ONLINE EMPLOYEE RECRUITMENT SYSTEM

A Project Report of Project - 2

Submitted By

ANJALI MANDOLIYA
(1713104023)

In partial fulfilment for the award of the degree
of

BACHELOR IN COMPUTER APPLICATION

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

Under the Supervision of

Mr. SHUBHAM KUMAR
Assistant Professor
May 2020

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report **" ONLINE EMPLOYEE RECRUITMENT SYSTEM "** is the bonafide work of **" ANJALI MANDOLIYA (1713104023) "** who carried out the project work under my supervision.

SIGNATURE OF HEAD	**SIGNATURE OF**
SUPERVISOR	
DR AVNEESH KUMAR**,**	Mr. SHUBHAM KUMAR,
, PhD (CS)	M-TECH
Associate Professor	Assistant Professor
School of Computing Science & Engineering	School of Computing Science &

Engineering

ABSTRACT

This project *Employee Recruitment System (ERS)* is a system in which jobseekers can register themselves online, view organization requirements and apply for the suitable job. Employee Recruitment System provides online help to the users all over the world. This kind of system plays an important role in simplifying the recruitment process. The system has facilities where prospective candidates can upload their CV and apply for jobs suited to them. It also makes it possible for organizations to post their staffing requirements and view profiles of interested candidates. Earlier recruitment was done manually and it was all at a time consuming work. Now it is all possible in a fraction of second. The system has been designed to do a whole lot more than just reduce paperwork. It can make a significant contribution to a company's marketing and sales activities. Employee recruitment systems make it possible for managers to access information that is crucial to managing their staff, which they can use for human resources management, staffing and planning activities. The primary purpose to develop this system is to optimize the recruitment process for an organization. Besides, the qualified applicants could be sorted by this system based on their qualifications and company requirements.

TABLE OF CONTENTS Page No.

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

1.1 Project Summary :

Employee Recruitment System by overall is aims to facilitate the applicant to apply for the job online. Indirectly, it is also to facilitate the managerial department of an organization for an optimized and systematic employee recruitment process.

A quick look at the overall trends in Online recruiting shows the rise in the importance of recruiting on the web site, online training, dawn of video interviews and emergence of professional Internet Recruiters. Online recruiting systems, with its emphasis on a more strategic decision making process is fast gaining ground as a popular outsourced function.

The system also provides the global platform for both - jobseekers and the organization, where the jobseekers can find their dream jobs and organization can find the right candidate to fulfill staff requirements.

This system has web-forms like registration form, login form and account pages like user applicant page, admin account page etc. It includes following main modules:-

- Admin module
- Candidate module

An administrator can be a manager who has full authority over the whole system. The administrator is able to update and retrieve data from the account of the candidate. The candidate is a center of this system. He has to register himself to use the services of the system.

1.2 Purpose:

The primary purpose to develop this system is to optimize the recruitment process for an organization. Besides, the qualified applicants could be sorted by this system based on their qualifications and company requirements.

The system has been designed to do a whole lot more than just reduce paperwork. It can make a significant contribution to a company's marketing and sales activities.

1.2 Project Scope:

Online Recruitment System enables the users to have the typical recruitment facilities and features at their disposal. It resolves typical issues of manual staffing processes and activities into a controlled and closely monitored work flow in the architecture of the application. This multi platform solution brings in by default, the basic intelligence and immense possibilities for further extension of the application as required by the user. The system makes it simpler to share and manage the organization's human resource requirements with higher efficiency and ease. The objective of these websites is to serve as a common meeting ground for jobseekers and organization, both locally and globally. This kind of system is specifically designed for organization to help in solving staffing problems and managing human resource department activities at a high degree of optimization.

1.3 Objectives:

- This software helps applicants to find suitable jobs within the organization and apply for that job easily.
- The software helps in managing and viewing details of interested applicants for the administrator.

- The system is capable of sorting and filtering best suitable candidates based on some criteria.

- Company will not have to waste his time finding the right employee at the right post.

1.4 TECHNOLOGY AND LITERATURE REVIEW

2.1 FRONT END (JAVA)

Overview of Java:

Java is a powerful but lean object oriented programming language. It has generated a lot of excitement because it makes it possible to program for the Internet by creating applets, programs that can be embedded in web pages. The context of an applet is limited only by one's imagination. Applets can be just little decorations to liven up web page, or they can be serious applications like word processors or spreadsheet.

But Java is more than a programming language for writing applets. It is becoming so popular that many people believe it will become a standard language for both general purposes and Internet programming.

Java from C++:

Java builds on the strength of C++. It has taken the best features of C++ and discarded the more problematic and error prone parts. To this, it has added garbage collection (automatic memory management), multi threading (the capacity for one program to do more than one thing at a time) and security capabilities. The result is that Java is simple, elegant, powerful and easy to use.

Java is actually a platform consisting of three components:

- Java Programming Language.
- Java library of classes and interfaces.
- Java virtual Machine.

But Java is more than a programming language for writing applets. It is becoming so popular that many people believe it will become a standard language for both general purpose and Internet programming.

Components of Java:

Java is actually a platform consisting of three components:

- Java Programming Language.
- Java Library of classes and interfaces.
- Java Virtual Machine.

Java is Object Oriented:

The Java programming language is object oriented, which makes program design focus on what you are dealing with rather than on how you are going to do something. This makes it more useful for programming in sophisticated projects because one can break the things down into understandable components. Reusability of these components is another big benefit.

Object oriented languages use the paradigm of classes. In simple terms, a class includes both the data and the functions to operate on that data. Object is an instance of the class that forms the actual run time entity of the class. Encapsulation of code and data makes it possible to make the changes in code without breaking other programs that use that code.

Java includes inheritance, or the ability to derive new classes from existing class referred to as the parent class. A subclass can add new data members to those inherited form the parent class. As far as methods are concerned, the subclass can reuse the inherited methods as it is, change them, and its own new methods.

Java's exciting features are:

- Ease in code correction.
- Garbage collection.
- Absence of pointers.
- Java is extensible.
- Java is secure.
- Java is robust.

- Java is multithreading.
- Simplicity.

Library Classes:

The Java platform includes an extensive class library so that programmers can use already existing classes, as it is, create subclasses to modify existing classes or implement interfaces and augment the capabilities of classes.

Classes contain data members i.e. fields and functions i.e. methods. In classes fields may be either variable or constant, and methods are fully implemented.

Interfaces:

Interfaces are also merely like classes. Interfaces also contain data members and functions. But the main difference is that in an interface, fields must be constants, and methods are just prototypes with no implementations. The prototype gives the method signature (the return type, the function name and the number of parameters with the type for each parameter), but the programmer must supply implementation. To use an interface, a programmer defines a class, declares that implements the interfaces, and then implements all the methods in that interface as the class.

The methods are implemented in a way that is appropriate for the class in which the methods are being used. Interface let one add functionality to a class and give a great deal of flexibility in doing it. In other words interfaces provide most of the advantages of multiple inheritances without its disadvantages.

Packages:

A package is a collection of related java classes and interfaces. The following list, gives examples of some java packages and what they cover.

JAVA.IO- Classes that manage reading data from input streams and writing data to the output streams.

JAVA.AWT- Classes that manage user interface components such as windows, dialog boxes, buttons, check boxes, lists, menus, scrollbars, and text fields; the 'AWT' stands Abstract Window Toolkit.

JAVA.APPLET- The applet class, which provides the ability to write applets, this package also includes several interfaces that connect an applet to its document and to resources for playing audio.

JAVA.AWT.EVENT- GUIs are event driven; it means they generate events when the user of the program interacts with the GUI.

JAVAX.SWING- This package enables the user to create interfaces which perform the GUI operations.

JAVA.SQL- The JDBC API, classes and interfaces that access databases and send SQL. In Java, packages serve as the basis for building other packages.

The Java Platform Builds in Security in Four Ways:

The way memory is allocated and laid out: In java an object's location in memory is not determined until the runtime, as opposed to C and C++, where the compiler makes memory layout decisions. As a result, a programmer cannot look at a class definition and figures out how it might be laid out in memory. Also since Java has no pointers, a programmer cannot forge pointers to memory.

The way incoming code is checked. The java virtual machine doesn’t trust any incoming code and subjects it to what is called ByteCode verification. The byte code verifier, part of the virtual machine, checks that

- The format of incoming code is correct
- Incoming code doesn’t forge pointers
- It doesn’t violate access restrictions
- It access objects as what they are

The way classes are loaded. The java bytecode loader, another Part of the virtual machine, checks whether classes loaded during program execution are local or from across a network. Imported classes cannot be substituted for built in classes cannot accidentally reference classes brought in over a network.

The way access is restricted for entrusted code. The java security manager allows users to restrict entrusted java applets so that they cannot access the local network, local files and other resources.

What is JDBC?

- JDBC is a java TM API for executing SQL statements.
- It consists of a set of classes and interfaces written in the java programming language that makes it easy to send SQL statements to virtually any relational databases.
- JDBC (Java Database Connectivity) is a front end tool for connecting servers to ODBC in that respect.
- JDBC is essentially a low-level application programming interface. It is called a low-level API since any data manipulation, storage and retrieval has to be done by the program itself. Some tools which provide a higher-level abstraction of, expected shortly.

The combination of java and JDBC lets a programmer write it once and run it anywhere.

Requirements to use JDBC:

- To use JDBC we need a basic knowledge of database and SQL.
- We need the jdk1.1 (Java Development Kit 1.1 available Java Soft's website) or a version of java since jdk1.1 and above come bundled with JDBC software.
- A back-end database engine for which a JDBC driver is available. When JDBC drivers area not available JDBC-ODBC bridge drivers are used to access the database through ODBC.

- Back-end is not needed when JDBC driver is capable of storing and retrieving the data itself, or if JDBC-ODBC bridge and the ODBC driver can be used to store and retrieve the information.

What does JDBC do?

JDBC makes it possible to do three things.

- Establishes the connection to database
- Send SQL statements
- Process the results.

JDBC is a low-level API and a base for Higher-level API. JDBC is a low-level interface, which means that it is used to invoke SQL commands directly. It works very well in this capacity and is easier to use than other to build higher-level interfaces and tools. A higher level interface such as JDBC. There are two kinds of higher-level APIs. • An embedded SQL for java and

- A direct mapping of relational database tables to java classes.

Java's Magig: The Byte Code

The key that allows java to solve both the security and the portability problems just described is that the output of a java compiler is not executable code. Rather, it is Byte code. Byte code is highly optimized set of instructions designed to be executed by the java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for Byte code. This may come as a bit of a surprise.

Translating a java program into Byte code helps makes easier to run a program in a wide variety of environments. The reason is straight forward; only the JVM needs to be implemented for each platform. Once the run-time package exists for a given system, any java program can run on it. Remember, although the details of the JVM will differ from platform to platform, all interrupt the same Java Byte Code.

JDBC Drivers:

The JDBC API founds in java.sql package, consists only a few concrete classes. Much of the API is distributed as database-neutral interface classes that specify behavior without providing any implementation. The actual implementations are provided by third-party vendors.

An individual database system is accessed via a specific JDBC driver that implements the java.sql.Driver interface. Drivers exist for nearly all popular RDBMS systems, though few are available for free. Sun bundles a free JDBC-ODBC bridge driver with the JDK to allow access to standard ODBC data sources, such as a Microsoft Access Database.

However, Sun advises against using the bridge driver for anything other than development and very limited deployment. Servlet developers in particular should need this warning because any problem in the JDBC-ODBC bridge driver's native code section can crash the entire server, not just your servlets.

JDBC drivers are available for most database platforms, from a number of vendors and in a number of different flavors. There are four driver categories:

Type1-JDBC-ODBC Bridge Driver

Type1 drivers use a bridge technology to connect a java client to an ODBC database service. Sun's JDBC-ODBC bridge is the most common Type1 driver. These drivers are implemented using native code.

Type2- Native-API Partly-Java Driver

Type2 drivers wrap this layer of java around database-specific native code libraries. For Oracle databases, the native libraries might be based on OCI (Oracle Call Interface) libraries, which were originally designed for C/C++ programmers. Because Type2 drivers are implementing using native code, in some cases they have better performance than their all-Java counterparts. They add an element of risk; however, because a defect in a driver's native code section can crash the entire server.

Type3- Net-Protocol All-Java Driver

Type3 drivers communicate via a generic network protocol to a piece of custom middleware. The middleware components might use any type of driver to provide the actual database access. WebLogic's Tengah product line is an example. These drivers are all java, which makes them useful for applet deployment and safe for servlet deployment.

Type4- Native-Protocol All-Java Driver

Type4 divers are the most direct of the lot. Written entirely in java, Type4 drivers understand database-specific networking protocols and can access the database directly without any additional software.

A list of currently available JDBC drivers can be found at

Getting a Connection

The first step in using a JDBC driver to get a database connection involves loading the specific driver class into the application's JVM. This makes the driver available later, when we need is for opening the connection. An easy way to load the driver class is to use the Class.forName () method:

Class.forName ("sun.jdbc.odbc_JdbcOdbcDriver"):

When the driver is loaded to memory, it registers itself with the java.sql.DriverManager class to open a connection to a given database, where the database is specified by a specially formatted URL. The method used to open the connection is DriverManager.GetConnection (). It returns a class that implements the java.sql.Connection interface:

A JDBC URL identifies an individual database in a driver-specific manner. Different drivers may need different information in the URL to specify the host database.

JDBC URLs usually begin with jdbc:sub protocol:subname during the call to get Connection(), the Driver Manager object asks each registered driver if it recognizes the URL. If a driver says yes, the driver manager uses that driver to create the connection object. Here is a snippet of code a servlet might use to load its database driver with the JDBC-ODBC Bridge and create an initial connection:

SERVLETS:

What are Java servlets?

Servlets are Java technology's answer to CGI programming. They are programs that run on a Web server and build Web pages. Building Web pages on the fly is useful (and commonly done) for a number of reasons:

The Web page is based on data submitted by the user.

For example the results pages from search engines are generated this way, and programs that process orders for E-comsitedo this as well.

The data changes frequently.

For example, a weather-report or news headlines page might build the page dynamically, perhaps returning a previously built page if it is still up to date.

The Web pages uses information from corporate databases or other such sources.

For example, you would use this for making a web page at On-line stores that lists current prices and number of items in stock.

What are the Advantages of Servlets Over "Traditional" CGI?

Java servlets are more efficient, easier to use, more powerful, more portable, and cheaper than traditional CGI and than many alternative CGI-like technologies.

- **Efficient:**

With traditional CGI, a new process is started for each HTTP request. If the CGI process does a relatively fast operation, the overhead of starting the process can dominate the execution time.

With servlets, the Java Virtual Machine stays up, and each request is handled by a lightweight Java Thread, not a heavyweight operating system process. Similarly, in traditional CGI, if there are N simultaneous request to the same CGI program, then the code for the CGI program is loaded into memory n times. With servlets, however, there are N threads but only a single copy of the servlet class. Servlets also have more alternatives than do regular CGI programs for optimizations such as caching previous computations, keeping database connections open, and the like.

- **Convenient:**

Hey, you are already known Java. Why learn Perl too? Besides the convenience of being able to use a familiar language, servlets have an extension infrastructure for automatically parsing and decoding HTML form data, reading and setting HTTP headers, handling cookies, tracking sessions, and many other such utilities.

- **Powerful:**

Java servlets let you easily do several things that are difficult or impossible with regular CGI. For one thing, servlets can talk directly to the Web server (regular CGI programs can't). This simplifies operations that need to look up images and other data stored in standard places. Servlets can also share data among each other, making useful thing like database connection pools easy to implement. They can also maintain information from request to request, simplifying things like session tracking and caching of previous computations.

- **Portable:**

Servlets are written in java and follows a well-standardized API. Consequently, servlets return for, I-Planet Enterprise Server can run virtually unchanged on Apache, Microsoft IIS, or Web Star. Servlets are supported directly or via plug-in on almost every major Web Server.

- **Inexpensive:**

There are a number of free or very inexpensive Web servers available that are good for “personal” use or low-level Web sites. However, with the major exception of Apache, which is free, most commercial-quality Web servers are relatively expensive. Nevertheless, once you have a Web server, no matter the cost server, adding servlet support to it (if it doesn’t come preconfigured to support servlets) is generally free or cheap.

What is JSP?

Java Serve Pages (JSP) is a technology that lets you mix regular, static HTML with dynamically-generated HTML. Many Web pages that are built by CGI programs are mostly static, with the dynamic part limited to a few small locations. But most CGI variations, including servlets, make you generate the entire page via your program, even though most of it is always the same. JSP lets you create the two parts separately. Here’s an example:

What are the Advantages of JSP?

- **Vs. Active Server Pages (ASP):**

ASP is a similar technology from Microsoft. The advantages of JSP are twofold. First, the dynamic part is written in Java not in Visual Basic or other MS-specific language. So it is more powerful and easier to use. Second, it portable to other operating systems and non-Microsoft web servers.

- **Vs. Pure Servlets:**

JSP doesn’t give you anything that you couldn’t in principle do with a servlet. But it is more convenient to write (and to modify!) regular HTML than to have a zillion println statements that generate the HTML. Plus, by separating the look from the content you can put different people on different tasks: your Web page design experts can build the HTML, leaving places for your servlet programmers to insert the dynamic content.

- **Vs. Server-Side Includes (SSI):**

SSI is a widely-supported technology for including externally-defined pieces into a static Web page. JSP is better because it lets you use servlets instead of a separate program to generate that dynamic part. Besides, SSI is really only intended for simple inclusions, not for “real” programs that use form data, make database connections, and the like.

- **Vs. JavaScript:**

JavaScript can generate HTML dynamically on the client. This is a useful capability, but only handles situations where the dynamic information is based on the client’s environment. With the exception of cookies, HTTP and form submission data is not available to JavaScript. And, since it runs on the client, JavaScript can’t access server-side resources like databases, catalogs, pricing information, and the like.

- **Vs. Static HTML:**

Regular, HTML, of course, cannot contain dynamic information. JSP is so easy and convenient that it is quite feasible to argument HTML pages that only benefit marginally by the insertion of small amounts of dynamic data. Previously, the cost of using dynamic data would preclude its use in all but the most valuable instances.

Basic Servlet Structure:

Here’s the outline of a basic servlet that handles GET requests. GET requests, for those unfamiliar with HTTP, are requests made by browsers when the types in a URL on the address line, follows a link from a web page, or makes an HTML form that does not specify a METHOD. Servlets can also very easily handle POST requests, which are generated when someone creates an HTML form that specifies METHOD=”POST”.

```
Import java.io.*;
Import javax.servlet.*;
Import javax.servlet.http.*;

Public class SomeServlets extends HttpServlet {
Public void doGet(HttpServletRequest request,
HttpServletResponse response) throws
ServletException.IOException {

//Use “request” to read incoming HTTP headers (e.g. cookies)
//and HTML form data (e.g. data the user entered and submitted)
//Use “response” to specify the HTTP response
line and headers // (e.g. specifying the content
type, setting cookies).

printWriter out=response.getWriter ();
//Use “out” to send content to browser
}
```

To be a servlet, a class should extend HttpServlet and override doGet or doPost (or both), depending on whether the data is being sent by GET or by POST. These methods take two arguments: an httpServletRequest and an HttpServletResponse. The HttpServletRequest has methods that let you find out about incoming information such as FORM data, HTTP request headers, and the like. The HttpServletResponse has methods that lets you specify the HTTP response line (200,404, etc.), response headers (ContentType, Set-Cookies, etc.), and, most importantly, lets you obtain a PrintWriter used to send output back to the client.

For simple servlets, most of the effort is println statements that generate the desired page. Note that doGet and doPost throw two exceptions, so you are required to include them in the declaration. Also note that you have to import classes in java.io (for printWriter, etc.), javax.servlet (for HttpServlet, etc.), and javax.servlet.http (for HttpServletRequest and HttpServletResponse). Finally, note that doGet and doPost are called by the service method, and sometimes you may want to override service directly, e.g. for a servlet that handles both GET and POST request.

Compiling and Installing the Servlet:

Note that the specific details for installing servlets vary from web server to web server. Please refer to your web server documentation for definitive directions. The online examples are running on Java Web Server (JWS) 2.0, where servlets are expected to be in a directory called Servlets in the JWS installation hierarchy. However, I placed this servlet in a separate package (hall) to avoid conflicts with other servlets on this server. You will want to be the same if you are using a web server that is used by other people and doesn't have a good infrastructure for "virtual servers" to prevent these conflicts automatically.

If you've never used packages before, there are two ways to compile classes that are in packages.

One way is to set your CLASSPATH to point to the directory above the actually containing your servlets. You can them compile normally from within the directory. For example, if your base directory is C:\JavaWebServer\servlets and your package name (and thus subdirectory name) is hall, and were on windows, you'd do:

```
DOS> set CLASSPATH=C: \JavaWebServlets; %CLASSPATH%
DOS> cd C:\JavaWebServer\servlets\hall
DOS> javac yourServlet.java
```

The first path, setting the CLASSPATH, you probably want to do permanently, rather than each time you start a new DOS window. On Windows 95/98 you'd typically put the "set CLASSPATH=……" Statement in your autoexec.bat file somewhere after the line that set the CLASSPATH to point to servlet.har and jsp.jar. On Windows NT, you'd go to the Start menu, select settings, select Control panel, select system, select Environment, then enter the variable and value. Note also that if your package were of the form name1.name2.name3 rather than simply name1 as here, you'd still have the CLASSPATH point to the top level directory of your package hierarchy (the one contain name1).

A Second way to compile classes that are in packages is to go to the directory above the one containing your servlets, and then do "javac directory\YourServlet.java" (UNIX; note the forward slash). For example, suppose again that your base directory is C:\JavaWebServer\Servlets and your package name (and thus subdirectory name) is hall, and you were on Windows. In that case, you'd do the following:

```
        DOS> cd C:\JavaWeb Server\Servlets
        DOS> javac hall\Yourservlet.java
```

Note that, on Windows, most JDK 1.1 versions of javac require a backslash, not a forward slash, after the directory name. This is fixed in JDK 1.1, many servlet authors stick with JDK 1.1.for portability.

Finally, another advanced option is to keep the source code in a location distinct from the .class files, and use javac's "-d" option to install them in the location the web server expects.

Running the Servlet:

With the Java web Sever, servlets are placed in the servlets directory within the main JWS installation directory, and are invoked via http://host/servlet/ServletName. Note that the directory is servlets, plural, while the URL refers to servlet, singular. Since this example was placed in the hall package, it would be invoked via http://host/servlet/hall.Helloworld. Other Web servers may have slightly different conventions on where to install servlets and how to invoke them. Most servers also let you define aliases for servlets. So that a servlet can be invoked via http://host/anypath/any-file.html. The process for doing this is completely server specific: check your server`s documentation for details.

A Servlet that generates HTML Most servlets generate HTML, not plain text as in the previous example. To do that, you need two additional steps: tell the browser that you are sending back HTML, and modify the println statements to build a legal Web page. The first step is done by setting the Content-Type response header. In general, headers can be set via the setHeader method of HttpServletResponse, but setting the content type is such a common task that there is also a special Setcontenttype method just for this purpose. Note that you need to set response headers before actually returning any of the content via the Printwriter. Here`s an example:

Maintenance Release of the Java Servlet 2.5 Specification:

Download the maintenance release of the Java Servlet Specification, version 2.5.

This version of Java Servlet technology.

Is included in the EE5 platform: Web Tier to go with Java EE5:

A Look at Resource Injection Read about the support for annotations by Java Web tier technologies and how they can simplify access to resources, environment data, and life-cycle control.

Form Processing Servlet:

This section shows how to

- Process form data
- Manage persistent data
- Use init parameters

The next Servlet that we are going to write provides a user interface to a mailing list through HTML forms. A user should be able to enter an email address in a text field and press a button to subscribe to the list or another button to unsubscribe.

The Servlet consists of two major parts:

- Data management
- Client interaction.

Data management:

The data management is rather straight-forward for an experienced Java Programmer. We use a java.lang.Vector object which contains the email addresses as Strings. Since a Servlet ca has data which persists between requests we load the address list only once, when the Servlet is initialized, and save it every time it has been changed by a request. An alternative approach would be keeping the list in memory while the

Servlet is active and writing it to disk in the destroy method. This would avoid the overhead of saving the address list after every change but is less fail-safe. If for some reason the address file can`t be written to disk or the server crashes and cannot destroy the Servlet, all changes to the list will be lost even though the users who submitted the requests to change the list received positive responses.

In init we first call super.init(config) to leave the ServletConfig management to the super class (HttpServlet),then we get the name of the address file from an init parameter(which is set up in the Web Server configuration). If the parameter is not available the Servlet throws a javax.servlet.UnavailableException (a subclass of javax.servletException) which indicates that a Servlet is temporarily (if a duration is specified) or permanently (as in this case) unavailable. Finally, the init method desterilizes the address file or creates an empty Vector if the address file does not exist yet. All exceptions that occur during the reserialization are transformed into Unavailable Exceptions.

The methods subscribe and unsubscribe are used to (un-)subscribe an address. They save the address list if it was modified by calling save () and return a Boolean success value. Note that these methods are both synchronized (on the Servlet object)to ensure the integrity of the address list, both, in memory and on disk.

The save method serializes the address list to the address file on disk which can be read again by init when the Servlet is restarted.

Client interaction:

The client interaction is handled by two of the standard HttpServlet methods, doGet and doPost.

- The doGet method replies to GET requests by sending an HTML page which contains the list of the currently subscribe or unsubscribe an address:
- The response content type is again set to text/html and the response is marked as not cacheable to proxy servers and clients (because it is dynamically created) by setting an HTTP header "pragma:no-cache". The form asks the client to use the POST method for submitting form data.
- Here is a typical output by this method:
- The doPost method receives the submitted form data, updates the address list and sends back a confirmation page:

Finally a confirmation page is sent with the usual method. Req.getRequestURI () is used to get the URI of the Servlet for a link back to the main page (which is created by doGet).

As usual, the Servlet extends javax.http.servlet.HttpServlet and overrides getServletInfo to provide a short notice. At last, here is the full source code of the ListManagerServlet.

3.2 BACK END (ORACLE)

Executing SQL Queries:

To really use a database, we need to have some way to execute queries. The simplest way to execute a query is to use the java.sql.Statement class. Statement objects are never instantiated directly; instead, a program calls the createStatement() method of Connection to obtain a new Statement object:

```
Statement stmt=con.crea:Statement();
```

A query that returns data can be executed using the executeQuery () method of Statement and returns a java.sql.ResultSet that encapsulates the retrieved data:

```
ResultSet rs-stmt.executeQuery(“SELECT * FROM CUSTOMERS”);
```

You can think of a ResultSet object as a representation of the query result returned one row at a time. You use the next() method of ResultSet to move from row to row. The ResultSet interface also boasts a multitude of methods designed for retrieving data from the current row. The getString () and getObject ()methods are among the most frequently used for retrieving column values:

```
while(rs.next()) {
String event=rs.getString(“even”);
Object count=(Integer)rs.getObject(“count”);
}
```

You should know that the ResultSet is linked to its parent Statement. Therefore, if a Statement is closed or used to execute another query, any related ResultSet objects are closed automatically.

Handling SQL Exceptions:

DBPhoneLookup encloses most of its code in a try/catch block. This block catches two exceptions: ClassNotFoundException and SQLException. The former is thrown by the Class.forName() method when the JDBC driver class can not be loaded. The latter is thrown by any JDBC method that has a problem. SQLException objects are just like any other exception type, with the additional feature that they can chain. The SQLException class defines an extra method, getNextException(), that allows the exception to encapsulate additional Exception objects. We didn’t bother with this feature in the previous example, but here’s how to use it:

```
Catch(SQLException e) {
out.println(e.getMessage());
while((e=e.getNextException())!=
null) { out.println(e.getMessage());
}
}
```

This code displays the message from the first exception and then loops through all the remaining exceptions, outputting the error message associated with each one. In practice, the first exception will generally include the most relevant information.

Results in Detail:

Before we continue, we should take a closer look at the ResultSet interface and related ResultSetMetaData interface. In Example9-1, we knew what our query looked like, and we knew what we expected to get back, so we formatted the output appropriately. But, if we want to display the results of a query in an HTML table, it would nice to have some Java code that builds the table automatically from the ResultSet rather than having to write the same loop-and-display code over and over. As an added bonus, this kind of code makes it possible to change the contents of the table simply by changing the query.

The ResultSetMetaData interface provides a way for a program to learn about the underlying structure of a query result on the fly. We can use it to build an object that dynamically generates an HTML table from a ResultSet, as shown in Example9-2. Many Java HTML generation tools have a similar capability.

Handling Null Fields:

Handling null database values with JDBC can be a little tricky(A database field can be set to null to indicate that no value is present, in much the same way that a Java object can be set to null). A method that does not return an object, like getInt(),has no way of indicating whether a column is null or whether it contains actual information. (Some drivers return a string that contains the text “null” when getString() is called on a null column!) any special value like -1, might be a legitimate value. Therefore, JDBC includes the wasNull() method in ResultSet, which returns true or false depending on whether the last column read was a true database null. This means that you must read data from the ResultSet into a variable, call was Null (), and proceed accordingly. It’s not pretty, but it works. Here’s an example:

```
int age=rs.getInt(“age”);
if(!rs.wasNull())
out.println(“Age:”+age);
```

Another way to check for null values is to use the getObject() method. If a column is null, getObject() always returns null. Compare this to the getString() method that has been known, in some implementations, to return the empty string if a column is null.

Using getObject() eliminates the need to call wasNull() and leads to simpler code.

Updating the Database:

Most database-enabled web sites need to do more than just perform dueries. When a client submits an order or provides some kind of information, the data needs to be entered into the database. When you know you’re executing a SQL UPDATE, INSERT, or DELETE statement and you know you don’t except a ResultSet, you can use the executeUpdate() method of statement. It returns a count that indicates the number of rows modified by the statement. It’s used like this:

```
int count= stmt.executeUpdate(“DELETE FROM CUSTOMERS
WHERE CUSTOMER_ID=5”)
```

If you are executing SQL, that may return either a ResultSet or a count (say, if you’re handling user-submitted SQL or building generic data-handling classes), use the generic execute() method of statement. It returns a Boolean whose value is true if the SQL statement produced one or more ResultSet objects or false if it resulted in an update count:

```
boolean b=stmt.execute(sql);
```

The getResultSet() and getUpdateCount() method of statement provide access to the results of the execute() method.

Using Prepared Statements:

A prepared statement object is like a regular statement object, in that it can be used to execute SQL statements. The important difference is that the SQL in a PreparedStatement is precompiled by the database for faster execution. Once a Prepared Statement has been compiled, it can still be customized by adjusting predefined parameters. Prepared statements are useful in applications that have to run the same general SQL command over and over.

Use the preparedStatement (String) method of connection to create preparedStatement objects. Use the? Character as a placeholder for values to be substituted later. For example:

```
PreparedStatement pstmt = con.preparedStatement
(“INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID.TOTAL)
VALUES (?,?,?)”);
INSERT INTO MUSKETEERS (NAME) VALUES (‘John
d'Artagnan’)
```

As you see, the string terminates twice. One solution is to manually replace the single quote’ with two single quotes”, the Oracle escape sequence for one single quote. This solution requires you to escape every character that your database treats as special—not an easy task and not consistent with writing platform-independent code. A far better solution is to use a preparedStatement and pass the string using its setString() method, as shown below. The preparedStatement automatically escapes the string as necessary for your database:

```
PreparedStatement pstmt = con.preparedStatement
(“INSERT INTO MUSKETEERS (NAME) VALUES (?)”)  ;
Pstmt.setString (1,”John d'Artagnan”);
Pstmt.executeUpdate ();
```

PROJECT MANAGEMENT

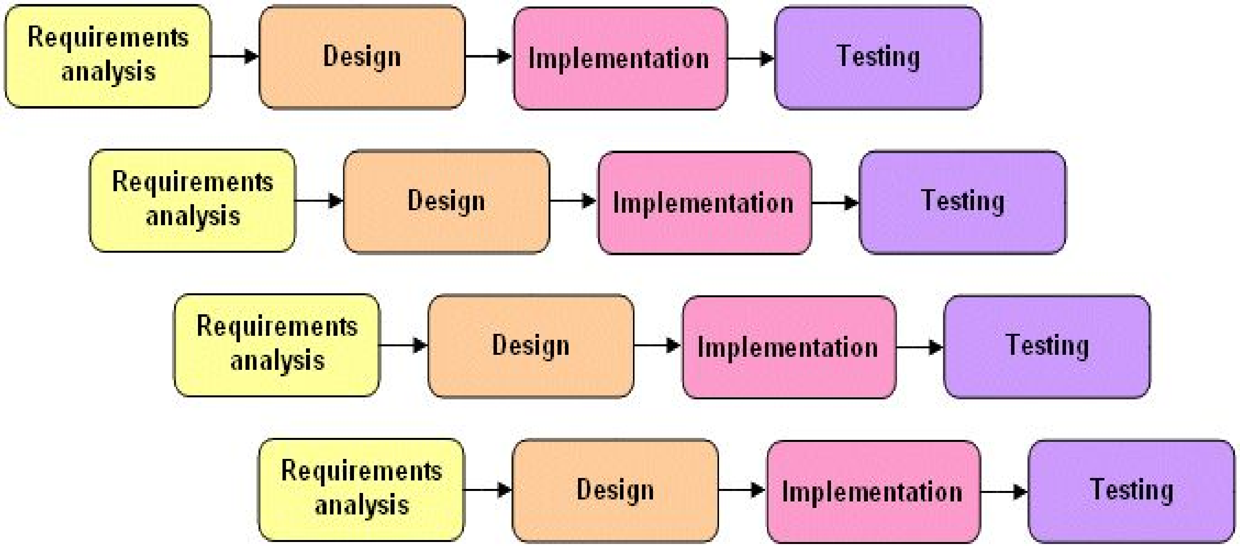
2.1 PROJECT PLANNING

Project planning includes description of the project task, activity and function, dependencies, resource requirements and detail schedules. Project planning involves estimating how much time, effort, money and resources will be required to build a specific software system.

2.1.1 Project Development Approach and Justification

A Software Process model is a simplified abstract representation of a software process, which is presented from a particular perspective. Planning prepares a framework that makes a reasonable estimate of the project. To accomplish it, software development models are used. Incremental models are used to satisfy this purpose.

Incremental Model:



The system uses this method of software development where the model is analyzed, designed, tested and implemented. In incremental models, systems are developed in various units. These units entail the requirement planning, development and test phases. This module is useful for big and small products. It is divided into units. It satisfies the customer’s needs.

The Incremental Model combines elements of the linear sequential model with the iterative philosophy of prototyping. The incremental model applies linear sequences in a staged fashion as calendar time progresses. Each linear sequence produces a deliverable “increment” of the software. In the incremental model the first increment is called core product. In core products basic requirements are added but some unknown supplementary features remain undelivered. This core product is used by customers to evolve the system and next increment is planned to develop.

During the first increment analysis phase, customers and developers specify as many requirements as possible and prepare documentation. Now a first version of product with minimal and essential feature is launched to market. Based on the feedback and experience with this version, list of additional features are added. This process is repeated following the delivery of each increment, until the complete product is produced.

With this approach first model may be available within few weeks or months. In this model, less cost and time is required to develop first increment called core product. Less risk is occurred to develop the smaller systems represented by the increments. Incremental funding is allowed, means only one or two increments might be funded when the program starts. It can results in better testing, because testing each increment is likely to be easier than testing entire system. The feedback providing at each increment is useful for determining the final requirement of system.

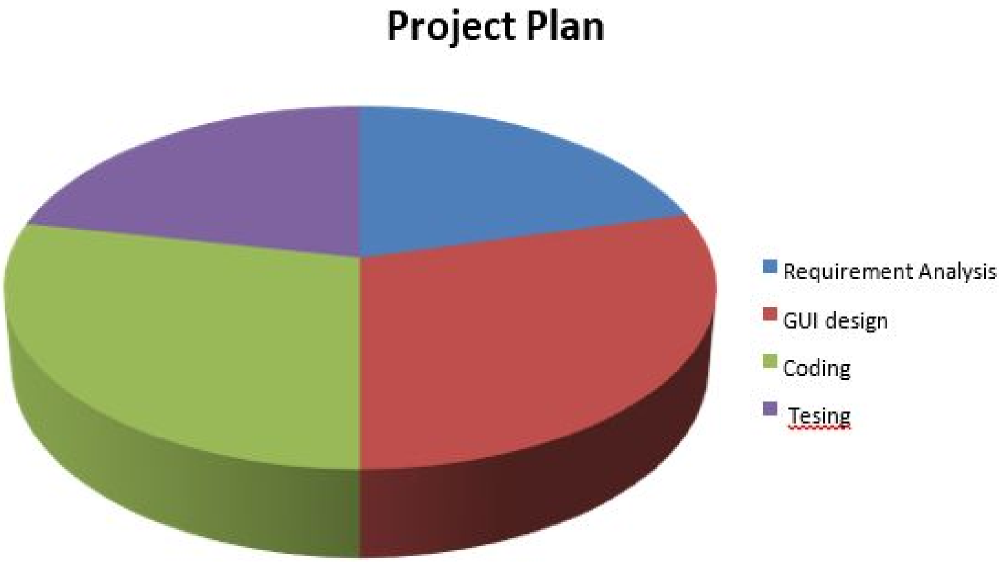
2.1.2 PROJECT PLAN

Before proceeding with the development of the Project, it is inevitable to plan its development right from the start till the end.

Task	From Date	To Date
Requirement Analysis	01/01/2020	22/01/2020

GUI design	23/01/2020	05/03./2020
Coding	06/03/2020	11/04/2020
Testing and debugging	12/04/2020	20/04/2020

Table 2.1 : Project Plan



2.1.3 MILESTONE AND DELIVERABLES

Milestones and Deliverables are the part of the Project Scheduling. In what time your Project is going to be ready, is known by Milestones. Milestone is an endpoint of the software process activity.

Software Project Activity	Milestone
Project Plan	Project schedule
Requirement Collection	User requirements

Data flow analysis	System Flow
Design 1. Database Design 2. GUI	System Design Document
Implementation 1. Code for giving security 2. Code for reports	Access rights Reports
Testing	Setting validations and error messages

Table2.2 : **MILESTONE AND DELIVERABLES**

SYSTEM REQUIREMENT

3.1 USER CHARACTERISTICS:

Three types of user interact with the system:

1. COMPANY (HR Department):

The following are the acts performed by the HR department of company throughout the system:

- Project Management
- Contact Management
- Team Member and Team Management
- Partner Management
- Data Management
- Activity Management
- Feedback Management

2. ADMINISTRATOR:

Admin can update, delete, modify the detail of the candidates which are filled by them only of their respective department..He also can schedule examination activities and sort candidates basis on exam results.

3. JOBSEEKER:

Jobseeker can register himself, upload CVs, find the appropriate job within organization's vacancy constraints, attend the exam and give feedback about the system.

3.2 HARDWARE AND SOFTWARE REQUIREMENT SPECIFICATION

Server Side:

Hardware

- Processor : Intel core processor I3
- RAM : 4 GB RAM
- Hard Disk : 80 GB HDD
- Monitor : Compatible Printing Device

Software

- Operating System : Microsoft Windows 7/8/8.1/9/10
- Package : Adobe Dreamweaver
- Database : MY SQL
- Language: java
- Design : Adobe Photoshop CS6.0 , Macromedia Flash Player
- Browser : IE (Version 6 or higher) , Mozilla Firefox or Google Chrome

Client side:

Hardware

- Processor : Intel core i3 or higher processor 2 GHz
- RAM : 520 MB RAM
- Hard Disk : 40 GB HDD
- Monitor : Compatible Printing Device
- Keyboard : Any Keyboard

Software

- netbeans Server

- MYSQL Server
- IE (Version 6 or higher)

3.3 CONSTRAINTS:

3.3.1 Hardware Limitations

The limitation of dream viewer is that it requires RAM that cannot be less than 520 MB and the processor cannot be less than 2 GHz speed as recommended in the hardware requirements.

3.3.2 Reliability Requirements

The main reliability requirement is the validation used. Without proper validation the system does not allow to enter that value into the database. All the required validation controls are kept controls are kept to keep the system secure.

The following are the some of the reliability requirements:

In the email ID the user cannot enter any dummy value, the validation checks that whether there is a '@' or '_' symbol in that.

- Any null value is not allowed in place of compulsory fields.
- In numeric field user cannot enter any character value.
- In date of birth, user cannot enter date and time other than given format

Entered password and confirm password must match to each other.

- User can not re –register an account on his primary email. **3.3.3 Safety and Security Considerations**

Safety:

The source of this software will be kept at more than one place with user ID, password and also in CD ROM in case of server failure.

Security:

Security in this software provide to different user in different ways by giving different user id. If user is admin, he has all the privileges and constraints. He can access the entire database. He can change or delete database from other user's accounts. HR department have limited access according to their role. Because of limited privileges one of them cannot update other details of the candidate.

3.4 Assumption and Dependencies:

- End user is the person having enough knowledge for the project operation.
- Only admin have all the privileges.
- Candidate can only fill the registration form.
- Candidate cannot apply for job after deadlines.

3.5 REQUIREMENTS of A NEW SYSTEM:

3.5.1 Functional requirements:

- The system should record all the details of an applicant.
- The system should provide applicant to edit his profile details.
- The system should allow user to give feedback about the system.
- The system should have the facility of administrative help for user to solve his important problems.
- The system should allow admin to have full authority over user accounts.
- The system should allow admin to display vacancy within thw organization.
- The system should allow admin to sort and filter applicants based on some criteria.
- The system should allow admin to send notifications about upcoming events and deadlines to applicants.
- Username & password are sent to the users via mobile sms and email after registration.
- Password recovery system is also provided in case of forgetting the password.

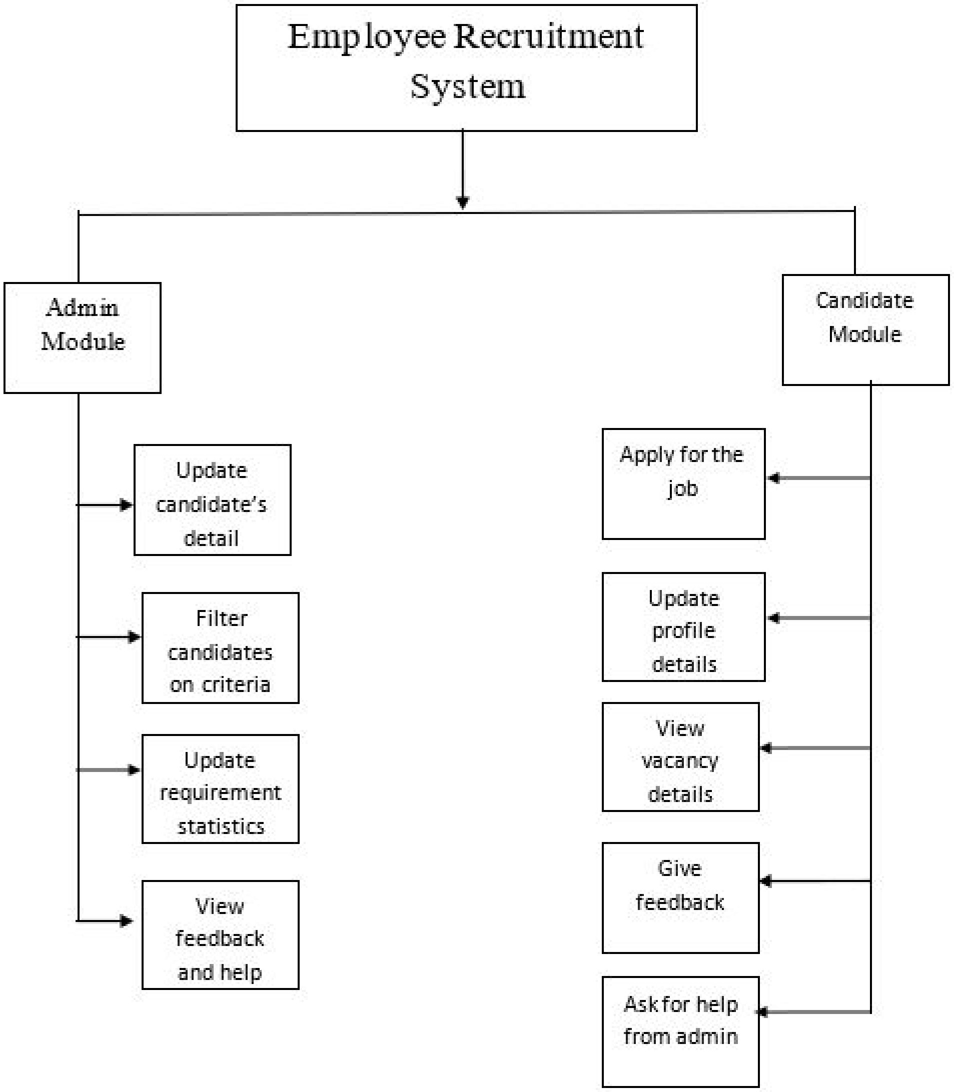
3.5.2 Non functional requirements:

- This application is secure for every kind of its users, because here is facility of session management. If any user logout from any session then nobody will be able to access his profile without knowing his confidential password.
- The database used here is robust, reliable & fast. So users will have to wait for the output for a very short time.
- This application can be accessed from any type of platform.

- There is no case of redundancy in the database so it will not take extra memory space.

SYSTEM ANALYSIS

STRUCTURE DIAGRAM:



4.2 OOPS DIAGRAMS

4.2.1 Use Case Diagram.

Use case diagrams are behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

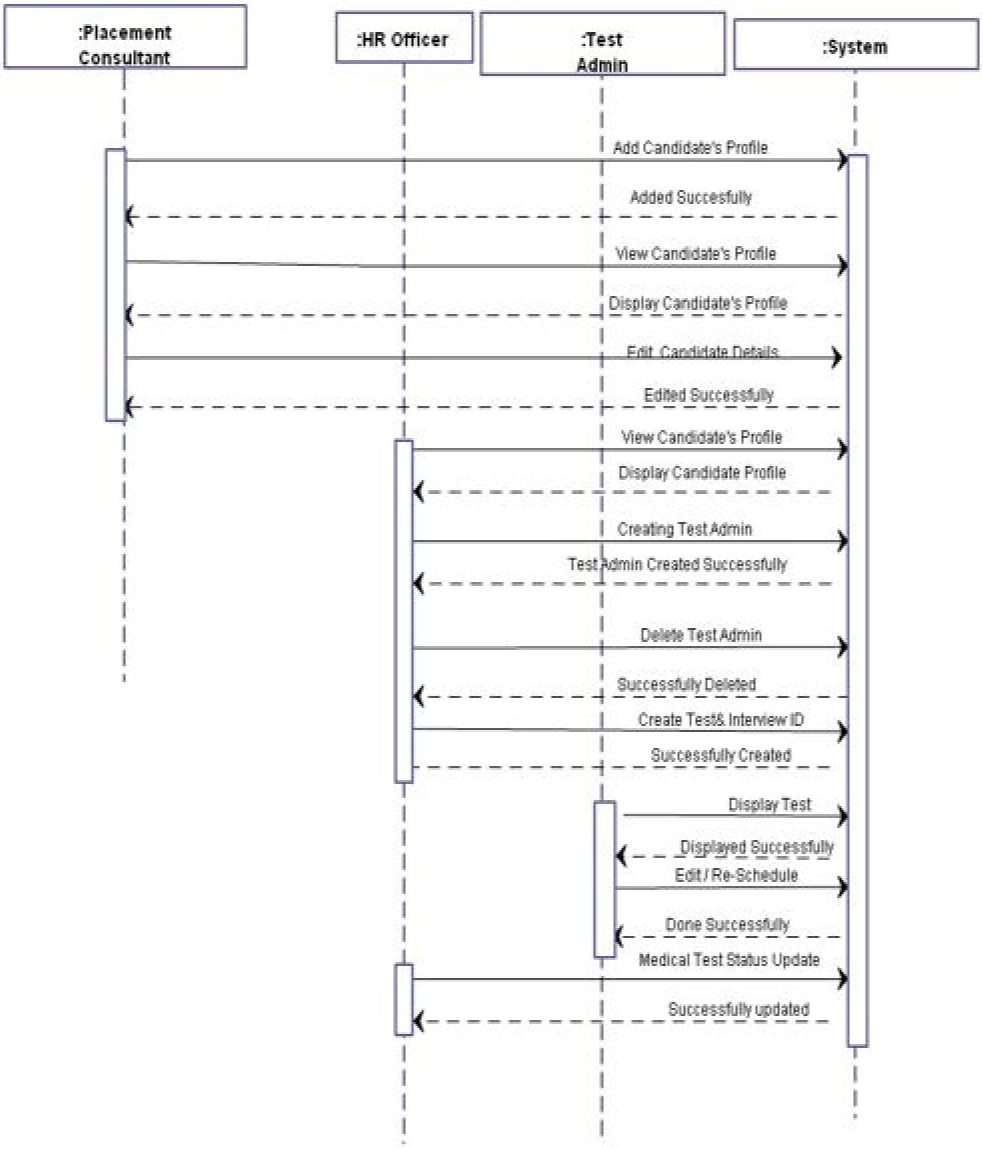
4.2.2 Sequence Diagram

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios

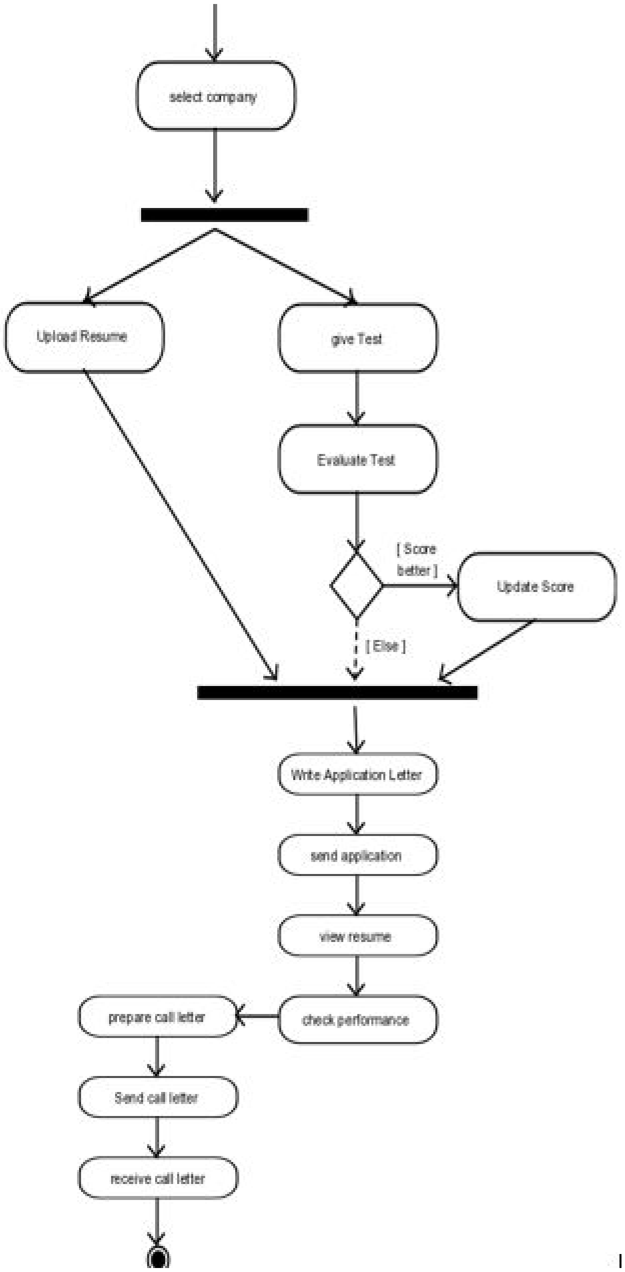
4.2.3 Activity Diagram

Activity diagrams are the object-oriented equivalent of flow charts and data-flow diagrams from structured development. It describes the workflow behavior of a system. The process flows in the system are captured in the activity diagram. Activity diagram illustrates the dynamic nature of a system by modeling the flow of control from Activity to activity.

USE CASE DIAGRAM:



ACTIVITY DIAGRAM:



4.3 Data Dictionary

- Data Dictionary is catalog – a repository of the elements in the systems.
- This element centre data and way they are structure to meet user requirements and organization needs.

- Data Dictionary contains list of all elements composing the following through a system.

- The major element of data flow, data stores and processes.
- The Data Dictionary stores details and description of this element.

Need of Data Dictionary:

- To manage details in a large system.
- To communicate a common meaning for all system elements.
- To document the futures of the system.
- To facilitate analysis of the details in order to evaluate characteristics and to determine where system should be made.
- To locate errors and omission in the system.

Table: Personal Details

Field	**Data Type**	**Size**	**Constraints**	**Description**
User id	int	100	Primary Key Auto Increment	It is the unique id of user.
First name	varchar	255		First name of user
Middle name	varchar	255		Middle name of user
Last name	varchar	255		last name of user
Gender	varchar	255		Male/female
Birth date	date			Date of birth of user

state	varchar	255		Current state of user
city	varchar	255		Current city of user

Table: Academic Details

Field	Data Type	Size	Constraints	Description
User id	int	100	Primary Key Auto Increment Foreign key	It is the unique id of the user.
university	varchar	255		University of user
institute	varchar	255		Institute of user
branch	varchar	255		Branch in which studying/studied
degree	varchar	255		BE/B.TECH/ ME/M.TECH/ BBA/MBA
status	varchar			Pursuing / Completed
CPI	float	255		CPI or grade achieved
semester	int	255		(If pursuing) – current semester
experience	int	255		(If completed) - experience

Table: Account Details

Field	Data Type	Size	Constraints	Description
User id	int	100	Primary Key Auto Increment Foreign key	It is the unique id of user.
Post	varchar	255		Post for which user is applying
Primary email	varchar	255	Unique key	Primary email – will be used for login
Secondary email	varchar	255	Unique key	Secondary email – can be used for account recovery
Password	varchar	255		Password defined by user

Table: Admin Details

Field	Data Type	Size	Constraint	Description
Admin id	int	100	Primary Key	It is the unique id of admin.
email	varchar	255	Unique key	email – will be used for login

password	int	255		Password defined by admin

Table: Requirement Statistics

Field	Data Type	Size	Constraints	Description
Post name	varchar	255		Post available in organization
vacancies	int	255		Vacancy for particular post within the company
Required experience	int	255		Required minimum experience to apply for the job
Min salary	int	255		Minimum salary for particular post
Max salary	int	255		Maximum salary for particular post

CODING

```
package Interface;

import static java.lang.Thread.sleep;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javafx.scene.control.RadioButton;
import javax.swing.ButtonModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author ccs
 */
public class Home extends javax.swing.JFrame {
```

```
/**
 * Creates new form Home
 */

Connection conn=null;
PreparedStatement pst=null;
ResultSet rs=null;
String sqr;

String epnumber;
String name;
String gender;
String nic;
String address;
String contactNumber;
String age;
String distric;

// O/L data
String olexamNumber;
String olas;
String olbs;
String olcs;
String olss;
String olmaths;
String olenglish;
```

```
// A/L data
String alexamNumber;
String alas;
String albs;
String alcs;
String alss;
String stream;

// advance
String degreeType;
String degree;
String duration;
String instute;

// experiance
String expostType;
String exmonth;
String exorg;



// shedule


String shpost;
String shjobtype;
String shgender;
String shagemin;

String shagemax;

int sholcheck=0;

String sholas;

String sholbs;

String sholcs;

String sholss;

int shalcheck=0;

String shalstream;

String shalas;

String shalbs;

String shalcs;

String shalss;

int shdegcheck=0;

String shdegtype;

String shdegduration;

String shdegcategory;

String sholmaths;

String sholenglish;


int olcheck=0;

int alcheck=0;

int degcheck=0;
int olvalue=0;

int alvalue=0;

int degvalue=0;

int agemin=0;

int agemax=0;

String selectolmath;

String selectolenglish;

String selectalstream;

String selectdegtype;





public Home() {

    initComponents();

    conn=DBConnect.connect();

    personalDataTablelord();

    olDataTablelord();

    alDataTablelord();

    aDDataTablelord();

    shDataTablelord();

    postpelord();

    olLoadDisable();

    alLoadDisable();

    degLoadDisable();

    exDataTablelord();
```

```
        empbasicLoad();
        clock();



    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        buttonGroup1 = new javax.swing.ButtonGroup();
        jLabel62 = new javax.swing.JLabel();
        jButton20 = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jLabel64 = new javax.swing.JLabel();
        lbltime = new javax.swing.JLabel();
        lbldate = new javax.swing.JLabel();
        jTabbedPane1 = new javax.swing.JTabbedPane();
        jPanel4 = new javax.swing.JPanel();
```

```
jLabel61 = new javax.swing.JLabel();

jPanel5 = new javax.swing.JPanel();

jTabbedPane2 = new javax.swing.JTabbedPane();

jPanel6 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

txtpdepnumber = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

jLabel9 = new javax.swing.JLabel();

txtpdname = new javax.swing.JTextField();

rbtmale = new javax.swing.JRadioButton();

rbtfemale = new javax.swing.JRadioButton();

txtpdnic = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

txtadress = new javax.swing.JTextArea();

txtpdcontactnumber = new javax.swing.JTextField();

txtpdage = new javax.swing.JTextField();

cmbpdcity = new javax.swing.JComboBox();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jScrollPane2 = new javax.swing.JScrollPane();
```

```
tblpd = new javax.swing.JTable();
txtpdsearch = new javax.swing.JTextField();
jLabel58 = new javax.swing.JLabel();
jPanel12 = new javax.swing.JPanel();
jLabel6 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
txtolname = new javax.swing.JTextField();
txtolas = new javax.swing.JTextField();
txtolbs = new javax.swing.JTextField();
txtolcs = new javax.swing.JTextField();
cmbolmaths = new javax.swing.JComboBox();
jButton4 = new javax.swing.JButton();
jButton5 = new javax.swing.JButton();
jButton6 = new javax.swing.JButton();
jScrollPane4 = new javax.swing.JScrollPane();
tblol = new javax.swing.JTable();
txtolsearch = new javax.swing.JTextField();
txtolexamnumber = new javax.swing.JTextField();
txtolss = new javax.swing.JTextField();
jLabel17 = new javax.swing.JLabel();
```

```
cmbolenglish = new javax.swing.JComboBox();
jPanel13 = new javax.swing.JPanel();
jLabel18 = new javax.swing.JLabel();
txtalepnumber = new javax.swing.JTextField();
jLabel19 = new javax.swing.JLabel();
jLabel20 = new javax.swing.JLabel();
jLabel21 = new javax.swing.JLabel();
jLabel22 = new javax.swing.JLabel();
jLabel23 = new javax.swing.JLabel();
jLabel24 = new javax.swing.JLabel();
txtalname = new javax.swing.JTextField();
txtalas = new javax.swing.JTextField();
txtalbs = new javax.swing.JTextField();
jButton7 = new javax.swing.JButton();
jButton8 = new javax.swing.JButton();
jButton9 = new javax.swing.JButton();
jScrollPane5 = new javax.swing.JScrollPane();
tblal = new javax.swing.JTable();
txtalsearch = new javax.swing.JTextField();
txtalexnumber = new javax.swing.JTextField();
txtalcs = new javax.swing.JTextField();
cmbalstream = new javax.swing.JComboBox();
jLabel25 = new javax.swing.JLabel();
txtalss = new javax.swing.JTextField();
jPanel14 = new javax.swing.JPanel();
jLabel26 = new javax.swing.JLabel();
```

```
txtadepnumber = new javax.swing.JTextField();
jLabel27 = new javax.swing.JLabel();
jLabel28 = new javax.swing.JLabel();
jLabel29 = new javax.swing.JLabel();
jLabel30 = new javax.swing.JLabel();
jLabel31 = new javax.swing.JLabel();
txtadname = new javax.swing.JTextField();
jButton10 = new javax.swing.JButton();
jButton11 = new javax.swing.JButton();
jButton12 = new javax.swing.JButton();
jScrollPane6 = new javax.swing.JScrollPane();
tblad = new javax.swing.JTable();
txtadsearch = new javax.swing.JTextField();
txtaddegree = new javax.swing.JTextField();
cmbaddegtype = new javax.swing.JComboBox();
txtadinstute = new javax.swing.JTextField();
cmbduration = new javax.swing.JComboBox();
jPanel15 = new javax.swing.JPanel();
jLabel33 = new javax.swing.JLabel();
txtexepnumber = new javax.swing.JTextField();
jLabel34 = new javax.swing.JLabel();
jLabel35 = new javax.swing.JLabel();
jLabel36 = new javax.swing.JLabel();
jLabel37 = new javax.swing.JLabel();
txtexname = new javax.swing.JTextField();
jButton13 = new javax.swing.JButton();
```

```
jButton14 = new javax.swing.JButton();
jButton15 = new javax.swing.JButton();
jScrollPane7 = new javax.swing.JScrollPane();
tblex = new javax.swing.JTable();
txtexsearch = new javax.swing.JTextField();
cmbpostType = new javax.swing.JComboBox();
txtexmonth = new javax.swing.JTextField();
txtexorg = new javax.swing.JTextField();
jPanel16 = new javax.swing.JPanel();
jLabel38 = new javax.swing.JLabel();
txtshpost = new javax.swing.JTextField();
jLabel39 = new javax.swing.JLabel();
jLabel40 = new javax.swing.JLabel();
jLabel41 = new javax.swing.JLabel();
jLabel42 = new javax.swing.JLabel();
jLabel43 = new javax.swing.JLabel();
jLabel44 = new javax.swing.JLabel();
jLabel45 = new javax.swing.JLabel();
txtshjobtype = new javax.swing.JTextField();
txtshagemax = new javax.swing.JTextField();
txtshagemin = new javax.swing.JTextField();
jButton16 = new javax.swing.JButton();
jButton17 = new javax.swing.JButton();
jButton18 = new javax.swing.JButton();
jScrollPane8 = new javax.swing.JScrollPane();
tblshedule = new javax.swing.JTable();
```

```
txtshsearch = new javax.swing.JTextField();

txtsholss = new javax.swing.JTextField();

txtsholcs = new javax.swing.JTextField();

txtsholas = new javax.swing.JTextField();

txtsholbs = new javax.swing.JTextField();

jLabel46 = new javax.swing.JLabel();

jLabel47 = new javax.swing.JLabel();

jLabel48 = new javax.swing.JLabel();

jLabel49 = new javax.swing.JLabel();

txtshalas = new javax.swing.JTextField();

txtshalbs = new javax.swing.JTextField();

jLabel50 = new javax.swing.JLabel();

jLabel51 = new javax.swing.JLabel();

txtshalcs = new javax.swing.JTextField();

jLabel52 = new javax.swing.JLabel();

txtshalss = new javax.swing.JTextField();

jLabel53 = new javax.swing.JLabel();

cmbshalstream = new javax.swing.JComboBox();

jLabel54 = new javax.swing.JLabel();

jLabel55 = new javax.swing.JLabel();

jLabel56 = new javax.swing.JLabel();

jLabel57 = new javax.swing.JLabel();

cmbshdegcategory = new javax.swing.JComboBox();

cmbshdegreeduration = new javax.swing.JComboBox();

cmbshgdegreetype = new javax.swing.JComboBox();

jButton19 = new javax.swing.JButton();
```

```
        cmbshgender = new javax.swing.JComboBox();
        jLabel32 = new javax.swing.JLabel();
        jLabel60 = new javax.swing.JLabel();
        txtsholmaths = new javax.swing.JTextField();
        txtsholenglish = new javax.swing.JTextField();
        chkol = new javax.swing.JCheckBox();
        chkal = new javax.swing.JCheckBox();
        chkdeg = new javax.swing.JCheckBox();
        jPanel17 = new javax.swing.JPanel();
        jLabel59 = new javax.swing.JLabel();
        jScrollPane9 = new javax.swing.JScrollPane();
        tblselectemp = new javax.swing.JTable();
        jTextField47 = new javax.swing.JTextField();
        cmbsepost = new javax.swing.JComboBox();

        jLabel62.setText("jLabel62");

        jButton20.setText("jButton20");


    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBackground(new java.awt.Color(255, 255, 255));
        jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jPanel2.setBackground(new java.awt.Color(0, 185, 205));
        jLabel64.setFont(new java.awt.Font("Sitka Heading", 3, 30)); // NOI18N

        jLabel64.setForeground(new java.awt.Color(255, 255, 255));

        jLabel64.setText("Employee Recruitment Management System");


        lbltime.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

        lbltime.setForeground(new java.awt.Color(255, 255, 255));

        lbltime.setText("Date");


        lbldate.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

        lbldate.setForeground(new java.awt.Color(255, 255, 255));

        lbldate.setText("Time");


        javax.swing.GroupLayout jPanel2Layout = new
    javax.swing.GroupLayout(jPanel2);

        jPanel2.setLayout(jPanel2Layout);

        jPanel2Layout.setHorizontalGroup(


    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
    G)

          .addGroup(jPanel2Layout.createSequentialGroup()

            .addContainerGap()

            .addComponent(jLabel64,
    javax.swing.GroupLayout.PREFERRED_SIZE, 599,
    javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(18, 18, 18)

            .addComponent(lbltime)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 55, Short.MAX_VALUE)

                .addComponent(lbldate)

                .addGap(53, 53, 53))

        );

        jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(lbltime)

                .addComponent(lbldate))

            .addComponent(jLabel64, javax.swing.GroupLayout.PREFERRED_SIZE, 40, javax.swing.GroupLayout.PREFERRED_SIZE)

        );


        jPanel1.add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 800, 40));


        jTabbedPane1.setBackground(new java.awt.Color(255, 255, 255));

        jTabbedPane1.setFont(new java.awt.Font("Sitka Heading", 1, 14)); // NOI18N


        jPanel4.setBackground(new java.awt.Color(255, 255, 255));


        jLabel61.setFont(new java.awt.Font("Sitka Heading", 3, 36)); // NOI18N

        jLabel61.setText("Right Job Candidate For Right Vacancy");
        javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);

        jPanel4.setLayout(jPanel4Layout);

        jPanel4Layout.setHorizontalGroup(


jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)

            .addGroup(jPanel4Layout.createSequentialGroup()

                .addContainerGap()

                .addComponent(jLabel61,
javax.swing.GroupLayout.PREFERRED_SIZE, 674,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addContainerGap(111, Short.MAX_VALUE))

        );

        jPanel4Layout.setVerticalGroup(


jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel4Layout.createSequentialGroup()

                .addComponent(jLabel61,
javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addContainerGap(323, Short.MAX_VALUE))

        );


        jTabbedPane1.addTab("HOME", jPanel4);
```

```
    jPanel5.setBackground(new java.awt.Color(255, 255, 255));
    jPanel5.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jTabbedPane2.setFont(new java.awt.Font("Sitka Heading", 1, 12)); // NOI18N

    jPanel6.setBackground(new java.awt.Color(255, 255, 255));
    jPanel6.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jLabel1.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel1.setText("EP Number");
    jPanel6.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 23, -1, -1));

    txtpdepnumber.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 185, 205)));
    jPanel6.add(txtpdepnumber, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 20, 171, -1));

    jLabel2.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel2.setText("Name");
    jPanel6.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 54, -1, -1));

    jLabel3.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel3.setText("Gender");
    jPanel6.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 82, -1, -1));
    jLabel4.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

    jLabel4.setText("NIC");

    jPanel6.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
111, -1, -1));


    jLabel5.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

    jLabel5.setText("Addess");

    jPanel6.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
139, -1, -1));


    jLabel7.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

    jLabel7.setText("Contact Number");

    jPanel6.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
220, -1, -1));


    jLabel8.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

    jLabel8.setText("Age");

    jPanel6.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
250, -1, -1));


    jLabel9.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

    jLabel9.setText("Distric");

    jPanel6.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
280, -1, -1));


    txtpdname.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
```

```
        jPanel6.add(txtpdname, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 50, 171, -1));


        buttonGroup1.add(rbtmale);

        rbtmale.setText("Male");

        jPanel6.add(rbtmale, new org.netbeans.lib.awtextra.AbsoluteConstraints(120,
80, 71, -1));


        buttonGroup1.add(rbtfemale);

        rbtfemale.setText("Female");

        jPanel6.add(rbtfemale, new
org.netbeans.lib.awtextra.AbsoluteConstraints(200, 80, -1, -1));


        txtpdnic.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel6.add(txtpdnic, new org.netbeans.lib.awtextra.AbsoluteConstraints(120,
110, 171, -1));


        txtadress.setColumns(20);

        txtadress.setRows(5);

        txtadress.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jScrollPane1.setViewportView(txtadress);


        jPanel6.add(jScrollPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 140, 170, 70));
```

```
txtpdcontactnumber.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel6.add(txtpdcontactnumber, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 220, 170, -1));


        txtpdage.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel6.add(txtpdage, new org.netbeans.lib.awtextra.AbsoluteConstraints(120,
250, 170, -1));


        cmbpdcity.setModel(new javax.swing.DefaultComboBoxModel(new String[]
{ "Item 1", "Item 2", "Item 3", "Item 4" }));
        cmbpdcity.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel6.add(cmbpdcity, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 280, 170, -1));


        jButton1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/delete-16 (1).png"))); //
NOI18N
        jButton1.setText("Delete");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        jPanel6.add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
310, 85, -1));
```

```
        jButton2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/available-updates-16
(1).png"))); // NOI18N
        jButton2.setText("Update");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });
        jPanel6.add(jButton2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 310, 90, -1));

        jButton3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/add-16.png"))); // NOI18N
        jButton3.setText("Add");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });
        jPanel6.add(jButton3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(200, 310, 85, -1));

        tblpd.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        tblpd.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {},
        {},
        {},
        {}
      },
      new String [] {


      }
    ));
    tblpd.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_OFF);
    tblpd.addMouseListener(new java.awt.event.MouseAdapter() {
      public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblpdMouseClicked(evt);
      }
    });
    jScrollPane2.setViewportView(tblpd);

    jPanel6.add(jScrollPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(300, 70, 462, 280));

    txtpdsearch.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
    txtpdsearch.addKeyListener(new java.awt.event.KeyAdapter() {
      public void keyReleased(java.awt.event.KeyEvent evt) {
        txtpdsearchKeyReleased(evt);
      }
    });
```

```
        jPanel6.add(txtpdsearch, new
org.netbeans.lib.awtextra.AbsoluteConstraints(592, 20, 183, -1));


        jLabel58.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel58.setText("EP Number");
        jPanel6.add(jLabel58, new
org.netbeans.lib.awtextra.AbsoluteConstraints(530, 20, -1, -1));


        jTabbedPane2.addTab("Personal Data", jPanel6);


        jPanel12.setBackground(new java.awt.Color(255, 255, 255));
        jPanel12.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());


        jLabel6.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel6.setText("EP Number");
        jPanel12.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
23, -1, -1));


        txtolepnumber.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel12.add(txtolepnumber, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 20, 171, -1));


        jLabel10.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel10.setText("Name");
        jPanel12.add(jLabel10, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 54, -1, -1));
```

```
        jLabel11.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel11.setText("Exam Number");
        jPanel12.add(jLabel11, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(10, 82, -1, -1));


        jLabel12.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel12.setText("A 's");
        jPanel12.add(jLabel12, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(10, 111, -1, -1));


        jLabel13.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel13.setText("B 's");
        jPanel12.add(jLabel13, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(10, 139, -1, -1));


        jLabel14.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel14.setText("C 's");
        jPanel12.add(jLabel14, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(10, 170, -1, -1));


        jLabel15.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel15.setText("S 's");
        jPanel12.add(jLabel15, new
    org.netbeans.lib.awtextra.AbsoluteConstraints(10, 200, -1, -1));


        jLabel16.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel16.setText("Maths");
        jPanel12.add(jLabel16, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 230, -1, -1));


        txtolname.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel12.add(txtolname, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 51, 171, -1));


        txtolas.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel12.add(txtolas, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
108, 171, -1));


        txtolbs.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel12.add(txtolbs, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
140, 170, -1));


        txtolcs.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel12.add(txtolcs, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
170, 170, -1));


        cmbolmaths.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Selete Grade", "A", "B", "C", "S", "F" }));

        cmbolmaths.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel12.add(cmbolmaths, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 230, 170, -1));
```

```
    jButton4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/delete-16 (1).png"))); //
NOI18N

    jButton4.setText("Delete");

    jButton4.addActionListener(new java.awt.event.ActionListener() {

      public void actionPerformed(java.awt.event.ActionEvent evt) {

        jButton4ActionPerformed(evt);

      }

    });

    jPanel12.add(jButton4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 308, 85, -1));



    jButton5.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/available-updates-16
(1).png"))); // NOI18N

    jButton5.setText("Update");

    jButton5.addActionListener(new java.awt.event.ActionListener() {

      public void actionPerformed(java.awt.event.ActionEvent evt) {

        jButton5ActionPerformed(evt);

      }

    });

    jPanel12.add(jButton5, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 310, 90, -1));



    jButton6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/add-16.png"))); // NOI18N

    jButton6.setText("Add");

    jButton6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton6ActionPerformed(evt);
            }
        });
        jPanel12.add(jButton6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(190, 310, 80, -1));

        tblol.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        tblol.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {

            }
        ));
        tblol.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_OFF);
        tblol.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                tblolMouseClicked(evt);
            }
        });
        jScrollPane4.setViewportView(tblol);

        jPanel12.add(jScrollPane4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(313, 51, 462, 280));
```

```
        txtolsearch.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        txtolsearch.addKeyListener(new java.awt.event.KeyAdapter() {

            public void keyReleased(java.awt.event.KeyEvent evt) {

                txtolsearchKeyReleased(evt);

            }

        });

        jPanel12.add(txtolsearch, new
org.netbeans.lib.awtextra.AbsoluteConstraints(592, 20, 183, -1));



txtolexamnumber.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel12.add(txtolexamnumber, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 80, 170, -1));


        txtolss.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel12.add(txtolss, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
200, 170, -1));


        jLabel17.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel17.setText("English");

        jPanel12.add(jLabel17, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 260, -1, -1));


        cmbolenglish.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Select Grade", "A", "B", "C", "S", "F" }));
```

```
        cmbolenglish.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel12.add(cmbolenglish, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 260, 170, -1));


        jTabbedPane2.addTab("O/L Exam ", jPanel12);


        jPanel13.setBackground(new java.awt.Color(255, 255, 255));
        jPanel13.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());


        jLabel18.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel18.setText("EP Number");
        jPanel13.add(jLabel18, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 23, -1, -1));


        txtalepnumber.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel13.add(txtalepnumber, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 20, 171, -1));


        jLabel19.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel19.setText("Name");
        jPanel13.add(jLabel19, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 54, -1, -1));


        jLabel20.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel20.setText("Exam Number");
        jPanel13.add(jLabel20, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 82, -1, -1));


        jLabel21.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel21.setText("Stream");

        jPanel13.add(jLabel21, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 111, -1, -1));


        jLabel22.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel22.setText("A 's");

        jPanel13.add(jLabel22, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 139, -1, -1));


        jLabel23.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel23.setText("B 's");

        jPanel13.add(jLabel23, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 170, -1, -1));


        jLabel24.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel24.setText("C 's");

        jPanel13.add(jLabel24, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 200, -1, -1));


        txtalname.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel13.add(txtalname, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 51, 171, -1));
```

```
        txtalas.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel13.add(txtalas, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
140, 170, -1));


        txtalbs.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel13.add(txtalbs, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
170, 170, -1));


        jButton7.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/delete-16 (1).png"))); //
NOI18N

        jButton7.setText("Delete");

        jButton7.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                jButton7ActionPerformed(evt);

            }

        });

        jPanel13.add(jButton7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 310, 90, -1));


        jButton8.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/available-updates-16
(1).png"))); // NOI18N

        jButton8.setText("Update");

        jButton8.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                jButton8ActionPerformed(evt);
            }
        });
        jPanel13.add(jButton8, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 310, 90, -1));

        jButton9.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/add-16.png"))); // NOI18N
        jButton9.setText("Add");
        jButton9.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton9ActionPerformed(evt);
            }
        });
        jPanel13.add(jButton9, new
org.netbeans.lib.awtextra.AbsoluteConstraints(190, 310, 90, -1));

        tblal.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        tblal.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {

            }
        ));
        tblal.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_OFF);
        tblal.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                tblalMouseClicked(evt);
            }
        });
        jScrollPane5.setViewportView(tblal);

        jPanel13.add(jScrollPane5, new
org.netbeans.lib.awtextra.AbsoluteConstraints(313, 51, 462, 280));

        txtalsearch.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        txtalsearch.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyReleased(java.awt.event.KeyEvent evt) {
                txtalsearchKeyReleased(evt);
            }
        });
        jPanel13.add(txtalsearch, new
org.netbeans.lib.awtextra.AbsoluteConstraints(592, 20, 183, -1));

        txtalexnumber.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel13.add(txtalexnumber, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 80, 170, -1));

        txtalcs.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
```

```
        jPanel13.add(txtalcs, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 200, 170, -1));


        cmbalstream.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Select AL Stream", "Commerce", "Maths", "Art", "Science" }));

        cmbalstream.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 185, 205)));

        jPanel13.add(cmbalstream, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 110, 170, -1));


        jLabel25.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel25.setText("S 's");

        jPanel13.add(jLabel25, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 230, -1, -1));


        txtalss.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 185, 205)));

        jPanel13.add(txtalss, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 230, 170, -1));


        jTabbedPane2.addTab("A/L Exam", jPanel13);


        jPanel14.setBackground(new java.awt.Color(255, 255, 255));

        jPanel14.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());


        jLabel26.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel26.setText("EP Number");

        jPanel14.add(jLabel26, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 23, -1, -1));
```

```
    txtadepnumber.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
    jPanel14.add(txtadepnumber, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 20, 171, -1));

    jLabel27.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel27.setText("Name");
    jPanel14.add(jLabel27, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 54, -1, -1));

    jLabel28.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel28.setText("Degree Type");
    jPanel14.add(jLabel28, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 82, -1, -1));

    jLabel29.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel29.setText("Instute");
    jPanel14.add(jLabel29, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 170, -1, -1));

    jLabel30.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel30.setText("Duration");
    jPanel14.add(jLabel30, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 139, -1, -1));

    jLabel31.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel31.setText("Degree ");
```

```
        jPanel14.add(jLabel31, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 110, -1, -1));


        txtadname.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel14.add(txtadname, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 51, 171, -1));


        jButton10.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/delete-16 (1).png"))); //
NOI18N
        jButton10.setText("Delete");
        jButton10.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton10ActionPerformed(evt);
            }
        });
        jPanel14.add(jButton10, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 310, 90, -1));


        jButton11.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/available-updates-16
(1).png"))); // NOI18N
        jButton11.setText("Update");
        jButton11.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton11ActionPerformed(evt);
            }
        });
```

```
    jPanel14.add(jButton11, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 310, 90, -1));


    jButton12.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/add-16.png"))); // NOI18N
    jButton12.setText("Add");
    jButton12.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton12ActionPerformed(evt);
      }
    });
    jPanel14.add(jButton12, new
org.netbeans.lib.awtextra.AbsoluteConstraints(190, 310, 90, -1));


    tblad.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
    tblad.setModel(new javax.swing.table.DefaultTableModel(
      new Object [][] {

      },
      new String [] {

      }
    ));
    tblad.addMouseListener(new java.awt.event.MouseAdapter() {
      public void mouseClicked(java.awt.event.MouseEvent evt) {
        tbladMouseClicked(evt);
```

```
          }
        });
        jScrollPane6.setViewportView(tblad);

        jPanel14.add(jScrollPane6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(313, 51, 462, 280));

        txtadsearch.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        txtadsearch.addKeyListener(new java.awt.event.KeyAdapter() {
          public void keyReleased(java.awt.event.KeyEvent evt) {
            txtadsearchKeyReleased(evt);
          }
        });
        jPanel14.add(txtadsearch, new
org.netbeans.lib.awtextra.AbsoluteConstraints(592, 20, 183, -1));

        txtaddegree.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel14.add(txtaddegree, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 110, 170, -1));

        cmbaddegtype.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Select Degree Category", "Accounting", "Software Engeneering",
"Information Technology", "Management Management", "Human Recource
Management", "Busines Management" }));
        cmbaddegtype.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
```

```
        jPanel14.add(cmbaddegtype, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 80, 170, -1));


        txtadinstute.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel14.add(txtadinstute, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 170, 170, -1));


        cmbduration.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "select Duration", "06 Month", "1 Year", "2 Years", "3 Years", "4 Years"
}));

        cmbduration.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel14.add(cmbduration, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 140, 170, -1));


        jTabbedPane2.addTab("Advance Education", jPanel14);


        jPanel15.setBackground(new java.awt.Color(255, 255, 255));

        jPanel15.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());


        jLabel33.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel33.setText("EP Number");

        jPanel15.add(jLabel33, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 23, -1, -1));


        txtexepnumber.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
```

```
        jPanel15.add(txtexepnumber, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 20, 171, -1));


        jLabel34.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel34.setText("Name");

        jPanel15.add(jLabel34, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 54, -1, -1));


        jLabel35.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel35.setText("Post Type");

        jPanel15.add(jLabel35, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 82, -1, -1));


        jLabel36.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel36.setText("Month");

        jPanel15.add(jLabel36, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 111, -1, -1));


        jLabel37.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel37.setText("Organization");

        jPanel15.add(jLabel37, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 139, -1, -1));


        txtexname.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel15.add(txtexname, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 50, 171, -1));
```

```
    jButton13.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/delete-16 (1).png"))); //
NOI18N
    jButton13.setText("Delete");
    jButton13.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton13ActionPerformed(evt);
      }
    });
    jPanel15.add(jButton13, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 310, 90, -1));


    jButton14.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/available-updates-16
(1).png"))); // NOI18N
    jButton14.setText("Update");
    jButton14.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton14ActionPerformed(evt);
      }
    });
    jPanel15.add(jButton14, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 310, 90, -1));


    jButton15.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/add-16.png"))); // NOI18N
    jButton15.setText("Add");
    jButton15.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton15ActionPerformed(evt);
            }
        });
        jPanel15.add(jButton15, new
org.netbeans.lib.awtextra.AbsoluteConstraints(190, 310, 90, -1));


        tblex.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        tblex.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {},
                {},
                {},
                {}
            },
            new String [] {


            }
        ));
        tblex.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                tblexMouseClicked(evt);
            }
        });
        jScrollPane7.setViewportView(tblex);
        jPanel15.add(jScrollPane7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(313, 51, 462, 280));


        txtexsearch.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        txtexsearch.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyReleased(java.awt.event.KeyEvent evt) {
                txtexsearchKeyReleased(evt);
            }
        });
        jPanel15.add(txtexsearch, new
org.netbeans.lib.awtextra.AbsoluteConstraints(592, 20, 183, -1));


        cmbpostType.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Select post Type", "Accounting", "Software Engeneering",
"Management", "HR", "Marketing" }));
        cmbpostType.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel15.add(cmbpostType, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 80, 170, -1));


        txtexmonth.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel15.add(txtexmonth, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 110, 170, -1));


        txtexorg.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
```

```
        jPanel15.add(txtexorg, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
140, 171, -1));


        jTabbedPane2.addTab("Experiance", jPanel15);


        jPanel5.add(jTabbedPane2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 790, 380));


        jTabbedPane1.addTab("Job Seekers Records", jPanel5);


        jPanel16.setBackground(new java.awt.Color(255, 255, 255));
        jPanel16.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());


        jLabel38.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel38.setText("Post");
        jPanel16.add(jLabel38, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 23, -1, -1));


        txtshpost.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel16.add(txtshpost, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 20, 171, -1));


        jLabel39.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel39.setText("Job Type");
        jPanel16.add(jLabel39, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 54, -1, -1));
```

```
    jLabel40.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel40.setText("Gender");
    jPanel16.add(jLabel40, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 82, -1, -1));

    jLabel41.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel41.setText("Age min");
    jPanel16.add(jLabel41, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 111, -1, -1));

    jLabel42.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel42.setText("Age max");
    jPanel16.add(jLabel42, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 139, -1, -1));

    jLabel43.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel43.setText("Education O/L");
    jPanel16.add(jLabel43, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 170, -1, -1));

    jLabel44.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel44.setText("A 's");
    jPanel16.add(jLabel44, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 200, -1, -1));

    jLabel45.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
    jLabel45.setText("C 's");
```

```
        jPanel16.add(jLabel45, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 200, -1, -1));


        txtshjobtype.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel16.add(txtshjobtype, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 51, 171, -1));


        txtshagemax.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel16.add(txtshagemax, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 140, 170, -1));


        txtshagemin.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel16.add(txtshagemin, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 110, 170, -1));


        jButton16.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/brush-16.png"))); //
NOI18N

        jButton16.setText("Clear");

        jButton16.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                jButton16ActionPerformed(evt);

            }

        });

        jPanel16.add(jButton16, new
org.netbeans.lib.awtextra.AbsoluteConstraints(590, 70, 90, -1));
```

```
    jButton17.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/delete-16 (1).png"))); //
NOI18N

    jButton17.setText("Delete");

    jButton17.addActionListener(new java.awt.event.ActionListener() {

      public void actionPerformed(java.awt.event.ActionEvent evt) {

        jButton17ActionPerformed(evt);

      }

    });

    jPanel16.add(jButton17, new
org.netbeans.lib.awtextra.AbsoluteConstraints(680, 70, 90, -1));


    jButton18.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/add-16.png"))); // NOI18N

    jButton18.setText("add");

    jButton18.addActionListener(new java.awt.event.ActionListener() {

      public void actionPerformed(java.awt.event.ActionEvent evt) {

        jButton18ActionPerformed(evt);

      }

    });

    jPanel16.add(jButton18, new
org.netbeans.lib.awtextra.AbsoluteConstraints(680, 130, 90, -1));


    tblshedule.setModel(new javax.swing.table.DefaultTableModel(

      new Object [][] {

        {},

        {},
        {},
        {}
      },
      new String [] {


      }
    ));
    tblshedule.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_OFF);
    tblshedule.addMouseListener(new java.awt.event.MouseAdapter() {
      public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblsheduleMouseClicked(evt);
      }
    });
    jScrollPane8.setViewportView(tblshedule);


    jPanel16.add(jScrollPane8, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 260, 770, 110));


    txtshsearch.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
    jPanel16.add(txtshsearch, new
org.netbeans.lib.awtextra.AbsoluteConstraints(590, 230, 190, -1));


    txtsholss.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
    jPanel16.add(txtsholss, new
org.netbeans.lib.awtextra.AbsoluteConstraints(230, 200, 30, -1));
```

```
        txtsholcs.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel16.add(txtsholcs, new
org.netbeans.lib.awtextra.AbsoluteConstraints(170, 200, 30, -1));


        txtsholas.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel16.add(txtsholas, new
org.netbeans.lib.awtextra.AbsoluteConstraints(40, 200, 30, -1));


        txtsholbs.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel16.add(txtsholbs, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 200, 30, -1));


        jLabel46.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel46.setText("S 's");
        jPanel16.add(jLabel46, new
org.netbeans.lib.awtextra.AbsoluteConstraints(210, 200, -1, -1));


        jLabel47.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel47.setText("B 's");
        jPanel16.add(jLabel47, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 200, -1, -1));


        jLabel48.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel48.setText("Education A/L");
        jPanel16.add(jLabel48, new
org.netbeans.lib.awtextra.AbsoluteConstraints(290, 20, -1, -1));
```

```
        jLabel49.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel49.setText("A 's");
        jPanel16.add(jLabel49, new
org.netbeans.lib.awtextra.AbsoluteConstraints(290, 70, -1, -1));

        txtshalas.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel16.add(txtshalas, new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 70, 30, -1));

        txtshalbs.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel16.add(txtshalbs, new
org.netbeans.lib.awtextra.AbsoluteConstraints(390, 70, 30, -1));

        jLabel50.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel50.setText("B 's");
        jPanel16.add(jLabel50, new
org.netbeans.lib.awtextra.AbsoluteConstraints(370, 70, -1, -1));

        jLabel51.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel51.setText("C 's");
        jPanel16.add(jLabel51, new
org.netbeans.lib.awtextra.AbsoluteConstraints(430, 70, -1, -1));

        txtshalcs.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
```

```
        jPanel16.add(txtshalcs, new
org.netbeans.lib.awtextra.AbsoluteConstraints(460, 70, 30, -1));


        jLabel52.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel52.setText("S 's");

        jPanel16.add(jLabel52, new
org.netbeans.lib.awtextra.AbsoluteConstraints(510, 70, -1, -1));


        txtshalss.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel16.add(txtshalss, new
org.netbeans.lib.awtextra.AbsoluteConstraints(530, 70, 30, -1));


        jLabel53.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel53.setText("Stream");

        jPanel16.add(jLabel53, new
org.netbeans.lib.awtextra.AbsoluteConstraints(290, 40, -1, -1));


        cmbshalstream.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Select AL Stream", "Commerce", "Maths", "Art", "Science" }));

        cmbshalstream.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel16.add(cmbshalstream, new
org.netbeans.lib.awtextra.AbsoluteConstraints(390, 40, 170, -1));


        jLabel54.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel54.setText("Degree");

        jPanel16.add(jLabel54, new
org.netbeans.lib.awtextra.AbsoluteConstraints(290, 110, -1, -1));
```

```
    jLabel55.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

    jLabel55.setText("Type");

    jPanel16.add(jLabel55, new
org.netbeans.lib.awtextra.AbsoluteConstraints(290, 140, -1, -1));


    jLabel56.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

    jLabel56.setText("Duration");

    jPanel16.add(jLabel56, new
org.netbeans.lib.awtextra.AbsoluteConstraints(290, 170, -1, -1));


    jLabel57.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

    jLabel57.setText("Degree Category");

    jPanel16.add(jLabel57, new
org.netbeans.lib.awtextra.AbsoluteConstraints(290, 200, -1, -1));


    cmbshdegcategory.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Select Degree Category", "Accounting", "Software Engeneering",
"Information Technology", "Management Management", "Human Recource
Management", "Busines Management" }));


cmbshdegcategory.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

    jPanel16.add(cmbshdegcategory, new
org.netbeans.lib.awtextra.AbsoluteConstraints(400, 200, 160, -1));


    cmbshdegreeduration.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "select Duration", "06
Month", "1 Year", "2 Years", "3 Years", "4 Years" }));
```

```
cmbshdegreeduration.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel16.add(cmbshdegreeduration, new
org.netbeans.lib.awtextra.AbsoluteConstraints(400, 170, 160, -1));


        cmbshgdegreetype.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Select Degree Type", "Accounting", "Software Engeneering",
"Management", "HR", "Marketing" }));


cmbshgdegreetype.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel16.add(cmbshgdegreetype, new
org.netbeans.lib.awtextra.AbsoluteConstraints(400, 140, 160, -1));


        jButton19.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/Image/available-updates-16
(1).png"))); // NOI18N
        jButton19.setText("update");
        jButton19.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton19ActionPerformed(evt);
            }
        });
        jPanel16.add(jButton19, new
org.netbeans.lib.awtextra.AbsoluteConstraints(590, 130, 90, -1));


        cmbshgender.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Select Gender", "Male", "Female" }));
```

```
        cmbshgender.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel16.add(cmbshgender, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 80, 170, -1));


        jLabel32.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel32.setText("Maths");

        jPanel16.add(jLabel32, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 230, -1, -1));


        jLabel60.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N

        jLabel60.setText("English");

        jPanel16.add(jLabel60, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 230, -1, -1));


        txtsholmaths.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel16.add(txtsholmaths, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 230, 70, -1));


        txtsholenglish.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

        jPanel16.add(txtsholenglish, new
org.netbeans.lib.awtextra.AbsoluteConstraints(190, 230, 70, -1));


        chkol.setText("Record");

        chkol.addMouseListener(new java.awt.event.MouseAdapter() {

            public void mouseClicked(java.awt.event.MouseEvent evt) {

                chkolMouseClicked(evt);
            }
        });
        jPanel16.add(chkol, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 170, -1, -1));

        chkal.setText("Record");
        chkal.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                chkalMouseClicked(evt);
            }
        });
        jPanel16.add(chkal, new org.netbeans.lib.awtextra.AbsoluteConstraints(480, 10, -1, -1));

        chkdeg.setText("Record");
        chkdeg.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                chkdegMouseClicked(evt);
            }
        });
        jPanel16.add(chkdeg, new org.netbeans.lib.awtextra.AbsoluteConstraints(480, 110, -1, -1));

        jTabbedPane1.addTab("Generate Job Post", jPanel16);

        jPanel17.setBackground(new java.awt.Color(255, 255, 255));
        jPanel17.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
        jLabel59.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabel59.setText("Post");
        jPanel17.add(jLabel59, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 20, -1, -1));

        tblselectemp.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {},
                {},
                {},
                {}
            },
            new String [] {


            }
        ));
        jScrollPane9.setViewportView(tblselectemp);

        jPanel17.add(jScrollPane9, new
org.netbeans.lib.awtextra.AbsoluteConstraints(15, 51, 760, 310));

        jTextField47.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));
        jPanel17.add(jTextField47, new
org.netbeans.lib.awtextra.AbsoluteConstraints(592, 20, 183, -1));
```

```
    cmbsepost.setModel(new javax.swing.DefaultComboBoxModel(new String[]
{ "Select Post" }));

    cmbsepost.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 185, 205)));

    cmbsepost.addPopupMenuListener(new
javax.swing.event.PopupMenuListener() {

      public void popupMenuCanceled(javax.swing.event.PopupMenuEvent evt)
{

      }

      public void
popupMenuWillBecomeInvisible(javax.swing.event.PopupMenuEvent evt) {

        cmbsepostPopupMenuWillBecomeInvisible(evt);

      }

      public void
popupMenuWillBecomeVisible(javax.swing.event.PopupMenuEvent evt) {

      }

    });

    jPanel17.add(cmbsepost, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 20, 170, -1));


    jTabbedPane1.addTab("Select   Candidate", jPanel17);


    jPanel1.add(jTabbedPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 40, 800, 410));


    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());

    getContentPane().setLayout(layout);

    layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );


        pack();
        setLocationRelativeTo(null);
    }// </editor-fold>



    public void clock()
    {
        Thread clock=new Thread()
                {
                public void run()
                {
                    try {
                   while(true)
                        {
                        Calendar cal=new GregorianCalendar();
                        int day=cal.get(Calendar.DAY_OF_MONTH);
                        int month=cal.get(Calendar.MONTH)+1;
                        int year=cal.get(Calendar.YEAR);

                        int second=cal.get(Calendar.SECOND);
                        int minute=cal.get(Calendar.MINUTE);
                        int hour=cal.get(Calendar.HOUR);

                        lbltime.setText( hour +":"+ minute +":"+ second);
                        lbldate.setText( year +"-"+ month +"-"+ day );
                      //  getTime=lbltime.getText();
                       // getDate=lbldate.getText();
                                        sleep(1000);
                        }
                    } catch (InterruptedException ex) {
                       // Logger.getLogger(main.class.getName()).log(Level.SEVERE,
null, ex);

                    }
                }
            };
        clock.start();
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    getPersonalData();


    try {
        String q= "INSERT INTO `personal_data`(`ep_number`, `name`,
`gender`, `nic`, `adress`, `contact_num`, `age`, `distric`)"
            + " VALUES ('"+epnumber+"',
'"+name+"','"+gender+"','"+nic+"','"+address+"','"+contactNumber+"','"+age+"','"+
distric+"')";
        pst=conn.prepareStatement(q);
        pst.execute();



         JOptionPane.showMessageDialog(rootPane,"Successfully Added");
           }
        catch (Exception e)
        {
        JOptionPane.showMessageDialog(rootPane,e);
        }
    personalDataTablelord();
    clearPersonalData();
    txtolepnumber.setText(epnumber);
    txtolname.setText(name);
  }


  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    getPersonalData();
```

```
        try {
            String squpdate = "UPDATE personal_data SET
name='"+name+"',gender='"+gender+"',nic='"+nic+"',adress='"+address+"',contact
_num='"+contactNumber+"',age='"+age+"',distric='"+distric+"' WHERE
ep_number='"+epnumber+"'";

            pst=conn.prepareStatement(squpdate);

              pst.execute();

          } catch (Exception e) {

              JOptionPane.showMessageDialog(rootPane,e);

          }


       personalDataTablelord();

       clearPersonalData();


    }


    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {


        int x=JOptionPane.showConfirmDialog(rootPane, "Do you realy want to
delete");

        if(x==0)

        {

            if(txtpdepnumber.getText().length()>0)

            {

        try {


              String sql="DELETE FROM `personal_data` WHERE ep_number='"+
txtpdepnumber.getText() +"'";
```

```
                pst=conn.prepareStatement(sql);
                pst.execute();
           JOptionPane.showMessageDialog(rootPane, "sccessfully Delete");
            } catch (Exception e) {
                JOptionPane.showMessageDialog(rootPane, e);
            }
            }
            else{
                JOptionPane.showMessageDialog(rootPane, "Select Employee to 
Delete");
            }
        }

        personalDataTablelord();
        clearPersonalData();
    }

    private void tblpdMouseClicked(java.awt.event.MouseEvent evt) {
       DefaultTableModel tmodel=(DefaultTableModel)tblpd.getModel();
        int selectrowindex=tblpd.getSelectedRow();
        txtpdepnumber.setText(tmodel.getValueAt(selectrowindex, 0).toString());
        txtpdname.setText(tmodel.getValueAt(selectrowindex, 1).toString());
        if(tmodel.getValueAt(selectrowindex, 2).toString().equals("Male"))
        {
        rbtmale.setSelected(true);
        }
        else
        {
        rbtfemale.setSelected(true);
        }
        txtpdnic.setText(tmodel.getValueAt(selectrowindex, 3).toString());
        txtadress.setText(tmodel.getValueAt(selectrowindex, 4).toString());
        txtpdcontactnumber.setText(tmodel.getValueAt(selectrowindex,
5).toString());
        txtpdage.setText(tmodel.getValueAt(selectrowindex, 6).toString());
        cmbpdcity.setSelectedItem(tmodel.getValueAt(selectrowindex, 7).toString());
    }


    private void txtpdsearchKeyReleased(java.awt.event.KeyEvent evt) {
        if(txtpdsearch.getText().length()>0)
        {
        try {
            String sql="SELECT `ep_number` as 'EP Number', `name` as 'Name',
`gender` as 'Gender', `nic` as 'NIC Number', `adress` as 'Address', `contact_num` as
'Contect Number', `age` as 'Age', `distric` as 'Distric' FROM `personal_data`
WHERE ep_number='"+txtpdsearch.getText()+"'";
            pst=conn.prepareStatement(sql);
            rs=pst.executeQuery();
            tblpd.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));


        } catch (Exception e) {
            JOptionPane.showConfirmDialog(rootPane, e);
        }
        }
        else
        {
        personalDataTablelord();


        }
    }


    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {


        getOLData();
        try {
            String q= "INSERT INTO `olevel_data`(`ep_number`, `name`,
`ol_ex_number`, `ol_as`, `ol_bs`, `ol_cs`, `ol_ss`, `ol_math`, `ol_english`)"
                    + " VALUES ('"+epnumber+"',
'"+name+"','"+olexamNumber+"','"+olas+"','"+olbs+"','"+olcs+"','"+olss+"','"+olma
ths+"','"+olenglish+"')";
            pst=conn.prepareStatement(q);
            pst.execute();



             JOptionPane.showMessageDialog(rootPane,"Successfully Added");
                }
            catch (Exception e)
            {
            JOptionPane.showMessageDialog(rootPane,e);
            }
        txtalepnumber.setText(epnumber);
       txtalname.setText(name);
        clearOLData();
        olDataTablelord();
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
      getOLData();
        try {
            String squpdate = "UPDATE olevel_data SET
ol_ex_number='"+olexamNumber+"',ol_as='"+olas+"',ol_bs='"+olbs+"',ol_cs='"+o
lcs+"',ol_ss='"+olss+"',ol_math='"+olmaths+"',ol_english='"+olenglish+"' WHERE
ep_number='"+txtolepnumber.getText()+"'";
            pst=conn.prepareStatement(squpdate);
              pst.execute();
          } catch (Exception e) {
              JOptionPane.showMessageDialog(rootPane,e);
          }
        olDataTablelord();
        clearOLData();
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        int x=JOptionPane.showConfirmDialog(rootPane, "Do you realy want to
delete");
        if(x==0)
        {
          if(txtolepnumber.getText().length()>0)

          {

        try {


              String sql="DELETE FROM `olevel_data` WHERE ep_number='"+
txtolepnumber.getText() +"'";

              pst=conn.prepareStatement(sql);

              pst.execute();

          JOptionPane.showMessageDialog(rootPane, "sccessfully Delete");

          } catch (Exception e) {

              JOptionPane.showMessageDialog(rootPane, e);

          }

          }

          else{

              JOptionPane.showMessageDialog(rootPane, "Select Employee to
Delete");

          }

        }



        clearOLData();


        olDataTablelord();

    }


    private void txtolsearchKeyReleased(java.awt.event.KeyEvent evt) {
```

```
if(txtolsearch.getText().length()>0)
{
    try {
        String sql="SELECT `ep_number` as 'EP Number', `name` as 'Name', `ol_ex_number` as 'Exam Number', `ol_as` as 'A Grades', `ol_bs` as 'B Grades', `ol_cs` as 'C Grades', `ol_ss` as 'S Grades', `ol_math` as 'Maths', `ol_english` as 'English' FROM `olevel_data` WHERE ep_number = '"+txtolsearch.getText()+"'";
        pst=conn.prepareStatement(sql);
        rs=pst.executeQuery();
        tblol.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));


    } catch (Exception e) {
        JOptionPane.showConfirmDialog(rootPane, e);
    }
 //  theader();
}
else
{
olDataTablelord();


}


}


private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
   getaLData();
```

```
      try {
              String q= "INSERT INTO `alevel_data`(`ep_number`, `name`,
`ex_number`, `al_stream`, `al_as`, `al_bs`, `al_cs`, `al_ss`)"
                    + " VALUES ('"+epnumber+"',
'"+name+"','"+alexamNumber+"','"+stream+"','"+alas+"','"+albs+"','"+alcs+"','"+als
s+"')";
              pst=conn.prepareStatement(q);
              pst.execute();



               JOptionPane.showMessageDialog(rootPane,"Successfully Added");
                 }
              catch (Exception e)
              {
              JOptionPane.showMessageDialog(rootPane,e);
              }
      alDataTablelord();
      clearALData();
      txtadepnumber.setText(epnumber);
      txtadname.setText(name);
    }

    private void tblolMouseClicked(java.awt.event.MouseEvent evt) {
       DefaultTableModel tmodel=(DefaultTableModel)tblol.getModel();
       int selectrowindex=tblol.getSelectedRow();
       txtolepnumber.setText(tmodel.getValueAt(selectrowindex, 0).toString());
```

```
        txtolname.setText(tmodel.getValueAt(selectrowindex, 1).toString());


        txtolexamnumber.setText(tmodel.getValueAt(selectrowindex, 2).toString());
        txtolas.setText(tmodel.getValueAt(selectrowindex, 3).toString());
        txtolbs.setText(tmodel.getValueAt(selectrowindex, 4).toString());
        txtolcs.setText(tmodel.getValueAt(selectrowindex, 5).toString());
        txtolss.setText(tmodel.getValueAt(selectrowindex, 6).toString());
        cmbolmaths.setSelectedItem(tmodel.getValueAt(selectrowindex,
7).toString());
        cmbolenglish.setSelectedItem(tmodel.getValueAt(selectrowindex,
8).toString());
    }


    private void tblalMouseClicked(java.awt.event.MouseEvent evt) {
      DefaultTableModel tmodel=(DefaultTableModel)tblal.getModel();
        int selectrowindex=tblal.getSelectedRow();
        txtalepnumber.setText(tmodel.getValueAt(selectrowindex, 0).toString());
        txtalname.setText(tmodel.getValueAt(selectrowindex, 1).toString());
        txtalexnumber.setText(tmodel.getValueAt(selectrowindex, 2).toString());
        cmbalstream.setSelectedItem(tmodel.getValueAt(selectrowindex,
3).toString());


        txtalas.setText(tmodel.getValueAt(selectrowindex, 4).toString());
        txtalbs.setText(tmodel.getValueAt(selectrowindex, 5).toString());
        txtalcs.setText(tmodel.getValueAt(selectrowindex, 6).toString());
        txtalss.setText(tmodel.getValueAt(selectrowindex, 7).toString());
    }


    private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
      getaLData();
        try {
              String squpdate = "UPDATE alevel_data SET
ex_number='"+alexamNumber+"',al_stream='"+stream+"',al_as='"+alas+"',al_bs='"
+albs+"',al_cs='"+alcs+"',al_ss='"+alss+"' WHERE
ep_number='"+txtalepnumber.getText()+"'";
              pst=conn.prepareStatement(squpdate);
                pst.execute();
            } catch (Exception e) {
                JOptionPane.showMessageDialog(rootPane,e);
            }
        alDataTablelord();
        clearALData();
    }


    private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
       int x=JOptionPane.showConfirmDialog(rootPane, "Do you realy want to
delete");
        if(x==0)
        {
            if(txtalepnumber.getText().length()>0)
            {
        try {
                String sql="DELETE FROM `alevel_data` WHERE ep_number='"+
txtalepnumber.getText() +"'";

                pst=conn.prepareStatement(sql);

                pst.execute();

             JOptionPane.showMessageDialog(rootPane, "sccessfully Delete");

            } catch (Exception e) {

                JOptionPane.showMessageDialog(rootPane, e);

            }

            }

            else{

                JOptionPane.showMessageDialog(rootPane, "Select Employee to
Delete");

            }

        }



        clearALData();


        alDataTablelord();

    }


    private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {

        getADData();


       try {
                String q= "INSERT INTO `degree_data`(`ep_number`, `name`,
`degree_type`, `degree`, `duration`, `instute`)"
               + " VALUES ('"+epnumber+"',
'"+name+"','"+degreeType+"','"+degree+"','"+duration+"','"+instute+"')";
                pst=conn.prepareStatement(q);
                pst.execute();



                 JOptionPane.showMessageDialog(rootPane,"Successfully Added");
                   }
                catch (Exception e)
                {
                JOptionPane.showMessageDialog(rootPane,e);
                }
       aDDataTablelord();
       clearADData();
       txtexname.setText(epnumber);
       txtexepnumber.setText(name);
    }

    private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {
       getADData();
        try {
                String squpdate = "UPDATE degree_data SET
degree_type='"+degreeType+"',degree='"+degree+"',duration='"+duration+"',instut
e='"+instute+"' WHERE ep_number='"+txtadepnumber.getText()+"'";
                pst=conn.prepareStatement(squpdate);
```

```
            pst.execute();
        } catch (Exception e) {
            JOptionPane.showMessageDialog(rootPane,e);
        }
     aDDataTablelord();
     clearADData();
   }

   private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {
     int x=JOptionPane.showConfirmDialog(rootPane, "Do you realy want to
delete");
      if(x==0)
      {
        if(txtadepnumber.getText().length()>0)
        {
      try {

            String sql="DELETE FROM `degree_data` WHERE ep_number='"+
txtadepnumber.getText() +"'";
            pst=conn.prepareStatement(sql);
            pst.execute();
       JOptionPane.showMessageDialog(rootPane, "sccessfully Delete");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(rootPane, e);
        }
        }
```

```
            else{
                JOptionPane.showMessageDialog(rootPane, "Select Employee to
Delete");
            }
        }
        aDDataTablelord();
       clearADData();
    }

    private void tbladMouseClicked(java.awt.event.MouseEvent evt) {
       DefaultTableModel tmodel=(DefaultTableModel)tblad.getModel();
        int selectrowindex=tblad.getSelectedRow();
        txtadepnumber.setText(tmodel.getValueAt(selectrowindex, 0).toString());
        txtadname.setText(tmodel.getValueAt(selectrowindex, 1).toString());
        cmbaddegtype.setSelectedItem(tmodel.getValueAt(selectrowindex,
2).toString());
        txtaddegree.setText(tmodel.getValueAt(selectrowindex, 3).toString());
        cmbduration.setSelectedItem(tmodel.getValueAt(selectrowindex,
4).toString());
        txtadinstute.setText(tmodel.getValueAt(selectrowindex, 5).toString());

    }

    private void txtalsearchKeyReleased(java.awt.event.KeyEvent evt) {

        if(txtalsearch.getText().length()>0)
        {
```

```
        try {

            String sql="SELECT `ep_number` as 'EP Number', `name` as 'Name',
`ex_number` as 'Exam Number', `al_stream` as 'As Stream', `al_as` as 'A Greades',
`al_bs` as 'B Grades', `al_cs` as 'C Grades', `al_ss` as 'S Gredes' FROM
`alevel_data` WHERE ep_number = '"+txtalsearch.getText()+"'";

            pst=conn.prepareStatement(sql);

            rs=pst.executeQuery();

            tblal.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));


        } catch (Exception e) {

            JOptionPane.showConfirmDialog(rootPane, e);

        }

        }

        else

        {

        alDataTablelord();


        }

    }


    private void txtadsearchKeyReleased(java.awt.event.KeyEvent evt) {


        if(txtadsearch.getText().length()>0)

        {

        try {

            String sql="SELECT `ep_number` as 'EP Number', `name` as 'Name',
`degree_type` as 'Degree Type', `degree` as 'Degree Name', `duration` as 'Duration',
`instute` as 'Instute' FROM `degree_data` WHERE ep_number =
'"+txtadsearch.getText()+"'";
            pst=conn.prepareStatement(sql);
            rs=pst.executeQuery();
            tblad.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));

        } catch (Exception e) {
            JOptionPane.showConfirmDialog(rootPane, e);
        }
        }
        else
        {
        aDDataTablelord();

        }
    }

    private void txtexsearchKeyReleased(java.awt.event.KeyEvent evt) {

        if(txtexsearch.getText().length()>0)
        {
        try {
            String sql="SELECT `ep_number` as 'EP Number', `name` as 'Name',
`post_type` as 'Post Type', `month` as 'Month', `organization` as 'Organization'
FROM `experiance_data` WHERE ep_number = '"+txtexsearch.getText()+"'";
            pst=conn.prepareStatement(sql);
            rs=pst.executeQuery();
            tblex.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));


        } catch (Exception e) {
            JOptionPane.showConfirmDialog(rootPane, e);
        }
        }
        else
        {
        exDataTablelord();


        }
    }


    private void jButton15ActionPerformed(java.awt.event.ActionEvent evt) {


        getexData();
        try {
                String q= "INSERT INTO `experiance_data`(`ep_number`, `name`,
`post_type`, `month`, `organization`)"
                        + " VALUES ('"+epnumber+"',
'"+name+"','"+expostType+"','"+exmonth+"','"+exorg+"')";
                pst=conn.prepareStatement(q);
                pst.execute();




                JOptionPane.showMessageDialog(rootPane,"Successfully Added");
                }
              catch (Exception e)
              {
              JOptionPane.showMessageDialog(rootPane,e);
              }
        exDataTablelord();
        clearexData();
    }


    private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
        getexData();
         try {
             String squpdate = "UPDATE experiance_data SET
post_type='"+expostType+"',month='"+exmonth+"',organization='"+exorg+"'
WHERE ep_number='"+txtexepnumber.getText()+"'";
             pst=conn.prepareStatement(squpdate);
              pst.execute();
           } catch (Exception e) {
              JOptionPane.showMessageDialog(rootPane,e);
           }
        exDataTablelord();
        clearexData();
    }


    private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {
         int x=JOptionPane.showConfirmDialog(rootPane, "Do you realy want to
delete");
```

```
        if(x==0)

        {

            if(txtexepnumber.getText().length()>0)

            {

        try {


                String sql="DELETE FROM `experiance_data` WHERE ep_number='"+
txtexepnumber.getText() +"'";

                pst=conn.prepareStatement(sql);

                pst.execute();

           JOptionPane.showMessageDialog(rootPane, "sccessfully Delete");

            } catch (Exception e) {

                JOptionPane.showMessageDialog(rootPane, e);

            }

            }

            else{

                JOptionPane.showMessageDialog(rootPane, "Select Employee to
Delete");

            }

        }

        exDataTablelord();

       clearexData();

    }


    private void tblexMouseClicked(java.awt.event.MouseEvent evt) {

        DefaultTableModel tmodel=(DefaultTableModel)tblex.getModel();
        int selectrowindex=tblex.getSelectedRow();

        txtexepnumber.setText(tmodel.getValueAt(selectrowindex, 0).toString());

        txtexname.setText(tmodel.getValueAt(selectrowindex, 1).toString());

        cmbpostType.setSelectedItem(tmodel.getValueAt(selectrowindex,
2).toString());

        txtexmonth.setText(tmodel.getValueAt(selectrowindex, 3).toString());

        txtexorg.setText(tmodel.getValueAt(selectrowindex, 4).toString());

    }


    private void jButton18ActionPerformed(java.awt.event.ActionEvent evt) {


        getshData();




        try {

                String q= "INSERT INTO `shedule`(`post`, `job_type`, `gender`,
`age_min`, `age_max`,olcheck, `ol_as`, "

                        + "`ol_bs`, `ol_cs`, `ol_ss`, `ol_maths`, `ol_english`, alcheck,
`al_stream`, `al_as`, `al_bs`, `al_cs`, `al_ss`, degcheck, `deg_type`, "

                        + "`deg_duration`, `deg_category`) VALUES ('"+shpost+"',
'"+shjobtype+"','"+shgender+"','"+shagemin+"',"

                        + "'"+shagemax+"',
'"+olcheck+"','"+sholas+"','"+sholbs+"','"+sholcs+"', '"+sholss+"','"+sholmaths+"',
'"+sholenglish+"', '"+alcheck+"','"+shalstream+"',"

                        + "'"+shalas+"','"+shalbs+"', '"+shalcs+"','"+shalss+"',
'"+degcheck+"','"+shdegtype+"','"+shdegduration+"',"

                        + "'"+shdegcategory+"')";

                pst=conn.prepareStatement(q);
                pst.execute();



                 JOptionPane.showMessageDialog(rootPane,"Successfully Added");
                    }
                catch (Exception e)
                {
                JOptionPane.showMessageDialog(rootPane,e);
                }

            shDataTablelord();
            clearshData();
        }

        private void jButton19ActionPerformed(java.awt.event.ActionEvent evt) {
          getshData();

            try {
                String squpdate = "UPDATE `shedule` SET `job_type`='"+shjobtype+"',"
                    +
"`gender`='"+shgender+"',`age_min`='"+shagemin+"',`age_max`='"+shagemax+"',o
lcheck='"+olcheck+"',"
                    +
"`ol_as`='"+sholas+"',`ol_bs`='"+sholbs+"',`ol_cs`='"+sholcs+"',`ol_ss`='"+sholss+
"',"
                    +
"`ol_maths`='"+sholmaths+"',`ol_english`='"+sholenglish+"',alcheck='"+alcheck+"'
, `al_stream`='"+shalstream+"',`al_as`='"+shalas+"',`al_bs`='"+shalbs+"',"
```

```
                  + "`al_cs`='"+shalcs+"',`al_ss`='"+shalss+"',
degcheck='"+degcheck+"', `deg_type`='"+shdegtype+"',"
                  +
"`deg_duration`='"+shdegduration+"',`deg_category`='"+shdegcategory+"'"
                  + " WHERE post='"+txtshpost.getText()+"'";
            pst=conn.prepareStatement(squpdate);
             pst.execute();
          } catch (Exception e) {
             JOptionPane.showMessageDialog(rootPane,e);
          }

       shDataTablelord();
       clearshData();
    }

    private void jButton17ActionPerformed(java.awt.event.ActionEvent evt) {
       getshData();

       int x=JOptionPane.showConfirmDialog(rootPane, "Do you realy want to
delete");
       if(x==0)
       {
          if(txtshpost.getText().length()>0)
          {
       try {
```

```
                String sql="DELETE FROM `shedule` WHERE post='"+
txtshpost.getText() +"'";

                pst=conn.prepareStatement(sql);

                pst.execute();

           JOptionPane.showMessageDialog(rootPane, "sccessfully Delete");

            } catch (Exception e) {

                JOptionPane.showMessageDialog(rootPane, e);

            }

            }

            else{

                JOptionPane.showMessageDialog(rootPane, "Select Employee to
Delete");

            }

        }




        shDataTablelord();

        clearshData();

    }


    private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {




        shDataTablelord();

        clearshData();

    }
```

```
   private void tblsheduleMouseClicked(java.awt.event.MouseEvent evt) {
     DefaultTableModel tmodel=(DefaultTableModel)tblshedule.getModel();
       int selectrowindex=tblshedule.getSelectedRow();


    txtshpost.setText(tmodel.getValueAt(selectrowindex, 0).toString());
    txtshjobtype.setText(tmodel.getValueAt(selectrowindex, 1).toString());
    cmbshgender.setSelectedItem(tmodel.getValueAt(selectrowindex,
2).toString());
    txtshagemin.setText(tmodel.getValueAt(selectrowindex, 3).toString());
    txtshagemax.setText(tmodel.getValueAt(selectrowindex, 4).toString());
    txtsholas.setText(tmodel.getValueAt(selectrowindex, 5).toString());
    txtsholbs.setText(tmodel.getValueAt(selectrowindex, 6).toString());
    txtsholcs.setText(tmodel.getValueAt(selectrowindex, 7).toString());
    txtsholss.setText(tmodel.getValueAt(selectrowindex, 8).toString());


    txtsholmaths.setText(tmodel.getValueAt(selectrowindex, 9).toString());
    txtsholenglish.setText(tmodel.getValueAt(selectrowindex, 10).toString());


    cmbshalstream.setSelectedItem(tmodel.getValueAt(selectrowindex,
11).toString());
    txtshalas.setText(tmodel.getValueAt(selectrowindex, 12).toString());
    txtshalbs.setText(tmodel.getValueAt(selectrowindex, 13).toString());
    txtshalcs.setText(tmodel.getValueAt(selectrowindex, 14).toString());
    txtshalss.setText(tmodel.getValueAt(selectrowindex, 15).toString());
    cmbshgdegreetype.setSelectedItem(tmodel.getValueAt(selectrowindex,
16).toString());
```

```
    cmbshdegreeduration.setSelectedItem(tmodel.getValueAt(selectrowindex,
17).toString());
    cmbshdegcategory.setSelectedItem(tmodel.getValueAt(selectrowindex,
18).toString());
   }

   private void chkolMouseClicked(java.awt.event.MouseEvent evt) {



      if(olcheck==0)
      {
       txtsholas.setEnabled(true);
       txtsholbs.setEnabled(true);
       txtsholcs.setEnabled(true);
       txtsholss.setEnabled(true);
       txtsholmaths.setEnabled(true);
       txtsholenglish.setEnabled(true);

       olcheck=1;
      }
      else if(olcheck==1)
      {
       txtsholas.setEnabled(false);
       txtsholbs.setEnabled(false);
       txtsholcs.setEnabled(false);
```

```
     txtsholss.setEnabled(false);
     txtsholmaths.setEnabled(false);
     txtsholenglish.setEnabled(false);
     olcheck=0;
   }

}

private void chkalMouseClicked(java.awt.event.MouseEvent evt) {
   if(alcheck==0)
   {
     txtshalas.setEnabled(true);
     txtshalbs.setEnabled(true);
     txtshalcs.setEnabled(true);
     txtshalss.setEnabled(true);
     cmbshalstream.setEnabled(true);

     alcheck=1;
   }
   else if(alcheck==1)
   {
      txtshalas.setEnabled(false);
     txtshalbs.setEnabled(false);
     txtshalcs.setEnabled(false);
     txtshalss.setEnabled(false);
     cmbshalstream.setEnabled(false);
```

```
          alcheck=0;
        }
    }

    private void chkdegMouseClicked(java.awt.event.MouseEvent evt) {
        if(degcheck==0)
        {
         cmbshdegcategory.setEnabled(true);
          cmbshdegreeduration.setEnabled(true);
          cmbshgdegreetype.setEnabled(true);


          degcheck=1;
        }
        else if(degcheck==1)
        {
           cmbshdegcategory.setEnabled(false);
          cmbshdegreeduration.setEnabled(false);
          cmbshgdegreetype.setEnabled(false);
          degcheck=0;
        }
    }

    private void
cmbsepostPopupMenuWillBecomeInvisible(javax.swing.event.PopupMenuEvent
evt) {
    postpelord();
    getValue();
    if(olvalue==0)
    {
     empbasicLoad();
    }
    else if(olvalue==1 && alvalue==0 && degvalue==0)
    {
    empolLoad();
    }
    else if(olvalue==1 && alvalue==1 && degvalue==0)
    {
    empalLoad();
    }
    else if(olvalue==1 && alvalue==1 && degvalue==1)
    {
    empdegLoad();
    }

    cmbsepost.removeAllItems();
    cmbsepost.addItem("Select Post");
    postpelord();
}
```

```
private void getPersonalData()
{

 epnumber = txtpdepnumber.getText();
 name = txtpdname.getText();
 if(rbtmale.isSelected())
 {
 gender = "Male";
 }
 else if(rbtfemale.isSelected())
 {
 gender = "Female";
 }
 else
 {
     JOptionPane.showMessageDialog(rootPane, "Select Gender");
 }
 nic = txtpdnic.getText();
 address = txtadress.getText();
 contactNumber = txtpdcontactnumber.getText();
 age = txtpdage.getText();
 distric = cmbpdcity.getSelectedItem().toString();

}

private void clearPersonalData()
```

```
   {


    txtpdepnumber.setText("");

    txtpdname.setText("");


    txtpdnic.setText("");

    txtadress.setText("");

    txtpdcontactnumber.setText("");

    txtpdage.setText("");

    cmbpdcity.setSelectedIndex(0);


   }


    private void personalDataTablelord()

   {

      try {

          String sql="SELECT `ep_number` as 'EP Number', `name` as 'Name',
`gender` as 'Gender', `nic` as 'NIC Number', `adress` as 'Address', `contact_num` as
'Contect Number', `age` as 'Age', `distric` as 'Distric' FROM `personal_data`";

          pst=conn.prepareStatement(sql);

          rs=pst.executeQuery();

          tblpd.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));


      } catch (Exception e) {

          JOptionPane.showConfirmDialog(rootPane, e);

      }
```

```
 //  theader();
}

 // Ol data handle

 private void getOLData()
{
   olexamNumber = txtolexamnumber.getText();
   olas = txtolas.getText();
   olbs = txtolbs.getText();
   olcs = txtolcs.getText();
   olss = txtolss.getText();
   olmaths = cmbolmaths.getSelectedItem().toString();
   olenglish = cmbolenglish.getSelectedItem().toString();
}

  private void clearOLData()
{
   txtolexamnumber.setText("");
   txtolas.setText("");
   txtolbs.setText("");
   txtolcs.setText("");
   txtolss.setText("");
   cmbolmaths.setSelectedIndex(0);
   cmbolenglish.setSelectedIndex(0);
}
```

```
    private void olDataTablelord()
    {
      try {
          String sql="SELECT `ep_number` as 'EP Number', `name` as 'Name',
`ol_ex_number` as 'Exam Number', `ol_as` as 'A Grades', `ol_bs` as 'B Grades',
`ol_cs` as 'C Grades', `ol_ss` as 'S Grades', `ol_math` as 'Maths', `ol_english` as
'English' FROM `olevel_data`";
          pst=conn.prepareStatement(sql);
          rs=pst.executeQuery();
          tblol.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));


      } catch (Exception e) {
          JOptionPane.showConfirmDialog(rootPane, e);
      }
    //  theader();
    }


    // AL data


    private void getaLData()
    {
      alexamNumber = txtalexnumber.getText();
      alas = txtalas.getText();
      albs = txtalbs.getText();
      alcs = txtalcs.getText();
      alss = txtalss.getText();
      stream = cmbalstream.getSelectedItem().toString();
    }


     private void clearALData()
    {
      txtalexnumber.setText("");
      txtalas.setText("");
      txtalbs.setText("");
      txtalcs.setText("");
      txtalss.setText("");
      cmbalstream.setSelectedIndex(0);


    }
     private void alDataTablelord()
    {
      try {
          String sql="SELECT `ep_number` as 'EP Number', `name` as 'Name',
`ex_number` as 'Exam Number', `al_stream` as 'As Stream', `al_as` as 'A Greades',
`al_bs` as 'B Grades', `al_cs` as 'C Grades', `al_ss` as 'S Gredes' FROM
`alevel_data`";
          pst=conn.prepareStatement(sql);
          rs=pst.executeQuery();
          tblal.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));


      } catch (Exception e) {
          JOptionPane.showConfirmDialog(rootPane, e);
      }
```

```
 //  theader();
}

 // advance

 private void getADData()
{
   degreeType = cmbaddegtype.getSelectedItem().toString();
   degree = txtaddegree.getText();
   duration = cmbduration.getSelectedItem().toString();
   instute = txtadinstute.getText();


}


  private void clearADData()
{
   txtaddegree.setText("");
   txtadinstute.setText("");
   cmbaddegtype.setSelectedIndex(0);
   cmbduration.setSelectedIndex(0);
}
 private void aDDataTablelord()
{
   try {
```

```
            String sql="SELECT `ep_number` as 'EP Number', `name` as 'Name',
`degree_type` as 'Degree Type', `degree` as 'Degree Name', `duration` as 'Duration',
`instute` as 'Instute' FROM `degree_data`";

            pst=conn.prepareStatement(sql);

            rs=pst.executeQuery();

            tblad.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));


        } catch (Exception e) {

            JOptionPane.showConfirmDialog(rootPane, e);

        }

    //  theader();

    }


    // experiance


    private void getexData()

    {

        exmonth = txtexmonth.getText();

        exorg = txtexorg.getText();

        expostType = cmbpostType.getSelectedItem().toString();




    }


     private void clearexData()

    {
      txtexmonth.setText("");

      txtexorg.setText("");

      cmbpostType.setSelectedIndex(0);


   }

    private void exDataTablelord()

   {

      try {

         String sql="SELECT `ep_number` as 'EP Number', `name` as 'Name',
`post_type` as 'Post Type', `month` as 'Month', `organization` as 'Organization'
FROM `experiance_data`";

         pst=conn.prepareStatement(sql);

         rs=pst.executeQuery();

         tblex.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));


      } catch (Exception e) {

         JOptionPane.showConfirmDialog(rootPane, e);

      }

    //  theader();

   }




    // shedule


    private void getshData()

   {
```

```
shpost = txtshpost.getText();
shjobtype = txtshjobtype.getText();
shgender = cmbshgender.getSelectedItem().toString();
shagemin = txtshagemin.getText();
shagemax = txtshagemax.getText();
sholas = txtsholas.getText();
sholbs = txtsholbs.getText();
sholcs = txtsholcs.getText();
sholss = txtsholss.getText();
shalstream = cmbshalstream.getSelectedItem().toString();
shalas = txtshalas.getText();
shalbs = txtshalbs.getText();
shalcs = txtshalcs.getText();
shalss = txtshalss.getText();
shdegtype = cmbshgdegreetype.getSelectedItem().toString();
shdegduration = cmbshdegreeduration.getSelectedItem().toString();
shdegcategory = cmbshdegcategory.getSelectedItem().toString();

sholmaths = txtsholmaths.getText();
sholenglish = txtsholenglish.getText();

if(chkol.isSelected())
{
   olcheck=1;
}
 if(chkal.isSelected())
 {
    alcheck=1;
 }
  if(chkdeg.isSelected())
 {
    degcheck=1;
 }



}


 private void clearshData()
{


 txtshpost.setText("");
 txtshjobtype.setText("");
 cmbshgender.setSelectedIndex(0);
 txtshagemin.setText("");
 txtshagemax.setText("");
 txtsholas.setText("");
 txtsholbs.setText("");
 txtsholcs.setText("");
 txtsholss.setText("");
 cmbshalstream.setSelectedIndex(0);
 txtshalas.setText("");
 txtshalbs.setText("");
```

```
    txtshalcs.setText("");

    txtshalss.setText("");

    cmbshgdegreetype.setSelectedIndex(0);

    cmbshdegreeduration.setSelectedIndex(0);

    cmbshdegcategory.setSelectedIndex(0);

    txtsholmaths.setText("");

    txtsholenglish.setText("");


    }

    private void shDataTablelord()

    {

      try {

          String sql="SELECT `post` as 'Job Post', `job_type` as 'Job Type', `gender`
as 'Gender', `age_min` as 'Min Age', `age_max` as 'Max Age', `ol_as` as 'O/L As',
`ol_bs` as 'O/L Bs', `ol_cs` as 'O/L Cs',"

                + " `ol_ss` as 'O/L Ss', `ol_maths` as 'O/L Math', `ol_english` as 'O/L
English', `al_stream` as 'A/L Stream', `al_as` as 'A/L As', `al_bs` as 'A/L Bs',
`al_cs` as 'A/L Cs', `al_ss` as 'A/L Ss',"

                + " `deg_type` as 'Degree Type', `deg_duration` as 'Duration', "

                + "`deg_category` as 'Degree Category' FROM `shedule`";

          pst=conn.prepareStatement(sql);

          rs=pst.executeQuery();

          tblshedule.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));


      } catch (Exception e) {

          JOptionPane.showConfirmDialog(rootPane, e);

      }
```

```
 //  theader();
}




 // select employee


  private void postpelord()
{



    String sq="SELECT `post` FROM `shedule`";
       try {
        pst=conn.prepareStatement(sq);
        rs=pst.executeQuery();
        while(rs.next())
        {
        String name=rs.getString("post");
        cmbsepost.addItem(name);


        }
     } catch (SQLException ex) {
       JOptionPane.showMessageDialog(rootPane, ex);
     }
}
```

```
private void olLoadDisable()
{

   txtsholas.setEnabled(false);
   txtsholbs.setEnabled(false);
   txtsholcs.setEnabled(false);
   txtsholss.setEnabled(false);
   txtsholmaths.setEnabled(false);
   txtsholenglish.setEnabled(false);

}
private void alLoadDisable()
{

   txtshalas.setEnabled(false);
   txtshalbs.setEnabled(false);
   txtshalcs.setEnabled(false);
   txtshalss.setEnabled(false);
   cmbshalstream.setEnabled(false);

}
private void degLoadDisable()
{

   cmbshdegcategory.setEnabled(false);
   cmbshdegreeduration.setEnabled(false);
```

```
        cmbshgdegreetype.setEnabled(false);



    }


    private void getValue()
    {


        String sq="SELECT `post`, `job_type`, `gender`, `age_min`, `age_max`,
`olcheck`, `ol_as`, `ol_bs`, "
                + "`ol_cs`, `ol_ss`, `ol_maths`, `ol_english`, `alcheck`, `al_stream`,
`al_as`, `al_bs`, "
                + "`al_cs`, `al_ss`, `degcheck`, `deg_type`, `deg_duration`,
`deg_category` "
                + "FROM `shedule` WHERE
post='"+cmbsepost.getSelectedItem().toString()+"'";
            try {
             pst=conn.prepareStatement(sq);
             rs=pst.executeQuery();
             while(rs.next())
             {
             olvalue =rs.getInt("olcheck");
             alvalue=rs.getInt("alcheck");
             degvalue=rs.getInt("degcheck");

             agemin = rs.getInt("age_min");
             agemax= rs.getInt("age_max");
        selectolmath = rs.getString("ol_maths");
        selectolenglish = rs.getString("ol_english");
        selectalstream = rs.getString("al_stream");
        selectdegtype = rs.getString("deg_type");



         }
      } catch (SQLException ex) {
        JOptionPane.showMessageDialog(rootPane, ex);
      }

    }

    private void empbasicLoad()
    {

       String sq="SELECT\n" +
"personal_data.`name`,\n" +
"personal_data.gender,\n" +
"personal_data.nic,\n" +
"personal_data.adress,\n" +
"personal_data.contact_num,\n" +
"personal_data.distric,\n" +
"personal_data.age\n" +
"FROM\n" +
"personal_data\n" +
"WHERE\n" +
"personal_data.age< '"+agemax+"' AND\n" +
"personal_data.age> '"+agemin+"'";
          try {
           pst=conn.prepareStatement(sq);
          rs=pst.executeQuery();

tblselectemp.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));
        } catch (SQLException ex) {
          JOptionPane.showMessageDialog(rootPane, ex);
        }


    }


      private void empolLoad()
    {


        String sq="SELECT personal_data.name, personal_data.gender,
personal_data.nic, personal_data.adress, personal_data.contact_num,
personal_data.distric, personal_data.age FROM personal_data INNER JOIN
olevel_data ON personal_data.ep_number = olevel_data.ep_number WHERE
personal_data.age< '"+agemax+"' AND personal_data.age> '"+agemin+"' AND
olevel_data.ol_math = '"+selectolmath+"'AND olevel_data.ol_english =
'"+selectolenglish+"'";
          try {
           pst=conn.prepareStatement(sq);
          rs=pst.executeQuery();
tblselectemp.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));
        } catch (SQLException ex) {
          JOptionPane.showMessageDialog(rootPane, ex);
        }


      }


        private void empalLoad()
      {


        String sq="SELECT personal_data.name, personal_data.gender,
personal_data.nic, personal_data.adress, personal_data.contact_num,
personal_data.distric, personal_data.age FROM personal_data INNER JOIN
olevel_data ON personal_data.ep_number = olevel_data.ep_number INNER JOIN
alevel_data ON personal_data.ep_number = alevel_data.ep_number WHERE
personal_data.age< '"+agemax+"' AND personal_data.age> '"+agemin+"' AND
olevel_data.ol_math = '"+selectolmath+"'AND olevel_data.ol_english =
'"+selectolenglish+"' AND al_stream='"+selectalstream+"'";
          try {
           pst=conn.prepareStatement(sq);
          rs=pst.executeQuery();

tblselectemp.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));
        } catch (SQLException ex) {
          JOptionPane.showMessageDialog(rootPane, ex);
        }


      }
```

```
        private void empdegLoad()
      {


        String sq="SELECT personal_data.name, personal_data.gender,
personal_data.nic, personal_data.adress, personal_data.contact_num,
personal_data.distric, personal_data.age FROM personal_data INNER JOIN
olevel_data ON personal_data.ep_number = olevel_data.ep_number INNER JOIN
alevel_data ON personal_data.ep_number = alevel_data.ep_number INNER JOIN
degree_data ON personal_data.ep_number = degree_data.ep_number WHERE
personal_data.age< '"+agemax+"' AND personal_data.age> '"+agemin+"' AND
olevel_data.ol_math = '"+selectolmath+"'AND olevel_data.ol_english =
'"+selectolenglish+"' AND al_stream='"+selectalstream+"' AND degree_type =
'"+selectdegtype+"'";
          try {
           pst=conn.prepareStatement(sq);
          rs=pst.executeQuery();

tblselectemp.setModel(net.proteanit.sql.DbUtils.resultSetToTableModel(rs));
        } catch (SQLException ex) {
          JOptionPane.showMessageDialog(rootPane, ex);
        }


      }
    // select employee




   /**
```

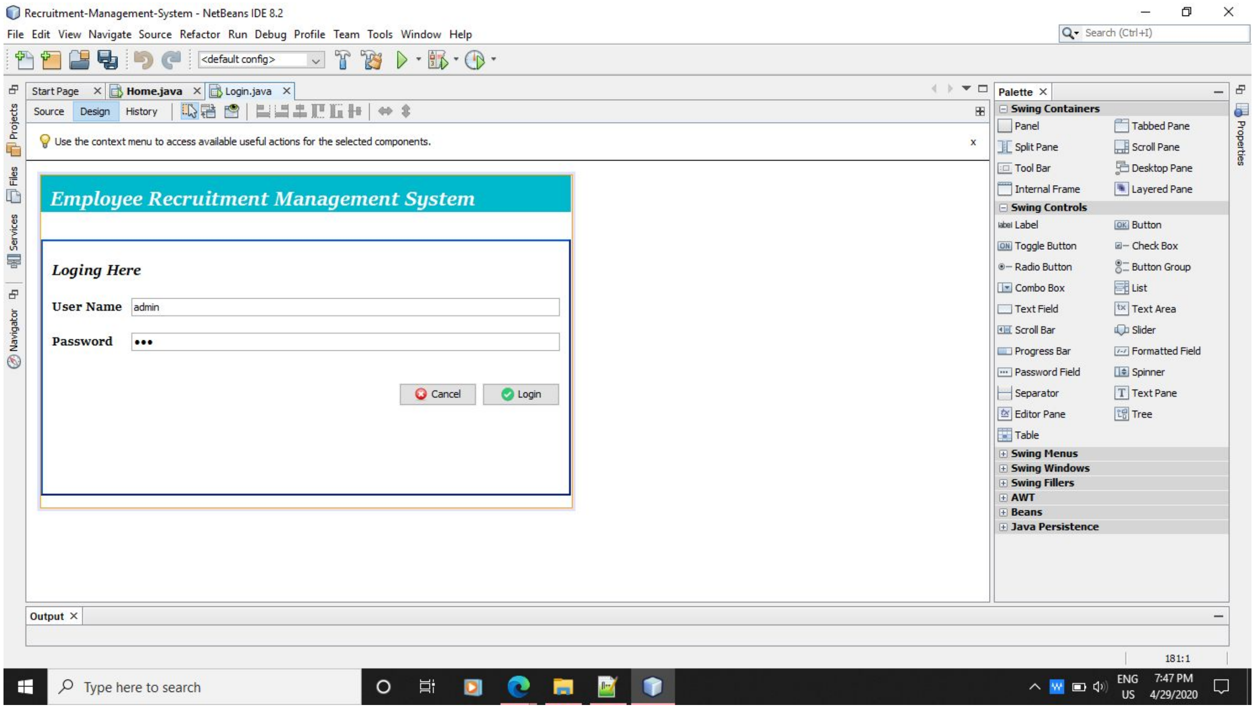
```
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default
look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
        }
        //</editor-fold>


        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Home().setVisible(true);
            }
        });
    }


    // Variables declaration - do not modify
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JCheckBox chkal;
    private javax.swing.JCheckBox chkdeg;
    private javax.swing.JCheckBox chkol;
    private javax.swing.JComboBox cmbaddegtype;
    private javax.swing.JComboBox cmbalstream;
    private javax.swing.JComboBox cmbduration;
    private javax.swing.JComboBox cmbolenglish;
    private javax.swing.JComboBox cmbolmaths;
    private javax.swing.JComboBox cmbpdcity;
    private javax.swing.JComboBox cmbpostType;
    private javax.swing.JComboBox cmbsepost;
    private javax.swing.JComboBox cmbshalstream;
    private javax.swing.JComboBox cmbshdegcategory;
    private javax.swing.JComboBox cmbshdegreeduration;
    private javax.swing.JComboBox cmbshgdegreetype;
    private javax.swing.JComboBox cmbshgender;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton10;
    private javax.swing.JButton jButton11;
    private javax.swing.JButton jButton12;
    private javax.swing.JButton jButton13;
    private javax.swing.JButton jButton14;
    private javax.swing.JButton jButton15;
    private javax.swing.JButton jButton16;
    private javax.swing.JButton jButton17;
    private javax.swing.JButton jButton18;
    private javax.swing.JButton jButton19;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton20;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JButton jButton7;
    private javax.swing.JButton jButton8;
    private javax.swing.JButton jButton9;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel19;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel20;
    private javax.swing.JLabel jLabel21;
    private javax.swing.JLabel jLabel22;
    private javax.swing.JLabel jLabel23;
    private javax.swing.JLabel jLabel24;
    private javax.swing.JLabel jLabel25;
    private javax.swing.JLabel jLabel26;
    private javax.swing.JLabel jLabel27;
    private javax.swing.JLabel jLabel28;
    private javax.swing.JLabel jLabel29;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel30;
    private javax.swing.JLabel jLabel31;
    private javax.swing.JLabel jLabel32;
    private javax.swing.JLabel jLabel33;
    private javax.swing.JLabel jLabel34;
    private javax.swing.JLabel jLabel35;
    private javax.swing.JLabel jLabel36;
    private javax.swing.JLabel jLabel37;
    private javax.swing.JLabel jLabel38;
    private javax.swing.JLabel jLabel39;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel40;
    private javax.swing.JLabel jLabel41;
    private javax.swing.JLabel jLabel42;
    private javax.swing.JLabel jLabel43;
    private javax.swing.JLabel jLabel44;
    private javax.swing.JLabel jLabel45;
    private javax.swing.JLabel jLabel46;
    private javax.swing.JLabel jLabel47;
    private javax.swing.JLabel jLabel48;
    private javax.swing.JLabel jLabel49;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel50;
    private javax.swing.JLabel jLabel51;
    private javax.swing.JLabel jLabel52;
    private javax.swing.JLabel jLabel53;
    private javax.swing.JLabel jLabel54;
    private javax.swing.JLabel jLabel55;
    private javax.swing.JLabel jLabel56;
    private javax.swing.JLabel jLabel57;
    private javax.swing.JLabel jLabel58;
    private javax.swing.JLabel jLabel59;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel60;
    private javax.swing.JLabel jLabel61;
    private javax.swing.JLabel jLabel62;
    private javax.swing.JLabel jLabel64;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel12;
    private javax.swing.JPanel jPanel13;
    private javax.swing.JPanel jPanel14;
    private javax.swing.JPanel jPanel15;
    private javax.swing.JPanel jPanel16;
    private javax.swing.JPanel jPanel17;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JPanel jPanel5;
    private javax.swing.JPanel jPanel6;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JScrollPane jScrollPane4;
    private javax.swing.JScrollPane jScrollPane5;
    private javax.swing.JScrollPane jScrollPane6;
    private javax.swing.JScrollPane jScrollPane7;
    private javax.swing.JScrollPane jScrollPane8;
    private javax.swing.JScrollPane jScrollPane9;
    private javax.swing.JTabbedPane jTabbedPane1;
    private javax.swing.JTabbedPane jTabbedPane2;
    private javax.swing.JTextField jTextField47;
    private javax.swing.JLabel lbldate;
    private javax.swing.JLabel lbltime;
    private javax.swing.JRadioButton rbtfemale;
    private javax.swing.JRadioButton rbtmale;
    private javax.swing.JTable tblad;
    private javax.swing.JTable tblal;
    private javax.swing.JTable tblex;
    private javax.swing.JTable tblol;
    private javax.swing.JTable tblpd;
    private javax.swing.JTable tblselectemp;
    private javax.swing.JTable tblshedule;
    private javax.swing.JTextField txtaddegree;
    private javax.swing.JTextField txtadepnumber;
    private javax.swing.JTextField txtadinstute;
    private javax.swing.JTextField txtadname;
    private javax.swing.JTextArea txtadress;
```

```
    private javax.swing.JTextField txtadsearch;
    private javax.swing.JTextField txtalas;
    private javax.swing.JTextField txtalbs;
    private javax.swing.JTextField txtalcs;
    private javax.swing.JTextField txtalepnumber;
    private javax.swing.JTextField txtalexnumber;
    private javax.swing.JTextField txtalname;
    private javax.swing.JTextField txtalsearch;
    private javax.swing.JTextField txtalss;
    private javax.swing.JTextField txtexepnumber;
    private javax.swing.JTextField txtexmonth;
    private javax.swing.JTextField txtexname;
    private javax.swing.JTextField txtexorg;
    private javax.swing.JTextField txtexsearch;
    private javax.swing.JTextField txtolas;
    private javax.swing.JTextField txtolbs;
    private javax.swing.JTextField txtolcs;
    public static final javax.swing.JTextField txtolepnumber = new
javax.swing.JTextField();
    private javax.swing.JTextField txtolexamnumber;
    private javax.swing.JTextField txtolname;
    private javax.swing.JTextField txtolsearch;
    private javax.swing.JTextField txtolss;
    private javax.swing.JTextField txtpdage;
    private javax.swing.JTextField txtpdcontactnumber;
    private javax.swing.JTextField txtpdepnumber;
    private javax.swing.JTextField txtpdname;
    private javax.swing.JTextField txtpdnic;
    private javax.swing.JTextField txtpdsearch;
    private javax.swing.JTextField txtshagemax;
    private javax.swing.JTextField txtshagemin;
    private javax.swing.JTextField txtshalas;
    private javax.swing.JTextField txtshalbs;
    private javax.swing.JTextField txtshalcs;
    private javax.swing.JTextField txtshalss;
    private javax.swing.JTextField txtshjobtype;
    private javax.swing.JTextField txtsholas;
    private javax.swing.JTextField txtsholbs;
    private javax.swing.JTextField txtsholcs;
    private javax.swing.JTextField txtsholenglish;
    private javax.swing.JTextField txtsholmaths;
    private javax.swing.JTextField txtsholss;
    private javax.swing.JTextField txtshpost;
    private javax.swing.JTextField txtshsearch;
    // End of variables declaration
}
```

RESULT

LOGIN PAGE:



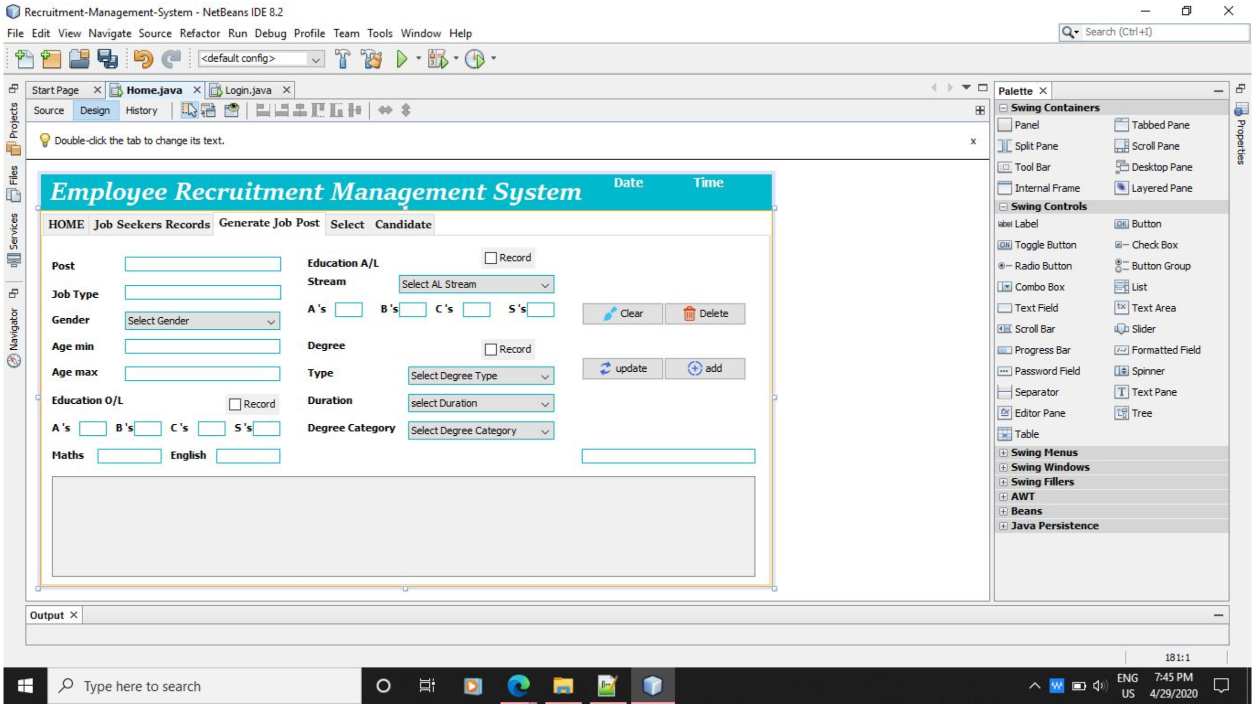
Employee Recruitment Management System

Loging Here

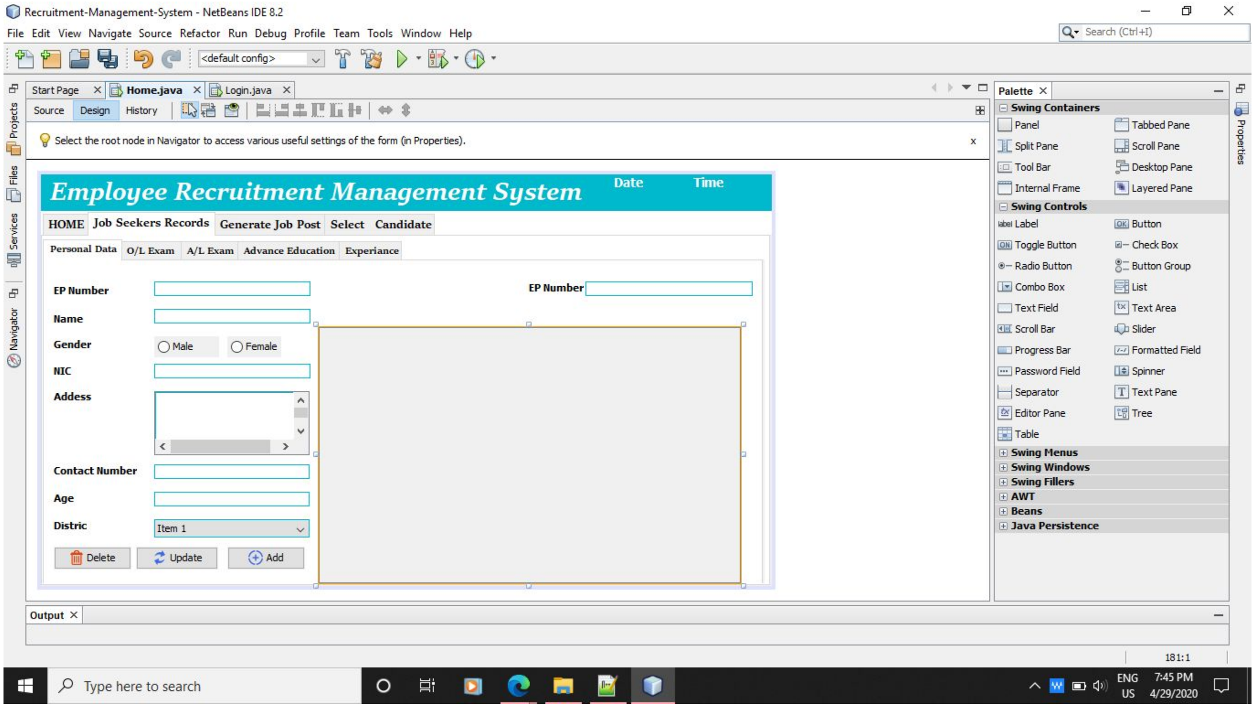
User Name admin

Password •••

CANDIDATE DETAILS:


Recruitment-Management-System - NetBeans IDE 8.2

Double-click the tab to change its text.


Recruitment-Management-System - NetBeans IDE 8.2

Select the root node in Navigator to access various useful settings of the form (in Properties).

CONCLUSION

Online Recruitment Management System can be used by employers to recruit the candidates based on their experience and the further interviews easily. Achieving this objective is difficult using the manual system as the information is scattered, can be redundant and collecting relevant information may be very time-consuming.

All these problems are solved by this project. This system helps in maintaining the information of potential candidates at one place. It can be easily accessed by both applicant and employers.

It is kept safe for a long period of time without any changes or omission. It reduces the time employer take to make few recruitments in their firm. This system helps the job seekers to get every information required for the process.