# Project/Dissertation Report

on

## Blogpost Website

***Submitted in partial fulfillment of the requirement for the award of the degree of***

# B. Tech CNCS


**Under The Supervision**
**of**
**Mr. Dhruv Kumar**

Submitted By

**Karan Kapoor**

18SCSE1140028

**Vikrant Chauhan**

18SCSE1140039

**Project ID: BT 4235**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERINGGALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA**

**May,**
**2021-2022**

# CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"Blogpost Website"** in partial fulfillment of the requirements for the award of the Bachelor of technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of January 2022 to May and 2022, under the supervision of Mr. Dhruv Kumar, Assistant Professor, Department of Computer Science and Engineering and Information and Science, of School of Computing Science and Engineering, Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Karan Kapoor (18SCSE1140028)

Vikrant Chauhan ( 18SCSE1140039 )

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. Dhruv Kumar

Assistant professor

## CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Karan Kapoor 18SCSE1140028, Vikrant Chauhan 18SCSE1140039 has been held on _____ and his/her work is recommended for the award of Bachelor of Technology

**Signature of Examiner(s)** **Signature of Supervisor(s)**

**Signature of Project Coordinator** **Signature of Dean**

Date: May,2022

Place : Greater Noida

# Table of content

## Acknowledgement

We take this occasion to thank God, almighty for blessing us with his grace and taking ourendeavor to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guide, Mr. Dhruv Kumar for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We would like to thank the other faculty members also, at this occasion. Last but not the least, we would like to thank friends for the support and encouragement they have given us during the course of our work.

## Abstract

As the name implies "Blog Post Website" is an educational website where there will be many blogs from a variety of topics. The main purpose of our website is to create a powerful platform where users can read various blogs and at the same time, users can publish their content or blogs on the website by creating an account, other users can log in / sign in. They can engage by doing likes, commenting on blogs. We will be adding this feature to increase engagement in our platform. Our Blogpost website is a website that is used for the chronological listing of the blog posts. Contains the most recent content first followed by previously updated content. In this there are many sections and posts written by a large number of authors. People can just create an account and start publishing their blogs. Blogs are a new and exciting way to communicate and to learn. Personal talk has evolved into a new and useful tool. Blogs have become an important form of emotional release and information for a growing number of people. With this website we are trying to enhance the learning perspective through blogs.

# Introduction

Our Blogpost website is basically a website which is used for chronological listing of blog posts. Contains the most recent content first followed by previously updated content. In this there are many sections and posts written by a large number of authors. People can just create an account and start publishing their blogs. Although the publication of blogs is subject to effective authorization from management. Viewers can read and engage with blogs by liking and commenting on them. This will increase interaction and give users a sense of interaction while browsing our site. Content will not be limited to certain popular topics; content writers can publish blogs of various domains to increase diversity to attract viewers of diverse interests.

Internet usage has increased dramatically and rapidly over the past decade. Websites have become a very important social networking site for many, if not all, businesses and organizations. Improperly designed websites can confuse users and cause a "jump rate", or people who visit the login page without checking other pages within the site. On the other hand, a well-designed and easy-to-use website has been found to have a positive impact on visitor retention (re-visit rates) and shopping behavior.

One of the most important steps in design is usability. The International Standardized Organization (ISO) defines usability as the degree to which users can access desirable functions (e.g., access to desirable information or purchases) effectively (completeness and accuracy of work), efficiency (time spent on work), and satisfaction (user experience) within the system.

In fact, before developing student writing tools, it is important to create an enabling environment for learning writing skills using a collaborative and deceptive approach. Not only can students improve their writing skills through blogging processes, they can build their confidence as writers can attract a wider audience. Over time, readers with various interests can become well-known bloggers with a large audience and can take ownership of their blog-based writing. Additionally, blogging processes play an important role in encouraging students to explore, take risks and promote their awareness to become independent and public writers. Through blogging, students can express their writing freely and can integrate different ideas when reading or publishing. This helps them to gain a solid sense of analysis and interpretation of the topics they write about.

Surprisingly, digitalization has successfully transformed the learning process. Blogging is one of the new tools used in education. Etymologically, 'blog' is a compound word 'web' and 'log'. Simply put, a 'blog' is a web page that contains multimedia, comments and links. Unlike online websites with text and images uploaded by web developers, which do not have the space for any interaction and communication, blogs provide access to readers to express their writing ideas and share their online writing skills. Blogging

quickly became popular among language readers supported by web designers and developers. This was quick and easy because using blogging for writing purposes required minimal technical complexity; in addition, blogs were available to different types of users and audiences. The rapid growth in the interest in blogging is reflected in the millions of people who write or read blogs. More interestingly, blogs are customized; this allows users to customize their presentation templates, themes, images, designs, colors and a few other options.

## Problem Stated

- In the existing system the blogs are posted manually by the admins or some hired content writers.
- Lack of security of data.
- More man power.
- Time consuming.
- Delay in publishing and not cost effective.
- Consumes large volume of pare work.
- No direct role for the higher officials.

## Problem Solution

- The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system.
- It will reduce the manual work. Ensure data accuracy. Content published will be firstly approved by the admins.
- Proper control of the higher officials. Minimize manual data entry by hired professionals.
- Normal users can publish their blogs, this would give them a homely experience.
- Minimum time needed for the various processing.
- Greater efficiency. Better service.
- User friendliness and interactive.

## Tools and Technology

- Hardware Configuration

  Pentium IV Processor 512

  MB RAM

  40GB HDD

  1024 * 768 Resolution Color Monitor

  **Note:** This is not the "System Requirements".

- **Softwares :**

· Eclipse IDE for development, MySQL workbench for Database and Tomcat server for Server Purposes.

This website will be made using different technologies like:

- **Technology :**

### 1. HTML

HTML (Hypertext Markup Language) is the code that is used to structure a web page and its content.
HTML is a markup language that defines the structure of your content. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way. The enclosing tags can make a word or image hyperlink to somewhere else, can italicize words, can make the font bigger or smaller, and so on. For example, take the following line of content:
My cat is very grumpy
If we wanted the line to stand by itself, we could specify that it is a paragraph by enclosing it in paragraph tags:
My cat is very grumpy

The main parts of our element are as follows:
1. The opening tag: This consists of the name of the element (in this case, p), wrapped in opening and closing angle brackets. This states where the element begins or starts to take effect — in this case where the

paragraph begins.

2. The closing tag: This is the same as the opening tag, except that it includes a forward slash before the element name. This states where the element ends — in this case where the paragraph ends. Failing to add a closing tag is one of the standard beginner errors and can lead to strange results.

3. The content: This is the content of the element, which in this case, is just text.

4. The element: The opening tag, the closing tag, and the content together comprise the element.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph. </p>

</body>
</html>
```

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

## 2. CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document

written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, development of various parts of CSS specification was done synchronously, which allowed versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, CSS3. However, CSS4 has never become an official version.

HTML was NEVER intended to contain tags for formatting a web page!

HTML was created to describe the content of a web page, like:

<h1>This is a heading</h1>

This is a paragraph. 

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

CSS removed the style formatting from the HTML page!

The style definitions are normally saved in external .CSS files.

With an external stylesheet file, you can change the look of an entire website by changing just one file!

Example

```
p {
   color: red;
   text-align: center;
}
```

Example Explained

- p is a selector in CSS (it points to the HTML element you want to style: ).
- color is a property, and red is the property value
- text-align is a property, and center is the property value

### 3. Bootstrap

Bootstrap is a free and open-source front end development framework for the creation of websites and web apps. The Bootstrap framework is built on HTML, CSS, and JavaScript (JS) to facilitate the development of responsive, mobile-first sites and apps. Bootstrap 5 is the newest version of Bootstrap. Bootstrap 5 supports

all major browsers except Internet Explorer 11 and down.

Responsive design makes it possible for a web page or app to detect the visitor's screen size and orientation and automatically adapt the display accordingly; the mobile first approach assumes that smartphones, tablets and task-specific Mobile apps are employees' primary tools for getting work done and addresses the requirements of those technologies in design.

Bootstrap includes user interface components, layouts and JS tools along with the framework for implementation. The software is available precompiled or as source code.

### 4. SQL

Basically, SQL stands for Structured Query Language which is basically a language used by databases. This language allows to handle the information using tables and shows a language to query these tables and other objects related (views, functions, procedures, etc.). Most of the databases like SQL Server, Oracle, PostgreSQL, MySQL, MariaDB handle this language (with some extensions and variations) to handle the data.

With SQL you can insert, delete, and update data. You can also create, delete, or alter database objects.

What is SQL?

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

What Can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database

- SQL can set permissions on tables, procedures, and views

### 4.1 RDBMS

RDBMS stands for Relational Database Management System.

RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.

Example

SELECT * FROM Customers;

Every table is broken up into smaller entities called fields. The fields in the Customers table consist of CustomerID, CustomerName, ContactName, Address, City, PostalCode and Country. A field is a column in a table that is designed to maintain specific information about every record in the table.

A record, also called a row, is each individual entry that exists in a table. For example, there are 91 records in the above Customers table. A record is a horizontal entity in a table.

A column is a vertical entity in a table that contains all information associated with a specific field in a table.

## 5. Java

What is Java?

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than 3 billion devices run Java.

It is used for:

- Mobile applications (especially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

### 5.1 Why Use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming languages in the world
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful
- It has a huge community support (tens of millions of developers)
- Java is an object-oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa.

### 5.2 How java works?

Java is a technology consisting of both a programming language and a software platform. To create an application using Java, you need to download the Java Development Kit (JDK), which is available for Windows, macOS, and Linux. You write the program in the Java programming language, then a compiler turns the program into Java bytecode—the instruction set for the Java Virtual Machine (JVM) that is a part of the Java runtime environment (JRE). Java bytecode runs without modification on any system that supports JVMs, allowing your Java code to be run anywhere.

The Java software platform consists of the JVM, the Java API, and a complete development environment. The JVM parses and runs (interprets) the Java bytecode. The Java API consists of an extensive set of libraries including basic objects, networking and security functions; Extensible Markup Language (XML) generation; and web services. Taken together, the Java language and the Java software platform create a powerful, proven technology for enterprise software development.

## 6. Spring core

Introduction to Spring Framework

Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java applications. Spring handles the infrastructure so you can focus on your application.

Spring enables you to build applications from “plain old Java objects” (POJOs) and to apply enterprise services non-invasively to POJOs. This capability applies to the Java SE programming model and to full and partial Java EE.

Examples of how you, as an application developer, can use the Spring platform advantage:

- Make a Java method execute in a database transaction without having to deal with transaction APIs.
- Make a local Java method a remote procedure without having to deal with remote APIs.
- Make a local Java method a management operation without having to deal with JMX APIs.
- Make a local Java method a message handler without having to deal with JMS APIs.

## 7. JDBC

Java database connectivity (JDBC) is the Java Soft specification of a standard application programming interface (API) that allows Java programs to access database management systems. The JDBC API consists of a set of interfaces and classes written in the Java programming language.

Using these standard interfaces and classes, programmers can write applications that connect to databases, send queries written in structured query language (SQL), and process the results.

Since JDBC is a standard specification, one Java program that uses the JDBC API can connect to any database management system (DBMS), as long as a driver exists for that particular DBMS.

What Are the Types of JDBC Drivers?

Today, there are five types of JDBC drivers in use:

- Type 1: JDBC-ODBC bridge
- Type 2: partial Java driver
- Type 3: pure Java driver for database middleware
- Type 4: pure Java driver for direct-to-database
- Type 5: highly-functional drivers with superior performance

What Type of JDBC Driver Should You Use?

For most applications, the best choice is a pure Java driver, either Type 3, Type 4, or even Type 5.

Type 5 JDBC drivers (such as Data Direct JDBC drivers) offer advanced functionality and superior performance over other driver types.

Type 4 drivers are the most common and are designed for a particular vendor's database.

In contrast, Type 3 is a single JDBC driver used to access a middleware server, which, in turn, makes the relevant calls to the database. A good example of Type 3 JDBC driver is the Data Direct Sequel ink JDBC driver.

Type 1 JDBC drivers are used for testing JDBC applications against an ODBC data source. Type 2 JDBC drivers require a native database API to be used. Both Type 1 and Type 2 JDBC driver types mix a Java-based API with another API.

The following figure shows a side-by-side comparison of the implementation of each of the JDBC driver types. All four implementations show a Java application or applet using the JDBC API to communicate through the JDBC Driver Manager with a specific JDBC driver type

object-relational mapping (ORM)

Object-relational mapping (ORM) is a mechanism that makes it possible to address, access and manipulate objects without having to consider how those objects relate to their data sources. ORM lets programmers maintain a consistent view of objects over time, even as the sources that deliver them, the sinks that receive them and the applications that access them change.

Based on abstraction, ORM manages the mapping details between a set of objects and underlying relational databases, XML repositories or other data sources and sinks, while simultaneously hiding the often-changing details of related interfaces from developers and the code they create.

ORM hides and encapsulates change in the data source itself, so that when data sources or their APIs change, only ORM needs to change to keep up—not the applications that use ORM to insulate themselves from this kind of effort. This capacity lets developers take advantage of new classes as they become available and also makes it easy to extend ORM-based applications. In many cases, ORM changes can incorporate new technology and capability without requiring changes to the code for related applications.

## 8. Spring MVC Tutorial

A Spring MVC is a Java framework which is used to build web applications. It follows the Model-View-Controller design pattern. It implements all the basic features of a core spring framework like Inversion of Control, Dependency Injection.

A Spring MVC provides an elegant solution to use MVC in spring framework by the help of Dispatcher Servlet. Here, Dispatcher Servlet is a class that receives the incoming request and maps it to the right resource such as controllers, models, and views

Spring Web Model-View-Controller

- Model - A model contains the data of the application. A data can be a single object or a collection of objects.
- Controller - A controller contains the business logic of an application. Here, the @Controller annotation is used to mark the class as the controller.
- View - A view represents the provided information in a particular format. Generally, JSP+JSTL is

used to create a view page. Although spring also supports other view technologies such as Apache Velocity, Thyme leaf and Free Marker.

o Front Controller - In Spring Web MVC, the Dispatcher Servlet class works as the front controller. It is responsible to manage the flow of the Spring MVC application.

### 8.1 Advantages of Spring MVC Framework

Let's see some of the advantages of Spring MVC Framework: -

o Separate roles - The Spring MVC separates each role, where the model object, controller, command object, view resolver, Dispatcher Servlet, validator, etc. can be fulfilled by a specialized object.

o Light-weight - It uses light-weight servlet container to develop and deploy your application.

o Powerful Configuration - It provides a robust configuration for both framework and application classes that includes easy referencing across contexts, such as from web controllers to business objects and validators.

o Rapid development - The Spring MVC facilitates fast and parallel development.

o Reusable business code - Instead of creating new objects, it allows us to use the existing business objects.

o Easy to test - In Spring, generally we create JavaBeans classes that enable you to inject test data using the setter methods.

o Flexible Mapping - It provides the specific annotations that easily redirect the page.

## 9. Servlet

Servlet technology is used to create a web application (resides at server side and generates a dynamic web page).

Servlet technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. We have discussed these disadvantages below.

There are many interfaces and classes in the Servlet API such as Servlet, Generic Servlet, HTTP Servlet, Servlet Request, Servlet Response, etc.

What is a Servlet?

Servlet can be described in many ways, depending on the context.

o Servlet is a technology which is used to create a web application.

o Servlet is an API that provides many interfaces and classes including documentation.

o Servlet is an interface that must be implemented for creating any Servlet.

o Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.

o Servlet is a web component that is deployed on the server to create a dynamic web page.

**10. JSP**

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

- Advantages of JSP over Servlet

There are many advantages of JSP over the Servlet. They are as follows:

1) Extension to Servlet

JSP technology is the extension to Servlet technology. We can use all the features of the Servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.

32.8M

601

History of Java

Next

Stay

2) Easy to maintain

JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.

3) Fast Development: No need to recompile and redeploy

If JSP page is modified, we don't need to recompile and redeploy the project. The Servlet code needs to be updated and recompiled if we have to change the look and feel of the application.

4) Less code than Servlet

In JSP, we can use many tags such as action tags, JSTL, custom tags, etc. that reduces the code. Moreover, we can use EL, implicit objects, etc.

The Lifecycle of a JSP Page

The JSP pages follow these phases:

- Translation of JSP Page
- Compilation of JSP Page
- Class loading (the class loader loads class file)
- Instantiation (Object of the Generated Servlet is created).
- Initialization (the container invokes jspInit() method).
- Request processing (the container invokes _jspService() method).
- Destroy (the container invokes jspDestroy() method).

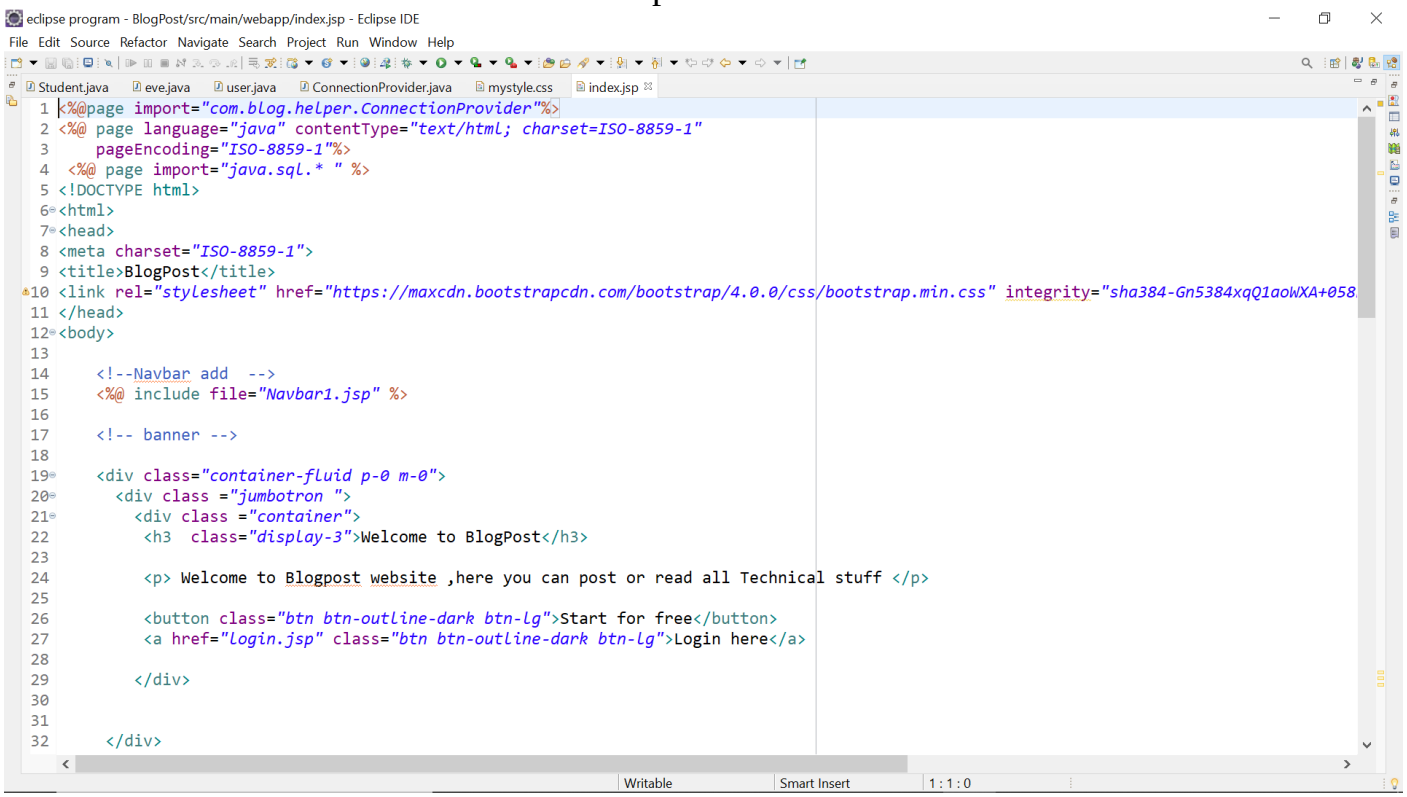
# Literature Survey

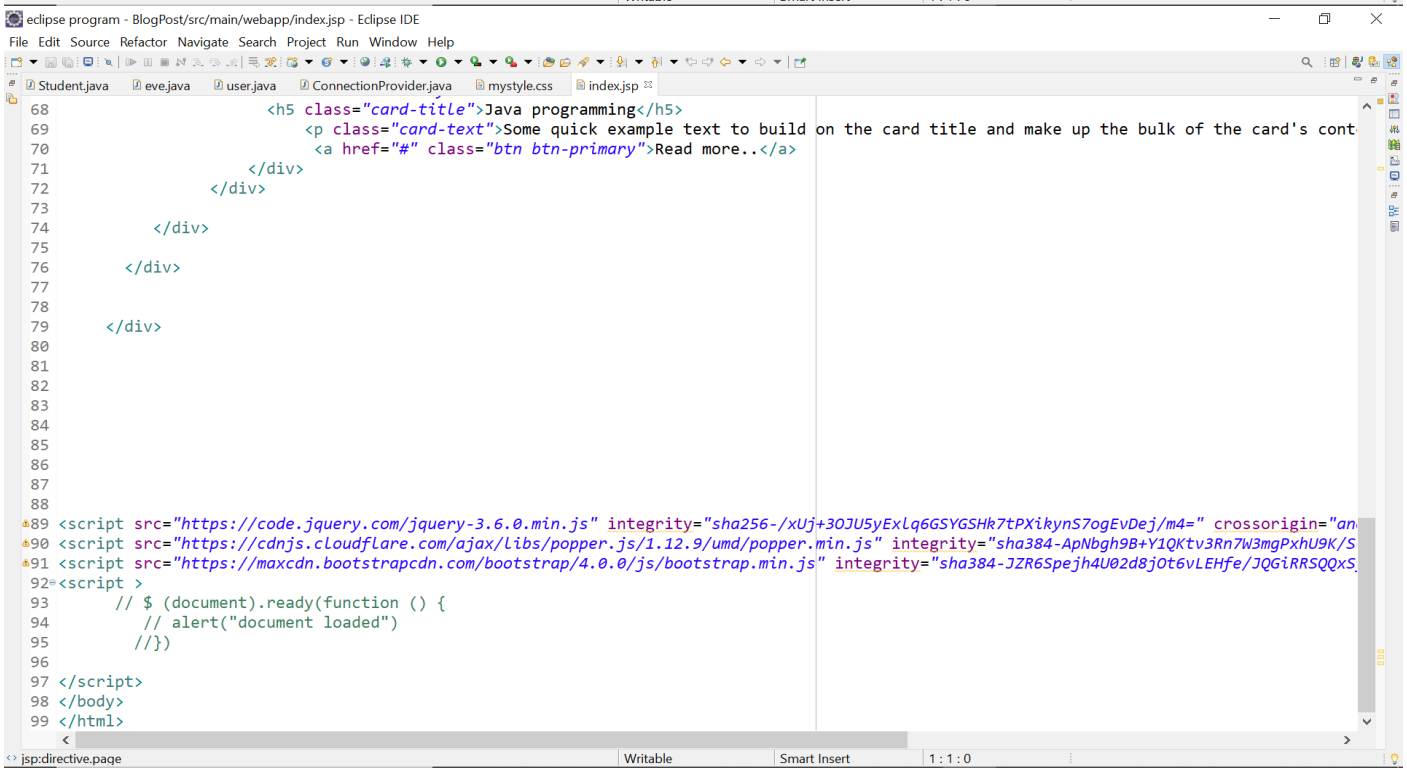
The interactive and attractive website "Blogpost" is developed using the latest technology and frameworks such as HTML, CSS, JavaScript, SQL, JDBC etc. Users will be able to create an account on the website, log in and read blogs. They can also like and comment on the blogs they want. If a user is a content writer and wants to publish their blogs then they should register and publish their blogs on our website but administrator approval will be required for the blog to be published for the final time. With our website, diverse readers or rather learners can often get help to learn new things as our site does not focus on a specific topic, the content will vary. Admins will be keeping an eye on the type of content that is being plagiarized and will be monitoring whether it is original or not. If plagiarism is more than 15%, blogs would not be published and if a same titled blog also exists, duplicating the same article will not be authorized until again unless further details or recent changes are made.

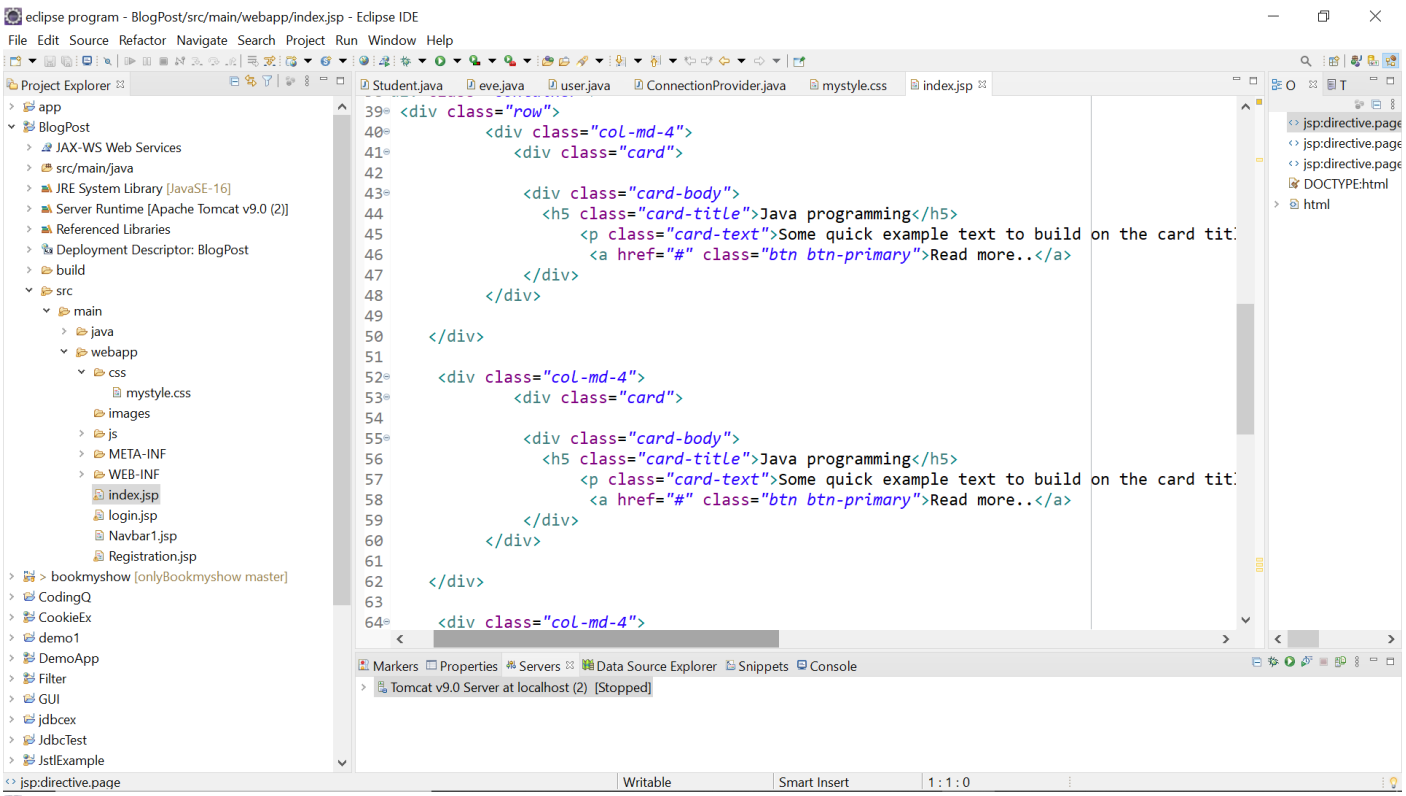
In addition to covering a lot of interests in blogs, bloggers are very slow, and deep, focused on interesting topics. Bloggers can work faster and communicate more deeply than mainstream media.

Blogging has recently gained a great deal of interest among teachers and students as a new way to teach writing in the classroom. Blogs focus on students' writing machines, empowering them, and giving them great writing power in the classroom. Blogging is a great way for students to connect with each other in a professionally community-based environment. Using blogs makes student writing more participatory and focuses on everyday language use. The study revealed that blogs play an important role in improving student appearance, classroom dialogue and social media. Blogs also develop writing skills for elementary, middle and high school students. The purpose of this research paper is to investigate the success of online blogging for the writing skills of each student or group and to ensure that through this Blogpost website we provide a platform for everyone to read, learn and write blogs. This paper also explores how blogs can help students improve their writing skills in non-traditional ways. Unlike traditional methods of teaching writing skills, blogging introduces students to do conversation, communication and speaking before the actual writing phase. Just as people are born with a natural desire to learn in a social setting that incorporates social skills, reading and writing involves all forms of communication that start at home and later develop continuously through communication with others, and blogging is an example. Blogs also encouraged interaction between students, and the use of blogging played a key role in understanding how students' response might work.

# Code part for UI :-



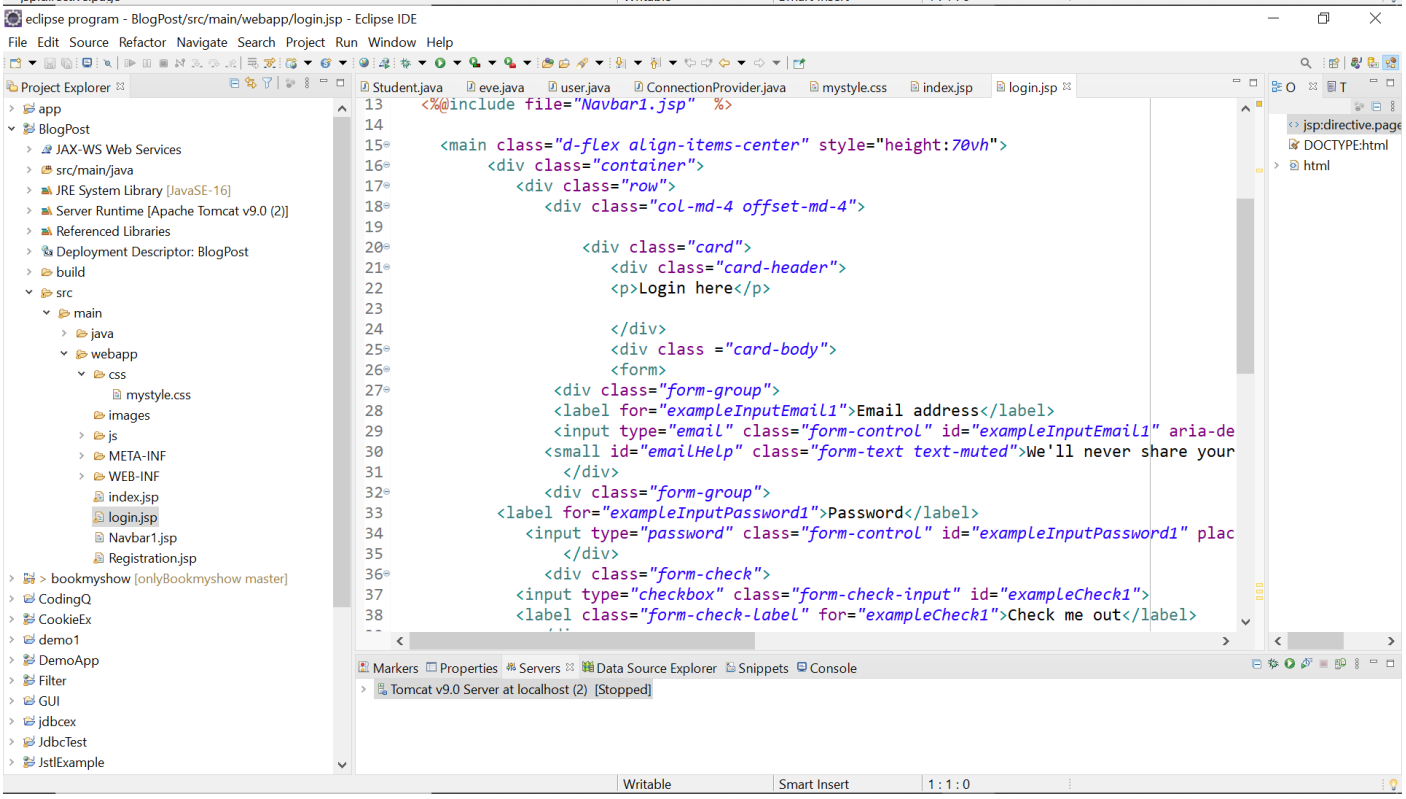




eclipse program - BlogPost/src/main/webapp/index.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
app
BlogPost
JAX-WS Web Services
src/main/java
JRE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0 (2)]
Referenced Libraries
Deployment Descriptor: BlogPost
build
src
main
java
webapp
css
mystyle.css
images
js
META-INF
WEB-INF
index.jsp
login.jsp
Navbar1.jsp
Registration.jsp
bookmyshow [onlyBookmyshow master]
CodingQ
CookieEx
demo1
DemoApp
Filter
GUI
jdbcex
JdbcTest
JstlExample
Student.java
eve.java
user.java
ConnectionProvider.java
mystyle.css
index.jsp

<h5 class="card-title">Java programming</h5>
Some quick example text to build on the card tit
<a href="#" class="btn btn-primary">Read more..</a>

<h5 class="card-title">Java programming</h5>
Some quick example text to build on the card tit
<a href="#" class="btn btn-primary">Read more..</a>

jsp:directive.page
jsp:directive.page
jsp:directive.page
DOCTYPE:html
html
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost (2) [Stopped]
jsp:directive.page
Writable
Smart Insert
1 : 1 : 0


eclipse program - BlogPost/src/main/webapp/login.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
app
BlogPost
JAX-WS Web Services
src/main/java
JRE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0 (2)]
Referenced Libraries
Deployment Descriptor: BlogPost
build
src
main
java
webapp
css
mystyle.css
images
js
META-INF
WEB-INF
index.jsp
login.jsp
Navbar1.jsp
Registration.jsp
bookmyshow [onlyBookmyshow master]
CodingQ
CookieEx
demo1
DemoApp
Filter
GUI
jdbcex
JdbcTest
JstlExample
Student.java
eve.java
user.java
ConnectionProvider.java
mystyle.css
index.jsp
login.jsp
<%@include file="Navbar1.jsp" %>
<main class="d-flex align-items-center" style="height:70vh">

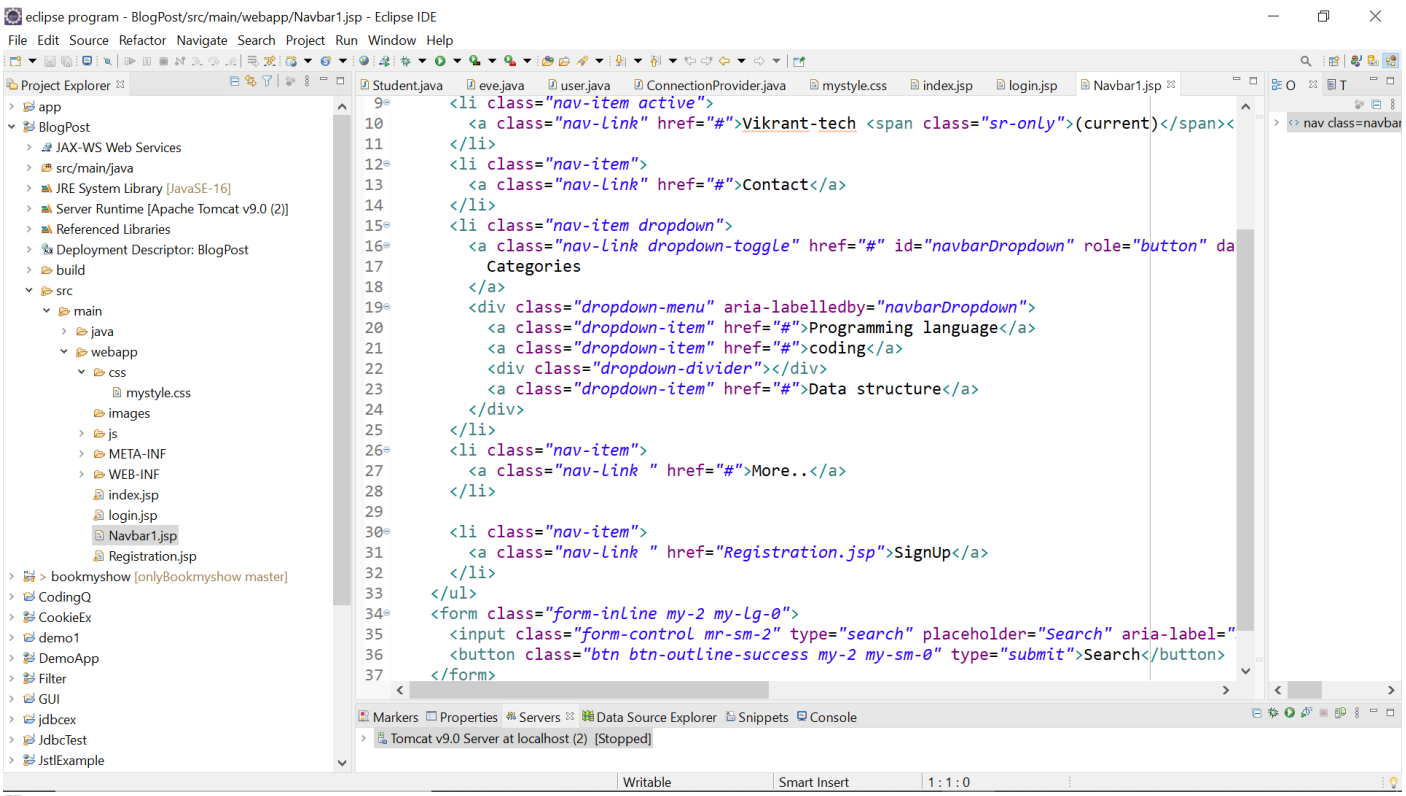
Login here

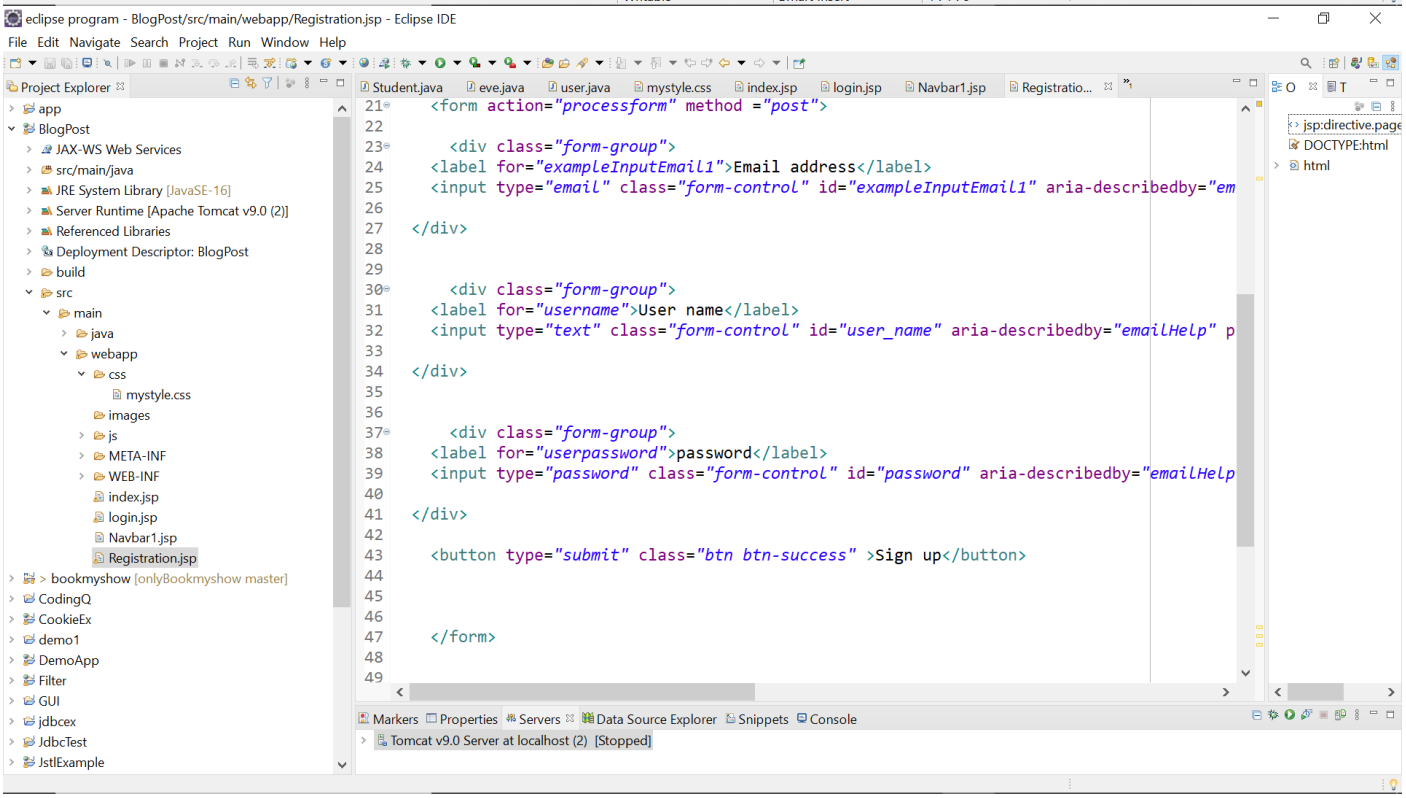
<form>

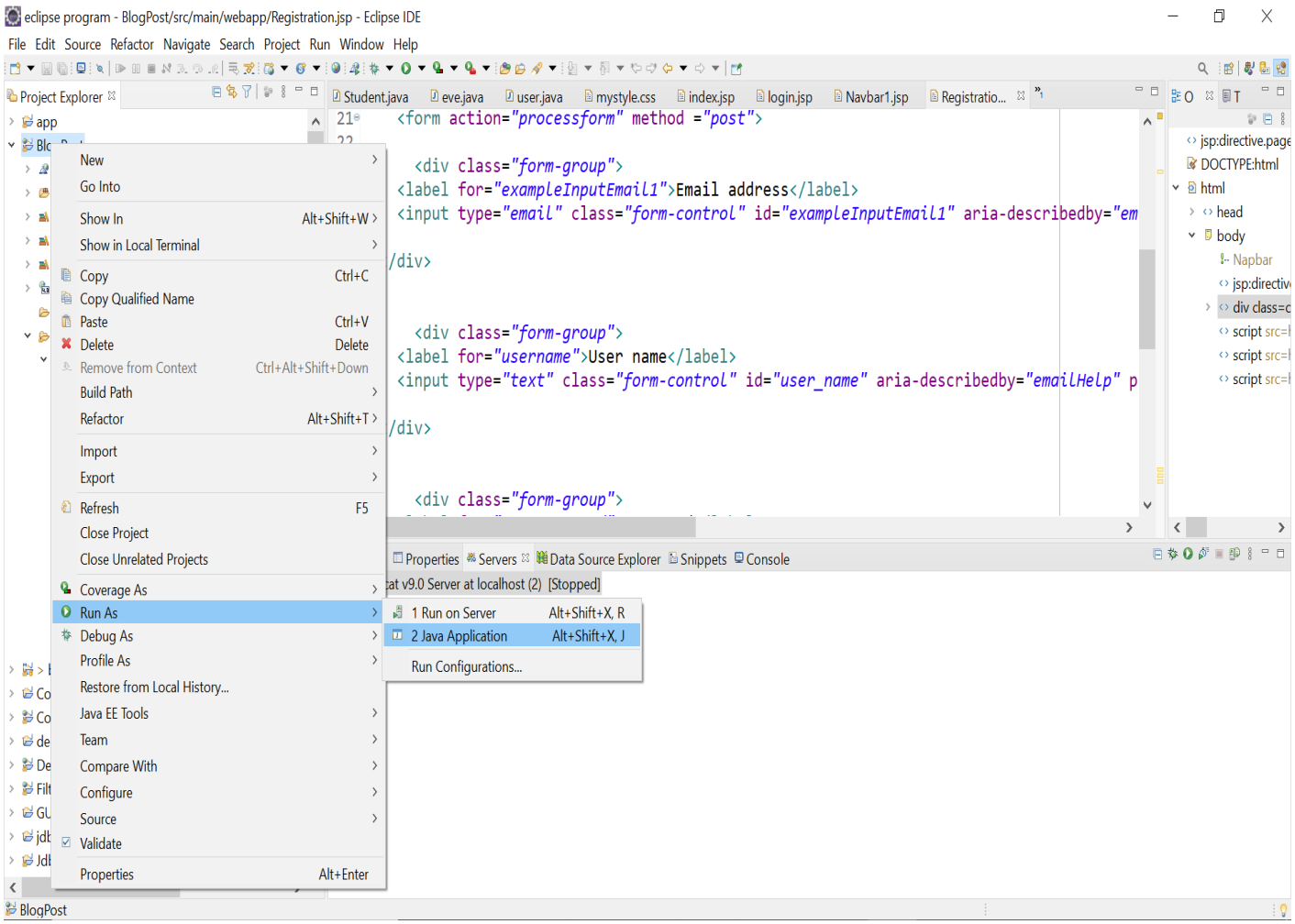
<label for="exampleInputEmail1">Email address</label>
<input type="email" class="form-control" id="exampleInputEmail1" aria-de
<small id="emailHelp" class="form-text text-muted">We'll never share your

<label for="exampleInputPassword1">Password</label>
<input type="password" class="form-control" id="exampleInputPassword1" plac

<input type="checkbox" class="form-check-input" id="exampleCheck1">
<label class="form-check-label" for="exampleCheck1">Check me out</label>
jsp:directive.page
DOCTYPE:html
html
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost (2) [Stopped]
Writable
Smart Insert
1 : 1 : 0


eclipse program - BlogPost/src/main/webapp/Navbar1.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
app
BlogPost
JAX-WS Web Services
src/main/java
JRE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0 (2)]
Referenced Libraries
Deployment Descriptor: BlogPost
build
src
main
java
webapp
css
mystyle.css
images
js
META-INF
WEB-INF
index.jsp
login.jsp
Navbar1.jsp
Registration.jsp
bookmyshow [onlyBookmyshow master]
CodingQ
CookieEx
demo1
DemoApp
Filter
GUI
jdbcex
JdbcTest
JstlExample
Student.java
eve.java
user.java
ConnectionProvider.java
mystyle.css
index.jsp
login.jsp
Navbar1.jsp
9 <li class="nav-item active">
10 <a class="nav-link" href="#">Vikrant-tech (current)<
11 </li>
12 <li class="nav-item">
13 <a class="nav-link" href="#">Contact</a>
14 </li>
15 <li class="nav-item dropdown">
16 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" da
17 Categories
18 </a>
19 
20 <a class="dropdown-item" href="#">Programming language</a>
21 <a class="dropdown-item" href="#">coding</a>
22 
23 <a class="dropdown-item" href="#">Data structure</a>
24 
25 </li>
26 <li class="nav-item">
27 <a class="nav-link " href="#">More..</a>
28 </li>
29
30 <li class="nav-item">
31 <a class="nav-link " href="Registration.jsp">SignUp</a>
32 </li>
33 </ul>
34 <form class="form-inline my-2 my-lg-0">
35 <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="
36 <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
37 </form>
nav class=navbar
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost (2) [Stopped]
Writable
Smart Insert
1 : 1 : 0

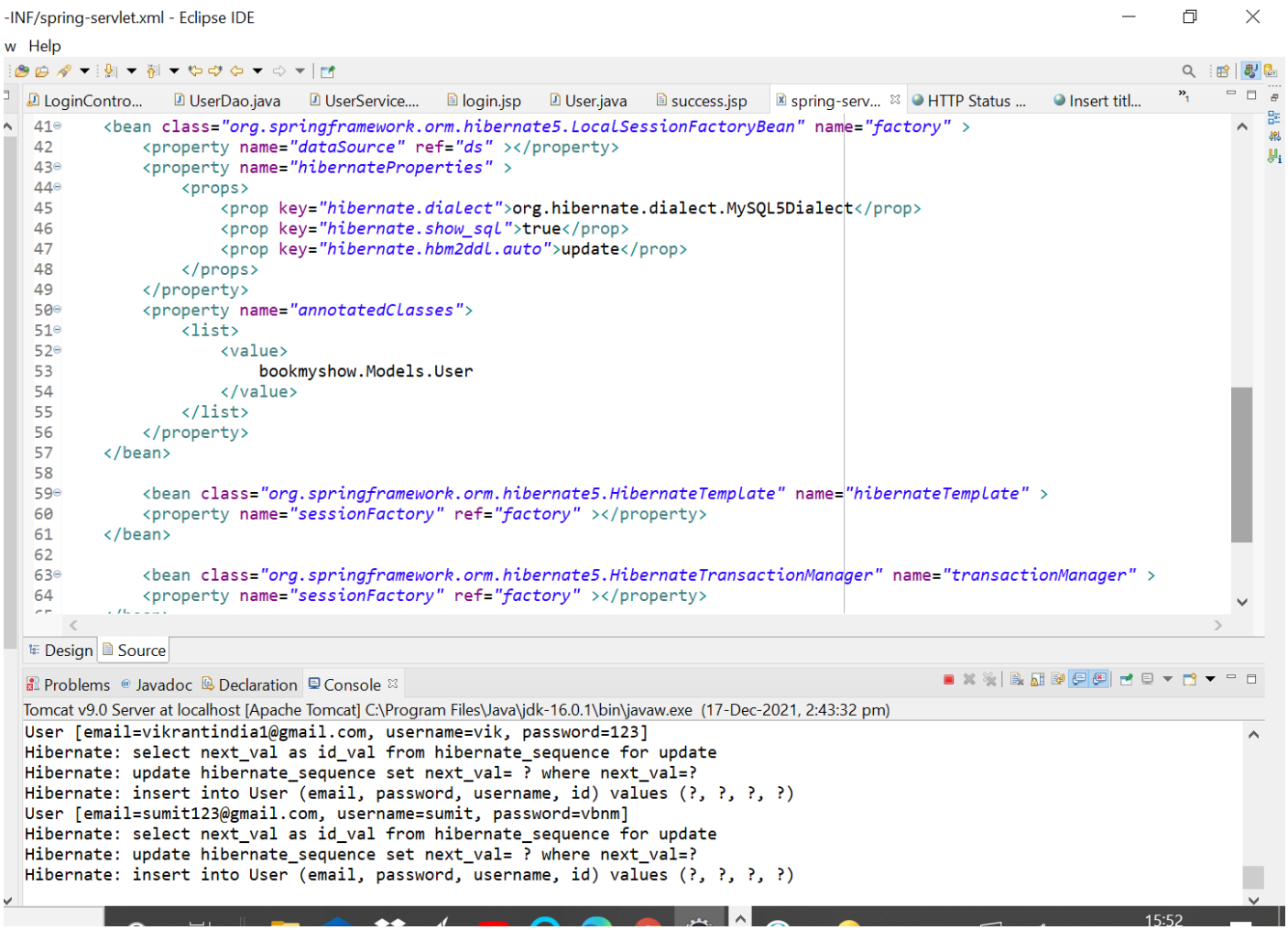

eclipse program - BlogPost/src/main/webapp/Registration.jsp - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Project Explorer
app
BlogPost
JAX-WS Web Services
src/main/java
JRE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0 (2)]
Referenced Libraries
Deployment Descriptor: BlogPost
build
src
main
java
webapp
css
mystyle.css
images
js
META-INF
WEB-INF
index.jsp
login.jsp
Navbar1.jsp
Registration.jsp
bookmyshow [onlyBookmyshow master]
CodingQ
CookieEx
demo1
DemoApp
Filter
GUI
jdbcex
JdbcTest
JstlExample
Student.java
eve.java
user.java
mystyle.css
index.jsp
login.jsp
Navbar1.jsp
Registratio...
21 <form action="processform" method ="post">
22
23 
24 <label for="exampleInputEmail1">Email address</label>
25 <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="em
26
27 
28
29
30 
31 <label for="username">User name</label>
32 <input type="text" class="form-control" id="user_name" aria-describedby="emailHelp" p
33
34 
35
36
37 
38 <label for="userpassword">password</label>
39 <input type="password" class="form-control" id="password" aria-describedby="emailHelp
40
41 
42
43 <button type="submit" class="btn btn-success" >Sign up</button>
44
45
46
47 </form>
48
49
jsp:directive.page
DOCTYPE:html
html
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost (2) [Stopped]

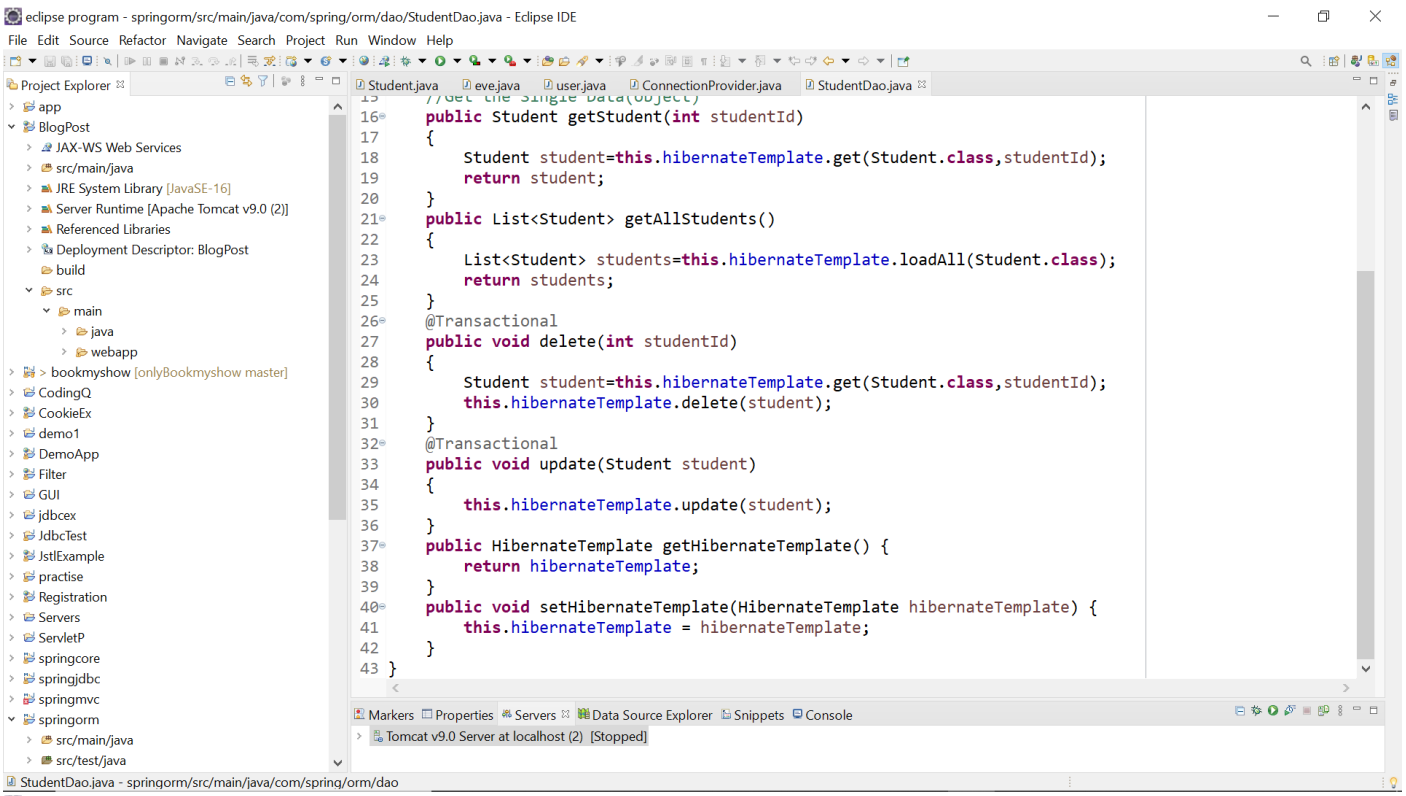

eclipse program - BlogPost/src/main/webapp/Registration.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
app
Student.java
eve.java
user.java
mystyle.css
index.jsp
login.jsp
Navbar1.jsp
Registratio...
<form action="processform" method ="post">

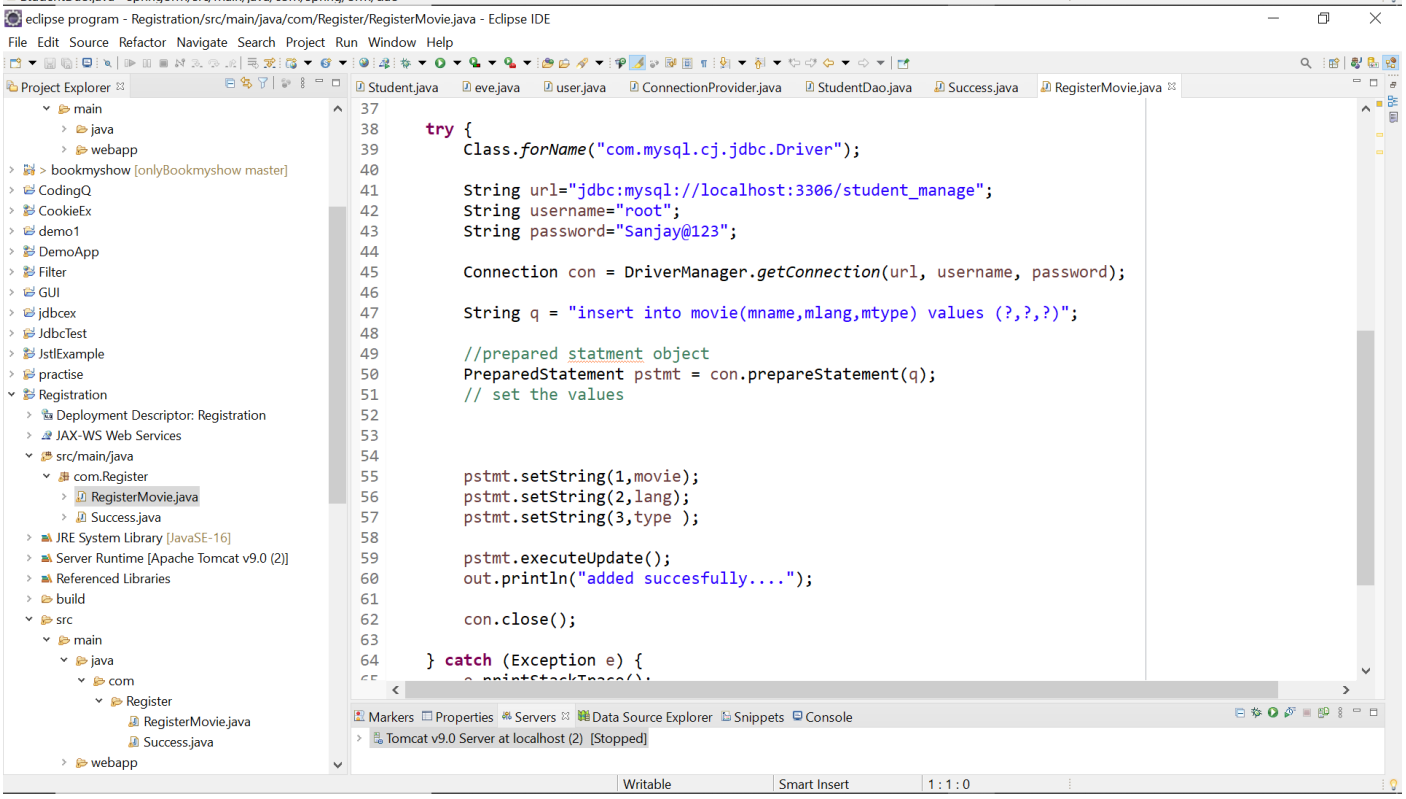
<label for="exampleInputEmail1">Email address</label>
<input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="em
/div>

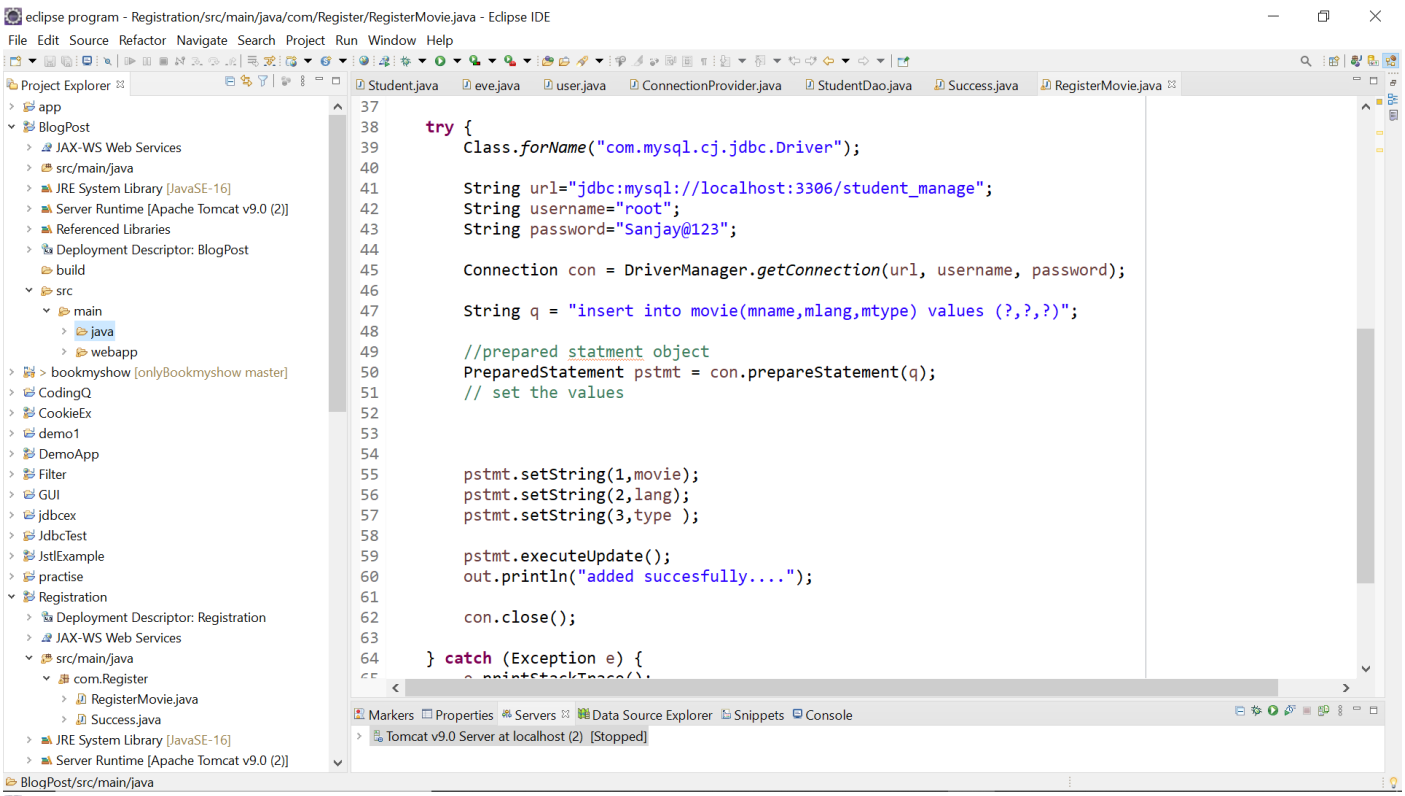
<label for="username">User name</label>
<input type="text" class="form-control" id="user_name" aria-describedby="emailHelp" p
/div>

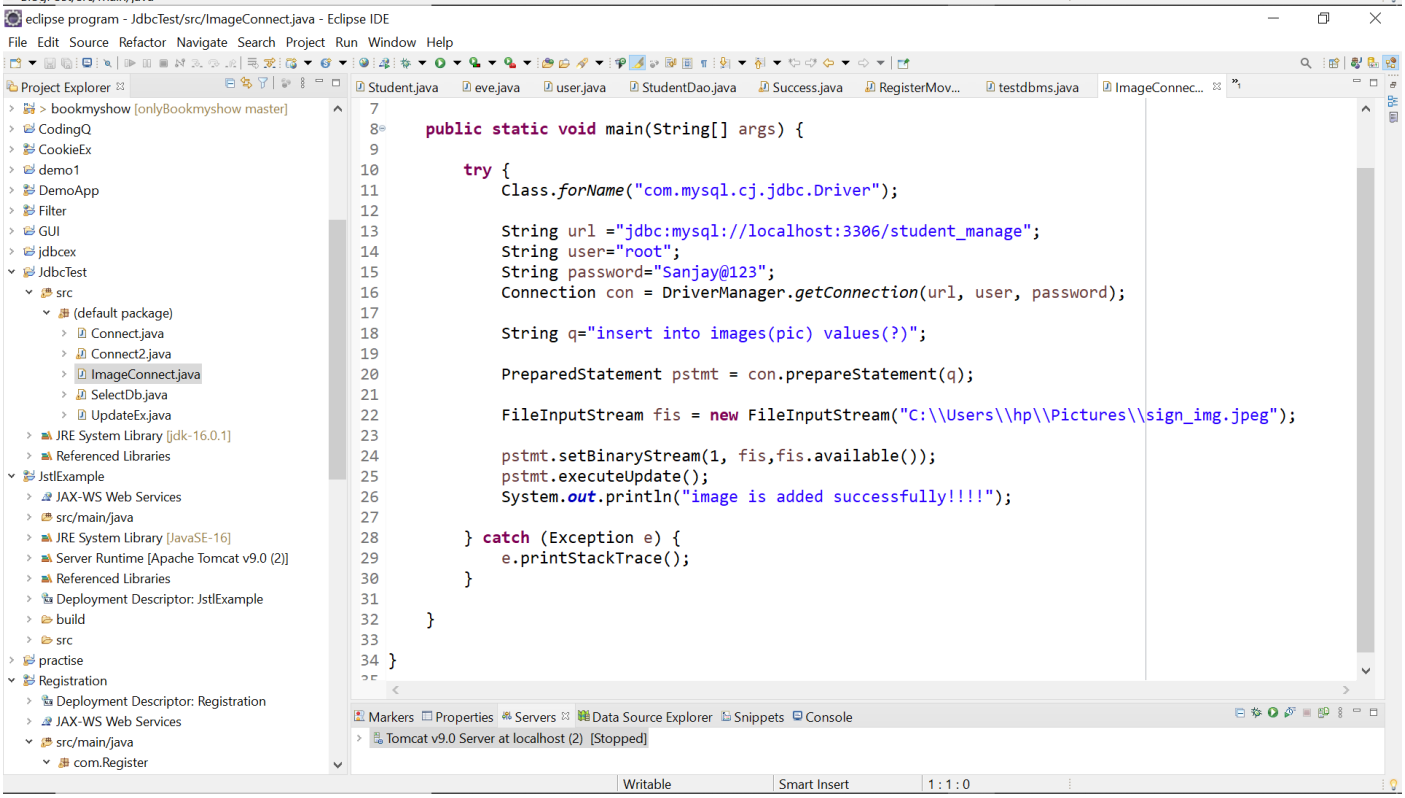
jsp:directive.page
DOCTYPE:html
html
head
body
Napbar
div class=c
script
script
script
New
Go Into
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Refactor Alt+Shift+T
Import
Export
Refresh F5
Close Project
Close Unrelated Projects
Coverage As
Run As
Debug As
Profile As
Restore from Local History...
Java EE Tools
Team
Compare With
Configure
Source
Validate
Properties Alt+Enter
1 Run on Server Alt+Shift+X, R
2 Java Application Alt+Shift+X, J
Run Configurations...
Properties
Servers
Data Source Explorer
Snippets
Console
at v9.0 Server at localhost (2) [Stopped]
BlogPost

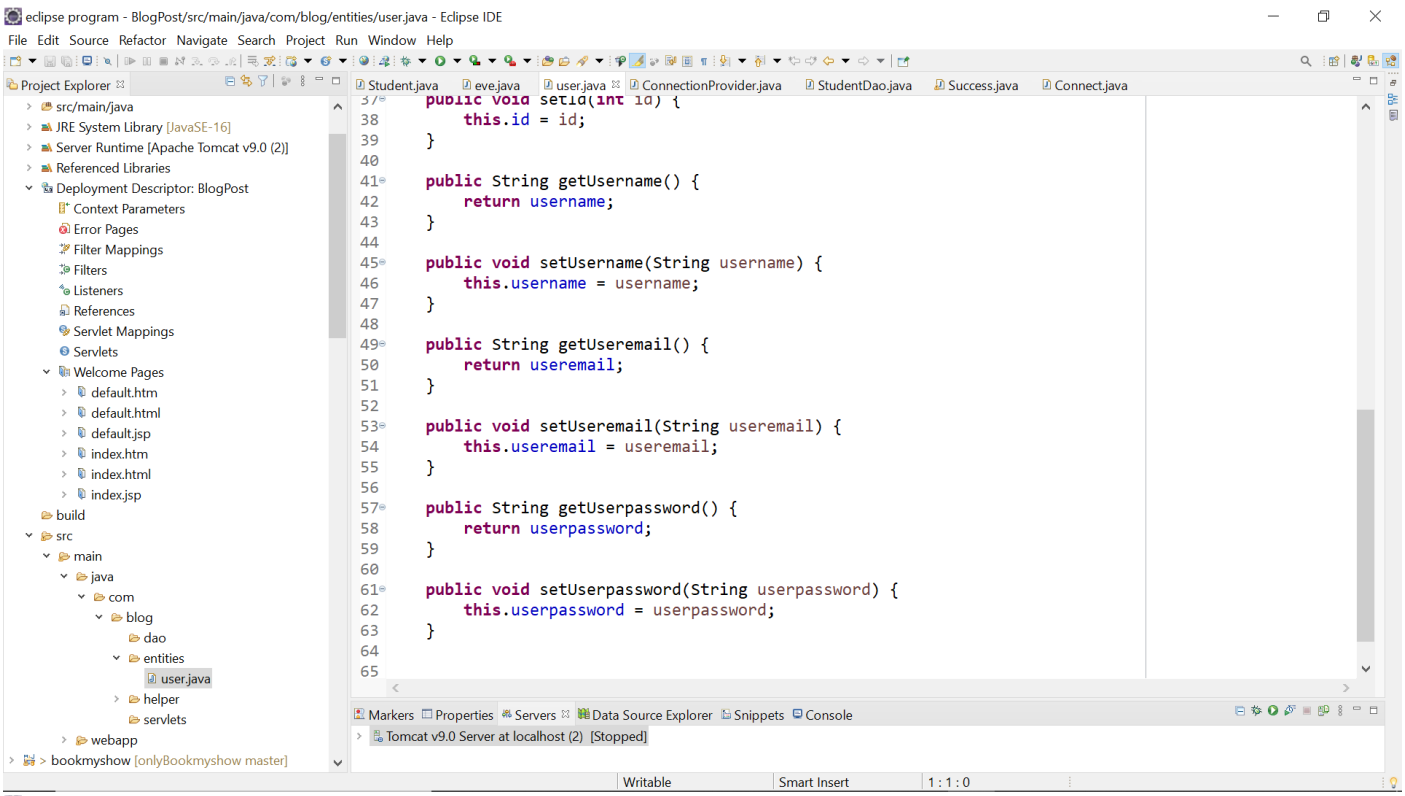

-INF/spring-servlet.xml - Eclipse IDE
w Help
LoginContro...
UserDao.java
UserService....
login.jsp
User.java
success.jsp
spring-serv...
HTTP Status ...
Insert titl...
<bean class="org.springframework.orm.hibernate5.LocalSessionFactoryBean" name="factory" >
<property name="dataSource" ref="ds" ></property>
<property name="hibernateProperties" >
<props>
<prop key="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</prop>
<prop key="hibernate.show_sql">true</prop>
<prop key="hibernate.hbm2ddl.auto">update</prop>
</props>
</property>
<property name="annotatedClasses">
<list>
<value>
bookmyshow.Models.User
</value>
</list>
</property>
</bean>
<bean class="org.springframework.orm.hibernate5.HibernateTemplate" name="hibernateTemplate" >
<property name="sessionFactory" ref="factory" ></property>
</bean>
<bean class="org.springframework.orm.hibernate5.HibernateTransactionManager" name="transactionManager" >
<property name="sessionFactory" ref="factory" ></property>
Design
Source
Problems
Javadoc
Declaration
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (17-Dec-2021, 2:43:32 pm)
User [email=vikrantindia1@gmail.com, username=vik, password=123]
Hibernate: select next_val as id_val from hibernate_sequence for update
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: insert into User (email, password, username, id) values (?, ?, ?, ?)
User [email=sumit123@gmail.com, username=sumit, password=vbnm]
Hibernate: select next_val as id_val from hibernate_sequence for update
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: insert into User (email, password, username, id) values (?, ?, ?, ?)

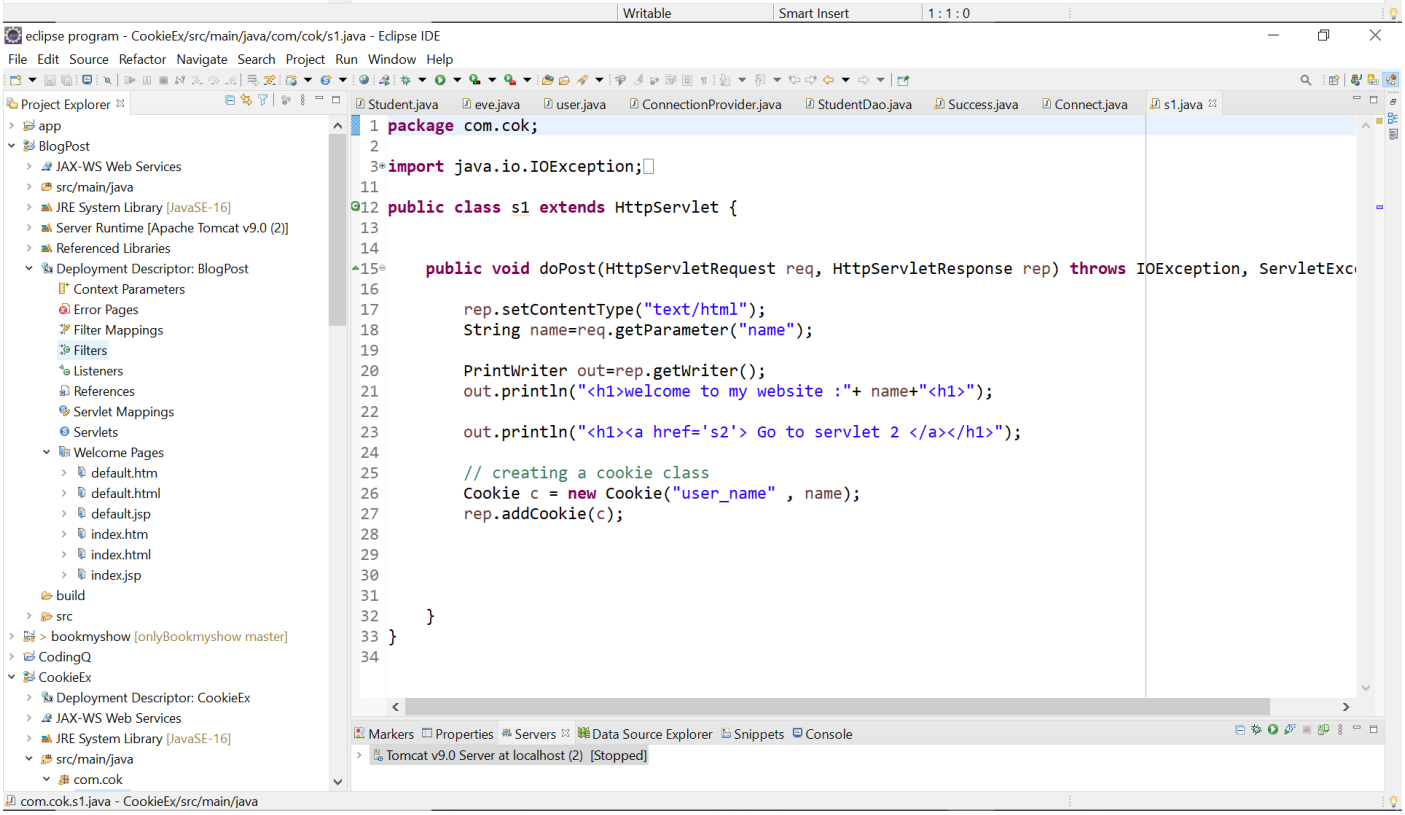

eclipse program - springorm/src/main/java/com/spring/orm/dao/StudentDao.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Student.java
eve.java
user.java
ConnectionProvider.java
StudentDao.java
app
BlogPost
JAX-WS Web Services
src/main/java
JRE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0 (2)]
Referenced Libraries
Deployment Descriptor: BlogPost
build
src
main
java
webapp
bookmyshow [onlyBookmyshow master]
CodingQ
CookieEx
demo1
DemoApp
Filter
GUI
jdbcex
JdbcTest
JstlExample
practise
Registration
Servers
ServletP
springcore
springjdbc
springmvc
springorm
src/main/java
src/test/java
16 public Student getStudent(int studentId)
17 {
18 Student student=this.hibernateTemplate.get(Student.class,studentId);
19 return student;
20 }
21 public List<Student> getAllStudents()
22 {
23 List<Student> students=this.hibernateTemplate.loadAll(Student.class);
24 return students;
25 }
26 @Transactional
27 public void delete(int studentId)
28 {
29 Student student=this.hibernateTemplate.get(Student.class,studentId);
30 this.hibernateTemplate.delete(student);
31 }
32 @Transactional
33 public void update(Student student)
34 {
35 this.hibernateTemplate.update(student);
36 }
37 public HibernateTemplate getHibernateTemplate() {
38 return hibernateTemplate;
39 }
40 public void setHibernateTemplate(HibernateTemplate hibernateTemplate) {
41 this.hibernateTemplate = hibernateTemplate;
42 }
43 }
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost (2) [Stopped]
StudentDao.java - springorm/src/main/java/com/spring/orm/dao

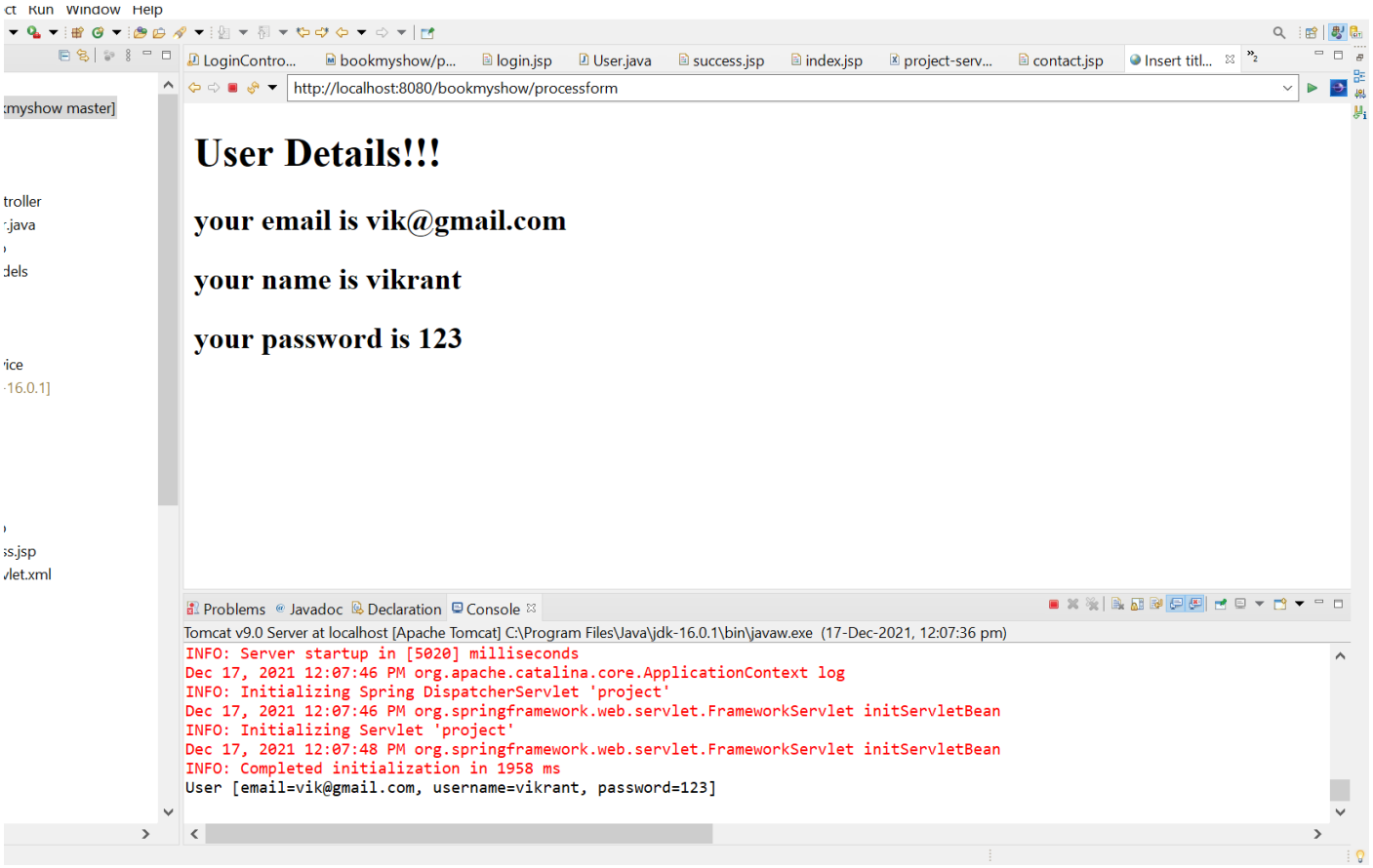

eclipse program - Registration/src/main/java/com/Register/RegisterMovie.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Student.java
eve.java
user.java
ConnectionProvider.java
StudentDao.java
Success.java
RegisterMovie.java
main
java
webapp
bookmyshow [onlyBookmyshow master]
CodingQ
CookieEx
demo1
DemoApp
Filter
GUI
jdbcex
JdbcTest
JstlExample
practise
Registration
Deployment Descriptor: Registration
JAX-WS Web Services
src/main/java
com.Register
RegisterMovie.java
Success.java
JRE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0 (2)]
Referenced Libraries
build
src
main
java
com
Register
RegisterMovie.java
Success.java
webapp
37
38 try {
39 Class.forName("com.mysql.cj.jdbc.Driver");
40
41 String url="jdbc:mysql://localhost:3306/student_manage";
42 String username="root";
43 String password="Sanjay@123";
44
45 Connection con = DriverManager.getConnection(url, username, password);
46
47 String q = "insert into movie(mname,mlang,mtype) values (?,?,?)";
48
49 //prepared statment object
50 PreparedStatement pstmt = con.prepareStatement(q);
51 // set the values
52
53
54
55 pstmt.setString(1,movie);
56 pstmt.setString(2,lang);
57 pstmt.setString(3,type );
58
59 pstmt.executeUpdate();
60 out.println("added succesfully....");
61
62 con.close();
63
64 } catch (Exception e) {
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost (2) [Stopped]
Writable
Smart Insert
1 : 1 : 0

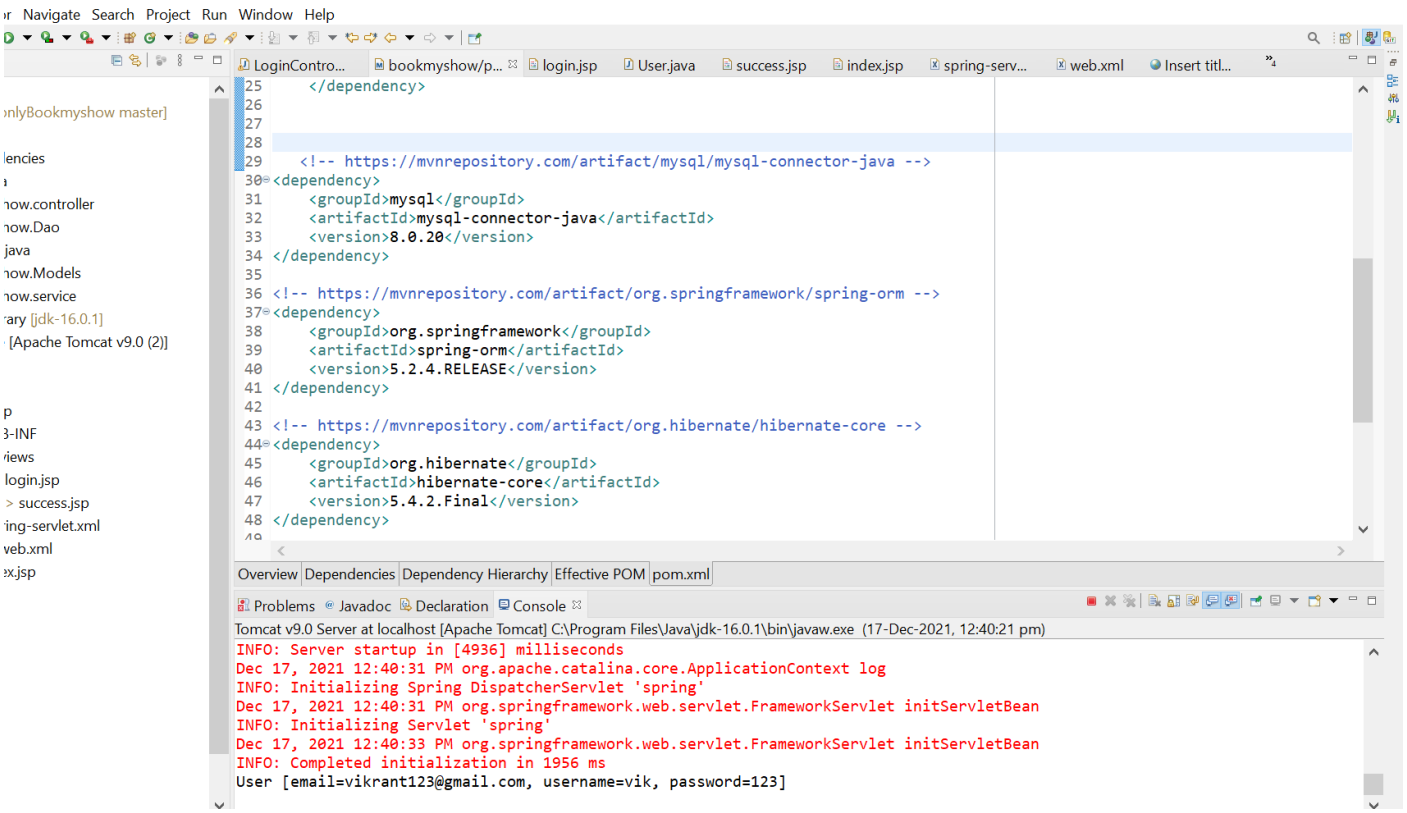

eclipse program - Registration/src/main/java/com/Register/RegisterMovie.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
app
BlogPost
JAX-WS Web Services
src/main/java
JRE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0 (2)]
Referenced Libraries
Deployment Descriptor: BlogPost
build
src
main
java
webapp
bookmyshow [onlyBookmyshow master]
CodingQ
CookieEx
demo1
DemoApp
Filter
GUI
jdbcex
JdbcTest
JstlExample
practise
Registration
Deployment Descriptor: Registration
JAX-WS Web Services
src/main/java
com.Register
RegisterMovie.java
Success.java
JRE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0 (2)]
Student.java
eve.java
user.java
ConnectionProvider.java
StudentDao.java
Success.java
RegisterMovie.java
37
38 try {
39 Class.forName("com.mysql.cj.jdbc.Driver");
40
41 String url="jdbc:mysql://localhost:3306/student_manage";
42 String username="root";
43 String password="Sanjay@123";
44
45 Connection con = DriverManager.getConnection(url, username, password);
46
47 String q = "insert into movie(mname,mlang,mtype) values (?,?,?)";
48
49 //prepared statment object
50 PreparedStatement pstmt = con.prepareStatement(q);
51 // set the values
52
53
54
55 pstmt.setString(1,movie);
56 pstmt.setString(2,lang);
57 pstmt.setString(3,type );
58
59 pstmt.executeUpdate();
60 out.println("added succesfully....");
61
62 con.close();
63
64 } catch (Exception e) {
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost (2) [Stopped]
BlogPost/src/main/java

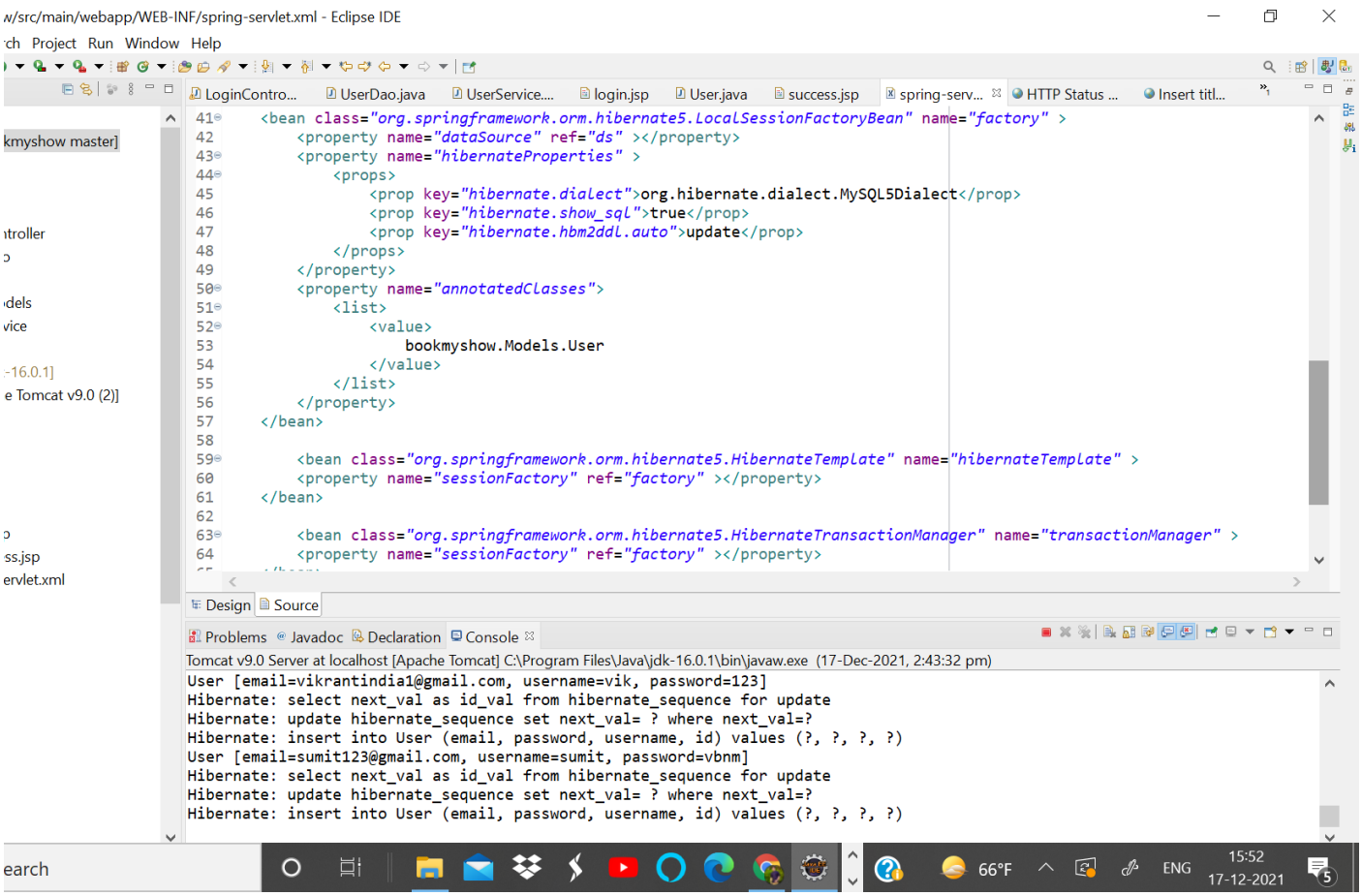

eclipse program - JdbcTest/src/ImageConnect.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
bookmyshow [onlyBookmyshow master]
CodingQ
CookieEx
demo1
DemoApp
Filter
GUI
jdbcex
JdbcTest
src
(default package)
Connect.java
Connect2.java
ImageConnect.java
SelectDb.java
UpdateEx.java
JRE System Library [jdk-16.0.1]
Referenced Libraries
JstlExample
JAX-WS Web Services
src/main/java
JRE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0 (2)]
Referenced Libraries
Deployment Descriptor: JstlExample
build
src
practise
Registration
Deployment Descriptor: Registration
JAX-WS Web Services
src/main/java
com.Register
Student.java
eve.java
user.java
StudentDao.java
Success.java
RegisterMov...
testdbms.java
ImageConnec...
7
8 public static void main(String[] args) {
9
10 try {
11 Class.forName("com.mysql.cj.jdbc.Driver");
12
13 String url ="jdbc:mysql://localhost:3306/student_manage";
14 String user="root";
15 String password="Sanjay@123";
16 Connection con = DriverManager.getConnection(url, user, password);
17
18 String q="insert into images(pic) values(?)";
19
20 PreparedStatement pstmt = con.prepareStatement(q);
21
22 FileInputStream fis = new FileInputStream("C:\\Users\\hp\\Pictures\\sign_img.jpeg");
23
24 pstmt.setBinaryStream(1, fis,fis.available());
25 pstmt.executeUpdate();
26 System.out.println("image is added successfully!!!!");
27
28 } catch (Exception e) {
29 e.printStackTrace();
30 }
31
32 }
33
34 }
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost (2) [Stopped]
Writable
Smart Insert
1 : 1 : 0


eclipse program - BlogPost/src/main/java/com/blog/entities/user.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
src/main/java
JRE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0 (2)]
Referenced Libraries
Deployment Descriptor: BlogPost
Context Parameters
Error Pages
Filter Mappings
Filters
Listeners
References
Servlet Mappings
Servlets
Welcome Pages
default.htm
default.html
default.jsp
index.htm
index.html
index.jsp
build
src
main
java
com
blog
dao
entities
user.java
helper
servlets
webapp
bookmyshow [onlyBookmyshow master]
Student.java
eve.java
user.java
ConnectionProvider.java
StudentDao.java
Success.java
Connect.java
public void setId(int id) {
this.id = id;
}
public String getUsername() {
return username;
}
public void setUsername(String username) {
this.username = username;
}
public String getUseremail() {
return useremail;
}
public void setUseremail(String useremail) {
this.useremail = useremail;
}
public String getUserpassword() {
return userpassword;
}
public void setUserpassword(String userpassword) {
this.userpassword = userpassword;
}
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost (2) [Stopped]
Writable
Smart Insert
1 : 1 : 0

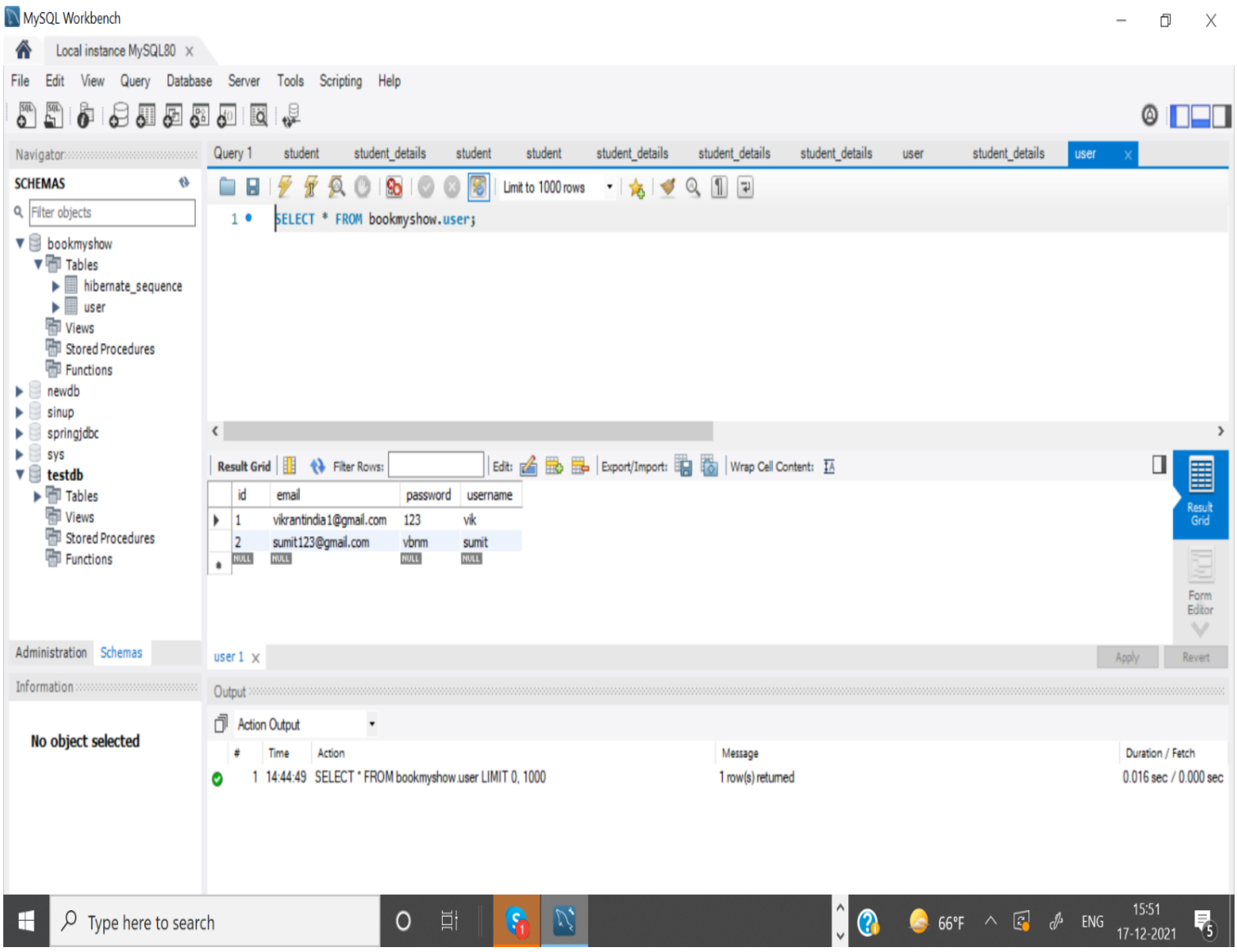

eclipse program - CookieEx/src/main/java/com/cok/s1.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
app
BlogPost
JAX-WS Web Services
src/main/java
JRE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0 (2)]
Referenced Libraries
Deployment Descriptor: BlogPost
Context Parameters
Error Pages
Filter Mappings
Filters
Listeners
References
Servlet Mappings
Servlets
Welcome Pages
default.htm
default.html
default.jsp
index.htm
index.html
index.jsp
build
src
bookmyshow [onlyBookmyshow master]
CodingQ
CookieEx
Deployment Descriptor: CookieEx
JAX-WS Web Services
JRE System Library [JavaSE-16]
src/main/java
com.cok
Student.java
eve.java
user.java
ConnectionProvider.java
StudentDao.java
Success.java
Connect.java
s1.java
package com.cok;
import java.io.IOException;
public class s1 extends HttpServlet {
public void doPost(HttpServletRequest req, HttpServletResponse rep) throws IOException, ServletExc
rep.setContentType("text/html");
String name=req.getParameter("name");
PrintWriter out=rep.getWriter();
out.println("<h1>welcome to my website :"+ name+"<h1>");
out.println("<h1><a href='s2'> Go to servlet 2 </a></h1>");
// creating a cookie class
Cookie c = new Cookie("user_name" , name);
rep.addCookie(c);
}
}
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost (2) [Stopped]
com.cok.s1.java - CookieEx/src/main/java


LoginContro...
bookmyshow/p...
login.jsp
User.java
success.jsp
index.jsp
project-serv...
contact.jsp
Insert titl...
http://localhost:8080/bookmyshow/processform
User Details!!!
your email is vik@gmail.com
your name is vikrant
your password is 123
Problems
Javadoc
Declaration
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (17-Dec-2021, 12:07:36 pm)
INFO: Server startup in [5020] milliseconds
Dec 17, 2021 12:07:46 PM org.apache.catalina.core.ApplicationContext log
INFO: Initializing Spring DispatcherServlet 'project'
Dec 17, 2021 12:07:46 PM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: Initializing Servlet 'project'
Dec 17, 2021 12:07:48 PM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: Completed initialization in 1958 ms
User [email=vik@gmail.com, username=vikrant, password=123]

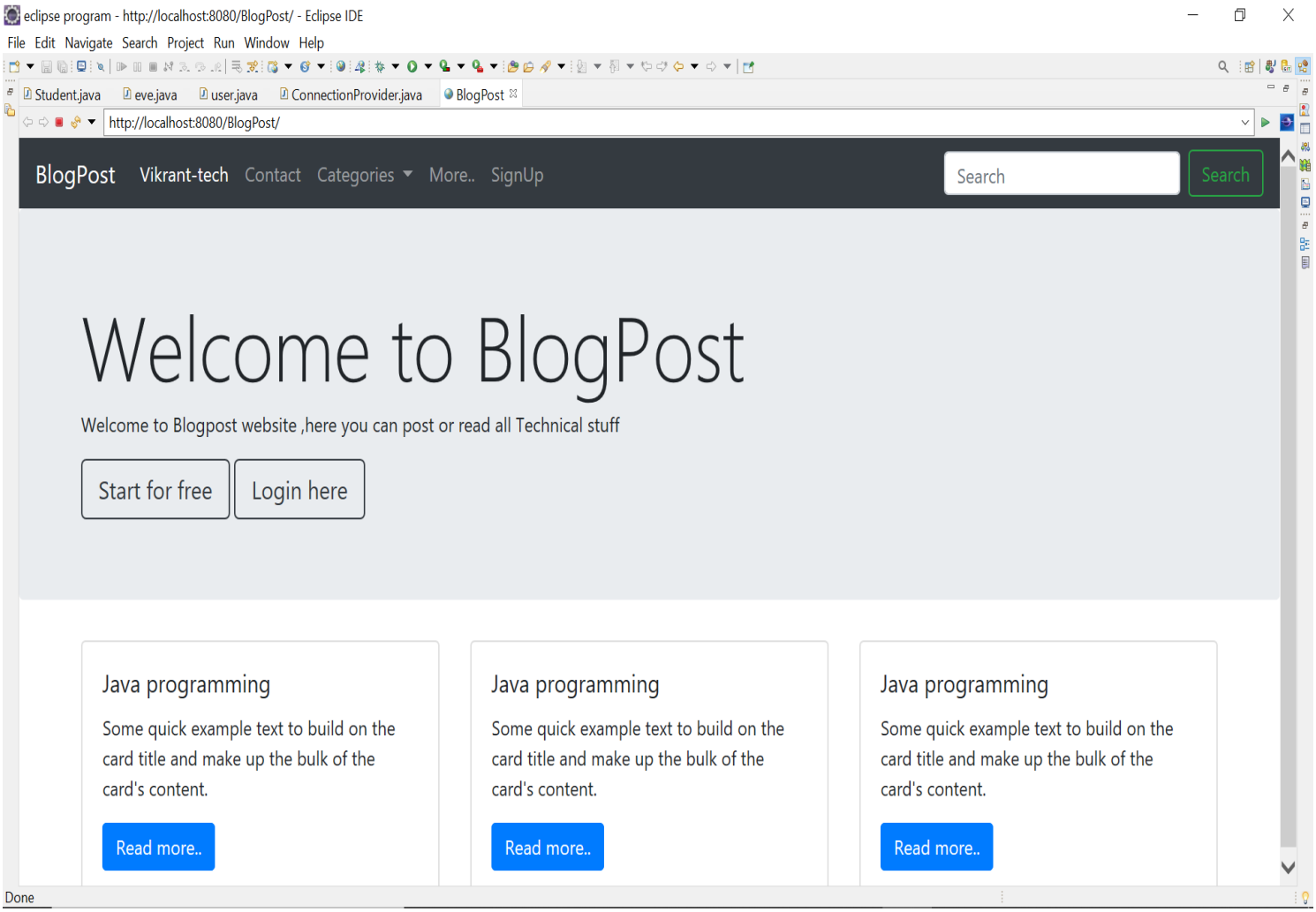

Navigate
Search
Project
Run
Window
Help
LoginContro...
bookmyshow/p...
login.jsp
User.java
success.jsp
index.jsp
spring-serv...
web.xml
Insert titl...
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.20</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-orm</artifactId>
<version>5.2.4.RELEASE</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
<dependency>
<groupId>org.hibernate</groupId>
<artifactId>hibernate-core</artifactId>
<version>5.4.2.Final</version>
</dependency>
Overview
Dependencies
Dependency Hierarchy
Effective POM
pom.xml
Problems
Javadoc
Declaration
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (17-Dec-2021, 12:40:21 pm)
INFO: Server startup in [4936] milliseconds
Dec 17, 2021 12:40:31 PM org.apache.catalina.core.ApplicationContext log
INFO: Initializing Spring DispatcherServlet 'spring'
Dec 17, 2021 12:40:31 PM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: Initializing Servlet 'spring'
Dec 17, 2021 12:40:33 PM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: Completed initialization in 1956 ms
User [email=vikrant123@gmail.com, username=vik, password=123]


w/src/main/webapp/WEB-INF/spring-servlet.xml - Eclipse IDE
ch Project Run Window Help
LoginContro... UserDao.java UserService.... login.jsp User.java success.jsp spring-serv... HTTP Status ... Insert titl...
41 <bean class="org.springframework.orm.hibernate5.LocalSessionFactoryBean" name="factory" >
42 <property name="dataSource" ref="ds" ></property>
43 <property name="hibernateProperties" >
44 <props>
45 <prop key="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</prop>
46 <prop key="hibernate.show_sql">true</prop>
47 <prop key="hibernate.hbm2ddl.auto">update</prop>
48 </props>
49 </property>
50 <property name="annotatedClasses">
51 <list>
52 <value>
53 bookmyshow.Models.User
54 </value>
55 </list>
56 </property>
57 </bean>
58
59 <bean class="org.springframework.orm.hibernate5.HibernateTemplate" name="hibernateTemplate" >
60 <property name="sessionFactory" ref="factory" ></property>
61 </bean>
62
63 <bean class="org.springframework.orm.hibernate5.HibernateTransactionManager" name="transactionManager" >
64 <property name="sessionFactory" ref="factory" ></property>
Design Source
Problems Javadoc Declaration Console
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (17-Dec-2021, 2:43:32 pm)
User [email=vikrantindia1@gmail.com, username=vik, password=123]
Hibernate: select next_val as id_val from hibernate_sequence for update
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: insert into User (email, password, username, id) values (?, ?, ?, ?)
User [email=sumit123@gmail.com, username=sumit, password=vbnm]
Hibernate: select next_val as id_val from hibernate_sequence for update
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: insert into User (email, password, username, id) values (?, ?, ?, ?)
66°F
ENG
15:52
17-12-2021

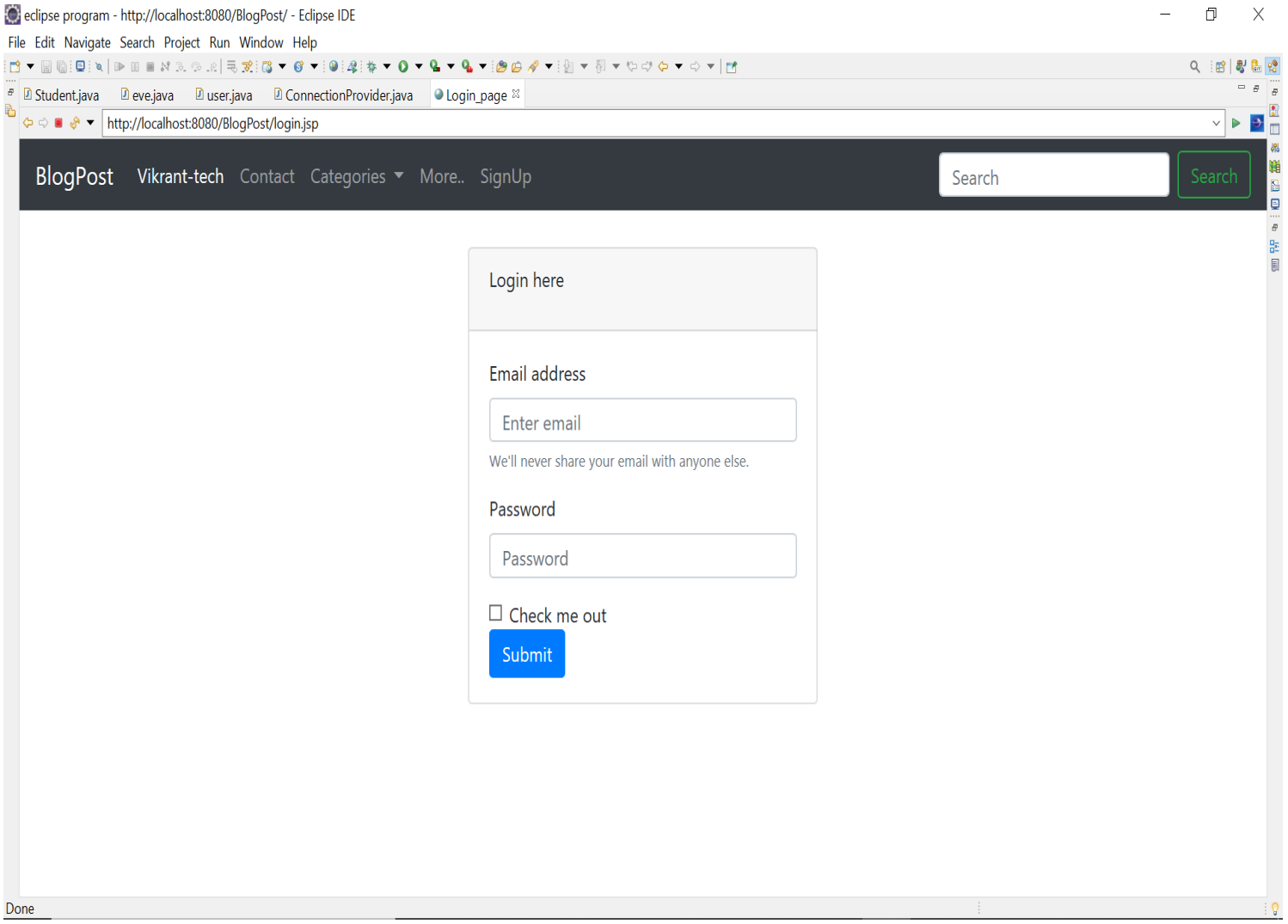
# Database:-

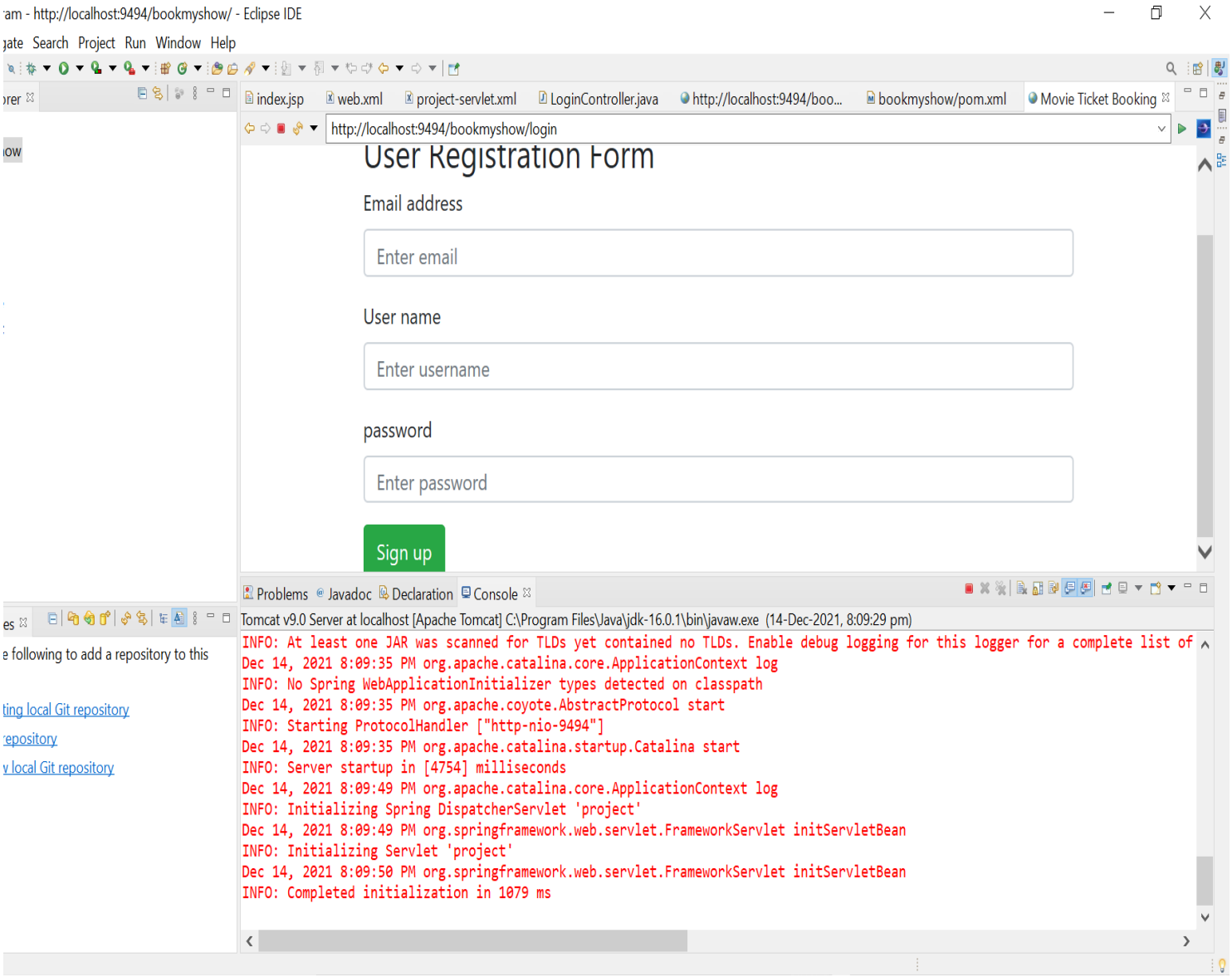

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
bookmyshow
Tables
hibernate_sequence
user
Views
Stored Procedures
Functions
newdb
sinup
springjdbc
sys
testdb
Tables
Views
Stored Procedures
Functions
Administration Schemas
Information
No object selected
Query 1 student student_details student student student_details student_details student_details user student_details user
Limit to 1000 rows
1 SELECT * FROM bookmyshow.user;
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
id email password username
1 vikrantindia1@gmail.com 123 vik
2 sumit123@gmail.com vbnm sumit
NULL NULL NULL NULL
Result Grid
Form Editor
user 1
Apply Revert
Output
Action Output
# Time Action Message Duration / Fetch
1 14:44:49 SELECT * FROM bookmyshow.user LIMIT 0, 1000 1 row(s) returned 0.016 sec / 0.000 sec
Type here to search
66°F
ENG
15:51
17-12-2021

- User Interface first appearance:



# Signup and registration page :-




am - http://localhost:9494/bookmyshow/ - Eclipse IDE
gate Search Project Run Window Help
index.jsp
web.xml
project-servlet.xml
LoginController.java
http://localhost:9494/boo...
bookmyshow/pom.xml
Movie Ticket Booking
http://localhost:9494/bookmyshow/login
User Registration Form
Email address
Enter email
User name
Enter username
password
Enter password
Sign up
Problems
Javadoc
Declaration
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (14-Dec-2021, 8:09:29 pm)
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of
Dec 14, 2021 8:09:35 PM org.apache.catalina.core.ApplicationContext log
INFO: No Spring WebApplicationInitializer types detected on classpath
Dec 14, 2021 8:09:35 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-9494"]
Dec 14, 2021 8:09:35 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [4754] milliseconds
Dec 14, 2021 8:09:49 PM org.apache.catalina.core.ApplicationContext log
INFO: Initializing Spring DispatcherServlet 'project'
Dec 14, 2021 8:09:49 PM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: Initializing Servlet 'project'
Dec 14, 2021 8:09:50 PM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: Completed initialization in 1079 ms
e following to add a repository to this
ting local Git repository
epository
y local Git repository

# Result

An interactive and engaging website is developed using the latest technologies and frameworks like HTML, CSS, JavaScript, SQL, JDBC etc.

Blogpost is an easy-to-use website that helps writers and readers improve their writing skills and helps the general public to read and learn about topics that interest them. In general, participants showed significant improvements in a few aspects of writing, such as content development., language resources, style, word, word choice and other small writing skills.

The time spent working on blogs was quite long, not only writing, posting and commenting, but also reading a lot of information contained on blogs, yet they found great satisfaction in their participation. All posts and responses were commented on and analyzed both in terms of their language perspective and the content of the information which often produces a fun conversation in the classroom.

# Conclusion

An interactive and engaging website is developed using the latest technologies and frameworks like HTML, CSS, JavaScript , SQL , JDBC etc.

Blogpost is an easy-to-use website that helps writers and readers improve their writing skills and helps the general public to read and learn about topics that interest them. In general, participants showed significant improvements in a few aspects of writing, such as content development., language resources, style, word, word choice and other small writing skills. The results also revealed that working as a team in addition to working as individuals helped improve our forum as anyone or anyone could literally publish a blog by creating an account. This ongoing discussion stimulates students' desire to continue writing and communicating in two ways: with the student and with themselves. With the like, comment and share feature it increased interaction and gave everyone a home feeling in our Blogpost.

By our website, various students or rather learners in general would get help to learn about new things as our site isn’t focusing on a particular topic, the content will be diversified.

# References


**Systems Analysis and Design**

Second Edition, 1998; By Elias M. Awad

**Software Engineering**

Fifth Edition, 2001; By Roger S. Pressman, Ph.D.

**SQL for SQL Server**

First Indian Edition 2003; By Bryan Syverson

**Visual Basic 6 Database How – To**

First Indian Edition 1999; By Eric Winemiller, Jason Roff, Bill Heyman, Ryan Groom

**SQL, PL/SQL**

Second Revised Edition; By Ivan Bayross

**An Introduction to Database Management System**

Bipin C. Desai

**HTML / CSS :** www.w3school.com

**JavaScript** : www.codementor.io