A Thesis/Project/Dissertation Report

on

AUTOMATIC TIMETABLE GENERATION

Submitted in partial fulfillment of the

requirement for the award of the degree of

BACHELOR OF ENGINEERING IN
COMPUTER SCIENCE
&ENGINEERING

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. Samson Ebenezer U.
Assistant Professor

Submitted By

Nidhi Rani-20SCSE1010096
Prachi Tanya-20SCSE1010580

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
MONTH, YEAR

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"AUTOMATIC TIMETABLE GENERATION"** in partial fulfillment of the requirements for the award of the **Bachelor of engineering in computer science** submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of August 2021Year to December 2021, under the supervision of **Mr. Samson Ebenezer U, Assistant Professor** , Department of Computer Science and Engineering/Computer Application 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 us for the award of any other degree of this or any other places.

Nidhi Rani (20SCSe1010096)
Prachi Tanya(20SCSE1010580)

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

Mr. Samson Ebenezer U.
Assistant Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Nidhi Rani (20SCSE1010096), Prachi Tanya(20SCSE1010580) has been held on 21/12/2021-24/12/2021 and our work is recommended for the award of Bachelor of Engineering in Computer Science.

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

Signature of Project Coordinator **Signature of Dean**

Date: 22 December, 2021
Place: Greater Noida

Acknowledgement

One of the pleasant aspects of preparing a project report is the opportunity to thank those who have contributed to make this project possible .

We are extremely thankful to **Mr. Samson Ebenezer U** sir whose active interest in the project &insight helped us to formulate, redefine implement our approach to the project Furthermore.

I would also like to acknowledge with much appreciation the crucial role of my teammates and all those who have guided and helped.

Abstract

The manual system of preparing time table in colleges is very time consuming and hard task which usually ends up with various classes clashing either at identical room or with same teachers having more than one class at a time. Due to manual approach, proper use of resources is neither effective nor efficient. To overcome all these problems we propose to make an automated system with computer assisted timetable generator. The system will take various inputs like number of subjects, teachers, maximal lectures a teacher can conduct, priority of subject and topics to be covered in a week or a lecture, depending upon these inputs it will generate possible time tables for working days of the week, making optimal use of all resources in a way that will best suit the constraints. An appropriate timetable is then chosen from the optimal solutions generated.

The proposed systems were developed to solve the timetable generation being faced by college every academic year

. • We can reduce high cost and slow turnaround involved in the generation of near-optimal timetables.

• The system has capabilities for input of the various courses, halls of lectures, departments, programs, lecturers and the specification of a few constraints from which the timetable is constructed.

• The proposed timetable system for this project seeks to generate maximum error free timetables using the principles of genetic algorithm (selection and crossover).

The tools used for the design and methodology include Algorithm and Flow chart. From the flow chart below, the steps carried out by the system administrator are as follow: The system administrator logs into the system. The administrator then input the courses with their codes and the unit. At this point, the admin will keep adding until the number of courses needed has been inputted. The admin can remove a course that has been inputted in the case of error.

The final system should be able to generate time tables in completely automated way which will save a lot of time and effort of a department administration. Focus on optimization of resources i.e., teachers, classrooms etc. Provide a facility for everyone to view the time table. This application is provided with necessary details of faculty and subjects which are stored in database and then by making use of available data it generates timetable with minimum time when compared to manual generation of timetable.

In future the Automatic Time Table Generator System can be used for the attendance system also, by which the teacher and student attendance problem can be solved. Different roles can be added to enhance security of time table such as Administrator, user.

Contents

List of Table

List of Figures

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering

CHAPTER-1 Introduction

1.1 Introduction

The manual timetable scheduling demands considerable time and efforts along with lots of paperwork. The timetable scheduling can also be considered as a Constraint satisfaction problem (CSP),which is a unique concept in Artificial Intelligence[4],in which we find a solution that satisfies the given set of constraints. Time table scheduling has been in human requirements since they thought of managing time effectively. It is widely used in schools, colleges and other fields of teaching. In early days, time table scheduling was done manually with a single person or some group involved in task of scheduling it with their hands, which take lot of effort and time. While scheduling, even the smallest constraints can take a lot of time and the case is even worse when the number of constraints or the amount of data to deal with increases. Other cases that can cause problem is when the number of Faculties (Teaching Staff) are less, resulting in rescheduling of time table or they need to fill on empty seats urgently. In such cases Automated time table, scheduling can be a very convenient method for managing it in computers with algorithms also proving to be eco-friendly for no paperwork.

The main objective of this research is to schedule courses to fully utilize available resources by assigning the course to lecturer at correct timeslot and place to appropriate event. The timetabling constraints are many and varied. In this research, genetic algorithm approach has been applied for solving lecturers timetabling problem. It represents a central concept and the relationship between them and other concepts of timetabling problem. An appropriate timetable is then chosen from the optimal solutions generated. Timetable creation is a very arduous and time consuming task. To create timetable it takes lots of patience and man hours. Time table is created for various purposes like to organize lectures in school and colleges, to create timing charts for train and bus schedule and many more.

1.2 Formulation of Problem

The time tabling problem can be modeled as a constraint satisfaction problem with many parameters and loose constraints. These constraints have to be modeled in a format that can be handled efficiently by the scheduling algorithm. The scheduling involves allowing for a number of pair wise restrictions on which tasks can be done simultaneously. For instance, in attempting to schedule classes at an institute, two courses taught by the same faculty member cannot be scheduled for the same time slot. Similarly, two courses that are required by the same group of students also should not conflict.

1.2.1 Tool and Technology Used

The tools used for the design and methodology include Algorithm and Flow chart. From the flow chart below, the steps carried out by the system administrator are as follow: The system administrator logs into the system. The administrator then input the courses with their codes and the unit. At this point, the admin will keep adding until the number of courses needed has been inputted. The admin can remove a course that has been inputted in the case of error. After inputting the courses, it moves to the next page where all the lecture halls or rooms that will be used will be inputted. After inputting these, the system then generates the timetable system.

CHAPTER-2 Literature Survey

In some years two main approaches have been successful for implementing the timetabling problem. The first approach is based on local search procedures method such as simulated annealing, tabu search and genetic algorithms.

These methods express constraints as some value of functions, which are minimized by a heuristic search of better solutions in reference of some initial feasible solution. The second approach is based on constraint programming (CP).

Its main advantage is declaratively a direct statement of the constraints serves as part of the program. This makes the program easy to modify, which is critical in timetabling problems. The constraints are handled through a system of constraint propagation, which decrease domains of variables, coupled with backtracking search.

The main disadvantages of these approaches are

1. Difficulties with expressing hard and soft constraints,
2. The need to determine their parameters through experimentation and
3. Possible problems with improving the initial feasible solution, Which – as a rule – may be determined without problem

An attempt to overcome the drawbacks along soft constraints was discussed, successfully combined local search with constraint satisfaction to decrease their drawbacks. A custom-tailored distribution strategy is able to introduce soft constraints during a search, leading immediately to a “good” timetable; giving the ability to effectively optimize the timetable.

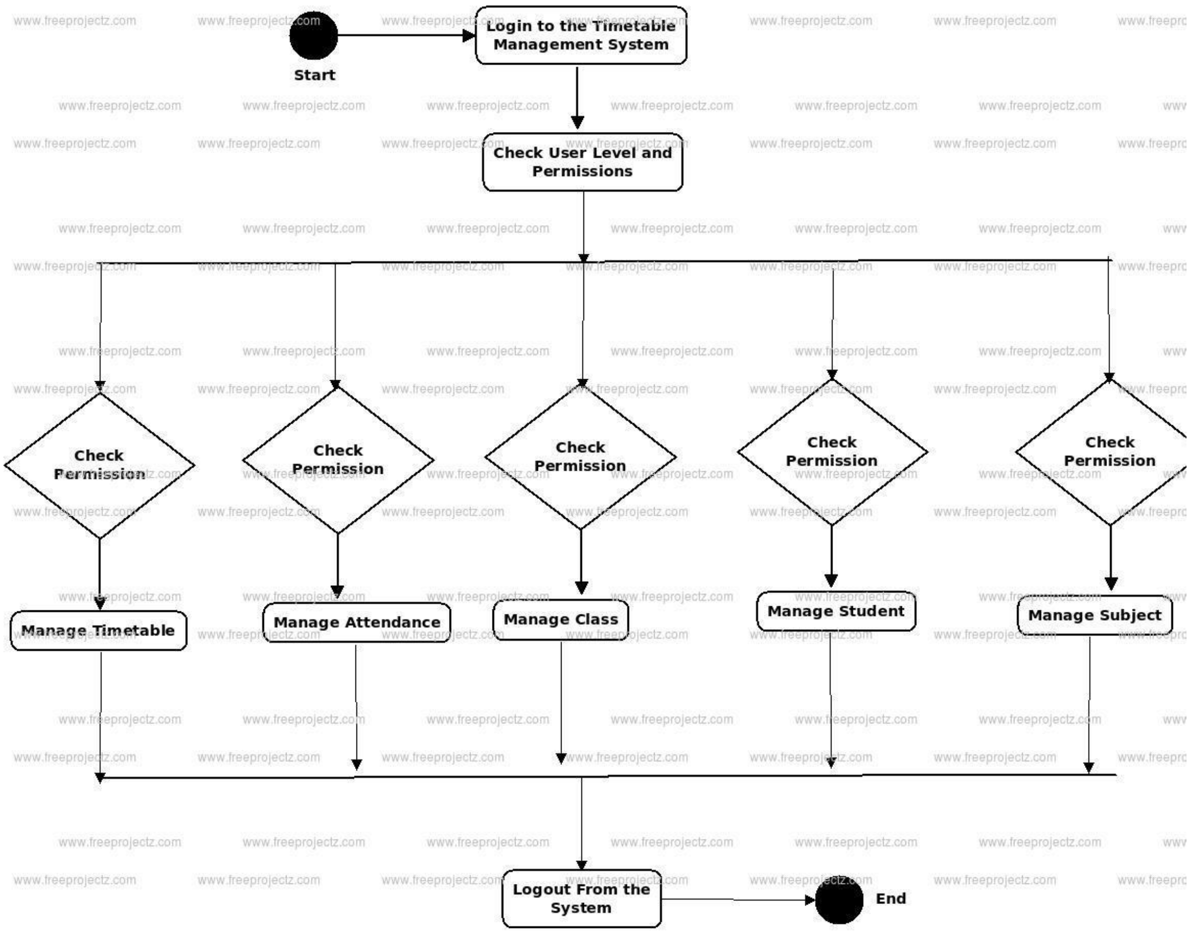
A. Bhaduri A, evolutionary techniques have been used to solve the time table scheduling problem. Methodologies like Genetic Algorithms, Evolutionary Algorithms etc. have been used with mixed success. In this paper, we have reviewed the problem of educational time table scheduling with genetic algorithm. We have further solved the problem with a mimetic hybrid algorithm, genetic artificial immune network and compare the result with that obtained from genetic algorithm. Results show that GAIN is able to reach the ideal feasible solution faster than that of GA.

B. Dipti Shrinivasan , Find a feasible tutorial timetable in a large university department is a challenging problem faced repeatedly in educational establishments. This paper represents an evolutionary algorithm (EA) based approach to solving a strong constrained university timetabling problem. The move toward a problem-specific chromosome representation. Heuristics and also context-based reasoning have been used for received feasible timetables in a reasonable computing time. An intelligent flexible mutation scheme has been employed for fast-moving up the convergence. The comprehensive course timetabling system presented in this paper has been approved, tested and discussed using real world data from a large university.

Chapter 3
Working of Project

The most creative and challenging phase of the life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementations of the system. The importance of software design can be stated in a single word “Quality”. Design provides us with representations of software that can be assessed for quality. Design is the only way where we can accurately translate user requirements into a complete software product or system. Without design we risk building an unstable system that might fail if small changes are made. It may as well be difficult to test, or could be one who’s quality can’t be tested. So it is an essential phase in the development of a software product.

3.1 UML Diagrams



UML DIAGRAM

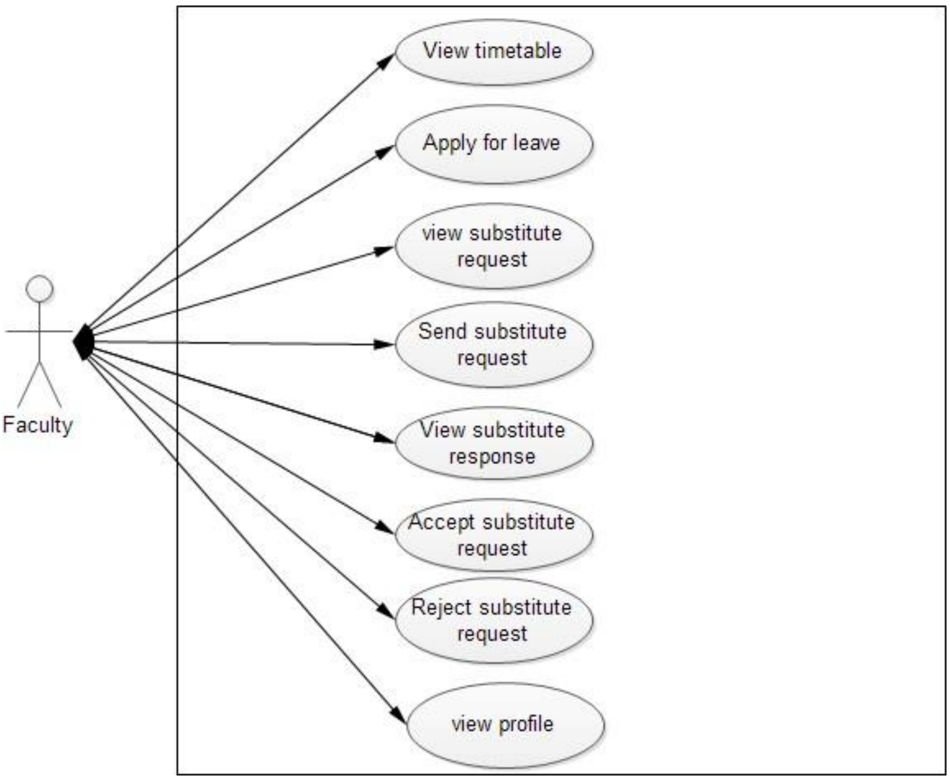
3.1.1 Use-Case diagram

This system is used by 3 types of users. They are

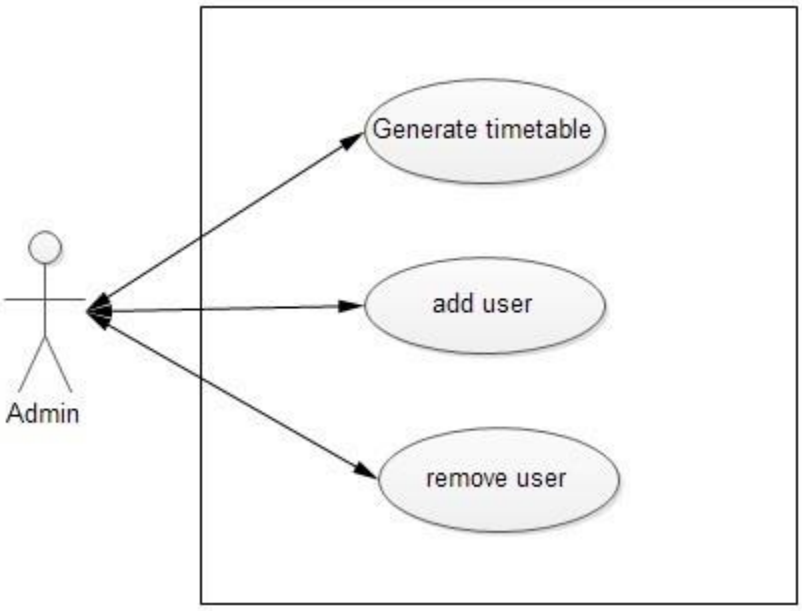
- Admin , Principal ,Users(faculty)

Principal

Faculty



Admin



3.2 Data Flow Diagrams

The DFD is also known as the bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out in these data and the output data generated by the system. Starting with a set of high-level functions that a system performance of DFD model in hierarchically it represents various sub functions. In a normal convention, logical DFD can be completed using only four notations:

- Represents source or destination of data
- Represents Data Flow
- Represents a process that transforms incoming data into outgoing data
- Represents Data Source

Function Symbol

A function is represented using a circle. This symbol is called a process or a bubble. Bubbles are annotated with the names of corresponding functions.

External Entity Symbol

An external entity such as a librarian, library member etc is represented by a rectangle. The external entities are essentially those physical entities external to the software system, which interact with the system by inputting data to the system or by consuming the data produced by the system.

Data Flow Symbol

A directed arc or an arrow is used as a Data Flow Symbol. This represents the data flow occurring between two processes or between an external entity and a process in direction of the Data Flow Arrow. Data Flow symbols are annotated with corresponding data names.

Data Store Symbol

A Data Store represents a logical file; it is represented using two parallel lines. A logical file can represent either Data Store Symbol, which can represent either data structure or a physical file on disk.

Output Symbol

The output symbol is used when a hardcopy is produced and the user of the copies cannot be clearly specified or there are several users of the output. The DFD at the simplest level is referred to as the Context Analysis Diagram. These are expanded by level, each explaining in process in detail. Processes are numbered for easy identification and are normally labeled in block letters. Each data flow is labeled for easy understanding.

Context Diagram

The context diagram is the most abstract data flow representation of a system. It represents the entire system as a single bubble. This bubble is labeled according to the main function of the system. The various external entities with the system interact and the data input to the system and the data output from the system are represented as incoming and outgoing arrows. These data flow arrows should be annotated with the corresponding data names.

Level-1 DFD

To develop the LEVEL-1 DFD, examine the high level functional requirements. It is between three to seven high-level functional requirements then these can be directly represented as bubbles as in the LEVEL-1 DFD. Wecan examine the input data to these functions and the data output by these functions, and represent them appropriately in the diagram.

Procedure to develop a DFD model of a system:

1. The SRS document is examined to determine:
   - Different high-level functions that the system need to perform.
   - Data input to every high-level function
   - Data output from every high-level functions
   - Interactions among the identified high-level functions

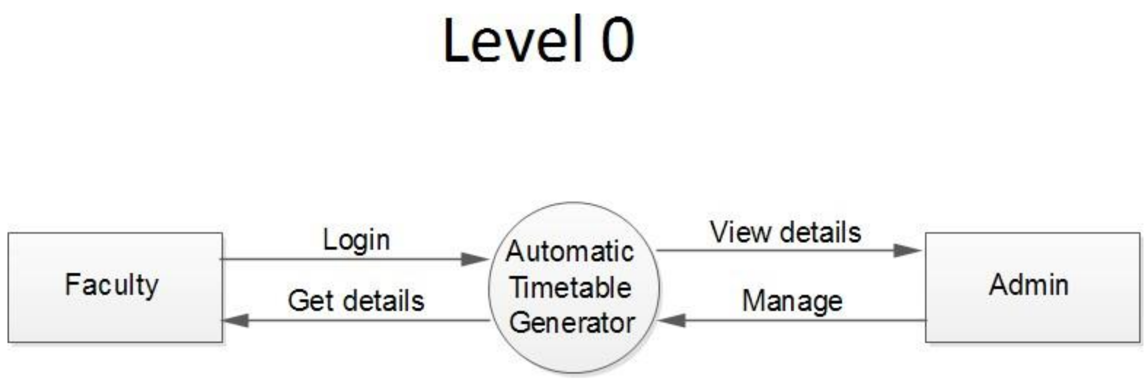
These aspects of the high-level functions are then represented in a diagrammatic form. This forms the top level Data Flow Diagram, usually called DFD.

2. The high-level functions described in the SRS Documents, and then each of the high-level functions can be represented in form of a bubble. Each high-level function is decomposed into constituent sub functions through the following set of activities.
   - Different sub functions of the high-level functions are identified.
   - Data input to each of these sub functions is identified.
   - Data output from each of these sub functions is identified.
   - Interactions among these sub functions are identified.

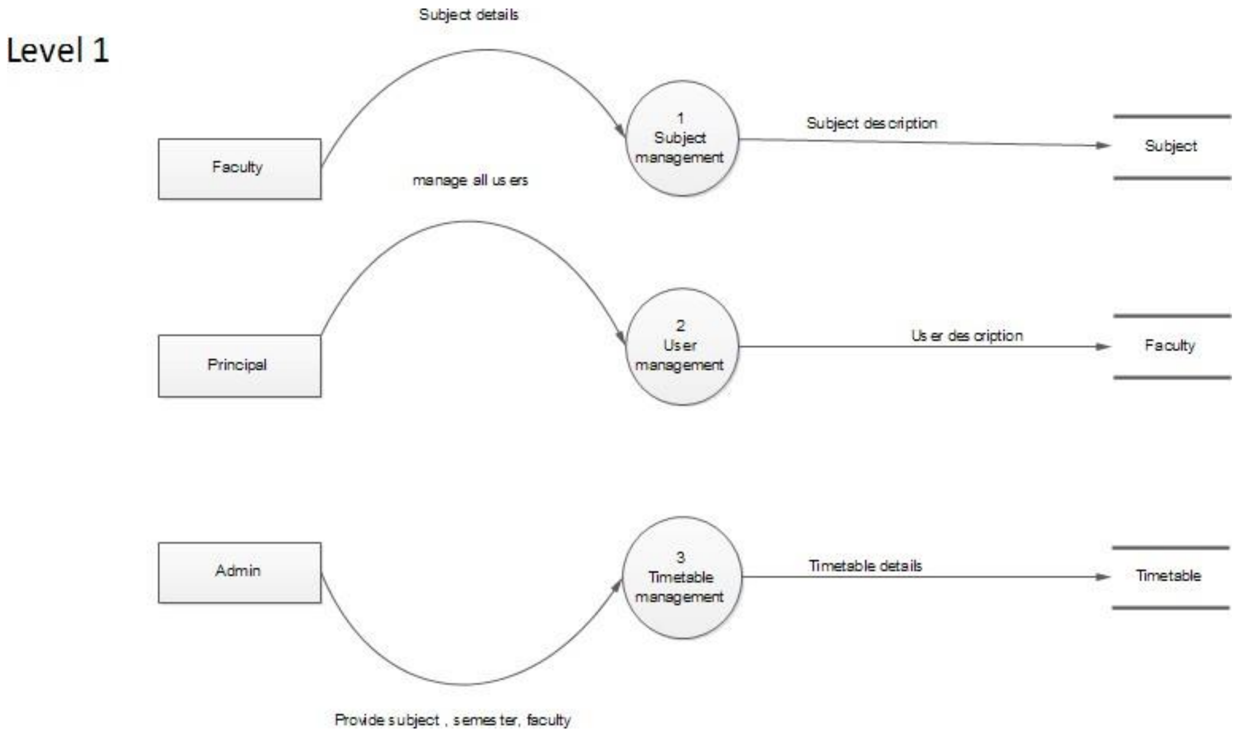
   These aspects are then represented in a diagrammatic form using a DFD.

3. Step three is repeated recursively for each sub function until a simple algorithm can represent a sub function

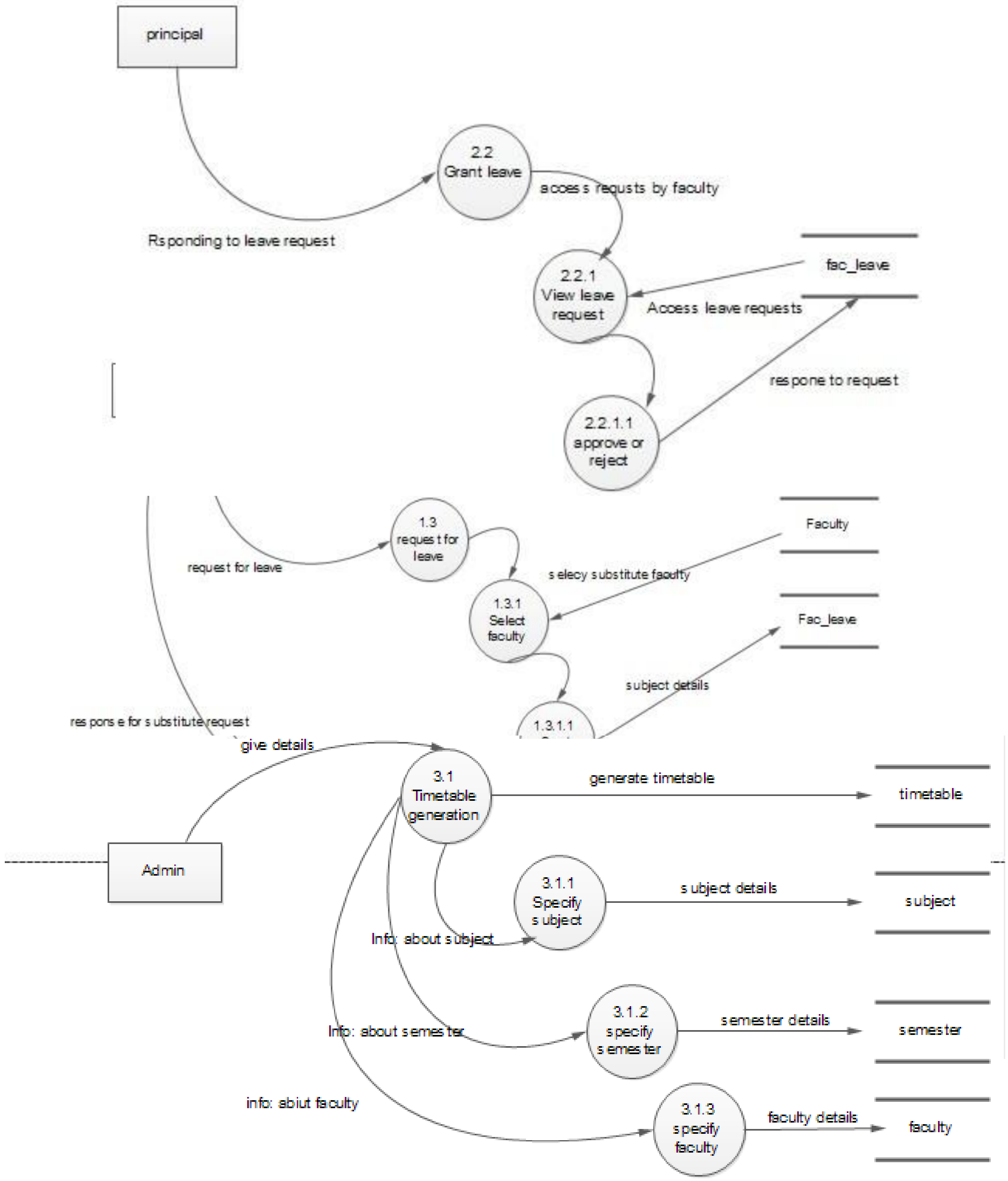
Level 0



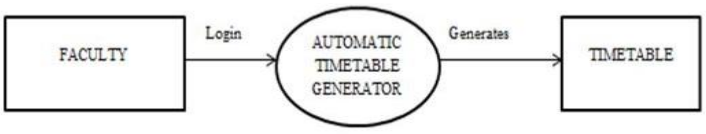
Level 1



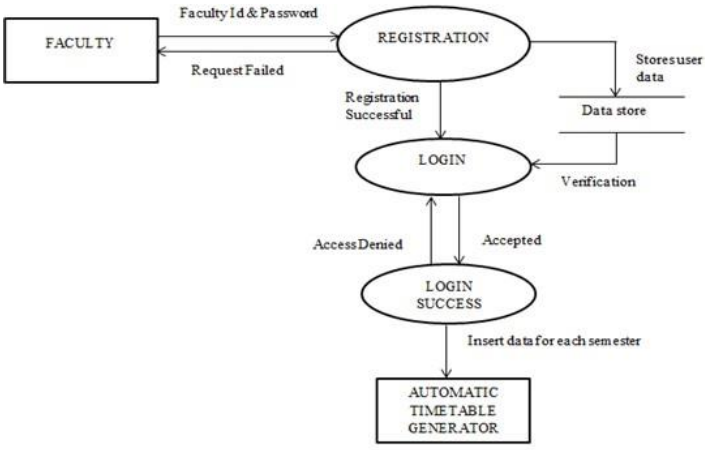
Level 2


principal
2.2
Grant leave
acces s requsts by faculty
Rsponding to leave request
fac_leave
2.2.1
View leave
request
Access leave requests
respone to request
2.2.1.1
approve or
reject
1.3
request for
leave
request for leave
Faculty
selecy substitute faculty
1.3.1
Select
faculty
Fac_leave
subject details
res ponse for substitute request
1.3.1.1
give details
3.1
Timetable
generation
generate timetable
timetable
Admin
3.1.1
Specify
subject
subject details
subject
Info: about subject
3.1.2
specify
semester
semester details
semester
Info: about semester
info: abiut faculty
3.1.3
specify
faculty
faculty details
faculty

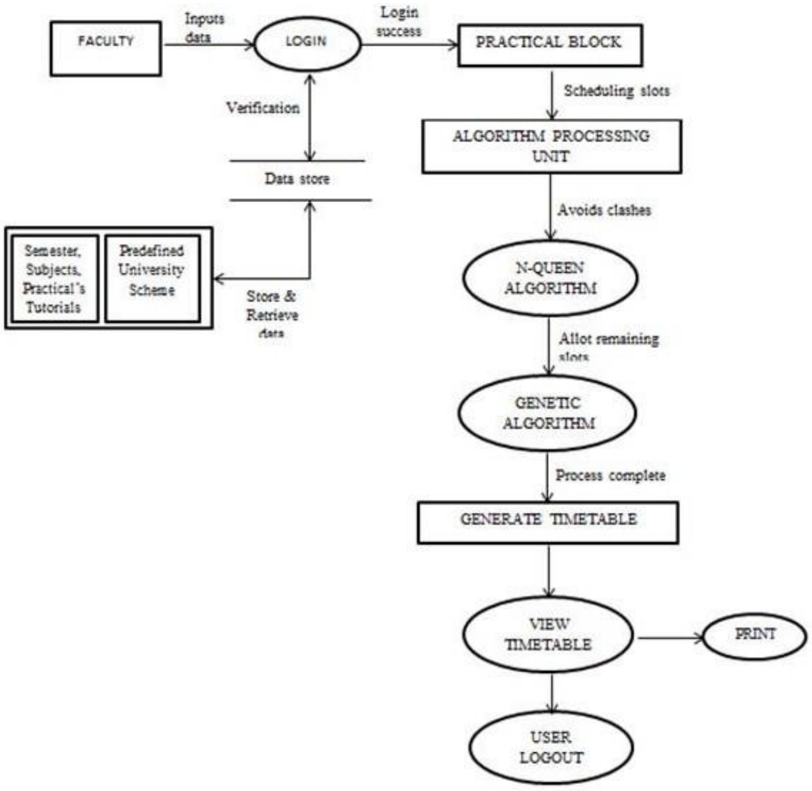
Data Flow Diagram



Level 0



Level 1



LEVEL 2

3.3 Database design

The system is implemented in such a way that all the valid information's are stored in the database. Database design converts the data model developed in logical designing to database definition. This is supported by database software. The management of data involves both the definition of structure of the storage of information and provision of mechanism for the manipulation of information. In addition database system must provide safety of information handle, despite the system crashes due to attempts of unauthorized access.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. In database design, several views of data must be considered along with the person who users them. The logical view is what the data look like, regardless of how they are stored. The physical view is the way data exist in physical storage. Tables are carefully designed aiming to achieve its main objectives ie, to make information access easy, quick, inexpensive and flexible for user.

NORMALIZATION

Normalization is the process of analyzing the given relation schemas based on their Functional Dependencies and primary keys to achieve the desirable properties of Minimizing Redundancy, Minimizing the insertion, deletion and updating anomalies.

Normalization is carried out for the following reasons:

- To structure the data so that perfect relationship between entries can be represented.
- To permit simple retrieval of data in response query and report requests.
- To reduce the need to restructure or reorganize data when new application requirement arises.

Normalization consists of various levels:

1. First Normal Form (1NF)

A table is in 1NF if there are no duplicate rows in the table. Each cell is single valued. Entries in a column are of the same kind.

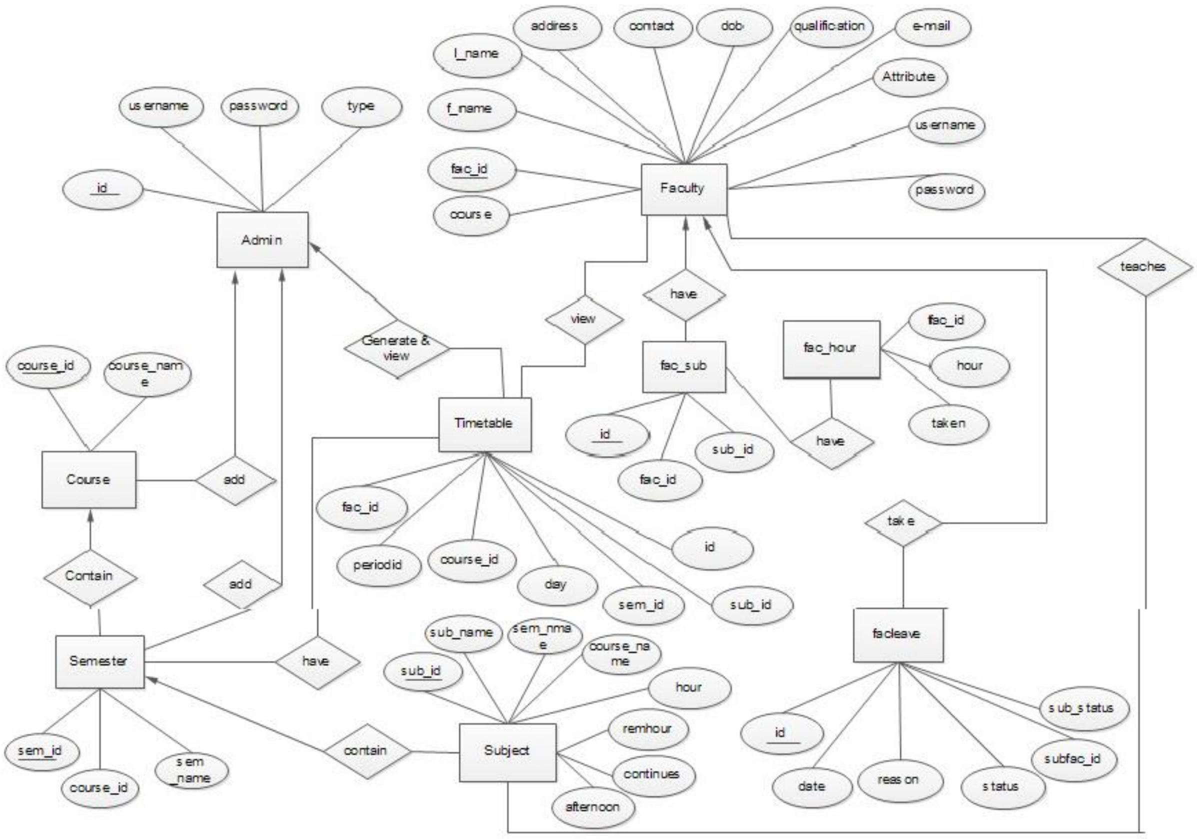
2. Second Normal Form (2NF)

Second Normal form is based on the concept of full functional dependency. A table (relation) is in 2NF if .It is in First Normal Form and if all non-key attributes are dependent on the key. Dependent on only a part of the (composite) key, the definition of 2NF is sometimes phrased as, "A table is in 2nF if it is in 1NF and if it has no partial dependencies.”

3. Third Normal Form (3NF)

Third Normal Form is based on the concept of transitive dependency. A table (relation) is in 3NF if it is in Second Normal Form and if it has no transitive dependencies

3.4 E-R DIAGRAM



3.5 Modules Description

Admin

- Login/Logout
- Manage Schools
- Manage Department
- Manage Students
- Manage Class Time Tables
- Manage Exam Time Tables
- Submit Notices
- Manage Users
- Admin Panel

Students

- Dashboard (Displays Time Tables scheduled for the current date and etc.)
- Personal Time Table Management
- Preview Student Details

Class Representative

- Dashboard
- Manage Class Time Tables
- Manage exam Time Tables
- Submit Notices

3.6 CODING AND CODE TEMPLATES

The system contains 3 users , Faculty , Principal and Admin. Each user have their own functionalities as follows .

Function : Subject allocation

Input :subject, faculty and semester

Output : timetable

Logic:

Step 1: Read subject , faculty and semester from the tables subject,faculty and semester respectively.

Step 2:Validate the details and processing is done

Step 3*:*subject allocated for facultys successfully in the table named Timetable.

Function : View Timetable

Input : Subject,Semester, Faculty

Output : Timetable

Logic:

Step 1: Read subject , faculty and semester from the tables subject, faculty and semester respectively.

Step 2: Timetable generated successfully in the table called timetable.

Function : Request for leave

Input *:* Date,Reason,Substitute

Output *:* Approve / Reject request

Logic:

Step 1: Read date, reason and substitute faculty from the tablefac_leave

Step 2: Get Substitute status from fac_leave

Step 3: Request succesfully sent (Approve / Reject)

If Facuty need to take a leave the he/she can apply for leave with setting Substitutes. Here Date at which leave is required , reason and substitute faculty should be specified.

Function : Substitute request

Input : Request

Output : Approve / Reject

Logic:

Step 1: Read the information and Apply for leave with providing date , reason and aubstitute facilty in the table fac_leave

Step 2: Substitute request successfully sent (approve/reject)

Function : Grant or deney

Input : Request

Output : Reply message sendApprove / Reject

Logic:

Step 1: Access substitute request send by the faculty from fac_leave

Step 2: Reply for Substitute request successfully sent (approve/reject)

Function : Grant leave

Input : Request

Output : Reply message send Approve / Reject

Logic:

Step 1: Access leave request send by the faculty from fac_leave

Step 2: Reply for Request for leave successfully sent (approve/reject)

Function : Subject allocation

Input :subject, faculty and semester

Output : timetable

Logic:

Step 1: Read subject , faculty and semester from the tables subject,faculty and semester respectively.

Step 2:Validate the details and processing is done

Step 3*:*Timetable generated Successfully

CODE:

```
<?php
// This script and data application were generated by AppGini 5.72
// Download AppGini for free from https://bigprof.com/appgini/download/

/*
 ajax-callable script that retrieves a list of users for admin, indicating which
ones have
 access to supplied table.

 REQUEST parameters:
 ===============
 t: table name
 id: optional, primary key value of current record
 p: page number (default = 1)
 s: search term
*/

 /* return json */
 header('Content-type: application/json');

 $start_ts = microtime(true);

 $curr_dir=dirname(__FILE__);
 require("{$curr_dir}/incCommon.php");

 // how many results to return per call, in case of json output
 $results_per_page = 50;

 $id = false;
 if(isset($_REQUEST['id'])) $id = from_utf8($_REQUEST['id']);

 $search_term = false;
 if(isset($_REQUEST['s'])) $search_term = from_utf8($_REQUEST['s']);

 $page = intval($_REQUEST['p']);
 if($page < 1) $page = 1;
 $skip = $results_per_page * ($page - 1);

 $table_name = $_REQUEST['t'];
 if(!in_array($table_name, array_keys(getTableList()))){
	/* invalid table */
	echo '{"results":[{"id":"","text":"Invalid
table"}],"more":false,"elapsed":0}';
	exit;
 }

 /* if id is provided, get owner */
 $owner = false;
 if($id){
	$owner = sqlValue("select memberID from
membership_userrecords where tableName='{$table_name}' and
pkValue='" . makeSafe($id) . "'");
 }

 $prepared_data = array();
 $where = "g.name!='{$adminConfig['anonymousGroup']}' and
p.allowView>0 ";
 if($search_term){
	$search_term = makeSafe($search_term);
	$where .= "and (u.memberID like '%{$search_term}%' or g.name
like '%{$search_term}%')";
 }
 $res = sql("select u.memberID, g.name from membership_users u left join
membership_groups g on u.groupID=g.groupID left join
membership_grouppermissions p on g.groupID=p.groupID and
p.tableName='{$table_name}' where {$where} order by g.name,
u.memberID limit {$skip}, {$results_per_page}", $eo);
 while($row = db_fetch_row($res)){
	$prepared_data[] = array('id' => to_utf8($row[0]), 'text' =>
to_utf8("<b>{$row[1]}</b>/{$row[0]}"));
 }

 echo json_encode(array(
        'results' => $prepared_data,
        'more' => (@db_num_rows($res) >= $results_per_page),
        'elapsed' => round(microtime(true) - $start_ts, 3)
 ));
```

```
<?php
 $currDir = dirname(__FILE__);
 require("{$currDir}/incCommon.php");
 $GLOBALS['page_title'] = $Translation['data records'];
 include("{$currDir}/incHeader.php");

 // process search
 $memberID = new Request('memberID', 'strtolower');
 $groupID = max(0, intval($_GET['groupID']));
 $tableName = new Request('tableName');
 $page = max(1, intval($_GET['page']));

 // process sort
 $sortDir = ($_GET['sortDir'] == 'desc' ? 'desc' : '');
 $sort = makeSafe($_GET['sort']);
 if($sort != 'dateAdded' && $sort != 'dateUpdated'){ // default sort is newly
created first
        $sort = 'dateAdded';
        $sortDir = 'desc';
 }

 if($sort){
        $sortClause = "order by {$sort} {$sortDir}";
 }

 if($memberID->sql != ''){
        $where .= ($where ? " and " : "") . "r.memberID like '{$memberID-
>sql}%'";
 }
 if($groupID){
         $where .= ($where ? " and " : "") . "g.groupID='{$groupID}'";
 }

 if($tableName->sql != ''){
         $where .= ($where ? " and " : "") . "r.tableName='{$tableName-
>sql}'";
 }

 if($where){
         $where = "where {$where}";
 }

 $numRecords = sqlValue("select count(1) from membership_userrecords r
left join membership_groups g on r.groupID=g.groupID {$where}");
 $noResults = false;
 if(!$numRecords){
         echo "<div class=\"alert alert-warning\">{$Translation['no
matching results found']}</div>";
         $noResults = true;
         $page = 1;
 }

 if($page > ceil($numRecords / $adminConfig['recordsPerPage']) &&
!$noResults){
         redirect("admin/pageViewRecords.php?page=" .
ceil($numRecords/$adminConfig['recordsPerPage']));
 }

 $start = ($page - 1) * $adminConfig['recordsPerPage'];

?>
<div class="page-header"><h1><?php echo $Translation['data records'] ;
?></h1></div>

<table class="table table-striped table-bordered table-hover">
 <thead>
```

```
    <tr>
        <th colspan="7">
            <form class="form-inline" method="get"
action="pageViewRecords.php" class="form-horizontal">
                <input type="hidden" name="page"
value="1">

                <div class="form-group">
                    <label for="groupID" class="control-
label"><?php echo $Translation["group"]; ?></label>
                    <?php echo
htmlSQLSelect("groupID", "select groupID, name from
membership_groups order by name", $groupID); ?>
                </div>
                <div class="form-group">
                    <label for="memberID"
class="control-label"><?php echo $Translation["member username"] ;
?></label>
                    <input type="text" class="form-
control" id="memberID" name="memberID" value="<?php echo
$memberID->attr; ?>">
                </div>
                <div class="form-group">
                    <label for="tableName"
class="control-label"><?php echo $Translation['show records'] ; ?></label>
                    <?php
                        $tables = array_merge(array(''
=> $Translation['all tables']), getTableList(true));
                        $arrFields =
array_keys($tables);
                        $arrFieldCaptions =
array_values($tables);
                        echo htmlSelect('tableName',
$arrFields, $arrFieldCaptions, $tableName->raw);
                    ?>
                </div>
                <div class="form-group">
```

```
                    <label for="sort" class="control-label"><?php echo $Translation['sort records'] ; ?></label>
                    <?php
                        $arrFields = array('dateAdded', 'dateUpdated');
                        $arrFieldCaptions = array( $Translation['date created'] , $Translation['date modified'] );
                        echo htmlSelect('sort', $arrFields, $arrFieldCaptions, $sort);
                    ?>
                    <span class="hspacer-md"></span>
                    <?php
                        $arrFields=array('desc', ");
                        $arrFieldCaptions = array( $Translation['newer first'] , $Translation['older first'] );
                        echo htmlSelect('sortDir', $arrFields, $arrFieldCaptions, $sortDir);
                    ?>
                </div>
                <div class="form-group">
                    <button type="submit" class="btn btn-primary"><i class="glyphicon glyphicon-search"></i> <?php echo $Translation['find'] ; ?></button>
                    <button type="button" id="reset-search" class="btn btn-warning"><i class="glyphicon glyphicon-remove"></i> <?php echo $Translation['reset'] ; ?></button>
                </div>
            </form>
        </th>
    </tr>
    <tr>
        <th>&nbsp;</td>
        <th><?php echo $Translation['username'] ; ?></th>
        <th><?php echo $Translation["group"] ; ?></th>
        <th><?php echo $Translation["table"] ; ?></th>
        <th><?php echo $Translation['created'] ; ?></th>
        <th><?php echo $Translation['modified'] ; ?></th>
```

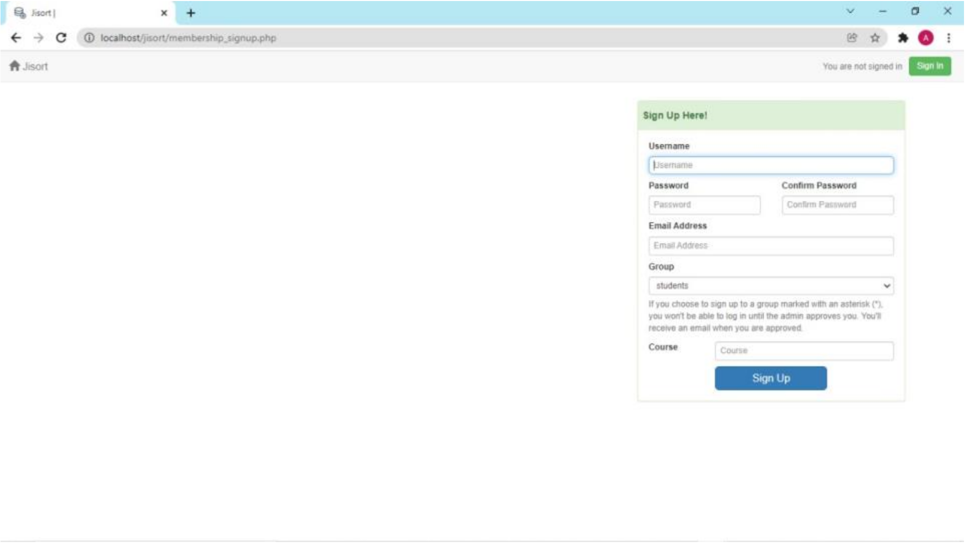
```
                <th><?php echo $Translation['data'] ; ?></th>
        </tr>
 </thead>
 <tbody>
<?php

 $res = sql("select r.recID, r.memberID, g.name, r.tableName, r.dateAdded,
r.dateUpdated, r.pkValue from membership_userrecords r left join
membership_groups g on r.groupID=g.groupID $where $sortClause limit
$start, " . $adminConfig['recordsPerPage'], $eo);
 while($row = db_fetch_row($res)){
        ?>
        <tr>
                <td class="text-center">
                        <a href="pageEditOwnership.php?recID=<?php
echo $row[0]; ?>" title="<?php echo $Translation['change record
ownership'] ; ?>"><i class="glyphicon glyphicon-user"></i></a>
                        <a href="pageDeleteRecord.php?recID=<?php echo
$row[0]; ?>" onClick="return confirm('<?php echo $Translation['sure delete
record'] ; ?>');" title="<?php echo $Translation['delete record'] ; ?>"><i
class="glyphicon glyphicon-trash text-danger"></i></a>
                </td>
                <td><?php echo $row[1]; ?></td>
                <td><?php echo $row[2]; ?></td>
                <td><?php echo $row[3]; ?></td>
                <td class="<?php echo ($sort == 'dateAdded' ? 'warning' :
");?>"><?php echo @date($adminConfig['PHPDateTimeFormat'],
$row[4]); ?></td>
                <td class="<?php echo ($sort == 'dateUpdated' ? 'warning' :
");?>"><?php echo @date($adminConfig['PHPDateTimeFormat'],
$row[5]); ?></td>
                <td>
                        <a href="#" class="view-record" data-record-
id="<?php echo $row[0]; ?>"><i class="glyphicon glyphicon-search
hspacer-md"></i></a>
                        <?php echo substr(getCSVData($row[3], $row[6]),
0, 80) . " ... "; ?>
```

```
        </td>
    </tr>
    <?php
}
?>
</tbody>
<tfoot>
    <tr>
        <td colspan="7" style="padding: .3em;">
            <table width="100%" cellspacing="0">
                <tr>
                    <th class="text-left flip"
style="width: 25%;">
                        <?php if($start){ ?>
                            <a
href="pageViewRecords.php?groupID=<?php echo $groupID;
?>&memberID=<?php echo $memberID->url; ?>&tableName=<?php echo
$tableName->url; ?>&page=<?php echo ($page > 1 ? $page - 1 : 1);
?>&sort=<?php echo $sort; ?>&sortDir=<?php echo $sortDir; ?>"
class="btn btn-default"><?php echo $Translation['previous'] ; ?></a>
                        <?php } ?>
                    </th>
                    <th class="text-center">
                        <?php
                            $record1 = $start + 1;
                            $record2 = $start +
db_num_rows($res);
                            $originalValues =
array('<RECORDNUM1>', '<RECORDNUM2>', '<RECORDS>');
                            $replaceValues =
array($record1, $record2, $numRecords);
                            echo
str_replace($originalValues, $replaceValues, $Translation['displaying
records']);
                        ?>
                    </th>
```

```
                                        <th class="text-right flip"
style="width: 25%;">
                                            <?php if($record2 <
$numRecords){ ?>
                                                <a
href="pageViewRecords.php?groupID=<?php echo $groupID;
?>&memberID=<?php echo $memberID->url; ?>&tableName=<?php echo
$tableName->url; ?>&page=<?php echo
($page<ceil($numRecords/$adminConfig['recordsPerPage']) ? $page+1 :
ceil($numRecords/$adminConfig['recordsPerPage'])); ?>&sort=<?php echo
$sort; ?>&sortDir=<?php echo $sortDir; ?>" class="btn btn-default"><?php
echo $Translation['next'] ; ?></a>
                                            <?php } ?>
                                        </th>
                                    </tr>
                                </table>
                            </td>
                        </tr>
 </tfoot>
</table>

<div class="modal fade" tabindex="-1" id="view-record-modal">
 <div class="modal-dialog modal-lg">
        <div class="modal-content">
                <button type="button" class="close hspacer-md vspacer-
md" data-dismiss="modal">&times;</button>
                <div class="modal-body" style="-webkit-overflow-
scrolling:touch !important; overflow-y: auto;">
                        <iframe width="100%" height="100%"
sandbox="allow-forms allow-scripts allow-same-origin" src="" id="view-
record-iframe"></iframe>
                </div>
        </div>
 </div>
</div>

<style>
```

```
 .form-inline .form-group{ margin: .5em 1em; }
</style>

<script>
 $j(function(){
        $j('.view-record').click(function(){
               var recID = $j(this).data('record-id');
               $j('#view-record-iframe').attr('src',
'pagePrintRecord.php?recID=' + recID);
               $j('#view-record-modal').modal('show');
               $j('#view-record-modal .modal-
body').height($j(window).height() * 0.7);

               return false;
        });

        $j('#reset-search').click(function(){
               window.location = 'pageViewRecords.php';
        });

        $j('#tableName, #groupID, #sort, #sortDir').addClass('form-
control');
 })
</script>

<?php
 include("{$currDir}/incFooter.php");
```

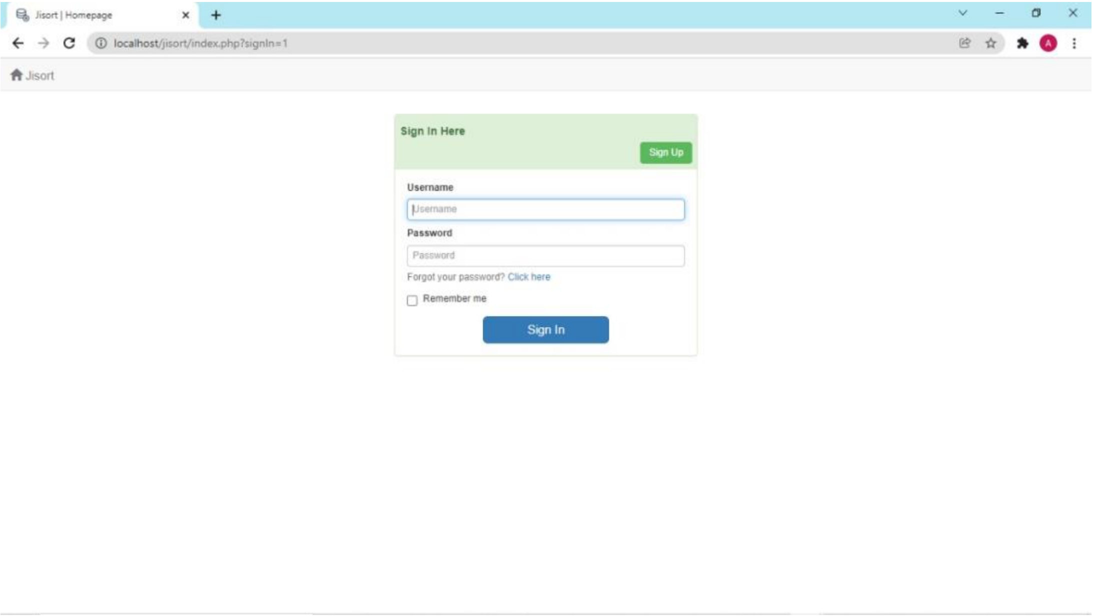
CHAPTER-4
RESULTS



Group: students

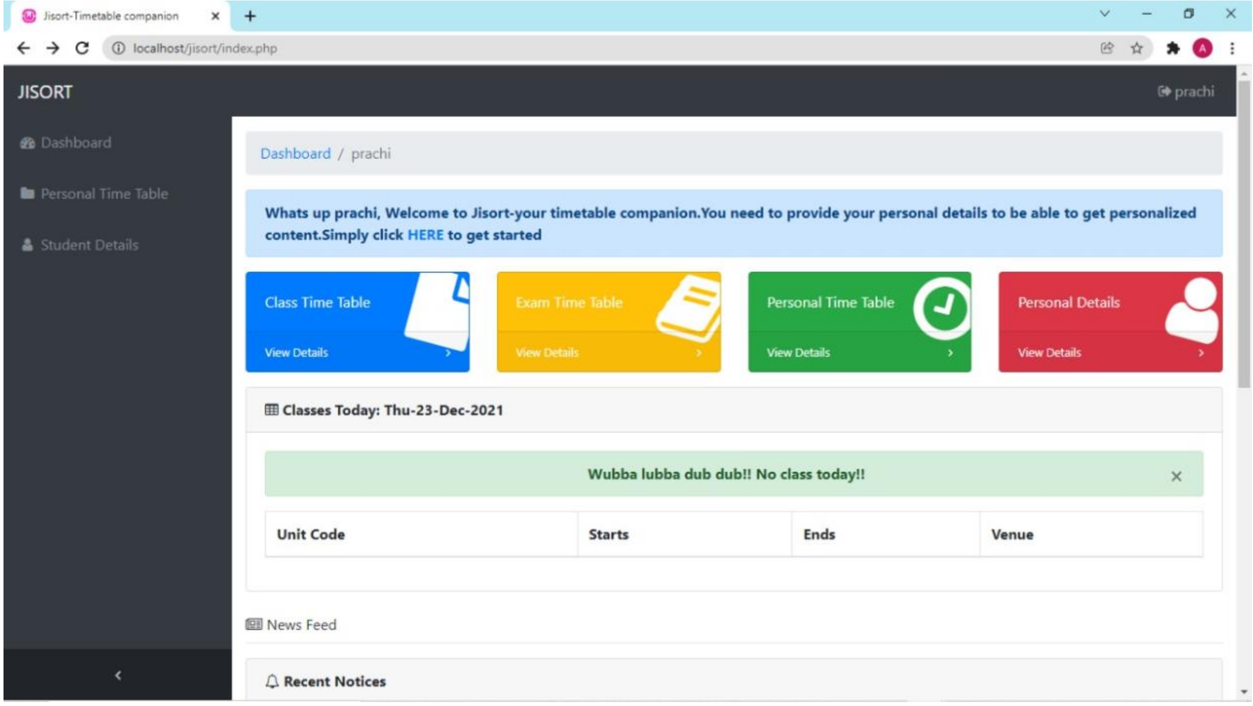
If you choose to sign up to a group marked with an asterisk (*), you won't be able to log in until the admin approves you. You'll receive an email when you are approved.

SIGN UP



□ Remember me

SIGN IN



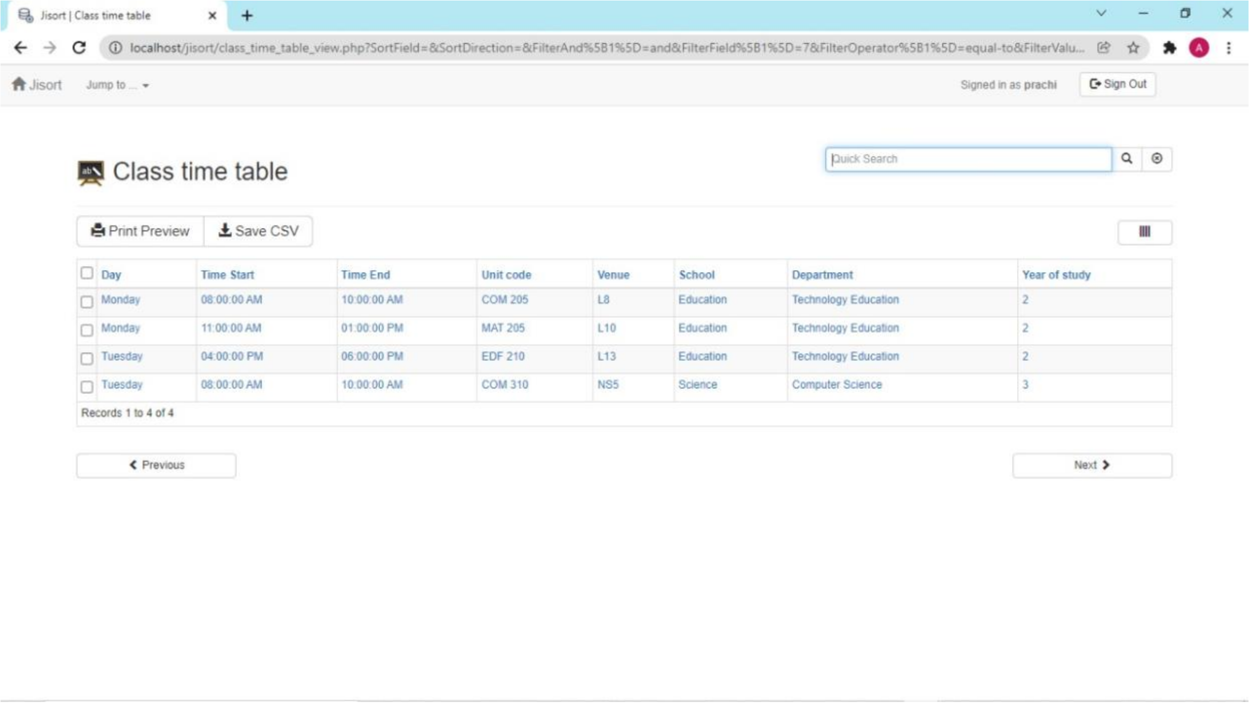
Whats up prachi, Welcome to Jisort-your timetable companion.You need to provide your personal details to be able to get personalized content.Simply click HERE to get started

Classes Today: Thu-23-Dec-2021

Wubba lubba dub dub!! No class today!!

Unit Code	Starts	Ends	Venue

DASHBOARD


Jisort | Class time table
localhost/jisort/class_time_table_view.php?SortField=&SortDirection=&FilterAnd%5B1%5D=and&FilterField%5B1%5D=7&FilterOperator%5B1%5D=equal-to&FilterValu...
Jisort
Jump to ...
Signed in as prachi
Sign Out

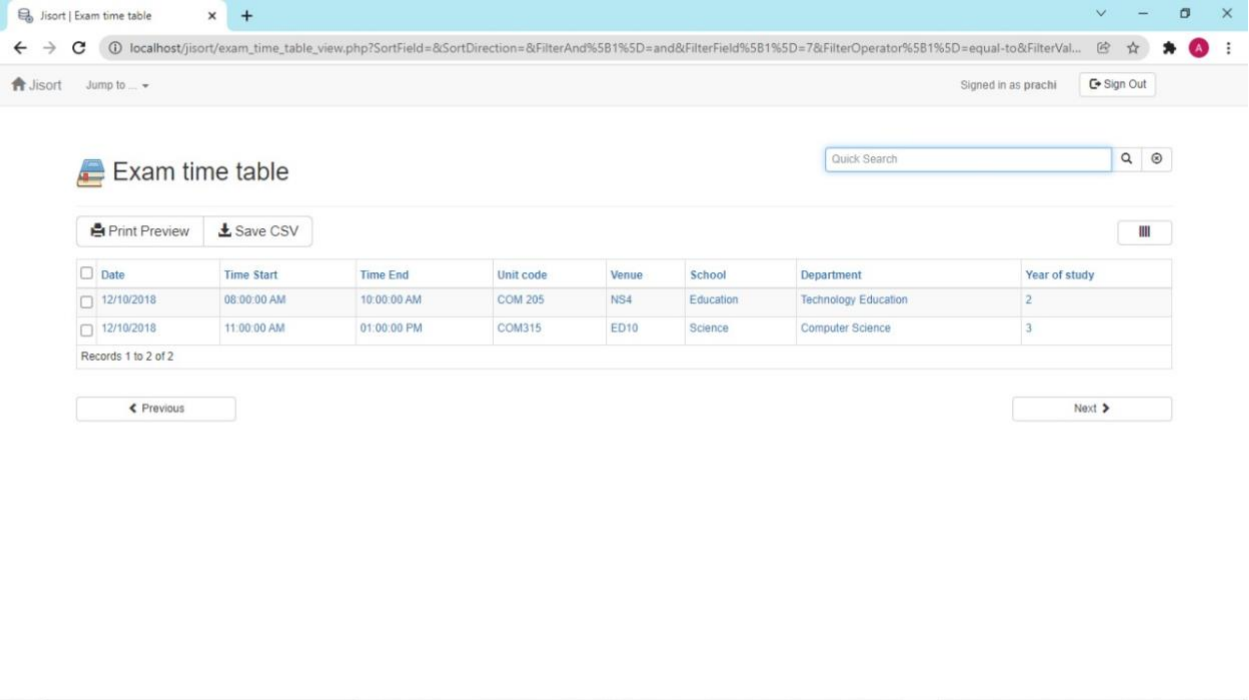
Class time table

Quick Search
Print Preview
Save CSV

□	Day	Time Start	Time End	Unit code	Venue	School	Department	Year of study
□	Monday	08:00:00 AM	10:00:00 AM	COM 205	L8	Education	Technology Education	2
□	Monday	11:00:00 AM	01:00:00 PM	MAT 205	L10	Education	Technology Education	2
□	Tuesday	04:00:00 PM	06:00:00 PM	EDF 210	L13	Education	Technology Education	2
□	Tuesday	08:00:00 AM	10:00:00 AM	COM 310	NS5	Science	Computer Science	3

Records 1 to 4 of 4

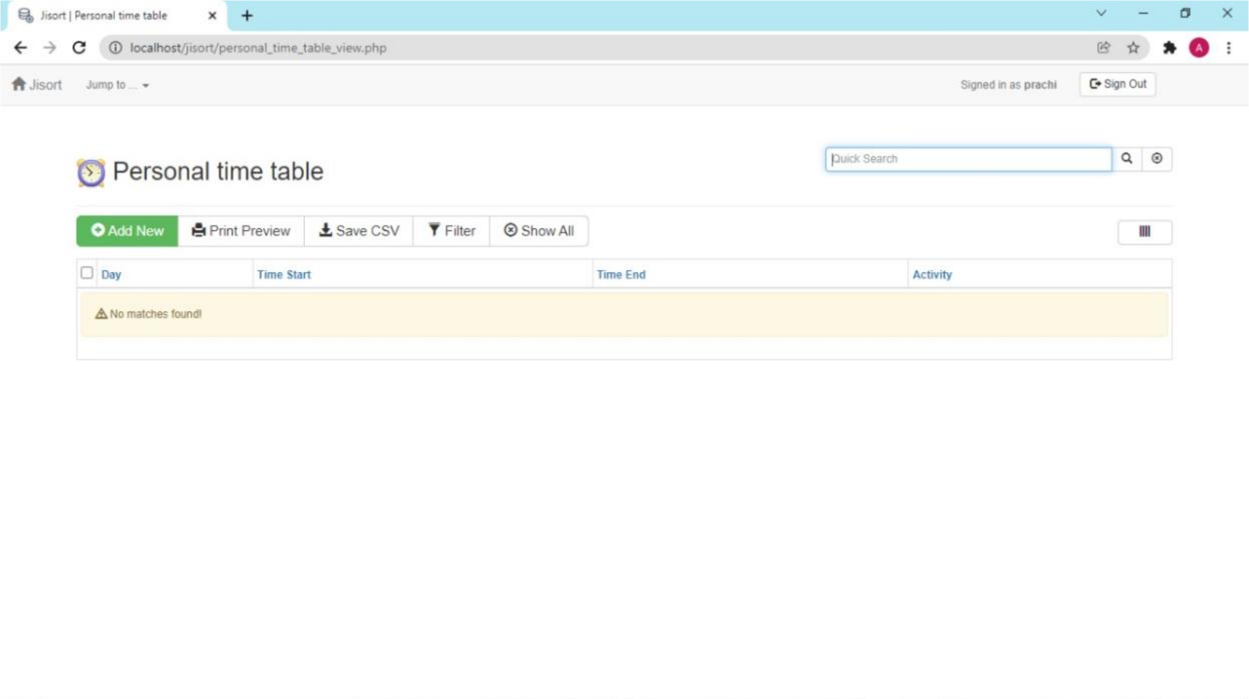
CLASS TIME TABLE


Jisort | Exam time table
localhost/jisort/exam_time_table_view.php?SortField=&SortDirection=&FilterAnd%5B1%5D=and&FilterField%5B1%5D=7&FilterOperator%5B1%5D=equal-to&FilterVal...
Jisort
Jump to ...
Signed in as prachi
Sign Out
Quick Search
Exam time table
Print Preview
Save CSV
Previous
Next

□	Date	Time Start	Time End	Unit code	Venue	School	Department	Year of study
□	12/10/2018	08:00:00 AM	10:00:00 AM	COM 205	NS4	Education	Technology Education	2
□	12/10/2018	11:00:00 AM	01:00:00 PM	COM315	ED10	Science	Computer Science	3

Records 1 to 2 of 2

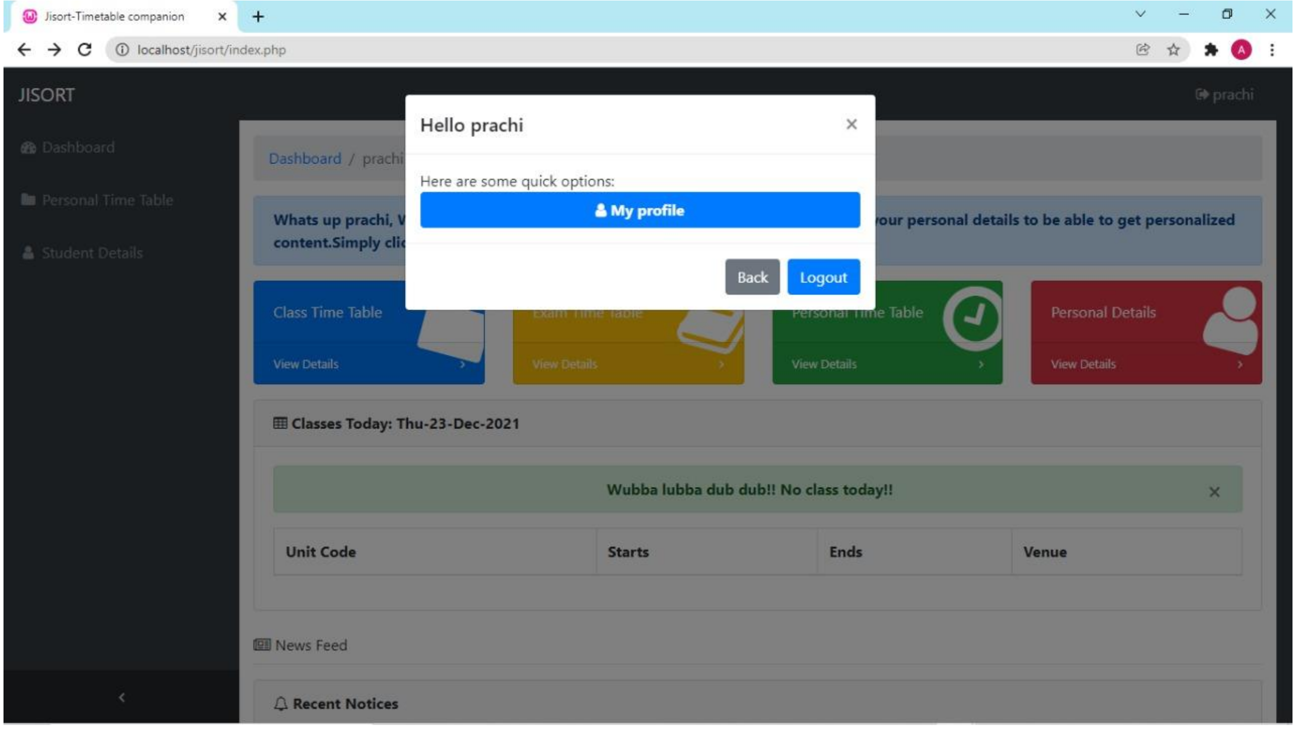
EXAM TIME TABLE



□	Day	Time Start	Time End	Activity

⚠ No matches found!

PEWRSONAL DETAILS



Hello prachi

Here are some quick options:

Whats up prachi, W… …your personal details to be able to get personalized content.Simply clic…

Classes Today: Thu-23-Dec-2021

Wubba lubba dub dub!! No class today!!

Unit Code	Starts	Ends	Venue

LOG OUT

CHAPTER-5
CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

Automatic Timetable Generator is a web based application for generating timetable automatically. It is a great difficult task that to manage many Faculty's and allocating subjects for them at a time manually. So proposed system will help to overcome this disadvantage. Thus we can generate timetable for any number of courses and multiple semesters. This system will help to create dynamic pages so that for implementing such an system we can make use of the different tools are widely applicable and free to use also. The purpose of the rule to get a timetable schedule mechanically is happy. The rule incorporates variety of methods, aimed to enhance the potency of the search operation. It also, addresses the necessary exhausting constraint of clashes during the supply of lecturers. The non stiff soft constraints i.e. optimization subjective for the search operation also are effectively handled. Taken the generality of the rule operation, it will additional be tailored to a lot of specific eventualities, e.g. School, examination planning and additional be increased to make railway time tables. Thus, through the method of automation of the time-table drawback, several an-hours of making a good timetable are decreased eventually. The foremost fascinating future directions within the development of the rule arise in its extension to constraint propagation. Once there's a price appointed to a variable, such assignment is propagated to unassigned variables to allow all values that get struggle with the present assignments. the data concerning such prohibited worth's is propagated furthermore.

5.2 FUTURE SCOPE

The Automatic Timetable Generator is driven portal for educational organization and is a web based application which will be helpful for creating Timetabels . This project will be a great helpful for the institiutions because, It is a great difficult task that to manage many Faculty's and allocating subjects for them at a time manually and this project will help to manage it properly. This manage timetable for faculty with considering maximum and minimum workload ,and can be managed easily .

REFERENCES

BOOKS

- Software Engineering Fifth edition by R. S.Pressman, McGraw-Hill Publication,1997
- DiptiShrinivasan “automated time table generation using multiple context reasoning for university modules” Published in: evolutionary computation, 2002. cec '02. Proceedings of the 2002 congress on (volume:2)
- AnujaChowdhary “TIME TABLE GENERATION SYSTEM” .Vol.3 Issue.2, February-2014, pg. 410-414
- Anirudha Nanda “An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach”. International Journal of Machine Learning and Computing, Vol. 2, No. 4, August 2012.

WEBSITES

http://en.wikipedia.org/wiki/MySQL

http://www.wpi.edu/Serlvets & JSP - Falkner Jones.pdf

http://edutechwiki.unige.ch/en/Educational_technology

http://www.wpi.edu/Images/Tutorial_JSP

http://en.wikipedia.org/wiki/JSP