Visvesvaraya Technological University

Jnana Sangama, Santhibastawad Road, Machhe
Belagavi – 590018, Karnataka, India

A PROJECT REPORT
ON

"TIMETABLE MANAGEMENT SYSTEM"

Submitted in partial fulfilment of the requirements for the DBMS Laboratory with Mini Project (18CSL58)

Bachelor of Engineering
in
Information Science & Engineering

for the Academic Year 2020-21

Submitted by

Patel Kavan	[1JS18IS063]
Rajiv Ranjan Singh	[1JS18IS072]

Under the Guidance of
Mrs. Sowmya K.N.
Asst. Professor, Dept. of ISE, JSSATE
Mrs. Punitha M
Asst. Professor, Dept. of ISE, JSSATE

Department of Information Science & Engineering
JSS Academy of Technical Education
JSS Campus, Dr. Vishnuvardhan Road, Bengaluru – 560060

2020-21

JSS Mahavidyapeetha, Mysuru

JSS Academy of Technical Education

JSS Campus, Dr. Vishnuvardhan Road, Bengaluru – 560060

Department of Information Science & Engineering

A PROJECT REPORT

ON

Certificate

This is to certify that the project work entitled **“TIMETABLE MANAGEMENT SYSTEM”** is a bonafide work carried out by **Mr. PATEL KAVAN (1JS18IS063)** and **Mr. RAJIV RANJAN SINGH (1JS18IS072)** in partial fulfilment for the DBMS Laboratory with Mini Project (18CSL58) of 5th semester **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during the year 2020-2021. It is certified that all corrections and suggestions have been incorporated in the report deposited in the department library.

Signature of the Guide	Signature of the HOD
Mrs. Sowmya K.N.	**Dr. Dayananda P**
Mrs. Punitha M	
Asst. Professor, Dept of ISE	**Prof. & Head,** Dept of ISE
JSSATE, Bengaluru	JSSATE, Bengaluru

Name of the Examiners	Signature with Date
1)	....
2)	....

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. So with gratitude, we acknowledge all those whose guidance and encouragement crowned my effort with success.

First and foremost, we would like to thank his **Holiness Jagadguru Sri Shivarathri Deshikendra Mahaswamiji** and **Dr. Mrityunjaya V Latte**, Principal, JSSATE, Bangalore for providing an opportunity to carry out the Presentation as a part of my curriculum in the partial fulfilment of the degree.

We express our sincere gratitude for my beloved Head of the department, **Dr. Dayananda P**, for his co-operation and encouragement at all the moments of my approach.

It is our pleasant duty to place on record our deepest sense of gratitude to our respected guides **Mrs. Sowmya K.N. and Mrs. Punitha M (**Asst. Professors), for the constant encouragement, valuable help and assistance in every possible way for continuous cooperation and support.

We would like to thank all **teaching** and **non-teaching faculty** for providing us with their valuable guidance and for being there at all stages of our work.

Patel Kavan

Rajiv Ranjan Singh

ABSTRACT

The Timetable Management System was developed for local use by any college institution to maintain a centralized database for timetables across various departments.

The database will store the timetables of all classes in all semesters of all departments, from which the timetables of all teachers and lab instructors can be extracted. Any teacher can view her free slots as well as that of any other faculty member. This can also be accessed by students. The faculty will have an option to show that they are not free at any particular time if they wish to do so.

The database also keeps track of all vacant classes at any particular time, so that it can be used by any other teacher if the need arises. This application can run on a low-end server easily to offer a smooth and secure environment to manage a college-level fest in a very professional way.

TABLE OF CONTENTS

Chapter 1:
PREAMBLE

1.1. Introduction

A database management system (DBMS) refers to the technology for creating and managing databases. Basically, is a software tool to organize (create, read, update, delete) data in the database. The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. A datum is a unit of data, meaningful data combined to form information. Hence, information is interpreted data – data provided with semantics. Microsoft Access is one of the most common examples of database management software. In other words, it is a group/package of information that is put in order so that it can be easily accessed, managed and updated.

The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data. As long as programs use the application programming interface (API) for the database that is provided by the DBMS, developers won‟t have to modify programs just because changes have been made to the database.

1.1.1. A History of DBMS

A Database Management System allows a person to organize, store, and retrieve data from a computer. It is a way of communicating with a computer‟is "stored memory." In the very early years of computers, "punch cards" were used for input, output, and data storage. Punch cards offered a fast way to enter data, and to retrieve it. Herman Hollerith is given credit for adapting the punch cards used for weaving looms to act as the memory for a mechanical tabulating machine, in 1890. Much later, Databases (or DBs) have played a very important part in the recent evolution of computers. The first computer programs were developed in the early 1950s, and focused almost completely on coding languages and algorithms. At the time, computers were basically giant calculators and data (names, phone numbers) was considered the leftovers of processing information. Computers were just starting to become commercially available, and when business people started using them for real-world purposes, this leftover data suddenly became important.

Enter the Database Management System (DBMS). A database, as a collection of information, can be organized so a Database Management System can access and pull specific information. In 1960, Charles W. Bachman designed the Integrated Database System, the "first" DBMS. IBM, not wanting to be left out, created a database system of their own, known as IMS. Both database systems are described as the forerunners of navigational databases.

By the mid-1960s, as computers developed speed and flexibility, and started becoming popular, many kinds of general use database systems became available. As a result, customers demanded a standard be developed, in turn leading to Bachman forming the Database Task Group. This group took responsibility for the design and standardization of a language called Common Business Oriented Language (COBOL). The Database Task Group presented this standard in 1971, which also came to be known as the "CODASYL approach".

1.1.2. Normalization

Database normalization is a database schema design technique, by which an existing schema is modified to minimize redundancy and dependency of data. Normalization splits a large table into smaller tables and defines relationships between them to increase the clarity in organizing data. Database normalization types are listed below.

First Normal Form (1NF):

- First normal form (1NF) deals with the `shape' of the record type
- A relation is in 1NF if, and only if, it contains no repeating attributes or groups of attributes

Second Normal Form (2NF):

- A relation is in 2NF if, and only if, it is in 1NF and every non-key attribute is fully functionally dependent on the whole key

Third Normal Form (3NF):

- A relation is in 3NF if, and only if, it is in 2NF and there are no transitive functional dependencies
- Transitive functional dependencies arise:
    - When one non-key attribute is functionally dependent on another non-key attribute:
    - When there is redundancy in the database

Boyce-Codd Normal Form (BCNF):

- When a relation has more than one candidate key, anomalies may result even though the relation is in 3NF
- 3NF does not deal satisfactorily with the case of a relation with overlapping candidate keys
    - Composite candidate keys with at least one attribute in common
    - BCNF is based on the concept of a determinant

Fourth Normal Form (4NF):

It is a normal form used in database normalization. Introduced by Ronald Fagin in 1977, 4NF is the next level of normalization after Boyce–Codd normal form (BCNF). Whereas the second, third, and

Boyce–Codd normal forms are concerned with functional dependencies, 4NF is concerned with a more general type of dependency known as a multivalued dependency.

1.2. About the Project

The Timetable Management System was developed for local use by any college institution to maintain a centralized database for timetables across various departments. The database will store the timetables of all classes in all semesters of all departments, from which the timetables of all teachers and lab instructors can be extracted. Any teacher can view her free slots as well as that of any other faculty member. This can also be accessed by students. The faculty will have an option to show that they are not free at any particular time if they wish to do so.

Chapter 2:
LITERATURE SURVEY

2.1. Problem Statement

The Timetable Management System was developed for local use by any college institution to maintain a centralized database for timetables across various departments. The database will store the timetables of all classes in all semesters of all departments, from which the timetables of all teachers and lab instructors can be extracted. Any teacher can view her free slots as well as that of any other faculty member. This can also be accessed by students. The faculty will have an option to show that they are not free at any particular time if they wish to do so.

This application aims to provide a system that eases the management of timetables that are being organised and the list of faculty who is going to take classes of different subjects.

2.2. Objectives

The system aims to check the following objectives:

- Any teacher can view her free slots as well as that of any other faculty member. This can also be accessed by students.
- If the timetable of a faculty is changed or they have any other works or are absent, the database can be updated and the timetable of the respective teacher is reorganized.
- The faculty will have an option to show that they are not free at any particular time if they wish to do so.
- The database also keeps track of all vacant classes at any particular time, so that it can be used by any other teacher if the need arises.

2.3. System Specifications

2.3.1. Hardware Specifications

- **Processor**: 8th gen Intel Core i5
- **System bus**: 64-bit
- **RAM**: 8GB

2.3.2. Software Specifications

- **Operating system**: macOS Big Sur 11.1
- **Frontend**: HTML, CSS and Bootstrap
- **Backend**: PHP
- **Query Language**: SQL
- **RDBMS**: PostgreSQL

2.4. Feasibility Study

2.4.1. Technical Feasibility

Technical Feasibility deals with the hardware as well as software requirements. Technology is not a constraint to type system development. We have to find out whether the necessary technology, the proposed equipment have the capacity to hold the data, which is used in the project, should be checked to carry out this technical feasibility. The technical feasibility issues usually raised during the feasibility stage of investigation includes these:

- The hardware required is a Pentium based server.
- The system can be expanded.

2.4.2. Behavioural Feasibility

This feasibility test asks if the system will work when it is developed and installed.
Operational feasibility in this project:

- The proposed system offers a greater level of user-friendliness.
- The proposed system produces best results and gives high performance. It can be implemented easily. So this project is operationally feasible

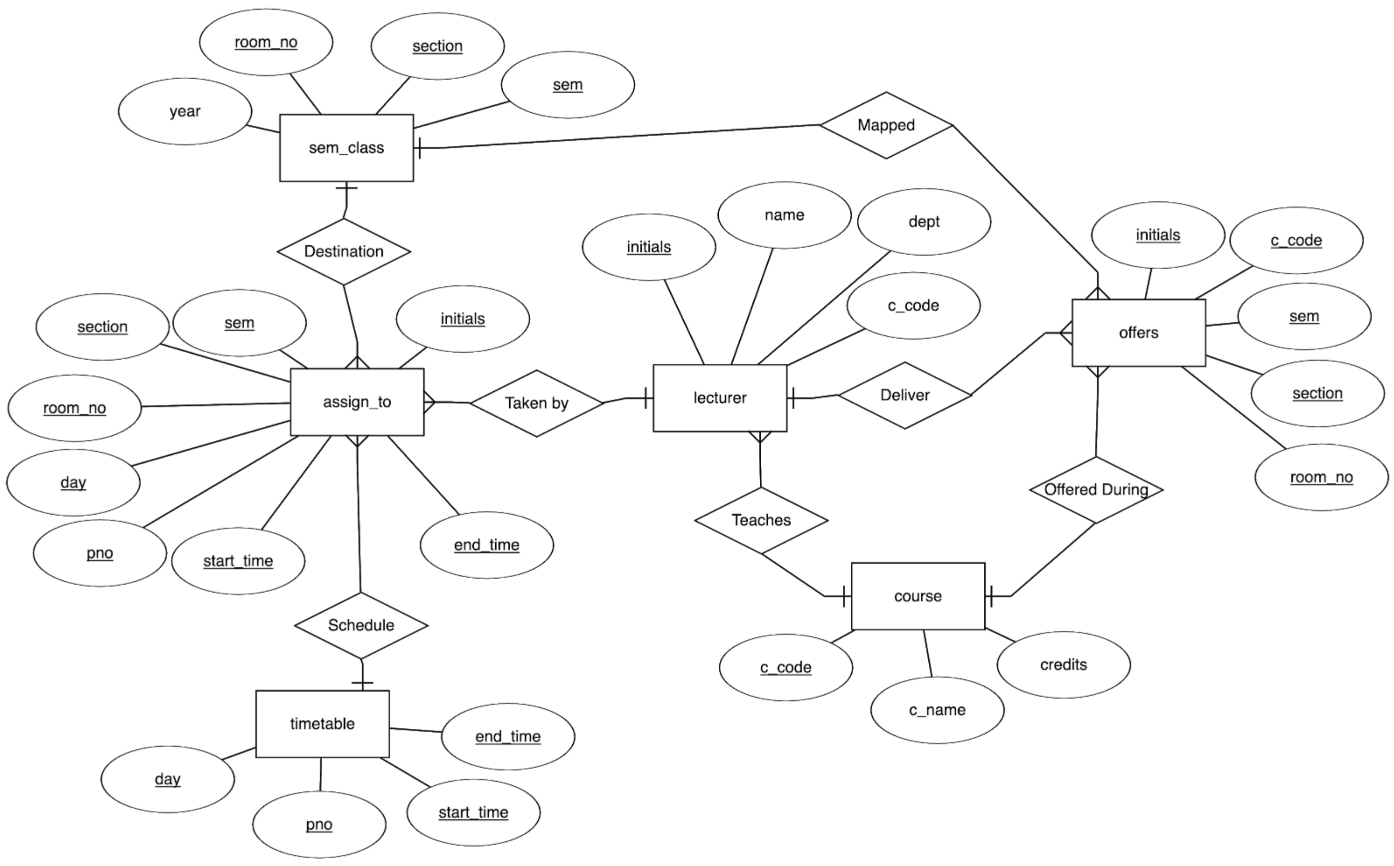
2.4.3. Economical Feasibility

Economic Feasibility deals about the economic impact faced by the organization to implement a new system. Financial benefits must equal or exceed the costs. The cost of conducting a full system, including software and hardware cost for the class of application being considered should be evaluated. Economic Feasibility in this project:

- The cost to conduct a full system investigation is possible.
- There is no additional manpower requirement.
- There is no additional cost involved in maintaining the proposed system.

Chapter 3:
SYSTEM DESIGN

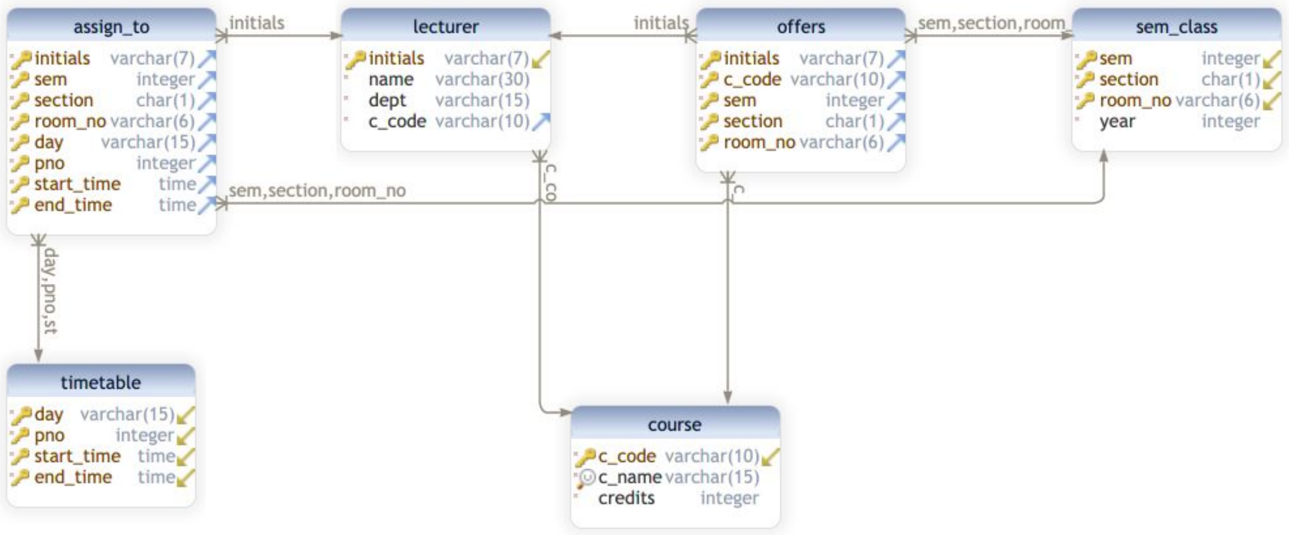
3.1. ER Diagram



An entity–relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types).

In software engineering, an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure which can be implemented in a database, typically a relational database.

3.2. Relational Schema Diagram



The schema diagram of a database system is its structure described in a formal language supported by the database management system (DBMS). The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database.

The term “schema” refers to the organization of data as a blueprint of how the database is constructed. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in realization of the database language. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modelled in the database.

The above schema is defined for the project. All the various tables used are described in the following schema. The necessary primary keys and the corresponding foreign keys are also represented.

Chapter 4:
IMPLEMENTATION

4.1. Front End and Back End Used

4.1.1. Front End – HTML and CSS

HTML is used as the front end tool to design web pages because:

- It is easy to write, use and understand.
- HTML also allows the use of templates, which makes designing a webpage easy.
- All browsers support HTML.

CSS is used along with html to design the web pages as it is relatively easy to learn and produces better and cleaner code than applying all the styles directly to the HTML code. Also the following reasons make CSS for helpful:

- Easy to maintain and update.
- Greater consistency in design and formatting options.
- Greater accessibility.

4.1.2. Back End – PHP and PostgreSQL

PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and SQL compliance. It was originally named POSTGRES, referring to its origins as a successor to the Ingres database developed at the University of California, Berkeley. It is a stable, reliable and powerful solution with advanced features like the following:

- Compatible with various platforms using all major languages and middleware
- It offers a most sophisticated locking mechanism
- Support for multi-version concurrency control
- Mature Server-Side Programming Functionality
- Compliant with the ANSI SQL standard
- Full support for client-server network architecture
- Log-based and trigger-based replication SSL
- Standby server and high availability
- Object-oriented and ANSI-SQL2008 compatible
- Support for JSON allows linking with other data stores like NoSQL which act as a federated hub

for polyglot databases.

PHP (Hypertext Preprocessor) is a server-side web programming language that is widely used for web development. MySQL is used with PHP as the back end tool in our web application.

- PHP also has powerful output buffering that further increases over the output flow.
- PHP is dynamic. PHP works in combination with HTML to display dynamic elements on the webpage.
- PHP can be used with a large number of relational database management systems and runs on all of the most popular web servers and is available for many different operating systems.

4.2. Discussion of Code Segments

4.2.1. Establish Connection with the Database

```
<?php
//Need to enter database name, username and password
$DATABASE_HOST = 'localhost';
$DATABASE_USER = ''; //Fill this
$DATABASE_PASS = ''; //Fill this
$DATABASE_NAME = ''; //Fill this
$db_connection = pg_connect("host='$DATABASE_HOST' port='5432' dbname='$DATABASE_NAME'
user='$DATABASE_USER' password='$DATABASE_PASS'") or die("unable to connect to database");
?>
```

The above code establishes connection with the database by taking into account the username and password for the PostgreSQL account and also the name of the database it is trying to establish the connection with. “$db_connection” is the variable that holds the connection.

4.2.2. Converting initials to caps and little bit of string formatting

```
if (isset($_POST["initials"]))
{
  $initials = strtoupper($_POST["initials"]);

  //In database Doctor is present as Dr
  if (strpos($initials, "DR") !== false)
  {
     $str_arr = explode('.', $initials);
     $str_arr[0] = "Dr";
     $initials = implode(".", $str_arr);
```

```
   }
}
```

4.2.3. Converts date to day

```
if (isset($_POST["date"]))
{
   $date = strtotime($_POST["date"]);
   $day = date("l", $date);
}
```

4.2.4. Frontend Code

```
<!DOCTYPE html>
<html>
<head>
          <title>Timetable Management System</title>
          <meta name="viewport" content="width=device-width, initial-scale=1">
          <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
          <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
          <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
          <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
   <meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<body>

                    <nav class="navbar navbar-default navbar-light bg-light shadow">
                              <a href="https://jssateb.ac.in/"><img src="logo.png" class="img-fluid"
width="50%"></a>
                              <a class="navbar-brand ml-auto text-secondary pull-left"
href="index.php"><h3>TIMETABLE MANAGEMENT SYSTEM</h3></a>
      </nav>

                    <div class="container pt-5">

                              <ul class="nav nav-pills mb-3" id="pills-tab" role="tablist">

                                        <li class="nav-item">
                                        <a class="nav-link active" id="pills-par-timetable-tab" data-toggle="pill"
href="#pills-par-timetable" role="tab" aria-controls="pills-home" aria-selected="true">Particular
Timetable</a>
                                        </li>

                                        <li class="nav-item">
                                        <a class="nav-link" id="pills-fulltime-tab" data-toggle="pill"
href="#pills-fulltime" role="tab" aria-controls="pills-fulltime" aria-selected="false">Full Timetable</a>
                                        </li>

                                        <li class="nav-item">
```

```
				<a class="nav-link" id="pills-freeslots-tab" data-toggle="pill"
href="#pills-freeslots" role="tab" aria-controls="pills-freeslots" aria-selected="false">Free Slots</a>
				</li>

				<li class="nav-item">
				<a class="nav-link" id="pills-vacancy-tab" data-toggle="pill"
href="#pills-vacancy" role="tab" aria-controls="pills-vacancy" aria-selected="false">Class Vacancy</a>
				</li>

				<li class="nav-item">
				<a class="nav-link" id="pills-findlec-tab" data-toggle="pill"
href="#pills-findlec" role="tab" aria-controls="pills-findlec" aria-selected="false">Find Lecturer</a>
				</li>

			</ul>
			<div class="tab-content" id="pills-tabContent">

				<div class="tab-pane fade show active" id="pills-par-timetable"
role="tabpanel" aria-labelledby="pills-par-timetable-tab">
					<form action="request.php" method="POST">

						<div class="form-group">
							<label for="initials"><strong>Faculty
Initials:</strong></label>
							<input type="text" class="form-control" name="initials"
placeholder="Enter Initials" required>
						</div>

						<div class="form-group">
							<label for="date"><strong>Date:</strong></label>
							<input type="date" class="form-control" name="date"
id="date" placeholder="Select the Date" required>
						</div>

						<button type="submit" class="btn btn-primary"
name="time_date">Get Info</button>

					</form>

				</div>

				<div class="tab-pane fade" id="pills-fulltime" role="tabpanel"
aria-labelledby="pills-fulltime-tab">
					<form action="request.php" method="POST">

						<div class="form-group">
							<label for="initials"><strong>Faculty
Initials:</strong></label>
							<input type="text" class="form-control" name="initials"
placeholder="Enter Initials" required>
						</div>

						<button type="submit" class="btn btn-primary"
name="timetable">Get Info</button>
```

```
                                        </form>
                                </div>

                                <div class="tab-pane fade" id="pills-freeslots" role="tabpanel"
aria-labelledby="pills-freeslots-tab">

                                        <form action="request.php" method="POST">

                                                <div class="form-group">
                                                        <label for="initials"><strong>Faculty
Initials:</strong></label>
                                                        <input type="text" class="form-control" name="initials"
placeholder="Enter Initials" required>
                                                </div>

                                                <button type="submit" class="btn btn-primary"
name="free_slots">Get Info</button>

                                        </form>

                                </div>

                                <div class="tab-pane fade" id="pills-vacancy" role="tabpanel"
aria-labelledby="pills-vacancy-tab">

                                        <form action="request.php" method="POST">

                                                <fieldset class="form-group">
                                                <div class="row">
                                                        <legend class="col-form-label col-sm-2
pt-0"><strong>Periods:</strong></legend>
                                                        <div class="col-sm-10">
                                                                <div class="form-check">
                                                                  <input class="form-check-input" type="radio"
name="gridRadios" id="gridRadios1" value="1" checked>
                                                                  <label class="form-check-label"
for="gridRadios1">
                                                                    08:15 - 09:15
                                                                  </label>
                                                                </div>
                                                                <div class="form-check">
                                                                  <input class="form-check-input" type="radio"
name="gridRadios" id="gridRadios2" value="2">
                                                                  <label class="form-check-label"
for="gridRadios2">
                                                                    09:15 - 10:15
                                                                  </label>
                                                                </div>
                                                                <div class="form-check">
                                                                  <input class="form-check-input" type="radio"
name="gridRadios" id="gridRadios3" value="3">
                                                                  <label class="form-check-label"
for="gridRadios3">
```

```
                                                                    10:45 - 11:45
                                                                  </label>
                                                                </div>
                                                                        <div class="form-check">
                                                                  <input class="form-check-input" type="radio"
name="gridRadios" id="gridRadios4" value="4">
                                                                  <label class="form-check-label"
for="gridRadios4">
                                                                    11:45 - 12:45
                                                                  </label>
                                                                </div>
                                                                        <div class="form-check">
                                                                  <input class="form-check-input" type="radio"
name="gridRadios" id="gridRadios5" value="5">
                                                                  <label class="form-check-label"
for="gridRadios5">
                                                                    13:30 - 14:30
                                                                  </label>
                                                                </div>
                                                                        <div class="form-check">
                                                                  <input class="form-check-input" type="radio"
name="gridRadios" id="gridRadios6" value="6">
                                                                  <label class="form-check-label"
for="gridRadios6">
                                                                    14:30 - 15:30
                                                                  </label>
                                                                </div>
                                                        </div>
                                                </div>
                                                </fieldset>

                                                <div class="form-group">
                                                        <label for="date"><strong>Date:</strong></label>
                                                        <input type="date" class="form-control" name="date"
id="date" placeholder="Select the Date" required>
                                                </div>

                                                <button type="submit" class="btn btn-primary"
name="class_vacancy">Get Info</button>

                                        </form>

                                </div>

                                <div class="tab-pane fade" id="pills-findlec" role="tabpanel"
aria-labelledby="pills-findlec-tab">

                                        <form action="request.php" method="POST">

                                                <fieldset class="form-group">
                                                <div class="row">
                                                        <legend class="col-form-label col-sm-2
pt-0"><strong>Periods:</strong></legend>
                                                        <div class="col-sm-10">
```

```
                                                  <div class="form-check">
                                                    <input class="form-check-input" type="radio" 
name="gridRadios" id="gridRadios1" value="1" checked>
                                                    <label class="form-check-label" 
for="gridRadios1">
                                                      08:15 - 09:15
                                                    </label>
                                                  </div>
                                                  <div class="form-check">
                                                    <input class="form-check-input" type="radio" 
name="gridRadios" id="gridRadios2" value="2">
                                                    <label class="form-check-label" 
for="gridRadios2">
                                                      09:15 - 10:15
                                                    </label>
                                                  </div>
                                                  <div class="form-check">
                                                    <input class="form-check-input" type="radio" 
name="gridRadios" id="gridRadios3" value="3">
                                                    <label class="form-check-label" 
for="gridRadios3">
                                                      10:45 - 11:45
                                                    </label>
                                                  </div>
                                                        <div class="form-check">
                                                    <input class="form-check-input" type="radio" 
name="gridRadios" id="gridRadios4" value="4">
                                                    <label class="form-check-label" 
for="gridRadios4">
                                                      11:45 - 12:45
                                                    </label>
                                                  </div>
                                                        <div class="form-check">
                                                    <input class="form-check-input" type="radio" 
name="gridRadios" id="gridRadios5" value="5">
                                                    <label class="form-check-label" 
for="gridRadios5">
                                                      13:30 - 14:30
                                                    </label>
                                                  </div>
                                                        <div class="form-check">
                                                    <input class="form-check-input" type="radio" 
name="gridRadios" id="gridRadios6" value="6">
                                                    <label class="form-check-label" 
for="gridRadios6">
                                                      14:30 - 15:30
                                                    </label>
                                                  </div>
                                            </div>
                                      </div>
                                      </fieldset>

                                      <div class="form-group">
                                            <label for="date"><strong>Date:</strong></label>
```

```
                                                                <input type="date" class="form-control" name="date"
id="date" placeholder="Select the Date" required>
                                                        </div>

                                                        <div class="form-group">
                                                                <label for="initials"><strong>Faculty
Initials:</strong></label>
                                                                <input type="text" class="form-control" name="initials"
placeholder="Enter Initials" required>
                                                        </div>

                                                        <button type="submit" class="btn btn-primary"
name="find_lecturer">Get Info</button>

                                                </form>

                                        </div>


                                </div>
                        </div>

                        <footer class="footer fixed-bottom">
                        <p><center>
                        Copyright &COPY; ISE @JSSATEB. All Rights Reserved
                        <br />
                        Made with <i class="fa fa-heart" style="color:red"></i> in India
                        </center></p>
                        </footer>

                        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
</body>
</html>
```

4.3. Stored Procedure and Triggers

4.3.1. Stored Procedure

A Stored Procedure is a set of SQL statements with an assigned name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs.

4.3.2. Triggers

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. It keeps the database active by performing actions on events like insertion or updating of values in specific tables.

Chapter 5:
TESTING

5.1. Introduction

Testing plays a vital role in the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. Once program code has been developed, testing begins. The testing process focuses on the logical internals of the software, ensuring that all statements have been tested, and on the functional externals, that is conducted tests to uncover errors and ensure that defined input will produce actual results that agree with required results. Broadly speaking, there are at least three levels of testing:

- Unit testing
- Integration testing
- System testing

5.2. Unit Testing

Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to ensure that the building blocks of the software work independently from each other. Unit testing is a software development process that involves a synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the software development lifecycle. Unit testing aims to eliminate construction errors before code is promoted to additional testing; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development process.

5.3. Integration Testing

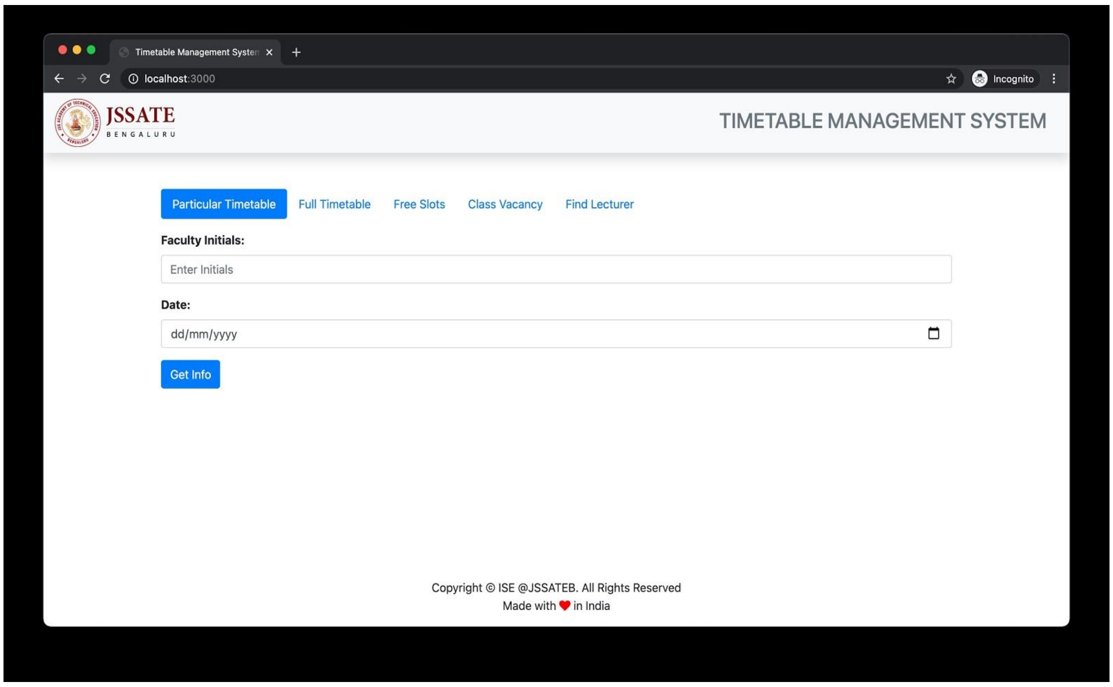
Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed. Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

5.4. System Testing

System testing tests a completely integrated system to verify that the system meets its requirements. For example, a system test might involve testing a login interface, then creating and editing an entry, plus sending or printing results, followed by summary processing or deletion (or archiving) of entries, then log off.

Chapter 6:
RESULTS

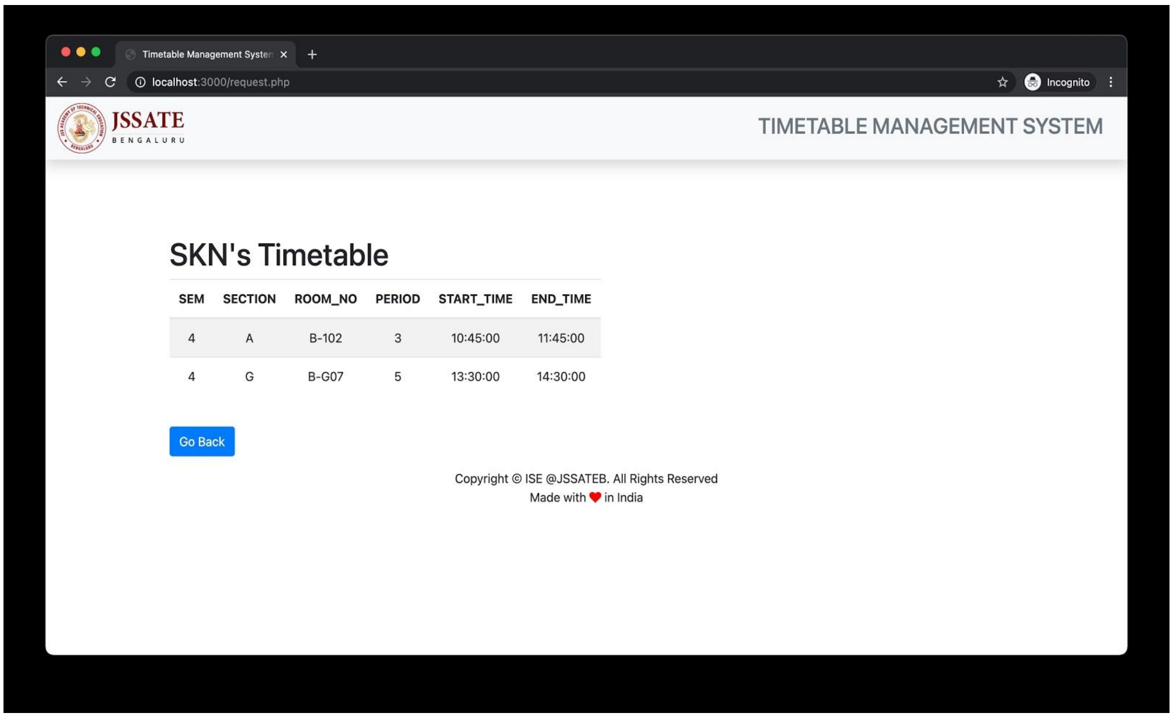
6.1. Snapshots



Faculty Initials: Enter Initials

Date: dd/mm/yyyy

Copyright © ISE @JSSATEB. All Rights Reserved
Made with ❤ in India



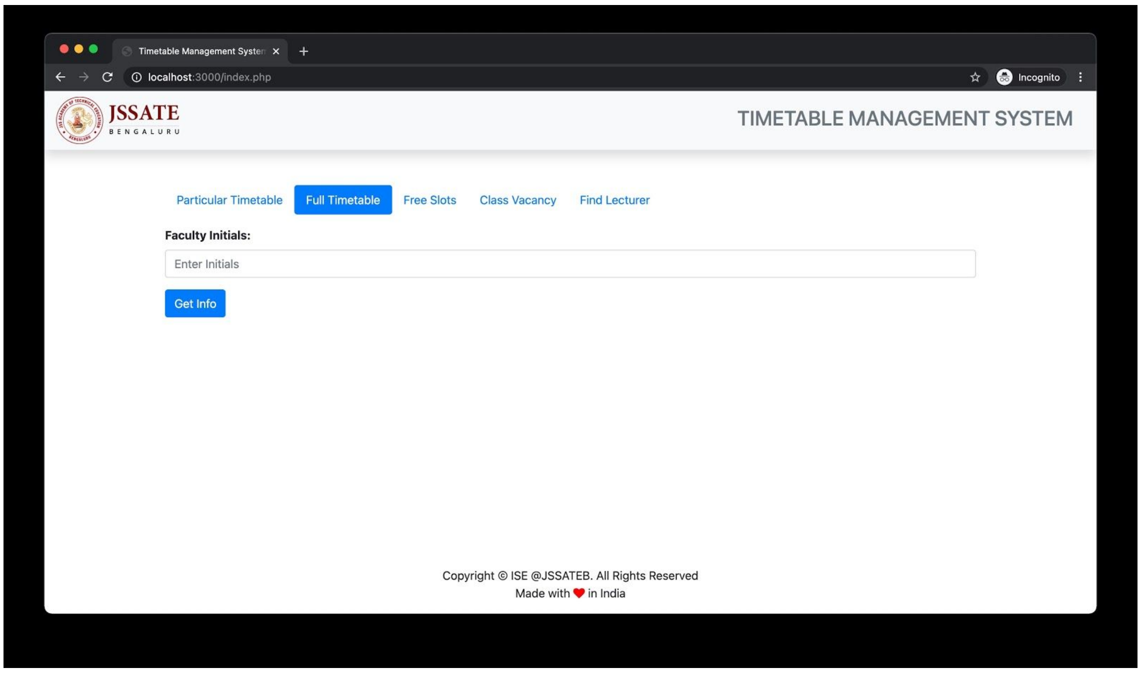
Timetable Management System
localhost:3000/request.php
Incognito
JSSATE BENGALURU
TIMETABLE MANAGEMENT SYSTEM
Go Back

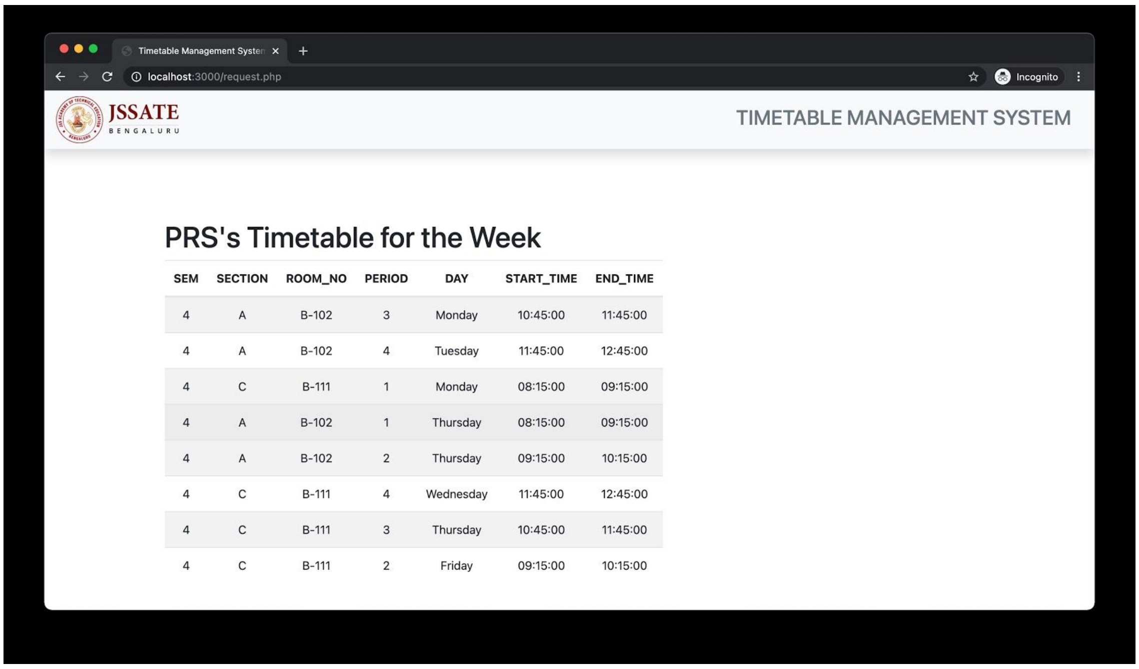
SKN's Timetable

SEM	SECTION	ROOM_NO	PERIOD	START_TIME	END_TIME
4	A	B-102	3	10:45:00	11:45:00
4	G	B-G07	5	13:30:00	14:30:00

Copyright © ISE @JSSATEB. All Rights Reserved
Made with ❤ in India

Particular Timetable



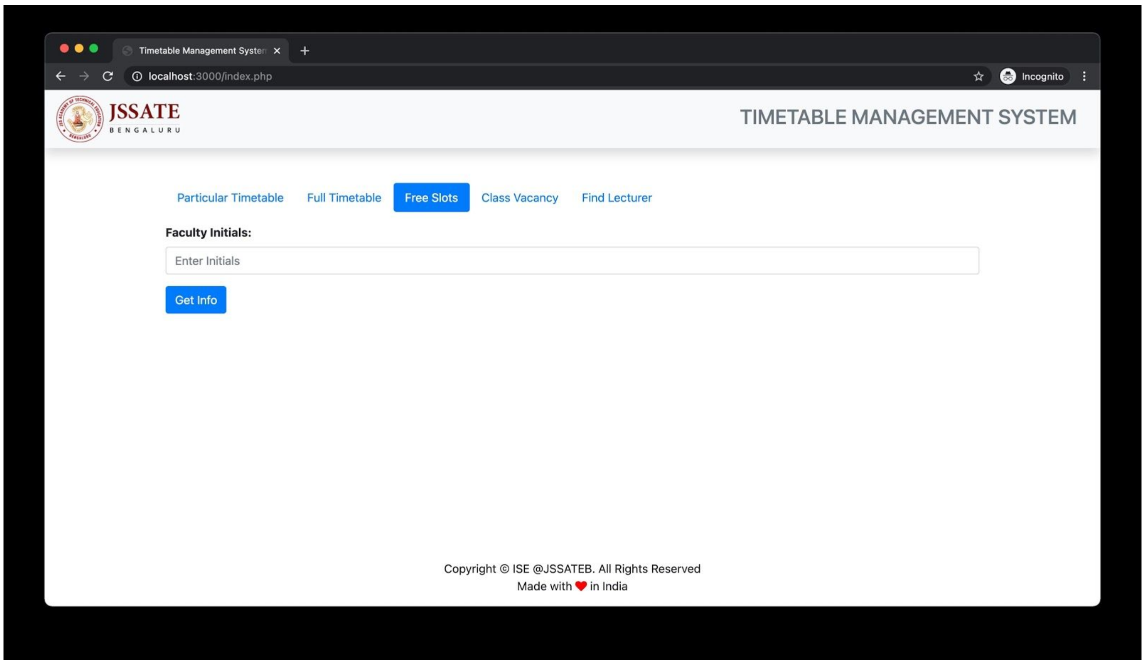


Timetable Management System
localhost:3000/request.php
Incognito
JSSATE
BENGALURU
TIMETABLE MANAGEMENT SYSTEM

PRS's Timetable for the Week

SEM	SECTION	ROOM_NO	PERIOD	DAY	START_TIME	END_TIME
4	A	B-102	3	Monday	10:45:00	11:45:00
4	A	B-102	4	Tuesday	11:45:00	12:45:00
4	C	B-111	1	Monday	08:15:00	09:15:00
4	A	B-102	1	Thursday	08:15:00	09:15:00
4	A	B-102	2	Thursday	09:15:00	10:15:00
4	C	B-111	4	Wednesday	11:45:00	12:45:00
4	C	B-111	3	Thursday	10:45:00	11:45:00
4	C	B-111	2	Friday	09:15:00	10:15:00

<u>Full Timetable</u>


Timetable Management System

localhost:3000/index.php

JSSATE BENGALURU

TIMETABLE MANAGEMENT SYSTEM

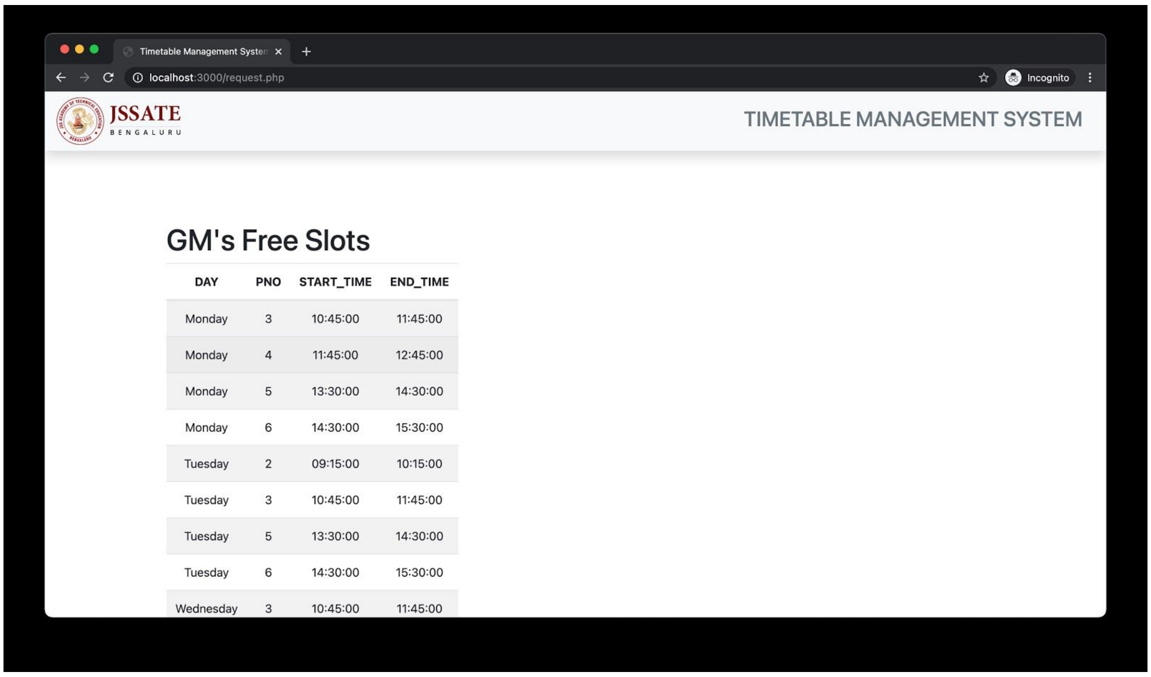
Particular Timetable | Full Timetable | Free Slots | Class Vacancy | Find Lecturer

Faculty Initials:

Enter Initials

Get Info

Copyright © ISE @JSSATEB. All Rights Reserved
Made with ❤ in India


Timetable Management System

localhost:3000/request.php

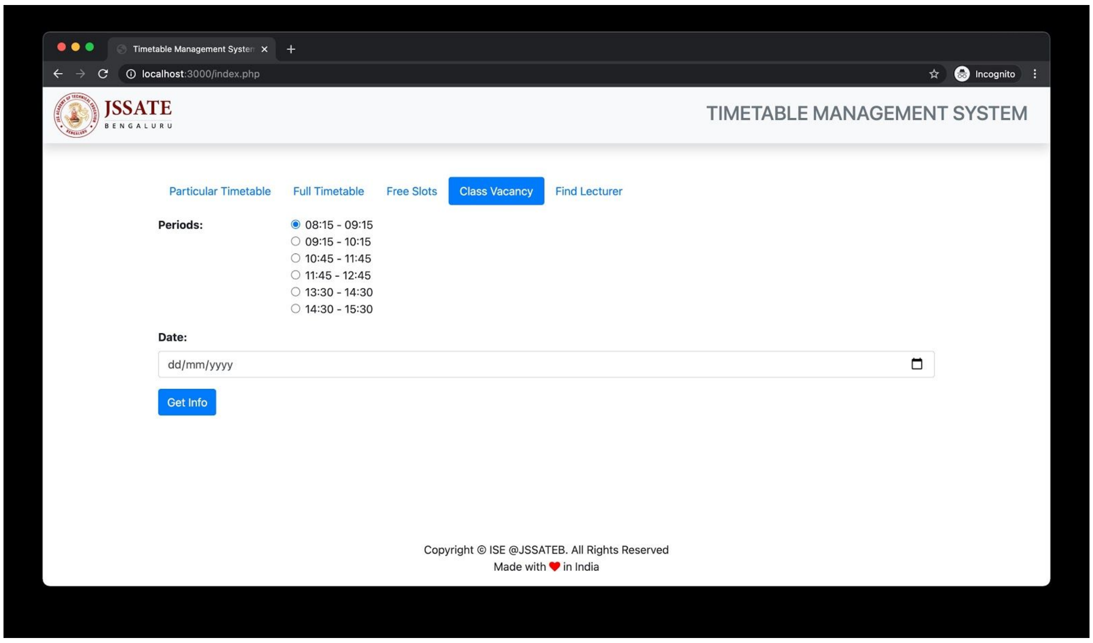
JSSATE BENGALURU

TIMETABLE MANAGEMENT SYSTEM

GM's Free Slots

DAY	PNO	START_TIME	END_TIME
Monday	3	10:45:00	11:45:00
Monday	4	11:45:00	12:45:00
Monday	5	13:30:00	14:30:00
Monday	6	14:30:00	15:30:00
Tuesday	2	09:15:00	10:15:00
Tuesday	3	10:45:00	11:45:00
Tuesday	5	13:30:00	14:30:00
Tuesday	6	14:30:00	15:30:00
Wednesday	3	10:45:00	11:45:00

Free Slots


localhost:3000/index.php

JSSATE BENGALURU

TIMETABLE MANAGEMENT SYSTEM

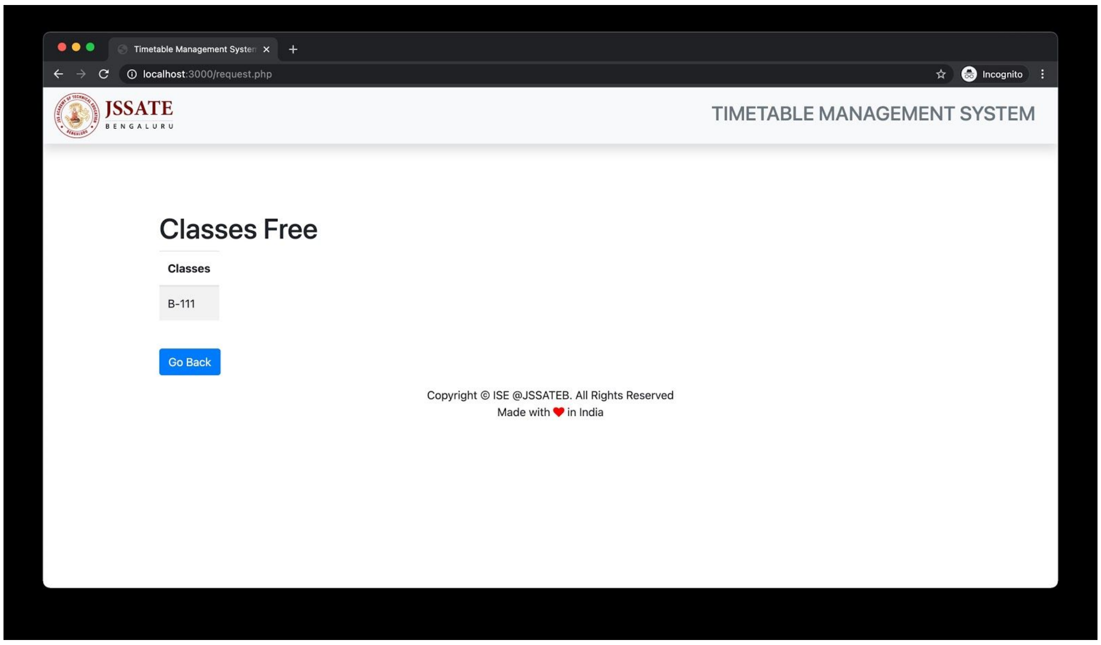
Particular Timetable | Full Timetable | Free Slots | Class Vacancy | Find Lecturer

Periods:
- (●) 08:15 - 09:15
- ( ) 09:15 - 10:15
- ( ) 10:45 - 11:45
- ( ) 11:45 - 12:45
- ( ) 13:30 - 14:30
- ( ) 14:30 - 15:30

Date: dd/mm/yyyy

Get Info

Copyright © ISE @JSSATEB. All Rights Reserved
Made with ♥ in India


localhost:3000/request.php

JSSATE BENGALURU

TIMETABLE MANAGEMENT SYSTEM

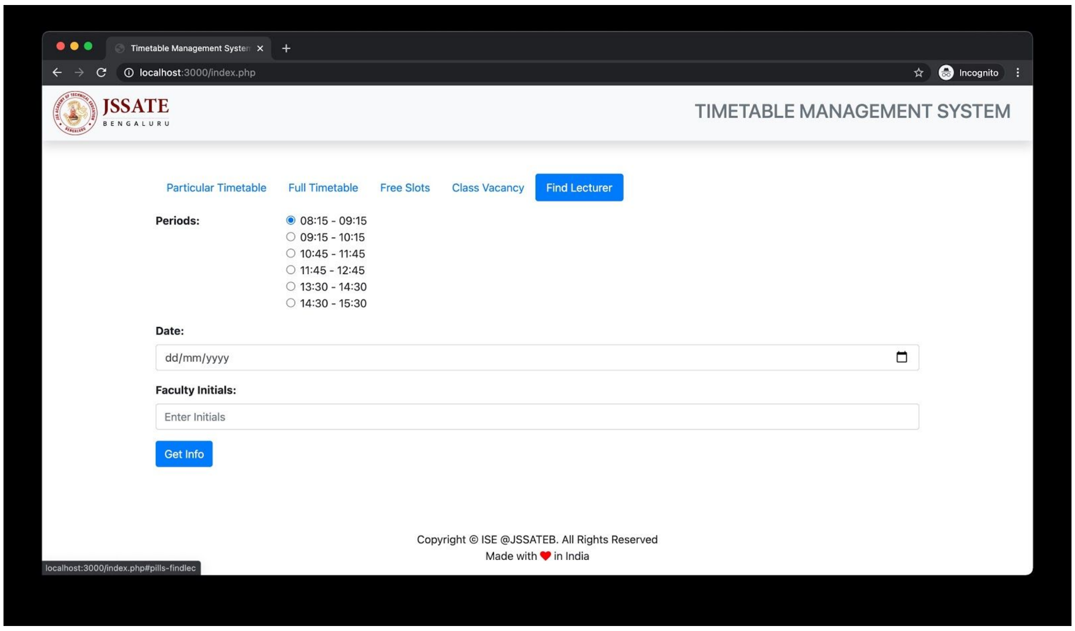
Classes Free

Classes
B-111

Go Back

Copyright © ISE @JSSATEB. All Rights Reserved
Made with ♥ in India

Class Vacancy



Timetable Management System

localhost:3000/index.php

Incognito

JSSATE BENGALURU

TIMETABLE MANAGEMENT SYSTEM

Particular Timetable | Full Timetable | Free Slots | Class Vacancy | Find Lecturer

Periods:
- [✓ 08:15 - 09:15]
- 09:15 - 10:15
- 10:45 - 11:45
- 11:45 - 12:45
- 13:30 - 14:30
- 14:30 - 15:30

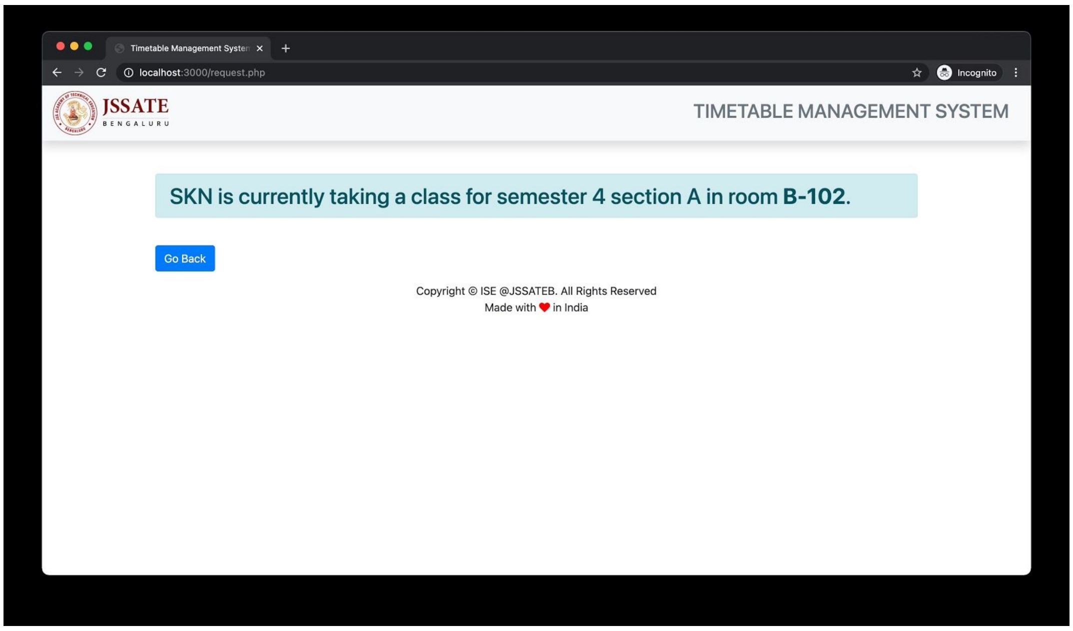
Date: dd/mm/yyyy

Faculty Initials: Enter Initials

Get Info

Copyright © ISE @JSSATEB. All Rights Reserved
Made with ♥ in India

localhost:3000/index.php#pills-findlec



Timetable Management System

localhost:3000/request.php

Incognito

JSSATE BENGALURU

TIMETABLE MANAGEMENT SYSTEM

SKN is currently taking a class for semester 4 section A in room **B-102**.

Go Back

Copyright © ISE @JSSATEB. All Rights Reserved
Made with ♥ in India

<u>Find Lecturer</u>

Chapter 7:
CONCLUSION AND FUTURE ENHANCEMENTS

7.1. Conclusion

The package was designed in such a way that future modifications can be done easily. The following conclusion can be deduced from the development of the project.

- It provides a friendly graphical user interface which proves to be better when compared to other existing systems.
- It gives appropriate access to the authorized users depending on their permissions
- Updating information becomes easier.
- System security, data security and reliability are the striking features.
- The system has adequate scope for modification in future if it's necessary.

7.2. Future Enhancements

The application can be enhanced with more features. Some of them are listed below.

- Develop an iOS and/or an Android application for the same.
- Databases can be migrated to a NoSQL one like MongoDB, Redis or Cassandra for large data flexibility.
- The backend can be migrated to a more efficient system like Node.js or Golang that allows more concurrency by handling thousands of requests per second with ease. This also allows the backend to be scalable horizontally too.
- The frontend can be migrated to a more sophisticated and modular component based application by using JavaScript libraries such as React.js or frameworks like Angular and Vue.js.

Chapter 8:
REFERENCES

8.1. Books

1. Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, McGRAW HILL, 3rd Edition.
2. Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7th Edition.

8.2. Websites

1. https://www.postgresql.org/docs/
2. https://www.tutorialspoint.com/postgresql/index.htm
3. https://getbootstrap.com/docs/5.0/getting-started/introduction/
4. https://www.w3schools.com/php/
5. https://www.tutorialspoint.com/php/
6. https://www.tutorialspoint.com/postgresql/postgresql_php.htm
7. https://www.php.net/manual/en/book.pgsql.php