VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018

A DBMS Lab Mini Project Report On

"E-COMMERCE"

SUBMITTED IN PARTIAL FULFILLMENT FOR 5TH SEMESTER

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

Monik L (1JB20CS064)

Pavan S G (1JB20CS078)

Under the Guidance of

Mrs. Anusha M
Assistant Professor
Dept. of CSE,
SJBIT, Bengaluru-560060

SJBIT

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SJB INSTITUTE OF TECHNOLOGY

No.67, BGS Health & Education City, Dr.Vishnuvardhan Rd, Kengeri,
Bengaluru, Karnataka 560060

2022 - 2023

|| Jai Sri Gurudev ||

Sri Adichunchanagiri Shikshana Trust ®

SJB INSTITUTE OF TECHNOLOGY

No.67, BGS Health & Education City, Dr.Vishnuvardhan Rd, Kengeri, Bengaluru, Karnataka 560060

Department of Computer Science and Engineering

SJBIT

CERTIFICATE

Certified that the Mini Project Work entitled "**E-COMMERCE**" carried out by Mr. **Monik L** & **Pavan S G** bearing USN **1JB20CS064** & **1JB20CS078** are bonafide students of **SJB Institute of Technology** in partial fulfilment for 5th semester of **BACHELOR OF ENGINEERING** in **Computer Science and Engineering** of the **Visvesvaraya Technological University**, **Belagavi during** the academic year **2022-23.** It is certified that all corrections/ suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the Departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work phase-1 prescribed for the said Degree.

Signature of Guide

Mrs. Anusha M
Assistant Professor
Dept. of CSE, SJBIT

Signature of HOD

Dr. Krishna A N
Professor & Head
Dept. of CSE, SJBIT

1. **Internal Examiner:** ______________________
2. **External Examiner:** ______________________

ACKNOWLEDGEMENT

We would like to express our profound grateful to His Divine Soul **Jagadguru Padmabhushan Sri Sri Sri Dr. Balagangadharanatha Mahaswamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Mahaswamiji** for providing us an opportunity to complete our academics in this esteemed institution.

We would also like to express our profound thanks to Revered **Sri Sri Dr. Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this Mini Project Work in this admired institution.

We express our gratitude to **Dr. K. V. Mahendra Prashanth**, Principal, SJB Institute of Technology, for providing us an excellent facilities and academic ambience; which have helped us in satisfactory completion of Mini Project Work.

We extend our sincere thanks to **Dr. Krishna A N**, Head of the Department, Computer Science and Engineering for providing us an invaluable support throughout the period of our Mini Project Work.

We wish to express our heartfelt gratitude to our guide **Mrs. Anusha M**, Assistant Professor, Dept. of CSE, for her valuable guidance, suggestions and cheerful encouragement during the entire period of our Mini Project Work.

Finally, we take this opportunity to extend our earnest gratitude and respect to our parents, Teaching & Non-teaching staffs of the department, the library staff and all our friends, who have directly or indirectly supported us during the period of our Mini Project Work.

Regards,
Monik L (1JB20CS064)
Pavan S G (1JB20CS078)

ABSTRACT

As this is generic application it can be used by a wide variety of users i.e, applicant, hostel Managers and admin. This software helps the applicant to keep track of all the applications along with the application number and the applied hostel's short details he/she has applied.

Through this software the applicant can apply applications to the hostels which are displayed on the homepage. To apply application the applicant has to login. Once the applicant submits the application successfully then the application number of newly applied application will be showed on the applicant's user account.

Hostel Manager(s) has been provided with an interface where they can check the newly arrived applications whose status is pending. Either they can approve the application or reject it. Even they can update the details of the hostels.

Once the status of application is changed from pending to accepted/rejected, the same will be notified in 'my applications' section of the respective user account where the admin can add/update/delete the details of hostels and hostel Managers.

Table of Contents

List of Tables and Figures

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 TRADITIONAL FILE SYSTEM

File System is collection of data. In this system, user has to write procedures for managing database. It provides details of data representation and storage of data. In this –

- Data is stored in files.
- Each file has specific format.
- Programs that use these files depend on knowledge about that format.
- In earlier days, database applications were built on top of file systems.
- Basically, it is a collection of application programs that performs services for end users such as production of reports. Each file defines and manages its own data.

1.1.1 PROS AND CONS OF TRADITIONAL APPROACH

PROS:

- File processing cost less and can be more speed than database.
- File processing design approach was well suited to mainframe hardware and batch input.
- Companies mainly use file processing to handle large volumes of structured data on a regular basis
- It can be more efficient and cost less than DBMS in certain situations.
- Design is simple.
- Customization is easy and efficient.

CONS:

- Data redundancy and inconsistency.
- Difficulty in accessing data.
- Data isolation – multiple files and formats.
- Integrity problems
- Unauthorized access is not restricted.
- It co-ordinates only physical access.

1.2-INTRODUCTION TO DBMS

Databases and database technology have had a major impact on the growing use of computers. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. Nowadays, this data is typically stored in mobile phones, which have their own simple database software. In other words, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents. A database can be of any size and complexity. For example, the list of names and addresses referred to earlier may consist of only a few hundred records, each with a simple structure. On the other hand, the computerized catalogue of a large library may contain half a million entries organized under different categories.

- A database has the following implicit properties:
- A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse. Changes to the mini world are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

A database management system (DBMS) is a computerized system that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.
Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalogue or dictionary; it is called meta-data.

Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world, and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

1.2.1 INDICATIVE AREAS FOR THE USE OF DBMS

Table 1.1 Indicative areas for the use of DBMS

SECTOR	USE OF DBMS
BANKING	For customer information, account activities, payments, deposits, loans, etc.
AIRLINES	For reservations and schedule information.
UNIVERSITIES	For student information, course registrations, colleges and grades.
TELECOMMUNICATION	It helps to keep call records, monthly bills, maintaining balances, etc.
FINANCE	For storing information about stock, sales, and purchases of financial instruments like stocks and bonds.
SALES	Use for storing customer, product & sales information.
MANUFACTURING	It is used for the management of supply chain and for tracking production of items. Inventories status in warehouses.
HR MANAGEMENT	For information about employees, salaries, payroll, deduction, generation of paychecks, etc.

Table 1.1 describes about the various areas of application of DBMS

1.2.2 ADVANTAGES OF DBMS

Compared to the File Based Data Management System, Database Management System has many advantages.

- **Reducing Data Redundancy**

The file based data management systems contained multiple files that were stored in many different locations in a system or even across multiple systems. Because of this, there were sometimes multiple copies of the same file which lead to data redundancy.

This is prevented in a database as there is a single database and any change in it is reflected immediately. Because of this, there is no chance of encountering duplicate data.

- **Data Integrity**

Data integrity means that the data is accurate and consistent in the database. Data Integrity is very important as there are multiple databases in a DBMS. All of these databases contain data that is visible to multiple users. So it is necessary to ensure that the data is correct and consistent in all the databases and for all the users.

- **Data Security**

  Data Security is vital concept in a database. Only authorized users should be allowed to access the database and their identity should be authenticated using a username and password. Unauthorized users should not be allowed to access the database under any circumstances as it violates the integrity constraints.

- **Privacy**

  The privacy rule in a database means only the authorized users can access a database according to its privacy constraints. There are levels of database access and a user can only view the data he is allowed to. For example - In social networking sites, access constraints are different for different accounts a user may want to access.

- **Backup and Recovery**

  Database Management System automatically takes care of backup and recovery. The users don't need to backup data periodically because this is taken care of by the DBMS. Moreover, it also restores the database after a crash or system failure to its previous condition.

- **Data Consistency**

  Data consistency is ensured in a database because there is no data redundancy. All data appears consistently across the database and the data is same for all the users viewing the database. Moreover, any changes made to the database are immediately reflected to all the users and there is no data inconsistency.

1.2.1 COMPONENTS OF DBMS

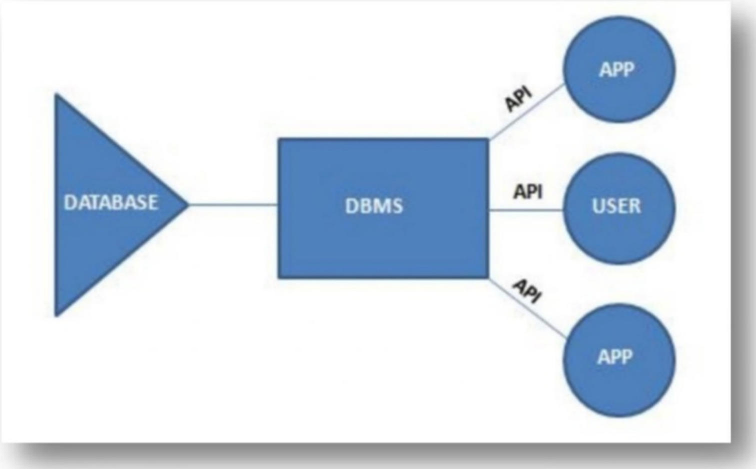


Fig 1.1 Components of DBMS

Figure 1.1 is a pictorial representation of various components of DBMS

- **Users**: Users may be of any kind, such as data base administrators, system developers or database users.
- **Database application**: Database application may be Departmental, Personal, Organizational and /or Internal.
- **DBMS**: Software that allows users to create and manipulate database access.
- **Database**: Collection of logical data as a single unit.

1.3-PROBLEM STATEMENT

The concept of 'Hostel Database of an Area' is useful for them who want to get a room in some hostel. There can be a lot of Hostels in an area with different features associated with the them. On the basis of these features, there are many candidates who want a place in those hostels. The whole objective of this project is to provide smooth way to access all the details of the hostels and applications, Hostel Managers for ease in processing application. This database helps in getting details of candidate as well as all the entities for user in easy and efficient manner.

1.4- OBJECTIVES

This software product the 'Hostel Database of an area' is to improve the services of requests of students to the hostel. This also reduce the manual work of the persons in admin penal and the bundle of registers that were search when to find the information of a application, because through this system you can store the data of the candidate that he has entered in the form. Through this the seeker can check the status of all the application that he has applied within few minutes the data base. The system will help hostel Managers to check newly applied requests and he can accept the request or even reject it. The candidate or hostel Manager will be recognized from the username.

- In the last, this system will improve the method of accepting the requests from interested candidates.
- To automate each and every activity of the manual system, which increases its throughput.
- To provide a quick response with very accurate information as and when required.
- To make the present manual system more interactive, speedy and user Friendly.
- To avail any information, whatever and whenever needed.
- Reduce the cost of maintenance.
- To reduce manual pen and paper involvement.
- To strive towards digitalization.

1.5- SCOPE

The scope of this project is clear to give a simple application to work as well as reduce the efforts of storing the database of application and the candidate.

Without this database system, it is a tedious task for any organization to maintain the record. This database will store all the details of the candidate including the application they applied for, hostels, hostel Managers.

Update/Delete: Suppose there are 100’s of hostels and 1000’s of applications and we need to process every request and have to inform candidate about the status of the application. In manual system this is a tedious task though we know the name of particular hostel and their facilities.

What contribution would the project make?

This is an era of information technology where automation is gaining importance. This system is much better than the manual approach of storing data and fetching it when needed. It allows neat handling of data rather than error prone records.

CHAPTER 2

SYSTEM REQUIREMENTS

There are no systems which can run without hardware and software requirements. So for any system in this world, the hardware and software requirements are the most basic necessity to work. For each and every system there will be different hardware and software requirements. So we shall see the particular requirement of our system. Software requirements concerned with portraying programming asset prerequisites and essentials that should be introduced on a computer to give best working of an application.

2.1- HARDWARE REQUIREMENTS

- Processor: Pentium core 2 duo and higher
- RAM: 512MB
- Hard Disk: 40GB
- Monitor: 15” Colour Monitor

2.2- SOFTWARE REQUIREMENTS

- OS: Windows 7 and higher
- Database: MYSQL
- Languages: PHP, HTML, CSS, JAVASCRIPT
- Software: XAMPP (any browser which supports XAMPP)

CHAPTER 3

DESIGN

3.1 Requirements and Constraints

Functional Requirements:

1.DISTRIBUTED DATABASE:

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network.

2.CLIENT/SERVER SYSTEM

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

3.USER INTERFACES

- Front-end software: HTML, CSS, Bootstrap
- Back-end software: PHP, myadmin SQL

4.HARDWARE INTERFACES

- Windows.
- A browser which supports PHP files

5.SOFTWARE INTERFACES

Table 3.1 Software Interfaces

Software used	**Description**
Operating system	We have chosen Windows operating system for its best support and user-friendliness.
Database	To save the customer records, tourist places records we have chosen PHP Myadmin SQL database.
HTML,CSS, Bootstrap	To implement the front-end of project we have chosen HTML, CSS, Bootstrap its more interactive support.
PHP	To implement back-end and to connect database we have used PHP as it is easy to

Table 3.1 Describes the various software interfaces used.

6-COMMUNICATION INTERFACES

This project supports all types of web browsers. We are using simple electronic forms for the reservation forms, ticket booking etc.

7-SECURITY REQUIREMENTS

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

CONSTRAINTS

Mainly Constraints on the relational database are of 4 types:

- Domain constraints
- Key constraints
- Entity Integrity constraints
- Referential integrity constraints

Domain constraints:

- Every domain must contain atomic values (smallest indivisible units) it means composite and multi-valued attributes are not allowed.
- We perform datatype check here, which means when we assign a data type to a column we limit the values that it can contain. Eg. If we assign the datatype of attribute age as int, we can’t give it values other than int datatype.

Key Constraints or Uniqueness Constraints:

- These are called uniqueness constraints since it ensures that every tuple in the relation should be unique.
- A relation can have multiple keys or candidate keys(minimal superkey), out of which we choose one of the keys as primary key, we don’t have any restriction on choosing the primary key out of candidate keys, but it is suggested to go with the candidate key with less number of attributes.
- Null values are not allowed in the primary key, hence Not Null constraint is also a part of key constraint.

Entity Integrity Constraints:

- Entity Integrity constraints says that no primary key can take NULL value, since using primary key we identify each tuple uniquely in a relation.

Referential Integrity Constraints:

- The Referential integrity constraints is specified between two relations or tables and used to maintain the consistency among the tuples in two relations.
- This constraint is enforced through foreign key, when an attribute in the foreign key of relation R1 have the same domain(s) as the primary key of relation R2, then the foreign key of R1 is said to reference or refer to the primary key of relation R2.
- The values of the foreign key in a tuple of relation R1 can either take the values of the primary key for some tuple in relation R2, or can take NULL values, but can't be empty.

3.2 Entities and Attribute

ADMIN

- Name
- userid
- password
- phno
- email

HOSTEL

- Name
- code
- location
- capacity
- vacancy
- Warden
- phno
- email
- type
- image

MANAGER

- name
- worksin
- hostelcode
- position

- salary
- address
- phno
- email
- userid
- Password

USERS

- name
- Phno
- email
- userid
- password
- gender

APPLICATION

- app_no
- fname
- mname
- lname
- dob
- Hostel
- hostelcode
- status
- userid
- Datetime

GUARDIAN_DETAILS

- app_no
- father
- mother
- gaurdian
- f_phno
- alt_phno
- email
- Address

3.3 ER Diagram

An **Entity–relationship model** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram** . An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

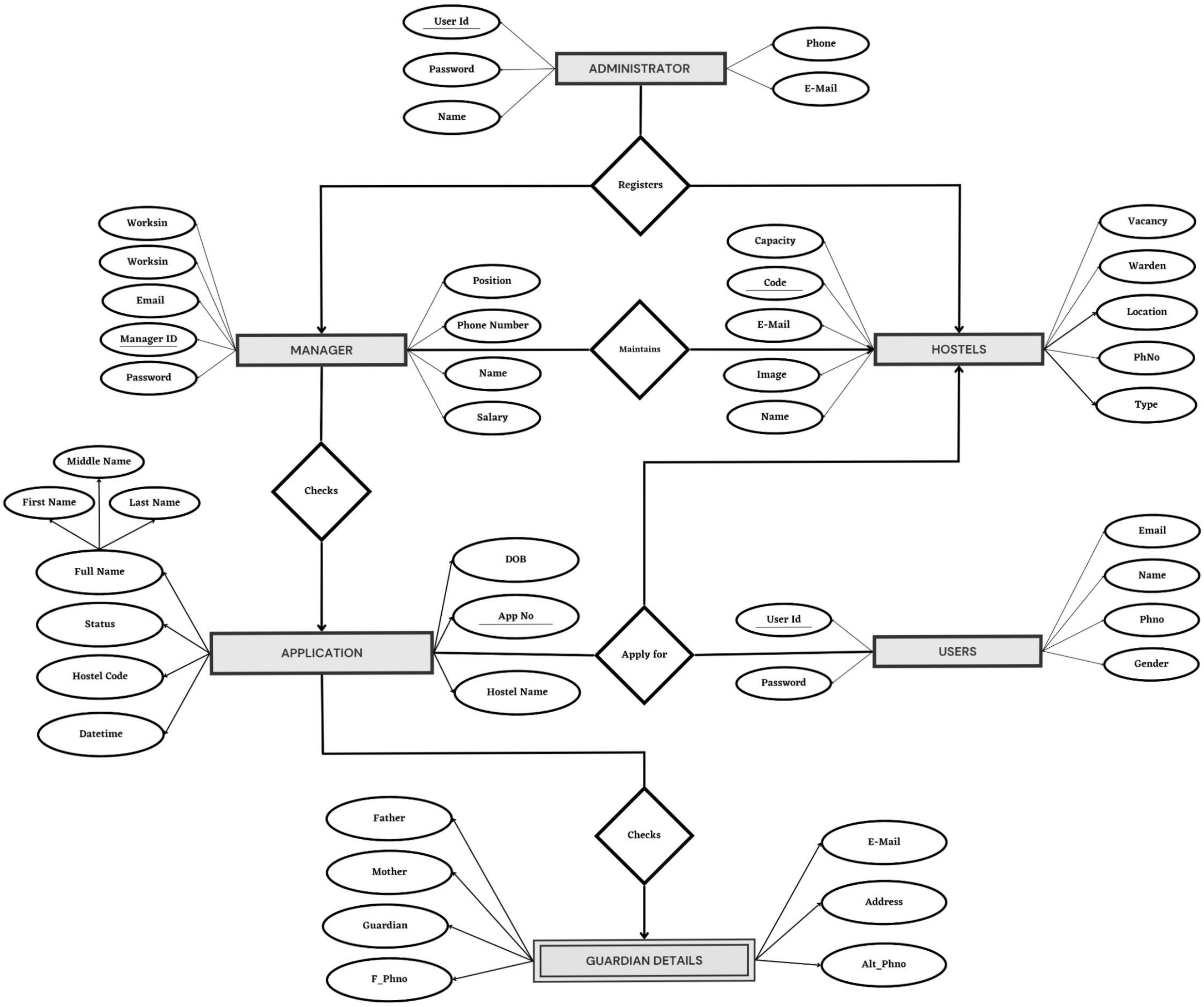


Fig 3.1 - ER Diagram of the relational database

Fig 3.1 represents the ER Diagram of the relational database that helps in understanding relationships between the entities and how they are related to each other.

Relationship table

Entity 1	Relationship	Entity 2
ADMIN	Modifies	HOSTEL
ADMIN	Modifies	MANAGERS
MANAGER	Supervises	HOSTEL
USERS	Apply	APPLICATION
USERS	Applies for	HOSTEL
MANAGER	Checks	APPLICATION
APPLICATION	Contains	GAURDIAN_DETAILS

Table 3.2 Relationship between entities

3.4 Relational Schema

Mapping Process:

Step 1: Convert all strong entity sets into relations

Admin
[Name,userid,password,phno,email]

Hostel
[Name,code,location,capacity,vacancy,warden,phno,email,type,image]

Manager
[name,worksin,position,salary,address,phno,email,userid,password]

Users
[name,phno,email,userid,password,gender]

Application
[app_no,fname,mname,lname,dob,Hostel,status,Datetime]

Step 2: Mapping weak entity

Create table for weak entity set and add all its attributes to table as field. Then Add the primary key of identifying entity set. Declare all foreign key constraint.

Gaurdian_details
[app_no,father,mother,gaurdian,f_phno,alt_phno,email,address]

Step 3: Mapping 1:1 relation

Add the primary key of partial participating entity to complete participating side. Add the primary key of application relation to Gaurdian_details relation as foreign key.

Application
[app_no,fname,mname,lname,dob,Hostel,status,Datetime]

Gaurdian_details
[app_no,father,mother,gaurdian,f_phno,alt_phno,email,address]

Step 4: Mapping 1: N or N:1 relation.

Consider the relation of N side, add primary key of that entity type as foreign key to '1' cardinality side.

Manager
[name,worksin,position,salary,address,phno,email,userid,password,hostelcode]

Application
[app_no,fname,mname,lname,dob,Hostel,status,Datetime,userid]

Step 5: Mapping M: N relation.

Create a new relation and primary key of both entity is considered as foreign key relation. Also add all descriptive attributes to it.

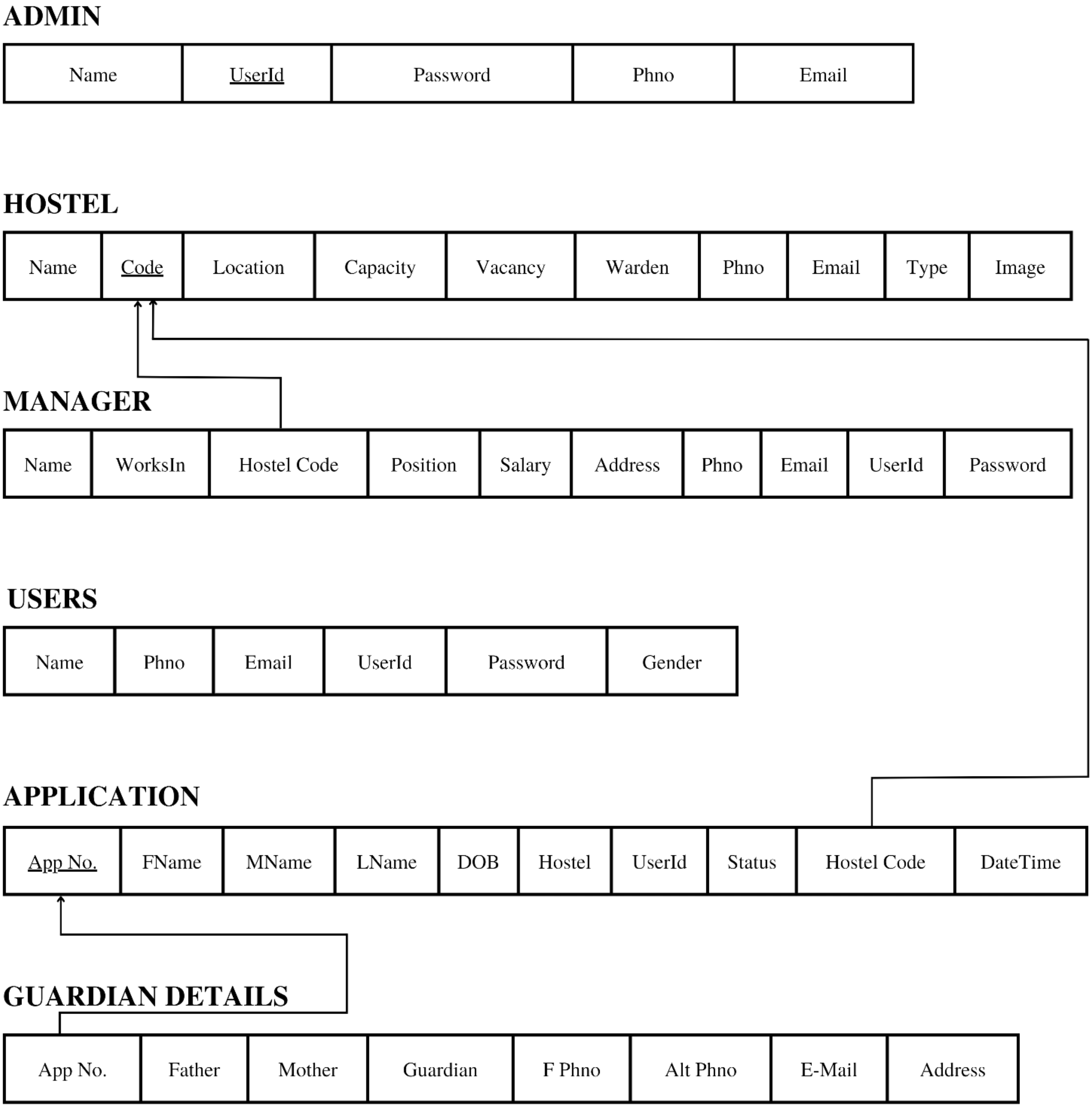
3.5 Normalization

Normalizing tables accordingly based on the normal forms.

Normal forms: Of a relation refers to the highest normal form condition that it meets and hence indicates the degree to which it has been normalized.

- **First NF:** States that the domain of the attribute must only include atomic values and the value of any attribute in a tuple must be a single value.
- **Second NF:** Is based on the concept of full functional dependency I.e. if removal of any attribute A from X in FD X->Y the dependency does not hold anymore.
- **Third NF:** A relation schema R is in 3NF if, whenever a non-trivial functional dependency X->A holds in R either: X is a super key of R or A is prime attribute of R.
- **Boyce-Codd NF:** A relation schema R is in BCNF if whenever a non-trivial functional dependency X->A holds in R, then X is a super key of R.
- **Fourth NF:** A relation schema R is in 4NF w.r.t a set of dependencies F if, for every non-trivial multivalued dependency X->>Y in F^+,X is a super key for R.

3.6 Schema Diagram



ADMIN

Name	UserId	Password	Phno	Email

HOSTEL

Name	Code	Location	Capacity	Vacancy	Warden	Phno	Email	Type	Image

MANAGER

Name	WorksIn	Hostel Code	Position	Salary	Address	Phno	Email	UserId	Password

USERS

Name	Phno	Email	UserId	Password	Gender

APPLICATION

App No.	FName	MName	LName	DOB	Hostel	UserId	Status	Hostel Code	DateTime

GUARDIAN DETAILS

App No.	Father	Mother	Guardian	F Phno	Alt Phno	E-Mail	Address

Fig 3.2 – SCHEMA DIAGRAM

Figure 3.2 Represents the Schema of the relational database, that helps in knowing the different entities and attributes in them, the way they are relate

CHAPTER 4

IMPLEMENTATION

4.1 Technologies/ Frameworks used in building the project

The language utilized for the implementation of project should be independent of platform, powerful and simple, object oriented, robust, powerful, interactive, simple to understand and learn, dynamic and extensible. The whole Project is divided in two parts the front end and the back end. The front end is designed using of PHP, CSS, HTML. The back end is designed using myself which is used to design the databases.

4.1.1 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Ramses Leadoff in 1995, the reference implementation of PHP is now produced by The PHP Group. While PHP originally stood for *Personal Home Page*, it now stands for *PHP: Hypertext Pre-processor*, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications. PHP is free software released under the PHP License. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

4.1.2 CSS

CSS Stands for "Cascading Style Sheet." Cascading style sheets are used to format the Layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

The basic purpose of CSS is to separate the content of a web document (written in any mark-up language) from its presentation (that is written using Cascading Style Sheets). There are lots of benefits that one can extract through CSS like improved content accessibility, better flexibility and moreover, CSS gives a level of control over various presentation characteristics of the document. It also helps in reducing the complexity and helps in saving overall presentation time.

4.1.3 Bootstrap

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

4.1.4 MYSQL

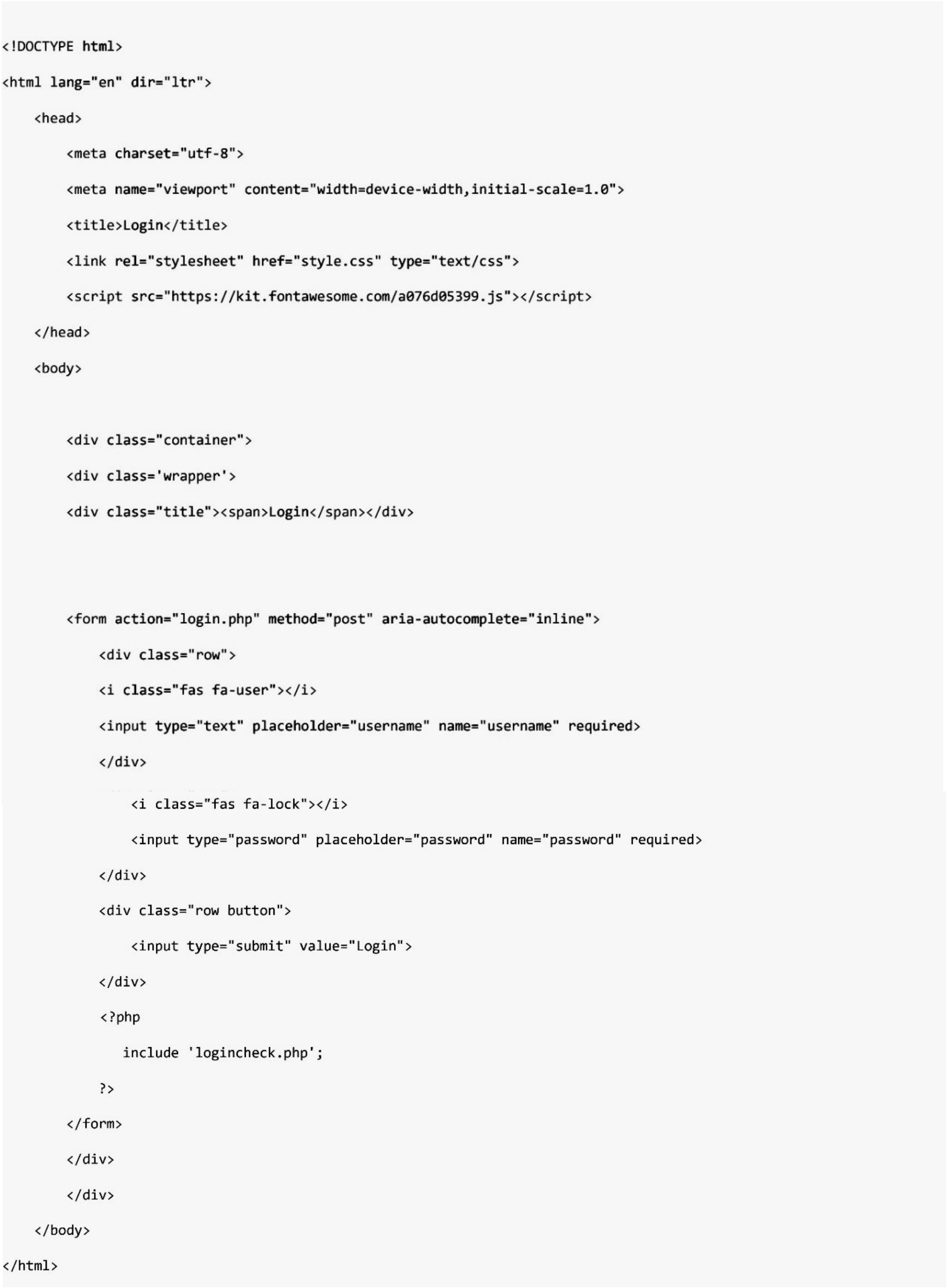
("My S-Q-L", officially, but also called "My Sequel") is (as of July 2013) the world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Wideners daughter; My SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. It was owned and sponsored by a single for-profit firm, the Swedish company Myself AB, now owned by Oracle Corporation. It was a popular choice of database for use in web applications, and am central components of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use me. For commercial use, several paid editions are available, and offer additional functionality.

4.1.5 HTML5

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard. It was published in October 2014 by the World Wide Web Consortium (W3C) to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc. HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML. HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind.

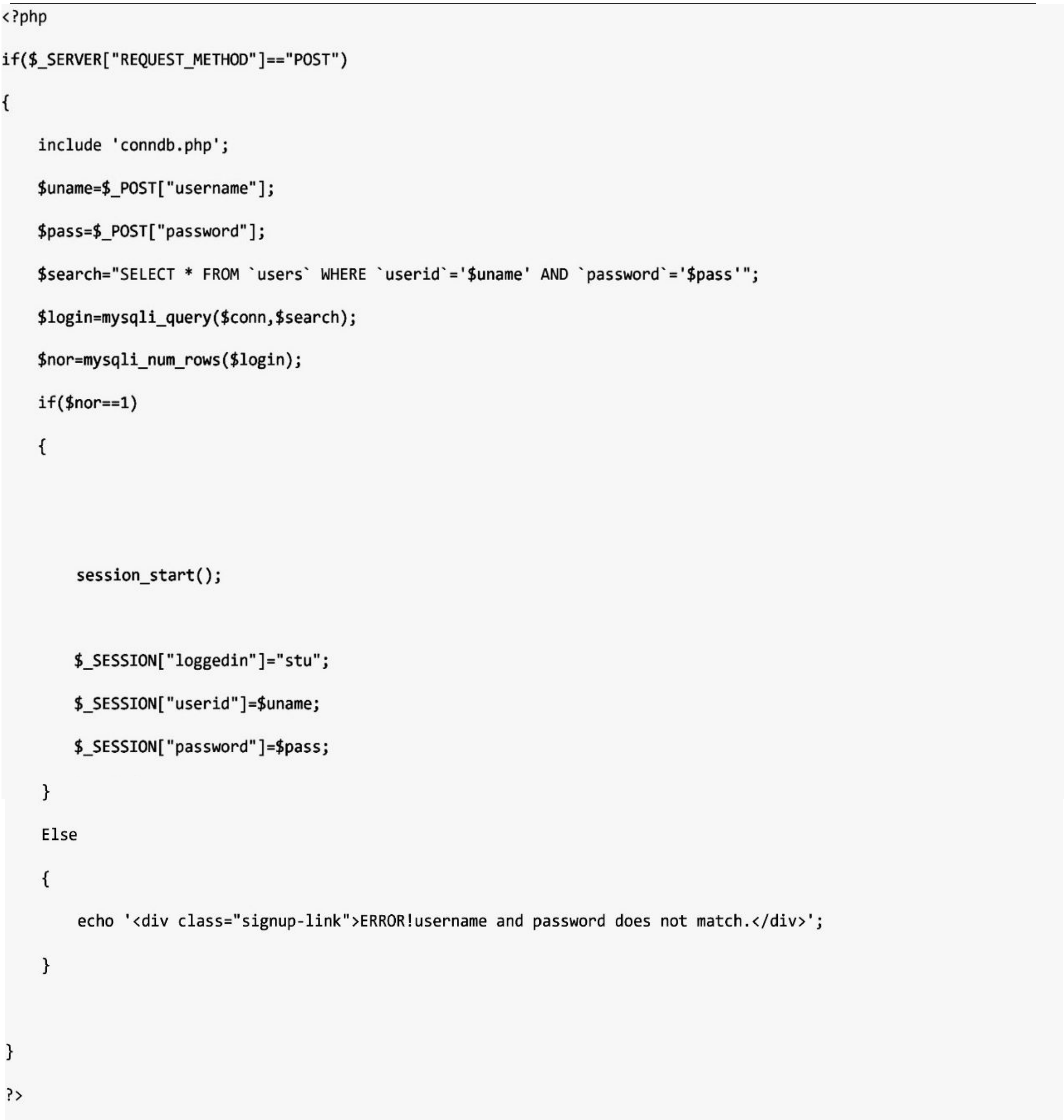
4.2 CODE SNIPPETS

4.2.1 Login page sample code



```
<!DOCTYPE html>
<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,initial-scale=1.0">
        <title>Login</title>
        <link rel="stylesheet" href="style.css" type="text/css">
        <script src="https://kit.fontawesome.com/a076d05399.js"></script>
    </head>
    <body>

        <div class="container">
        <div class='wrapper'>
        <div class="title"><span>Login</span></div>


        <form action="login.php" method="post" aria-autocomplete="inline">
            <div class="row">
            <i class="fas fa-user"></i>
            <input type="text" placeholder="username" name="username" required>
            </div>
                <i class="fas fa-lock"></i>
                <input type="password" placeholder="password" name="password" required>
            </div>
            <div class="row button">
                <input type="submit" value="Login">
            </div>
            <?php
               include 'logincheck.php';
            ?>
        </form>
        </div>
        </div>
    </body>
</html>
```

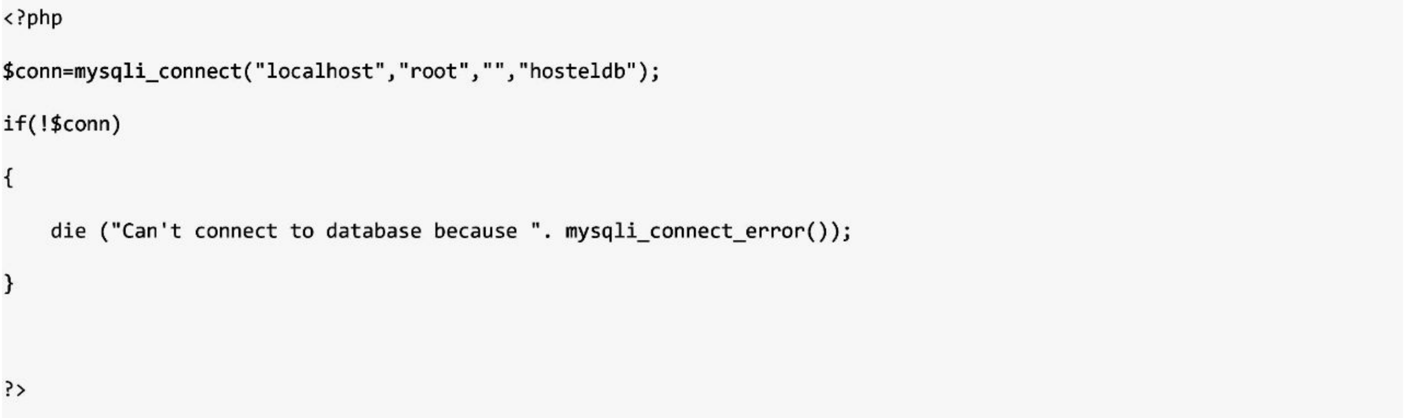
4.2.2 PHP code for login page



```
<?php
if($_SERVER["REQUEST_METHOD"]=="POST")
{
    include 'conndb.php';
    $uname=$_POST["username"];
    $pass=$_POST["password"];
    $search="SELECT * FROM `users` WHERE `userid`='$uname' AND `password`='$pass'";
    $login=mysqli_query($conn,$search);
    $nor=mysqli_num_rows($login);
    if($nor==1)
    {


        session_start();

        $_SESSION["loggedin"]="stu";
        $_SESSION["userid"]=$uname;
        $_SESSION["password"]=$pass;
    }
    Else
    {
        echo '<div class="signup-link">ERROR!username and password does not match.</div>';
    }

}
?>
```

This was the sample code used for the login page that checks the login credentials with the information stored in the database (refer figure no.). If the username and the password given by the user matches with the stored information, the user is logged in. Else the error message “Invalid Credentials” is displayed and the user is redirected to the same page.

4.2.3 Code for connecting database with the PHP



```
<?php
$conn=mysqli_connect("localhost","root","","hosteldb");
if(!$conn)
{
    die ("Can't connect to database because ". mysqli_connect_error());
}


?>
```

4.2.4 TRIGGER:

This method is implemented to add the warden as one of the Manager with his name as userid and password as 123456.

CREATE TRIGGER `defsuf`

AFTER INSERT ON `hostel`

FOR EACH ROW

INSERT INTO `supervisior` (`name`, `worksin`, `hostelcode`, `position`, `salary`, `address`, `phno`, `email`, `userid`, `password`) VALUES (new.warden, new.name, new.code, 'warden','30000',new.location,new.p hno,new.email,new.warden,'123456');

4.3 TABLE CREATION:

1. Creating table users:

CREATE TABLE `hosteldb`.`users` (

`name` VARCHAR(20) NOT NULL ,

`phno` INT(10) NOT NULL ,

`email` VARCHAR(30) NOT NULL ,

`userid` VARCHAR(20) NOT NULL ,

```
`password` VARCHAR(20) NOT NULL ,
`gender` VARCHAR(10) NOT NULL ,
 PRIMARY KEY (`userid`(20)))
 ENGINE = InnoDB;
```

2. Creating table admin:

```
 CREATE TABLE `hosteldb`.`admin` (
`Name` VARCHAR(20) NOT NULL ,
`userid` VARCHAR(20) NOT NULL ,
`password` VARCHAR(20) NOT NULL ,
`phno` BIGINT(10) NOT NULL ,
`email` VARCHAR(30) NOT NULL ,
 PRIMARY KEY  (`userid`(20)))
 ENGINE = InnoDB;
```

3. Creating table manager:

```
CREATE TABLE `hosteldb`.`supervisior`
( `name` VARCHAR(20) NOT NULL ,
`worksin` VARCHAR(30) NOT NULL ,
`hostelcode` VARCHAR(5) NOT NULL ,
`poition` VARCHAR(15) NOT NULL ,
`salary` FLOAT(20) NOT NULL ,
`address` VARCHAR(80) NOT NULL ,
`phno` BIGINT(10) NOT NULL ,
`email` VARCHAR(30) NOT NULL ,
`userid` VARCHAR(20) NOT NULL ,
`password` VARCHAR(20) NOT NULL ,
PRIMARY KEY (`userid`(20)))
ENGINE = InnoDB;
```

4. Creating table gaurdian_details:

```
  CREATE TABLE `hosteldb`.`gaurdian_details` (
`app_no` INT(11) NOT NULL ,
`father_name` VARCHAR(20) NOT NULL ,
`mother_name` VARCHAR(20) NOT NULL ,
`gaurdian_name` VARCHAR(20) NOT NULL ,
`f_phno` BIGINT(10) NOT NULL ,
`alt_phno` BIGINT(10) NOT NULL ,
`email` VARCHAR(30) NOT NULL ,
`address` VARCHAR(100) NOT NULL ,
PRIMARY KEY (`app_no`(11)))
ENGINE = InnoDB;
```

5. Creating table hostel:

```
 CREATE TABLE `hosteldb`.`hostel` (
`name` VARCHAR(50) NOT NULL ,
`code` VARCHAR(3) NOT NULL ,
`location` TEXT NOT NULL ,
`capacity` INT(5) NOT NULL ,
`vacancy` INT(5) NOT NULL ,
`warden` VARCHAR(20) NOT NULL ,
`phno` BIGINT(10) NOT NULL ,
`email` VARCHAR(30) NOT NULL ,
`type` TEXT NOT NULL ,
`image` BLOB NOT NULL ,
PRIMARY KEY  (`code`(3)))
 ENGINE = InnoDB;
```

6.Creating table application:

```
Create table ‘application’ (
‘appno’ VARCHAR(20) NOT NULL ,
‘fname’ VARCHAR (20) NOT NULL ,
‘mname’ VARCHAR(20) NOT NULL ,
‘lname’ VARCHAR(20) NOT NULL ,
‘dob’ DATE,
‘hostel’ VARCHAR(20) NOT NULL ,
‘userid` VARCHAR(20) NOT NULL ,
‘status’VARCHAR(20) NOT NULL ,
‘hostelcode’ references hostel(code) ,
Primary key(‘userid’ , ’hostelcode’)))
ENGINE = InnoDB;
```

CHAPTER 5

SNAPSHOTS

5.1 Applicant Login page

A user can enter his/her login details, which will take them to home page.

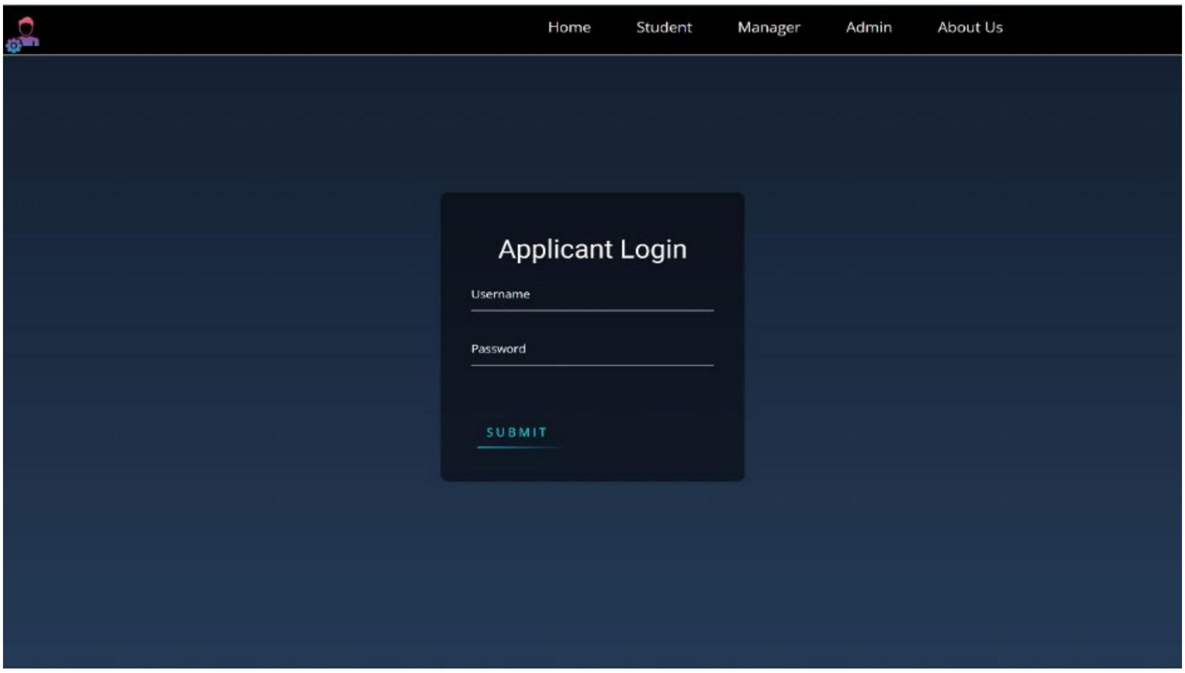


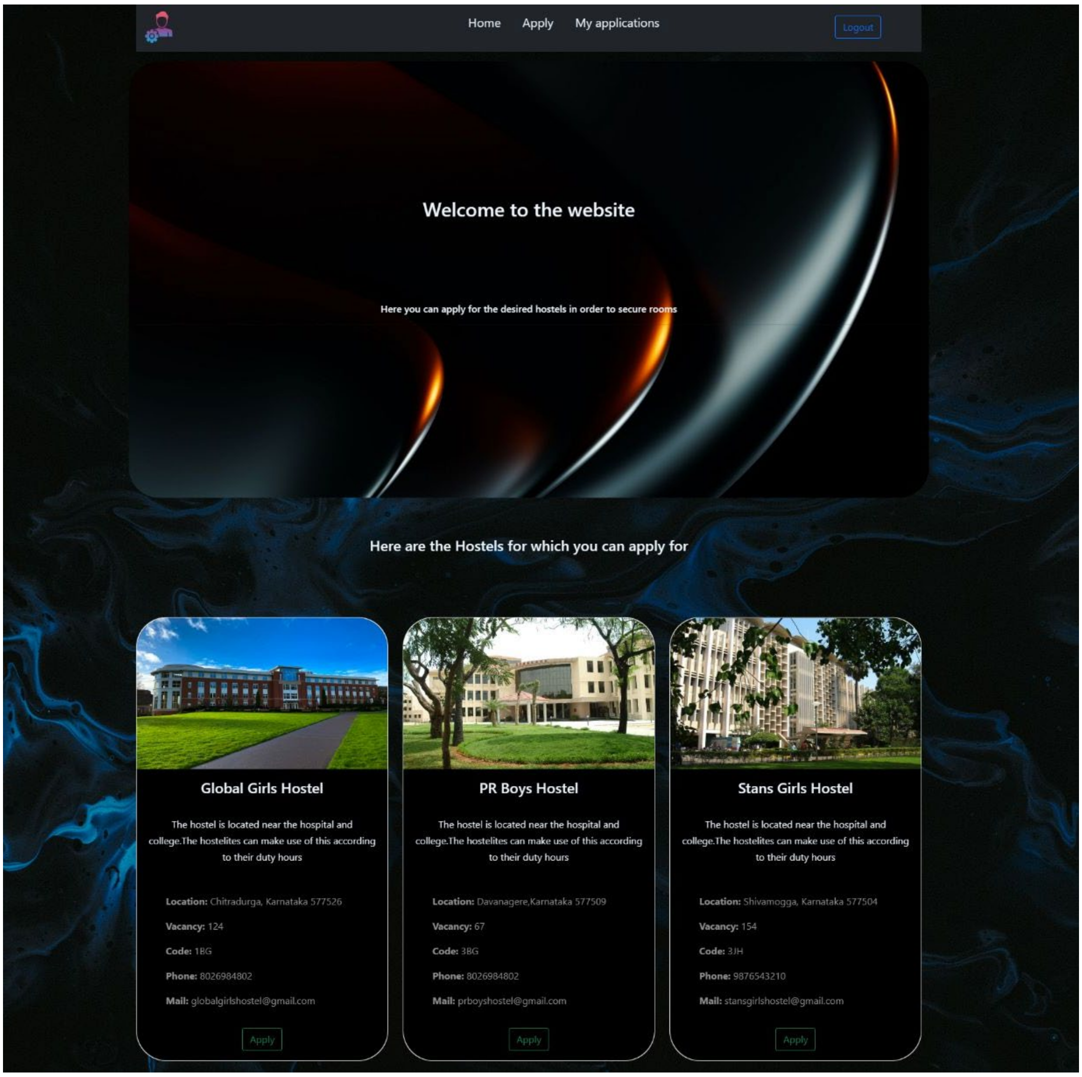
Fig 5.1 Login page for applicant

Figure 5.1 shows the login page for the applicant where in the applicant can log in using their credentials

- Each time a user needs to avail the services they need to login in using their username and password set by them.
- The username and password for a user is chosen by the user themselves at the time of creating the user account.
- New users are used to create the user account using the signup page by providing all the required information.
- It a the process of authenticating in order to let user access their stored information.
- It also facilitates user to continue their progress from where they had left before.
- Wrong password else an incorrect username is notified on the login page letting only authorized users to access the contents of their account.
- This authentication step is very much essential as a user account contains has personal details such as address, contact details etc. which must be prevented from getting in to wrong hands.

5.2 Home page

As soon as the User enters the website, the very first page visible is the home page.



Welcome to the website

Here you can apply for the desired hostels in order to secure rooms

Here are the Hostels for which you can apply for

Global Girls Hostel

The hostel is located near the hospital and college.The hostelites can make use of this according to their duty hours

Location: Chitradurga, Karnataka 577526

Vacancy: 124

Code: 1BG

Phone: 8026984802

Mail: globalgirlshostel@gmail.com

PR Boys Hostel

The hostel is located near the hospital and college.The hostelites can make use of this according to their duty hours

Location: Davanagere,Karnataka 577509

Vacancy: 67

Code: 3BG

Phone: 8026984802

Mail: prboyshostel@gmail.com

Stans Girls Hostel

The hostel is located near the hospital and college.The hostelites can make use of this according to their duty hours

Location: Shivamogga, Karnataka 577504

Vacancy: 154

Code: 3JH

Phone: 9876543210

Mail: stansgirlshostel@gmail.com

Fig 5.2 Home page of user

Figure 5.2 shows the home page of the user which is visible to them as soon as they enter and can redirect according to their needs.

- Every user lands upon the Home page each time upon successful login.
- All the available services can be accessed from the home page.
- It posses a welcome page and all the prioritized, most important notifications and information is displayed on the home page.
- The most relevant information and popular available hostels are displayed on the home page along with a short description clicking on which redirects the user to the information page of the hostel.

5.3 Sign up page

Here, a customer can enter their details to sign up for the service.

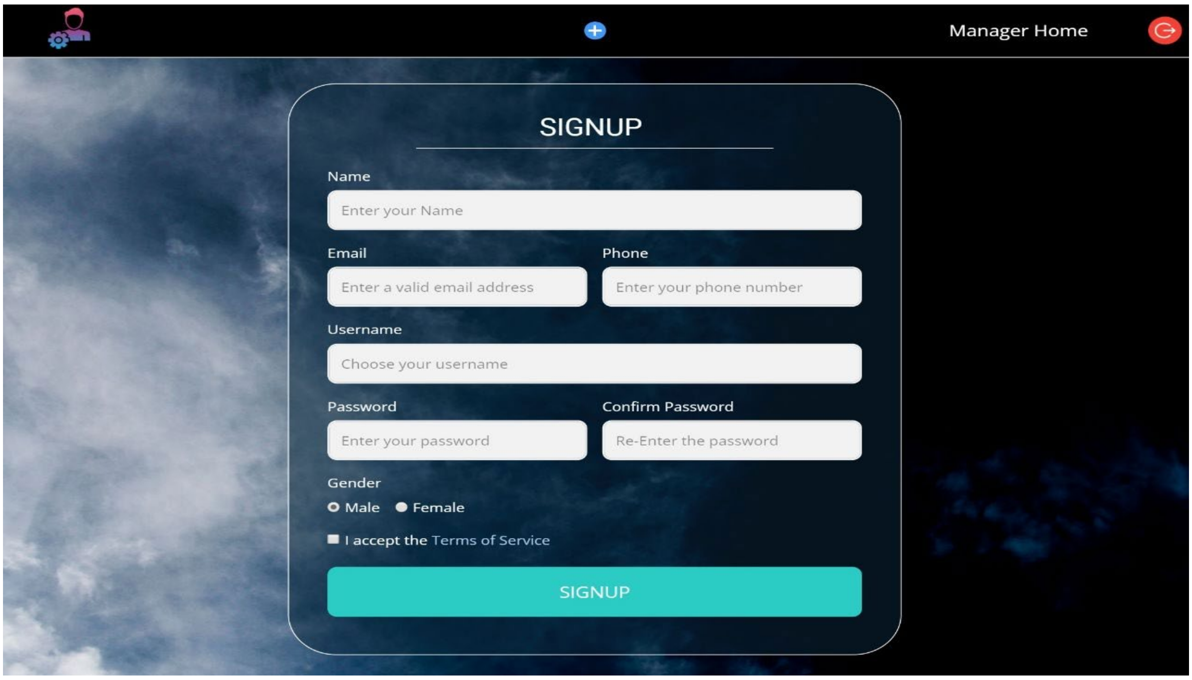
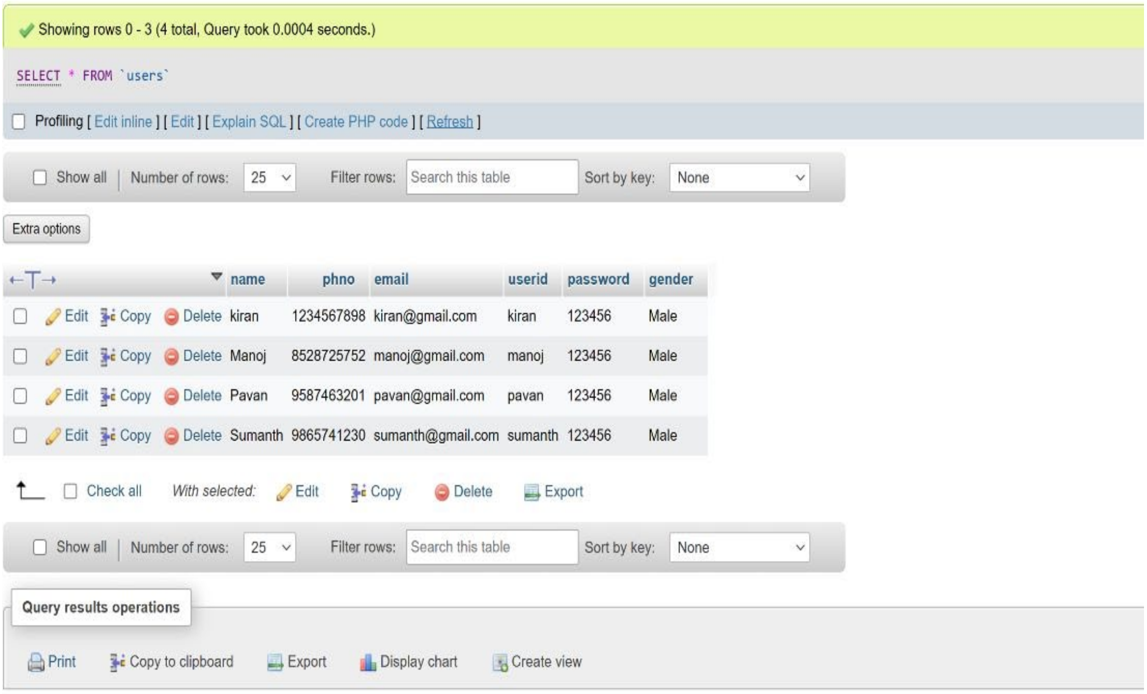


Fig 5.3 Sign up page for new user

Figure 5.3 displays the signup page for new users where they can create their account.

Table 5.1 Users table



✓ Showing rows 0 - 3 (4 total, Query took 0.0004 seconds.)

```
SELECT * FROM `users`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

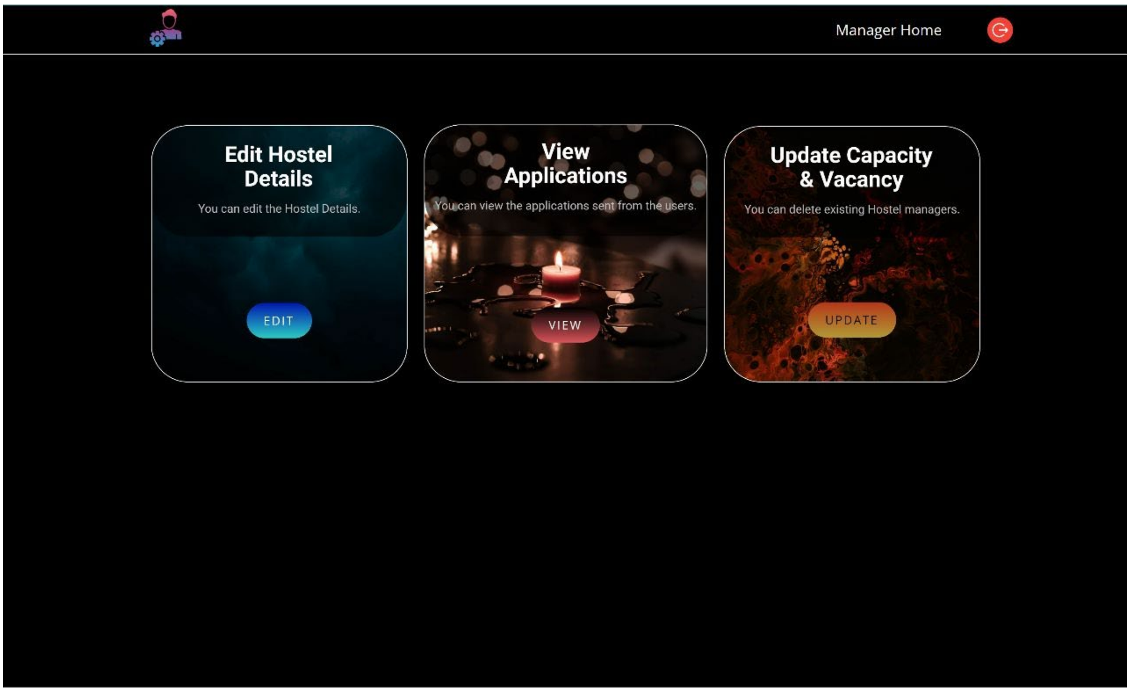
	name	phno	email	userid	password	gender
□ Edit Copy Delete	kiran	1234567898	kiran@gmail.com	kiran	123456	Male
□ Edit Copy Delete	Manoj	8528725752	manoj@gmail.com	manoj	123456	Male
□ Edit Copy Delete	Pavan	9587463201	pavan@gmail.com	pavan	123456	Male
□ Edit Copy Delete	Sumanth	9865741230	sumanth@gmail.com	sumanth	123456	Male

□ Check all With selected: Edit Copy Delete Export
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view

Table 5.1 Show the list of users who have signed up using the signup page along with their user credentials.

5.4 Manager home page

As soon as the Manager enters the website, the very first page visible is the Managers' home page.



You can edit the Hostel Details.

You can view the applications sent from the users.

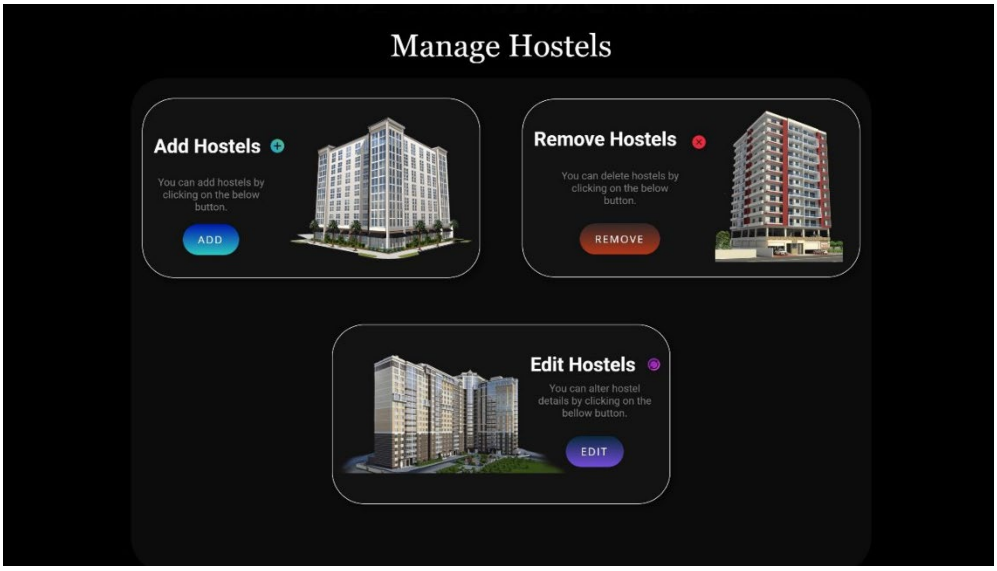
You can delete existing Hostel managers.

Fig 5.4 Manager home page

Figure 5.4 Shows the Home page of the manager from where they access their services .

5.5 Admin home page

As soon as the Admin enters his credentials in login page, the very first page visible is the home page.



You can add hostels by clicking on the below button.

You can delete hostels by clicking on the below button.

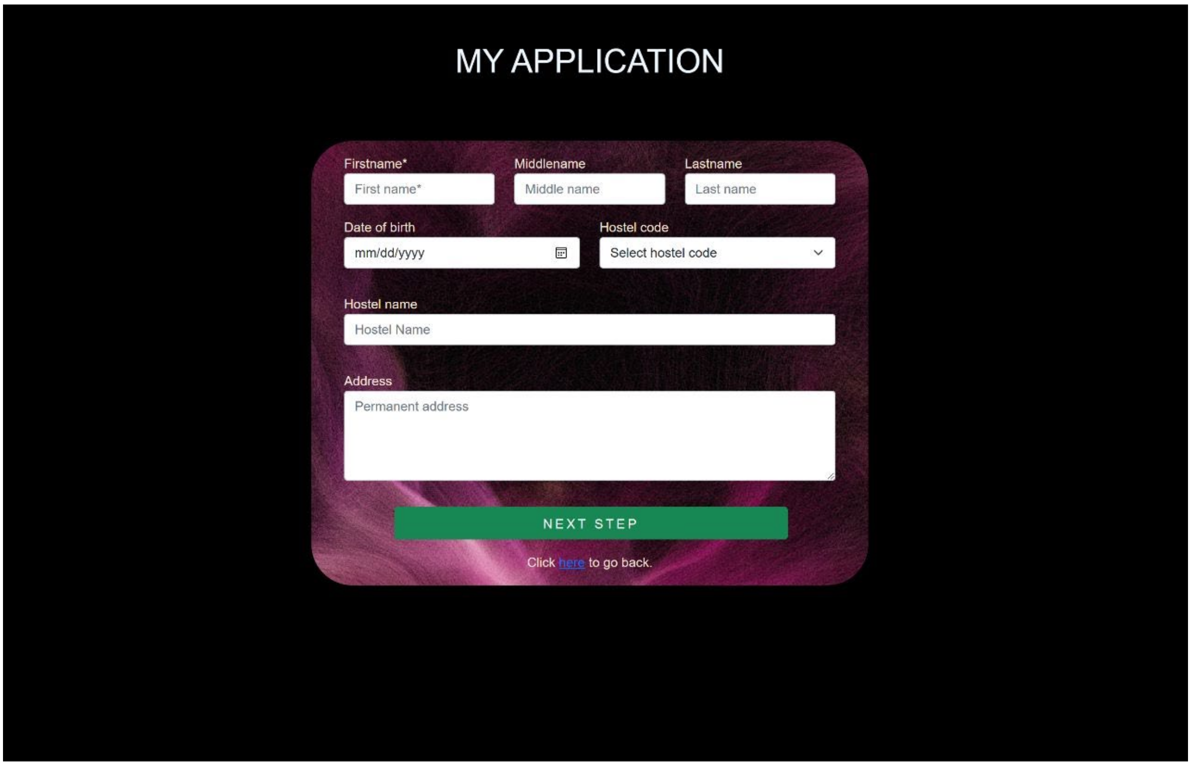
You can alter hostel details by clicking on the bellow button.

Fig 5.5 Admin home page

Figure 5.5 displays the home page of the admin using which the admin can add, edit and remove hostels.

5.6 Application form

This is a snap of application form where user can apply for the desired hostel.



MY APPLICATION

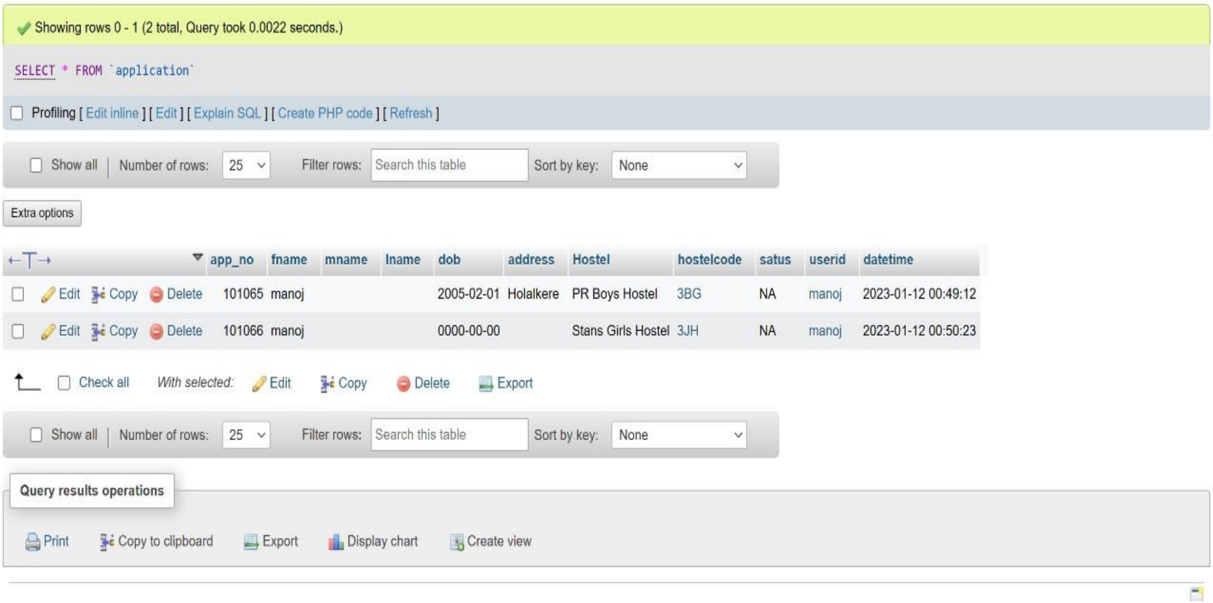
Firstname* First name*
Middlename Middle name
Lastname Last name
Date of birth mm/dd/yyyy
Hostel code Select hostel code
Hostel name Hostel Name
Address Permanent address

Fig 5.6 Application form for candidate

The figure 5.6 is the application which the applicant needs fill up with appropriate information in order to proceed further to apply for a hostel.

Submitted information in applications form will be stored in applications table.

Table 5.2 Applications table



✓ Showing rows 0 - 1 (2 total, Query took 0.0022 seconds.)

```
SELECT * FROM `application`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

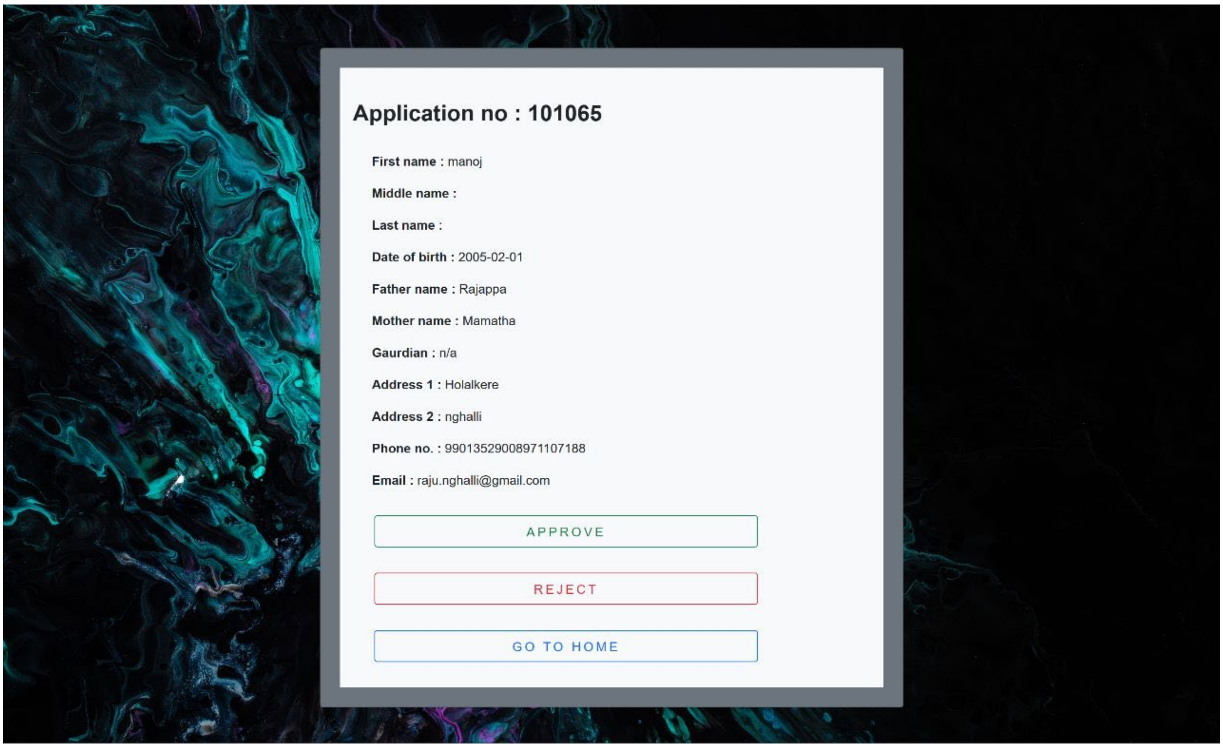
	app_no	fname	mname	lname	dob	address	Hostel	hostelcode	satus	userid	datetime
□ Edit Copy Delete	101065	manoj			2005-02-01	Holalkere	PR Boys Hostel	3BG	NA	manoj	2023-01-12 00:49:12
□ Edit Copy Delete	101066	manoj			0000-00-00		Stans Girls Hostel	3JH	NA	manoj	2023-01-12 00:50:23

□ Check all With selected: Edit Copy Delete Export
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view

Table 5.2 shows all the application forms that are successfully submitted by applicants.

5.7 Manager checking the applied applications

This is a image of the application showing to Manager to accept or reject.



Application no : 101065

First name : manoj

Middle name :

Last name :

Date of birth : 2005-02-01

Father name : Rajappa

Mother name : Mamatha

Gaurdian : n/a

Address 1 : Holalkere

Address 2 : nghalli

Phone no. : 99013529008971107188

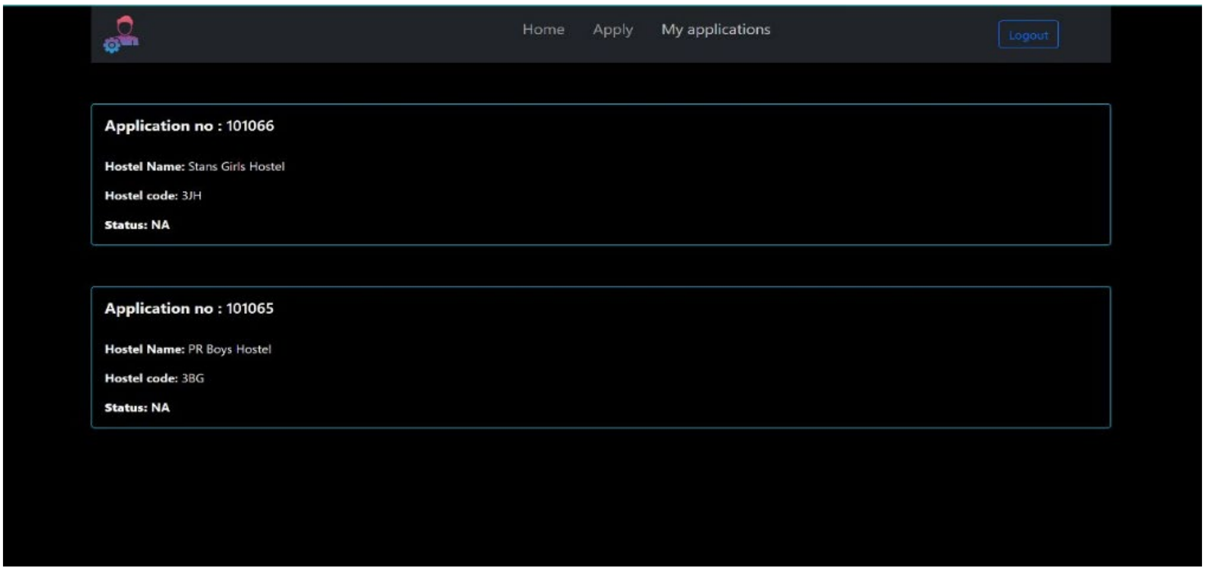
Email : raju.nghalli@gmail.com

Fig 5.7 Manager checking new applications

Figure 5.7 shows the received applications displayed to the manager for approval wherein the manager can review the application before approving else rejecting the application.

5.8 Displaying the status of application to user

View of all the applications and their status that user has applied.



Application no : 101066

Hostel Name: Stans Girls Hostel

Hostel code: 3JH

Status: NA

Application no : 101065

Hostel Name: PR Boys Hostel

Hostel code: 3BG

Status: NA

Fig 5.8 Displaying status of users' application

Figure 5.8 how the status of the of the application is displayed to applicant .

CHAPTER 6

CONCLUSION

Thus we have successfully implemented Hostel database of an area with the help of centralizing the data used for managing the records of candidates and their request.. We have successfully implemented the various functionalities of HTML, CSS, Bootstrap to design our GUI and various features of PHP myadmin SQL to design our database and thus created a fully functional database for the project Hostel database of an area.

At the end it is concluded that we have made effort on following points:

- Made statement of the aims and objectives of the project.
- The description of purpose, scope and applicability.
- We define the topic on which we are working in the project.
- We include features and operations in detail, including screen layouts.
- We designed user interface and security issues related to system.
- Finally, the system is implemented and tested according to test case.

REFERENCES

[1] https://www.w3schools.com/sql/sql_create_table.asp

[2] https://www.w3schools.com/sql/sql_drop_table.asp

[3] https://www.geeksforgeeks.org/python-gui-tkinter/

[4] https://www.w3schools.com/sql/sql_select.asp

[5] https://www.nicepage.com

[6] Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson

[7] Reference Book - Database System Concepts, Silberschatz Korth and Sudharshan, 6th Edition, McGraw Hill, 2013.