SE Project Report

Hostel Management System

Abstract:
In this emerging world of computers, the almost all-manual system has switched to an automated and computerized system. Therefore, we are developing the software for the "Hostel Management System" to model the present system and to remove the drawbacks of the present system and in every college it is mandatory to build hostels. In earlier days the hostel management was controlled through books under the control of the corresponding warden. Today the technology is improving a lot, controlling the whole hostel for several things through hand-written documents is very hard for warden's and the students who are coming from several places,they don't know how many rooms are free and how many rooms are booked,payment method etc. At this time to resolve this problem "Hostel management "comes into picture. The goal of the Hostel Management System is to automate the existing manual system using computerized equipment and full-featured computer software to meet their needs, so that their valuable data/information may be stored for a longer period of time with easy access and manipulation. The necessary software and hardware are both readily available for ease of use and we developed this system to reduce the complexity and efficiency of a versatile and outsourcing hostel management system.

Software Requirements Specification(SRS):

1. Introduction

1.1 Purpose
The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's intended audience and its user interface, hardware and software requirements for the product. It defines how our end client, team and audience see the product and its functionality.

1.2 Intended Audience
The intended audience is the students who are staying in hostels. The students face a lot of problems while they are staying in their hostels, so we want to automate the entire hostel management so that they can enjoy staying at hostels without facing any problems and if there are no problems faced by students in the hostel then few more students would be interested to use the hostel facility.

1.3 Intended Use
This document is to provide a detailed overview of our software product, its parameters and goals and also serves as a contract between the manager of the software and the

developers and testers where the manager can clearly see what and how the developers intend to do to make the software.

1.4 Scope

The goal is to design robust software for hostel management. In the project, we will fully automate the entire work of the hostels. Whenever a student chooses to stay in the hostel, one should visit our website and then he/she should create an account by filling required details and then once the account was created he/she should choose their choice of rooms in the available rooms such as (2person, 4person, etc.) and then we will be proceeding with the payment and confirmation of the room. When a student residing in a hostel requests to get his room cleaned, a notification will be sent to the help desk staff and then the help desk staff will send the housekeeping staff employee to clean the room at the time mentioned in the request. In case a student faces a problem related to electrical works in the room, a notification will be sent to the help desk again and they will notify the electrician and he will reach the room in the mutually accepted time and any other complaints regarding carpentry works, plumbing issues and AC repairs will be catered similarly.

The software must be able to perform the following operations:

I. Student details: It must be able to take the required student details to create an account.

II. Room details: It must be able to show the available rooms for the hostellers to choose the rooms of their choice to stay in the hostel.

III. Payment and confirmation: It must be able to confirm if the payment is done for the chosen room and allocate that room.

IV. Take requests: It must be able to take requests from the students in the hostel about the type of work required i.e. plumbing, room cleaning, AC repair etc.

V. Notify staff: It must be able to notify the help desk staff about the request made by hostellers and once the help desk staff got the request then a person is allotted to do that task.

Initially, we plan to implement the project for only one of the hostels in the city, in which only limited students residing in different rooms of the same hostel will be given access as a part of the Pilot Phase. Once the Pilot Phase is successful then we plan to implement it in the entire hostel. After we get to know all the vulnerabilities of the project, we can implement it in the different hostels and slowly increase the number.

The scope of this project is not just limited to the one hostel or hostels in the same city or one college campus only as the same mechanism can be reused in other campuses as well. There are an uncountable number of institutes in the country where automation of the hostels is required so that students can enjoy their stay at hostels. This system can also be implemented in big cities where service apartments are quite common.

1.5 Document conventions

We will display the Hostel Management System (HRM) on a user-friendly and user-centric website.

1.6 Contact information/SRS team members

K.Sravan

T.Anudeep

C.Abhinav

1.7 Definitions and Acronyms

- SRS – System Requirement Specifications
- DFD – Data Flow Diagram
- ERD – Entity Relationship Diagram
- HMS – Hostel Management System
- User – Student who lives in the hostel
- Database – Records of every hosteller

2. Overall Description

The product will run as a website wherein when the concerned person opens the homepage of the website, the person will be asked to sign up if it’s the first time he or she is using the portal. If they have been using the site, they will be asked to log in. Once they log in, they can see the different options available in the portal and choose the required option and then proceed with the room details and payment methods there will also be an option like help/form where they complain about the issues they are facing and these are taken care by help desk staff and then they will inform the required ff about the issues and then they will be resolved.

2.1 User Neons

The product is going to be used by colleges and universities who are providing hostel facilities to the students and college management is the primary user as the primary user is in direct contact with the system interface and they are having access to the database too and students are the secondary users as they create an account on the portal and choose the rooms of the hostel to stay in them and raise the complaints in their rooms if they have any.

2.2 Assumptions and Dependencies

The following list prevents the assumptions, dependencies or guidelines that are imposed upon the implementation of System:

I. The product must have a user-friendly interface that is simple enough for all types of users to understand.

II. Response time should not be longer than 5 seconds

III. General knowledge of basic computer skills or usage of a smartphone knowledge is required to use the product.

IV. They should have the internet to use the product.

2.3 Operating environment

Operating environment for the hostel management system includes

Operating system: Windows XP, 7/ Mac OS

Front End: HTML, CSS, JavaScript, PHP

Database(Back End): SQL SERVER

Admin and Student side system

The admin and student side components of the software system must operate within common web browser environments. The browsers that must be included and supported are

- Apple Safari 7+
- Google Chrome 44+
- Microsoft Internet Explorer 10+
- Mozilla Firefox 40+

2.5 User environment

Any user is the target audience for the software system that is provided. It gives users access and guidelines to register and view different modules that are present in the

software. The users will be given a detailed description of how to register and about every step on the website.

2.6 Design/implementation constraints

The Administration Department should implement a Security Policy to ensure the safety of the information contained in our database. A Bug Bounty Team can help catch these errors in the initial stages.

Since there would be many departments in a hostel for the maintenance of hostel information and student databases in any hostel. All these departments provide various records regarding students. Most of the records must contain information about the students. General details like student name, address, number, etc. are stored there. All the modules in hostel administration are interdependent and maintained manually. So they need to be automated and centralized to avoid redundancy.

3. System Features and Requirements

3.1 Functional Requirements

- Admin can be able to enter the student details and he can perform a few actions such as deletion of student records or adding new student records. Admins will no longer have to maintain a book and register the student details. They can register them with one click through or web portal and can save them or edit them whenever they want.
- Admin can search for the student records and the data stored should be obtained within the stipulated time. Delays in time may result in the usage of the software and usage of the software is preferred over the book so that we can save time, while manually it takes a lot of time.
- Provide hostlers with the ability to complain about problems facing the hostel. Residents will no longer have to write about the problems and complaints in the register which are not read by anyone in the management. They can register a complaint with one click through or web portal. Taking into consideration different parameters, a form is made to monitor the quality of the hostel. After filling out the form, the residents can also give suggestions or register a complaint in the other box.
- Admin can maintain the track of the payment records by using the website and they can go through that whenever it is necessary.

- Students are provided with different options of rooms and bed choices in the website so that they can choose according to their choice and proceed with the payment gateway and while registration they are requested to enter some personal details about them for a successful registration.
- **Admin Login:** This facility is for authorizing access to the system. (Admin)Then he should be able to retrieve the information of his students and enter the choice of hostel room details accordingly. It is the job of the administrator to insert, update and monitor the whole process.
- **Student Login:** They will be able to log in and proceed with the registration for a room for having a comfortable stay.

3.2 External Interface Requirements

User Interfaces:

The goal is to design the software used for the proper management of hostels and automate the current process. The user types are listed as followed

I. Students/Hostellers
II. Hostel staff(people like electrician/plumber)
III. Help desk staff
IV. Administrator

Our goal is to develop a software that should be easy to use for all types of users. Thus while designing the software one can assume that each user type has the following characteristics:

I.The user is a computer-literate and has little or no difficulty in using the software keeping in mind the software is user friendly.
II. In order to use software a user must be aware of the internal working and expected to know how things work.
III. All the guidelines about the use of software will be informed to the user once the user signs up on the software or web page.

Hardware Interfaces:

I.Computer: A computer will be required to open the website and use the software
II. Smartphone: A smartphone can also be required in case there is no availability of computers for them
III. Internet: A good internet connection is required to access the website and without this if the person is having a computer or smartphone then there would be no use.

Software Interfaces:
I. A SQL Database Server will be required to store and retrieve data of the activities done in the hostel management system.
II. A web browser will be required to open the website.

Communication Interfaces:
I.The system shall be a standalone product that does not require any communication interfaces.

3.3 System Features

System feature- A

In the Student View, he can visit all the following pages:

- Rooms and bed details
- Payment related details
- Help desk page to lodge a complaint

3.3.1 Description and priority

The hostel management system maintains information about rooms, beds, information about student details, personal information requirements and hostel registration. This project has a high priority because it is very difficult for the hostel administrator to keep track of the hostlers data without this software and it is very beneficial for the students to check in quickly without having any delay and they don't need anyone to help them out and paperwork is eliminated with this software.

3.3.2 Action/result

Gives desired output of the students based on his interaction with the software. It finally allocates the room booked and helps students to have an identity.

System feature B

- Student Helpdesk is staffed by hostel wardens who are there to help you and You may contact the help desk at any point of time and raise a query that needs to be addressed.
- Queries are solved by contacting authorized persons and then students are notified about that and changing rooms or editing personal information also can be done through software.

3.4 Non-functional Requirements

Performance Requirements:

The application shall be based on web technology and has to be run on any platform. the application shall task initial load time depending on performance of the operating system. The performance shall depend upon hardware and software components of the computer or smartphone that we are using.

Safety Requirements:

The database may get crashed at any certain time due to the virus or operating system failure. Therefore, it is required to take backup of the database.

Security Requirements:

This project provides genuine security to all those individuals who are having their details on the database as they are password protected. This is a very important aspect of the design and should cover areas of hardware reliability, fallback procedures, physical security of data and provision for detection of fraud and abuse.

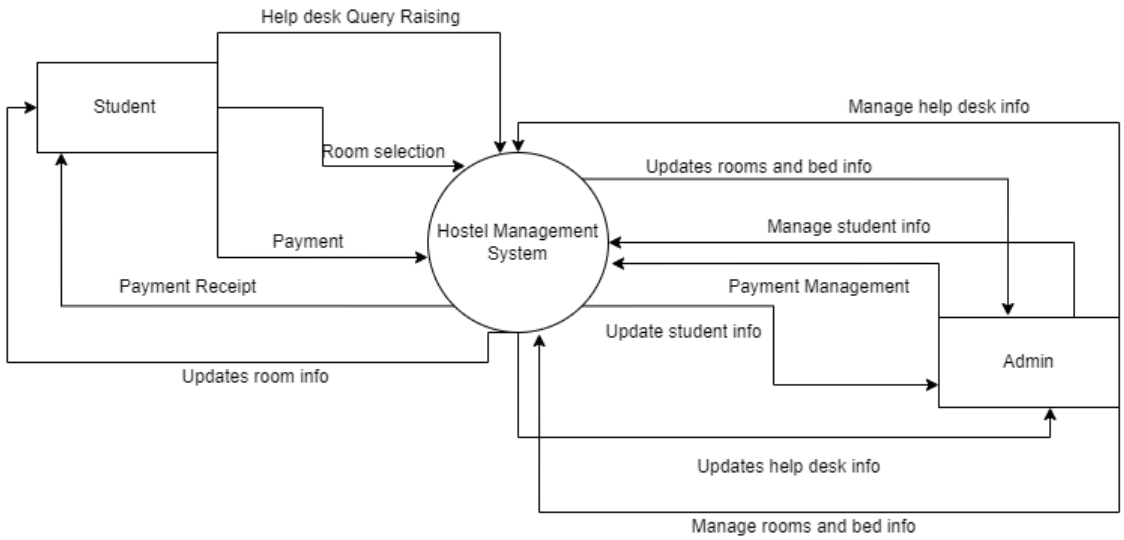
Quality Requirements:

The software is prepared to define six quality characteristics such as functionality, reliability, usability, maintainability, portability, and efficiency.

Design of the Project:

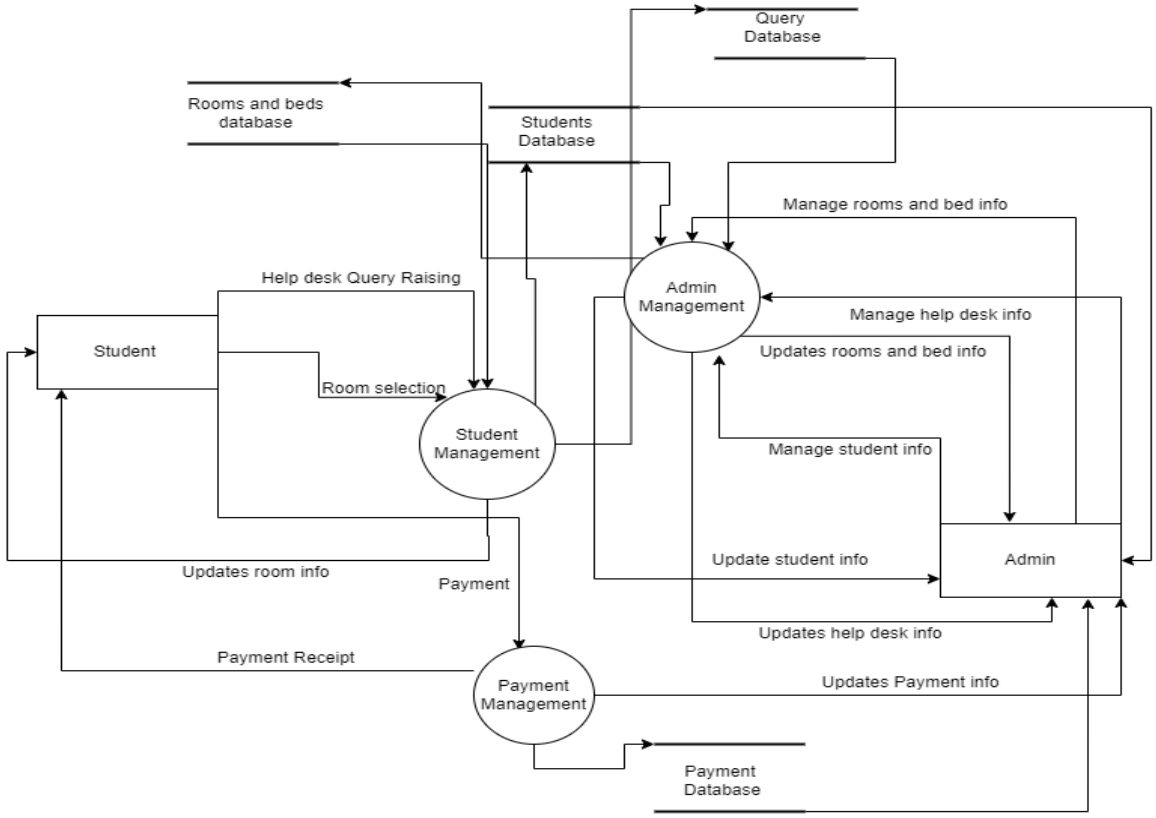
Level 0 DFD

https://app.diagrams.net/#G1edPuFm9UTnMlsDfoEBpsiWe5IMId51Ls



Level 1 DFD

https://app.diagrams.net/#G1jppwoqmfW8Bt4J_VnXHDcHY5hOwTdEZ1



For more diagrams refer to: Design of the HMS

Implementation of code:

Index:

```
<!doctype html>
<html lang="en" class="no-js">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1,
minimum-scale=1, maximum-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <meta name="theme-color" content="#3e454c">
    <title>Student Hostel Registration</title>
```

```
    <link rel="stylesheet" href="css/style.css">
<script type="text/javascript"
src="js/jquery-1.11.3-jquery.min.js"></script>
<script type="text/javascript" src="js/validation.min.js"></script>
<script type="text/javascript"
src="http://code.jquery.com/jquery.min.js"></script>
<script type="text/javascript">
function valid()
{
if(document.registration.password.value!=
document.registration.cpassword.value)
{
alert("Password and Retype Password Field do not match  !!");
document.registration.cpassword.focus();
return false;
}
return true;
}
</script>
</head>
<body>
    <?php include('includes/header.php');?>
    <div class="ts-main-content">
        <?php include('includes/sidebar.php');?>
        <div class="content-wrapper">
            <div class="container-fluid">
                <div class="row">
                    <div class="col-md-12">
                        <h2 class="page-title">User Login </h2>
                        <div class="row">
                    <div class="col-md-6 col-md-offset-3">
                        <div class="well row pt-2x pb-3x bk-light">
                            <div class="col-md-8 col-md-offset-2">

                                <form action="" class="mt" method="post">
                                    <label for="" class="text-uppercase
text-sm">Email</label>
                                    <input type="text" placeholder="Email"
name="email" class="form-control mb">
```

```
                                    <label for="" class="text-uppercase
text-sm">Password</label>
                                    <input type="password"
placeholder="Password" name="password" class="form-control mb">
                                    <input type="submit" name="login"
class="btn btn-primary btn-block" value="login" >
                                </form>
                            </div>
                        </div>
                        <div class="text-center text-light"
style="color:black;">
                            <a href="forgot-password.php"
style="color:black;">Forgot password?</a>
                        </div>
```

Room booking:

```
<div class="form-group">
<label class="col-sm-4 control-label"><h4 style="color: green"
align="left">Room Related info </h4> </label>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Room no. </label>
<div class="col-sm-8">
<select name="room" id="room"class="form-control"
onChange="getSeater(this.value);" onBlur="checkAvailability()" required>
<option value="">Select Room</option>
<?php $query ="SELECT * FROM rooms";
$stmt2 = $mysqli->prepare($query);
$stmt2->execute();
$res=$stmt2->get_result();
while($row=$res->fetch_object())
{
?>
<option value="<?php echo $row->room_no;?>"> <?php echo
$row->room_no;?></option>
<?php } ?>
</select>
<span id="room-availability-status" style="font-size:12px;"></span>

</div>
</div>
```

```
<div class="form-group">
<label class="col-sm-2 control-label">Seater</label>
<div class="col-sm-8">
<input type="text" name="seater" id="seater"  class="form-control"  >
</div>
</div>

<div class="form-group">
<label class="col-sm-2 control-label">Fees Per Month</label>
<div class="col-sm-8">
<input type="text" name="fpm" id="fpm"  class="form-control" >
</div>
</div>

<div class="form-group">
<label class="col-sm-2 control-label">Food Status</label>
<div class="col-sm-8">
<input type="radio" value="0" name="foodstatus" checked="checked"> Without
Food
<input type="radio" value="1" name="foodstatus"> With Food(Rs 2000.00 Per
Month Extra)
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Stay From</label>
<div class="col-sm-8">
<input type="date" name="stayf" id="stayf"  class="form-control" >
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Duration</label>
<div class="col-sm-8">
<select name="duration" id="duration" class="form-control">
<option value="">Select Duration in Month</option>
<option value="1">1</option>
<option value="2">2</option>
. . . . . .
<option value="12">12</option>
</select>
```

```
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Total Amount</label>
<div class="col-sm-8">
<input type="text" name="ta" id="ta"  class="result form-control" >
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label"><h4 style="color: green"
align="left">Personal info </h4> </label>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">course </label>
<div class="col-sm-8">
<select name="course" id="course" class="form-control" required>
<option value="">Select Course</option>
<?php $query ="SELECT * FROM courses";
$stmt2 = $mysqli->prepare($query);
$stmt2->execute();
$res=$stmt2->get_result();
while($row=$res->fetch_object())
{
?>
<option value="<?php echo $row->course_fn;?>"><?php echo
$row->course_fn;?>  (<?php echo $row->course_sn;?>)</option>
<?php } ?>
</select> </div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Registration No : </label>
<div class="col-sm-8">
<input type="text" name="regno" id="regno"  class="form-control"
value="<?php echo $row->regNo;?>" readonly >
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">First Name : </label>
<div class="col-sm-8">
<input type="text" name="fname" id="fname"  class="form-control"
value="<?php echo $row->firstName;?>" readonly>
```

```
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Middle Name : </label>
<div class="col-sm-8">
<input type="text" name="mname" id="mname"  class="form-control"
value="<?php echo $row->middleName;?>"  readonly>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Last Name : </label>
<div class="col-sm-8">
<input type="text" name="lname" id="lname"  class="form-control"
value="<?php echo $row->lastName;?>" readonly>
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Gender : </label>
<div class="col-sm-8">
<input type="text" name="gender" value="<?php echo $row->gender;?>"
class="form-control" readonly>
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Contact No : </label>
<div class="col-sm-8">
<input type="text" name="contact" id="contact" value="<?php echo
$row->contactNo;?>"  class="form-control" readonly>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Guardian  Name : </label>
<div class="col-sm-8">
<input type="text" name="gname" id="gname"  class="form-control"
required="required">
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Guardian  Relation : </label>
<div class="col-sm-8">
<input type="text" name="grelation" id="grelation"  class="form-control"
required="required">
</div>
<div class="form-group">
```

```
<label class="col-sm-2 control-label">Guardian Contact no : </label>
<div class="col-sm-8">
<input type="text" name="gcontact" id="gcontact"  class="form-control"
required="required">
</div>
<div class="form-group">
<label class="col-sm-3 control-label"><h4 style="color: green"
align="left">Correspondense Address </h4> </label>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Address : </label>
<div class="col-sm-8">
<textarea  rows="5" name="address"  id="address" class="form-control"
required="required"></textarea>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">City : </label>
<div class="col-sm-8">
<input type="text" name="city" id="city"  class="form-control"
required="required">
</div>
<div class="form-group">
<label class="col-sm-2 control-label">State </label>
<div class="col-sm-8">
<select name="state" id="state"class="form-control" required>
<option value="">Select State</option>
<option value="<?php echo $row->State;?>"><?php echo
$row->State;?></option>
<?php } ?>
</select> </div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Pincode : </label>
<div class="col-sm-8">
<input type="text" name="pincode" id="pincode"  class="form-control"
required="required">
</div>
</div>
<div class="form-group">
```

```
<label class="col-sm-3 control-label"><h4 style="color: green"
align="left">Permanent Address </h4> </label>
</div>
<div class="form-group">
<label class="col-sm-5 control-label">Permanent Address same as
Correspondense address : </label>
<div class="col-sm-4">
<input type="checkbox" name="adcheck" value="1"/>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Address : </label>
<div class="col-sm-8">
<textarea  rows="5" name="paddress"  id="paddress" class="form-control"
required="required"></textarea>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">City : </label>
<div class="col-sm-8">
<input type="text" name="pcity" id="pcity"  class="form-control"
required="required">
</div>
```

Student registration:

```
<div class="form-group">
<label class="col-sm-2 control-label">Room no. </label>
<div class="col-sm-8">
<select name="room" id="room"class="form-control"
onChange="getSeater(this.value);" onBlur="checkAvailability()" required>
<option value="">Select Room</option>
<?php $query ="SELECT * FROM rooms";
$stmt2 = $mysqli->prepare($query);
$stmt2->execute();
$res=$stmt2->get_result();
</select>
<span id="room-availability-status" style="font-size:12px;"></span>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Seater</label>
<div class="col-sm-8">
```

```
<input type="text" name="seater" id="seater"  class="form-control"  >
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Fees Per Month</label>
<div class="col-sm-8">
<input type="text" name="fpm" id="fpm"  class="form-control" >
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Food Status</label>
<div class="col-sm-8">
<input type="radio" value="0" name="foodstatus" checked="checked"> Without
Food
<input type="radio" value="1" name="foodstatus"> With Food(Rs 2000.00 Per
Month Extra)
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Stay From</label>
<div class="col-sm-8">
<input type="date" name="stayf" id="stayf"  class="form-control" >
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Duration</label>
<div class="col-sm-8">
<select name="duration" id="duration" class="form-control">
<option value="">Select Duration in Month</option>
<option value="1">1</option>
<option value="2">2</option>
. . . . . .
<option value="12">12</option>
</select>
</div>
<div class="form-group">
<label class="col-sm-2 control-label"><h4 style="color: green"
align="left">Personal info </h4> </label>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">course </label>
<div class="col-sm-8">
<select name="course" id="course" class="form-control" required>
<option value="">Select Course</option>
```

```
<option value="<?php echo $row->course_fn;?>"><?php echo
$row->course_fn;?>  (<?php echo $row->course_sn;?>)</option>
<?php } ?>
</select> </div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Registration No : </label>
<div class="col-sm-8">
<input type="text" name="regno" id="regno"  class="form-control"
required="required" >
</div>
<div class="form-group">
<label class="col-sm-2 control-label">First Name : </label>
<div class="col-sm-8">
<input type="text" name="fname" id="fname"  class="form-control"
required="required" >
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Middle Name : </label>
<div class="col-sm-8">
<input type="text" name="mname" id="mname"  class="form-control">
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Last Name : </label>
<div class="col-sm-8">
<input type="text" name="lname" id="lname"  class="form-control"
required="required">
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Gender : </label>
<div class="col-sm-8">
<select name="gender" class="form-control" required="required">
<option value="">Select Gender</option>
<option value="male">Male</option>
<option value="female">Female</option>
<option value="others">Others</option>
</select>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Contact No : </label>
```

```
<div class="col-sm-8">
<input type="text" name="contact" id="contact"  class="form-control"
required="required">
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Email id : </label>
<div class="col-sm-8">
<input type="email" name="email" id="email"  class="form-control"
required="required">
</div>
</div>

<div class="form-group">
<label class="col-sm-2 control-label">Emergency Contact: </label>
<div class="col-sm-8">
<input type="text" name="econtact" id="econtact"  class="form-control"
required="required">
</div>
</div>

<div class="form-group">
<label class="col-sm-2 control-label">Guardian  Name : </label>
<div class="col-sm-8">
<input type="text" name="gname" id="gname"  class="form-control"
required="required">
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Guardian  Relation : </label>
<div class="col-sm-8">
<input type="text" name="grelation" id="grelation"  class="form-control"
required="required">
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Guardian Contact no : </label>
<div class="col-sm-8">
<input type="text" name="gcontact" id="gcontact"  class="form-control"
required="required">
</div>
```

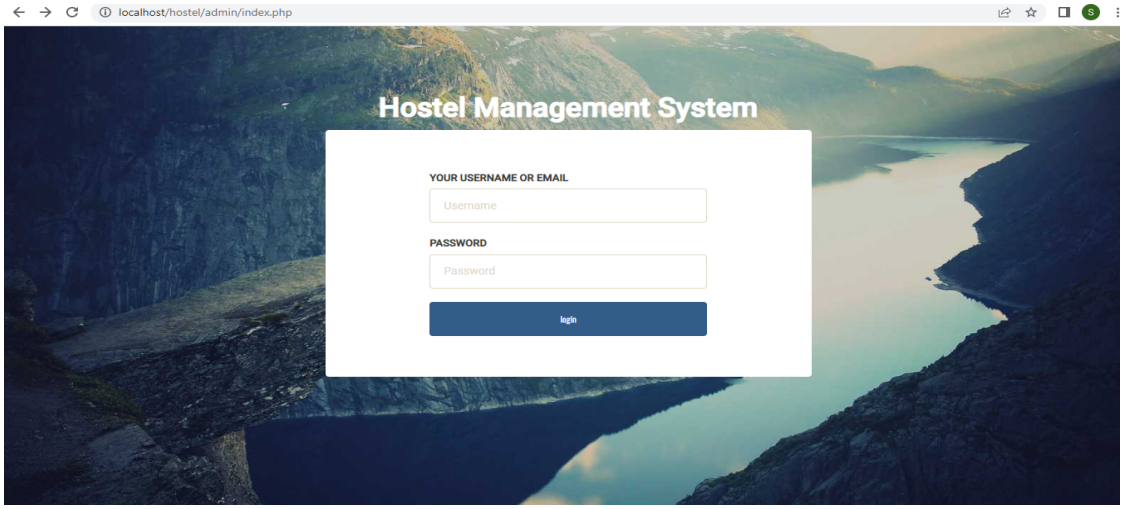
```
<div class="form-group">
<label class="col-sm-3 control-label"><h4 style="color: green"
align="left">Correspondense Address </h4> </label>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Address : </label>
<div class="col-sm-8">
<textarea  rows="5" name="address"  id="address" class="form-control"
required="required"></textarea>
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">City : </label>
<div class="col-sm-8">
<input type="text" name="city" id="city"  class="form-control"
required="required">
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">State </label>
<div class="col-sm-8">
<select name="state" id="state"class="form-control" required>
<option value="<?php echo $row->State;?>"><?php echo
$row->State;?></option>
<?php } ?>
</select> </div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Pincode : </label>
<div class="col-sm-8">
<input type="text" name="pincode" id="pincode"  class="form-control"
required="required">
</div>
</div>
<div class="form-group">
<label class="col-sm-3 control-label"><h4 style="color: green"
align="left">Permanent Address </h4> </label>
</div>
<div class="form-group">
```

```
<label class="col-sm-5 control-label">Permanent Address same as
Correspondense address : </label>
<div class="col-sm-4">
<input type="checkbox" name="adcheck" value="1"/>
</div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Address : </label>
<div class="col-sm-8">
<textarea  rows="5" name="paddress"  id="paddress" class="form-control"
required="required"></textarea>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">City : </label>
<div class="col-sm-8">
<input type="text" name="pcity" id="pcity"  class="form-control"
required="required">
</div>
<div class="form-group">
<label class="col-sm-2 control-label">State </label>
<div class="col-sm-8">
<select name="pstate" id="pstate"class="form-control" required>
<option value="">Select State</option>
<option value="<?php echo $row->State;?>"><?php echo
</select> </div>
</div>
<div class="form-group">
<label class="col-sm-2 control-label">Pincode : </label>
<div class="col-sm-8">
<input type="text" name="ppincode" id="ppincode"  class="form-control"
required="required">
</div>
<div class="col-sm-6 col-sm-offset-4">
<button class="btn btn-default" type="submit">Cancel</button>
<input type="submit" name="submit" Value="Register" class="btn
btn-primary">
</div>
</form>
 </div>
```

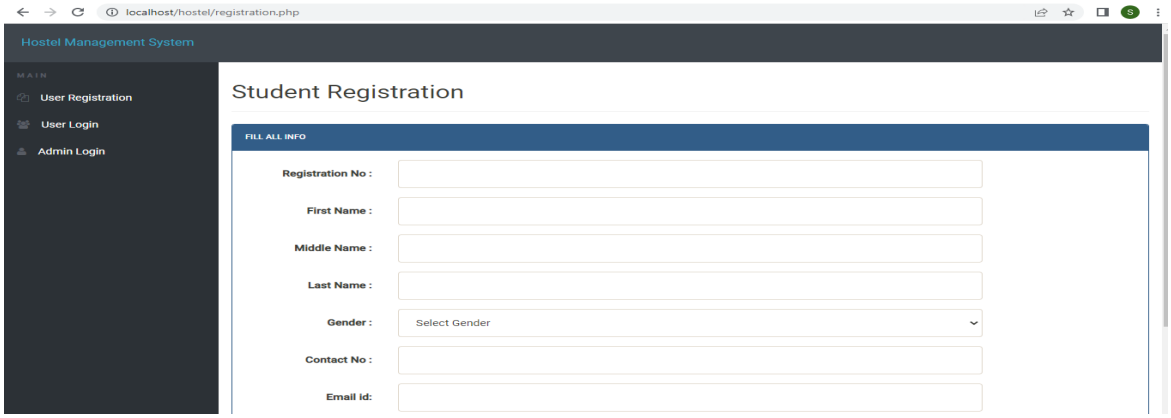
For total system code refer to the ZIP file submitted along with the report.

Sample Results:

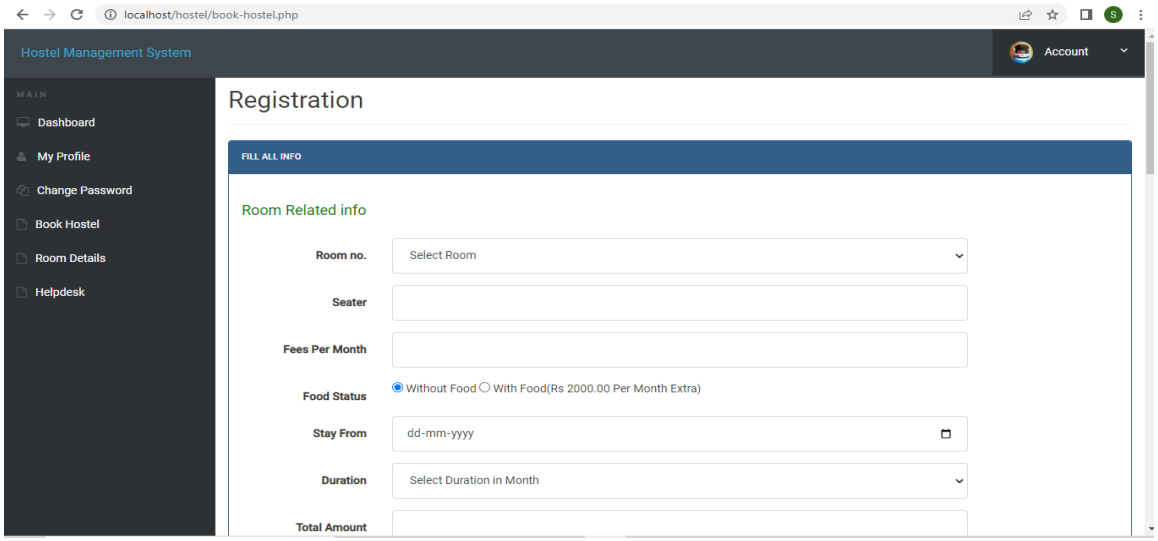
Admin login page:



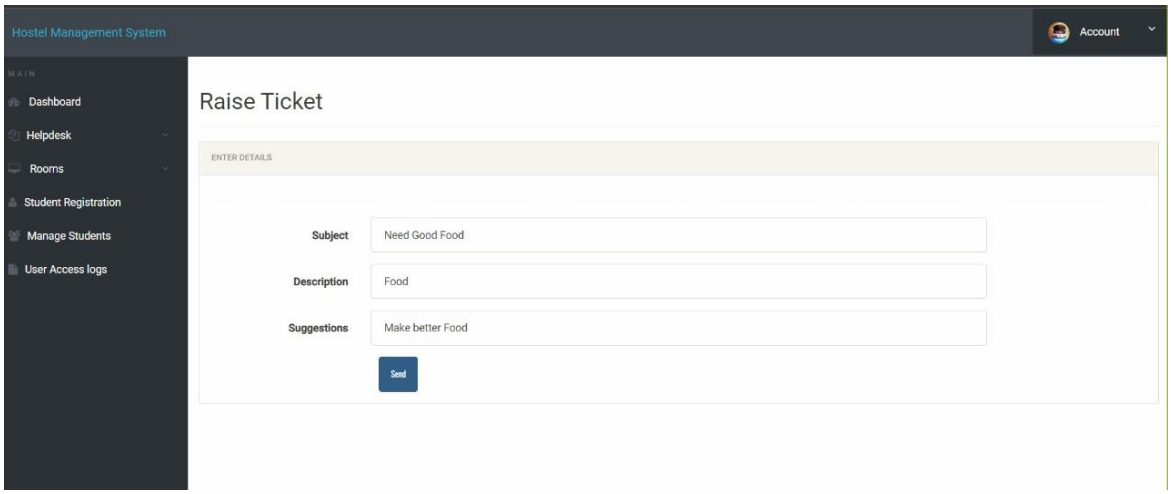
Student registration:



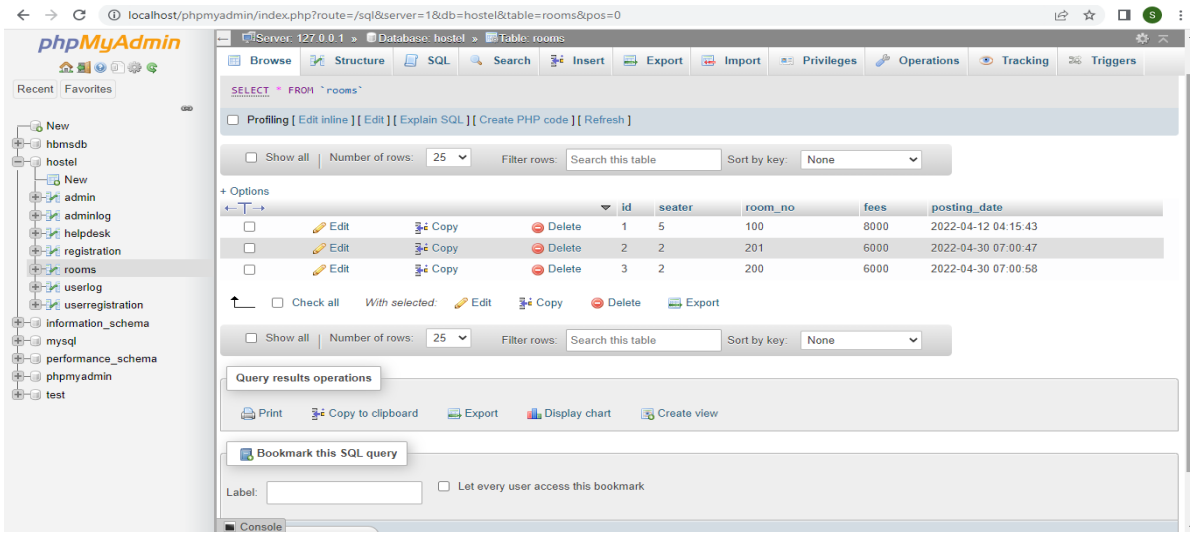
Registration of a room:



Help desk (ticket raising):



Available rooms database:



References:

- https://www.slideshare.net/hiraakram11/hostel-management-system-srs
- https://www.researchgate.net/publication/312490400_Development_of_an_Automated_Hostel_Facility_Management_System
- https://www.academia.edu/37040228/HOSTEL_MANAGEMENT_SYSTEM

Acknowledgement:

Here we gladly present this project report on "HOSTEL MANAGEMENT SYSTEM" as part of our course in B.TECH in Computer Science and Engineering. At this time of submitting this report we use this opportunity to mention our sincere and heartfelt thanks to our esteemed professor Dr. Hiren Kumar Deva Sharma for giving us the project work to do and gain real time experience in the Software world and providing us with the right guidance and advice at the crucial movements and for showing us the right way. This will help us to be way forward in the real world in the near future. He's been very cooperative and motivated us to complete this project for our own good. We will always be thankful to him and our university SRM University, AP.