

A Project Report On

**“HOSTEL MANAGEMENT SYSTEM”**

Submitted in partial fulfillment of the

**MASTER OF COMPUTER APPLICATION**

By

**ARUNABH DAS**

**(Enrolment Number – AJU/221788)**

Under the esteemed guidance of

**Dr. Arvind Kumar Pandey**

(Head of Department)

**Dr. Arun Kumar Marandi**

(Internal Guide)

ARKA JAIN UNIVERSITY

JAMSHEDPUR, JHARKHAND

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

2022 – 2024

****

**CERTIFICATE**

This is to certify that the project entitled, “**HOSTEL MANAGEMENT SYSTEM”**, is bonafide work of **Arunabh Das** bearing **Enrolment no – AJU/221788** under the guidance of **Dr. Arun Marandi** submitted in partial fulfillment of the requirements for the award of degree of MASTER OF COMPUTER APPLICATION from ARKA JAIN UNIVERSITY, Jharkhand.

Internal Guide H.O.D

Dr. Arun Kumar Marandi Dr. Arvind Kumar Pandey

(Associate Professor) (Head of the Department)

Department of Cs & IT Department of CS & IT

ARKA JAIN, University ARKA JAIN , University

Jamshedpur,Jharkhand Jamshedpur,Jharkhand

Date: University Seal

****

**ABSTRACT**

As the name specifies “**HOSTEL MANAGEMENT SYSTEM**” is software developed for managing various activities in the hostel. For the past few years the number of educational institutions is increasing rapidly.

Thereby the number of hostels is also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software’s are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.

Identification of the drawbacks of the existing system leads to the development of computerized hostel management system that will be compatible to the existing system with the system which is more users friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing hostel management system less human error, Strength and strain of manual labor can be reduced, High security, Data redundancy can be avoided to some extent, Data consistency, Easy to handle, Easy data updating, Easy record keeping, Backup data can be easily generated.

****

**ACKNOWLEDGEMENT**

We take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination.

It is a genuine pleasure to express my profound gratitude and deep regard to my guide “**Dr. Arun Marandi”** for their exemplary guidance, monitoring and constant encouragement.

I would like to specially thank “**Dr. Arvind Kumar Pandey**“ our Head of the Department who gave me the golden opportunity to do this wonderful project on the topic “Hotel Management System”, which helped me in research and I came to know about so many things.

****

**DECLARATION**

I, Arunabh Das hear by declare that the Project entitled “**Hostel Management System**” done at **ARKA JAIN UNIVERSITY** has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

This project is done in the fulfillment of the requirement for the award degree of MASTER OF COMPUTER APPLICATION to be submitted as a mini project as part of our curriculum.

ARUNABH DAS

(AJU/221788)

Signature of Student

TABLE OF CONTENTS

[Ch 1. INTRODUCTION 1](#_Toc140878867)

[1.1 OVERVIEW 1](#_Toc140878868)

[1.2 OBJECTIVES 1](#_Toc140878869)

[1.3 ADMIN PANEL 2](#_Toc140878870)

[1.3.1 Administrator login 2](#_Toc140878871)

[1.3.2 Admin Dashboard Home 2](#_Toc140878872)

[1.3.3 Register Students 2](#_Toc140878873)

[1.3.4 Manage Students 2](#_Toc140878874)

[1.3.5 Book Hostel 2](#_Toc140878875)

[1.3.6 Hostel Students 2](#_Toc140878876)

[1.3.7 Manage Rooms 2](#_Toc140878877)

[1.4 STUDENT PANEL 3](#_Toc140878878)

[1.4.1 Student login 3](#_Toc140878879)

[1.4.2 Student Dashboard Home 3](#_Toc140878880)

[1.4.3 Book Hostel 3](#_Toc140878881)

[1.4.4 My Room Details 3](#_Toc140878882)

[Ch 2. REQUIREMENT AND ANALYSIS 4](#_Toc140878883)

[2.1 HARWARE AND SOFTWARE CONFIGURATION 4](#_Toc140878884)

[Ch 3. DETAILED DESIGN 5](#_Toc140878885)

[3.1 Data Flow Diagram 5](#_Toc140878886)

[3.2 ENTITY RELATIONSHIP DIAGRAM 9](#_Toc140878887)

[3.3 DATA DICTIONARY 10](#_Toc140878888)

[Ch 4. NORMALIZATION 12](#_Toc140878889)

[Ch 5. PROGRAM CODE 20](#_Toc140878890)

[5.1 CODE DETAILS AND CODE EFFECIENCY 20](#_Toc140878891)

[Ch 6. RESULT AND DISCUSSION 42](#_Toc140878892)

[6.1 USER DOCUMENTATION 42](#_Toc140878893)

[Ch 7. TESTING 50](#_Toc140878894)

[7.1 Test Case 50](#_Toc140878895)

[Ch 8. CONCLUSION 52](#_Toc140878896)

[8.1 CONCLUSION 52](#_Toc140878897)

[8.2 FUTURE SCOPE 52](#_Toc140878898)

[8.3 REFERENCES 53](#_Toc140878899)

# Ch 1. INTRODUCTION

## OVERVIEW

This Hostel Management System is developed in favor of the hostel management team which helps them to save the records of the students about their rooms and other things. It helps them from the manual work from which it is very difficult to find the record of the students and the information about those ones who had left the hostel years before.

This solution is developed on the plight of the hostel management team, through this they cannot require so efficient person to handle and manage the affairs of the students in the hostel, all you need to do is to login as administrator and you can see the information of all the students who have obtained and registered their hostel form, click verify to ascertain their eligibility and allocate them to the available hostel.

Identification of the problems of the existing hostel management leads to the development of computerized solution that will be compatible to the existing hostel management with the solution which is more users friendly and more GUI oriented. We can improve the efficiency of the hostel management, thus overcome the drawbacks of the existing management.

## 1.2 OBJECTIVES

The main objectives of this project work are:

1. Student will no longer apply for their hostels manually.
2. The admin can see and access the student’s data.
3. This also will reduce the manual work of the persons in admin penal.
4. It helps the admin to know the number of students that can leave in a room..
5. To ensure that the number of student staying in a room does not exceed specified limit.
6. Help the admin to disseminate information to students without calling for a meeting or moving room to room.

## 1.3 ADMIN PANEL

### 1.3.1 Administrator login

* Enter the credentials (username and password) to login.

### 1.3.2 Admin Dashboard Home

* Admin Dashboard lands on a welcome page and shows no of students registered, no of rooms, and no of students allotted the hostel.
* Admin can register student, manage student’s records, book hostel for students, manage hostel students, and manage rooms.

### 1.3.3 Register Students

* Registration form where student’s details were to be filled.

### 1.3.4 Manage Students

* To manage the records of registered students and allow to edit and delete the records of students.

### 1.3.5 Book Hostel

* Book Hostel Form to allot hostel to the registered students by fetching the records of students by their registration number.

### 1.3.6 Hostel Students

* To manage the records of registered hostel students and allow editing and deleting the records of hostel allotted to the students.

### 1.3.7 Manage Rooms

* Here the rooms are managed by the room no.
* Can view, add, delete and modify the rooms.

## 1.4 STUDENT PANEL

### 1.4.1 Student login

* Enter the credentials (regnum and password) to login.

### 1.4.2 Student Dashboard Home

* A welcome page having About US.

### 1.4.3 Book Hostel

* Book Hostel Form to allot hostel to the logged in student.
* If already booked cannot book another one.

### 1.4.4 My Room Details

* Shows all the details records of student and hostel of a logged in student.

# Ch 2. REQUIREMENT AND ANALYSIS

## 2.1 HARWARE AND SOFTWARE CONFIGURATION

**Hardware Configuration**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

* Desktop PC or Laptop
* Printer
* Processor – Intel (R) core (TM) i5-6200U CPU @ 2.30 GHz 2.40 GHz
* RAM – 4.00 GB
* System Type – 64-bit Operating System , x64-based processor
* Display – 1024 x 768 monitor resolution
* Keyboard and Mouse

**Software Configuration**

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application

* Operating System – Windows 10
* Frontend – Html , CSS
* Backend – PHP
* Database – MySQL, PHP MyAdmin
* Software – Vs Code , XAMPP Control Panel

# Ch 3. DETAILED DESIGN

## 3.1 DATA FLOW DIAGRAM

DFD is an important tool used by system analysis. A data flow diagram model, a system using external entities from which data flows through a process which transforms the data and creates output data transforms which go to other processes external entities such as files. The main merit of DFD is that it can provide an overview of what data a system would process.

• A data-flow diagram is a way of representing a flow of data through a process or a system.

• The DFD also provides information about the outputs and inputs of each entity and the process itself.

CONTEXT LEVEL DFD

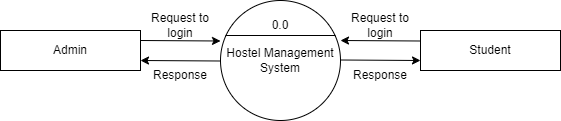
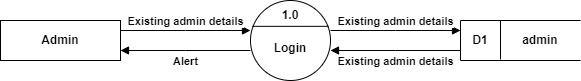
****

Figure 1: Context level DFD

LEVEL 1 DFD

**Admin**

****

****

****

****

****

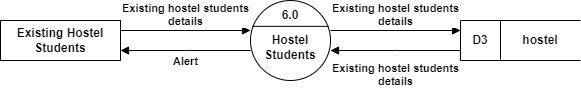
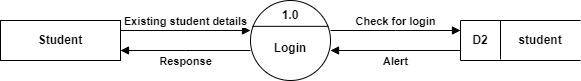
****

Figure 2: DFD Level 1 (Admin)

**Student**

****

****

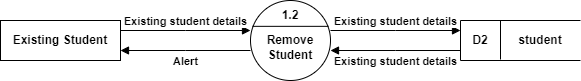
****

Figure 3: DFD Level 1 (Student)

LEVEL 2 DFD

**Admin**

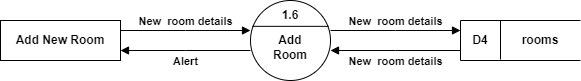
****

****

****

****









## 

Figure 4: DFD Level 2

## 3.2 ENTITY RELATIONSHIP DIAGRAM

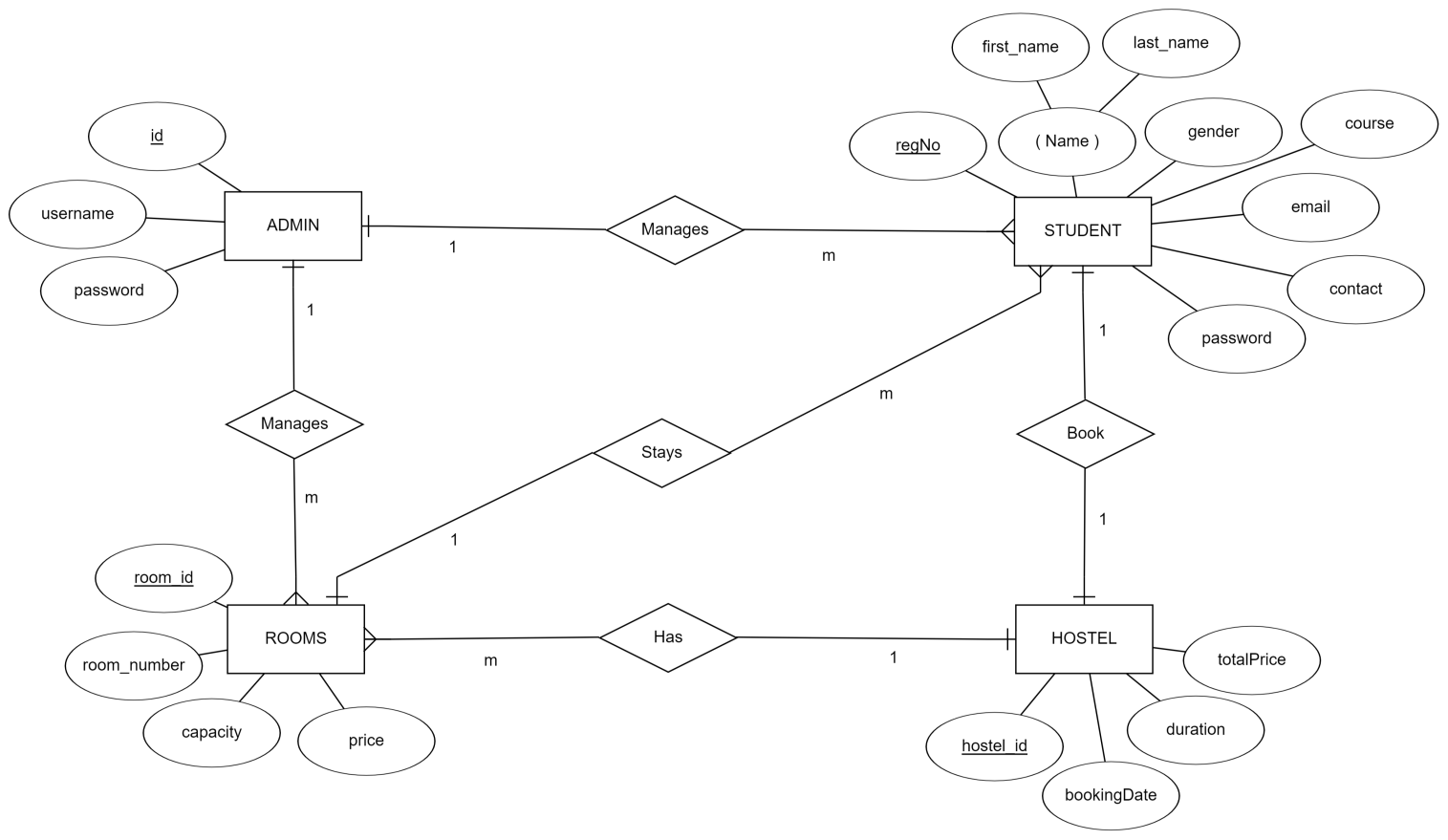
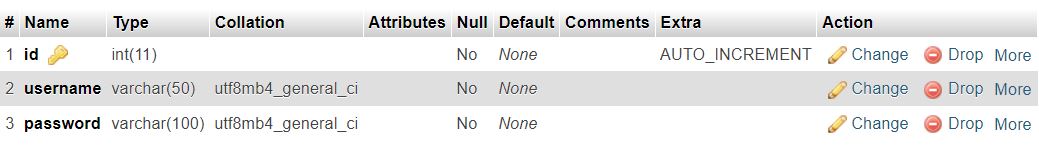
****

Figure 5: ER Diagram

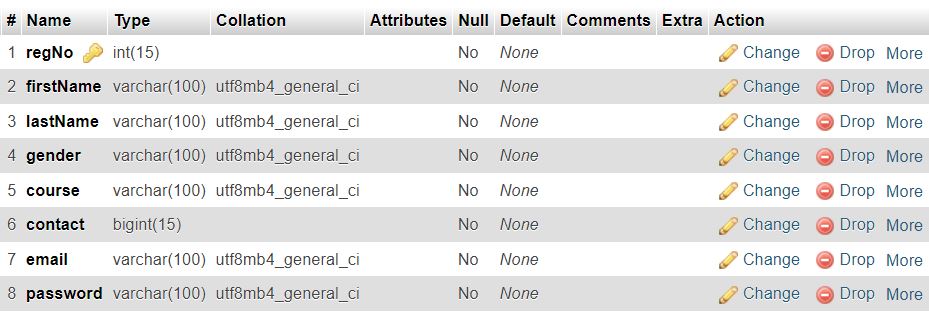
## 

## 3.3 DATA DICTIONARY

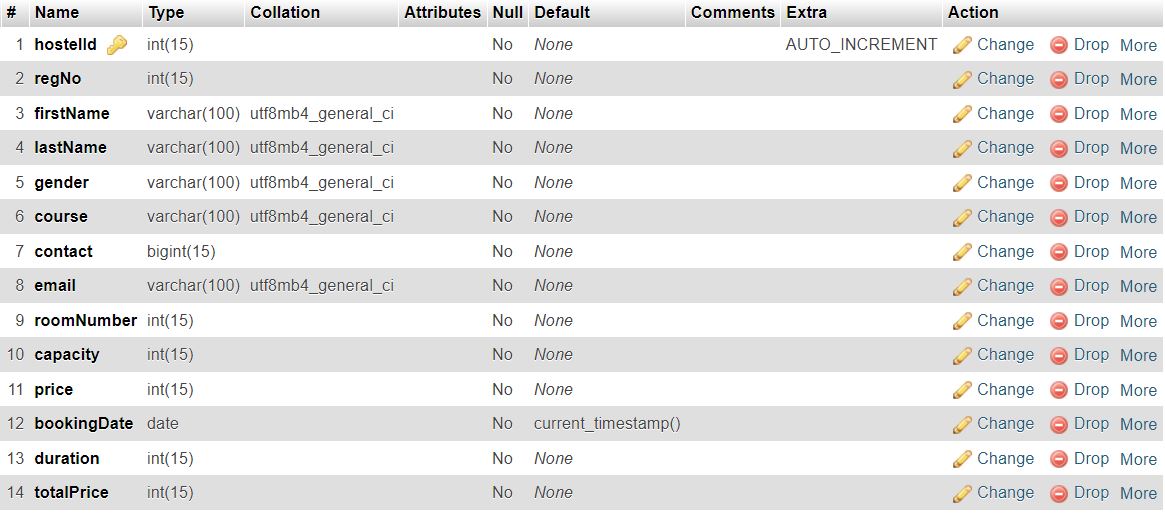
**Admin Table**

****

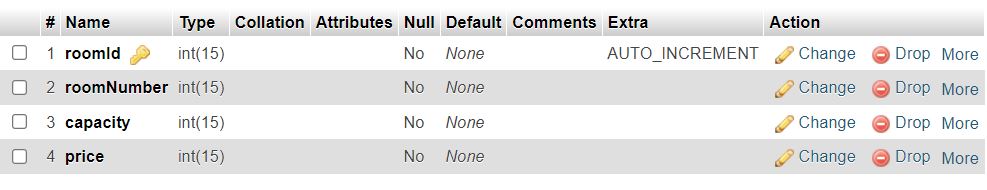
**Student Table**

****

**Hostel Table**

****

**Rooms Table**

****

# Ch 4. NORMALIZATION

**First Normal Form (1NF)**

Admin Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| id | int(11) | Primary Key NOTNULL |
| username | varchar(50) | NOT NULL |
| password | varchar(100) | NOT NULL |

Rooms Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| roomid | int(15) | Primary Key NOT NULL |
| roomNumber | int(15) | NOT NULL |
| capacity | int(15) | NOT NULL |
| price | int(15) | NOT NULL |

Student Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| regNo | int(15) | Primary Key NOTNULL |
| firstName | varchar(100) | NOT NULL |
| lastName | varchar(100) | NOT NULL |
| gender | varchar(100) | NOT NULL |
| course | varchar(100) | NOT NULL |
| contact | bigint(15) | NOT NULL |
| email | varchar(100) | NOT NULL |

Hostel Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| hostelid | int(15) | Primary Key NOTNULL |
| regNo | int(15) | NOTNULL |
| firstName | varchar(100) | NOT NULL |
| lastName | varchar(100) | NOT NULL |
| gender | varchar(100) | NOT NULL |
| course | varchar(100) | NOT NULL |
| contact | bigint(15) | NOT NULL |
| email | varchar(100) | NOT NULL |
| roomNumber | int(15) | NOT NULL |
| capacity | int(15) | NOT NULL |
| price | int(15) | NOT NULL |
| bookingDate | date | NOT NULL |
| duration | int(15) | NOT NULL |
| totalPrice | int(15) | NOT NULL |

**Second Normal Form (2NF)**

Rooms Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| roomid | int(15) | Primary Key NOT NULL |
| roomNumber | int(15) | NOT NULL |
| capacity | int(15) | NOT NULL |
| price | int(15) | NOT NULL |

Rooms Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| roomid | int(15) | Primary Key NOTNULL |
| roomNumber | int(15) | NOT NULL |

Room\_Details Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| roomid | int(15) | Foreign Key NOT NULL |
| capacity | int(15) | NOT NULL |
| price | int(15) | NOT NULL |

Student Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| regNo | int(15) | Primary Key NOTNULL |
| firstName | varchar(100) | NOT NULL |
| lastName | varchar(100) | NOT NULL |
| gender | varchar(100) | NOT NULL |
| course | varchar(100) | NOT NULL |

Student\_Contact Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| regNo | int(15) | Foreign Key NOT NULL |
| contact | bigint(15) | NOT NULL |
| email | varchar(100) | NOT NULL |

Hostel Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| hostelid | int(15) | Primary Key NOTNULL |
| regNo | int(15) | NOTNULL |
| firstName | varchar(100) | NOT NULL |
| lastName | varchar(100) | NOT NULL |
| gender | varchar(100) | NOT NULL |
| course | varchar(100) | NOT NULL |
| roomNumber | int(15) | NOT NULL |
| capacity | int(15) | NOT NULL |
| price | int(15) | NOT NULL |
| bookingDate | date | NOT NULL |
| duration | int(15) | NOT NULL |
| totalPrice | int(15) | NOT NULL |

Hostel \_Contact Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| regNo | int(15) | Foreign Key NOT NULL |
| contact | bigint(15) | NOT NULL |
| email | varchar(100) | NOT NULL |

**Third Normal Form (3NF)**

Rooms Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| roomid | int(15) | Primary Key NOTNULL |
| roomNumber | int(15) | NOT NULL |
| capacity | int(15) | NOT NULL |
| price | int(15) | NOT NULL |

Rooms Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| roomid | int(15) | Primary Key NOTNULL |
| roomNumber | int(15) | NOT NULL |

Rooms\_Capacity Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| roomid | int(15) | Foreign Key NOT NULL |
| capacity | int(15) | NOT NULL |

Rooms\_Price Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| roomid | int(15) | Foreign Key NOT NULL |
| price | int(15) | NOT NULL |

Student Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| regNo | int(15) | Primary Key NOTNULL |
| firstName | varchar(100) | NOT NULL |
| lastName | varchar(100) | NOT NULL |
| gender | varchar(100) | NOT NULL |
| course | varchar(100) | NOT NULL |

Student\_Contact Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| regNo | int(15) | Foreign Key NOT NULL |
| contact | bigint(15) | NOT NULL |

Student\_Email Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| regNo | int(15) | Foreign Key NOT NULL |
| email | varchar(100) | NOT NULL |

Hostel Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| hostelid | int(15) | Primary Key NOTNULL |
| regNo | int(15) | NOTNULL |
| firstName | varchar(100) | NOT NULL |
| lastName | varchar(100) | NOT NULL |
| gender | varchar(100) | NOT NULL |
| course | varchar(100) | NOT NULL |
| roomNumber | int(15) | NOT NULL |
| capacity | int(15) | NOT NULL |
| price | int(15) | NOT NULL |
| bookingDate | date | NOT NULL |
| duration | int(15) | NOT NULL |

Hostel\_Contact Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| regNo | int(15) | Foreign Key NOT NULL |
| contact | bigint(15) | NOT NULL |

Hostel\_Price Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Key Constraints** |
| regNo | int(15) | Foreign Key NOT NULL |
| totalPrice | int(15) | NOT NULL |

# Ch 5. PROGRAM CODE

## 5.1 CODE DETAILS AND CODE EFFECIENCY

**Index.php**

<?php

session\_start();

if (isset($\_SESSION['admin\_id'])) {

header("Location: admin\_dashboard.php");

exit();

}

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

// Database connection credentials

$db\_host = 'localhost';

$db\_user = 'root';

$db\_pass = '';

$db\_name = 'hostel\_manage';

// Connect to the database

$conn = mysqli\_connect($db\_host, $db\_user, $db\_pass, $db\_name);

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

// Get the entered username and password

$username = $\_POST['username'];

$password = $\_POST['password'];

// Query to check admin credentials

$sql = "SELECT \* FROM admin\_t WHERE username='$username' AND password='$password'";

$result = mysqli\_query($conn, $sql);

if (mysqli\_num\_rows($result) === 1) {

// Admin login success

$admin = mysqli\_fetch\_assoc($result);

$\_SESSION['admin\_id'] = $admin['id'];

header("Location: admin\_dashboard.php");

exit();

} else {

$error\_message = "Invalid username or password";

}

mysqli\_close($conn);

}

?>

<!DOCTYPE html>

<html>

<head></head>

<body>

<div class="container">

<h1>

<i class="fas fa-home"></i> HOSTEL MANAGEMENT SYSTEM

</h1>

<div class="logo">

<i class="fas fa-user-shield"></i>

<!-- Update with the desired admin logo icon -->

</div>

<h2>

<i class="fas fa-user-shield"></i>Admin Login

</h2>

<form method="POST" action="

<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>">

<div class="card">

<i class="fas fa-user"></i>

<input type="text" id="admin-username" name="username" placeholder="Username" required>

</div>

<div class="card">

<i class="fas fa-lock"></i>

<input type="password" id="admin-password" name="password" placeholder="Password" required>

</div>

<input type="submit" value="Login">

</form>

<div class="footer">

<h3>Student Login Panel</h3>

<p>If you are an Student, please <a href="student\_login.php">click here</a> to login. </p>

</div> <?php

if (isset($error\_message)) {

echo '<p class="error">' . $error\_message . '</p>';

}

?>

</div>

</body>

</html>

**student\_login.php**

<?php

include 'db.php';

include 'check\_login.php';

if (isset($\_SESSION['regNo'])) {

header("Location: student\_dashboard.php");

exit;

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Student Login</title>

</head>

<body>

<div class="container">

<h1>

<i class="fas fa-home"></i> HOSTEL MANAGEMENT SYSTEM

</h1>

<div class="logo">

<i class="fas fa-graduation-cap"></i>

<!-- Update with your desired logo icon -->

</div>

<h2>Student Login</h2>

<form method="post" action="">

<div class="card">

<i class="fas fa-user"></i>

<label for="regNo">Registration Number:</label>

<input type="text" name="regNo" id="regNo" placeholder="Enter your registration number" required>

</div>

<div class="card">

<i class="fas fa-lock"></i>

<label for="password">Password:</label>

<input type="password" name="password" id="password" placeholder="Enter your password" required>

</div>

<input type="submit" name="login" value="Login">

</form>

<div class="footer">

<h3>Admin Login Panel</h3>

<p>If you are an Admin, please <a href="index.php">click here</a> to login. </p>

</div> <?php

if (isset($\_SESSION['login\_errors'])) {

$errors = $\_SESSION['login\_errors'];

foreach ($errors as $error) {

echo '

<p class="error">' . $error . '</p>';

}

unset($\_SESSION['login\_errors']);

}

?>

</div>

</body>

</html>

**register\_student.php**

<!DOCTYPE html>

<html>

<head>

<title>Register Student</title>

</head>

<body>

<h2><i class="fas fa-user-plus"></i> REGISTER STUDENT</h2>

<?php include 'db.php'; ?>

<form method="post" action="">

<div class="card">

<label for="regNo"><i class="fas fa-id-card"></i> Registration Number:</label>

<input type="text" name="regNo" id="regNo" required placeholder="Enter registration number">

</div>

<div class="card">

<label for="firstName"><i class="fas fa-user"></i> First Name:</label>

<input type="text" name="firstName" id="firstName" required placeholder="Enter first name">

</div>

<div class="card">

<label for="lastName"><i class="fas fa-user"></i> Last Name:</label>

<input type="text" name="lastName" id="lastName" required placeholder="Enter last name">

</div>

<div class="card">

<label for="gender"><i class="fas fa-venus-mars"></i> Gender:</label>

<select name="gender" id="gender" required>

<option value="">Select Gender</option>

<option value="male">Male</option>

<option value="female">Female</option>

<option value="other">Other</option>

</select>

</div>

<div class="card">

<label for="course"><i class="fas fa-book"></i> Course:</label>

<input type="text" name="course" id="course" required placeholder="Enter course">

</div>

<div class="card">

<label for="contact"><i class="fas fa-phone"></i> Contact Number:</label>

<input type="text" name="contact" maxlength="10" id="contact" required placeholder="Enter contact number">

</div>

<div class="card">

<label for="email"><i class="fas fa-envelope"></i> Email ID:</label>

<input type="email" name="email" id="email" required placeholder="Enter email ID">

</div>

<div class="card">

<label for="password"><i class="fas fa-lock"></i> Password:</label>

<input type="password" name="password" id="password" required placeholder="Enter password">

</div>

<div class="card">

<label for="confirmPassword"><i class="fas fa-lock"></i> Confirm Password:</label>

<input type="password" name="confirmPassword" id="confirmPassword" required placeholder="Confirm password">

</div>

<input type="submit" name="register" value="Register">

<input type="reset" value="Reset">

<a class="dashboard-button" href="admin\_dashboard.php">Back to Dashboard</a>

<?php

if (isset($\_POST['register'])) {

$regNo = $\_POST['regNo'];

$firstName = $\_POST['firstName'];

$lastName = $\_POST['lastName'];

$gender = $\_POST['gender'];

$course = $\_POST['course'];

$contact = $\_POST['contact'];

$email = $\_POST['email'];

$password = $\_POST['password'];

$confirmPassword = $\_POST['confirmPassword'];

$errors = array();

if ($password !== $confirmPassword) {

$errors[] = "Passwords do not match.";

}

// Check for other validations and add errors to the $errors array

// If there are any errors, display them

if (!empty($errors)) {

echo '<div class="error-message">';

foreach ($errors as $error) {

echo '<p>' . $error . '</p>';

}

echo '</div>';

} else {

// SQL query to insert student data into the student table

$sql = "INSERT INTO student (regNo, firstName, lastName, gender, course, contact, email, password)

VALUES ('$regNo', '$firstName', '$lastName', '$gender', '$course','$contact', '$email', '$password')";

if ($conn->query($sql) === TRUE) {

echo "<b><p>Registration successful!</p></b>";

echo "<p>Registration Number: " . $regNo . "</p>";

echo "<p>Name: " . $firstName . " " . $lastName . "</p>";

echo "<p>Gender: " . $gender . "</p>";

echo "<p>Course: " . $course . "</p>";

echo "<p>Contact: " . $contact . "</p>";

echo "<p>Email: " . $email . "</p>";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

}

}

// Close the database connection

$conn->close();

?>

</form>

</body>

</html>

**Manage\_student.php**

<!DOCTYPE html>

<html>

<head>

<title>Manage Students</title>

</head>

<body>

<div class="container">

<h2><i class="fas fa-users"></i> MANAGE STUDENTS</h2>

<a class="dashboard-button" href="admin\_dashboard.php">Back to Dashboard</a>

<div class="search-form">

<input type="text" class="search-input" id="searchInput" placeholder="Search by Registration Number, Name, Course..." autocomplete="off" />

</div>

<table>

<tr>

<th><i class="fas fa-id-card"></i> Registration Number</th>

<th><i class="fas fa-user"></i> Name</th>

<th><i class="fas fa-venus-mars"></i> Gender</th>

<th><i class="fas fa-book"></i> Course</th>

<th><i class="fas fa-phone"></i> Contact Number</th>

<th><i class="fas fa-envelope"></i> Email ID</th>

<th><i class="fas fa-cog"></i> Action</th>

</tr>

<!-- PHP code to fetch and display student data -->

<?php

// Include the database connection file

include 'db.php';

// SQL query to retrieve student data from the database

$sql = "SELECT \* FROM student";

$result = $conn->query($sql); if ($result->num\_rows > 0) { // Loop through each row of the result set while ($row = $result->fetch\_assoc()) { $regNo = $row['regNo']; $firstName = $row['firstName']; $lastName = $row['lastName'];

$gender = $row['gender']; $course = $row['course']; $contact = $row['contact']; $email = $row['email']; // Display the student data in the table echo '

<tr>

'; echo '

<td>' . $regNo . '</td>

'; echo '

<td>' . $firstName . ' ' . $lastName . '</td>

'; echo '

<td>' . $gender . '</td>

'; echo '

<td>' . $course . '</td>

'; echo '

<td>' . $contact . '</td>

'; echo '

<td>' . $email . '</td>

'; echo '

<td>

'; echo '<a href="edit\_student.php?regNo=' . $regNo . '" class="edit-button"><i class="fas fa-edit"></i> Edit</a>'; echo '

<a href="delete\_student.php?regNo=' . $regNo . '" class="delete-button"><i class="fas fa-trash"></i> Delete</a>'; echo '

</td>

'; echo '

</tr>

'; } } else { echo '

<tr>

<td colspan="7">No students found.</td>

</tr>

'; }

$conn->close(); ?>

</table>

</div>

<script>

// JavaScript code to handle search functionality

document.addEventListener("DOMContentLoaded", function () {

const searchInput = document.getElementById("searchInput");

const table = document.querySelector("table");

searchInput.addEventListener("input", handleSearch);

function handleSearch() {

const searchKeyword = searchInput.value.toLowerCase();

const rows = table.getElementsByTagName("tr");

for (let i = 1; i < rows.length; i++) {

const cells = rows[i].getElementsByTagName("td");

let foundMatch = false;

for (let j = 0; j < cells.length; j++) {

const cellText = cells[j].textContent.toLowerCase();

if (cellText.includes(searchKeyword)) {

foundMatch = true;

break;

}

}

if (foundMatch) {

rows[i].style.display = "";

} else {

rows[i].style.display = "none";

}

}

}

});

</script>

</body>

</html>

**Manage\_room.php**

<!DOCTYPE html>

<html>

<head>

<title>Room Management</title>

</head>

<body>

<div class="container">

<h1>

<center><i class="fas fa-door-open"></i> ROOM MANAGEMENT</center>

</h1>

<br />

<a class="add-button" href="add\_room.php"><i class="fas fa-plus-circle"></i> Add New Room</a>

<a class="dashboard-button" href="admin\_dashboard.php"><i class="fas fa-arrow-left"></i> Back to Dashboard</a>

<br />

<br />

<?php

// Include database connection file

include 'db.php';

// Fetch room records from the database

$sql = "SELECT \* FROM rooms";

$result = $conn->query($sql); if ($result->num\_rows > 0) { echo '

<table>

'; echo '

<tr>

<th><i class="fas fa-id-badge"></i> Room ID</th>

<th><i class="fas fa-door-closed"></i> Room Number</th>

<th><i class="fas fa-users"></i> Capacity</th>

<th><i class="fas fa-dollar-sign"></i> Price</th>

<th><i class="fas fa-cogs"></i> Actions</th>

</tr>

'; while ($row = $result->fetch\_assoc()) { echo '

<tr>

'; echo '

<td>' . $row['roomId'] . '</td>

'; echo '

<td>' . $row['roomNumber'] . '</td>

'; echo '

<td>' . $row['capacity'] . '</td>

'; echo '

<td>' . $row['price'] . '</td>

'; echo '

<td>

<a class="edit-button" href="edit\_room.php?id=' . $row['roomId'] . '"><i class="fas fa-edit"></i> Edit</a>

<a class="delete-button" href="delete\_room.php?id=' . $row['roomId'] . '"><i class="fas fa-trash-alt"></i> Delete</a>

</td>

'; echo '

</tr>

'; } echo '

</table>

'; } else { echo '

<p class="no-rooms"><i class="fas fa-exclamation-circle"></i> No rooms found.</p>

'; }

connection $conn->close(); ?>

</div>

</body>

</html>

**Book\_hostel.php**

<?php

session\_start();

if (isset($\_SESSION['admin\_id'])) {

$userRole = 'admin'; // Set the user role as 'admin' if the 'admin\_id' session variable is set

} else if (isset($\_SESSION['regNo'])) {

$userRole = 'student'; // Set the user role as 'student' if the 'regNo' session variable is set

} else {

header('Location: login.php'); // Redirect to the login page

exit; // Terminate the current script

}

$dashboardURL = ($userRole === 'admin') ? 'admin\_dashboard.php' : 'student\_dashboard.php';

include 'db.php'; // Include the database connection file

// Fetch student details if regNo session variable is set

if ($userRole === 'student' && isset($\_SESSION['regNo'])) {

$regNo = $\_SESSION['regNo'];

// Fetch the student details from the hostel table

$sql = "SELECT \* FROM hostel WHERE regNo = '$regNo'";

$result = $conn->query($sql);

$studentData = $result->fetch\_assoc();

// Fill in the form fields with the fetched data

if ($studentData) {

$hostelId = $studentData['hostelId'];

$firstName = $studentData['firstName'];

$lastName = $studentData['lastName'];

$gender = $studentData['gender'];

$course = $studentData['course'];

$contact = $studentData['contact'];

$email = $studentData['email'];

$roomNumber = $studentData['roomNumber'];

$capacity = $studentData['capacity'];

$price = $studentData['price'];

$bookingDate = $studentData['bookingDate'];

$duration = $studentData['duration'];

$totalPrice = $studentData['totalPrice'];

}

}

// Restrict booking if the user has already made a booking

$message = '';

if ($userRole === 'student' && isset($\_SESSION['regNo'])) {

$regNo = $\_SESSION['regNo'];

// Check if the student has already made a booking

$sql = "SELECT COUNT(\*) AS count FROM hostel WHERE regNo = '$regNo'";

$result = $conn->query($sql);

$row = $result->fetch\_assoc();

$bookingCount = $row['count'];

if ($bookingCount > 0) {

$message = "You have already made a booking.";

}

}

// Handle form submission

if (isset($\_POST['bookHostel'])) {

if (!empty($message)) {

// Display the message if the user has already made a booking

$message = "You have already made a booking.";

} else {

// Retrieve form data

$regNo = $\_POST['regNo'];

$firstName = $\_POST['firstName'];

$lastName = $\_POST['lastName'];

$gender = $\_POST['gender'];

$course = $\_POST['course'];

$contact = $\_POST['contact'];

$email = $\_POST['email'];

$roomNumber = $\_POST['roomNumber'];

$capacity = $\_POST['capacity'];

$price = $\_POST['price'];

$hostelId = $\_POST['hostelId'];

$bookingDate = $\_POST['bookingDate'];

$duration = $\_POST['duration'];

$totalPrice = $\_POST['totalPrice'];

// Check if the student has already made a booking

$sql = "SELECT COUNT(\*) AS count FROM hostel WHERE regNo = '$regNo'";

$result = $conn->query($sql);

$row = $result->fetch\_assoc();

$bookingCount = $row['count'];

if ($bookingCount > 0) {

$message = "You have already made a booking.";

} else {

// Perform validations

$errors = array();

// Validate booking date

if (empty($bookingDate)) {

$errors[] = "Please select a booking date.";

}

// Validate duration

if (empty($duration)) {

$errors[] = "Please select the duration.";

}

// Convert duration to months if needed

if (!empty($duration)) {

$durationMonths = intval($duration);

if ($durationMonths <= 0) {

$errors[] = "Invalid duration.";

}

}

// Check if the selected room has available capacity

$sql = "SELECT COUNT(\*) AS count FROM hostel WHERE roomNumber = '$roomNumber'";

$result = $conn->query($sql);

$row = $result->fetch\_assoc();

$bookedCount = $row['count'];

if ($bookedCount >= $capacity) {

$errors[] = "The selected room is fully booked. Please choose another room.";

}

// If there are no errors, insert the booking data into the hostel table

if (empty($errors)) {

$sql = "INSERT INTO hostel (hostelId, regNo, firstName, lastName, gender, course, contact, email, roomNumber, capacity, price, bookingDate, duration, totalPrice)

VALUES ('$hostelId', '$regNo', '$firstName', '$lastName', '$gender', '$course', '$contact', '$email', '$roomNumber', '$capacity', '$price', '$bookingDate', '$duration', '$totalPrice')";

if ($conn->query($sql) === TRUE) {

$message = "Hostel booking successful!";

// Display the booking details

} else {

$message = "Error: " . $sql . "<br>" . $conn->error;

}

} else {

// Display validation errors

$message = implode('<br>', $errors);

}

}

}

}

// Close the database connection

$conn->close();

?>

<!DOCTYPE html>

<html>

<head>

<title>Book Hostel</title>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script>

$(document).ready(function() {

// Fetch student details based on registration number

$('#regNo').change(function() {

var regNo = $(this).val();

$.ajax({

url: 'fetch\_student.php',

type: 'POST',

data: { regNo: regNo },

dataType: 'json',

success: function(response) {

$('#firstName').val(response.firstName);

$('#lastName').val(response.lastName);

$('#gender').val(response.gender);

$('#course').val(response.course);

$('#contact').val(response.contact);

$('#email').val(response.email);

}

});

});

// Fetch room details based on room number

$('#roomNumber').change(function() {

var roomNumber = $(this).val();

$.ajax({

url: 'fetch\_room.php',

type: 'POST',

data: { roomNumber: roomNumber },

dataType: 'json',

success: function(response) {

$('#capacity').val(response.capacity);

$('#price').val(response.price);

}

});

});

// Calculate the price based on the selected duration

$('#duration').change(function() {

calculatePrice(); // Call the calculatePrice function

});

// Function to calculate the price based on duration and original price

function calculatePrice() {

var duration = parseInt($('#duration').val());

var price = parseFloat($('#price').val());

var totalPrice = duration \* price;

$('#totalPrice').val(totalPrice.toFixed(2));

}

});

</script>

</head>

<?php

if (!isset($firstName)) {

$firstName = '';

}

if (!isset($lastName)) {

$lastName = '';

}

if (!isset($gender)) {

$gender = '';

}

if (!isset($course)) {

$course = '';

}

if (!isset($contact)) {

$contact = '';

}

if (!isset($email)) {

$email = '';

}

if (!isset($roomNumber)) {

$roomNumber = '';

}

if (!isset($capacity)) {

$capacity = '';

}

if (!isset($price)) {

$price = '';

}

if (!isset($bookingDate)) {

$bookingDate = '';

}

if (!isset($duration)) {

$duration = '';

}

if (!isset($totalPrice)) {

$totalPrice = '';

}

?>

<body>

<h2><i class="fas fa-hotel"></i> BOOK HOSTEL</h2>

<form method="post" action="">

<?php if ($userRole === 'student') { ?>

<input type="hidden" name="regNo" value="<?php echo $\_SESSION['regNo']; ?>">

<?php } ?>

<?php if (!empty($message)): ?>

<div class="message">

<?php echo $message; ?>

</div>

<?php endif; ?>

<div class="card">

<label for="hostelId"><i class="fas fa-id-card"></i> Hostel Id:</label>

<input type="text" name="hostelId" id="hostelId" required placeholder="Enter hostel ID">

</div>

<div class="card">

<label for="regNo"><i class="fas fa-id-badge"></i> Registration Number:</label>

<input type="text" name="regNo" id="regNo" required placeholder="Enter registration number" <?php if ($userRole === 'student') { echo 'value="' . $\_SESSION['regNo'] . '" readonly'; } ?>>

</div>

<div class="card">

<label for="firstName"><i class="fas fa-user"></i> First Name:</label>

<input type="text" name="firstName" id="firstName" required placeholder="Enter first name" <?php if ($userRole === 'student') { echo 'value="' . $firstName . '"'; } ?>>

</div>

<div class="card">

<label for="lastName"><i class="fas fa-user"></i> Last Name:</label>

<input type="text" name="lastName" id="lastName" required placeholder="Enter last name" <?php if ($userRole === 'student') { echo 'value="' . $lastName . '"'; } ?>>

</div>

<div class="card">

<label for="gender"><i class="fas fa-venus-mars"></i> Gender:</label>

<select name="gender" id="gender" required>

<option value="">Select Gender</option>

<option value="male" <?php if ($userRole === 'student' && $gender === 'male') { echo 'selected'; } ?>>Male</option>

<option value="female" <?php if ($userRole === 'student' && $gender === 'female') { echo 'selected'; } ?>>Female</option>

<option value="other" <?php if ($userRole === 'student' && $gender === 'other') { echo 'selected'; } ?>>Other</option>

</select>

</div>

<div class="card">

<label for="course"><i class="fas fa-book"></i> Course:</label>

<input type="text" name="course" id="course" required placeholder="Enter course" <?php if ($userRole === 'student') { echo 'value="' . $course . '"'; } ?>>

</div>

<div class="card">

<label for="contact"><i class="fas fa-phone"></i> Contact Number:</label>

<input type="text" name="contact" id="contact" required placeholder="Enter contact number" <?php if ($userRole === 'student') { echo 'value="' . $contact . '"'; } ?>>

</div>

<div class="card">

<label for="email"><i class="fas fa-envelope"></i> Email ID:</label>

<input type="email" name="email" id="email" required placeholder="Enter email ID" <?php if ($userRole === 'student') { echo 'value="' . $email . '"'; } ?>>

</div>

<div class="card">

<label for="roomNumber"><i class="fas fa-door-open"></i> Room Number:</label>

<input type="text" id="roomNumber" name="roomNumber" required placeholder="Enter room number">

</div>

<div class="card">

<label for="capacity"><i class="fas fa-users"></i> Capacity:</label>

<input type="text" id="capacity" name="capacity" required placeholder="Enter capacity">

</div>

<div class="card">

<label for="price"><i class="fas fa-dollar-sign"></i> Price:</label>

<input type="text" id="price" name="price" required placeholder="Enter price" readonly>

</div>

<div class="card">

<label for="bookingDate"><i class="far fa-calendar-alt"></i> Booking Date:</label>

<input type="date" name="bookingDate" id="bookingDate" required>

</div>

<div class="card">

<label for="duration"><i class="fas fa-clock"></i> Duration (in months):</label>

<select name="duration" id="duration" required>

<option value="">Select Duration</option>

<option value="1">1 Month</option>

<option value="2">2 Months</option>

<option value="3">3 Months</option>

<option value="4">4 Months</option>

<option value="5">5 Months</option>

<option value="6">6 Months</option>

<option value="7">7 Months</option>

<option value="8">8 Months</option>

<option value="9">9 Months</option>

<option value="10">10 Months</option>

<option value="11">11 Months</option>

<option value="12">12 Months</option>

</select>

</div>

<div class="card">

<label for="totalPrice"><i class="fas fa-dollar-sign"></i> Total Price:</label>

<input type="text" id="totalPrice" name="totalPrice" readonly>

</div>

<input type="submit" name="bookHostel" value="Book Hostel">

<input type="reset" value="Reset">

<a class="dashboard-button" href="<?php echo $dashboardURL; ?>"> Back to Dashboard</a>

</form>

</body>

</html>

**Db.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "hostel\_manage";

// Create a database connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) { die("Connection failed: " . $conn->connect\_error); }?>

**Add\_room.php**

<!DOCTYPE html>

<html>

<head>

<?php

include 'db.php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$roomId = $\_POST['roomId'];

$roomNumber = $\_POST['roomNumber'];

$capacity = $\_POST['capacity'];

$price = $\_POST['price'];

$sql = "INSERT INTO rooms (roomId, roomNumber, capacity, price) VALUES ('$roomId', '$roomNumber', '$capacity', '$price')";

if ($conn->query($sql) === TRUE) {

echo '<p><center>Room added successfully.</center></p>';

} else {

echo '<p>Error: ' . $conn->error . '</p>';

}

$conn->close();

}

?>

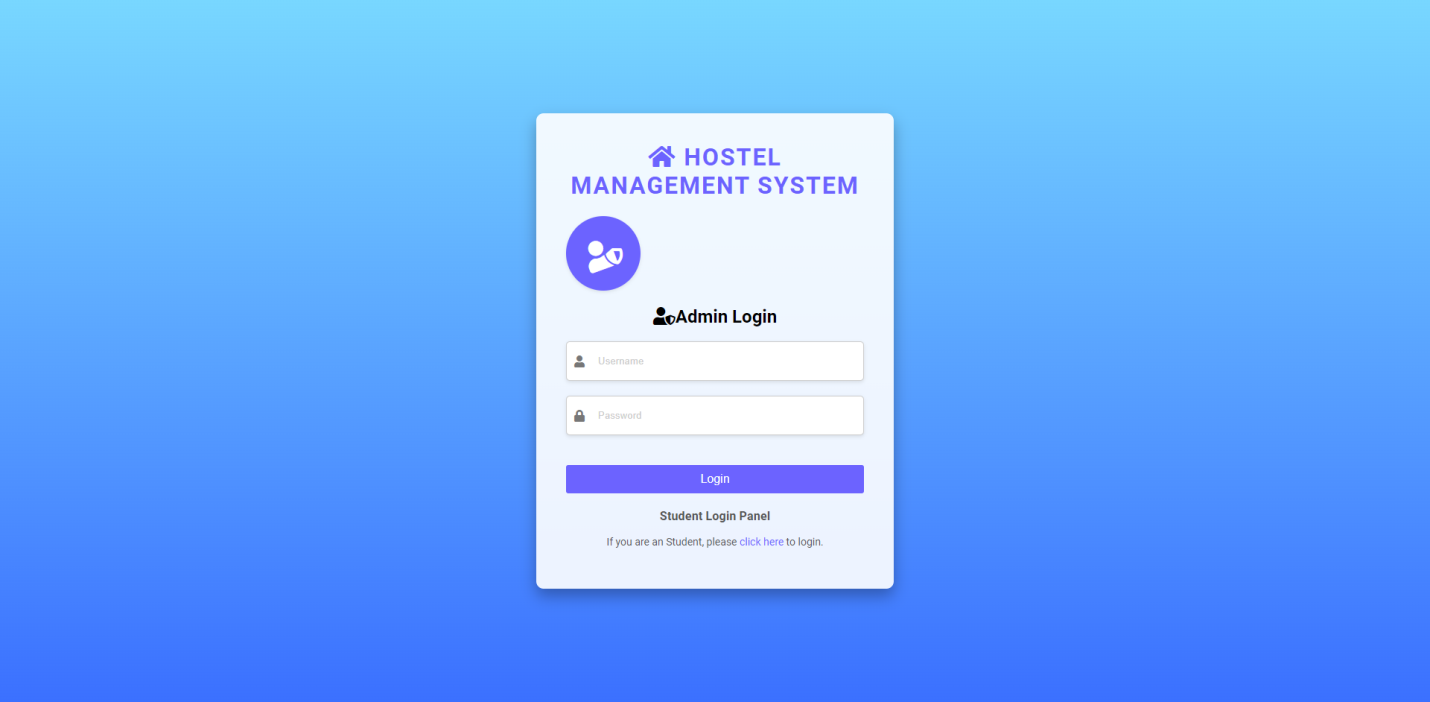
</body>

</html>

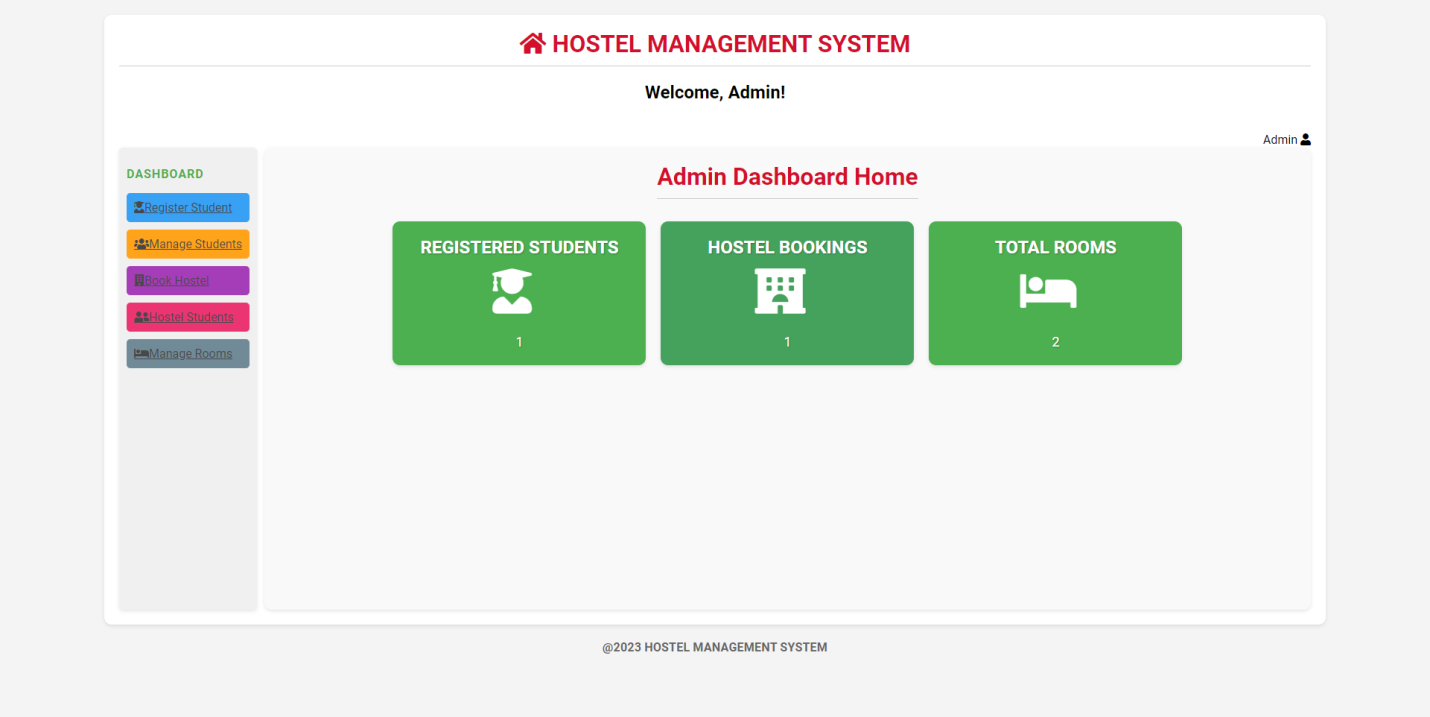
# Ch 6. RESULT AND DISCUSSION

## 6.1 USER DOCUMENTATION

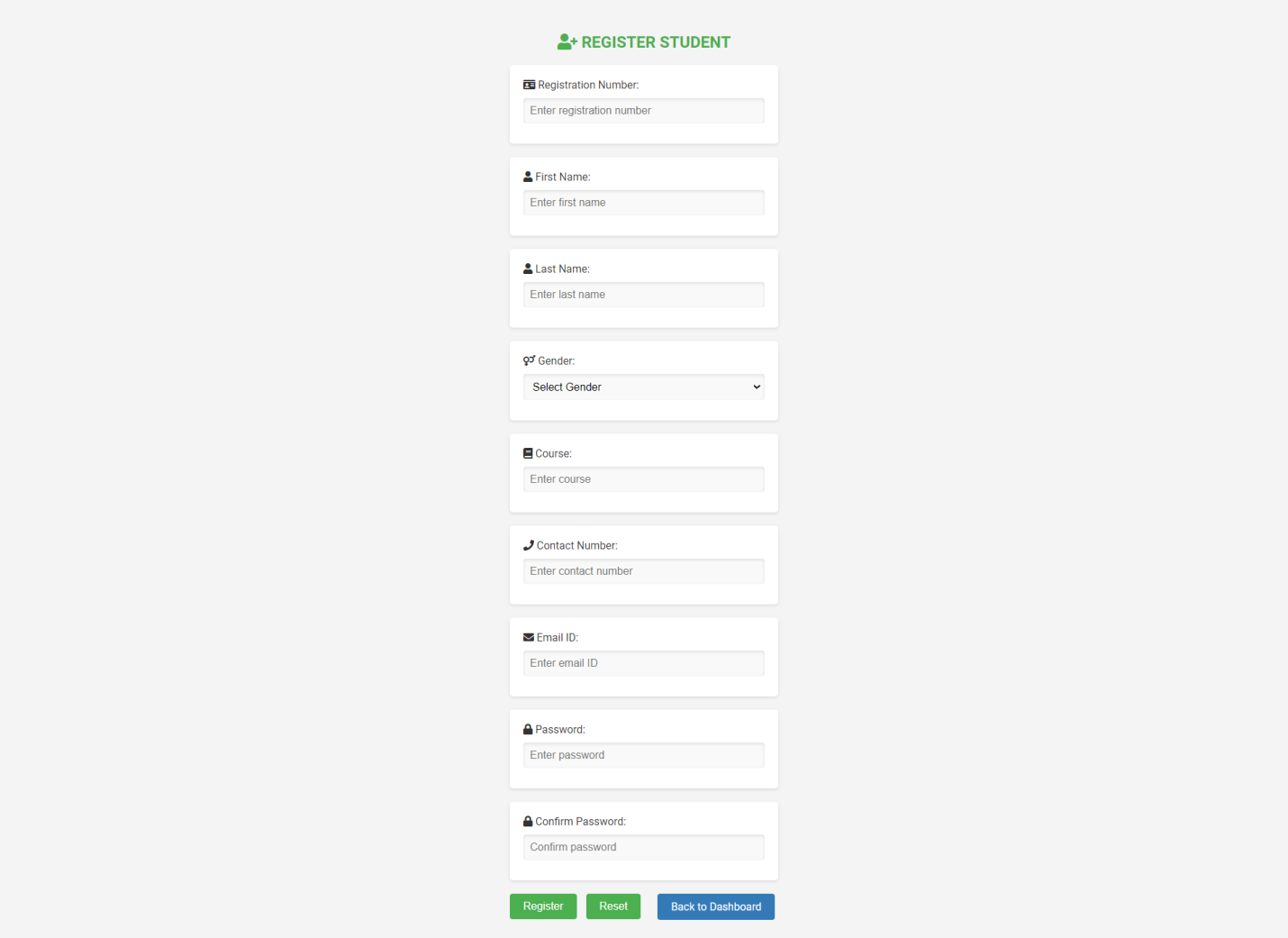
1. **Admin Login Panel**

****

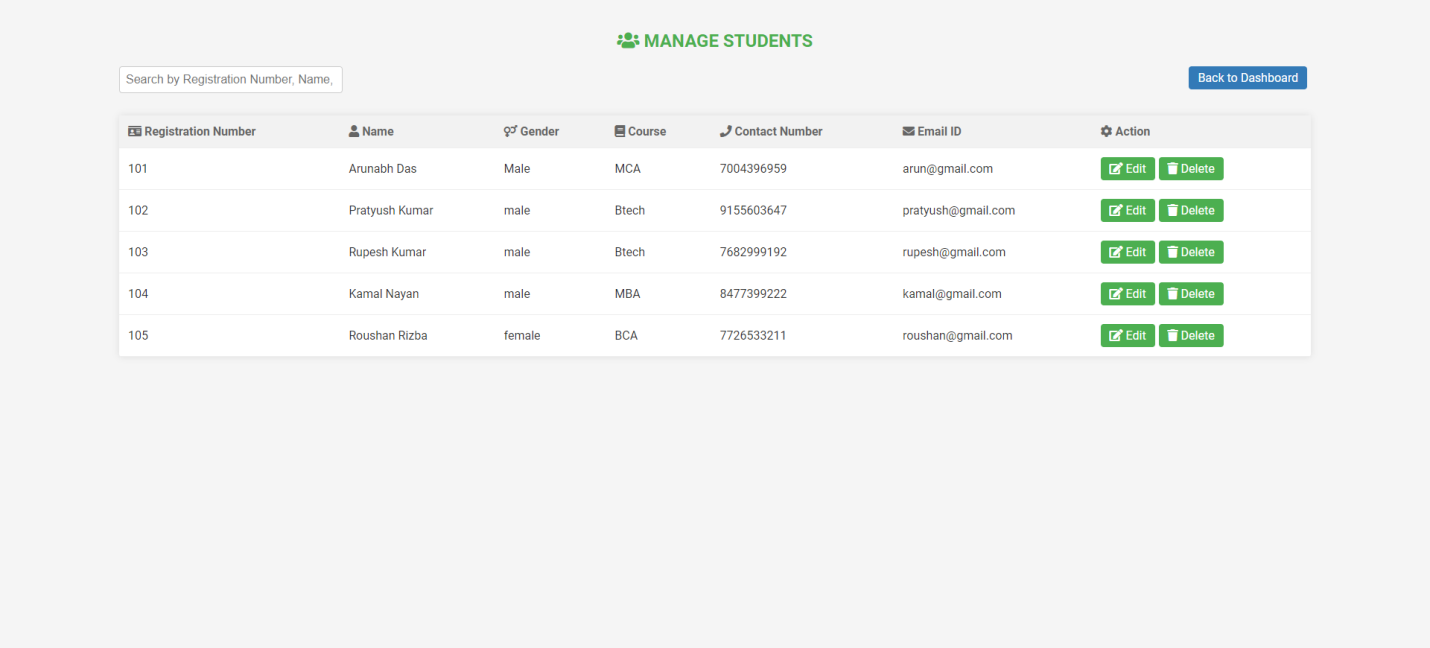
1. **Admin Dashboard Home**

****

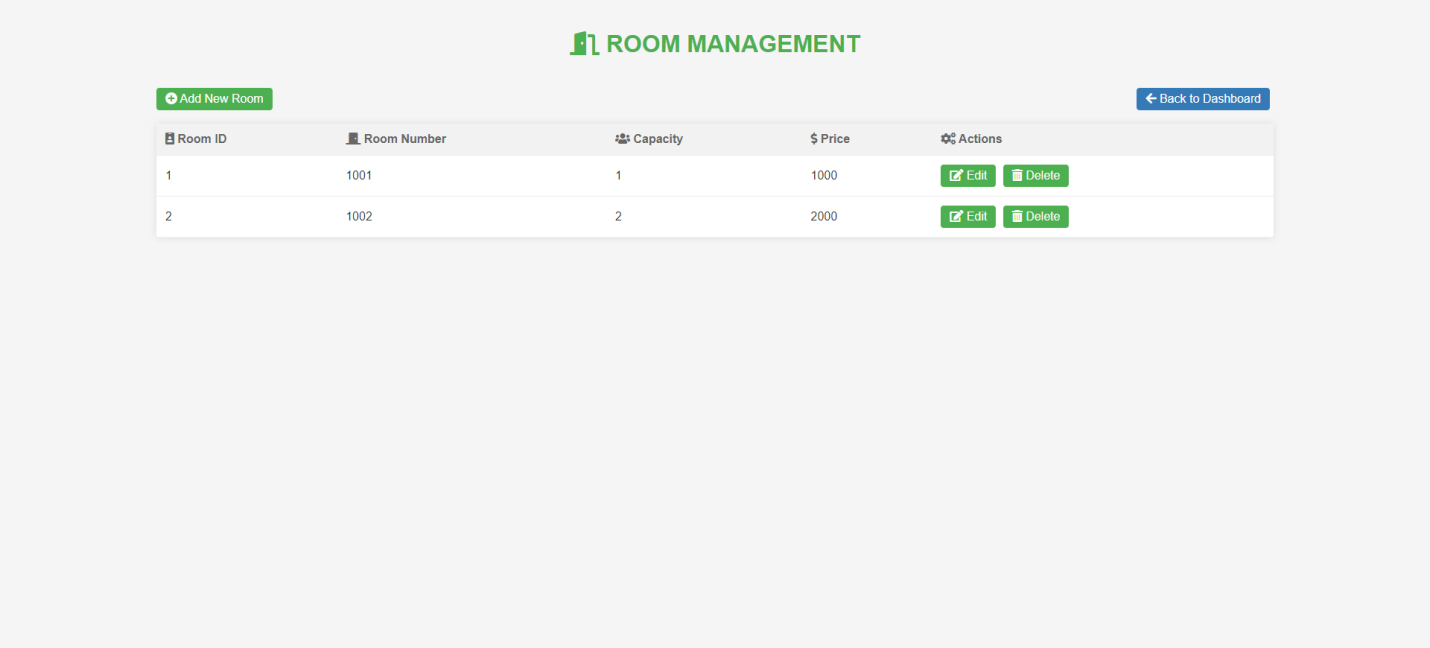
1. **Student Registration Form**

****

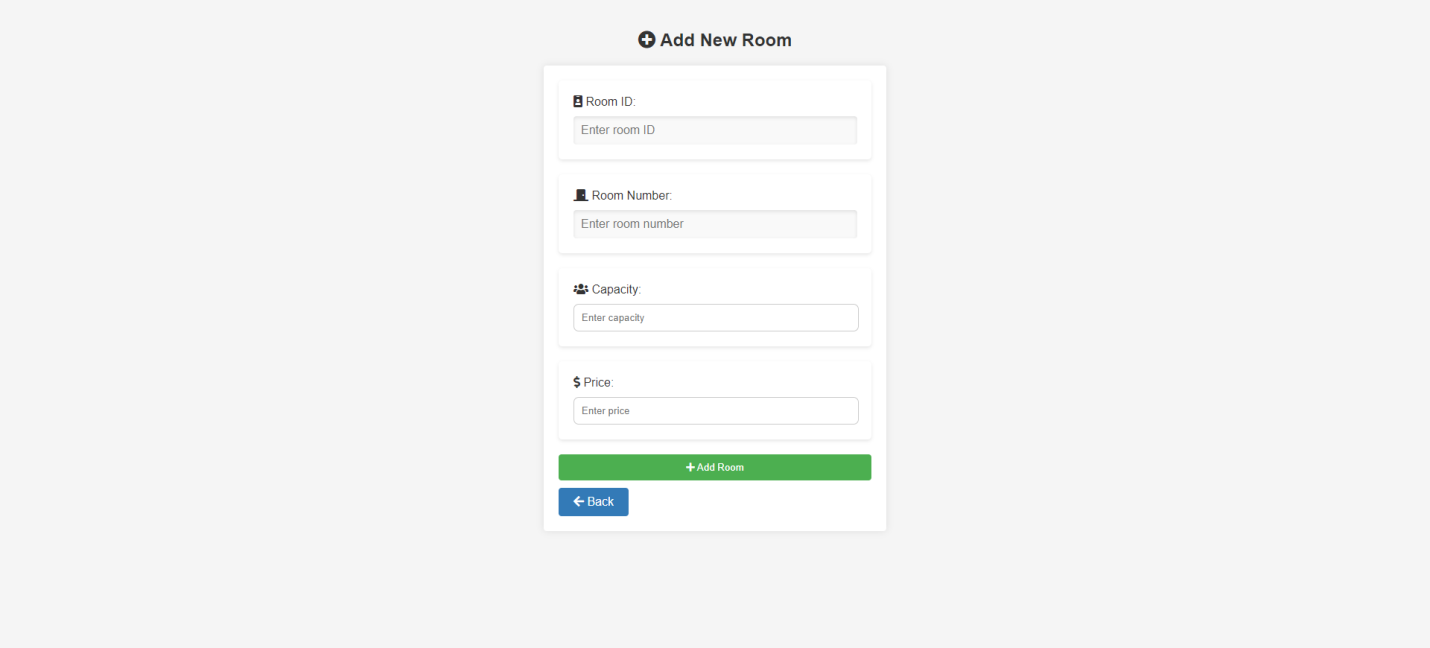
1. **Manage Students**

****

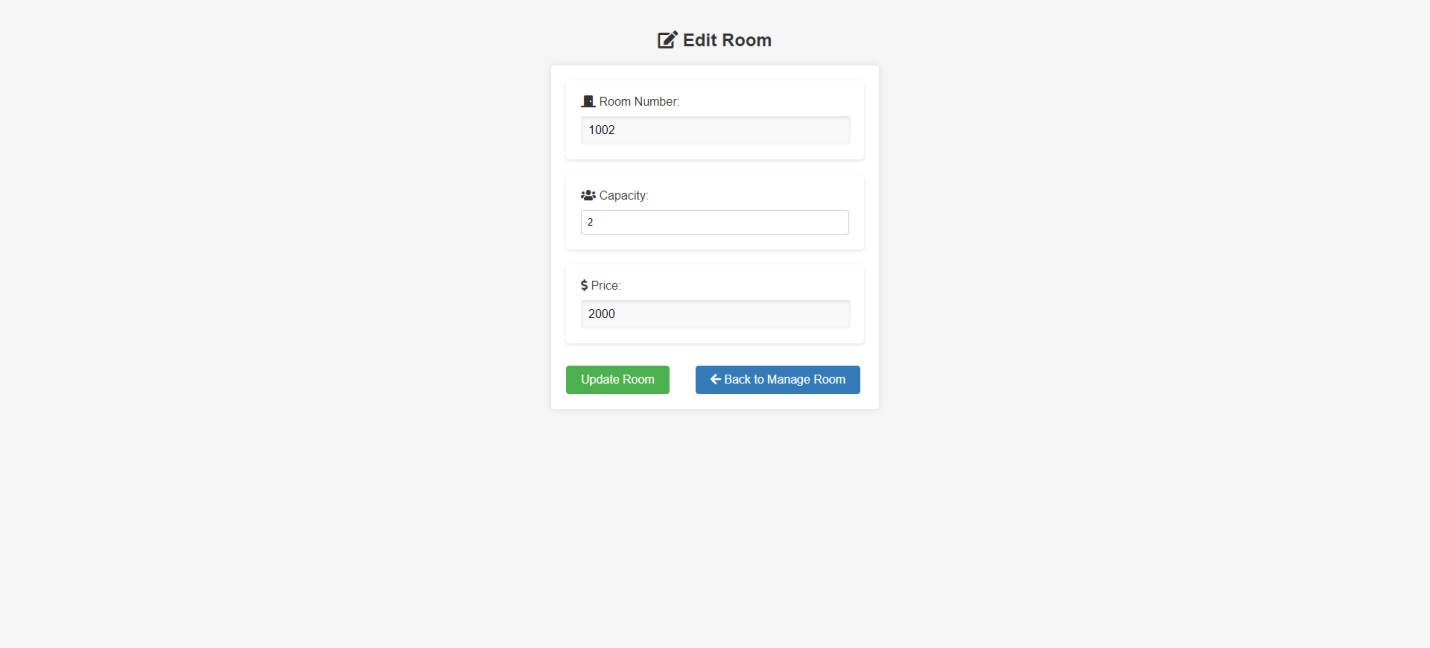
1. **Room Management**

****

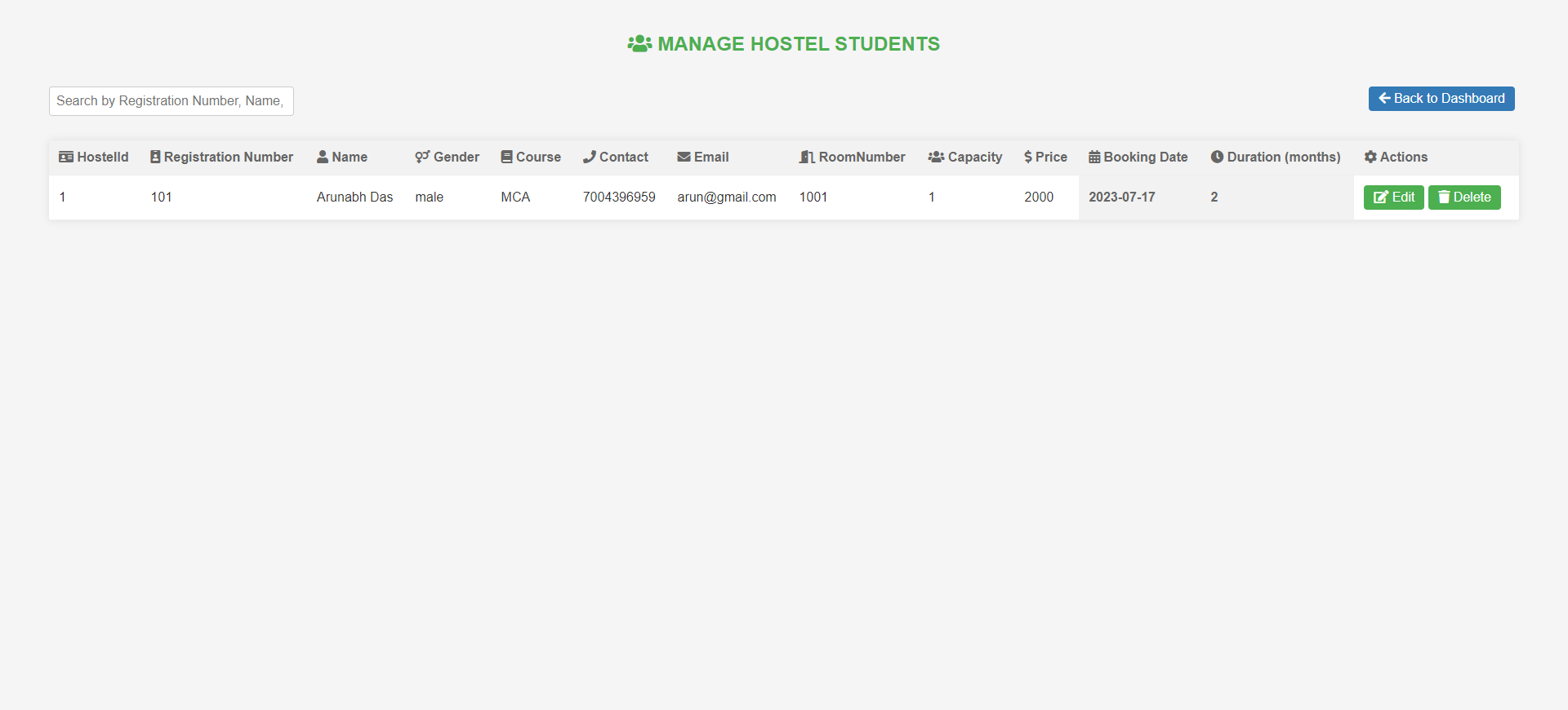
1. **Add New Room**

****

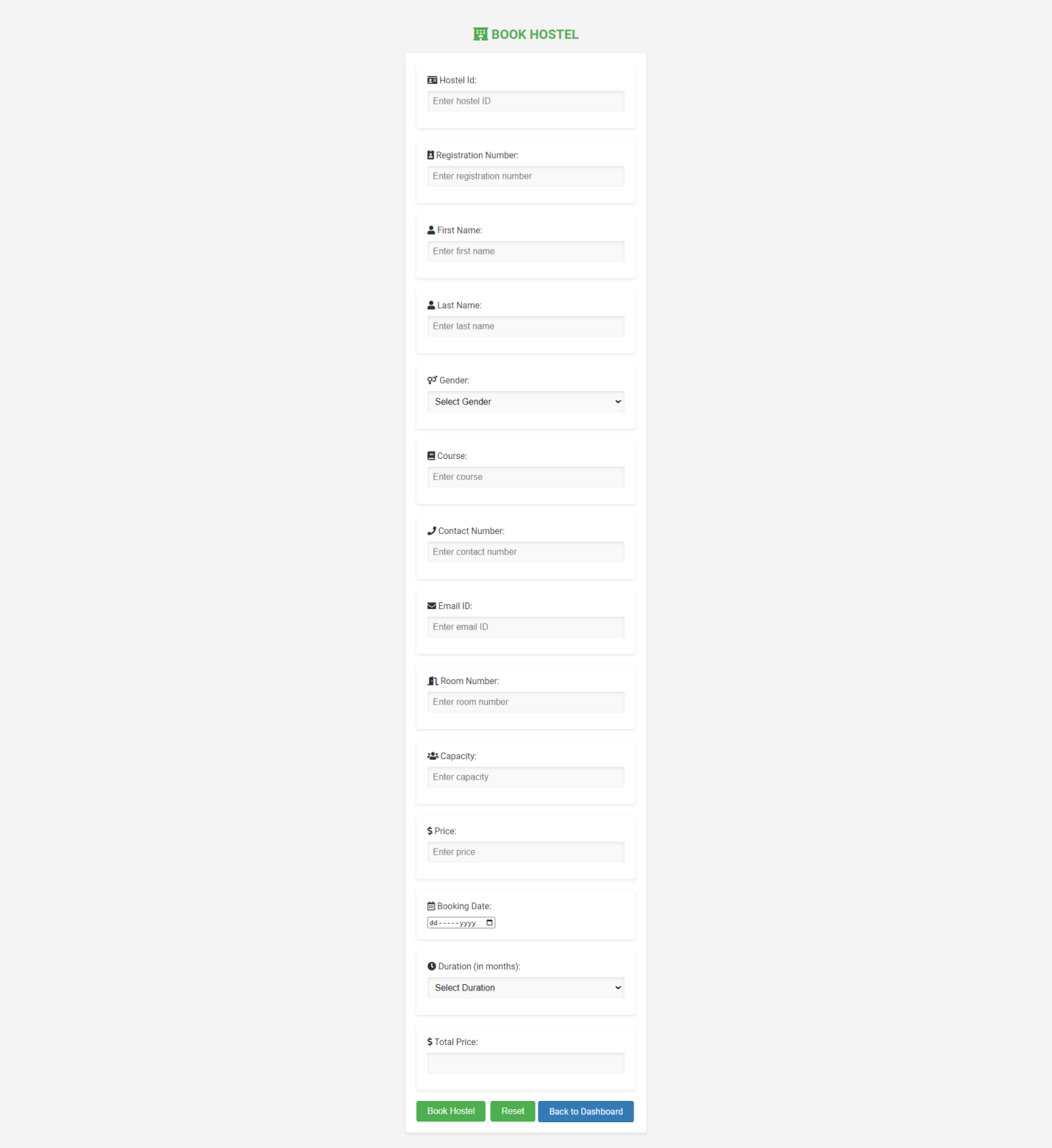
1. **Edit Rooms Details**

****

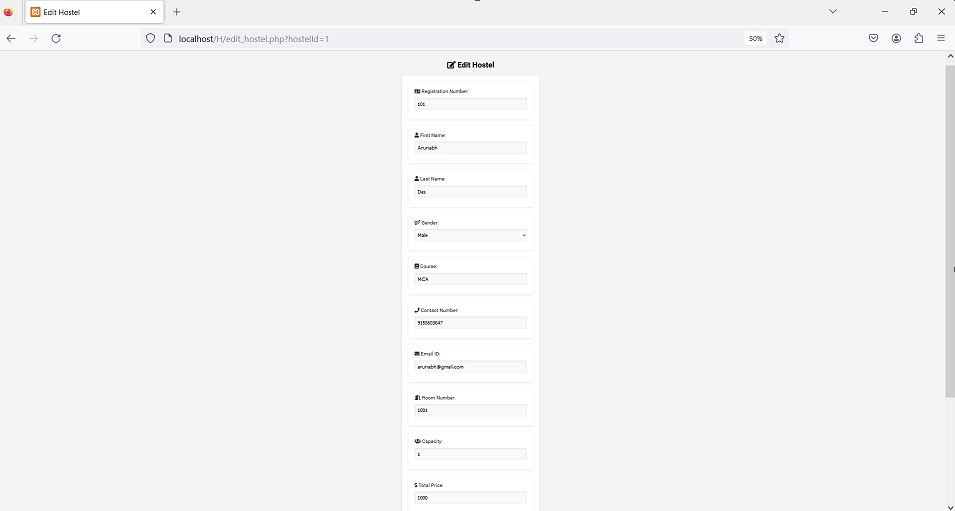
1. **Manage Hostel Students Records**



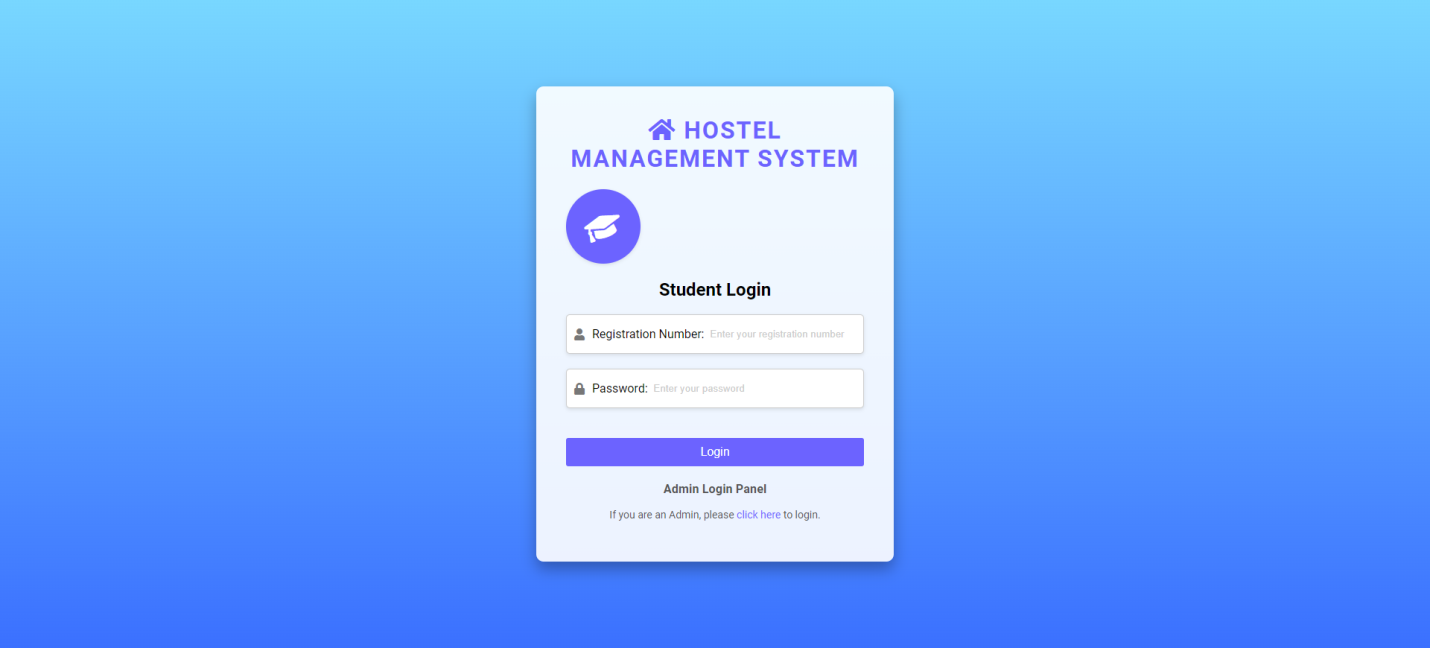
1. **Hostel Booking Form**

****

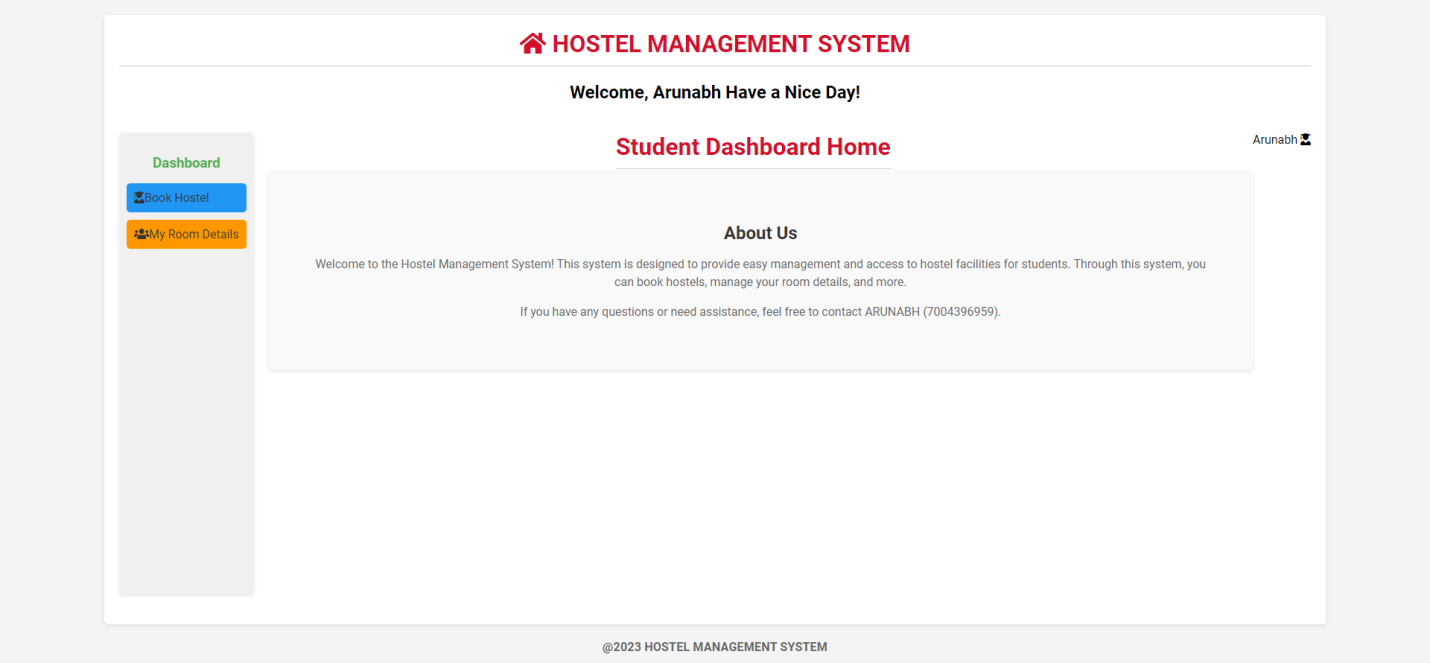
1. **Edit Hostel Students records**

****

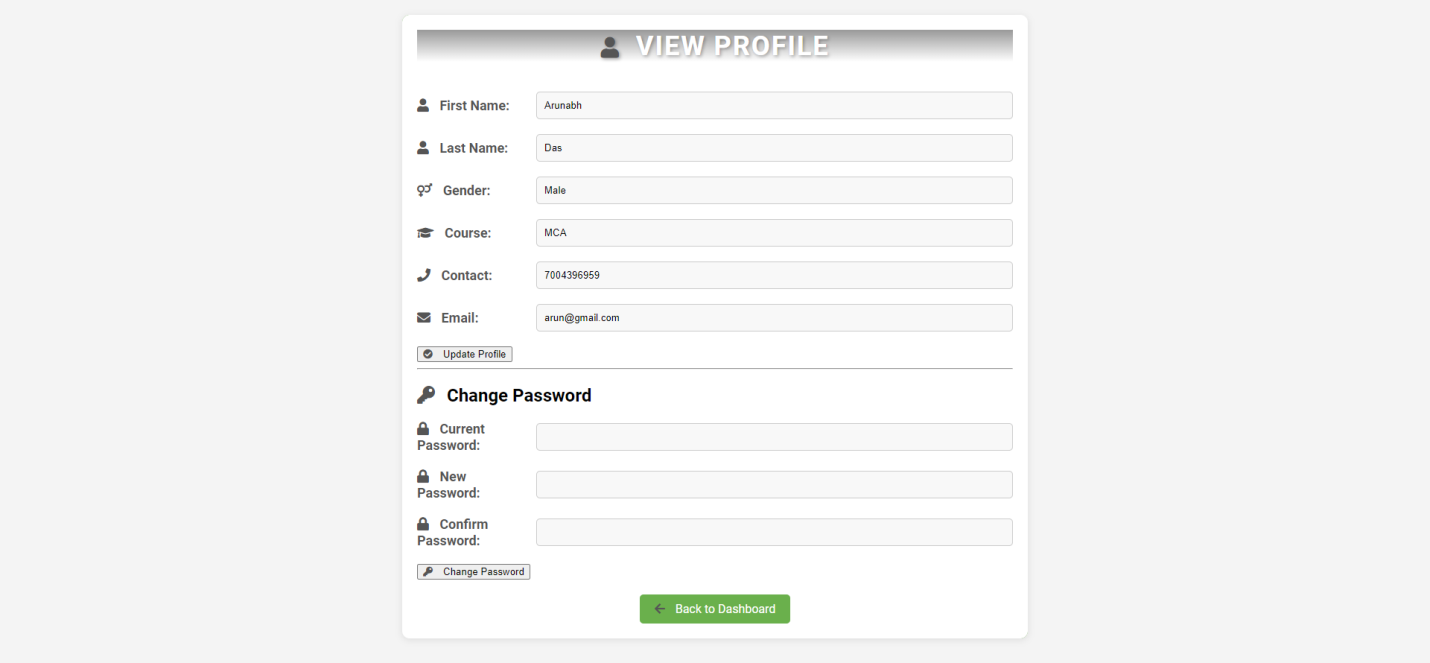
1. **Student Login Panel**

****

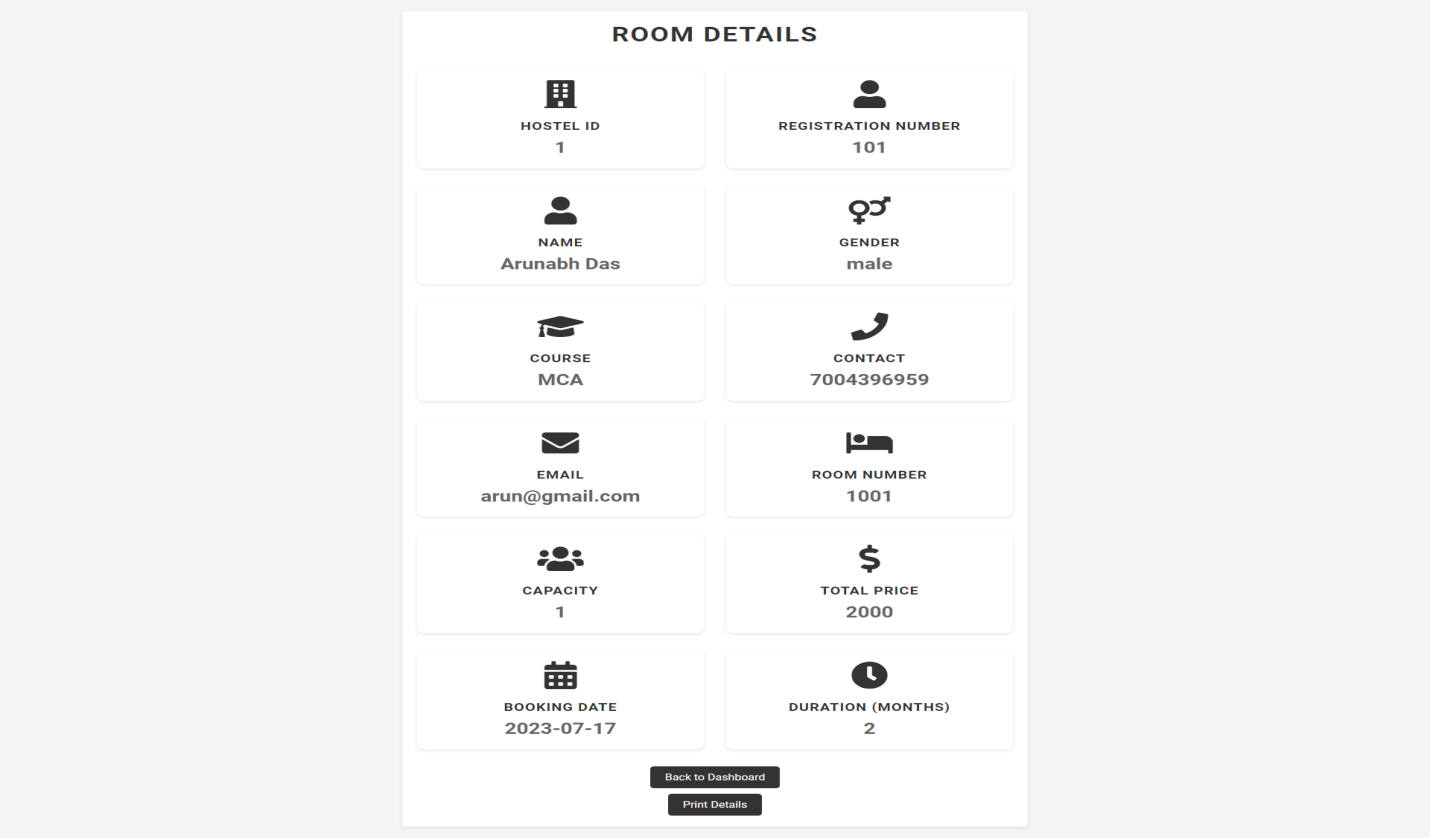
1. **Student Dashboard Home**

****

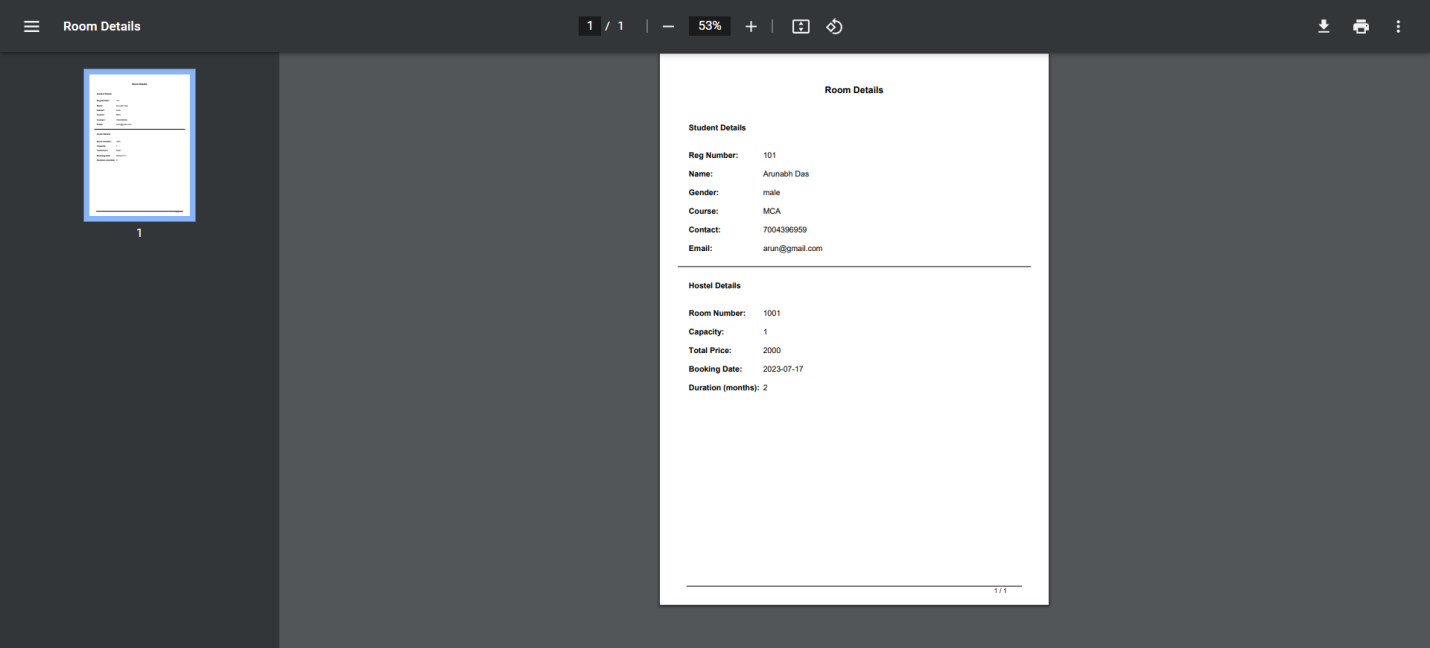
1. **View Profile**

****

1. **Booked Room Details**

****

1. **Print Record**

****

# Ch 7. TESTING

## 7.1 Test Case

These test cases cover various functionalities of the hostel management system, including adding new students, room allocation and deallocation, searching for student information, updating and deleting student data, calculating hostel fees, and generating occupancy and revenue reports.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Text ID** | **Test Purpose** | **Test Condition** | **Expected Output** | **Output** | **Remarks** |
| TC\_001 | Add new student | Valid student information provided | Success message | Success message | Test Pass |
| TC\_002 | Add new student | Missing required fields | Error message | Error message | Test Fail |
| TC\_003 | Add new student | Student already exists | Error message | Error message | Test Fail |
| TC\_004 | Allocate room to student | Valid student RegNo and available room | Room allocation success | Room allocation success | Ensure room availability |
| TC\_005 | Allocate room to student | Invalid student RegNo or unavailable room | Room allocation failed | Room allocation failed | Verify error message |
| TC\_006 | Deallocate room from student | Valid student Reg No | Room deallocation success | Room deallocation success | Test Pass |
| TC\_007 | Deallocate room from student | Invalid student RegNo | Room deallocation failed | Room deallocation failed | Verify error message |
| TC\_008 | Search student by name | Name exists in the database | List of students with name | List of students with name | Test Pass |

# 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC\_009 | Search student by name | Name does not exist in the database | Empty list | Empty list |  |
| TC\_010 | Search student by room number | Room number exists in the database | Student details | Student details | Test Pass |
| TC\_011 | Search student by room number | Room number does not exist in the database | Error message | Error message | Verify error message |
| TC\_012 | Update student information | Valid student ID and updated information | Update successful | Update successful | Test Pass |
| TC\_013 | Update student information | Invalid student ID | Update failed | Update failed | Verify error message |
| TC\_014 | Delete student | Valid student ID | Deletion successful | Deletion successful | Test Pass |
| TC\_015 | Delete student | Invalid student ID | Deletion failed | Deletion failed | Verify error message |
| TC\_016 | Calculate hostel fees for a student | Valid student ID with valid check-in and check-out dates | Correct total fees | Correct total fees | Test Pass |
| TC\_017 | Calculate hostel fees for a student | Invalid student ID or missing dates | Error message | Error message | Verify error message |
| TC\_018 | Generate Report | All Details of Logged in Student | Print report | Print report | Test Pass |

# Ch 8. CONCLUSION

## 8.1 CONCLUSION

To conclude the description about the project, it is developed using PHP and MySQL. It is based on the requirement specifications of the user and the analysis of the existing system, with flexibility for future enhancement.

This project is aimed at developing a system for keeping records and showing information about or in the hostel. It will provide complete information about a student in the hostel. It will show rooms available or not and number of people in a particular room.

This project deals with the problems occur when carried out manually. Identification of the drawbacks of the existing system with the system which is more users friendly and more GUI oriented. We can improve the efficiency of the system, and thus overcome the drawbacks of the existing system. Less human error, strain of manual labor can be reduced, high security, data redundancy can be avoided to some extent, data consistency, easy to handle, easy data updating, easy record keeping and backup data can be easily generated.

Through this project we do not require very efficient person to handle and manage the affairs of the students in the hostel, all we need to do is to login as the administrator and can see the information of the students who have obtained and registered their hostel form, click verify to ascertain their eligibility and allocate them to the available hostel. It helps them to get rid of the boring manual work from which it is very difficult to find the records of the students and rooms.

## 8.2 FUTURE SCOPE

Developing a dedicated mobile application would extend the system's accessibility and convenience for administrators and students on-the-go. Integrating email or SMS notifications would enhance communication by keeping students informed about hostel bookings, payment reminders, and important updates. Enhancing the search functionality with filters and sorting options would facilitate quicker access to specific records, improving user efficiency.

Furthermore, implementing role-based access control would ensure that different user roles have appropriate permissions, enhancing data security and privacy. Ongoing improvements in security practices, including regular security audits and updates.

## 8.3 REFERENCES

1. Frontend (HTML/CSS):
   1. MDN Web Docs HTML: <https://developer.mozilla.org/en-US/docs/Web/HTML>
   2. MDN Web Docs CSS: <https://developer.mozilla.org/en-US/docs/Web/CSS>
   3. Flexbox Froggy (Learn CSS Flexbox): <https://flexboxfroggy.com/>
   4. Responsive : [https:// freecodecamp.org/learn/responsive-web-design/](https://flexboxfroggy.com/)
2. Backend (PHP):
   1. PHP Manual: <https://www.php.net/manual/en/>
   2. PHP: The Right Way: <https://phptherightway.com/>
3. Database (MySQL):
   1. MySQL Documentation: <https://dev.mysql.com/doc/>
   2. PHP and MySQL CRUD Application Tutorial: <https://www.tutorialrepublic.com/php-tutorial/php-mysql-crud-application.php>