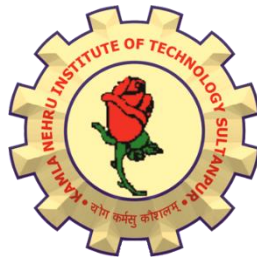


A PROJECT REPORT
ON
ONLINE HOSTEL ALLOTMENT

BY
MOHIT VERMA- (13623)
SAURABH KUMAR - (13629)
VAIBHAV KAUSHAL - (13642)
&
DHARMPAL CHAUDHARY - (148604)

UNDER THE SUPERVISION OF
Prof. ALKA SINGH

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF DEGREE OF
BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
KAMLA NEHRU INSTITUTE OF TECHNOLOGY
SULTANPUR (U.P.) – 228118

(An Autonomous State Government Institute)

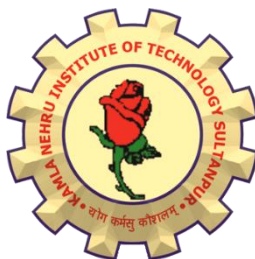
AFFILIATED TO
DR A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY
LUCKNOW (U.P.) INDIA

2016-2017

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

KAMLA NEHRU INSTITUTE OF TECHNOLOGY

SULTANPUR (U.P.) – 228118



CERTIFICATE

This is to certify that **Mohit Verma - (13623)**, **Saurabh Kumar- (13629)**, **Vaibhav Kaushal - (13642)** & **Dharmpal Chaudhary -(148604)** have carried out the project work in this report entitled “**ONLINE HOSTEL ALLOTMENT**” for the award of Bachelor of Technology in Information Technology at Kamla Nehru Institute of Technology, affiliated to Dr. A. P. J. Abdul Kalam Technical University (AKTU), Lucknow.

This report is the record of the candidates’ own work carried out by them under our supervision and guidance. This project work is the part of their Bachelor of Technology in Information Technology curriculum.

Their performance was excellent and we wish them good luck for their future endeavours.

Prof. Alka Singh
(Project Guide)

Prof Samir Srivastava
(Project In-Charge)

Dr. Anil Kumar Malviya
(Professor & Head)

ACKNOWLEDGEMENT

Here, we gladly present this project report on “**ONLINE HOSTEL ALLOTMENT**” as part of the 8th semester B.TECH in Computer Science and Engineering. We take this occasion to thank God, almighty for blessing us with his grace and taking our endeavour to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed guide, **Prof. Alka Singh** for providing us with the right guidance and advice at the crucial junctures and for showing us the right way. We extend our sincere thanks to our respected head of the department **Prof. A.K. Malaviya**, for allowing us to use the facilities available.

We would like to thank the other faculty members also, at this occasion. Last but not the least, we would like to thank our friends for the support and encouragement they have given us during the course of our work.

Submitted By -

Mohit Verma (13623)

Saurabh Kumar (13629)

Vaibhav Kaushal (13642)

Dharmpal Chaudhary (148604)

ABSTRACT

As the name specifies “**ONLINE HOSTEL ALLOTMENT** ” is a software developed for managing various activities in the hostel. For the past few years, the number of educational institutions have increased rapidly. Thereby, the number of hostels are 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 designing of computerized system that will be compatible to the existing system with the system which is more user friendly and easy to handle.

We can improve the efficiency of the system, thus overcoming the drawbacks of the existing system. The project will have the following advantages :

- Less human error
- High security
- Data consistency
- Easy to handle
- Easy data updating
- Easy record keeping

CONTENTS

Certificate	i
Acknowledgement	ii
Abstract	iii
1. Introduction	
1.1 Project Overview	1
1.2 Motivation	1
1.3 Project Objectives	1
2. System Analysis	
2.1 Existing System	2
2.2 Proposed System	2
3. Feasibility Study	
3.1 Technical Feasibility	3
3.2 Economical Feasibility	3
3.3 Operational Feasibility	3
4. Requirement Analysis And Specifications	
4.1 Modules	
4.1.1 Administrative Module	4
4.1.2 Student Module	4
4.2 Configurations	
4.2.1 Hardware Configuration	4
4.2.2 Software Configuration	5
5. System Design	
5.1 Administrator	6
5.2 Data Flow Diagrams	6
5.3 Use Case Diagrams	9
6. System Testing	
6.1 Unit Testing	10

6.2 Integration Testing	10
6.3 User Acceptance Testing	10
Conclusion	11
References	12
Appendix	
➤ Appendix A : Source Code.....	13
➤ Appendix B : Screenshots.....	37

1. INTRODUCTION

1.1 PROJECT OVERVIEW

The online hostel management system is web based software to provide college students accommodation to the university hostel more efficiently. This project also keeps details of the hostellers and applied students. It is headed by Dean. He will be the administrator. For accommodate a large number of students into hostel.

This document is intended to minimize human works and make hostel allocation is an easier job for college students and hostel authorities by providing online application for hostel, automatically select the students from the waiting list and mess calculation, complaint registration, notice board etc. etc. Students will come to know about the allotted room status once the entire process gets over.

1.2 Motivation

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and easy to handle. The existing system being more redundant and difficult to manipulate, motivated us to design a new system which would help in overcoming these obstacles.

1.3 Project Objectives

- Maintain the students as hostellers and waiting list students separately
- Process allotment list.
- Admin can send the approval notification to every approved student via email .
- Students can register their complaints.
- Admin can edit notice board and each student can view it.
- Hostel secretary can calculate hostel fee including mess fee and can edit mess menu
- Hostellers can check the status of every month's hostel fee.

2. SYSTEM ANALYSIS

2.1 Existing System

The existing system is manual based and need lot of efforts and consume enough time. In the existing system we can apply for the hostels online but the allotment processes are done manually. It may lead to corruptions in the allocation process as well as hostel fee calculation. The existing system does not deals with mess calculation and complaint registration.

Drawbacks of the existing system :

- More human power
- More strength and strain of manual labor needed
- Repetition of same procedure
- Less security
- Data redundancy
- Difficulty to handle
- Difficulty to update data

2.2 Proposed System

The proposed system is having many advantages over the existing system. It require less overhead and very efficient. The proposed system deals with the hostel allotment process efficiently. This system is intended to minimize human work and make the hostel allocation an easier job for college students and hostel authorities. The students are made available the online application and they simply need to register themselves by entering their basic details. Then, they need to submit their choice or preference for the hostel and the ones with whom they wish to live in. This application is submitted to the Admin who then decides which hostel is to be allotted to whom according to their merit ranks. Students will come to know about the allotted room status once the entire process gets over.

Advantages of the proposed system :

- Less human error
- High security
- Data consistency
- Easy to handle
- Easy data updating
- Easy record keeping

3. FEASIBILITY STUDY

3.1 Technical Feasibility

The technical feasibility in the proposed system deals with the technology used in the system. It deals whether the hardware and software used in the system are of the latest technology or not. It happens that after a system is prepared a new technology arises and the user wants the system based on that technology. This system use windows platform, HTML,CSS,Bootstrap and JSP as front end technology and Microsoft MySQL Server as backend technology. Thus, “ ONLINE HOSTEL ALLOTMENT ” is technically feasible.

3.2 Economical Feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis. Microsoft MySQL database easily available on the internet.

3.3 Operational Feasibility

The project has been developed in such a way that it becomes very easy even for a person with little computer knowledge to operate it. This software is very user friendly and does not require any technical person to operate .The model is easy to handle and user compatible. Thus, the project is operationally feasible.

4.REQUIREMENT ANALYSIS AND SPECIFICATION

4.1 Modules

4.1.1 Administrative Module

In administrator module, the administrator manages the master data's like server details and student details, accept the application of students, view the application forms, reject the fake applications, view the complaints of the students in the hostel, accept the vacating form and delete from the database, edit the notice boards and view complaints.

4.1.2 Student Module

In student module, the student can submit the application form, change password, can check status, view notice board and submit the vacating form.

4.2 Configuration

4.2.1 Hardware Configuration

The section of hardware configuration is an important task related to the software development insufficient random access memory may affect adversely on the speed and efficiency of the entire system. The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application.

Processor : Pentium IV and above

Processor speed : 1.4 GHz Onwards

System memory : 128 Mb minimum 256 Mb recommended

Cache size : 512 KB

RAM : 512 MB(Minimum)

Network card : Any card can provide a 100mbps speed

Network connection : UTP or Coaxile cable connection

Printer : Inkjet/Laser Color printer provides at least 1000 Dpi

Hard disk : 80Gb

Monitor : SVGA Color 15"

4.2.2 Software Configuration

A major element in building a system is the selection of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system. This document gives a detailed description of the software requirement specification. The study of requirement specification is focused specially on the functioning of the system. It allow the developer or analyst to understand the system, function to be carried out the performance level to be obtained and corresponding interfaces to be established.

Front end tool: HTML, CSS, Bootstrap, JSP

Backend: Microsoft SQL Server

Operating system: Windows 8.1

Client Side: HTML, JSP, Javascript, Bootstrap

Server Side: Servlets, JSP

5. SYSTEM DESIGN

5.1 Administrator

- The Administrator can allot different students to the different hostels.
- He can vacate the students for the hostels.
- He can control the status of the fee payment.
- He can edit the details of the students, can change their rooms, edit and delete the student records.
- He can edit the news board
- He can check the complaints

5.2 Dataflow Diagrams

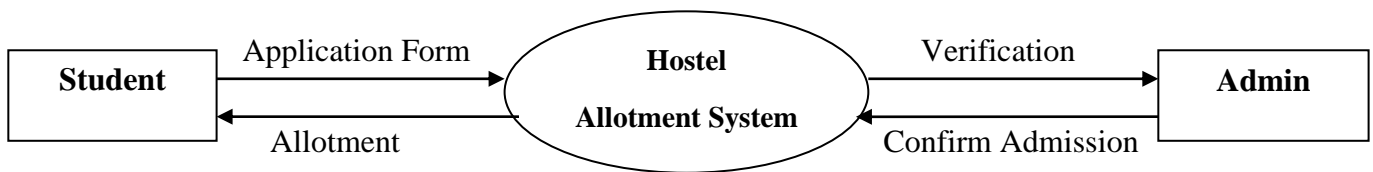


Fig 5.1 DFD for Allotment Process

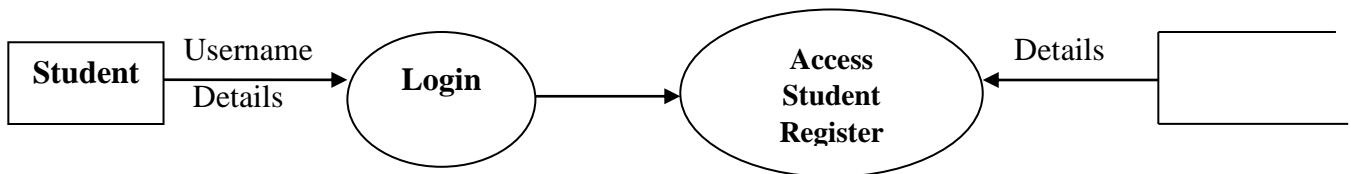


Fig 5.2 DFD for Student Module

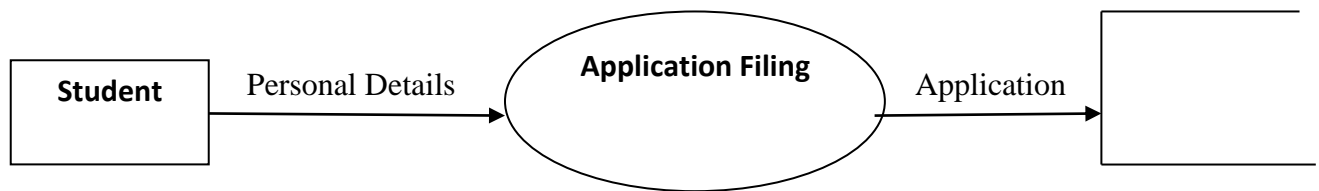


Fig 5.3 DFD for Student Registration

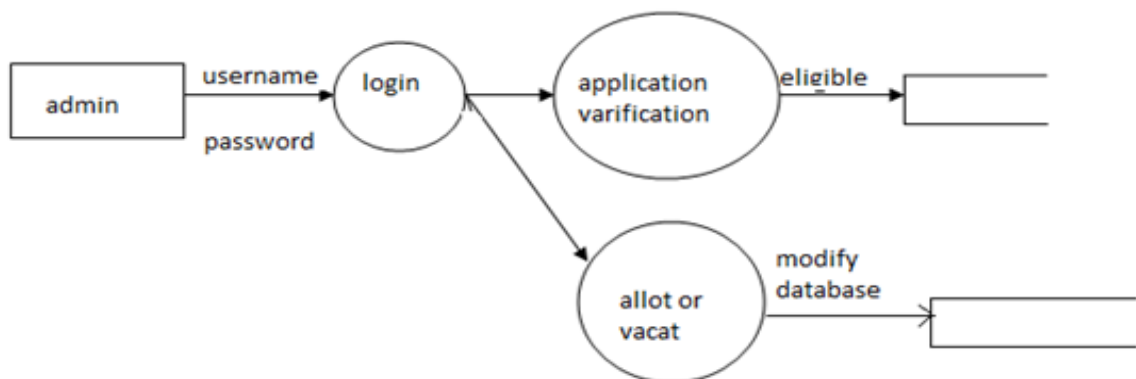


Fig 5.4 DFD to Allot and Vacate the room

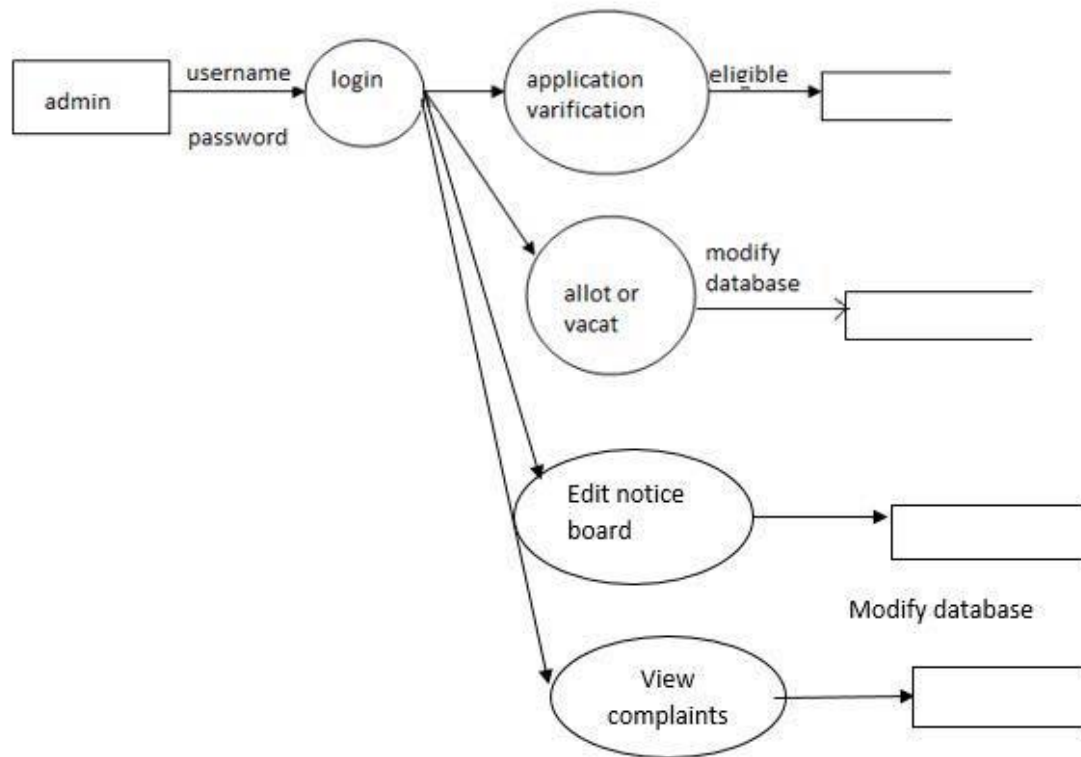


Fig 3.5 DFD for admin module

5.3 Use Case Diagrams



Fig 5.5 Use Case Diagram

6. SYSTEM TESTING

As the part of system testing we execute the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance.

Tests are carried out and the results are compared with the expected document. In the case of erroneous results, debugging is done. Using detailed testing strategies a test plan is carried out on each module. The various tests performed are unit testing, integration testing and user acceptance testing.

6.1 Unit Testing

The software units in the system is are modules and routines that are assembled and integrated to perform a specific function. As a part of unit testing we executed the program for individual modules independently. This enables, to detect errors in coding and logic that are contained within each of the three module. This testing includes entering data that is filling forms and ascertaining if the value matches to the type and entered into the database. The various controls are tested to ensure that each performs its action as required.

6.2 Integration Testing

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here the admin module, sec module and student module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

6.3 User Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keep the records of applicants and making changes to the details and password whenever required.

CONCLUSION

To conclude the description about the project The project, developed using ASP.net with c# and SQL SERVER is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement. ONLINE HOSTEL MANAGEMENT SYSTEM is very useful for hostel allotment and mess fee calculation . This hostel management software is designed for people who want to manage various activities in the hostel. For the past few years the numbers of educational institutions are increasing rapidly. Thereby the numbers of hostels are 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 designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented.

REFERENCES

- [1] Navathe & Korth for DBMS
- [2] W3schools for HTML, CSS, Bootstarp
- [3] Css files, from W3 schools.
[http://w3 school.com/css_file](http://w3school.com/css_file)
- [4] Javatpoint for JSP, Servlet applications
- [5] Css files, from Wikipedia, the free encyclopedia.
http://www.en.wikipedia.org/wiki/Wireless_security.
- [6] Stackoverflow

APPENDIX

Appendix A - SOURCE CODE

➤ Index.jsp

```
<!doctype html>

<html>

<head>

<link href="css/bootstrap.min.css" rel="stylesheet">

<link href="css/Mycss.css" rel="stylesheet">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width,initial-scale=1">

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

<title>Hostel Allotment</title>

<meta name="keywords" content="Hostel Allotment">

<style>

<!-- These three lines are written for carousals and after that are for making nav bar(home, hostel, rules
etc.) hoverable -->

.mySlides1 {display:none;}

.mySlides2 {display:none;}

.mySlides3 {display:none;}

.dropdown:hover .dropdown-menu {

                                display: block;

                                margin-top: 0;

                                }

.dropdown:hover {

                                text-decoration:none;

                                }

</style>

<script type="text/javascript">

    function showlogin()
```

[illegible]

```

</br>

</br>

</div>

<!-- For giving 3rd full division for third section-->

<div class="col-md-12" >

<div class="col-md-1" style="background-color:WHITE" ></div>

<!--For giving middle section-->

<div class="col-md-10" style="background-color:WHITE">

<div class="col-md-12" style="background-color:black" >

<div class="col-md-2" style="color::font-weight:bold;margin-top:10px;margin-bottom:10px;background-
color:black" ><a style="color:white" href="index.jsp"> Home</a></div>

<div class="col-md-2" style="color::font-weight:bold;margin-top:10px;margin-bottom:10px;background-
color:black">

<li class="dropdown">

<a href="#" class="dropdown-toggle alink" data-toggle="dropdown" role="button" aria-haspopup="true"
aria-expanded="false">Hostel<span class="caret"></span></a>

<ul class="dropdown-menu">

<li><a href="ramanujam.html">Ramanujam Hostel</li>

<li><a href="newvs.html">New VS Hostel</a></li>

<li><a href="oldvs.jsp">Old VS Hostel</a></li>

<li><a href="aryabhatt.html">Aryabhatt Hostel</li>

<li><a href="khosla.html">Khosla Hostel</a></li>

<li><a href="raman.html">Raman Hostel</a></li>

<li><a href="kalam.html">Kalam Hostel</li>

<li><a href="gargi.html">Gargi Hostel</a></li>

<li><a href="maitreyee.html">Maitreyee Hostel</a></li>

</ul>

</li>

</div>

```

```

<div class="col-md-2" style="color::font-weight:bold;margin-top:10px;margin-bottom:10px;background-color:black">

<li class="dropdown">

<a href="#" class="dropdown-toggle alink" data-toggle="dropdown" role="button" aria-haspopup="true"
aria-expanded="false">Rules<span class="caret"></span></a>

    <ul class="dropdown-menu">

        <li><a href="generalhostelrules.jsp">General Hostel Rules</li>

        <li><a href="girlrules.jsp">Girls Rules </a></li>

        <li><a href="messrules.jsp">Mess Rules</a></li>

    </ul>

</li>

</div>

<div class="col-md-2" style="color::font-weight:bold;margin-top:10px;margin-bottom:10px;background-color:black"><a style="color:white" href="Contacts.jsp"> Contacts</a></div>

<div class="col-md-2" style="color::font-weight:bold;margin-top:10px;margin-bottom:10px;background-color:black"><a style="color:white" href="Help.jsp"> Help</a></div>

<div class="col-md-2" style="color::font-weight:bold;margin-top:10px;margin-bottom:10px;background-color:black"><a style="color:white" href="loginregister.jsp"> Login</a></div>

</div>

<!--For giving a little space-->

<div class="col-md-12" style="background-color:white"></br></br></div>

<!--Everything is right above this line-->

<!--For giving photo sections-->

<div class="col-md-12">

<!--This is for carousal 1-->

<div class="col-md-4">

<h2 class="w3-center"></h2>

<div class="w3-content w3-section" style="max-width:500px">

```

```








</div>
```

```
</div>
```

```
<!--This is for carousal 2-->
```

```
<div class="col-md-4">
```

```
<h2 class="w3-center"></h2>
```

```
<div class="w3-content w3-section" style="max-width:500px">
```

```












</div>
```

```
</div>
```

```
<!--This is for carousal 3-->
```



```

<div class="col-md-4">
<h2 class="w3-center"></h2>
<div class="w3-content w3-section" style="max-width:500px">
  
  
  
  
  
  
  
  
  
  
</div>
</div>
</div>
</br></br></br>
<!--For giving a pre formatted more data window-->
  <div class="col-md-12"><pre>
The hostel can be allotted just by filling form so I would like to tell yyouu to fill the form first nd then
apply so that wwee can understand

what youu reeally looking fo

---<a href="page.jsp">more</a>
</pre></div>
<BR><BR><BR><BR><BR>
  <div class="col-md-1"></div>
</BR></BR></BR></BR></BR></div>
<div class="col-md-1" style="background-color:WHITE"></div>
</div>

```

```

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<!-- Include all compiled plugins (below), or include individual files as needed -->
<script src="js/bootstrap.min.js"></script>
<!--this line is for footer-->
<div class="col-md-12" style="height:50px;background-color:#222222"></div>
<script>
    //All these scripts are written for carousals
var myIndex1 = 0;
var myIndex2 = 0;
var myIndex3 = 0;
carousel1();
carousel2();
carousel3();
function carousel1() {
    var i;
    var x = document.getElementsByClassName("mySlides1");
    for (i = 0; i < x.length; i++) {
        x[i].style.display = "none";
    }

    myIndex1++;
    if (myIndex1 > x.length) { myIndex1 = 1 }
    x[myIndex1-1].style.display = "block";
    setTimeout(carousel1, 2000); // Change image every 2 seconds
    }
function carousel2()    {
    var i;
    var x = document.getElementsByClassName("mySlides2");
    for (i = 0; i < x.length; i++)

```

```

        {
            x[i].style.display = "none";
        }

        myIndex2++;
        if (myIndex2 > x.length) { myIndex2 = 1 }
        x[myIndex2-1].style.display = "block";
        setTimeout(carousel2, 2000); // Change image every 2 seconds
    }
}

function carousel3() {
    var i;
    var x = document.getElementsByClassName("mySlides3");
    for (i = 0; i < x.length; i++) {
        x[i].style.display = "none";
    }

    myIndex3++;
    if (myIndex3 > x.length) { myIndex3 = 1 }
    x[myIndex3-1].style.display = "block";
    setTimeout(carousel3, 2000); // Change image every 2 seconds
}

</script>
</body>
</html>

```

➤ Applyhostel.jsp

```
<!doctype html>

<html>

<head>

<link href="css/bootstrap.min.css" rel="stylesheet">

<link href="css/Mycss.css" rel="stylesheet">

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width,initial-scale=1">

<title>Apply Hostel</title>

<meta name="keywords" content="Hostel Allotment">

</head>

<body>

<div class="container">

    <h2>Apply for Hostel Application Form</h2>

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

    <div class="col-md-4">

        <div class="form-group">

            <label for="exampleSelect1">Select Your Year</label>

            <select class="form-control" name="year" id="exampleSelect1">

                <option>1</option>

                <option>2</option>

                <option>3</option>

                <option>4</option>

            </select>

        </div>

        <div class="form-group">

            <label for="exampleInputEmail1">Enter Your Rollno</label>
```

```

<input type="text" class="form-control" id="exampleInputEmail1" name="rollno" aria-
describedby="emailHelp" placeholder="Enter Your Rollno">

<div> <% String message=(String)request.getAttribute("message1");

    if(message != null)

    {

        out.println(message);

    }

    %>

</div>

</div>

<div class="form-group">

<label for="exampleInputEmail1">Enter Your Branch</label>

<input type="text" class="form-control" id="exampleInputEmail1" name="branch" aria-
describedby="emailHelp" placeholder="Enter Your Branch">

</div>

<div class="form-group">

<label for="exampleInputPassword1">Enter Your Marks</label>

<input type="text" class="form-control" name="marks" id="exampleInputPassword1"
placeholder="Last Year Marks">

</div>

<div class="form-group">

<label for="exampleInputPassword1">Enter Your Partener Rollno</label>

<input type="text" class="form-control" name="partenerrollno" id="exampleInputPassword1"
placeholder="Your Partener Rollno">

<div> <% String message1=(String)request.getAttribute("message2");

    if(message1 != null)

    {

        out.println(message1);

    }

    %>

```

```

        </div>
</div>
<div class="form-group">
    <label for="exampleSelect1">Select Your Hostel</label>
    <select class="form-control" name="hostelchoice" id="exampleSelect1">
        <option>Ramanujam Hostel</option>
        <option>Aryabhatt Hostel</option>
        <option>Khosla Hostel</option>
        <option>Kalam Hostel</option>
        <option>Maitrayee Hostel</option>
        <option>OldVs Hostel</option>
    </select>
<br>
<div> <%   String message2=(String)request.getAttribute("message3");
        if(message2 != null)
        {
            out.println(message2);
        }
        %>
    </div>
    <br>
    <button type="submit" class="btn btn-primary">Submit</button>
</div>
</form>
</div>
</body>
</html>

```

➤ Addroom.jsp

```
<% @ page import="java.sql.*"%>
<%
    ResultSet resultset = null;
%>
<!doctype html>
<html>
<head>
<link href="css/bootstrap.min.css" rel="stylesheet">
<link href="css/Mycss.css" rel="stylesheet">
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,initial-scale=1">
<title>Add Room</title>
<meta name="keywords" content="Hostel Allotment">
<script language="javascript">
    function fncSubmit() {
        if (document.addroomForm.hostelname.value == "choose") {
            alert('Please Select Hostel Name');
            document.addroomForm.hostelname.focus();
            return false;
        }
        if (document.addroomForm.roomname.value == "")
        {
            alert('Please Enter Room Name');
            document.addroomForm.roomname.focus();
            return false;
        }
    }
}
```

```

    }
</script>

</head>

<body>

<div class="container">

<div class="col-md-4"></div>

<div class="col-md-4" style="background-color: #D5DBDB">

    <h2>Add Room</h2>

    <br>

<form name="addroomForm" action="addroomservlet" method="post"

        OnSubmit="return fncSubmit()">

<!-- printing message hostel added successfully -->

    <div style="color: red">

        <%

                                String message1 = (String)
request.getAttribute("message1");

                                if (message1 != null) {
                                        out.println(" " + message1);
                                }

                                %>

        </div>

        <div style="color: green">

            <%

                                String message = (String)
request.getAttribute("message");

                                if (message != null) {
                                        out.println(" " + message);
                                }

                                %>

```



```

        </div>

<!-- fetching hostels name from database -->

<%
        try {
                Class.forName("com.mysql.jdbc.Driver");

                Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/HostelAllotementDB",
                                "root", "saurabh");

                Statement statement = connection.createStatement();
                resultset = statement.executeQuery("select hostel_name
from hostel");

        %>

<div class="form-group">

                <label for="exampleInputroom">Select Hostel</label> <select
                                class="form-control" name="hostelname"
id="exampleInputroom">

                                <option>choose</option>

                                <%
                                        while (resultset.next()) {
                                                String name =
resultset.getString("hostel_name");

                                %>

                                <option value="<%=name%>"><%=name%></option>

                                <%
                                        }

                                %>

                                </select>

                </div>

<!-- end fetching hostels -->

                <div class="form-group">

```

```

<input
    <label style="text-align: center">Enter Room Name </label>
    type="text" class="form-control" name="roomname"
    id="exampleInputroom" placeholder="Room Name">
</div>
<button type="submit" class="btn btn-primary">Submit</button>
<br> <br>
<%
    /**Should I input the codes here?*/
    } catch (Exception e) {
        out.println("wrong entry" + e);
    }
%>
</form>
<div class="col-md-4"></div>
</div>
</body>
</html>

```

➤ **Logout.jsp**

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>logout</title>

</head>

<body>

<% session.invalidate(); %>

<p>You have been successfully logout</p>

<% request.getRequestDispatcher("index.jsp").forward(request, response); %>

</body>

</html>
```

➤ **Register.java**

```
import java.io.*;
import java.sql.*;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.*;

public class Register extends HttpServlet {

    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        String username = request.getParameter("username");
        String name = request.getParameter("name");
        String year = request.getParameter("year");
        String branch = request.getParameter("branch");
        String password=request.getParameter("password");

        // validating that rollno is available in database or not
        if(validatestudent.checkStudentExist(username))
        {

            String message1="you have already registered";

            request.setAttribute("message1", message1);

            RequestDispatcher rs = request.getRequestDispatcher("loginregister.jsp");
            rs.include(request, response);

            return;
        }
    }
}
```

```

    }

    try {

        Class.forName("com.mysql.cj.jdbc.Driver");

        java.sql.Connection con =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/HostelAllotementDB","root", "saurabh");
        PreparedStatement ps = con.prepareStatement("insert into student_register values(?,?,?,?)");

        ps.setString(1, username);

        ps.setString(2, name);

        ps.setString(3, year);

        ps.setString(4, branch);

        ps.setString(5, password);

        int i = ps.executeUpdate();

        if (i > 0) {

            String message2="you have successfully registered now you can login and apply for hostel";

            request.setAttribute("message2", message2);

            RequestDispatcher rs = request.getRequestDispatcher("loginregister.jsp");

            rs.include(request, response);

            return;

        }

        else

        {

        }

        } catch (Exception e2) {

        System.out.println(e2);

        }

    out.close();

}}

```

➤ **Login.java**

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/login")
public class login extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        PrintWriter out = response.getWriter();
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String logintype = request.getParameter("logintype");
        if(logintype==null){
            String message="please select any logintype";
            request.setAttribute("message", message);
            RequestDispatcher rs = request.getRequestDispatcher("index.jsp");
            rs.forward(request, response);
```

```

    }

    if(validatestudent.checkUser(username, password) &&
logintype.equals("student") )
    {
        // session checking

        HttpSession session = request.getSession();
        session.setAttribute("username", username);
        request.setAttribute("username", username);

        RequestDispatcher rs = request.getRequestDispatcher("home.jsp");
        rs.forward(request, response);

    }

    else if(validatewarden.checkUser(username, password) &&
logintype.equals("Warden") )
    {
        // session checking

        HttpSession session = request.getSession();
        session.setAttribute("username", username);
        request.setAttribute("username", username);

        RequestDispatcher rs = request.getRequestDispatcher("wardenhome.jsp");
        rs.forward(request, response);

    }

    else if(validateDSW.checkUser(username, password) &&
logintype.equals("DSW") )
    {
        // session checking

        HttpSession session = request.getSession();
        session.setAttribute("username", username);

```

```
        request.setAttribute("username", username);
        RequestDispatcher rs = request.getRequestDispatcher("DSWhome.jsp");
        rs.forward(request, response);
    }
    else
    {
        String message="Username or Password incorrect";
        request.setAttribute("message", message);
        RequestDispatcher rs = request.getRequestDispatcher("loginregister.jsp");
        rs.include(request, response);
    }
}

}
```


➤ **Logout.java**

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
 * Servlet implementation class logout
 */
@WebServlet("/logout")
public class logout extends HttpServlet {
    private static final long serialVersionUID = 1L;

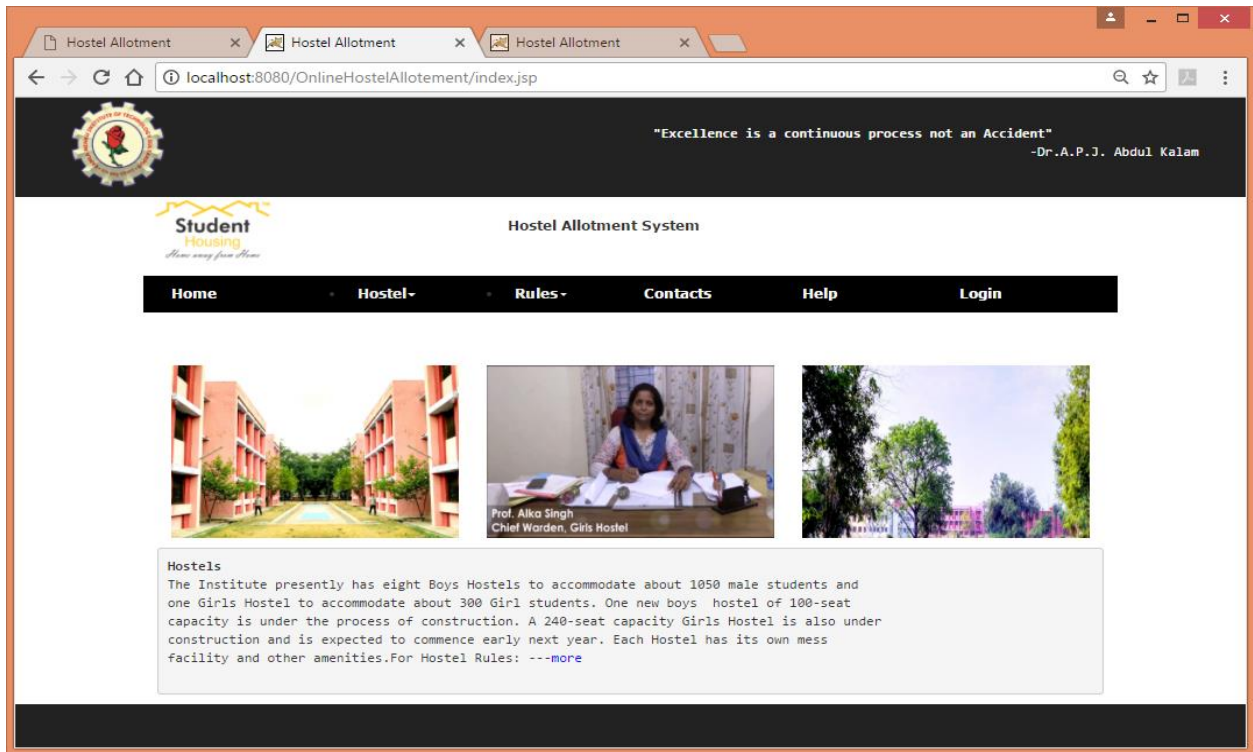
    /**
     * @see HttpServlet#HttpServlet()
     */
    public logout() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException
    {
        HttpSession session = request.getSession();
```

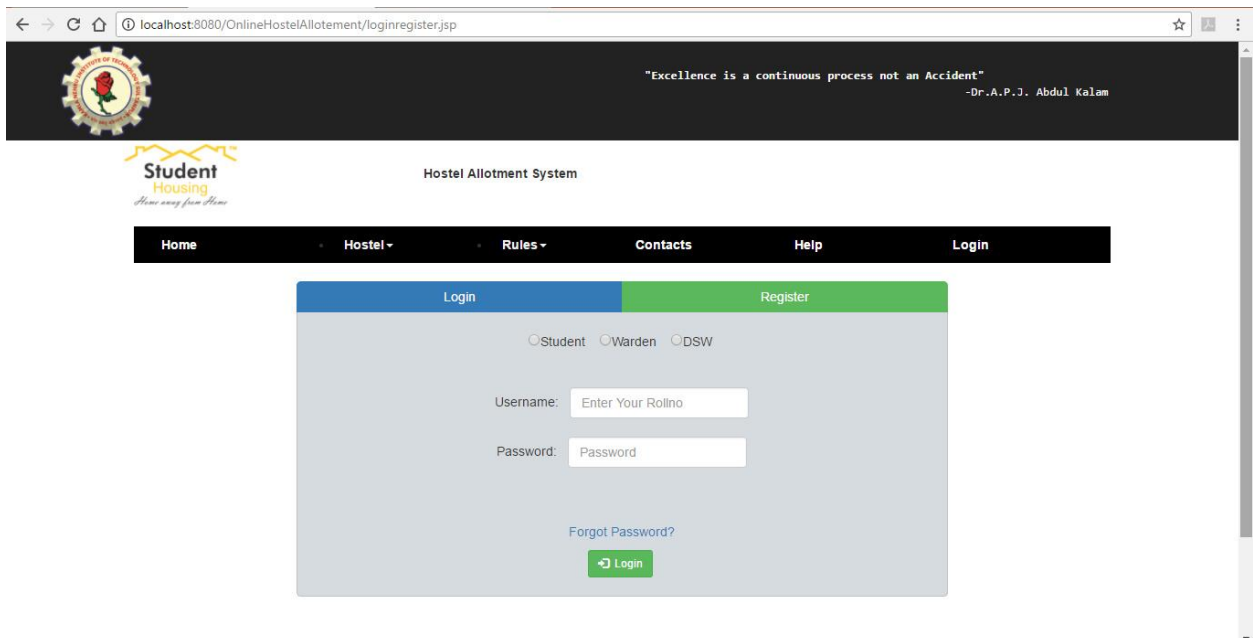
```
        session.removeAttribute("username");  
        session.invalidate();  
        response.sendRedirect("index.jsp");  
    }  
}
```

Appendix B - Screenshots

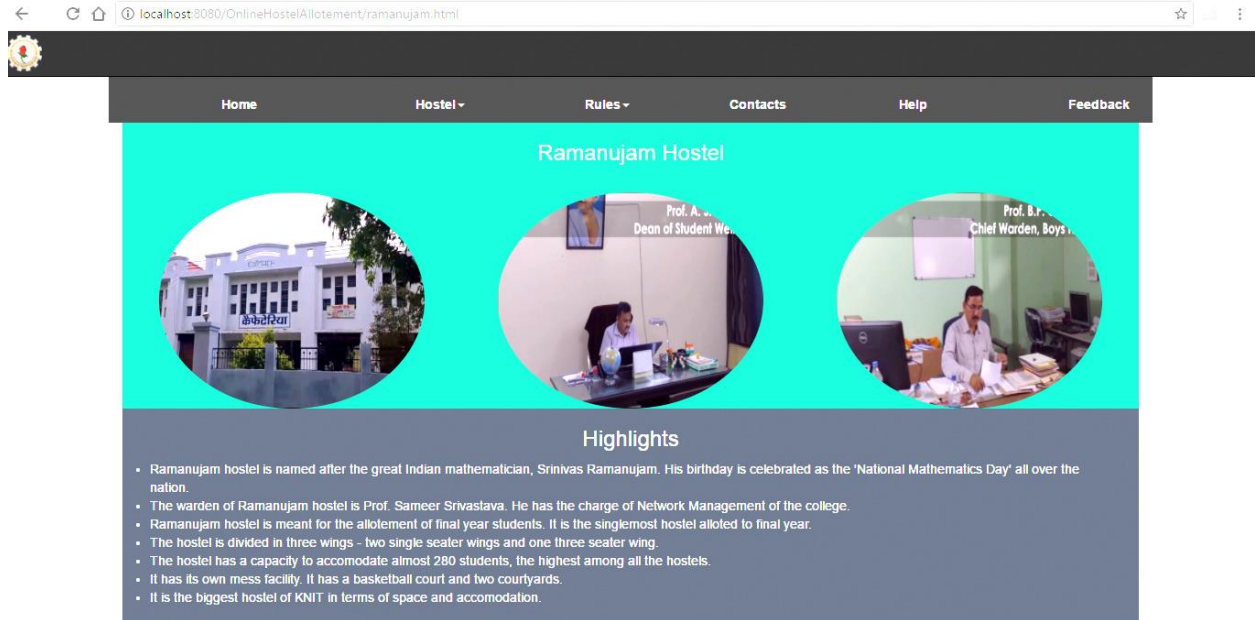
➤ Home Page



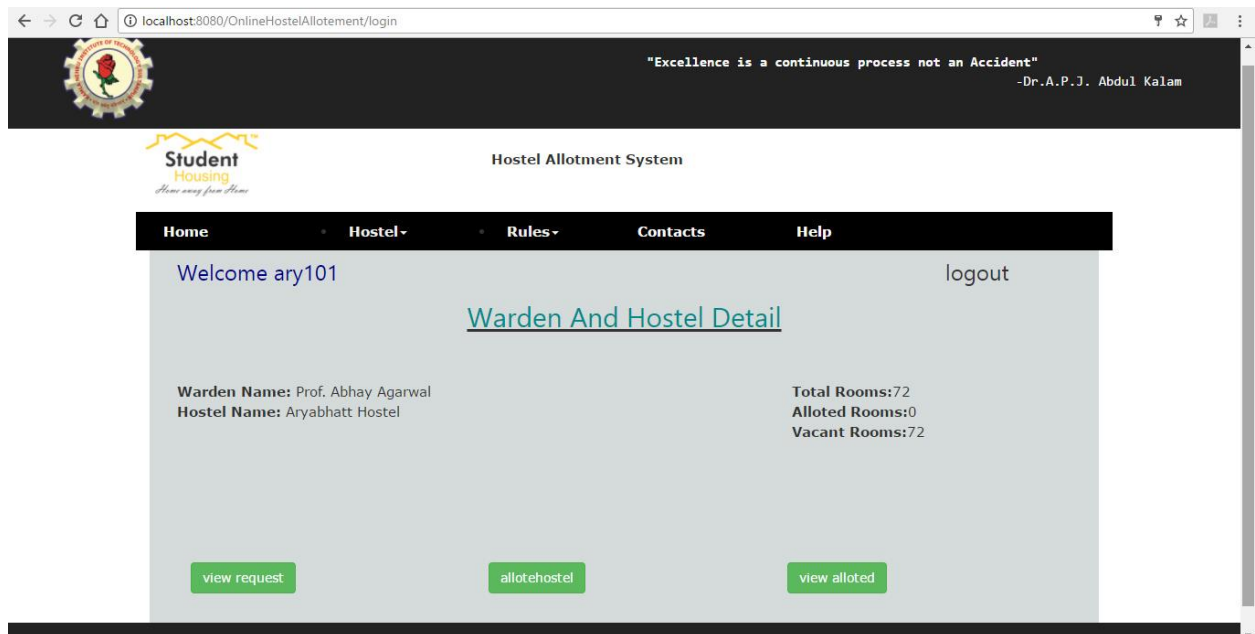
➤ Student Login



➤ Ramanujam Hostel



➤ Warden and Hostel Details



➤ Hostel Allotment

localhost:8080/OnlineHostelAllotement/allotedRooms.jsp

Alloted Rooms						
Year	Rollno	Branch	Marks	Partenerrollno	Hostel	Alloted Room
3	13601	it	1500	13602	Aryabhata Hostel	f2-05
3	13604	it	1490	13606	Aryabhata Hostel	g1-11
3	13607	it	1508	13608	Aryabhata Hostel	f1-01
3	13609	it	1504	13610	Aryabhata Hostel	f2-14
3	13611	it	1503	13612	Aryabhata Hostel	g1-02
3	13613	it	1545	13614	Aryabhata Hostel	g2-16
3	13615	it	1589	13616	Aryabhata Hostel	f1-11
3	13617	it	1600	13618	Aryabhata Hostel	f2-07
3	13619	it	1610	13620	Aryabhata Hostel	g1-05
3	13621	it	1690	13622	Aryabhata Hostel	g1-03
3	13623	it	1620	13624	Aryabhata Hostel	g2-12
3	13625	it	1700	13626	Aryabhata Hostel	g2-03
1	13629	it	1500	13630	Aryabhata Hostel	f2-02
1	13629	it	1500	13630	Aryabhata Hostel	f2-02
1	13629	it	1200	13630	Aryabhata Hostel	f2-02
1	13629	it	1200	13501	Aryabhata Hostel	f2-02

➤ Application for Hostel Allotment

localhost:8080/OnlineHostelAllotement/applyhostel.jsp

Apply for Hostel Application Form

Select Your Year

1

Enter Your Rollno

Enter Your Rollno

Enter Your Branch

Enter Your Branch

Enter Your Marks

Last Year Marks

Enter Your Partener Rollno

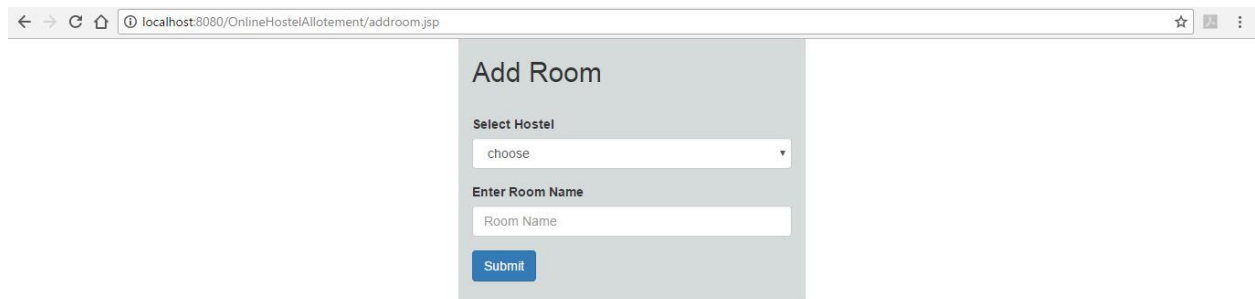
Your Partener Rollno

Select Your Hostel

Ramanujam Hostel

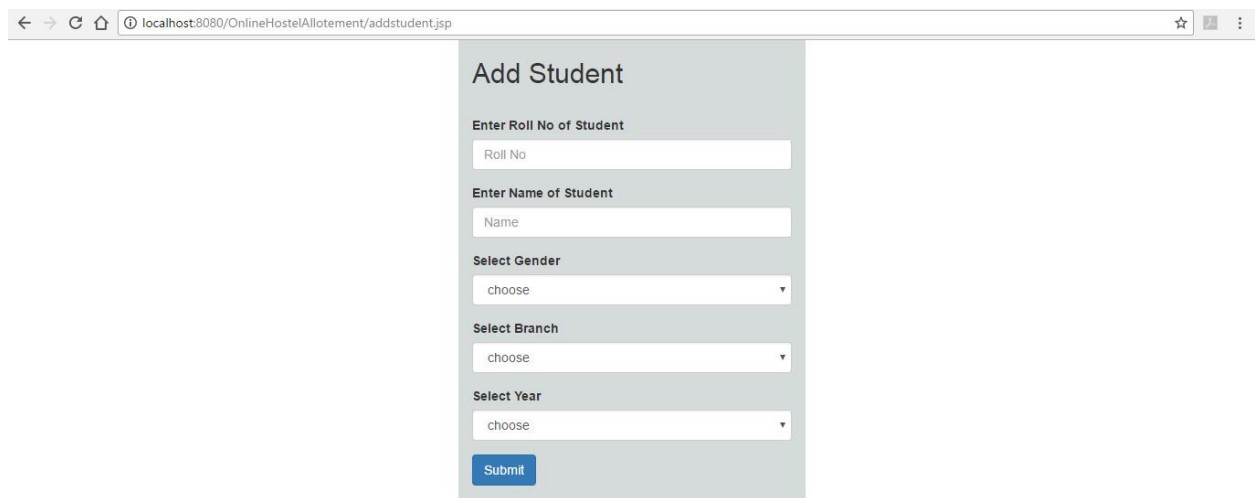
Submit

➤ Add Room



A screenshot of a web browser displaying the 'Add Room' form. The browser's address bar shows the URL 'localhost:8080/OnlineHostelAllotement/addroom.jsp'. The form itself is a light gray box with the title 'Add Room' at the top. It contains a 'Select Hostel' dropdown menu with 'choose' selected, an 'Enter Room Name' text input field with 'Room Name' entered, and a blue 'Submit' button at the bottom.

➤ Add Student



A screenshot of a web browser displaying the 'Add Student' form. The browser's address bar shows the URL 'localhost:8080/OnlineHostelAllotement/addstudent.jsp'. The form is a light gray box with the title 'Add Student' at the top. It contains five input fields: 'Enter Roll No of Student' (text input with 'Roll No'), 'Enter Name of Student' (text input with 'Name'), 'Select Gender' (dropdown menu with 'choose'), 'Select Branch' (dropdown menu with 'choose'), and 'Select Year' (dropdown menu with 'choose'). A blue 'Submit' button is located at the bottom of the form.