



A project report on  
**HOSTEL PORTAL**

Submitted in partial fulfilment of the requirements for the Degree of

B. Tech in Computer Science and Engineering

by

**Avinab Sen (1705298)**  
**Dattatraya Deb (1705304)**  
**Nabankur Dey (1705418)**  
**Supriyo Nandi (1705470)**

under the guidance of

**Prof. N Biraja Isac**

School of Computer Engineering  
Kalinga Institute of Industrial Technology  
Deemed to be University  
Bhubaneswar

April 2021



Deemed to be University U/S 3 of the UGC Act, 1956

## CERTIFICATE

This is to certify that the project report entitled "**HOSTEL PORTAL**" submitted by

|                       |                |
|-----------------------|----------------|
| <b>AVINAB SEN</b>     | <b>1705298</b> |
| <b>DATTATRAYA DEB</b> | <b>1705304</b> |
| <b>NABANKUR DEY</b>   | <b>1705418</b> |
| <b>SUPRIYO NANDI</b>  | <b>1705470</b> |

in partial fulfilment of the requirements for the award of the **Degree of Bachelor of Technology in Computer Science and Engineering** is a bona fide record of the work carried out under our guidance and supervision at School of Computer Engineering, Kalinga Institute of Industrial Technology, Deemed to be University.

**Date: 20/04/2021**

.....  
**(Prof. N Biraja Isac)**

**(Project Supervisor)**

## **ACKNOWLEDGEMENTS**

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We are profoundly grateful to Prof. N Biraja Isac, our Supervisor, for her expert guidance and encouragement throughout to see that this project rights its target since its commencement to its completion. We are grateful to Dr. Bhabani Sankar Prasad Mishra, Dean of Computer Science and Engineering, who gave us the opportunity to do our project. The work is a team effort without which the completion of this project was not possible.

**AVINAB SEN  
DATTATRAYA DEB  
NABANKUR DEY  
SUPRIYO NANDI**

## ABSTRACT

As the name specifies “HOSTEL PORTAL” is a software developed for managing various activities in the hostel. For the past few years the number of educational institutions are increasing 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 more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

*Less human error*

*Strength and strain of manual labour can be  
reduced High security*

*Data redundancy can be avoided to some extent*

*Data consistency*

*Easy to handle*

**Keywords:** Hostel Portal, Management, Student Allocation, Data Consistency, Upcoming events, Chat bot, Contact us

## TABLE OF CONTENTS

|           |  |              |
|-----------|--|--------------|
| <b>1.</b> | <b>Introduction</b>                        | <b>1 - 3</b> |
| 1.1       | PROBLEM DEFINITION.....                    | 1            |
| 1.2       | AUDIENCE AND VIEWER ELIGIBILITY.....       | 2            |
| 1.3       | SCOPE OF THE PRODUCT.....                  | 2            |
| 1.4       | 1.4.1 OBJECTIVES.....                      | 2            |
|           | 1.4.2 GOALS.....                           | 3            |
| <b>2.</b> | <b>Literature Survey</b>                   | <b>4</b>     |
| 2.1       | GATHERED INFORMATION FROM CLIENTS.....     | 4            |
| <b>3.</b> | <b>Software Requirements Specification</b> | <b>5 - 9</b> |
| 3.1       | FUNCTIONAL REQUIREMENTS.....               | 5            |
|           | 3.1.1 LOGIN.....                           | 5            |
|           | 3.1.2 STUDENT PROFILE.....                 | 5            |
|           | 3.1.3 COMPLAINT.....                       | 5            |
| 3.2       | NON-FUNCTIONAL REQUIREMENTS.....           | 6            |
|           | 3.2.1 SAFETY REQUIREMENTS.....             | 6            |
|           | 3.2.2 SECURITY REQUIREMENTS.....           | 6            |
|           | 3.2.3 SOFTWARE QUALITY ATTRIBUTES.....     | 6            |
| 3.3       | EXTERNAL INTERFACE REQUIREMENTS.....       | 7 - 9        |
|           | 3.3.1 USER INTERFACES.....                 | 7            |
|           | 3.3.2 HARDWARE INTERFACES.....             | 7            |
|           | 3.3.3 SOFTWARE INTERFACES.....             | 8            |
|           | 3.3.4 COMMUNICATION INTERFACES.....        | 9            |
| <b>4.</b> | <b>Requirement Analysis</b>                | <b>10</b>    |
| 4.1       | PURPOSE.....                               | 10           |
| 4.2       | SCOPE.....                                 | 10           |
| <b>5.</b> | <b>System Design</b>                       | <b>11-15</b> |
| 5.1       | DETAIL DESIGN.....                         | 11           |
| 5.2       | ARCHITECTURE OF THE PROPOSED SYSTEM.....   | 11           |
|           | 5.2.1 STUDENT USER.....                    | 11           |
|           | 5.2.2 ADMINISTRATOR.....                   | 11           |

|            |                                    |  |              |
|------------|------------------------------------|--|--------------|
|            | <b>5.3</b>                         | PROCESS DESIGN.....  | <b>12-14</b> |
|            | 5.3.1                              | STUDENT LOGIN.....   | 13           |
|            | 5.3.2                              | WARDEN LOGIN.....  | 13           |
|            | 5.3.3                              | COMPLAINT STATUS.....  | 14           |
|            | <b>5.4</b>                         | DATABSE DESIGN.....  | <b>14-15</b> |
|            | 5.4.1                              | STUDENT DETAILS DATABASE.....                                      | 15           |
|            | 5.4.2                              | LOGIN DETAILS DATABASE.....  | 15           |
|            | 5.4.3                              | WARDEN LOGIN DATABASE.....   | 15           |
|            | 5.4.4                              | HOSTEL DETAILS DATABASE.....                                       | 15           |
| <b>6.</b>  | <b>System Testing</b>              |  | <b>16</b>    |
|            | 6.1                                | Test Cases and Test Results.....                                   | 16           |
| <b>7.</b>  | <b>Implementation</b>              |  | <b>17</b>    |
|            | 7.1                                | CODE FOR LOGIN.....  | 17-18        |
|            | 7.2                                | CODE FOR COMPLAINT.....  | 19           |
|            | 7.3                                | CODE FOR APPROVAL AND REJECTION OF<br>COMPLAINTS (BY WARDENS)..... | 20           |
|            | 7.4                                | CODE FOR CONTACT US  | 21           |
|            | 7.5                                | CODE FOR DARK MODE   | 21           |
| <b>8.</b>  | <b>Screenshots of Project</b>      |  | <b>22-28</b> |
|            | 8.1                                | WEBSITE AND APP SCREENSHOTS.....                                   | 22-28        |
|            | 8.1.1                              | LOGIN SCREEN.....  | 22           |
|            | 8.1.2                              | HOME PAGE.....   | 23           |
|            | 8.1.3                              | CONTACT SCREEN.....  | 24           |
|            | 8.1.4                              | COMPLAIN BOX.....  | 25           |
|            | 8.1.5                              | DASHBOARD.....   | 26           |
|            | 8.1.6                              | STUDENT PROFILE.....   | 27           |
|            | 8.1.7                              | HOSTEL DETAILS.....  | 28           |
|            | 8.1.8                              | UPCOMING EVENTS.....   | 29           |
|            | 8.1.9                              | CHAT BOT.....  | 30           |
| <b>9.</b>  | <b>Conclusion and Future Scope</b> |  | <b>31</b>    |
|            | 9.1                                | CONCLUSION.....  | 31           |
|            | 9.2                                | FUTURE SCOPE.....  | 32           |
| <b>10.</b> | <b>References</b> .....            |  | <b>33</b>    |

## LIST OF FIGURES

| <b>Figure ID</b> | <b>Figure Title</b>  | <b>Page No.</b> |
|------------------|--|-----------------|
| 5.1              | Level - 0 DFD  | 12              |
| 5.2              | Level-1 DFD  | 12              |
| 5.3              | Hostel Portal for Student Login                            | 13              |
| 5.4              | Hostel Portal for Warden Login                             | 12              |
| 5.5              | Hostel Portal for Complaint Status                         | 15              |
| 7.1              | Code For Login   | 17-18           |
| 7.2              | Code For Complaint   | 19              |
| 7.3              | Code For Approval And Rejection of Complaints (By Wardens) | 20              |
| 7.4              | Code For Contact Us  | 21              |
| 7.5              | Code For Dark Mode   | 21              |
| 8.1.1            | Login Screen   | 22              |
| 8.1.2            | Home Page  | 23              |
| 8.1.3            | Contact Screen   | 24              |
| 8.1.4            | Complain Box   | 25              |
| 8.1.5            | Dashboard  | 26              |
| 8.1.6            | Student Profile  | 27              |
| 8.1.7            | Hostel Details   | 28              |
| 8.1.8            | Upcoming Events  | 29              |
| 8.1.9            | Chat Bot   | 30              |
| 8.1.10           | Contact Us   | 30              |

## LIST OF TABLES

| <b>Table Id</b> | <b>Table Title</b>          | <b>Page No.</b> |
|-----------------|-----------------------------|-----------------|
| 5.4.1           | Student Details Database    | 15              |
| 5.4.2           | Login Details Database      | 15              |
| 5.4.3           | Warden Login Database       | 15              |
| 5.4.4           | Hostel Details Database     | 15              |
| 6.1             | Test Cases And Test Results | 16              |

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 PROBLEM DEFINITION**

The purpose of this project to make an automated system to carry out different operations of a hostel. The system will provide the ease, comfort of use to the staff of the hostel by performing all work on computer system rather than following a paper pen approach. This approach helps improving the reliability of data maintained and provides a fast and efficient for the users of the website.

We have got seventeen hostels in our university, which consist of eleven boy's hostel and six girl's hostel. All these hostels at present are managed manually by the hostel office. The Registration form verification to the different data processing are done manually.

Thus there are a lot of repetitions which can be easily avoided. And hence there is a lot of

Hostel Management Portal is a customized and user friendly website for hostel which provide Hostel information, Hostel room information and Hostel accounts information. It helps admin to manage Student record, staff record generating report of students, etc.

## 1.2 AUDIENCE VIEWER AND ELIGIBILITY

The Software Requirements Specification is primarily intended for all project developers associated with this project. This Specification is organized into several sections or members that can be read and referenced as follows:-

1. **Project Manager-** To manage all processes in the project.
2. **Software Designer-** To design the models and diagrams that helps the programmer in implementation phase.
3. **Software Tester-** To test system by using dummy data.
4. **Database Administrator-** To perform database operations
5. **Software Analyst-** To analyze the requirements of hostel management system.

## 1.3 SCOPE OF THE PRODUCT

The software product “Hostel Portal” will be an application that will be used for maintaining the records in an organized manner and to replace old paper working on the system. This projects aims at automating the hostel management for smooth working of the hostel by automating almost all the calculations and accounting work would be accurate. Hostel Management Portal is designed for hostel like (Schools and Universities).

## 1.4 OBJECTIVES AND GOALS

### 1.4.1 OBJECTIVES

The basic objective of the “Hostel Portal” is to maintain hostel management activities as follows:

- Room/Hostel Change Request
- Maintaining Student’s Records
- Maintaining Warden’s Records
- Provide to student’s Complaints

#### **1.4.2 GOALS**

The goals of the system are to accept these problems in an effective and optimal manner by:

1. Centralizing the database and thus providing consistent data to all the employees in the Hostel.
2. Make the system more user friendly by providing an intensive user interface.
3. Easy access through reports.
4. Restricted data access to employees thus providing additional security to data.
5. A platform where students can complaint regarding hostel issues.
6. Students can also request for hostel/room change directly through website.

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 GATHERED INFORMATION FROM VARIOUS SOURCES**

Hostel management system development using PHP program has lots of codes. Using Internet in gathering information partially contributed to the success of this project. Due to the fact that PHP is an open source program, development of hostel management system was not too difficult. However, thanks to the cyber world (Internet) that makes it possible to study and make comparison in needs of some code function. A number of hostel portal system documents were examined and compared to the need of KIIT's proposed hostel portal system. Among other web site that was used in my research is "<http://freesourcecode.com/>". It provided me with various lines of codes which I used in the developmental process of this project.

- a. "A review of PHP Compilers and Outputs" Favre, Nicolas (2010-02-16) - gave a good layout of product design.
- b. Personal Home Page Tools (PHP Tools) Lerdorf, Rasmus (1995-06-08), was very helpful and insightful in the product development of this project.
- c. W3schools have prototypes that helped building interfaces for the project.

# **CHAPTER 3**

## **SOFTWARE REQUIREMENTS SPECIFICATION**

### **3.1 FUNCTIONAL REQUIREMENTS**

#### **3.1.1 LOGIN**

- ✓ The website/app allows a visitor to login by authenticating whether the user is a hosteller student or a warden of the designated hostel.
- ✓ The hosteller has a dedicated account which is mapped to his/her respective User ID and Password by the system.
- ✓ The system allows an account holder to change the password used for logging in.
- ✓ All the wardens share a common account dedicated only to a particular hostel or block.

#### **3.1.2 STUDENT PROFILE**

- ✓ The system allows a hosteller to view and update his/her details in the ‘My Profile’ section.
- ✓ The hosteller is also able to see the Mess Menu which is updated as per the season and accompanied by the availability of regional foods.
- ✓ The hosteller can view the hostel details wherein there is the list of wardens and their contact details, room details and Hostel name of the hosteller.
- ✓ The system also allows the hosteller to check out the facilities available in their respective hostels.

#### **3.1.3 COMPLAINT**

- ✓ The system allows a hosteller to raise complaints where each complaint associated is given an ID and stored in the database.
  - ✓ Once an warden resolves a complaint, the system marks it as approved.
  - ✓ The wardens of the designated hostel is allowed to view the approved/disapproved complaints as the system keeps track of the raised complaints.
- The hostellers may request to change rooms or report any other issues related to the particular hostel.

## 3.2 NON-FUNCTIONAL REQUIREMENTS

### 3.2.1 SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

### 3.2.2 SECURITY REQUIREMENTS

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

### 3.2.3 SOFTWARE QUALITY ATTRIBUTES

- **AVAILABILITY:** The Wardens should be available online periodically as many Students may be raising issues frequently in the need of the hour.
- **CORRECTNESS:** The Students and Wardens should edit their profile and fill them according to the original data (official use). Also the Students should raise an issue if it is genuine.
- **MAINTAINABILITY:** The Administrators and Wardens in chargers should maintain correct information in the database.
- **USABILITY:** The Wardens should resolve maximum number of issues raised within a given period of time.

### 3.3 EXTERNAL INTERFACE REQUIREMENTS

#### 3.3.1 USER INTERFACES

There are three types of user Interfaces as:

- i. Administration Interface
- ii. Student Interface
- iii. Warden Interface

##### ADMINISTRATOR :

An administrator is one who monitors all users. Admin has to maintain data of every student profile and warden profile in database. When the request is given by user admin checks the availability of user account then it forwarded to the student database. Admin have the complete information related to every student and warden database and all the information related to the students and wardens. All data is maintained at the admin level.

##### STUDENT :

Every student who have room in hostel have a database and a student account to access his data. these permissions shall be showed after administrator approval.

##### DATABASE MANAGER :

Database manager is a user who have the administrator permission to update the entire database. In this project the database manager has the permission to update the student update details, room details, warden details and update of mess.

#### 3.3.2 HARDWARE INTERFACES

Hardware Interfaces exist in computing systems between many of the components such as various storage devices, other I/O devices these are following for project:

➤Processor:  
Pentium IV and above

➤RAM: 3GB or more

### **3.3.3 SOFTWARE INTERFACES**

The software is developed with all the basic controls and class provided in JAVA and SQL, Windows XP or above installed on the system. Application package must be installed.

- Operating system : Windows 10.
- Front End tools : PHP as scripting language
- Back End : SQL Server
- Client Side : HTML,CSS,Bootstrap and JavaScript (ES-5)

#### **HTML (Hyper-Text Markup Language) :**

HTML elements forms the building blocks of all websites, allows Image sand objects to be embedded and can to be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as heading, paragraphs, lists, links, quotes and so on. It can also embed scripts written in languages such as JavaScript which affect the behaviour of HTML web pages.

#### **CSS (Cascading Style Sheets) :**

CSS is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is designed basically to enable the separation of document content from document presentation, including elements such as layout, colours and fonts.

#### **Bootstrap :**

Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of colour, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements.

#### **PHP (Hypertext Preprocessor) :**

PHP is a popular general-purpose scripting language that is especially suited to web development. PHP code is usually processed on a web server by a PHP interpreter

implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of a HTTP response.

#### SQL (Structured Query Language) :

To work with data in a database, you must use a set of commands and statements (language) defined by the DBMS software. There are several different languages that can be used with relational databases; the most common is SQL.

#### JavaScript :

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. It is used to make the page dynamic

#### XAMPP Server :

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

### 3.3.4 COMMUNICATION INTERFACES

The system shall be a standalone product that does not require any communication interfaces. The application shall communicate with the databases and software services via API function calls. Because the application will be written in Java, Java functions will make these calls to the APIs. The exact formats and protocols for incoming and outgoing messages should be abstracted by the APIs.

## **CHAPTER 4**

### **REQUIREMENT ANALYSIS**

#### **4.1 PURPOSE**

To build an online Hostel Portal to manage Hostel activities to ease management efficiently.

#### **4.2 SCOPE**

The purpose of the online Hostel Portal is to ease management and to create a convenient and easy-to-use application for Hostellers and Wardens to handle various complaints, resolving issues of mess or room changing facilities and other concerns of the residing Hostel. The system is based on a relational database. We will have a database server supporting a number of hostels of an Institution and the people associated with the respective hostels. Above all, we hope to provide a comfortable user experience.

# **Chapter 5**

## **System Design**

### **5.1 DETAIL DESIGN**

The design of our implemented system is discussed here. The Design phase involves planning different stages for implementing the Software. In a system, result of a process can be seen in the output. The design of our output specification is carried out with much user friendliness. The Hostel Portal System Design is simple and efficient and is just made of a login page where users can login as per their respective id which leads them to their respective homepage,further where they can process their complaint through the complain window and wardens can access them through the complaint box.

### **5.2 ARCHITECTURE OF THE PROPOSED SYSTEM**

The shown below is the system architecture and how the system is implemented and how their functions are being achieved.

The system design is divided in two portions: the *Administrator* section and the *User* (student's) section.

#### **5.2.1 STUDENT USER**

The student has to login to register a complaint.The complaint given by the student of the respective hostels' will be taken care of by their respective wardens.

#### **5.2.2 ADMINISTRATOR**

1. The Administrator can allot different students to the different hostels.
2. He can vacate the students for the hostels.
3. He can change their rooms, edit and delete the student records.

## Level 0 →



Figure 5.1: Level - 0 DFD

### 5.3 PROCESS DESIGN

Process design plays an important role in project development. In order to understand the working procedure, process design is necessary. Data Flow Diagram and System Flow chart are the tools used for process design. Data Flow Diagram is the logical representation of the data flow of the project. The DFD is drawn using various symbols. It has a source and a destination. The process is represented using circles and source and destination are represented using squares. The data flow is represented using arrows. One reader can easily get the idea about the project through Data Flow Diagram.

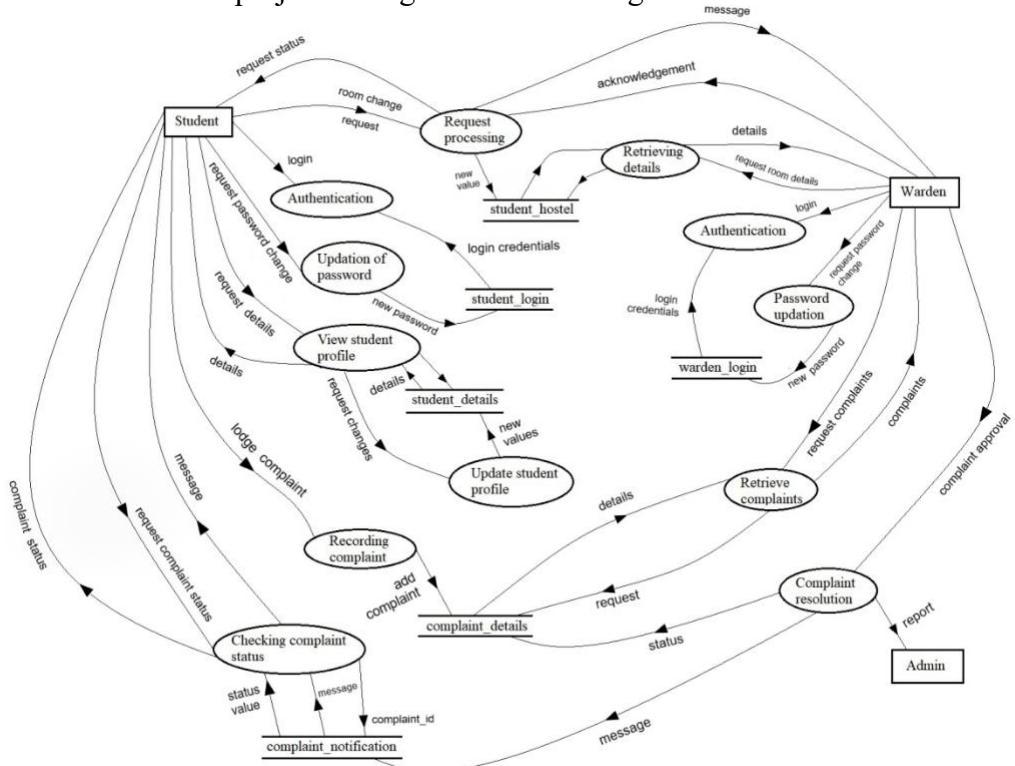


Figure 5.2: Level-1 DFD

### 5.3.1 STUDENT LOGIN

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped.here the use case diagram shows the students can login and register their complaints.

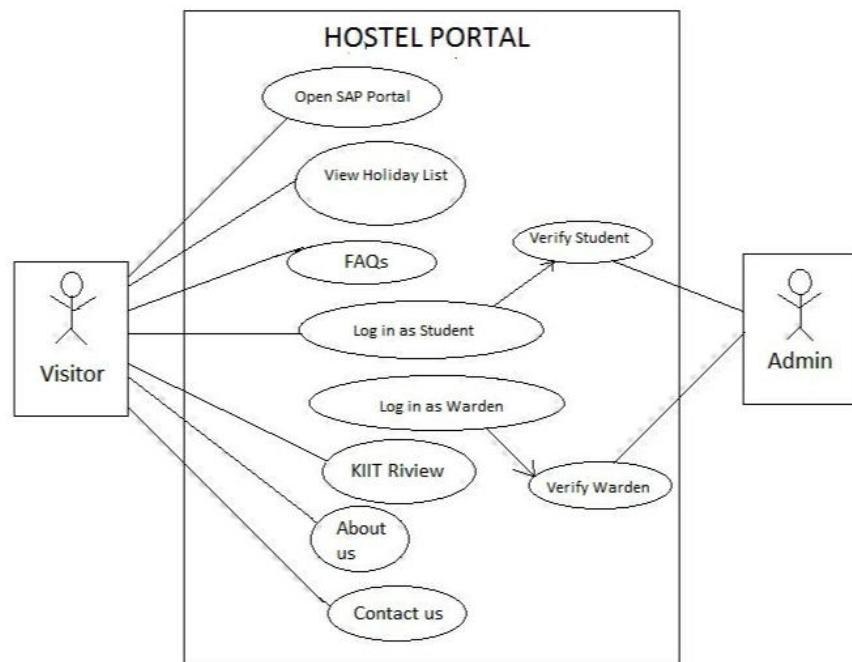


Figure 5.3: Hostel Portal for Student Login

### 5.3.2 WARDEN LOGIN

Now comes the part where the warden logs in first to see all the complaints submitted which is shown in the dashboard.

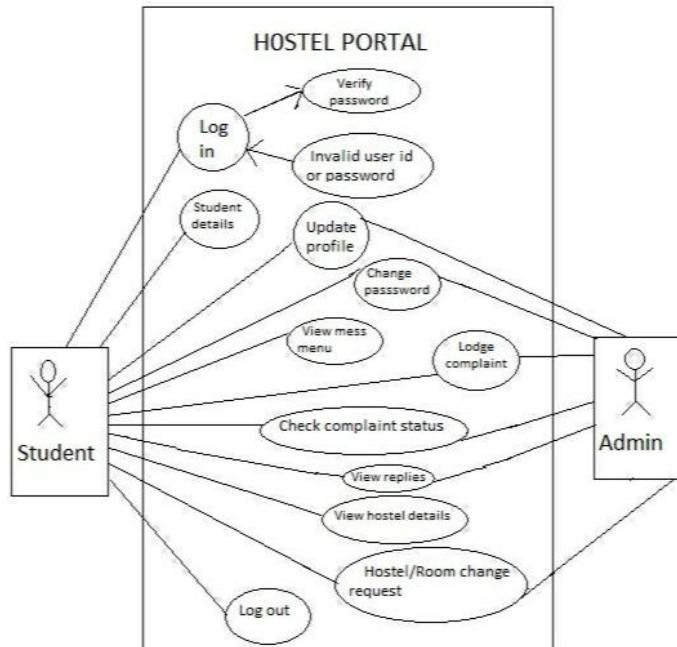


Figure 5.4: Hostel Portal for Warden Login

### 5.3.3 COMPLAINT STATUS

The wardens can give a suitable reply to the students after checking the complaints and can arrange possible solutions to their problems which will be notified to students.

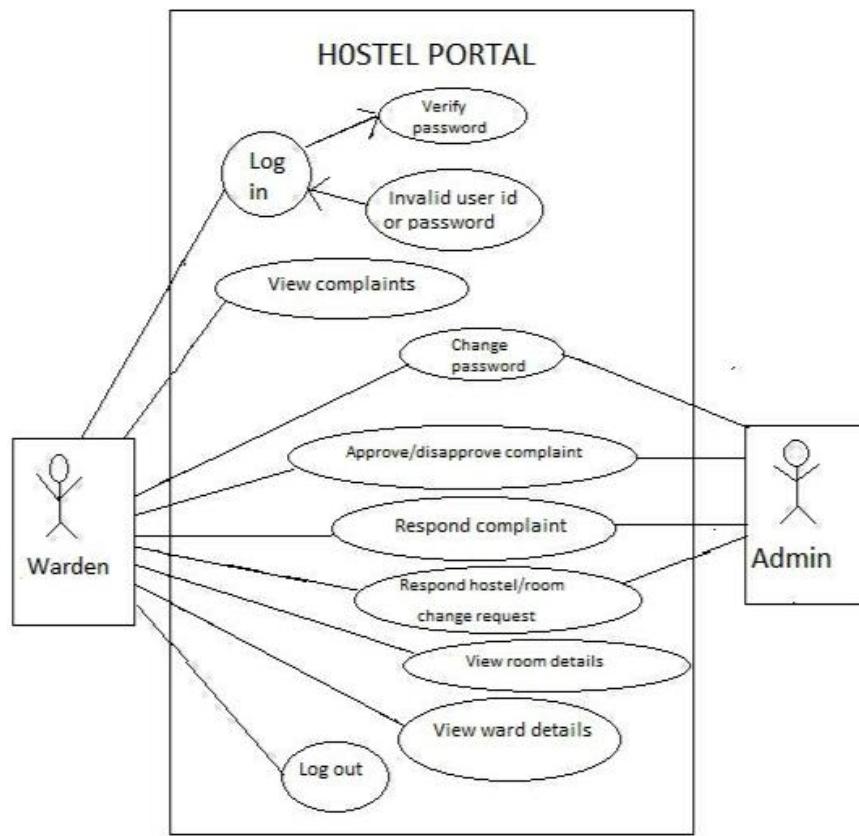


Figure 5.5: Hostel Portal for Complaint Status

## 5.4 DATABASE DESIGN

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed.

#### 5.4.1 STUDENT DETAILS DATABASE

| <b>student_rollno</b> | <b>student_name</b> | <b>student_gender</b> | <b>student_age</b> | <b>student_contact_number</b> | <b>student_dob</b> |
|-----------------------|---------------------|-----------------------|--------------------|-------------------------------|--------------------|
| 1705291               | SUPRIYO NANDI       | M                     | 20                 | 9749636258                    | 1999-06-10         |
| 1705298               | AVINAB SEN          | M                     | 21                 | 6370536552                    | 1999-01-19         |
| 1705353               | RISHI RAJ SAXENA    | M                     | 21                 | 9556901757                    | 1999-02-14         |
| 1705323               | MAYUKH MAJHI        | M                     | 21                 | 9126157829                    | 1999-10-23         |
| 1705418               | NABANKUR DEY        | M                     | 20                 | 8787638080                    | 1999-05-18         |

#### 5.4.2 LOGIN DETAILS DATABASE

| <b>student_id</b> | <b>student_rollno</b> | <b>student_password</b> |
|-------------------|-----------------------|-------------------------|
| 1                 | 1705353               | Rishiraj                |
| 2                 | 1705323               | Mayukh                  |
| 3                 | 1705298               | Avinab                  |
| 4                 | 1705470               | Supriyo                 |
| 5                 | 1705418               | Nabankur                |

#### 5.4.3 WARDEN LOGIN DATABASE

| <b>hostel_id</b> | <b>hostel_name</b> | <b>hostel_password</b> |
|------------------|--------------------|------------------------|
| KP9A             | KP-9A              | KP9A                   |
| KP9C             | KP-9C              | KP9C                   |

#### 5.4.4 HOSTEL DETAILS DATABASE

| <b>student_rollno</b> | <b>student_roomno</b> | <b>student_hostelname</b> | <b>student_name</b> | <b>student_contact</b> |
|-----------------------|-----------------------|---------------------------|---------------------|------------------------|
| 1705298               | B-104                 | KP-9C                     | Avinab              | 6370536552             |
| 1705353               | B-104                 | KP-9C                     | Mayukh              | 9126157829             |
| 1705323               | B-104                 | KP-9C                     | Rishiraj            | 9556901757             |

## CHAPTER 6

### SYSTEM TESTING

#### 6.1 TEST CASES AND TEST RESULTS

| Test ID | Test Case Title                    | Test Condition   | System Behavior   | Expected Result   |
|---------|------------------------------------|--|---|---|
| T01     | Validating user input              | Input of data in appropriate format  | The system takes specified input by the user e.g. int value for integer type.   | The system should take appropriate value from the user for the respective column.   |
| T02     | Validating form                    | Input of inappropriate data by the user in particular column.  | The system displays a pop-up alert window displaying the error message for the incorrect input  | The value of each input selected should match it's corresponding validation rules. E.g - Full Name input field cannot contain a digit.            |
| T03     | Validating login submit button     | Authorize the user typed username and password(check whether any user exists in the database with same username and password). | If a match is found,then the system will redirect to a web-page after login and SESSION gets created for a user.  | After being validated the system gets redirected to the login page and also the SESSION variable is activated for the user being logged in.       |
| T04     | Validating complaint submit button | Submission of the complaint form through HTTP via POST request and Storing the data in the database.                           | The system displays the pop-up alert window displaying an success message if form is submitted else it displays the error message(In case any input field is left empty). | The form should get submitted and the data must get stored in the database.In case of any empty mandatory input field error message is displayed. |

# CHAPTER 7

## IMPLEMENTATION

In this Application we have two types of user: *Student* and *Warden*. Student and Warden can Login to portal by their UserId and Password. The Student can see their profile ,hostel details and they can complain to warden via app to respective hostel wardens. And warden basically review that complain can take required actions.

### 7.1 CODE FOR LOGIN

#### ANDROID APP

```
loginWarden.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        if (ed_wardenId.getText().toString().equals("")) {
            Toast.makeText(MainLoginActivity.this, "Enter Roll No", Toast.LENGTH_SHORT).show();
        } else if (ed_warden_pass.getText().toString().equals("")) {
            Toast.makeText(MainLoginActivity.this, "Enter Password", Toast.LENGTH_SHORT).show();
        } else {

            final ProgressDialog progressDialog = new ProgressDialog(MainLoginActivity.this);
            progressDialog.setMessage("Please Wait..");

            progressDialog.show();

            str_userid = ed_wardenId.getText().toString().trim();
            str_password = ed_warden_pass.getText().toString().trim();

            StringRequest request = new StringRequest(Request.Method.POST, urlWarden,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        progressDialog.dismiss();
                        if (response.equalsIgnoreCase("logged in successfully")) {

                            ed_wardenId.setText("");
                            ed_warden_pass.setText("");
                            Intent i = new Intent(getApplicationContext(), WardenMainActivity.class);
                            i.putExtra("UserRoll", str_userid);
                            startActivity(i);
                            // startActivity(new Intent(getApplicationContext(), StudentHomeActivity.class));
                            Toast.makeText(MainLoginActivity.this, response, Toast.LENGTH_SHORT).show();
                        } else {
                            str_userid = null;
                            Toast.makeText(MainLoginActivity.this, response, Toast.LENGTH_SHORT).show();
                        }
                    }
                }, new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        progressDialog.dismiss();
                        Toast.makeText(MainLoginActivity.this, error.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }
            ) {
                @Override
                protected Map<String, String> getParams() throws AuthFailureError {
                    Map<String, String> params = new HashMap<String, String>;
                    params.put("hostel_id", str_userid);
                    params.put("hostel_password", str_password);
                    return params;
                }
            };

            RequestQueue requestQueue = Volley.newRequestQueue(MainLoginActivity.this);
            requestQueue.add(request);

        }
    });
});
```

```
loginBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String str_userid = ed_userid.getText().toString();
        if (ed_userid.getText().toString().equals("")) {
            Toast.makeText(MainLoginActivity.this, "Enter Roll No", Toast.LENGTH_SHORT).show();
        } else if (ed_password.getText().toString().equals("")) {
            Toast.makeText(MainLoginActivity.this, "Enter Password", Toast.LENGTH_SHORT).show();
        } else {

            final ProgressDialog progressDialog = new ProgressDialog(MainLoginActivity.this);
            progressDialog.setMessage("Please Wait..");

            progressDialog.show();

            str_userid = ed_userid.getText().toString().trim();
            str_password = ed_password.getText().toString().trim();

            StringRequest request = new StringRequest(Request.Method.POST, urlStud,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        progressDialog.dismiss();
                        if (response.equalsIgnoreCase("logged in successfully")) {

                            ed_userid.setText("");
                            ed_password.setText("");
                            Intent i = new Intent(getApplicationContext(), StudentHomeActivity.class);

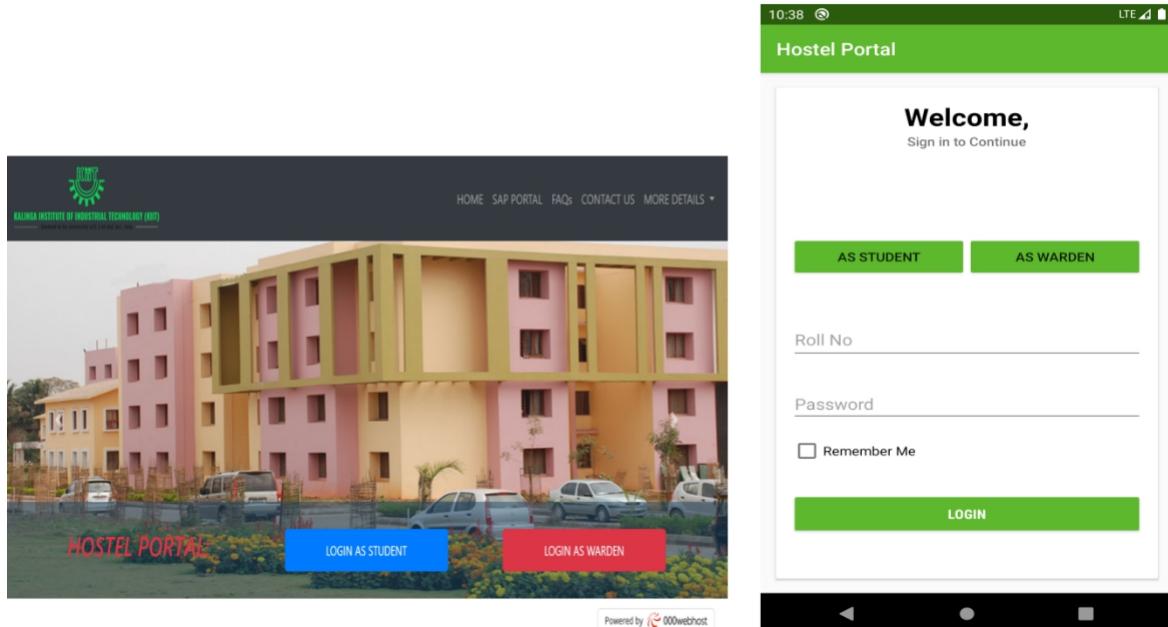
                            SharedPreferences preferences = getSharedPreferences("UserRoll", MODE_PRIVATE);
                            Sharedpreferences.Editor editor = preferences.edit();
                            editor.putString("str_roll", "" + str_userid);
                            editor.apply();

                            i.putExtra("UserRoll", str_userid);
                            startActivity(i);
                            // startActivity(new Intent(getApplicationContext(), StudentHomeActivity.class));
                            Toast.makeText(MainLoginActivity.this, response, Toast.LENGTH_SHORT).show();
                        } else {
                            str_userid = null;
                            Toast.makeText(MainLoginActivity.this, response, Toast.LENGTH_SHORT).show();
                        }
                    }
                }, new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        progressDialog.dismiss();
                        Toast.makeText(MainLoginActivity.this, error.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }
            ) {
                @Override
                protected Map<String, String> getParams() throws AuthFailureError {
                    Map<String, String> params = new HashMap<String, String>;
                    params.put("student_rollno", str_userid);
                    params.put("student_password", str_password);
                    return params;
                }
            };

            RequestQueue requestQueue = Volley.newRequestQueue(MainLoginActivity.this);
            requestQueue.add(request);

        }
    });
});
```

## **WEBSITE**



## 7.2 CODE FOR COMPLAINT

### ANDROID APP

```
public void ComplainRegister() {  
  
    final int complainType=complainInt;  
    final String compSub=CompDetails.getText().toString().trim();  
  
    StringRequest request=new StringRequest(Request.Method.POST, "https://hostelportal123.000webhostapp.com/android/RegisterComplain.php", new  
    @Override  
    public void onResponse(String response) {  
        if(response.equals("some error occurred"))  
            Toast.makeText(getApplicationContext(),"some error occurred",Toast.LENGTH_SHORT).show();  
        else  
        {  
            Toast.makeText(getApplicationContext(),response,Toast.LENGTH_SHORT).show();  
        }  
    }  
, new Response.ErrorListener() {  
    @Override  
    public void onErrorResponse(VolleyError error) {  
        Toast.makeText(getApplicationContext(), error.getMessage(), Toast.LENGTH_SHORT).show();  
    }  
}  
){  
    @Override  
    protected Map<String, String> getParams() throws AuthFailureError {  
        Map<String, String> params = new HashMap<String, String>();  
  
        params.put("complaint_type",""+complainInt);  
        params.put("complaint_subject",compSub);  
        params.put("complaint_student_name",arrayList.get(3));  
        params.put("complaint_student_rollno",arrayList.get(2));  
        params.put("complaint_hostel_name",arrayList.get(0));  
        params.put("complaint_roomno",arrayList.get(1));  
        Log.e("PARAMS",""+params);  
        return params;  
    }  
};  
RequestQueue requestQueue= Volley.newRequestQueue(getApplicationContext());  
requestQueue.add(request);  
}
```

### WEBSITE

```
<?php  
include "includes/student_header.php";  
if((!isset($_SESSION['student_rollno']) || !isset($_SESSION['student_password'])))  
{  
    ?>  
</php  
// session_start();  
?  
  
<?php  
if(!isset($_POST['complaint_submit']))  
{  
    $complaint_type=$_POST['complaint_type'];  
    $complaint_details=$_POST['complaint_details'];  
    $complaint_student_name=$_POST['complaint_student_name'];  
    $complaint_student_rollno=$_POST['complaint_student_rollno'];  
    $complaint_hostelname=$_POST['complaint_hostel_name'];  
    $complaint_roomno=$_POST['complaint_room_no'];  
  
    //uploading the image for the complaint  
    $post_image = $_FILES['image'][name];  
    $post_image_temp = $_FILES['image']['tmp_name'];  
  
    $complaint_number=rand(1000,9999);  
    $query="INSERT INTO complaint_details(complaint_subject,complaint_type,complaint_student_name,complaint_student_rollno,complaint_hostel_name,complaint_roomno,complaint_number) VALUES  
    ('$complaint_details','$complaint_type','$complaint_student_name','$complaint_student_rollno','$complaint_hostelname','$complaint_roomno','$complaint_number')";  
  
    move_uploaded_file($post_image_temp, "../images/$post_image");  
  
    $insert_complaint_query=mysql_query($connection,$query);  
    if($insert_complaint_query)  
    die("QUERY FAILED".mysql_error($connection));  
}  
?  
  
<div class="d-flex" id="wrapper">  
    <!-- Sidebar -->  
</php  
    include "includes/student_sidebar.php";  
    ?>  
    <!-- /sidebar-wrapper -->  
    <!-- Page Content -->  
    <div id="page-content-wrapper">  
        </p>  
        include "includes/student_navigation.php";  
        ?>  
        <br>  
        <center><button class="btn btn-primary" id="menu-toggle">Click Here for Student Details</button></center>
```

## 7.3 CODE FOR APPROVAL AND REJECTION OF COMPLAINTS (BY WARDENS)

```
private void ApproveComplain() {
    final String notId=WardenDashboardFragment.complaintsStack.get(position).getmComplaintNo();
    final String studRoll=WardenDashboardFragment.complaintsStack.get(position).getComplainStudentRoll();
    final String warden_name=ed_warden_name.getText().toString().trim();
    final String warden_message=ed_warden_msg.getText().toString().trim();
    final String complaint_status="Approved";
    final String compUrl="http://192.168.1.10:8080/SmartWarden/approveComplain.php";

    StringRequest request=new StringRequest(Request.Method.POST, compUrl, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            if(response.equals("Insertion Successful")){
                Toast.makeText(ComplainClickActivity.this,"Reply Sent",Toast.LENGTH_SHORT).show();
            }
            else{
                Toast.makeText(ComplainClickActivity.this,response,Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(ComplainClickActivity.this, error.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });
    //Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<String, String>();

        params.put("notification_id",notId);
        params.put("warden_name",warden_name);
        params.put("warden_message",warden_message);
        params.put("complaint_status",complaint_status);
        params.put("student_rollno",studRoll);
        Log.e("PARAMS",""+params);
        return params;
    }
}
RequestQueue requestQueue= Volley.newRequestQueue(ComplainClickActivity.this);
requestQueue.add(request);
}

private void RejectComplain(){
    final String notId=WardenDashboardFragment.complaintsStack.get(position).getmComplaintNo();
    final String warden_name=ed_warden_name.getText().toString().trim();
    final String warden_message=ed_warden_msg.getText().toString().trim();
    final String complaint_status="Rejected";

    StringRequest request=new StringRequest(Request.Method.POST, compUrl, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            if(response.equals("Insertion unsuccessful")){
                Toast.makeText(ComplainClickActivity.this,"Reply Sent",Toast.LENGTH_SHORT).show();
            }
            else{
                Toast.makeText(ComplainClickActivity.this,response,Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(ComplainClickActivity.this, error.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });
    //Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> params = new HashMap<String, String>();

        params.put("notification_id",notId);
        params.put("warden_name",warden_name);
        params.put("warden_message",warden_message);
        params.put("complaint_status",complaint_status);
        Log.e("PARAMS",""+params);
        return params;
    }
}
RequestQueue requestQueue= Volley.newRequestQueue(ComplainClickActivity.this);
requestQueue.add(request);
}
```

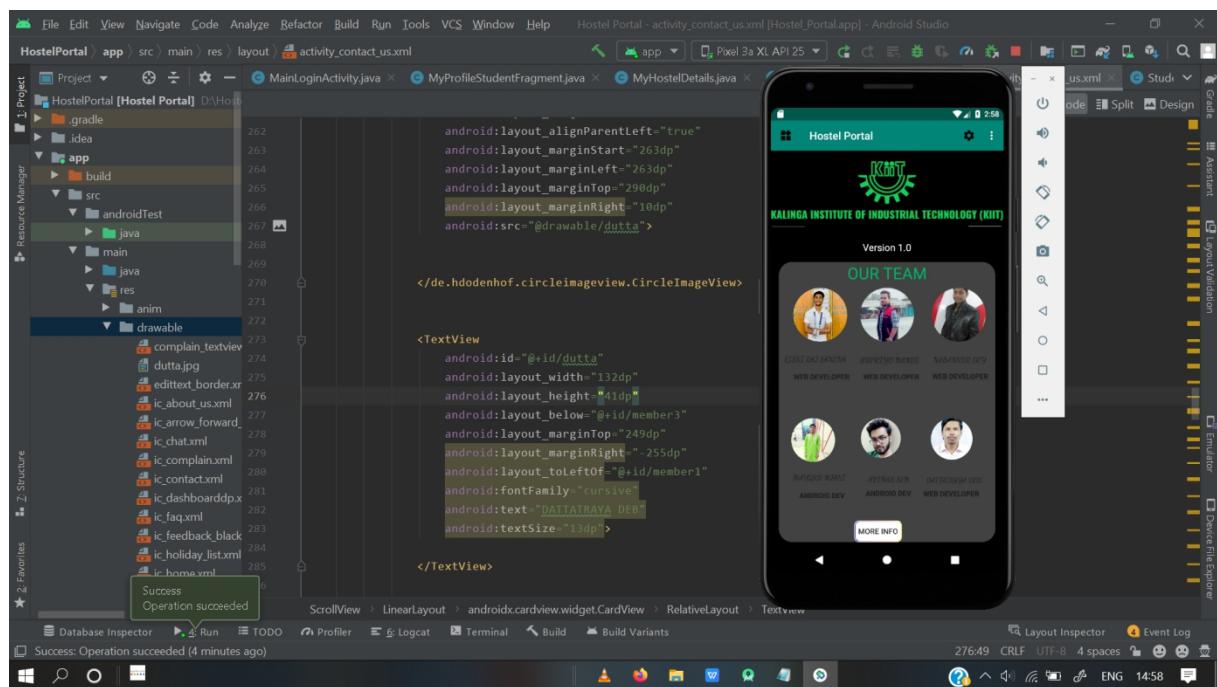
## 7.4 CODE FOR CONTACT US

```
File Edit Selection View Go Run Terminal Help • contactus.html - public_html - Visual Studio Code

EXPLORER contactus.html
OPEN EDITORS 1 UNSAVED contactus.html > html > body
PUBLIC.HTML contactus.html
db.php
footer.php
header.php
navigation.php
student.hostel.login...
student.hostel.logo...
warden.hostel.login...
warden.hostel.logo...
js
pages
includes
  footer.php
  header.php
  navigation.php
  contact_us.php
  holiday_list.php
  student_login_page...
  warden_login_page...
student_area
styling.folder
.htaccess
contactus.html 1
faqs.php
hostel_details.php
index.php
login.php
messmenu.php
retrHostelDetails.php
retrStudDetails.php
OUTLINE
Ln 129, Col 7 (249 selected) Spaces: 2 UFT-8 CRLF HTML Go Live R C

87 border-radius: 7px; "
88 }
89 </style>
90 </head>
91 <body>
92
93 <h2>Responsive Contact Section</h2>
94 <p>Resize the browser window to see the effect.</p>
95
96 <div class="container">
97   <div style="text-align:center">
98     <h2>Contact Us</h2>
99     <p>Swing by for a cup of coffee, or leave us a message:</p>
100    </div>
101    <div class="row">
102      <div class="column">
103
104        </div>
105        <div class="column">
106          <form action="action_page.php">
107            <label for="fname">First Name</label>
108            <input type="text" id="fname" name="firstname" placeholder="Your name..">
109            <label for="lname">Last Name</label>
110            <input type="text" id="lname" name="lastname" placeholder="Your last name..">
111            <label for="country">Country</label>
112            <select id="country" name="country">
113              <option value="australia">Australia</option>
114              <option value="canada">Canada</option>
115              <option value="usa">USA</option>
116            </select>
117            <label for="subject">Subject</label>
118            <textarea id="subject" name="subject" placeholder="Write something.. " style="height:170px"></textarea>
119            <input type="submit" value="Submit">
120          </form>
121        </div>
122
123      </div>
```

## 7.5 CODE FOR DARK MODE

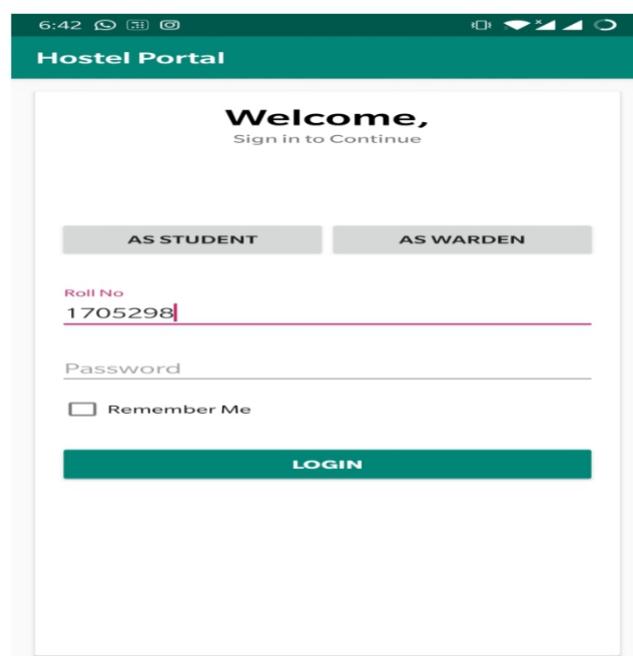
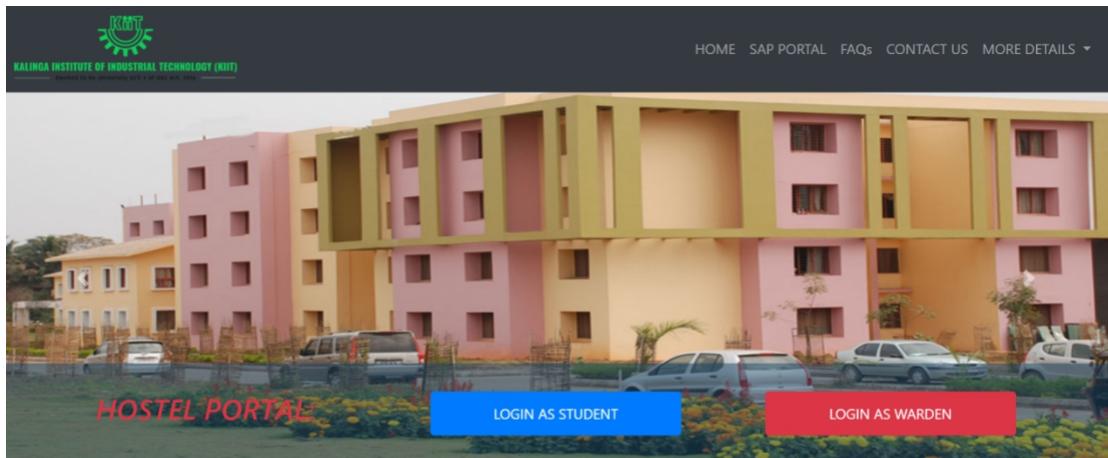


# CHAPTER 8

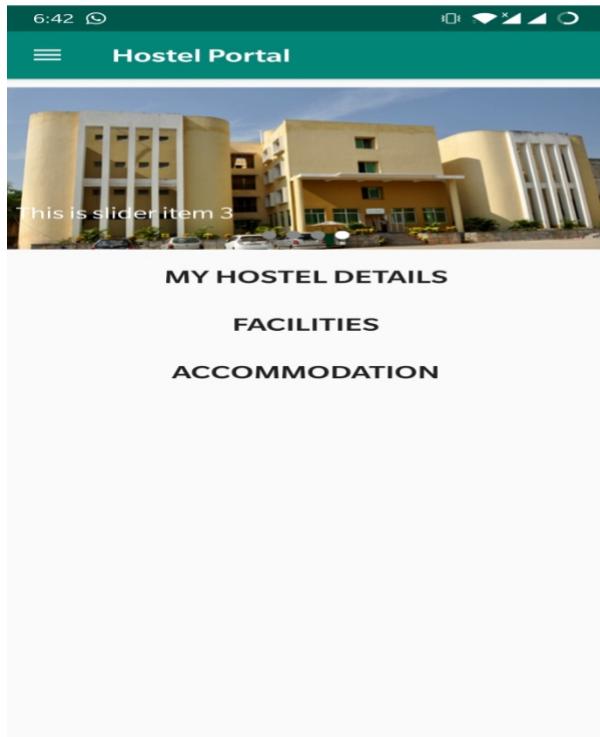
## SCREEN SHOTS OF PROJECT

### 8.1 WEBSITE AND APP SCREENSHOTS

#### 8.1.1 LOGIN SCREEN



## 8.1.2 HOME PAGE

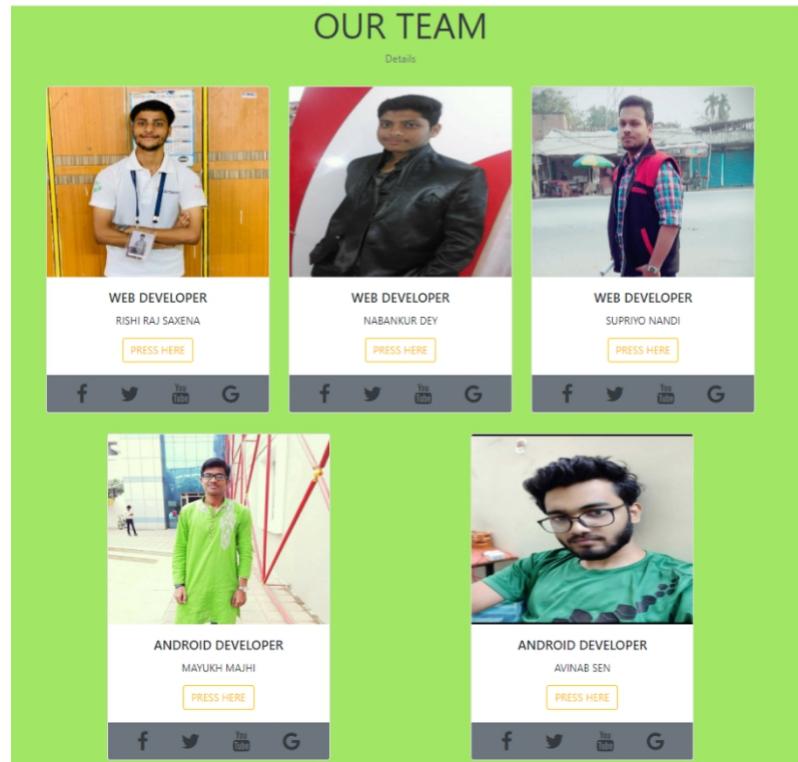


A screenshot of the KIIT Hostel Portal website. At the top, there is a navigation bar with links: "HOME", "SAP PORTAL", "HOLIDAY LIST", "FAQs", "CONTACT US", and "LOG OUT". The logo of "KALINJA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)" is displayed. On the left, a sidebar menu titled "Start Bootstrap" lists: "Home" (which is highlighted), "Notifications", "My Profile", "Mess Menu", "Complain", "My Hostel Details", "Facilities", and "Log Out". To the right of the sidebar, a button says "Click Here for Student Details". The main content area has a heading "Welcome to KIIT Hostel Portal" and a note about the responsive menu state.

The starting state of the menu will appear collapsed on smaller screens, and will appear non-collapsed on larger screens. When toggled using the button below, the menu will change.

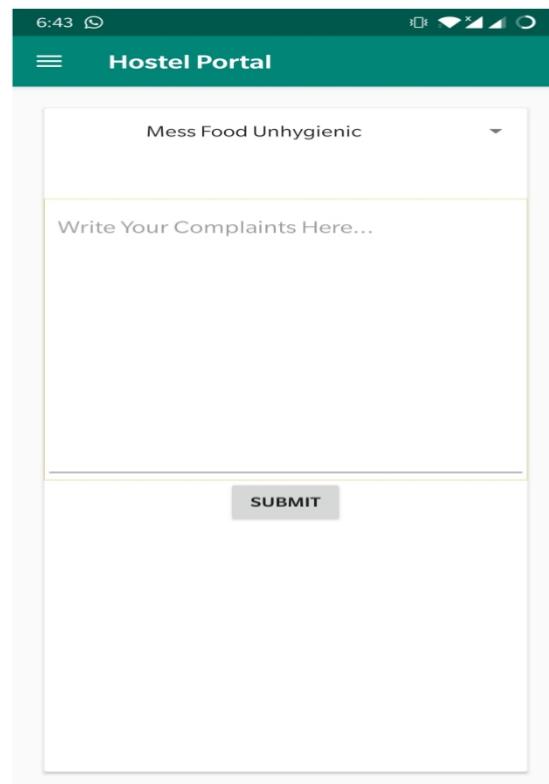
Make sure to keep all page content within the `#page-content-wrapper`. The top navbar is optional, and just for demonstration. Just create an `element` with the `menu-toggle` ID which will toggle the menu when clicked.

### 8.1.3 CONTACT SCREEN

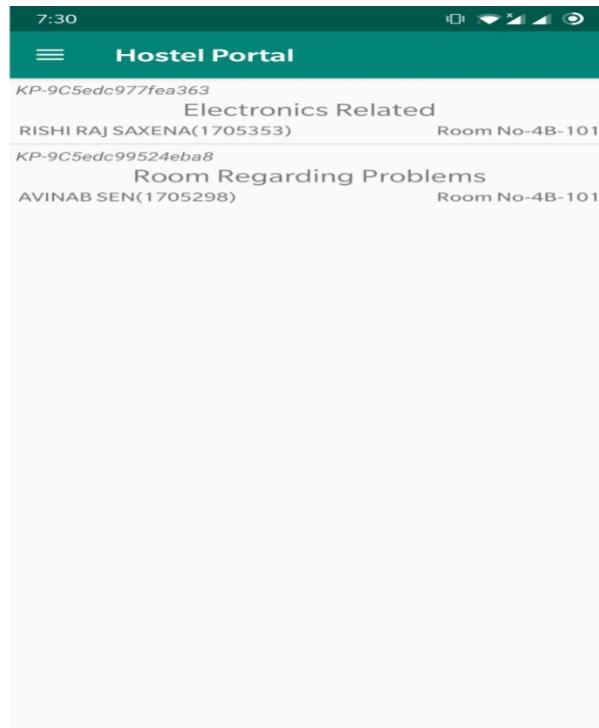


## 8.1.4 COMPLAIN BOX

The screenshot shows the 'Welcome to KIIT Hostel Portal' page. On the left is a vertical sidebar menu with options: Start Bootstrap, Home, Notifications, My Profile, Mess Menu, Complain (which is selected), My Hostel Details, Facilities, and Log Out. The main content area has a header 'Welcome to KIIT Hostel Portal' and a 'Click Here for Student Details' button. A dropdown menu for 'Complaint Type' is set to 'Mess Food Unhygienic'. Below it is a file upload field for 'Upload an image regarding the issue' with a placeholder 'Choose file No file chosen'. A large text area for 'Complaint Details' is labeled 'Enter your complaint here...'. At the bottom, there are fields for 'Student Name' (RISHI RAJ SAXENA), 'Student Roll No' (1705353), 'Hostel Name' (KP-9C), and 'Room No' (4B-101). A blue 'SUBMIT' button is at the bottom right.



## 8.1.5 DASHBOARD



The screenshot shows the KIIT Hostel Portal web application. On the left, there is a sidebar with the following menu items: "Start Bootstrap", "Home", "Dashboard", "Approved Complaints", and "Unapproved Complaints". The main content area has a dark header with the KIIT logo and the text "KALIGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)". Below the header, there is a button labeled "Click Here for Details". The main title is "Welcome to KIIT Hostel Portal". Below the title, there is a table with the following data:

| Complaint Id       | Student Name     | Roll No | Current Status | Seen Status | Details                         |
|--------------------|------------------|---------|----------------|-------------|---------------------------------|
| KP-9C5edc977fea363 | RISHI RAJ SAXENA | 1705353 | approved       | seen        | <a href="#">More details...</a> |
| KP-9C5edc99524eba8 | AVINAB SEN       | 1705298 | approved       | seen        | <a href="#">More details...</a> |

## 8.1.6 STUDENT PROFILE



The screenshot shows a web-based application for the Kalinga Institute of Industrial Technology (KIIT) Hostel Portal. On the left, there is a vertical sidebar menu with the following items: Start Bootstrap, Home, Notifications, My Profile, Mess Menu, Complain, My Hostel Details, Facilities, and Log Out. The main content area has a dark header bar with the KIIT logo and navigation links: HOME, SAP PORTAL, HOLIDAY LIST, FAQS, CONTACT US, and LOG OUT. Below the header, a blue button says "Click Here for Student Details". The main title "My Profile" is displayed in red. The student's profile information is listed in input fields:

- Student Name: RISHI RAJ SAXENA
- Gender: M
- Roll No: 1705353
- Age: 21
- Date of Birth: 14-02-1999

In the bottom right corner, there is a small text "Powered by 000webhost".

## 8.1.7 HOSTEL DETAILS



The screenshot shows the "Welcome to KIIT Hostel Portal" page. On the left, there is a vertical sidebar with the following menu items:

- Start Bootstrap
- Home
- Notifications
- My Profile
- Mess Menu
- Complain
- My Hostel Details
- Facilities
- Log Out

On the right, the main content area includes the KIIT logo and navigation links: HOME, SAP PORTAL, HOLIDAY LIST, FAQs, CONTACT US, and LOG OUT. A blue button labeled "Click Here for Student Details" is visible. The page displays the following information:

Student Name: RISHI RAJ SAXENA

Student Hostel Name: KP-9C

Student Room No: 2B-129

Student Contact: 9556901757

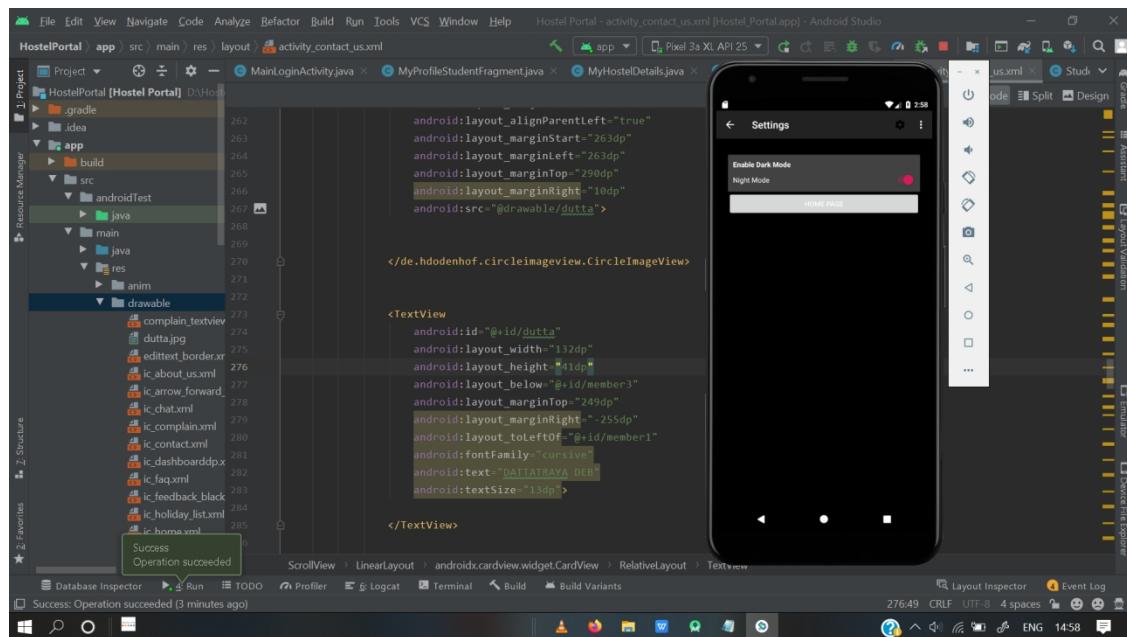
WARDEN DETAILS

| S.No | Warden Name | Hostel Name | Contact No |
|------|-------------|-------------|------------|
| 1    | jyoti sir   | KP-9C       | 1234567890 |
| 2    | madan sir   | KP-9C       | 2345678901 |

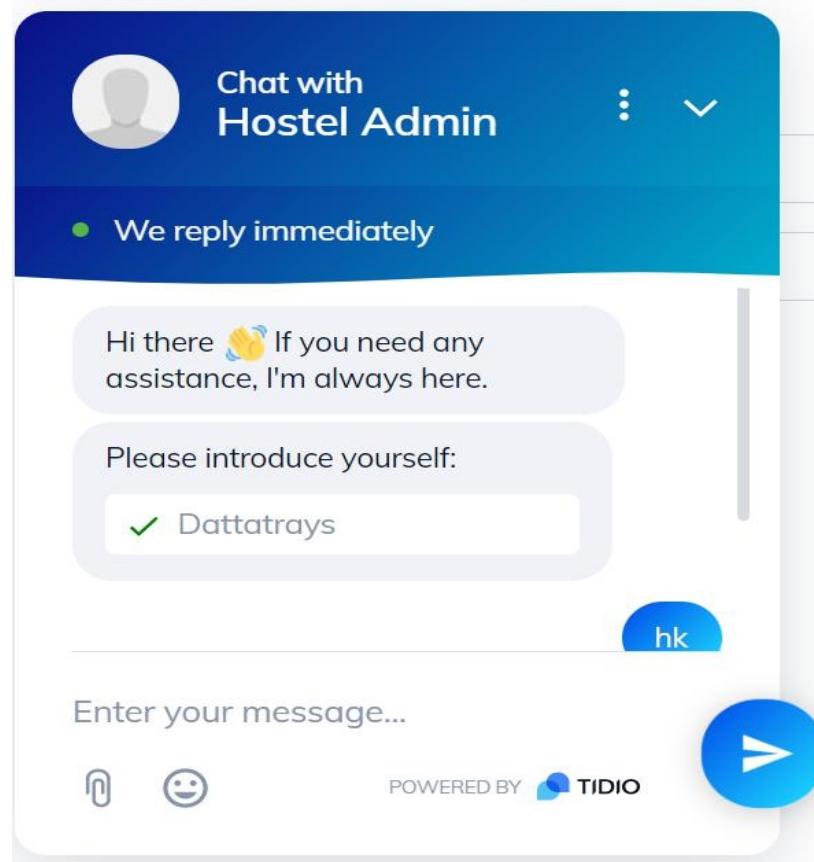
### 8.1.8 UPCOMING EVENTS

The screenshot shows a web browser window with the URL [hostelportal123.000webhostapp.com/eventForm.php](http://hostelportal123.000webhostapp.com/eventForm.php). The page title is "Upcoming Event Form". It contains two input fields: "Enter KIIT Email" and "Enter your Roll Number". Below these is a radio button group for "Select the Event" with options: "Basket Ball" (selected), "FootBall", and "Quiz competition". A blue "Register" button is at the bottom right. At the bottom left is a blue circular icon with a white phone receiver and a red notification badge, labeled "Chat with us".

### 8.1.9 DARK MODE



## 8.1.10 CHAT BOT



## 8.1.11 CONTACT US

### Responsive Contact Section

Resize the browser window to see the effect.

### Contact Us

Swing by for a cup of coffee, or leave us a message:

First Name

Last Name

Country

Subject

## **CHAPTER 9**

### **CONCLUSION AND FUTURE SCOPE**

#### **9.1 CONCLUSION**

The project, developed using PHP with SQL Server and various front end tools technologies which is based on the software's requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement. The online "Hostel Portal" is very helpful for hostel allotment and mess fee calculation and information regarding pupils accessing the premises. This hostel management software is designed for people who want to manage various activities in the hostel perspective. In past few years the number of educational institutions are increasing rapidly which resulted in number of hostels are also increasing for the accommodation of the students studying in a particular institution, hence there is a lot of strain on the person who is running the hostel since software's or websites are not usually used in this context. This particular project deals with the problems on managing a hostel resulting in eradicating the problems that 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.

#### **9.2 FUTURE SCOPE**

This Hostel Portal Management System would help the admins/end user with instantaneous updated information regarding any student/warden/worker in the current hostel premises hence reducing the opacity that has been a concern in the previous generation. The Portal System enables each student to put up a query at point of time (if any) regarding an issue he/she is facing with the warden obliged to look after the situation and find a solution to it. The transparency of the system helps in authorizing every situation without any compromise or faulty execution from misleading or unforeseen future.

1. Instantaneous Queries
2. Proficient agreement
3. Transparency in Co-operation

### **9.3 PROJECT PLANNING:**

This project idea was developed unanimously among the members to develop a project which would be very beneficial to the college , students and wardens. The idea was to create a hostel portal to solve common problems a hosteller faces in his daily hostel life

| <b>MEETINGS</b> | <b>SPRINTS</b>     | <b>COMPLETED DATE</b>    |
|-----------------|--------------------|--------------------------|
| 1.              | Project Discussion | 20-10-2019 to 27-11-2019 |
| 2.              | Project Designing  | 12-12-2019 to 26-04-2020 |
| 3.              | Initiation Process | 09-02-2020 to 18-02-2020 |
| 4.              | Front end          | 19-02-2020 to 05-06-2020 |
| 5.              | Back End           | 14-01-2020 to 10-05-2020 |
| 6.              | Report Preparation | 18-05-2020 to 01-06-2020 |

## **REFERENCES**

[1] Fundamentals of Database Systems,Ramez Elmasri and Shamkant B.Navathe.

[2] <https://www.php.net/manual/en/index.php>

## **INDIVIDUAL CONTRIBUTION REPORT:**

### **HOSTEL PORTAL**

AVINAB SEN  
1705298

#### **Abstract:**

“HOSTEL PORTAL” is a software developed for managing various activities that occurs in a hostel. For the past few years the number of educational institutions are increasing rapidly and so are the number of hostels for accommodation of the students studying in the institution. And hence there is a lot of strain on the person who runs the hostel and software's are not usually used in this context. To avoid such situation, this particular project has been built to deal with the problems on managing a hostel. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system which is more efficient, user-friendly, and more GUI oriented.

#### **Individual contribution and findings:**

This project consists of three phases: Project Selection, Implementation and Documentation. We came out with the idea of doing something innovative for our college (KIIT), so I and Mayukh thought of creating a user-friendly app, “HOSTEL PORTAL”, which could benefit the college students. In addition to the website, an app gives a better user-friendly experience and easy access to users. I worked on the front-end part of the app giving minute details to make it innovative and unique. I created the database tables of different activities to implement and retrieve information through JSON format and also the elements of the navigation drawer which leads to pages: Contact Us, FAQ, Student and Hostel Details, Mess Menu, etc. I had to learn the basics of android from online tutorials like fragments, intent, toast, different layouts and views. I collected the data for this project by clicking pictures of hostels, gathering information of wardens of different hostels along with their contact numbers.

### **Individual contribution to project report preparation:**

In project report preparation, I helped my team to make the cover page with proper margin and borders with our roll numbers and names. I made the UML diagrams in digital platform, “Paint”, which were originally drawn by my teammate, Nabankur, and I later added them in the document. I made the required database tables and its layout in the report. I added screenshots of the app of various working activities. I also helped my team in editing and formatting the project report.

### **Individual contribution for project presentation and demonstration:**

I helped with the final modification and editing of the presentation of our project to showcase our work during this whole project.

*Avinab Sen*

.....  
Signature of the Supervisor  
(Prof. N BIRAJA ISAC)

.....  
Signature of the student  
(AVINAB SEN)

# **HOSTEL PORTAL**

**DATTATRAYA DEB  
1705304**

## **Abstract:**

“HOSTEL PORTAL” is a software developed for managing various activities that occurs in a hostel. For the past few years the number of educational institutions are increasing rapidly and so are the number of hostels for accommodation of the students studying in the institution. And hence there is a lot of strain on the person who are runs the hostel and software’s are not usually used in this context. To avoid such situation, this particular project has been built to deal with the problems on managing a hostel. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system which is more efficient, user-friendly, and more GUI oriented.

## **Individual contribution and findings:**

This project consists of three phases: Project Selection, Implementation and Documentation. We came out with the idea of doing something innovative for our college (KIIT), so I and Mayukh thought of creating a user-friendly app, “HOSTEL PORTAL”, which could benefit the college students. In addition to the website, an app gives a better user-friendly experience and easy access to users. I worked on the front-end part of the app giving minute details to make it innovative and unique. I created the database tables of different activities to implement and retrieve information through JSON format and also the elements of the navigation drawer which leads to pages: Conatct Us, FAQ, Student and Hostel Details, Mess Menu, etc. I had to learn the basics of android from online tutorials like fragments, intent, toast, different layouts and views. I collected the data for this project by clicking pictures of hostels, gathering information of wardens of different hostels along with their contact numbers. I have contributed for Chatbot App.

### **Individual contribution to project report preparation:**

In project report preparation, I helped my team to make the cover page with proper margin and borders with our roll numbers and names. I made the UML diagrams in digital platform, “Paint”, which were originally drawn by my teammate, Nabankur, and I later added them in the document. I made the required database tables and its layout in the report. I added screenshots of the app of various working activities. I also helped my team in editing and formatting the project report. I also contributed to designing report for screenshots.

### **Individual contribution for project presentation and demonstration:**

Project Implementation and Screenshot of the Codes.

.....

Signature of the Supervisor  
(Prof. N BIRAJA ISAC)

Dattatraya Deb

Signature of the student  
(DATTATRAYA DEB)



# **HOSTEL PORTAL**

NABANKUR DEY  
1705418

## **Abstract:**

“HOSTEL PORTAL” is a software developed for managing various activities that occurs in a hostel. For the past few years the number of educational institutions are increasing rapidly and so are the number of hostels for accommodation of the students studying in the institution. And hence there is a lot of strain on the person who are runs the hostel and software's are not usually used in this context. To avoid such situation, this particular project has been built to deal with the problems on managing a hostel. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system which is more efficient, user-friendly, and more GUI oriented.

## **Individual contribution and findings:**

I have done the footer section for every page of the website (including the home page). Also I have done the login form for the warden where a warden of their respective hostel can login with their hostel id and password. The holiday list page is also designed by me. After logging in as an student a home page is needed where it will get redirected to which is designed by me. The hostel details page of the logged in student is also designed by me.

I along with Supriyo designed the complaint details page where the logged in warden can see the details of the complaint being registered by a student of their hostel. The front end for the page was designed by me whereas the server side code was written by Supriyo for the same. Also I along with Rishi designed the login form whose front end was being done by me and the login redirect and new session creation was done by Rishi.

### **Individual contribution to project report preparation:**

In project report preparation, I helped my team to make the Software requirements Specification (SRS) and also the Requirements Analysis as well. Also I have drawn the original UML diagrams which were further being converted into a digital diagram by my teammate, Avinab.

### **Individual contribution for project presentation and demonstration:**

Need of the project and it's future scope.



.....  
Signature of the Supervisor  
(Prof. N BIRAJA ISAC)

.....  
Signature of the student  
(NABANKUR DEY)

# **HOSTEL PORTAL**

SUPRIYO NANDI  
1705470

## **Abstract:**

“HOSTEL PORTAL” is a software developed for managing various activities that occurs in a hostel. For the past few years the number of educational institutions are increasing rapidly and so are the number of hostels for accommodation of the students studying in the institution. And hence there is a lot of strain on the person who runs the hostel and software's are not usually used in this context. To avoid such situation, this particular project has been built to deal with the problems on managing a hostel. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system which is more efficient, user-friendly, and more GUI oriented.

## **Individual contribution and findings:**

I have designed the header and the navigation section of the home page of the website. Also I have designed the FAQs page for the website. Login form for the student was also designed by me where the student needs to enter his college roll number along with the password. After being logged in as a student the “My Profile” page is needed which displays the information about the student and is being designed by me.I have also designed the mess menu page to display the mess menu for the hostels.

I along with Rishi designed a tab for the unapproved complaints just like the approved complaints being designed by him.In this tab all the complaints being unapproved by the warden will be displayed.Also I designed the server side code for the complaint details along with Nabankur who designed the front end of the page.

### **Individual contribution to project report preparation:**

In project report preparation, I helped my team to make the “INTRODUCTION” section which contains a brief introduction of the project. Also I made the “LITERATURE SURVEY” for the project report. At last I made the Conclusion and Reference of the project report

### **Individual contribution for project presentation and demonstration:**

Literature survey and it's requirements in the future.

*Supriyo Nandi*

.....  
Signature of the Supervisor  
(Prof. N BIRAJA ISAC)

.....  
Signature of the student  
(SUPRIYO NANDI)

# PLAGIARISM REPORT

The screenshot shows a web-based plagiarism checker interface. At the top, the URL https://searchenginereports.net/plagiarism-checker is displayed. Below the header, the word "Result" is centered above three performance metrics: "Checked 100%", "Plagiarism 15%", and "Unique 85%". Each metric is accompanied by a colored icon (blue for checked, red for plagiarism, green for unique). Below these metrics are three navigation buttons: "Sentence Wise Results", "Matched Sources", and "Document View". The main content area displays four rows of results, each consisting of a status indicator (green "Unique") and a snippet of text. The snippets are as follows:

|        |  |
|--------|--|
| Unique | Keywords: Hostel Portal, Management, Student Allocation, Data Consistency, Upcoming events, Chat bot, C...   |
| Unique | We have got seventeen hostels in our university, which consist of eleven boy's hostel and six girl's hostel. |
| Unique | All these hostels at present are managed manually by the hostel office.                                      |
| Unique | The Registration form verification to the different data processing are done manually.                       |