

# FiveStarHostels



**Session 2023 – 2027**

**Submitted by:**

ALI HAMZA	2023-CS-174
MIAN SAAD TAHIR	2023-CS-62

**Supervised by:**

Dr. Amjad Farooq

**Course:**

CSC-104 Database Systems

**Department of Computer Science**

**University of Engineering and Technology**

**Lahore Pakistan**

## Contents

<b>1.1 About FIVE STAR HOSTELS .....</b>	<b>4</b>
<b>1.1.1 Why Five Star Hostels? .....</b>	<b>4</b>
<b>1.1.2 Expectations from Five Star Hostel.....</b>	<b>4</b>
<b>1.1.3 Contribution towards CS .....</b>	<b>4</b>
<b>1.2 User Types on FiveStar Hostel.....</b>	<b>5</b>
<b>1.2.1 Warden (Owner) .....</b>	<b>5</b>
<b>1.2.2 Resident Tutor .....</b>	<b>5</b>
<b>1.2.3 Student .....</b>	<b>5</b>
<b>1.3 Functional Requirements of FiveStarHostel .....</b>	<b>6</b>
<b>1.4 ERD Model .....</b>	<b>9</b>
<b>1.5 Wireframes:.....</b>	<b>10</b>
<b>1.5.1 Basic Features (For All types of Users):.....</b>	<b>10</b>
<b>1.5.1.1 Startup Face: .....</b>	<b>10</b>
<b>1.5.1.2 Log Out .....</b>	<b>12</b>
<b>1.5.2 User Type (Warden):.....</b>	<b>13</b>
<b>1.5.2.1 Warden Menu.....</b>	<b>13</b>
<b>1.5.2.2 Allot Student.....</b>	<b>13</b>
<b>1.5.2.3 Manage Rooms .....</b>	<b>14</b>
<b>1.5.2.4 Manage Hostel .....</b>	<b>14</b>
<b>1.5.2.5 Add Resident Tutor .....</b>	<b>15</b>
<b>1.5.2.6 Remove RT and Student.....</b>	<b>15</b>
<b>1.5.2.8 Rules .....</b>	<b>16</b>
<b>1.5.3 User Type (Resident Tutor) .....</b>	<b>16</b>
<b>1.5.3.1 Rt Menu .....</b>	<b>16</b>
<b>1.5.3.2 Hostel Details .....</b>	<b>17</b>
<b>1.5.3.3 Check Complains .....</b>	<b>17</b>
<b>1.5.3.4 Leave Hostel .....</b>	<b>17</b>
<b>1.5.4 User Type (Student).....</b>	<b>18</b>
<b>1.5.4.1 Student Menu .....</b>	<b>18</b>
<b>1.5.4.1 Student Profile.....</b>	<b>18</b>
<b>1.5.4.2 Complains .....</b>	<b>19</b>

1.5.4.3 Change Password .....	19
1.6 Queries .....	20
1.6.1 Complaints.....	20
1.6.2 Dwellers.....	20
1.6.3 Hostels .....	21
1.6.4 Rooms.....	21
1.6.5 Resident Tutors .....	22
1.6.6 Rules .....	22
1.6.7 Students.....	22
1.7 Future Direction.....	23
Figure 1 FiveStar Main Page .....	10
Figure 2 FiveStar Information .....	11
Figure 3 Sign In .....	11
Figure 4 Sign Up.....	12
Figure 5 Log Out.....	12
Figure 6 warden menu .....	13
Figure 7 Allot Student.....	13
Figure 8 Manage Rooms .....	14
Figure 9 Manage Hostel.....	14
Figure 10 AddRT .....	15
Figure 11 Remove Rt and Student .....	15
Figure 13 Rules .....	16
Figure 14 RT Menu.....	16
Figure 15 Hostel Details .....	17
Figure 16 Check Complains.....	17
Figure 17 Leave .....	17
Figure 18 Student Menu.....	18
Figure 19 Student Profile .....	18
Figure 20 Complains.....	19
Figure 21 Change Password.....	20

## 1.1 About FIVE STAR HOSTELS

The **FIVE STAR HOSTELS** is a user-friendly platform with distinct logins for students, resident tutors, and senior wardens. It streamlines hostel operations by allowing students to manage profiles, request leave, and submit complaints. Resident tutors oversee students in their respective hostels, while senior wardens coordinate across all hostels, ensuring efficient management.

### 1.1.1 Why Five Star Hostels?

The Five Star Hostel is a preferred choice due to its user-friendly interface, which simplifies administrative tasks, such as room assignments, leave requests, and complaint submissions. It streamlines processes, saving time and reducing manual effort for students and staff. Effective communication through notifications and updates keeps all stakeholders informed. Senior wardens benefit from centralized control over multiple hostels, facilitating efficient management. The app contributes to an enhanced hostel experience by offering a structured and organized platform.

### 1.1.2 Expectations from Five Star Hostel

Users can expect an efficient and user-friendly experience with the Hostel Management App. The platform simplifies various administrative tasks, such as room assignments, leave requests, and complaint submissions, reducing manual work and saving time. Effective communication is facilitated through timely notifications and updates, ensuring that students, resident tutors, and senior wardens are well-informed. With centralized control, senior wardens can oversee multiple hostels, contributing to more organized and effective management. The app's structured and organized platform enhances the overall hostel experience for all users. Additionally, access to student data, leave records, and complaint history simplifies issue resolution and trend monitoring, promoting transparency and a smoother living environment.

### 1.1.3 Contribution towards CS

Five Star Hostels make substantial contributions to the field of computer science by incorporating various principles and techniques. These applications leverage optimization algorithms for efficient room allocation, a fundamental aspect of computer science that involves balancing multiple factors to arrive at the most optimal solution. Additionally, the integration of robust security measures, such as encryption and secure authentication processes, addresses the critical domain of cybersecurity, emphasizing the importance of safeguarding personal data within these systems. The development of user-friendly interfaces within hostel apps follows principles of Human-Computer Interaction (HCI) and user experience design, fostering advancements in the application of HCI methodologies. These apps also play a role in database management, handling extensive student information and transaction records, contributing to the broader field of database systems and data modeling. The use of machine learning algorithms for predictive analysis further extends their impact, providing insights into student behavior and resource utilization, thereby contributing to the evolving landscape of artificial intelligence and machine learning in computer science.

## 1.2 User Types on FiveStar Hostel

FiveStar Hostel provides different access for different type of users. The users are divided into three categories, Students, Resident tutor and Senior Warden (Admin). Each user type has access to different types of command related to their need and requirements. User can login and verify their type by inputting the issued username and password for authentication. Furthermore, each client type user has different database connected to it. So that data is stored and kept for specific person.

The hierarchy and functionality of user types is as under: -

### 1.2.1 Warden (Owner)

The Warden Profile functionality encompasses a comprehensive suite of features, allowing wardens to efficiently oversee hostel operations. Wardens can check the availability of students, allot rooms, manage room occupancy, and access detailed student data. The module includes tools for statistical analysis, presenting graphical representations of hostel statistics for informed decision-making. Wardens can also take actions such as removing students or resident tutors, adding new resident tutors, making announcements, displaying and managing rules, changing , and logging out. This multifaceted functionality equips wardens with the necessary tools to maintain order, address issues promptly, and ensure the overall well-being of the hostel community.

### 1.2.2 Resident Tutor

The Resident Tutor (RT) Profile module is designed to empower RTs with tools for effective hostel management. RTs can conveniently view the list of students assigned to them, set and enforce rules, and access the mess menu to facilitate a smooth living experience for the residents. Furthermore, the RT can stay informed about announcements, hostel events, and rules, enhancing communication and rule adherence. RTs can play a pivotal role in resolving complaints, making announcements, reviewing student complaints, and updating their passwords. This feature-rich functionality aims to streamline the RT's responsibilities and foster a positive and organized living environment.

### 1.2.3 Student

The Student Profile functionality caters to the diverse needs of hostel residents. Students can effortlessly navigate through options such as checking their profile details, submitting complaints or requests for issue resolution, reviewing and understanding hostel rules, and staying updated on notices and announcements. Additionally, students have quick access to the mess menu, hostel events calendar, and can efficiently check replies to their submitted complaints. The module further allows students to mark their attendance, pay their mess bills, change their passwords securely, and log out seamlessly, promoting a user-centric and interactive platform for managing various facets of hostel life.

### 1.3 Functional Requirements of FiveStarHostel

Some of the functional requirements expected from FiveStarHostel are as under

User Type	Required Functions to be performed	Result of Action Performed
<b>1 – Warden</b>	Manage Hostels	He can add or remove the hostel of his own choice.
	Allot Student	Allot a room to the students in their respective hostels according to their gender.
	Manage Rooms	He can add or remove the room of his own choice.
	Manage rules	Set the rules for resident tutors and students as well as he can remove announcements
	Add Rt	He can add Rt and assign him hostel according to his respective gender.
	Remove RT and Student	Remove the rt and student and his respective data from all associated classes.
	Log out	Warden can log out from his menu and going back to login Menu.

<b>2-Resident Tutor</b>	View assigned students	Display all the students that are assigned to his hostel.
	Manage rules	Set the rules for resident tutors and students as well as he can remove announcements
	Leave Hostel	Leave Hostel and his all associated Data is deleted.
	Check complains	Display complains from only those students which are assigned to his hostel
	Log out	Rt can log out from his menu and going back to login Menu.
<b>3-Student</b>	Check Profile	Student can check his profile which includes about his details of room and hostel.
	Manage Complains	Set the Complains to resident tutors As well as he can remove announcements
	Check rules	Display the rules that are implemented by both warden and Rt.
<b>3-Student</b>	Manage Mess Bill	

Leave Room

Leave Room and his all associated Data is deleted.

Change password

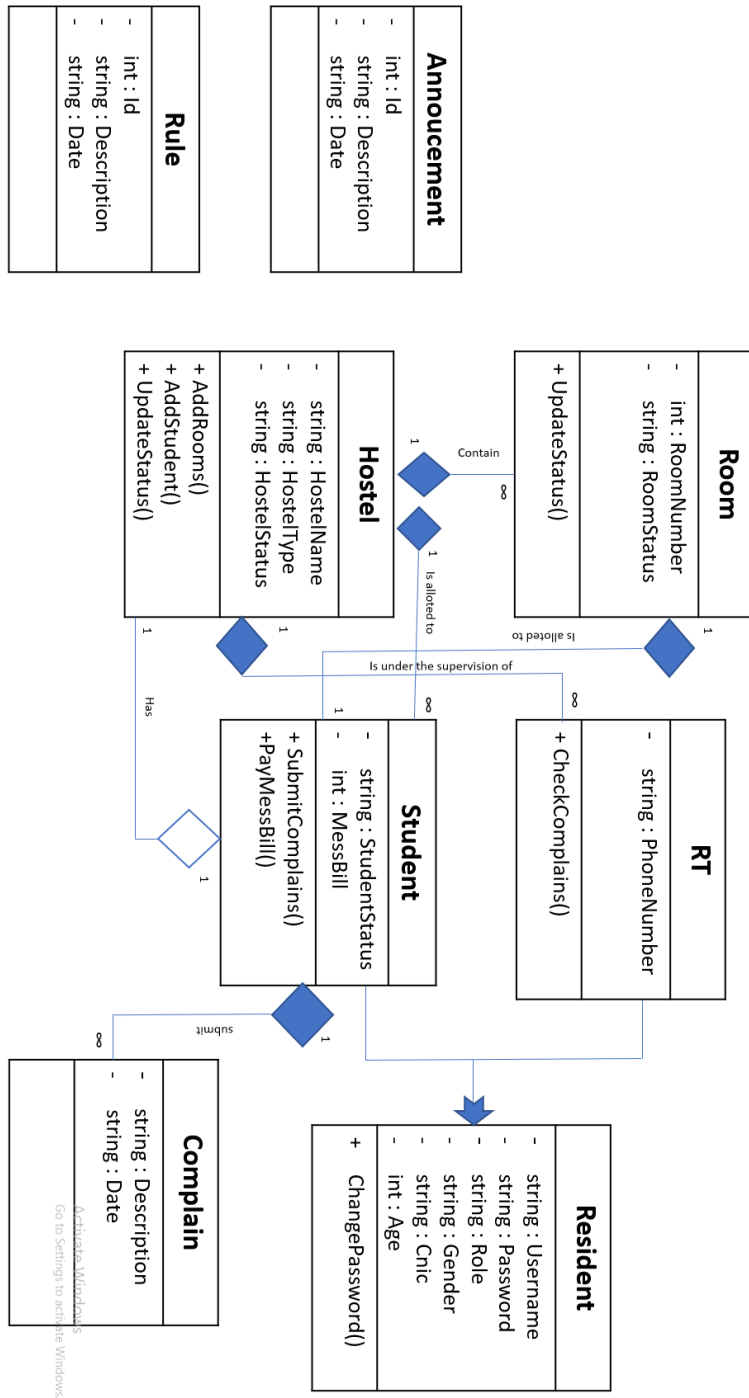
Change the password after verification of old password.

Log out

Student can log out from his menu and going back to login Menu.



## 1.4 Class Responsibility Collaborator Model



## 1.5 Wireframes:

The following is the wireframe of FiveStarHostel Displayed in Command Line Interface:

### 1.5.1 Basic Features (For All types of Users):

#### 1.5.1.1 Startup Face:

Startup interface is filled with animations to provide a premium experience to user about using the application.

##### 1.5.1.1.1 Main Header

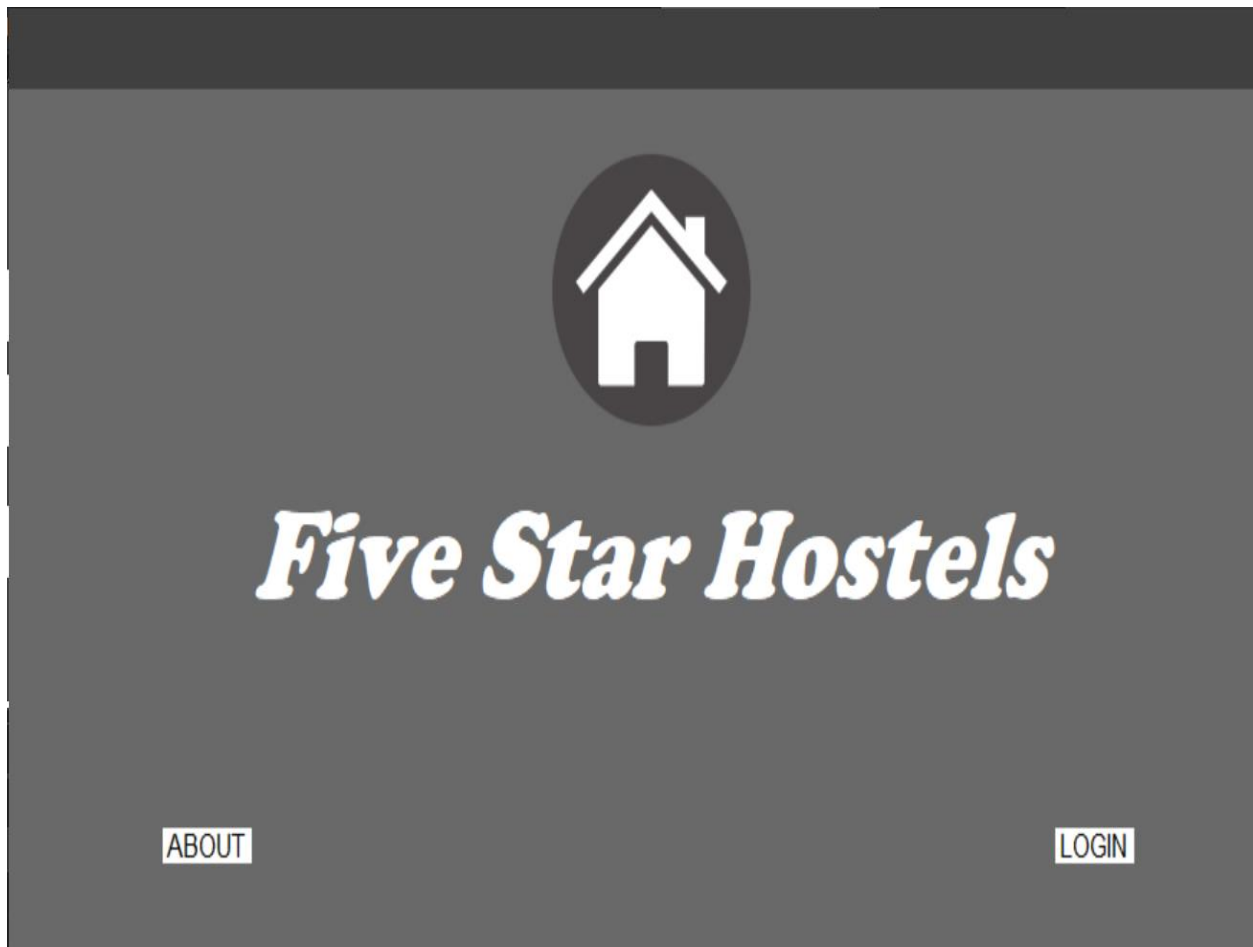


Figure 1 FiveStar Main Page

##### 1.5.1.1.2 FiveStarHostel Information



Figure 2 Five Star Information

### 1.5.1.1.3 Sign In

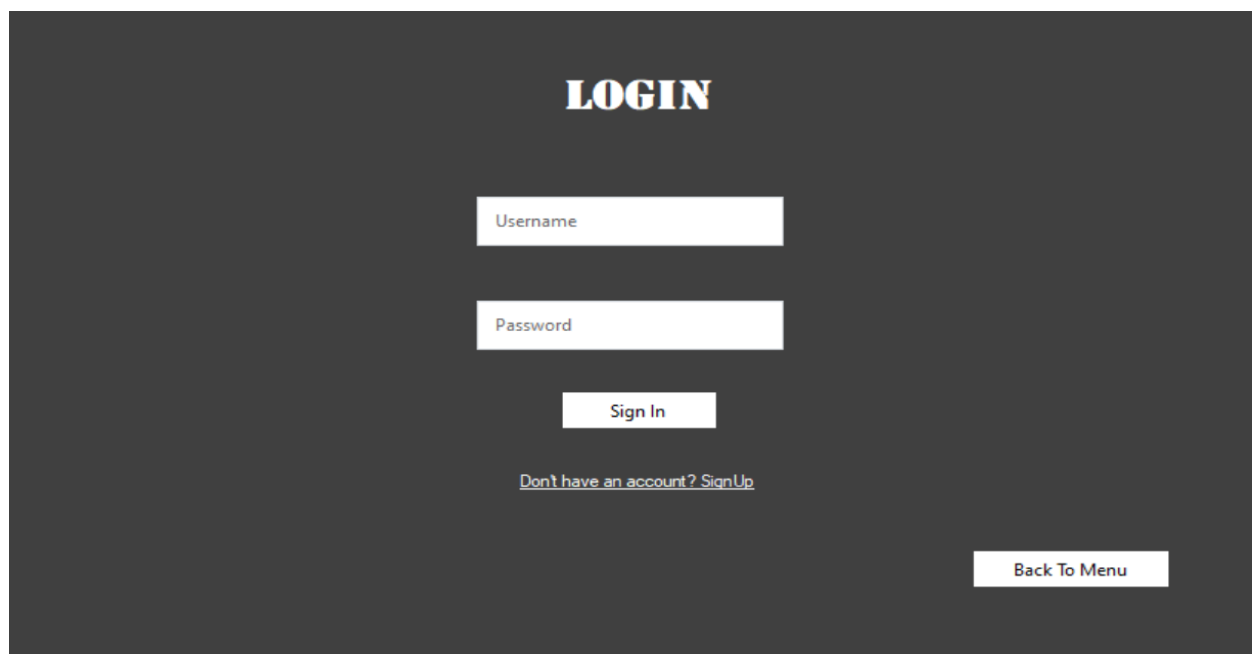


Figure 3 Sign In

### 1.5.1.1.4 Sign Up



A dark-themed sign-up form titled "SIGN UP" in a bold, serif font. The form contains seven input fields stacked vertically: Username, Password, Age, CNIC, Gender, Role, and Confirm Password. Each field is a white rounded rectangle with a small blue cursor icon on the left. Below the fields are two buttons: "Register" on the left and "Reset" on the right, both in a light gray box with dark text.

Figure 4 Sign Up

### 1.5.1.2 Log Out



A dark-themed confirmation dialog titled "ARE YOU SURE TO LOG OUT?" in a bold, serif font. Below the title are two buttons: "Yes" on the left and "No" on the right, both in a light gray box with dark text.

Figure 5 Log Out

## 1.5.2 User Type (Warden):

### 1.5.2.1 Warden Menu

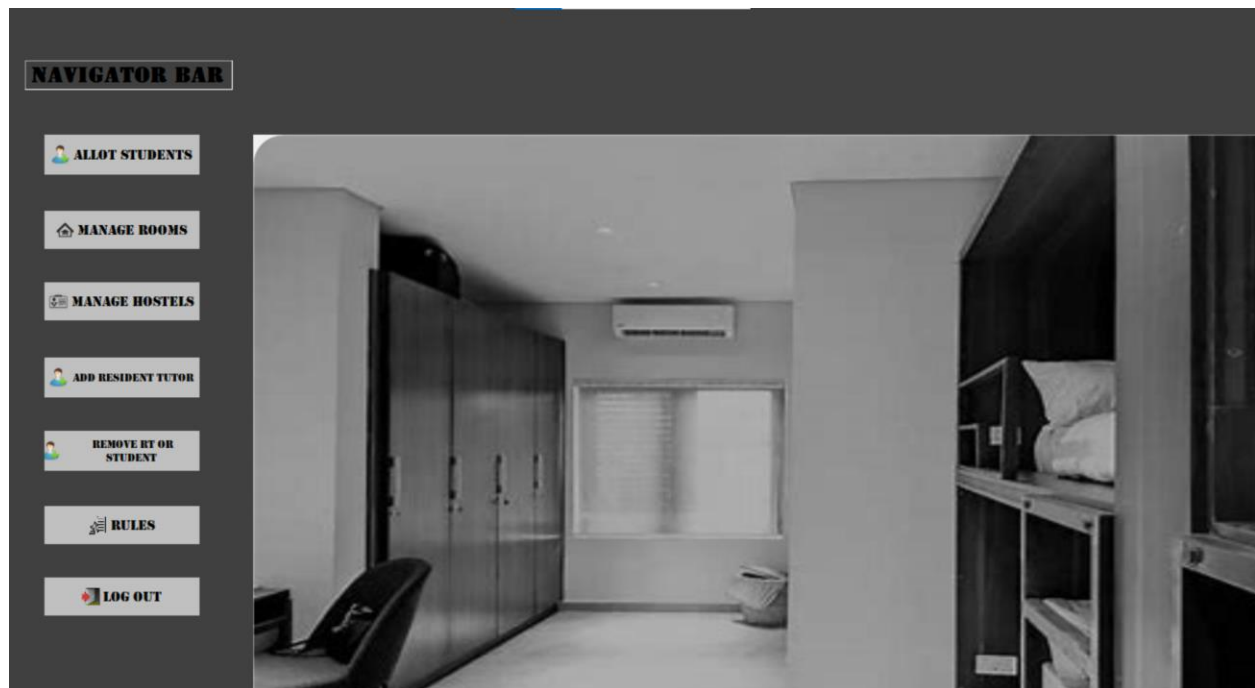


Figure 6 warden menu

### 1.5.2.2 Allot Student

**ALLOT STUDENT HOSTEL**

Select Student:  Select Hostel:

SelectRoom:

**Alloted Students**

AliHamza	male	32334	22	khalid hall	1
ali	male	33232	2	Zubair	1

Figure 7 Allot Student

### 1.5.2.3 Manage Rooms

**ADD**

Enter Room Number  Select Hostel Eidhi

**REMOVE ROOM**

Select Room  Select Hostel ayesha

**ALL AVAILABLE ROOMS**

1	Vacant	ayesha

Figure 8 Manage Rooms

### 1.5.2.4 Manage Hostel

**ADD NEW HOSTEL**

Enter Hostel Name  Select Hostel Type

**REMOVE HOSTEL**

Hostel Name Eidhi

**ALL HOSTEL**

Eidhi	Boys	Checked
ayesha	Girls	Checked
fatima	Girls	Unchecked
khalid hall	Boys	Unchecked
Zubair	Boys	Unchecked

Figure 9 Manage Hostel

### 1.5.2.5 Add Resident Tutor

**ADD**

Hostel

**ALL RTS**

bilal	female	3333333333333	2	ayesha	03147714171
bilal	male	3232323232323	3	Eidhi	03232323232

Figure 10 AddRT

### 1.5.2.6 Remove RT and Student

**REMOVE**

Enter Student

**REMOVE**

Enter Rt

Figure 11 Remove Rt and Student

### 1.5.2.8 Rules

**RULES AND**

Enter a new Rule here

Date  
 Friday, April 26, 2024

Add

**DELETE**

Give Rule Id

▼

Remove

All Rules

2	Saturday, February 8, 2020	Kisay ne munshi nu gaalan ni kdnia
2	Friday, April 26, 2024	bnday bno oye
3	Friday, April 26, 2024	koi bhi msla ho,bhai hazir ha
4	Friday, April 5, 2024	fees time pr diya kro bhai
5	Friday, April 26, 2024	kisi ny bilal ko tajnbg

[Back To Menu](#)

Figure 12 Rules

### 1.5.3 User Type (Resident Tutor)

#### 1.5.3.1 Rt Menu



Figure 13 RT Menu



### 1.5.3.2 Hostel Details

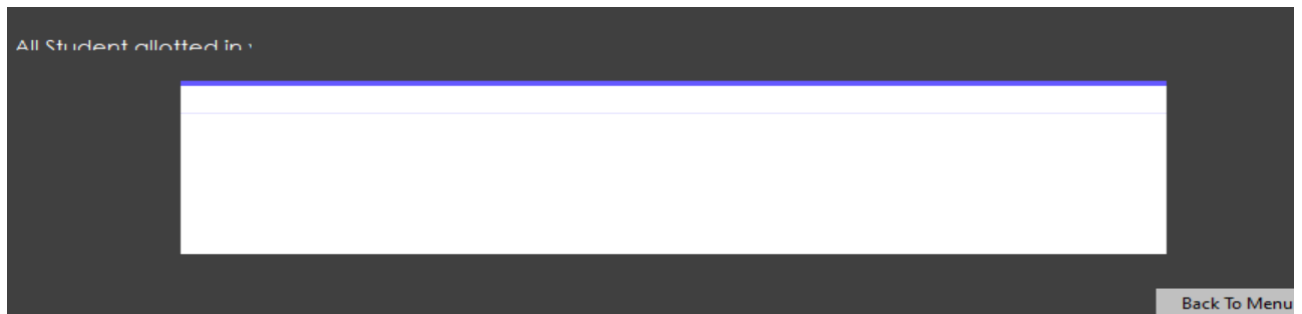


Figure 14 Hostel Details

### 1.5.3.3 Check Complains

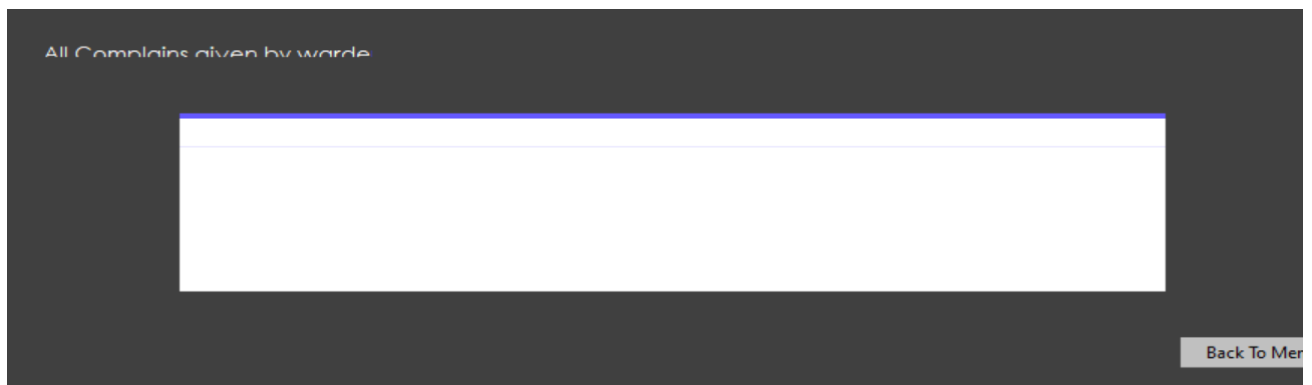


Figure 15 Check Complains

### 1.5.3.4 Leave Hostel

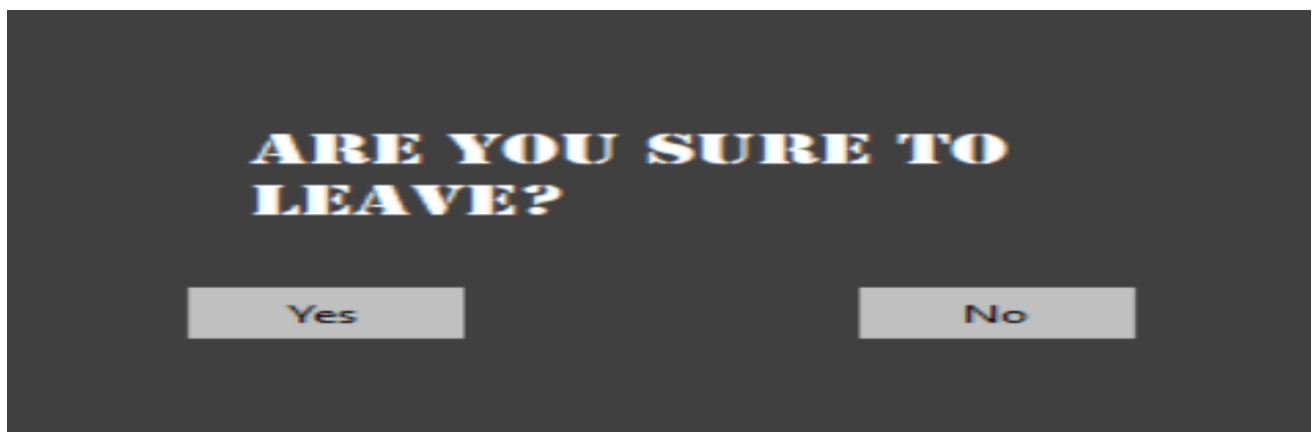


Figure 16 Leave

## 1.5.4 User Type (Student)

### 1.5.4.1 Student Menu



Figure 17 Student Menu

### 1.5.4.1 Student Profile

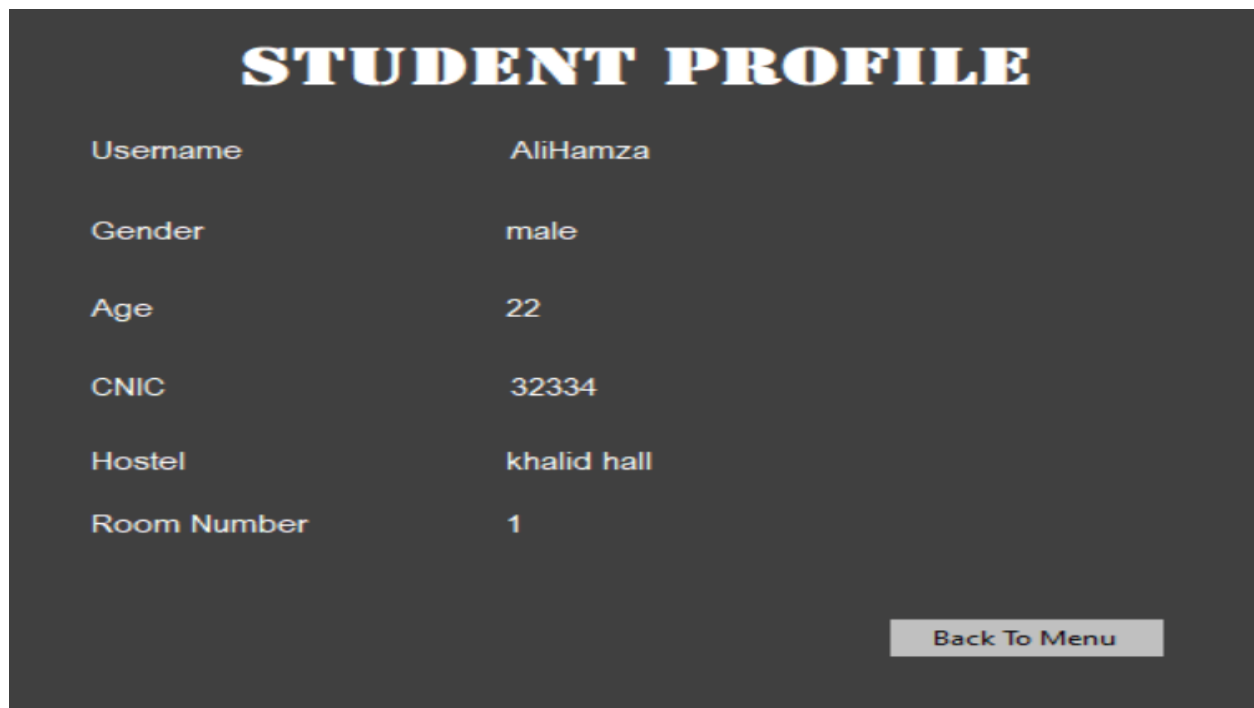


Figure 18 Student Profile

### 1.5.4.2 Complains

The screenshot shows a dark-themed interface for managing complaints. At the top, the title "COMPLAINS" is centered. Below it, there are two main sections: "Set Complain" and "DELETE COMPLAIN".

**Set Complain:** This section includes a large, rounded white input field for the complaint text. To its right, there is a "Date" label followed by a date picker showing "Friday, April 26, 2024". A grey "Add" button is positioned to the right of the date picker.

**DELETE COMPLAIN:** This section features a label "Give Complain Id" followed by a white input field with a dropdown arrow. A grey "Remove" button is located to the right of the input field.

Below these sections, the text "All Complains" is centered above a large, empty white rectangular area, likely intended for a list of complaints. In the bottom right corner, there is a grey "Back To Menu" button.

Figure 19 Complains

### 1.5.4.3 Change Password

The screenshot displays a dark-themed interface for updating a password. The title "UPDATE PASSWORD" is centered at the top in a large, stylized font.

Below the title, there are two white input fields. The first field is labeled "Current Password" and the second field is labeled "New Password".

At the bottom center, there is a grey "Update" button. In the bottom right corner, there is a grey "Back To" button.

Figure 20 Change Password

## 1.6 Queries

### 1.6.1 Complaints

#### 1.6.1.1 Add Complaint

```
insert into Complain (Detail,Date,StudentName) VALUES('{0}','{1}','{2}')
```

#### 1.6.1.2 Delete Complaint

```
DELETE FROM Complain WHERE ComplainId = {0}
```

#### 1.6.1.3 See all Complaint

```
SELECT * FROM Complain where StudentName = '{0}'
```

#### 1.6.1.4 View Complaint of a student

```
SELECT ComplainId FROM Complain where StudentName= '{0}'
```

#### 1.6.1.5 Delete Complaint of a student

```
DELETE FROM Complain WHERE StudentName = '{0}'
```

### 1.6.2 Dwellers

#### 1.6.2.1 Add Dweller

```
insert into Resident (Username, Password, Age,Gender,Cnic,Role) VALUES('{0}',  
'{1}', '{2}','{3}', '{4}', '{5}')
```

#### 1.6.2.2 Delete Dweller

```
DELETE FROM Resident WHERE UserName = '{0}'
```

#### 1.6.2.3 Find Dweller

- select \* from Resident where Username = '{0}' and Password = '{1}'

#### 1.6.2.4 Update Dweller Password

```
UPDATE Resident SET Password = '{0}' WHERE Username = '{1}'
```

#### 1.6.2.5 Update Dweller Hostel Name

```
UPDATE Resident SET HostelName = '{0}' WHERE Username = '{1}'
```

### 1.6.2.6 Find Dweller Gender

```
SELECT Gender FROM Resident where Username = '{0}'
```

## 1.6.3 Hostels

### 1.6.3.1 Add Hostel

```
INSERT INTO Hostel (HostelName, HostelType, HostelStatus) VALUES ('{0}', '{1}', '{2}')
```

### 1.6.3.2 Delete Hostel

```
DELETE FROM Hostel WHERE HostelName = '{0}'
```

### 1.6.3.3 View all Hostels

```
SELECT * FROM Hostel
```

### 1.6.3.4 View all Hostel Names

```
SELECT HostelName FROM Hostel
```

### 1.6.3.5 Update Hostel Status

```
UPDATE Hostel SET HostelStatus = 'Checked' WHERE HostelName = '{0}'
```

## 1.6.4 Rooms

### 1.6.4.1 Add Room

```
INSERT INTO Room (RoomNumber, HostelName, RoomStatus) VALUES ('{0}', '{1}', '{2}')
```

### 1.6.4.2 Delete Room

```
DELETE FROM Room WHERE HostelName = '{0}'
```

### 1.6.4.3 Update Room Status

```
UPDATE Room SET RoomStatus = 'Alloted' WHERE RoomNumber = '{0}'and HostelName = '{1}'
```

### 1.6.4.4 View Room Numbers

```
SELECT RoomNumber FROM Room WHERE HostelName = '{0}'and RoomStatus ='Vacant'
```

### 1.6.4.5 View Vacant Rooms

```
SELECT * FROM Room WHERE RoomStatus = 'Vacant'
```

## 1.6.5 Resident Tutors

### 1.6.5.1 Add Resident Tutor

```
insert into Rt (Username, Password,
Age,Gender,Cnic,Role,PhoneNo,ResidentId,HostelName) VALUES('{0}', '{1}',
'{2}','{3}', '{4}', '{5}','{6}','{7}','{8}')
```

### 1.6.5.2 Delete Resident Tutor

```
DELETE FROM Rt WHERE Username = '{0}'
```

### 1.6.5.3 View all Resident Tutors

```
SELECT * FROM Rt
```

### 1.6.5.4 Update Resident Tutor Password

```
UPDATE Rt SET Password = '{0}' WHERE Username = '{1}'
```

### 1.6.5.5 View Resident Tutor Hostel

```
SELECT HostelName FROM Rt where HostelName = '{0}'
```

## 1.6.6 Rules

### 1.6.6.1 Add Rule

```
insert into Rules (Description, Date) VALUES('{0}', '{1}')
```

### 1.6.6.2 Delete Rule

```
DELETE FROM Rules WHERE rulesId = {0}
```

### 1.6.6.3 View all Rules

```
SELECT * FROM Rules
```

## 1.6.7 Students

### 1.6.7.1 Add Student

```
insert into Student (Username, Password,
Age,Gender,Cnic,Role,ResidentId,StudentStatus) VALUES('{0}', '{1}', '{2}','{3}',
'{4}', '{5}','{6}','{7}')
```

### 1.6.7.2 Delete Student

```
DELETE FROM Student WHERE HostelName = '{0}'
```

### 1.6.7.3 View all Unalloted Students

```
SELECT Username FROM Student where StudentStatus = 'Unalloted'
```

#### 1.6.7.4 View Student's Hostel

```
SELECT HostelName FROM Student where Username = '{0}'
```

#### 1.6.7.5 View Student's Room Number

```
SELECT HostelName FROM Student where Username = '{0}'
```

#### 1.6.7.6 Update Student's Status

```
UPDATE Student SET StudentStatus = 'Alloted', RoomNumber = '{0}', HostelName = '{1}' WHERE Username = '{2}'
```

#### 1.6.7.7 View all Students

```
SELECT * FROM Student
```

#### 1.6.7.6 Update Student's Password

```
UPDATE Student SET Password = '{0}' WHERE Username = '{1}'
```

### 1.7 Future Direction

I want to Implement an intelligent room allotment system that considers various factors such as student preferences, compatibility, and proximity to common facilities. This can enhance the overall satisfaction of residents. I can Foster a sense of community by incorporating features that encourage social interactions among hostel residents. This could include discussion forums, event planning, and group messaging to facilitate communication and collaboration.