

---

# Hostel Room Allocation System

Project Report

---

Mohith Kumar Thummaluru - B180299CS

Avinash Samudrala - B180409CS

Rahul Kumawat - B180635CS

Tushar Kumar Patni - B180122CS

Ritik Gautam - B180630CS

**Final Draft**

**Date : 26/12/2020**

# Table of Contents

<b>1. Introduction</b>	<b>3</b>
<b>2. Objective</b>	<b>3</b>
<b>3. Scope of the project</b>	<b>3</b>
<b>4. Overview of project</b>	<b>3</b>
<b>5. Background and Motivation:</b>	<b>4</b>
<b>6. User Requirements Definition</b>	<b>4</b>
<b>7. System Requirements</b>	<b>4</b>
7.1 Functional System Requirement	4
7.2 Non-functional System Requirement	5
7.2.1 Performance Requirements	5
7.2.2 Safety Requirements	5
7.2.3 Security Requirements	5
<b>8. E-R Diagram</b>	<b>7</b>
<b>9. Relational Database Design</b>	<b>8</b>
<b>10. Database Normalisation</b>	<b>9</b>
10.1 Normalisation	9
10.2 Algorithm	10
10.3 Tables used in our Relational Schema	11
<b>11. Tools used</b>	<b>12</b>
<b>12. Implementation</b>	<b>12</b>
<b>13. Results</b>	<b>13</b>
<b>14. User Manual</b>	<b>14</b>
14.1 For Admin	14
14.2 For Student	18
<b>15. Testing</b>	<b>23</b>
<b>16. Conclusion</b>	<b>30</b>
<b>References:</b>	<b>30</b>

# 1. Introduction

As the name specifies “HOSTEL ROOM ALLOCATING SYSTEM” is software developed for managing various activities in the hostel. As we can notice that educational institutions are increasing year by year, thereby the number of hostels is also increasing at the same pace. Since this can be a lot of strain on the person who is running the hostel and software’s are not usually used in this context. Therefore there is a greater need for software for the accommodation of the students studying in these institutions which manages this activity of maintaining hostels. And also because a well-reputed college like NITC which is ours, is taking the choices of students for allocation of rooms from Google forms and not by a GUI(in our knowledge), which is why we wanted to target this aspect and bring this software to life.

## 2. Objective

This particular project aims at the problems of managing a hostel and avoids the problems/errors which occur when carried manually. Identification of the drawbacks of the existing system which includes the terrible procedures like filling many forms and waiting in long queues for hours to submit those forms in hostel offices which ultimately leads to a lot of paperwork, which is very inefficient. This problem leads us to the development of computerized hostel management systems that will be compatible with the current manual system. And also which is more user friendly and more GUI oriented and can sustain from any kind of inside and outside attacks of data.

## 3. Scope of the project

- Hostel Management System is designed for Hostel (like schools, Universities).
- There will be predefined criteria for the Reservation to the hostels.
- He/She checks the attested application forms of the students obtained from the internet and verifies it with the student database.
- If the students are found eligible then they are allotted to the hostel room.
- There will be a portal for complaints as well which updates daily.
- This can be further modified accordingly to the later requirements of the institute.

## 4. Overview of project

Hostel Room Allocation System is a web application that aims at computerization of the current procedure of allocating hostel rooms. Currently, the process involves students filling up the forms and submitting them in respective hostel offices which involves a lot of paperwork, hence less efficient.

## 5. Background and Motivation

Currently, most of the colleges do manual work to allocate hostels and rooms to the students every year. We all know that it's an overwhelming process to do allocation where the student's strength is pretty high as local colleges. Also, hostlers find it difficult to raise their complaints to the hostel administration faculty. Even if they do, they won't have a clear idea about the status of that complaint or issue. So, to reduce this hectic work, we introduced a web-app by which students can choose their hostels and also roomies through an online portal. It reduces the work of hostel administration faculty and also students will find this useful due to its feature of sending and tracking their complaints directly to admins. Hostel administration faculty will find this useful to post any important updates or daily feeds to students.

## 6. User Requirements Definition

The user requirement for this system is to make the system fast, flexible, less prone to error, reduce expenses, and save time.

1. Less human error
2. Strength and strain of manual labor can be reduced
3. High security
4. Data redundancy can be avoided to some extent
5. Data consistency
6. Easy to handle
7. Easy data updating
8. Easy record-keeping
9. Backup data can be easily generated.

## 7. System Requirements

### 7.1 Functional System Requirement

This section gives functional requirements that apply to the HRAS. These are sub modules in this phase.

- Administrative Module
- User Module
- Application Module

**The Functionalities are as follows:**

- Administrative Module

The Administrator can:

1. Allot different students in different hostels.
2. Vacate the students from the hostels.
3. Edit the details of the students & modify the student records.
4. Add new employees
5. Impose fine on students.
6. Can post announcements.

- User Module

The user can:

1. Can submit the application form.
2. Can view the notice board.
3. Can submit the vacating form.
4. Can choose roommates.
5. Can raise complaints.
6. Can find room numbers by Roll No.

- Application Module

This section provides a form to the students which can be filled by them, and a copy of the filled page can be taken in the printed form. This is later submitted to the Hostel authorities can be verified by them before allotting them to the respective hostels' Rooms.

## 7.2 Non-functional System Requirement

### 7.2.1 Performance Requirements

Some Performance requirements identified are listed below:

1. The database shall be able to accommodate around thousand records to store.
2. The software shall support the use of multiple users at a time.

### 7.2.2 Safety Requirements

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

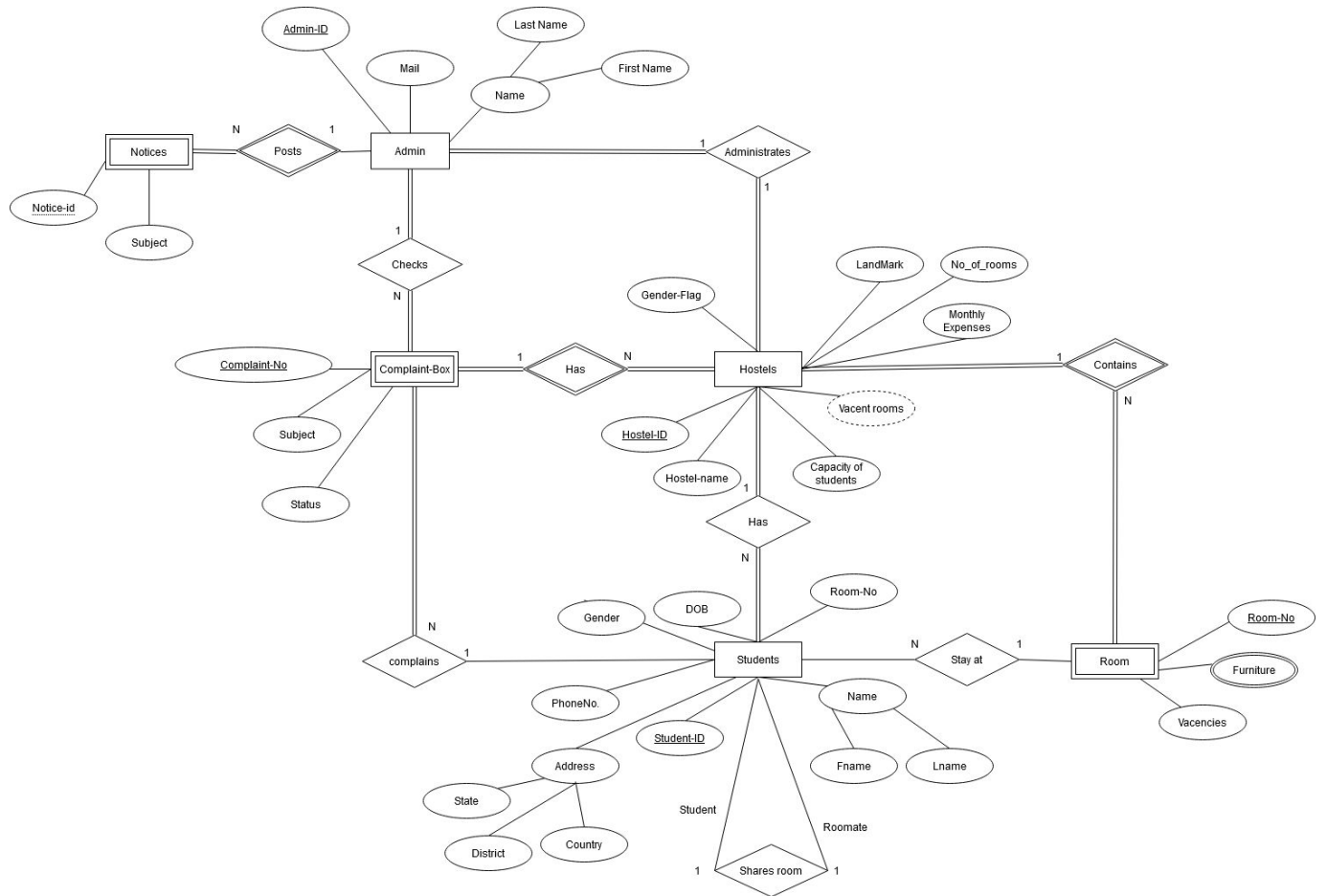
### 7.2.3 Security Requirements

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below. Keep a specific log or history data sets.

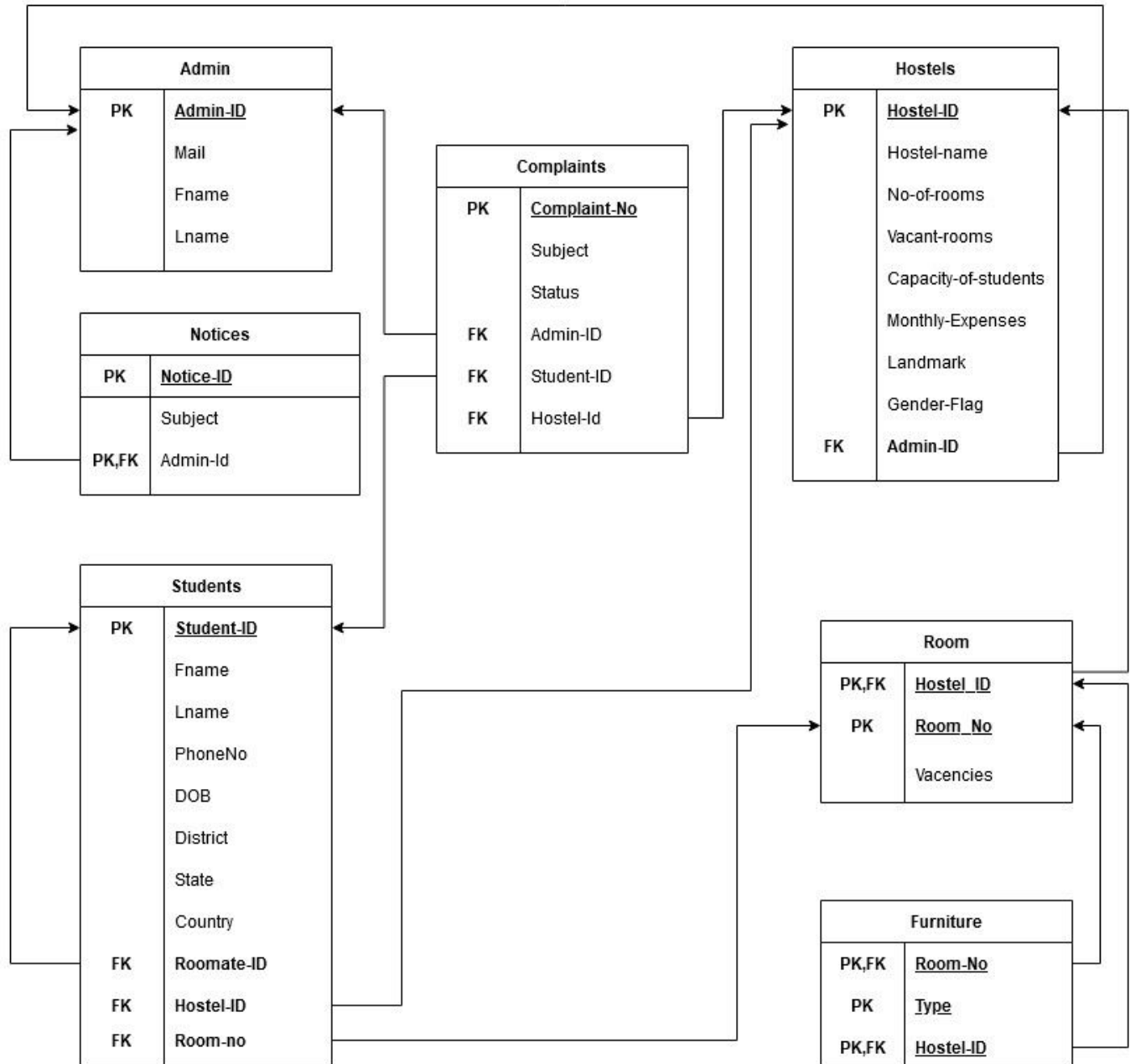
1. Assign certain functions to different modules.

2. Restrict communications between some areas of the program
3. Check data integrity for critical variables.
4. Later versions of the software will incorporate encryption techniques in the user/license authentication process.

## 8. E-R Diagram



## 9. Relational Database Design





# 10. Database Normalisation

## 10.1 Normalisation

The process of organizing the data in the database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

- (a) First Normal form (1NF)
- (b) Second Normal form (2NF)
- (c) Third Normal form (3NF)

a) First Normal Form (1NF) :

As per the rule of the first normal form, an attribute(column) of a table cannot hold multiple values i.e., it should store only atomic values.

b) Second Normal Form (2NF) :

A table is said to be in 2NF if both the following conditions hold:

- The table is in 1NF (First normal form)
- No non-prime attribute is dependent on the proper subset of any candidate key of the table.

An attribute that is not part of any candidate key is known as a non-prime attribute.

c) Third Normal Form (3NF) :

A table design is said to be in 3NF if both the following conditions hold:

- The table must be in 2NF
- The transitive functional dependency of the non-prime attribute on any super key should be removed.

In other words, 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency  $X \rightarrow Y$  at least one of the following conditions hold:

- X is a super key of the table
- Y is a prime attribute of the table

An attribute that is a part of one of the candidate keys is known as a prime attribute.

## 10.2 Algorithm

### 1) Attributes preservation:-

While decomposing universal relation R to relations  $R_1, R_2, \dots, R_m$ , all the attributes are preserved. The result of the union of attributes in all relations is the same as the universal relation attributes.

### 2) Functional dependency preservation:-

In the final result of relations in RDBMS, all the functional dependencies are preserved from universal relation functional dependencies. Given RDBMS satisfies the Dependence Preservation Property.

$$(\pi_{R_1}(F) \cup \pi_{R_2}(F) \cup \pi_{R_3}(F) \cup \dots \cup \pi_{R_n}(F))^+ = F^+$$

### 3) Lossless JOIN:-

While decomposing relations from global relations it is checked that while applying JOIN operation between those relations there is not any spurious tuple. Given RDBMS satisfies Lossless JOIN Property.

$$^*(\pi_{R_1}(r), \pi_{R_2}(r), \pi_{R_3}(r), \dots, \pi_{R_n}(r)) = r$$

\* denotes natural join of all the relation

## 10.3 Tables used in our Relational Schema

Admin	
PK	<u>Admin-ID</u>
	Mail
	Fname
	Lname
FK	Hostel-ID

(i)

Hostels	
PK	<u>Hostel-ID</u>
	Hostel-name
	No-of-rooms
	Vacant-rooms
	Capacity-of-students
	Monthly-Expenses
	Landmark
	Gender-Flag
FK	Admin-ID

(ii)

Students	
PK	<u>Student-ID</u>
	Fname
	Lname
	PhoneNo
	DOB
	District
	State
	Country
FK	Roomate-ID
FK	Hostel-ID
FK	Room-no

(iii)

Complaints	
PK	<u>Complaint-No</u>
	Subject
	Status
FK	Admin-ID
FK	Student-ID
FK	Hostel-Id

(iv)

Room	
PK,FK	<u>Hostel_ID</u>
PK	<u>Room_No</u>
	Vacencies

(v)

Furniture	
PK,FK	<u>Room-No</u>
PK	<u>Type</u>
PK,FK	<u>Hostel-ID</u>

(vi)

Notices	
PK	<u>Notice-ID</u>
	Subject
PK,FK	Admin-Id

(vii)

## 11. Tools used

1. Front- end : Html5, CSS3, Bootstrap
2. Back-end : PHP, javascript
3. Database : SQL

## 12. Implementation

1. Admin Module:
  - a. Admin\_login.php : this page avails the admins to login.
  - b. Add\_admin.php : this page enables the main admin to add new admins to maintain hostels (this is accessible only to the main admin i.e admin1).
  - c. Update\_hostels.php : the main admin(admin1) can make new entries of hostels using this page.
  - d. Admin\_land.php : this displays the dashboard of all admin details and complaints of all students under the logged-in admin.
  - e. Admin\_list.php : this page displays the details of all students in that hostel maintained by the logged-in admin.
  - f. Admin\_feed.php : using this page admin can post notices to students.
2. Student Module:
  - a. Index.php : this is the login page for students. New users (students) can sign up by providing the required credentials.
  - b. Signup.php : students can sign up using this page
  - c. student\_main.php : students get to choose a hostel and roommate using this page.
    - i. Through this page, a student can send requests for the students who are supposed to stay in that particular hostel of the logged-in student.
    - ii. Here one can check their requests and correspondingly accept or reject them and there might be cases where students might misuse this feature and start spamming the database so duplicate requests are not entertained (as implementation is concerned).
    - iii. Students can post their complaints, which will be sent to the corresponding admin.

The feedback of these operations is displayed as a snack bar component.
  - d. Profile.php : this page displays the details of logged-in students where one can also edit their information.
  - e. Notifications.php : using this page students can view the notifications posted by the admin and also the status of the complaint posted by the logged-in student.

## 13. Results

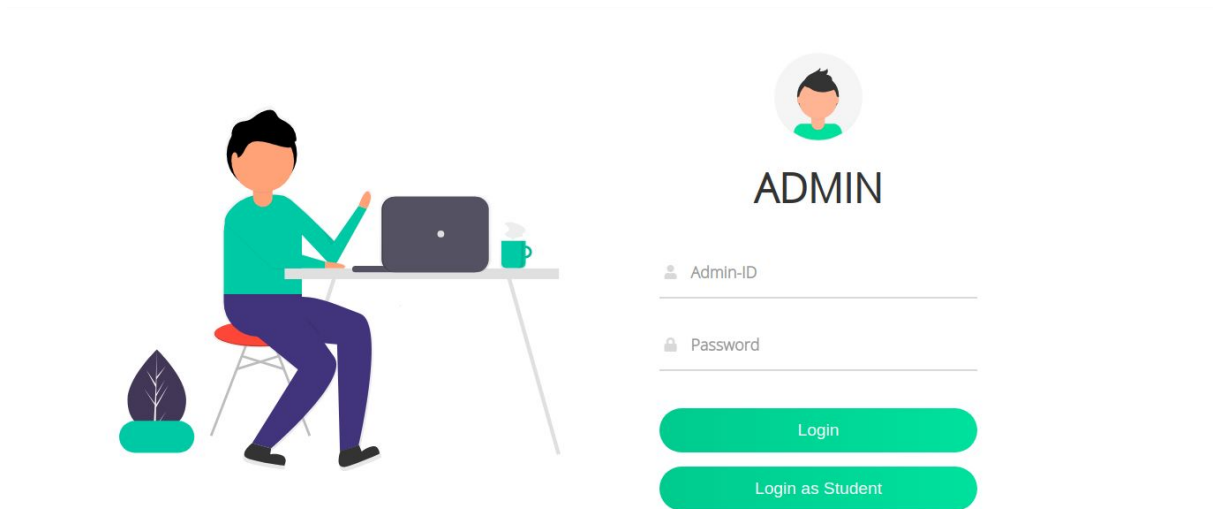
Through this software, a very laborious task of allocating the hostel rooms has been made automated and also provided the users to choose the hostel and their roommates according to their choice. This platform created a direct bridge between hostel officers and students, thereby reducing the gap between them and making interactions simple. This also avails the hostel officers to reach the students in no time just by posting a news feed. This also clears the dilemma of students whether the complaint made by them is being processed or not by displaying the status of their complaints, thereby increasing the transparency in running the system.

# 14. User Manual

## 14.1 For Admin

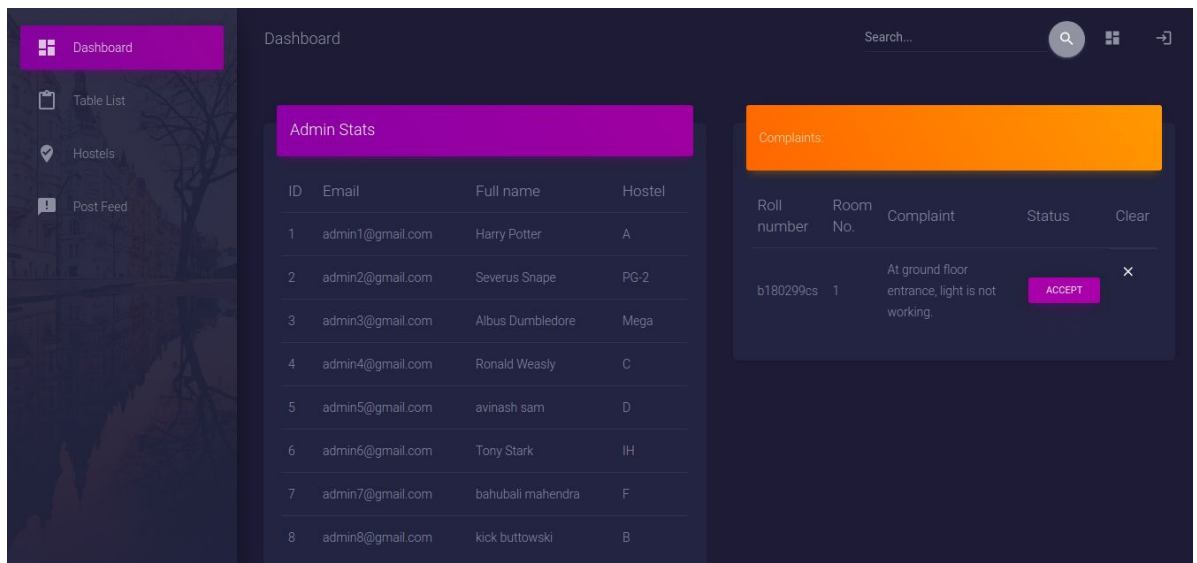
### 1. Sign-in :

Every admin is provided with a username and the password by the main admin i.e admin1. So all the admin users can only sign-in to the webpage using their credentials.



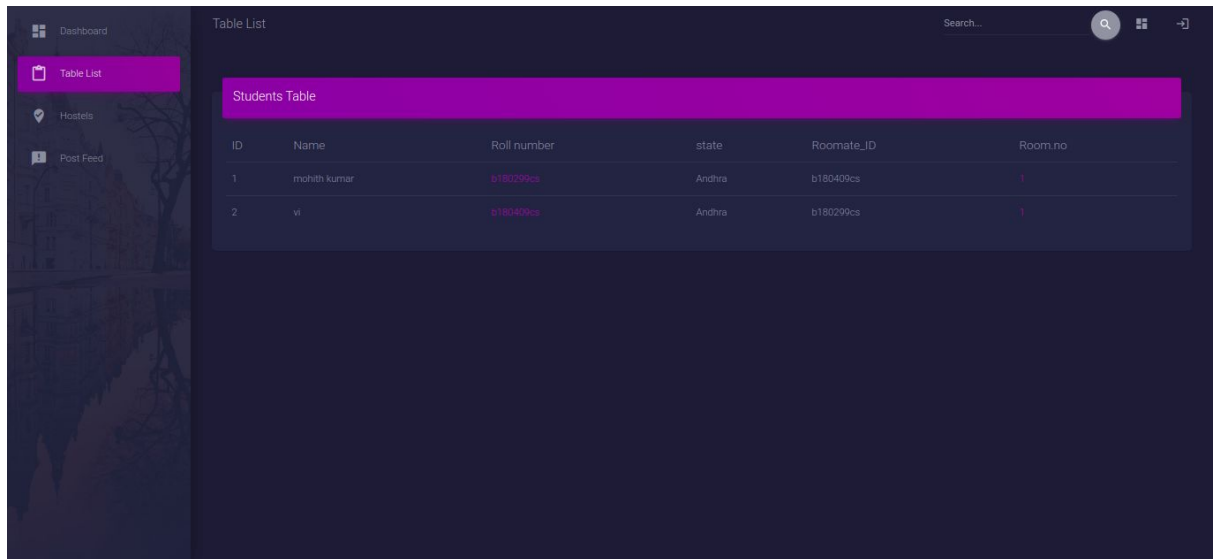
### 2. Admin Dashboard:

After logging in the landing page of the admin users displays the dashboard containing details of the peer admins and the complaints in their corresponding hostels.



### 3. Student Details:

In Table-list (present in the left nav bar of the landing page), all the details of the students maintained by the logged-in admin are displayed.

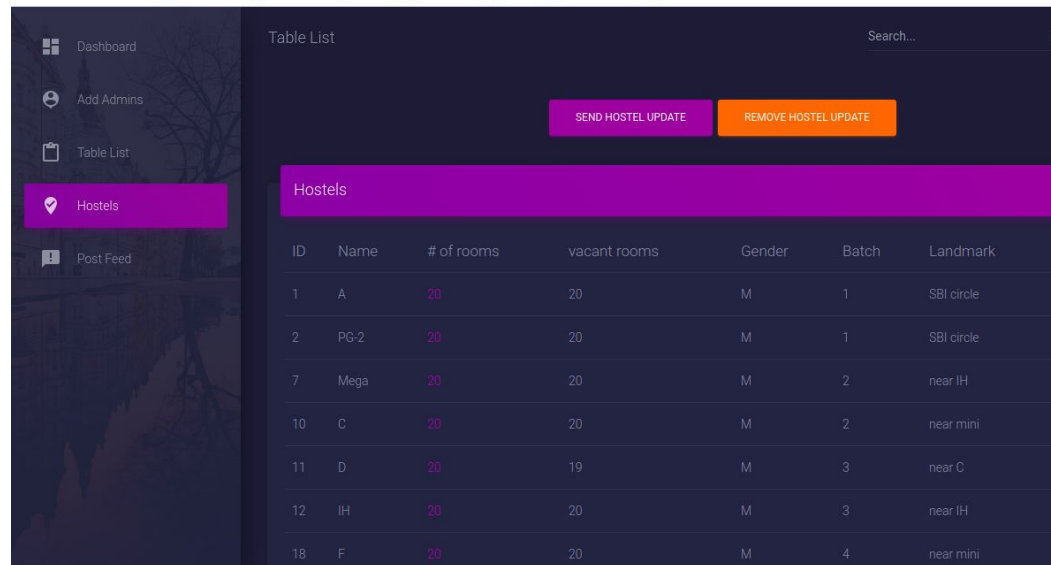


The screenshot displays a web application interface. On the left is a dark sidebar with a navigation menu containing 'Dashboard', 'Table List' (highlighted in red), 'Hostels', and 'Post Feed'. The main content area is titled 'Table List' and features a search bar. Below this is a red header for the 'Students Table'. The table itself has six columns: ID, Name, Roll number, state, Roomate\_ID, and Room.no. It contains two rows of data.

ID	Name	Roll number	state	Roomate_ID	Room.no
1	mohith kumar	b180299cs	Andhra	b180409cs	1
2	vi	b180409cs	Andhra	b180299cs	1

#### 4. Main admin privileges:

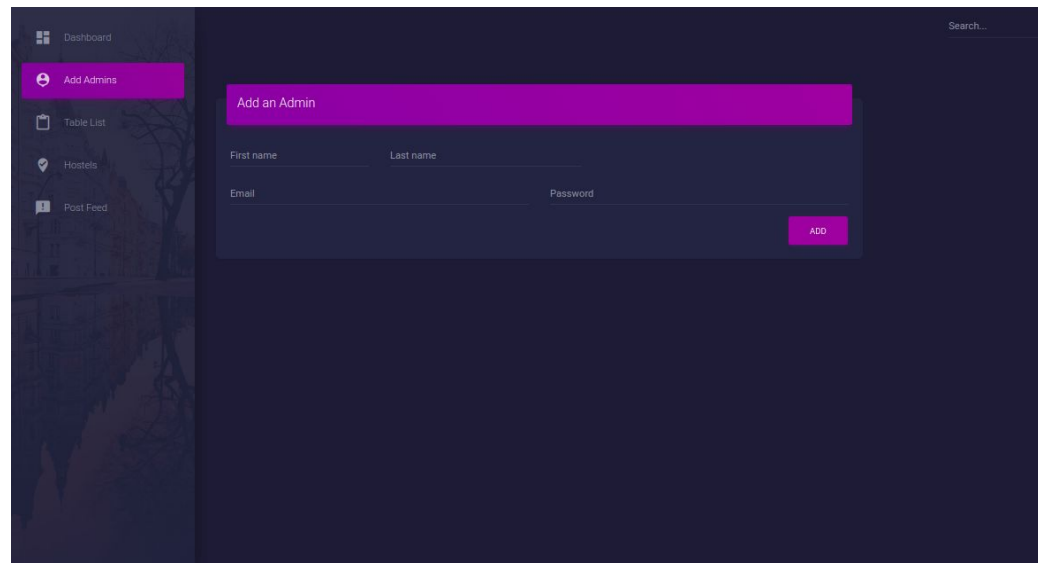
- At the beginning of every academic year, the main admin (admin1) sends the hostel update, so that the rooms can be reallocated to the other students.
- And after a certain deadline, the main admin can remove the update which was given earlier.



The screenshot shows the 'Hostels' table in the admin dashboard. The left sidebar contains navigation links: Dashboard, Add Admins, Table List, Hostels (selected), and Post Feed. The main content area is titled 'Table List' and includes a search bar and two buttons: 'SEND HOSTEL UPDATE' and 'REMOVE HOSTEL UPDATE'. The table itself has the following data:

ID	Name	# of rooms	vacant rooms	Gender	Batch	Landmark
1	A	20	20	M	1	SBI circle
2	PG-2	20	20	M	1	SBI circle
7	Mega	20	20	M	2	near IH
10	C	20	20	M	2	near mini
11	D	20	19	M	3	near C
12	IH	20	20	M	3	near IH
18	F	20	20	M	4	near mini

- The main admin has the privilege to appoint new admins using the **Add Admin** tab present in the left navbar.



The screenshot shows the 'Add an Admin' form in the admin dashboard. The left sidebar contains navigation links: Dashboard, Add Admins (selected), Table List, Hostels, and Post Feed. The main content area is titled 'Add an Admin' and includes a search bar and an 'ADD' button. The form fields are:

First name \_\_\_\_\_ Last name \_\_\_\_\_

Email \_\_\_\_\_ Password \_\_\_\_\_

**ADD**



- d. The newly appointed admins can be allocated to one of the hostels.

The screenshot displays the 'Hostels' management interface. On the left is a sidebar with navigation links: Dashboard, Add Admins, Table List, Hostels (active), and Post Feed. The main content area features a table of existing hostels and a form to add a new one.

ID	Hostel Name	# of rooms	Gender	Admins	Landmark
2	PG-2	20	M	1	SBI circle
7	Mega	20	M	2	near IH
10	C	20	M	2	near mini
11	D	17	M	3	near C
12	IH	20	M	3	near IH
18	F	19	M	4	near mini

### Add Hostels

Hostel Name  # of rooms

Batch  ☒ Male ☐ Female

Landmark

Admin-ID

**ADD HOSTEL**

## 5. Post Feed:

Admins can post their feed or notices to the students of the corresponding hostel using this page. All their previous feeds they have posted are also visible on this page and admins can delete their previously posted feeds on this page.

The screenshot shows the 'Post Feed' interface. The sidebar on the left includes: Dashboard, Table List, Hostels, and Post Feed (active). The main content area has a search bar and a form to post a new feed.

### Feed

Hello Everyone,  
Wish you a very happy new year!!!

**POST**

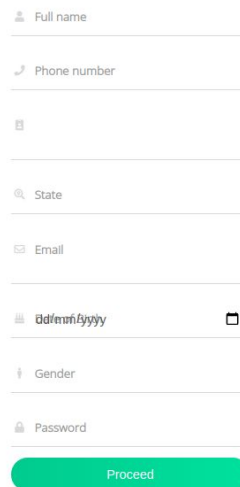
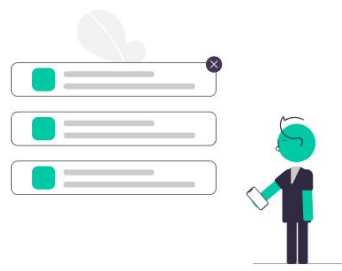
### Recent Feeds

#	Feed	Clear

## 14.2 For Student

### 1. Sign-up page:

Students can register to this web-app using this signup page by mentioning all their details.



Full name

Phone number

State

Email

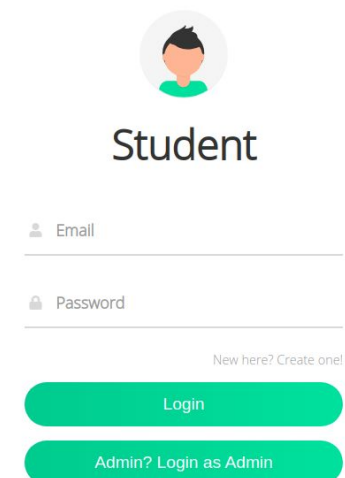

Gender

Password

Proceed

### 2. Sign-in page:

Existing users(students) can log in to the web app using the correct credentials through the sign-in page as shown below.



Student

Email

Password

New here? Create one!

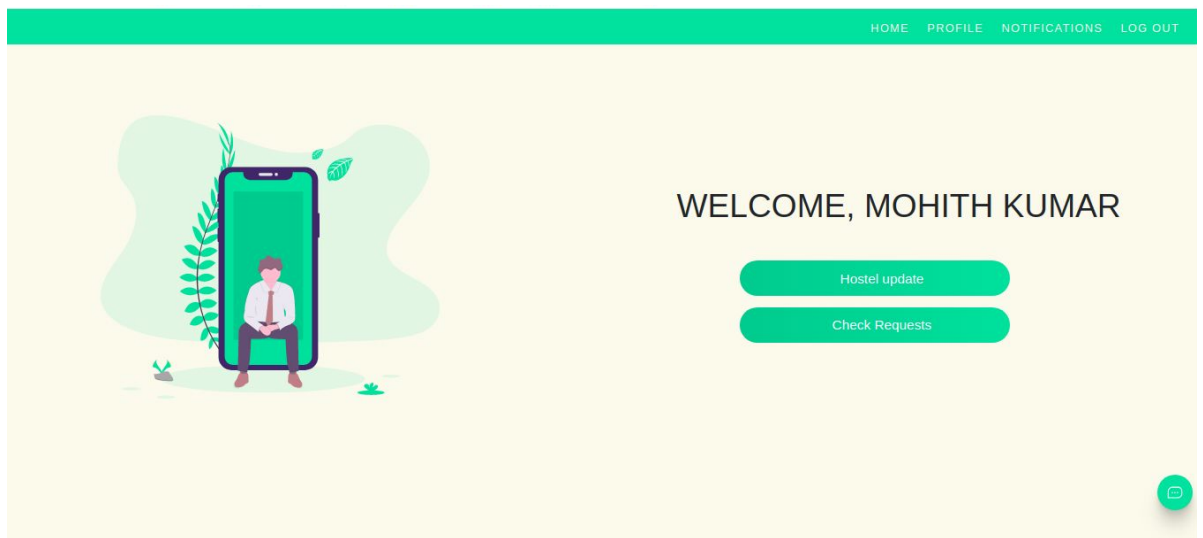
Login

Admin? Login as Admin

### 3. Hostel Update:

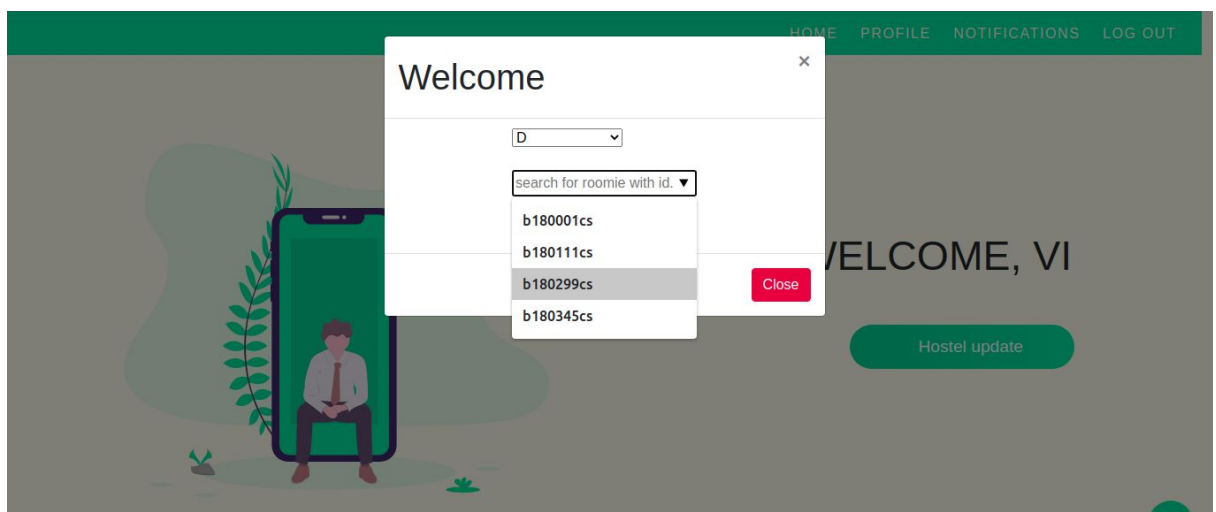
After signing in users will be redirected to the landing page which will appear as shown below.

- a. When the main admin sends the notification for the hostel update then the Hostel update button in the landing page of the student becomes active and when there are room requests for the logged-in student the check request button becomes active.



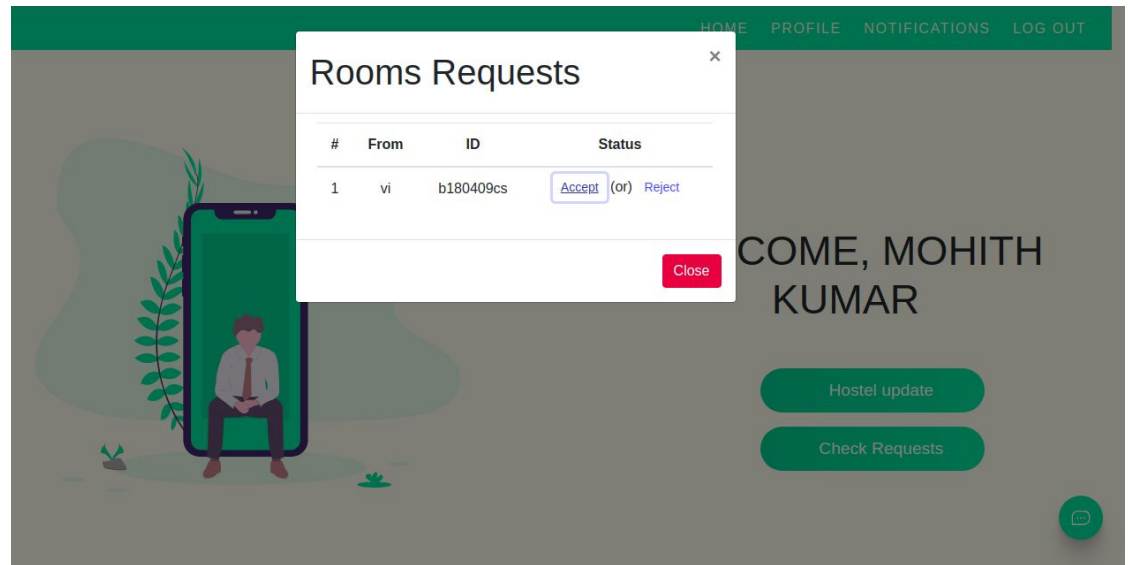
#### b. Choose Roommate (send request):

After clicking the Hostel update button, a pop-up window will appear through which each student can get to choose a hostel from one of the hostels allocated to their batch (year) and this pop-up window allows them to send a request to their batchmates only if the other batchmate is not paired and this page does not allow the users to send duplicate requests to the receiver (to avoid spamming).



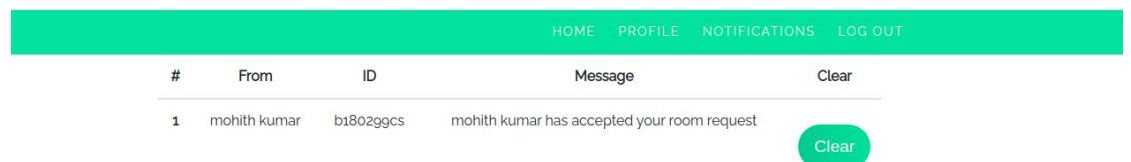
### c. Accept Request:

Users(students) can accept or reject the room requests which they have received from other users. If they accept the request then the user will be allocated to the same room and hostel as his request's sender.



### d. Notification for request acceptance:

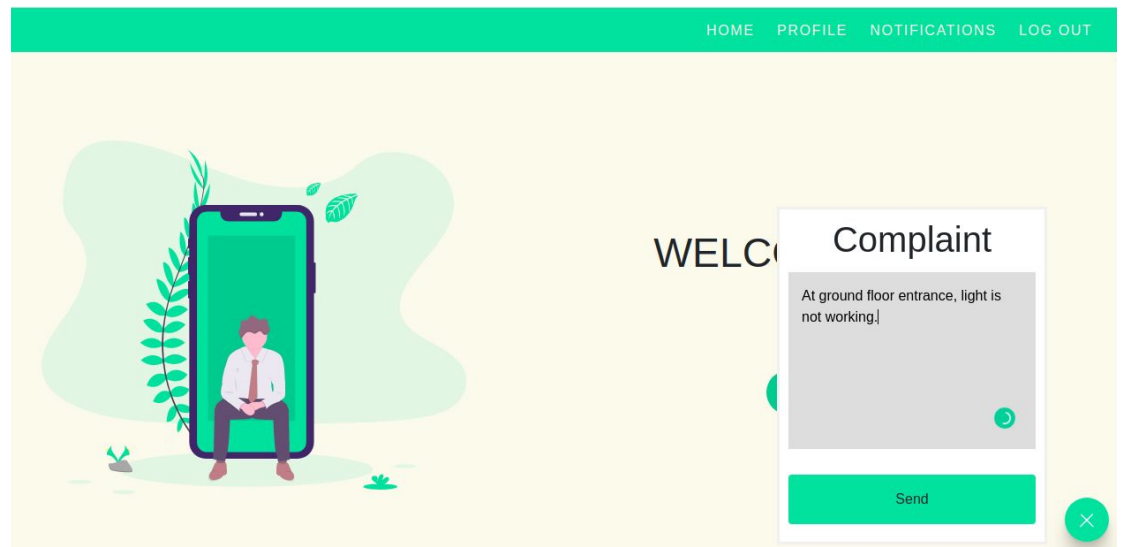
Notifications regarding the acceptance or rejection of your room requests are visible on the notifications page.



#### 4. Complaints:

##### a. Send Complaints:

Users can send complaints directly to the admin using the floating button present in the bottom right corner of the landing page of the users. All these complaints are sent directly to the admin of that particular hostel.



##### b. Notification for the feedback of the complaint:

All the notifications regarding the status(accepted, rejected, cleared) of your complaint sent by the admin are visible here. Users can clear unwanted notifications.

HOME PROFILE NOTIFICATIONS LOG OUT				
#	From	ID	Message	Clear
1	avinash sam (Admin)	5	Your complaint :At ground floor entrance, light is not working. has been Accepted..	Clear

(i) accepted indicates that the complaint is under process

HOME PROFILE NOTIFICATIONS LOG OUT				
#	From	ID	Message	Clear
1	avinash sam (Admin)	5	Your complaint :At ground floor entrance, light is not working. has been Accepted..	Clear
2	avinash sam (Admin)	5	Your complaint :At ground floor entrance, light is not working. has been Cleared..	Clear

(ii) cleared indicates that the complaint is solved

## 5. Check feed posted by admin:

This functions as a virtual notice board where the news feed posted by the admin is notified to the students as a notification that can only be cleared by the admin.

[HOME](#)
[PROFILE](#)
[NOTIFICATIONS](#)
[LOG OUT](#)

#	From	ID	Message	Clear
1	avinash sam (Admin)	5	Your complaint :At ground floor entrance, light is not working. has been Accepted..	Clear
2	avinash sam (Admin)	5	Your complaint :At ground floor entrance, light is not working. has been Cleared..	Clear
3	avinash sam (Admin)	5	Hello Everyone, Wish you a very happy new year!!!	Clear

# 15. Testing

## 1. Send hostel Update:

When the main admin enters the hostel update button then the notification enters into the NOTICES table which can be seen in the below figures.

ID	Name	# of rooms	vacant rooms	Gender	Batch	Landmark	Admin
1	A	20	20	M	1	SBI circle	1
2	PG-2	20	20	M	1	SBI circle	2
7	Mega	20	20	M	2	near IH	3
10	C	20	20	M	2	near mini	4
11	D	20	20	M	3	near C	5
12	IH	20	20	M	3	near IH	6
18	F	20	20	M	4	near mini	7
20	B	20	20	M	4	Next to Mega	8
21	B	20	20	M	4	Next to Mega	8

Notice_ID	Subject	Admin_ID	student_id
1	New Hostel Update	1	0

## 2. Sign-up:

When the students hit the proceed button by entering the required details then the entry made by the student will be added to the STUDENTS table in the DB, which can be seen in the following figures.

Full name	Phone number	b180299cs	State	Email	Date of Birth	Gender	Password
Mohith	9490915217	b180299cs	andhra	mohith@gmail.com	23/10/2000	M	****

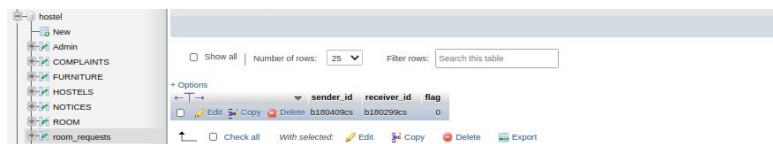
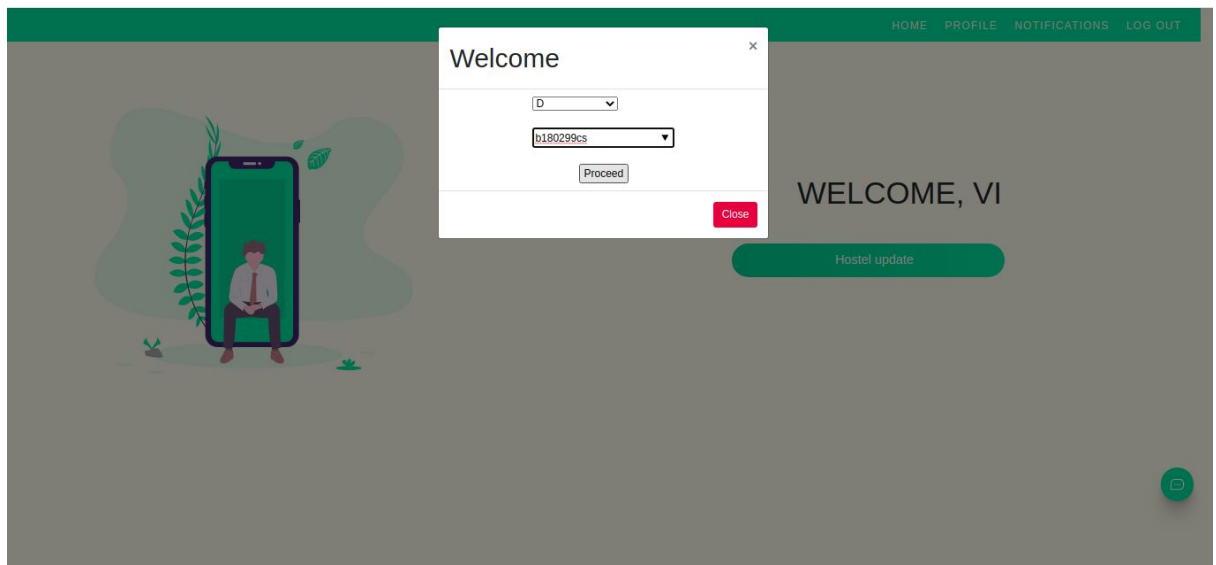
  

Full name	Phone number	b180299cs	State	Email	Date of Birth	Gender	Password
Mohith	9490915217	b180299cs	andhra	mohith@gmail.com	23/10/2000	M	****

### 3. Room requests:

#### a. Sending requests

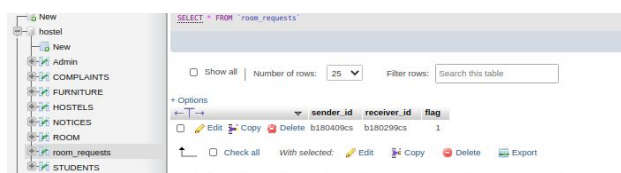
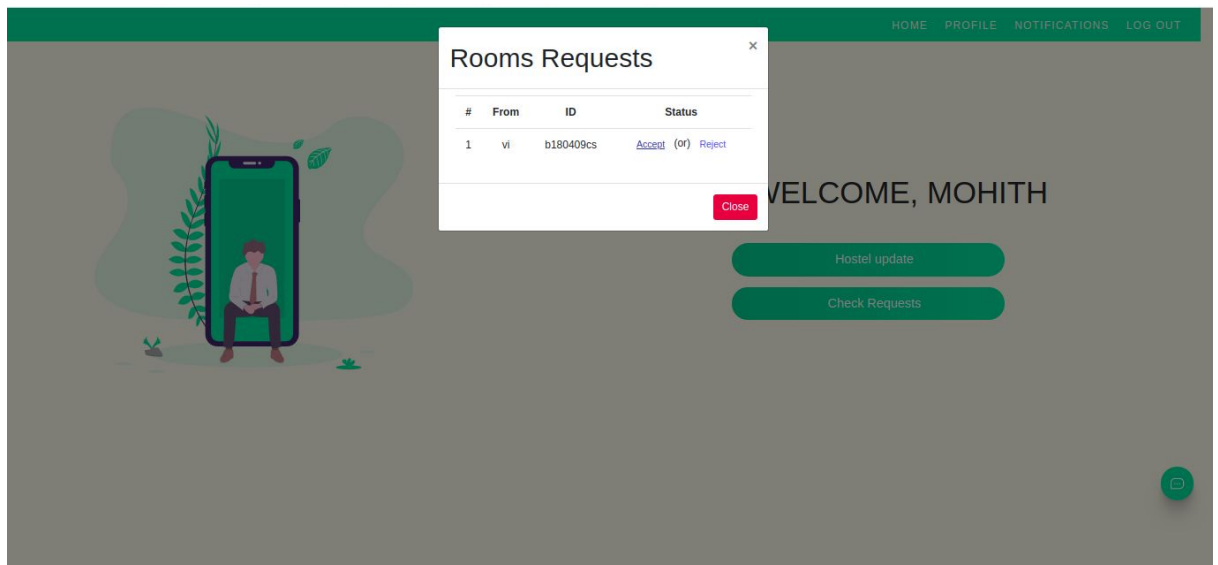
When a student chooses a hostel and sends the request to the receiver (the other student) then an entry will be made in the ROOM\_REQUESTS table in the DB with *flag* = 0, which can be seen in the following figures.



#### b. Accepting Requests:

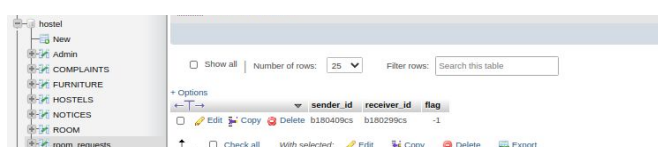
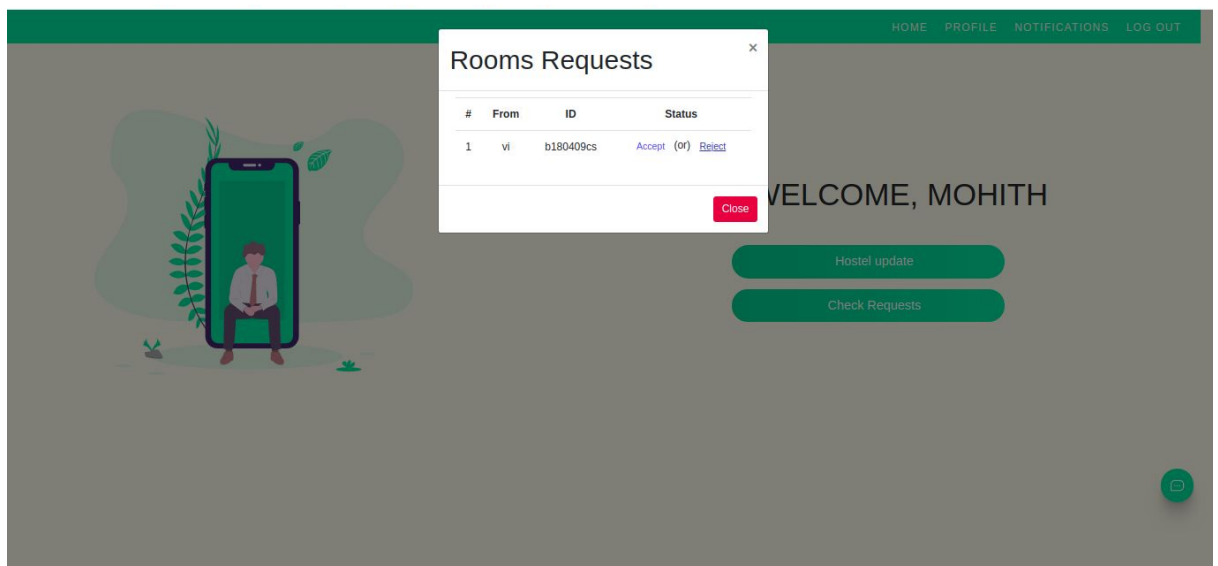
When the receiver accepts the request sent by the sender then the flag of the entry made by the sender will be updated to 1 i.e *flag* = 1, which can be seen from the below figures.





### c. Rejecting request

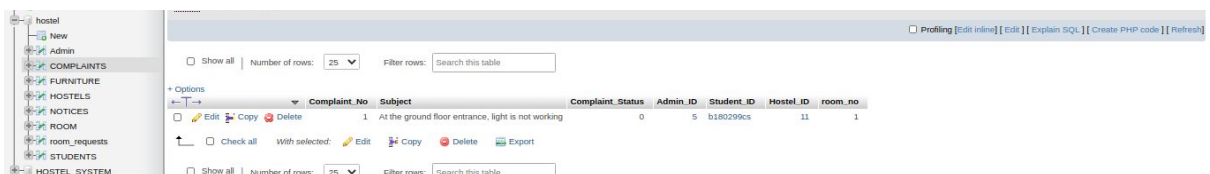
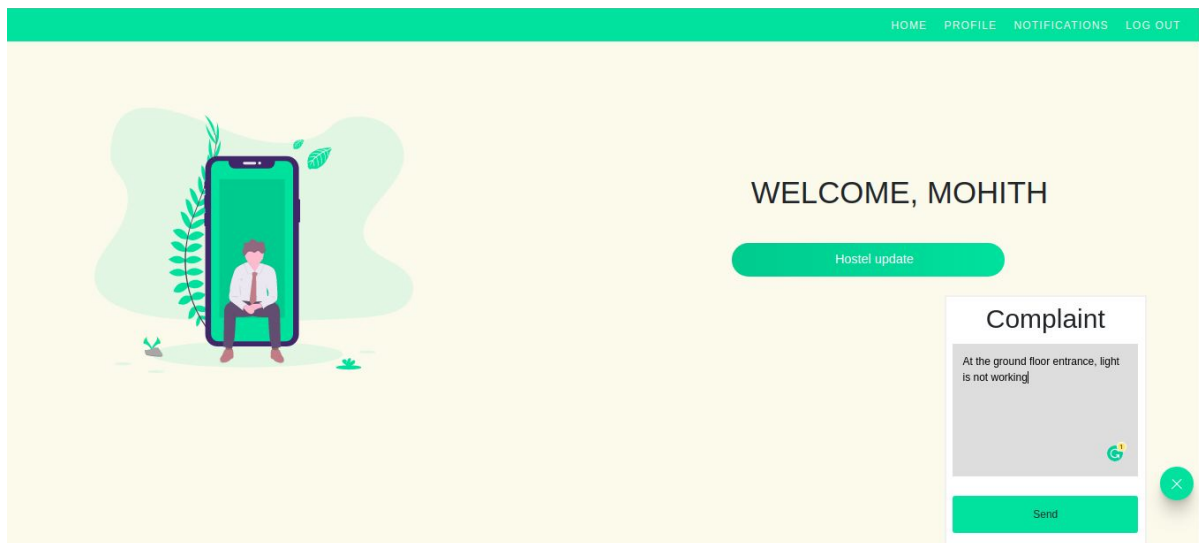
When the receiver accepts the request sent by the sender then the flag of the entry made by the sender will be updated to 1 i.e  $flag = -1$ , which can be seen from the below figures.



## 4. Complaints:

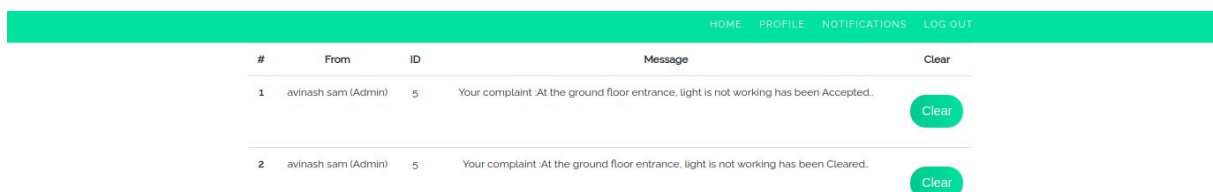
### a. Sending Complaint

When a student enters the complaint and sends it, then an entry will be made in the COMPLAINTS table in the DB with *complaint\_status* = 0, which can be seen from the following figures.



### b. Accepting Complaint

When the corresponding admin accepts the complaint a notification saying '*your complaint has been accepted*' will be sent to the student who complained and the *complaint\_status* will be changed to 1 i.e *complaint\_status* = 1, in the COMPLAINTS entry made by the complaint sender, which can be seen in the following figures.



Complaint_No	Subject	Complaint_Status	Admin_ID	Student_ID	Hostel_ID	room_no
1	At the ground floor entrance, light is not working	1	5	b180299cs	11	1

## 5. Sending Feed:

When an admin posts the feed then a new entry will be made in the NOTICES table in the DB, which can be seen from the following figures.

Feed

Hello Everyone,  
Wish you a very happy new year

POST

Recent Feeds

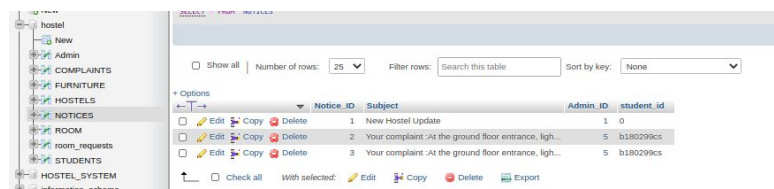
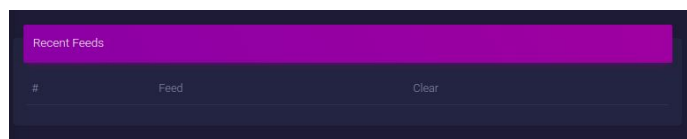
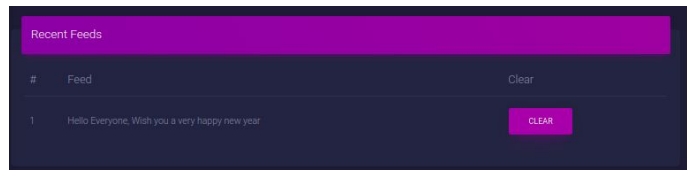
#	Feed	Clear
1	New Hostel Update	1 0
2	Your complaint :At the ground floor entrance, ligh...	5 b180299cs
3	Your complaint :At the ground floor entrance, ligh...	5 b180299cs
4	Hello Everyone, Wish you a very happy new year	5 1

Notice_ID	Subject	Admin_ID	student_id
1	New Hostel Update	1	0
2	Your complaint :At the ground floor entrance, ligh...	5	b180299cs
3	Your complaint :At the ground floor entrance, ligh...	5	b180299cs
4	Hello Everyone, Wish you a very happy new year	5	1

## 6. Clearing:

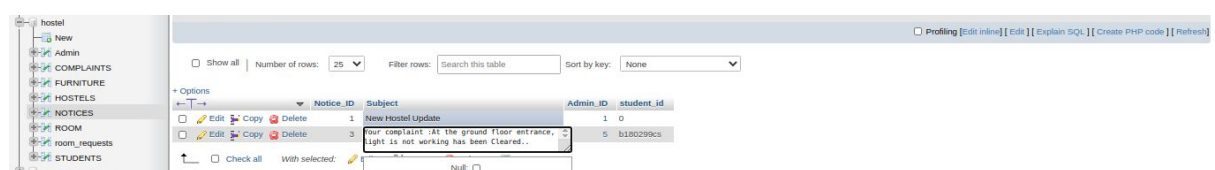
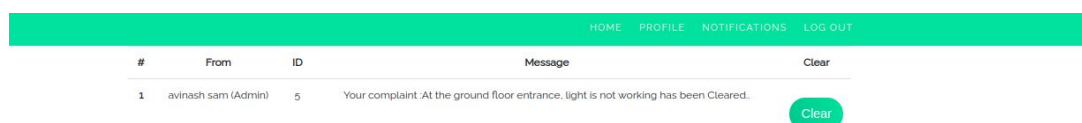
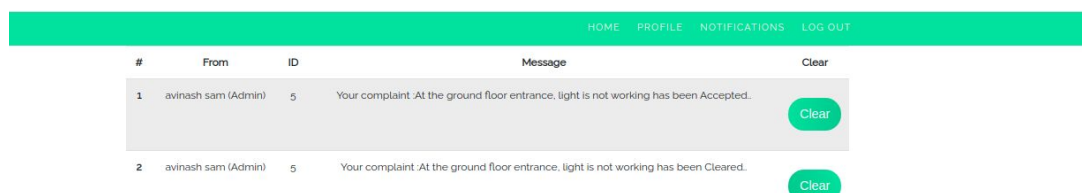
### a. Clearing feed:

When the admin who posted the feed hits the clear button then the entry in the NOTICES table will be cleared, which can be seen from the following figures.



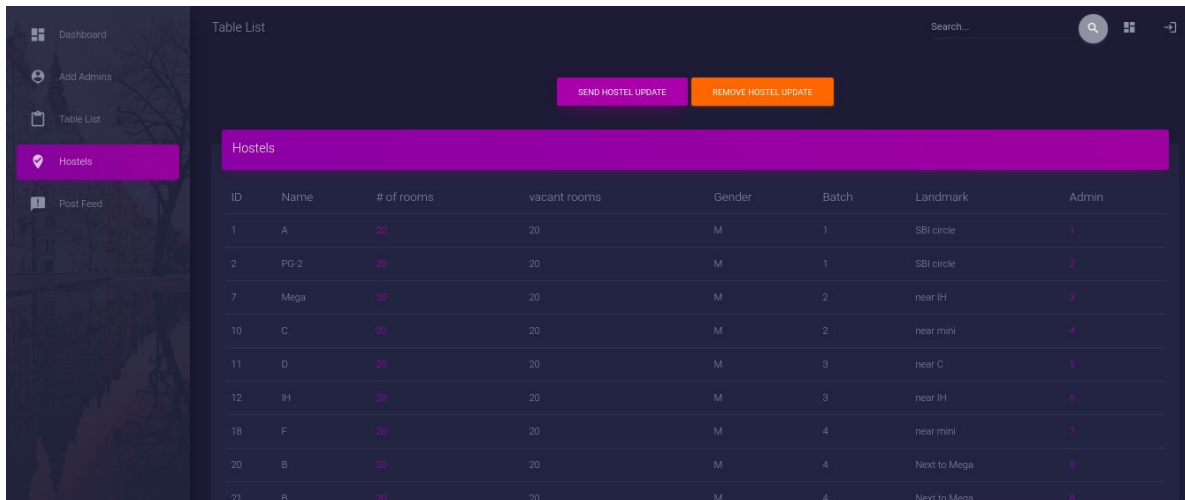
### b. Clearing Notifications:

When the student hits the clear button in the notification tab the notifications (other than the feed posted by the admin) will be cleared in the notifications tab and in the NOTICES table as well, which can be seen from the following figures.

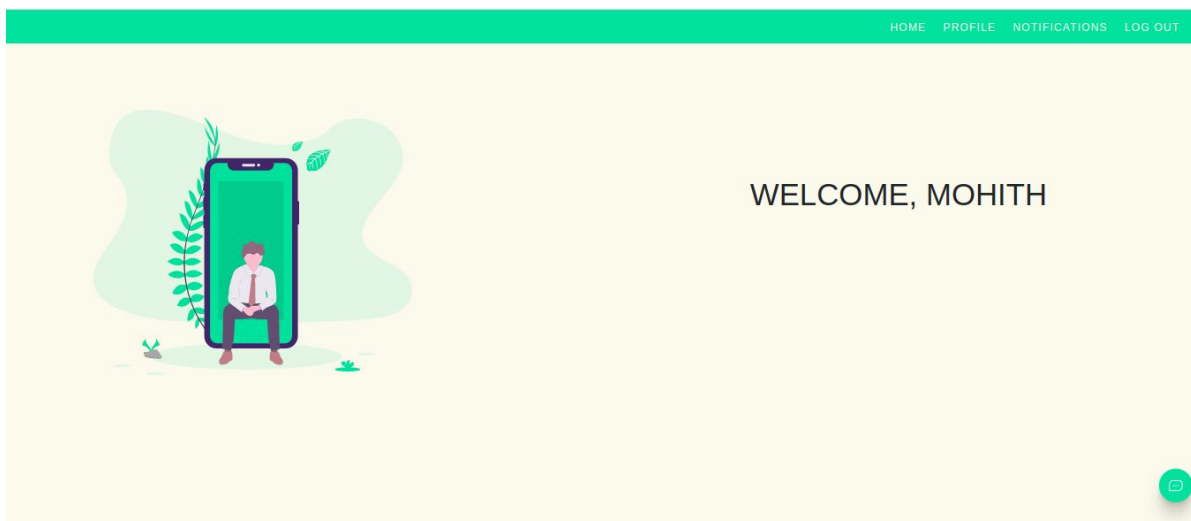


## 7. Removing Hostel Update:

When the admin hits the remove hostel update button, then *hostel update* and *check requests* will become inactive, which can be seen from the following figures.



ID	Name	# of rooms	vacant rooms	Gender	Batch	Landmark	Admin
1	A	20	20	M	1	SBI circle	1
2	PG-2	20	20	M	1	SBI circle	2
7	Mega	20	20	M	2	near IH	3
10	C	20	20	M	2	near mini	4
11	D	20	20	M	3	near C	5
12	IH	20	20	M	3	near IH	6
18	F	20	20	M	4	near mini	7
20	B	20	20	M	4	Next to Mega	8
21	B	20	20	M	4	Next to Mega	8



## 16. Conclusion

All the overwhelming work of hostel-room allocation have been reduced since all the room allocations where done automatically by this web-app. Thereby reduceing the strain of manual work. This mobile responsive webapp allows users to directly interact with the admins and send their complaints to get them solved. This also facilitates the admins to interact the students very efficiently by sending the notifications to students under this particular admin.

## References:

1. <https://www.w3schools.com/php/default.asp>
2. [https://www.w3schools.com/php/php\\_mysql\\_intro.asp](https://www.w3schools.com/php/php_mysql_intro.asp)
3. <https://www.javatpoint.com/php-tutorial>