

---

# Project Documentation

for

# Hostel Management System

Version 1.0

Prepared by

Group Number: 4

Emmanuel Marian Mathew  
Mathew Jose Mammoottil  
Shaaheen A M  
Sidharth Menon  
Vishnu Sajith

B180347CS  
B180586CS  
B181134CS  
B180561CS  
B180474CS

[emmanuel\\_b180347cs@nitc.ac.in](mailto:emmanuel_b180347cs@nitc.ac.in)  
[mathew\\_b180586cs@nitc.ac.in](mailto:mathew_b180586cs@nitc.ac.in)  
[shaaheen\\_b181134cs@nitc.ac.in](mailto:shaaheen_b181134cs@nitc.ac.in)  
[sidharth\\_b180561cs@nitc.ac.in](mailto:sidharth_b180561cs@nitc.ac.in)  
[vishnu\\_b180474cs@nitc.ac.in](mailto:vishnu_b180474cs@nitc.ac.in)

Instructor:	Abdul Nazeer
Course:	Database Management Systems
Date:	29 - 12 - 2020

## CONTENTS

<b>1. LOGIN AND LOGOUT FUNCTIONALITIES</b>	<b>3</b>
1.1. LOGIN OF STUDENT / WARDEN / ADMINISTRATOR INTO THE DATABASE	3
1.2. LOGOUT FUNCTION	3
<b>2. STUDENT FUNCTIONALITIES</b>	<b>4</b>
2.1. ADD AND VIEW REQUESTS / COMPLAINS	4
2.2. CHANGE AND VACATE HOSTEL ROOMS	5
2.3. ISSUING OF THE MESS CARD	5
2.4. ADD AND VIEW LOST / FOUND REQUESTS	6
2.5. ADD AND VIEW VISITOR DETAILS	7
2.6. EDIT ADDRESS, MOBILE NUMBER, PASSWORD	8
2.7. VIEW LOST / FOUND, REQUEST / COMPLAINS, DUES TABLES	8
<b>3. ADMINISTRATOR / WARDEN FUNCTIONALITIES</b>	<b>9</b>
3.1. ALLOCATING A NEW STUDENT	9
3.2. ALLOCATING A FELLOW ADMINISTRATOR / WARDEN	10
3.3. ALLOCATING A MESS FOR A STUDENT	10
3.4. ALLOCATING DUES FOR THE STUDENT	11
3.5. VACATING ROOMS USED BY A STUDENT	11
3.6. MANAGING LOST / FOUND ENTRIES OF THE STUDENT	12
3.7. MANAGING REQUEST / COMPLAINTS OF THE STUDENT	12
3.8. VIEW ALL THE STUDENTS MANAGED BY THE HOSTEL MANAGEMENT SYSTEM	13
3.9. VIEW ALL THE VISITORS VISITING A PARTICULAR STUDENT	13
3.10. VIEW HOSTEL AND MESS STATISTICS BEFORE ALLOCATION	13
<b>4. OUR TEAM, HELP, HOSTEL STATISTICS PAGES</b>	<b>14</b>
4.1. OUR TEAM PAGE	14
4.2. HELP PAGE	14
4.3. HOSTEL STATISTICS PAGE	14

## Login and Logout Functionalities:

### ❖ Login of the Student/Admin/Warden into the Website:

- **Input:** Enter Email, Password and choose privilege: Admin or Student
- **Output:** Verifies the above data with the data in the database and redirects to respective user pages ( Administrator / Warden ).
- Get the user email, password and user privilege using a form
- If any field is left blank, the user is notified and asked to fill all the forms before submitting
- POST method is used to send data to the backend server
- In the backend page, the data is extracted using \$\_POST
- The extracted data is verified with the data present in users table in the database
- If the data matches, the user is redirected to their respective user profile pages
- Otherwise user is notified with an error message
- Also, the user email is stored in a session variable for future use

The image displays two screenshots of the NITC Hostel Management System login page. The top screenshot shows the login form with the email field containing 'admin@nitc.ac.in', the password field containing 'admin123', and the privilege dropdown set to 'Admin'. The bottom screenshot shows the same form with the email field containing 'mathew@nitc.ac.in', the password field masked with '\*\*\*\*\*', and the privilege dropdown set to 'Student'. Both screenshots feature a dark background with a starry pattern and large white text on the sides reading 'ABC DEF F'G PG1 PG2 MBH LH MLH'. A navigation bar at the top includes links for HOSTELS, DSS, CONTACT, HELP, and LOGIN. A footer message reads 'Unable to login? GET IN TOUCH WITH THE NITC HOSTEL OFFICE NOW!'.

### ❖ Logout:

- The value stored in the session variable is deleted
- User is redirected to the login page

## The Student Has The Following Functionalities:

- Add and View Requests/Complaints
- Change/Vacate Hostel Rooms
- Issue of the Mess Card
- Add and View Lost/Found Details
- Add and View Visitor Details
- Edit Address, Password, Mobile Number
- View Lost and Found, Requests and Complaints and Dues Table

### ◆ Add and View Requests/Complaints:

- **Input:** Select Request/Complaint, Add a detailed description
- **Output:** Addition of the requests to the req\_com database.
- Requests and complaints form
  - Get the type of concern (Request/Complaint) and description from the user using a form.
  - POST method is used to send data to the server to update the relation req\_com in the database. This is a more secure method to transfer data than the GET method.
  - After submission, data entered by the user is extracted from the global array \$\_POST.
  - Validation
    - Concern type
      - ◆ If the user has not selected any concern type, then an appropriate error message is displayed below the input column
    - Concern description
      - ◆ If the user has left concern description empty, then an appropriate error message is displayed below the input column
  - In case of any errors, the data will not be entered into the database
  - User's email-id is extracted using \$\_SESSION['username']
  - If there are no errors, then a mysql query is implemented to insert the tuple in the req\_com relation.

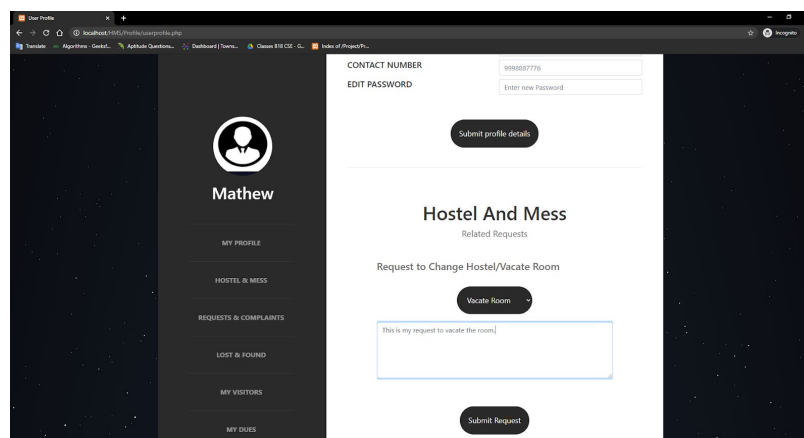
The top screenshot shows a web application interface with a sidebar menu on the left and a main content area. The sidebar menu includes options like 'MY PROFILE', 'HOSTEL & MESS', 'REQUESTS & COMPLAINTS', 'LOST & FOUND', 'MY VISITORS', and 'MY DUES'. The main content area has a section titled 'Request And Complaints' with a form to 'Fill the form in case of Requests and Complaints'. The form has a dropdown menu for 'Concern' (set to 'Complaint') and a text input field for 'This is my complaint'. Below the input field are buttons for 'Submit Request&Complaint' and 'View Request&Complaint'. There is also a section titled 'Lost And Found' with a form to 'Fill the form in case of Lost/Found Item'.

The bottom screenshot shows a table with the following data:

ID	EMAIL	SUBJECT	CONCERN	DATETIME	STATUS
2	mathew@nitc.ac.in	CHANGE HOSTEL	This is my concern to change the hostel.	2020-12-28 19:58:38	PENDING
3	mathew@nitc.ac.in	VACATE ROOM	This is my request to vacate the room.	2020-12-28 19:59:25	PENDING
4	mathew@nitc.ac.in	REQUEST	This is my request.	2020-12-28 20:01:30	PENDING
5	mathew@nitc.ac.in	COMPLAINT	This is my complaint.	2020-12-28 20:01:56	PENDING

## ❖ Change and Vacate Hostel Rooms:

- **Input:** Select Change/Vacate, Add a detailed description
- **Output:** Addition of the requests to the req\_com database.
- Requests and complaints form
  - Get the type of concern (Change/Vacate) and description from the user using a form.
  - POST method is used to send data to the server to update the relation req\_com in the database. This is a more secure method to transfer data than the GET method.
  - After submission, data entered by the user is extracted from the global array \$\_POST.
  - Validation
    - Concern type
      - ◆ If the user has not selected any change/vacate type, then an appropriate error message is displayed below the input column
    - Concern description
      - ◆ If the user has left description field empty, then an appropriate error message is displayed below the input column
  - In case of any errors, the data will not be entered into the database
  - User's email-id is extracted using \$\_SESSION['username']
  - If there are no errors, then a mysql query is implemented to insert the tuple in the req\_com relation.



## ❖ Issue of the Mess Card For a Student:

- **Input:** Select the Mess interested in after looking at the current statistics of the available messes.
- **Output:** Addition of the mess request to the req\_com database.
- Requests and complaints form
  - Get the mess entry from the user using a form.
  - POST method is used to send data to the server to update the relation req\_com in the database. This is a more secure method to transfer data than the GET method.
  - Validation
    - Concern type
      - ◆ If the user has not selected any mess, then an appropriate error message is displayed below the input column .
  - In case of any errors, the data will not be entered into the database.
  - User's email-id is extracted using \$\_SESSION['username']
  - If there are no errors, then a mysql query is implemented to insert the tuple in the req\_com relation.

## ❖ Add and View Lost/Found Details:

- Get the mess entry from the user using a form.
- POST method is used to send data to the server to update the relation req\_com in the database. This is a more secure method to transfer data than the GET method.
- Validation
  - Concern type
    - If the user has not selected a lost/found type, then an appropriate error message is displayed below the input column .
  - Concern name
    - If the user has not inputted a name, then an appropriate error message is displayed below the input column.
  - Concern description
    - If the user has left the description field empty, then an appropriate error message is displayed below the input column.
- In case of any errors, the data will not be entered into the database.
- User's email-id is extracted using `$_SESSION['username']`
- If there are no errors, then a mysql query is implemented to insert the tuple in the lost\_found relation.

ID	STATUS	ITEM	DESCRIPTION	CONTACT	DATE
L2	LOST	Lost item	Description of lost item	mathew@nitc.ac.in	2020-12-28 20:02:58
F2	FOUND	Found item	Description of found item	mathew@nitc.ac.in	2020-12-28 20:03:46

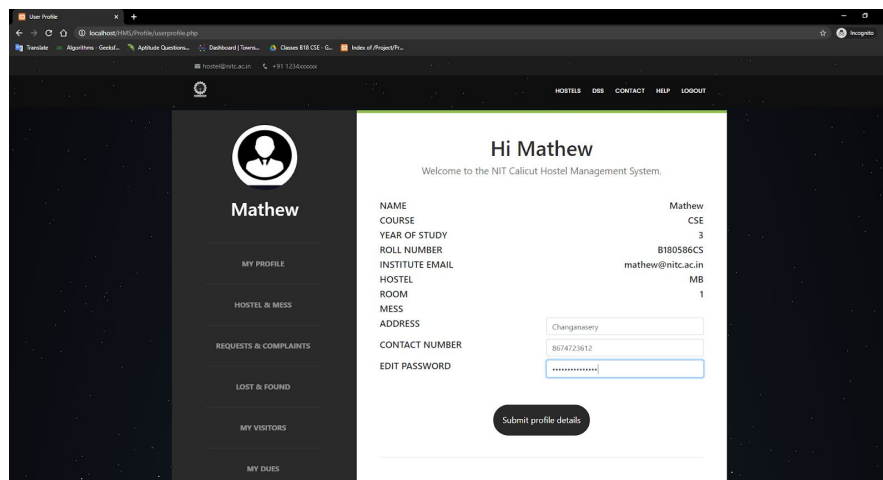
## ❖ Add and View Visitors Details:

- ❖ **Input:** Get the visitor name, visitor's relation to the user, arrival date and time, departure date and time and visitor's contact number from the user using a form.
- ❖ **Output:** Addition of the requests to the visitors database.
- ❖ POST method is used to send data to the server to update the visitors relation in the database.
- ❖ After submission, data entered by the user is extracted from the global array `$_POST`.
- ❖ Validation
  - Visitor name
    - If the user has left visitor name column empty, then appropriate error message is displayed below the input column
  - Visitor's relation to the user
    - If the user has relation column empty, then appropriate error message is displayed below the input column
  - Arrival date
    - If the user has left the arrival date field empty, then an appropriate error message is displayed below the input column
    - A regular expression is used to check if the arrival date is in a valid dd/mm/yyyy format. If the arrival date entered is invalid, then an appropriate error message is displayed below the input column.
  - Arrival time
    - If the user has left the arrival time field empty, then an appropriate error message is displayed below the input column
    - A regular expression is used to check if the arrival time is in a valid hh:mm in 24-hour format. If the arrival time entered is invalid, then an appropriate error message is displayed below the input column.
  - Departure time
    - If the user has left the departure time field empty, then an appropriate error message is displayed below the input column
    - A regular expression is used to check if the departure time is in a valid hh:mm in 24-hour format. If the departure time entered is invalid, then an appropriate error message is displayed below the input column.
- ❖ In case of any errors, the data will not be entered into the database.
- ❖ User's email-id is extracted using `$_SESSION['username']`.

The screenshot displays a web application interface. On the left, a dark sidebar contains a user profile for 'Mathew' with a circular icon. Below the profile, there are navigation links: 'MY PROFILE', 'HOSTEL & MESS', 'REQUESTS & COMPLAINTS', 'LOST & FOUND', 'MY VISITORS', and 'MY DUES'. The main content area is titled 'Visitors' and 'Add Visitor Details'. It features a form with the following fields: 'VISITOR NAME' (with 'Visitor1' entered), 'RELATION' (with 'Relationship' entered), 'CONTACT NUMBER' (with '1234567890' entered), 'ARRIVAL DATE' (with '29/12/2020' entered), 'ARRIVAL TIME' (with '10:00' entered), and 'DEPARTURE TIME' (with '12:00' entered). A 'Submit Visitor Details' button is located at the bottom of the form.

## ❖ Edit Address, Password, Mobile Number:

- The student has the chance to edit their password, mobile number, and address. The previous values will be displayed as a placeholder and these values can be edited by submitting the form.



Hi Mathew

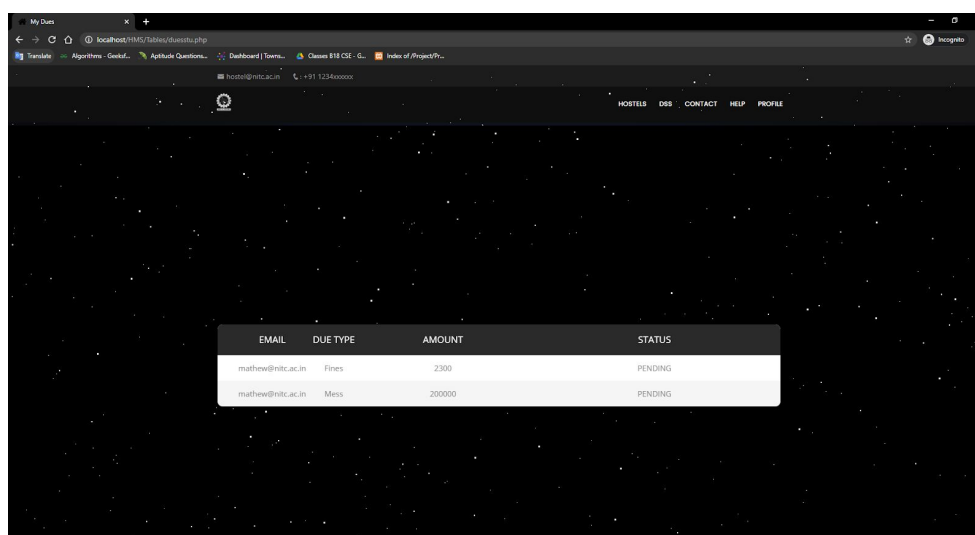
Welcome to the NIT Calicut Hostel Management System.

NAME	Mathew
COURSE	CSE
YEAR OF STUDY	3
ROLL NUMBER	B180506CS
INSTITUTE EMAIL	mathew@nitc.ac.in
HOSTEL	MB
ROOM	1
MESS	
ADDRESS	
CONTACT NUMBER	
EDIT PASSWORD	

Submit profile details

## ❖ View Statuses of the Lost/Found, Requests/Complaints and Dues Table:

- The student has a chance to view the status of any of the requests they have forwarded to the Administrator/Warden. He/she would receive real time updates on the statuses of their issues/ requests.



EMAIL	DUE TYPE	AMOUNT	STATUS
mathew@nitc.ac.in	Fines	2300	PENDING
mathew@nitc.ac.in	Mess	200000	PENDING



## The Administrator/Warden Has The Following Functionalities:

- Allocating a new student.
- Allocating a fellow administrator/warden.
- Allocating a Mess For the Student.
- Allocating Dues For the Students.
- Vacating Rooms for the Students.
- Allocating New Rooms for the Students.
- Manage Lost and Found Entries of the Students.
- Manage Request and Complaint Entries of the Students.
- View all the Students enlisted in the Hostel Management System.
- View all the Visitors visiting a particular Student.

### ❖ Allocating a new student.

- **Input:** Get the student's name, roll number, year of study, email ID, DoB, hostel ID, address, contact and course from the system user using a form.
- **Output:** Addition of the requests to the students and users database.
- Get the student's name, roll number, year of study, email ID, DoB, hostel ID, address, contact and course from the system user using a form.
- POST method is used to send data to the server to update the relation reqcom in the database. This is a more secure method to transfer data than the GET method.
- After submission, data entered by the user is extracted from the global array \$\_POST.
- Next, it is confirmed that the users relation does not already contain this entry.
  - If it does, an appropriate message is displayed. Data is not entered into the database.
  - If it does not, then a MySQL query is implemented to insert the tuple in the users relation using the extracted data.
    - If the above MySQL query gives an error, an appropriate error message is displayed.
- The room capacity and updated number of students in the hostel is extracted from the hostel relation using MySQL commands. The room number of room the student is allocated to is calculated using

◆  $\text{ceiling}(\text{updated\_no\_of\_students} / \text{room\_capacity for a hostel})$ .

The screenshot displays the Admin interface for allocating a new student. On the left is a dark sidebar with a white circular profile icon at the top, followed by the text 'Admin'. Below this are several menu items: 'MY PROFILE', 'ALLOCATE', 'REQUESTS & COMPLAINTS', 'LOST & FOUND', 'VIEW VISITORS', and 'VACATE'. The main content area has a light gray background and is titled 'Enter Details To Allocate A New Student.' It contains a form with the following fields: NAME (Student1), COURSE (Course1), YEAR OF STUDY (2), ROLL NUMBER (8123456XX), INSTITUTE EMAIL (email@nitcac.in), DATE OF BIRTH (28-02-2002 with a calendar icon), HOSTEL ID (A), ADDRESS (Kerala, India), and CONTACT NUMBER (6785412903). Below the form are two buttons: 'Allocate Student' and 'View All Students'. At the bottom of the main area, the text 'Allocate Warden' is visible. To the right of the form is a dark vertical sidebar.

## ❖ Allocating a fellow administrator/warden.

- **Input:** Get the student's name, roll number, year of study, email ID, DoB, hostel ID, address, contact and course from the system user using a form.
- **Output:** Addition of the requests to the students and admins database.
- Get the warden's email, their name, DoB, hostel and contact from the user using a form.
- POST method is used to send data to the server to update the relation reqcom in the database. This is a more secure method to transfer data than the GET method.
- After submission, data entered by the user is extracted from the global array \$\_POST.
- If there is no error, the user is redirected to the admin profile page.
- In case of an error, an appropriate error message is displayed.
- Next, it is confirmed that the users relation does not already contain this entry.
  - If it does, an appropriate message is displayed. Data is not entered into the database.
  - If it does not, then a MySQL query is implemented to insert the tuple in the users relation using the extracted data.

The screenshot shows the 'Admin' panel on the left with a sidebar menu containing 'MY PROFILE', 'ALLOCATE', 'REQUESTS & COMPLAINTS', 'LOST & FOUND', 'VIEW VISITORS', and 'VACATE'. The main content area is titled 'Allocate Warden' and includes a sub-header 'Enter Details To Allocate A New Hostel Incharge'. The form fields are: NAME (with a dropdown menu), INSTITUTE EMAIL (text input), DATE OF BIRTH (date input), HOSTEL INCHARGE OF (dropdown menu), and CONTACT NUMBER (text input). There are two buttons: 'Allocate Student' and 'View All Students' at the top, and 'Allocate Warden' at the bottom.

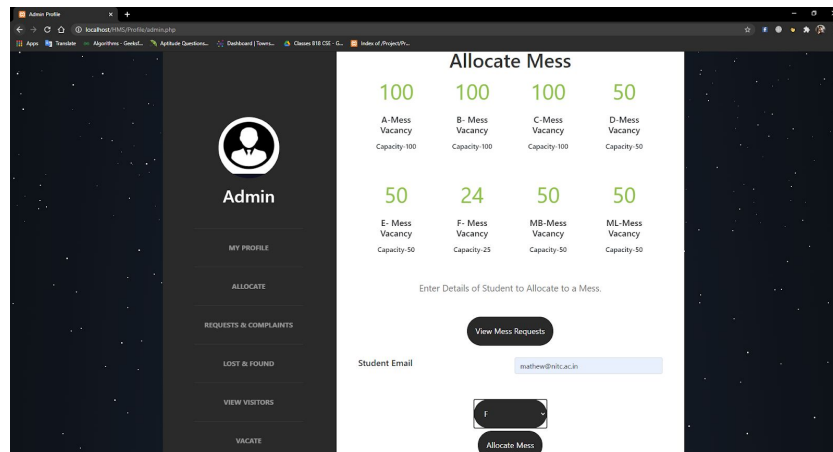
## ❖ Allocating Dues

- **Input:** Enter Student Email, Enter the due amount, Select type of due: Mess, Hostel and Fine
- **Output:** Addition of the above details to dues table in database.
- Get the student email, due amount and due type using a form
- POST method is used to send data to the backend server and insert the values to the database
- In the backend page, the data is extracted using \$\_POST
- After that, an SQL query is used to insert the extracted data into the database
  - If there is any error in executing the SQL, then an appropriate error message is displayed accordingly.
  - If there is no error, then its redirected back to the admin page

The left screenshot shows the 'Allocate Dues' form. It has a sidebar menu similar to the previous one. The main content area is titled 'Allocate Dues' with the sub-header 'Enter Dues of Student'. The form fields are: Student Email (text input), Dues (text input), and a dropdown menu for 'Type'. There are buttons for 'Choose Mess', 'Allocate Mess', 'Allocate Dues', and 'View Dues'. The right screenshot shows the 'Vacate Student' form. It has the same sidebar menu. The main content area is titled 'Vacate Student' with the sub-header 'Enter Details To Vacate Student'. The form has a single text input field for 'Student Email' and a 'Vacate Student' button.

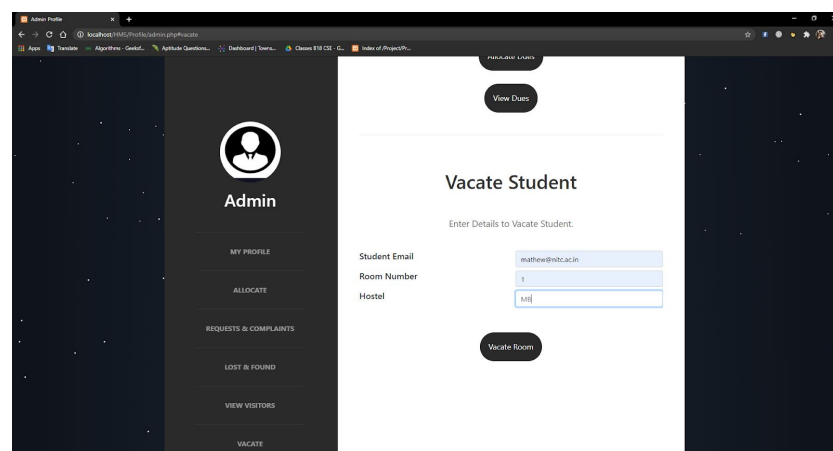
## ❖ Allocating Mess

- **Input:** Enter Student Email, Select Mess from the mess options
- **Output:** Updating the mess value in the student table in the database corresponding to the given student email.
- Get the student email, mess option using a form
- POST method is used to send data to the backend server and insert the values to the database
- In the backend page, the data is extracted using `$_POST`
- An SQL query is used to update the data in the database
  - If there is any error in executing the SQL, then an appropriate error message is displayed accordingly.
  - If there is no error, then its redirected back to the admin page



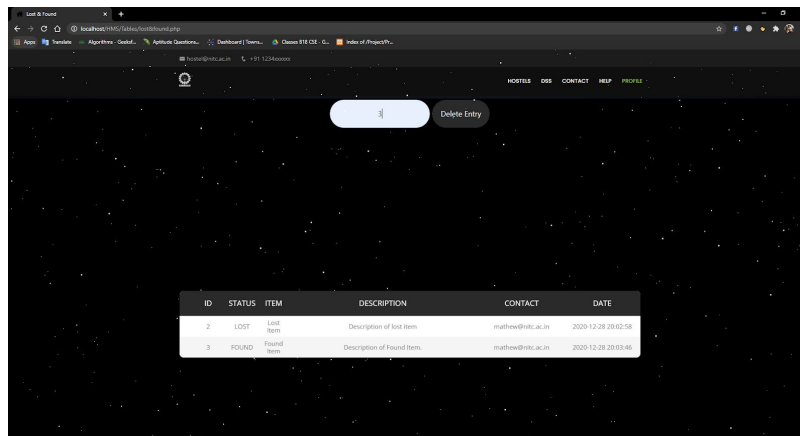
## ❖ Vacating Rooms for the Students.

- **Input:** Student's email, room number and hostel ID from the system user using a form.
- **Output:** The Student Room Entry and Corresponding Hostel Entry will be deleted.
- Get the vacating student's email, room number and hostel ID from the system user using a form.
- POST method is used to send data to the server to update the relation reqcom in the database. This is a more secure method to transfer data than the GET method.
- After submission, data entered by the user is extracted from the global array `$_POST`.
- The tuple corresponding to the student from 'student' relation is fetched. In case of an error, an appropriate error message is displayed. Next, it is checked if the room number and hostel ID match with those in the tuple
  - If they do not match, the user is asked to enter the correct details. Data is not updated in the database.



## ❖ Manage Lost and Found Entries of the Students.

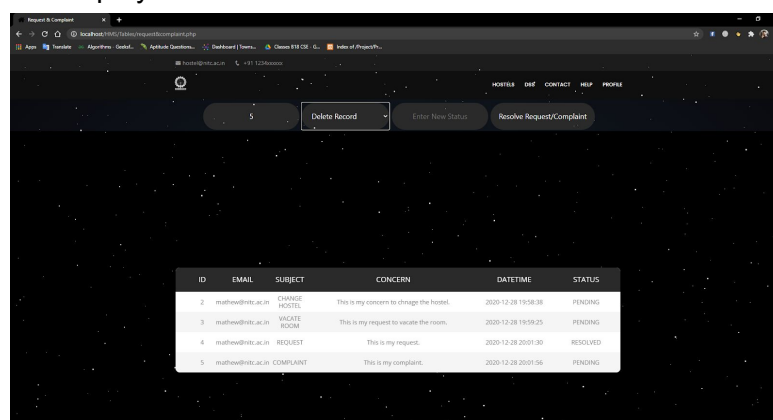
- **Input:** Enter LF\_ID
- **Output:** Deletion of tuples in lost\_and\_found relation in the database.
- POST method is used to send data to the server to update the relation reqcom in the database. This is a more secure method to transfer data than the GET method.
- After submission, data entered by the user is extracted from the global array \$\_POST.
- On clicking the Lost & Found tab, the admin is redirected to a new page of lost and found.
- The new page displays all the lost and found items the students have submitted and also a form for the admin, where he can manage the lost and found.
- **Manage Form:**
  - Deletion of the tuples of lost\_and\_found relation in the database, thereby modifying the table displayed.



ID	STATUS	ITEM	DESCRIPTION	CONTACT	DATE
2	LOST	Lost Item	Description of lost item	mathew@nitc.ac.in	2020-12-28 20:02:58
3	FOUND	Found Item	Description of Found Item	mathew@nitc.ac.in	2020-12-28 20:03:40

## ❖ Manage Request and Complaint Entries of the Students.

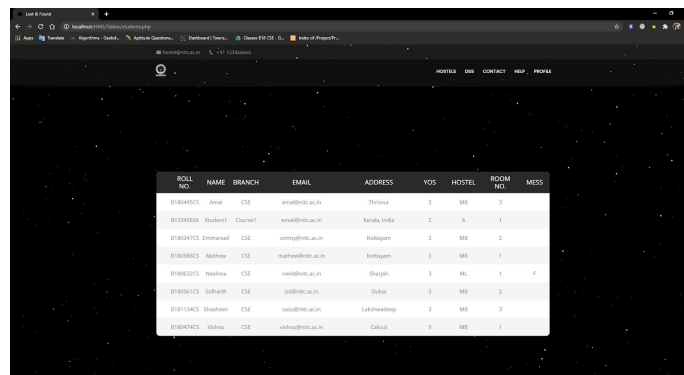
- **Input:** Enter RC\_ID, Select Update Status/Delete Record, Enter New Status
- **Output:** Updation or Deletion of tuples in req\_com relation in the database.
- POST method is used to send data to the server to update the relation reqcom in the database. This is a more secure method to transfer data than the GET method.
- After submission, data entered by the user is extracted from the global array \$\_POST.
- On clicking the Request & Complaints tab, the admin is redirected to a new page of request and complaints.
- The new page displays all the requests and complaints the students have submitted and also a form for the admin, where he can update and manage the requests and complaints.
- **Update/Manage Form:**
  - Updation and deletion to the req\_comp relation in the database, thereby modifying the table displayed.



ID	EMAIL	SUBJECT	CONCERN	DATETIME	STATUS
2	mathew@nitc.ac.in	CHANGE HOSTEL	This is my concern to change the hostel.	2020-12-28 19:08:36	PENDING
3	mathew@nitc.ac.in	VACATE ROOM	This is my request to vacate the room.	2020-12-28 19:58:25	PENDING
4	mathew@nitc.ac.in	REQUEST	This is my request.	2020-12-28 20:01:30	RESOLVED
5	mathew@nitc.ac.in	COMPLAINT	This is my complaint.	2020-12-28 20:01:56	PENDING

## ❖ View all the Students enlisted in the Hostel Management System:

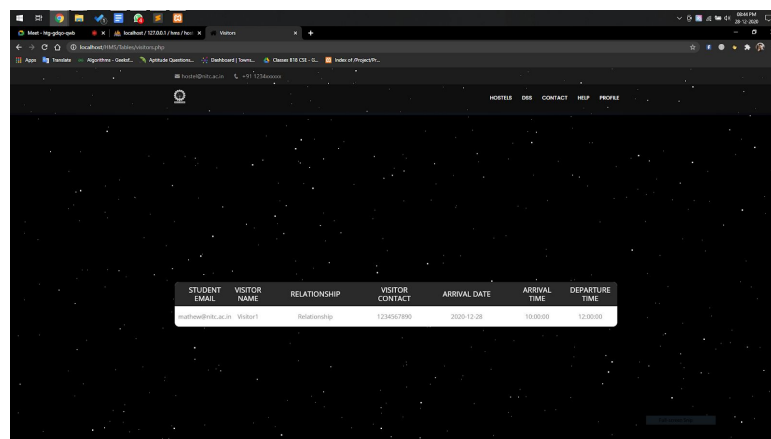
- The Administrator has the option to view all the students that are part of the hostel management system. This is done by merging the student table and user table joining them by the email attribute of both tables.



ROLL NO.	NAME	BRANCH	EMAIL	ADDRESS	YOS	HOSTEL	ROOM NO.	MESS
0180806103	Amit	CSE	amit@nitk.ac.in	Hydrabad	3	MB	3	
0122455004	Student1	Chemist	email@nitk.ac.in	Kerala India	2	A	1	
0180806103	Emmanuel	CSE	emmy@nitk.ac.in	Kottayam	3	MB	2	
0180806103	Mathew	CSE	mathew@nitk.ac.in	Kottayam	3	MB	1	
0180806103	Neelima	CSE	neel@nitk.ac.in	Shangai	3	ML	1	F
0180806103	Subhath	CSE	subhath@nitk.ac.in	Dubai	3	MB	2	
0180806103	Shashan	CSE	shashan@nitk.ac.in	Lakshwarp	3	MB	3	
0180806103	Vishnu	CSE	vishnu@nitk.ac.in	Calicut	2	MB	1	

## ❖ View all the Visitors visiting a particular Student:

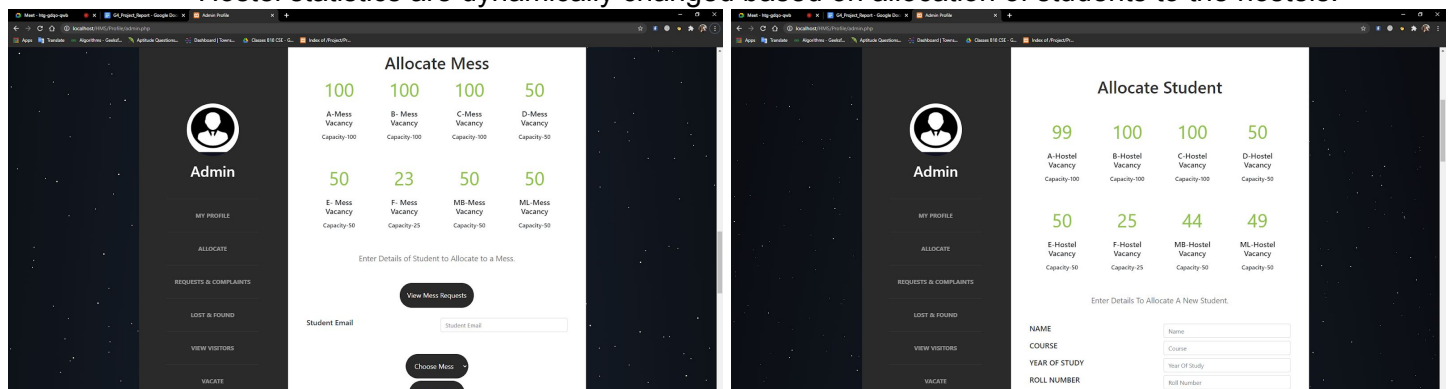
- The student has a chance to view the status of any of the requests they have forwarded to the Administrator/Warden. He/she would receive real time updates on the statuses of their issues/ requests.



STUDENT EMAIL	VISITOR NAME	RELATIONSHIP	VISITOR CONTACT	ARRIVAL DATE	ARRIVAL TIME	DEPARTURE TIME
mathew@nitk.ac.in	Visitor1	Relationship	1234567890	2020-12-28	10:00:00	12:00:00

## ❖ Hostel and Mess Statistics for all the Hostels:

- Mess statistics are dynamically changed based on allocation of students to the messes.
- Hostel statistics are dynamically changed based on allocation of students to the hostels.

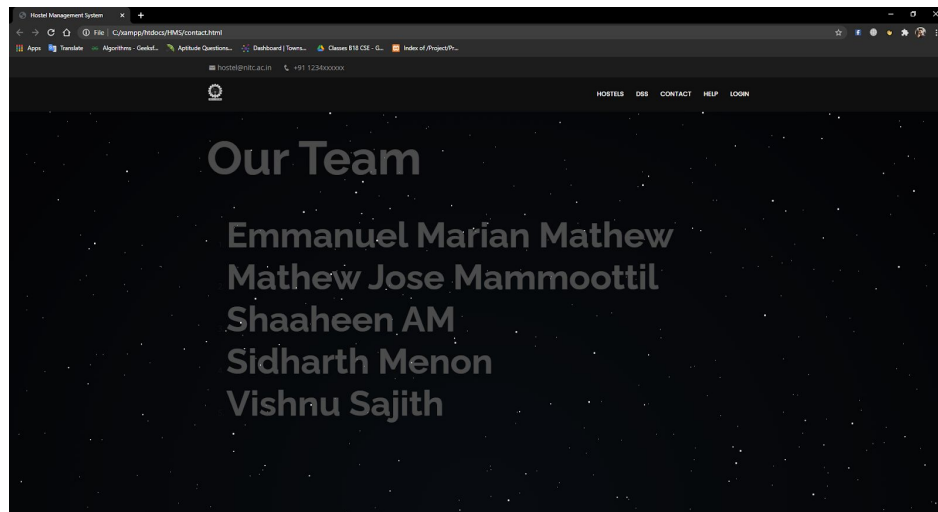


**Room Number Calculated Using the Formula:**  

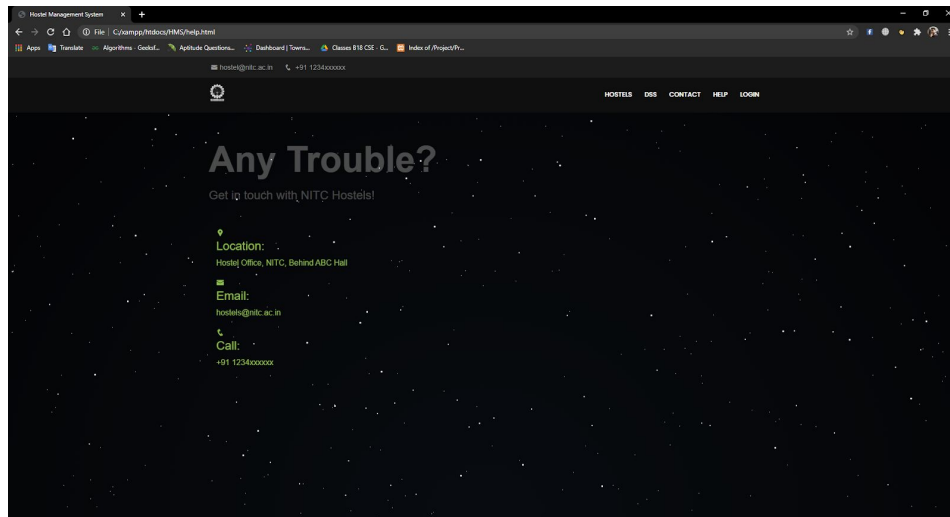
$$\text{ceiling}(\text{updated\_no\_of\_students} / \text{room\_capacity for a hostel}).$$

## OUR TEAM, HELP, HOSTEL STATISTICS PAGES :

### ❖ OUR TEAM



### ❖ HELP PAGE



### ❖ HOSTEL STATISTICS

