

HOSTEL MANAGEMENT SYSTEM (HMS)

Software Engineering Project

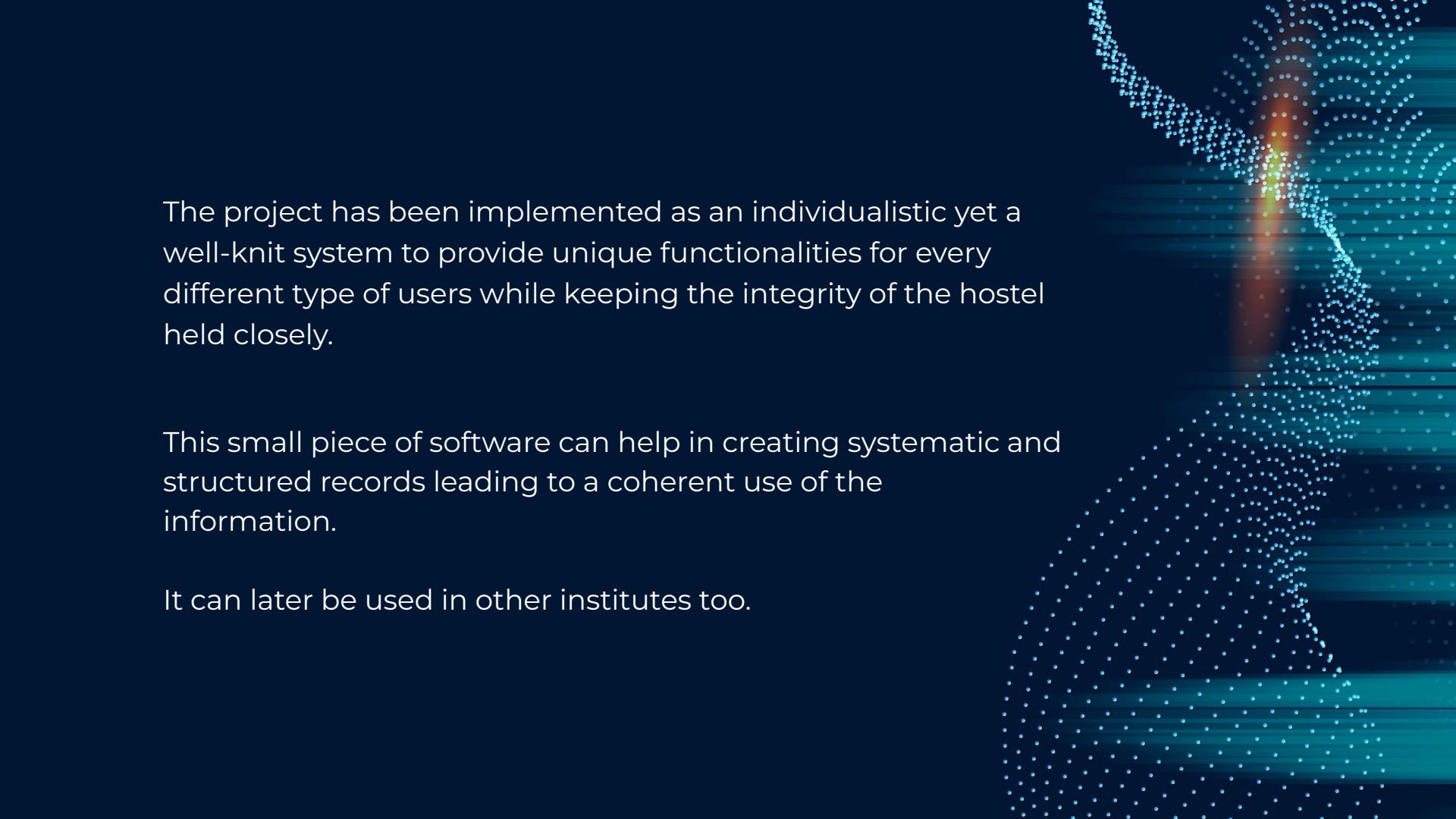
By - Omang Rawat	(1901126)
Rahul Soni	(1901154)
Vaibhav Kesarwani	(1901212)
Vishesh Sharma	(1901217)

INTRODUCTION

The purpose of developing this “Hostel Management System” is to keep the students and the hostel administration connected through a virtual platform rather than maintaining records manually using registers.

It will integrate the benefits of better organizability with effortless accessibility leading to an increased efficiency for the workforce and convenience for all.





The project has been implemented as an individualistic yet a well-knit system to provide unique functionalities for every different type of users while keeping the integrity of the hostel held closely.

This small piece of software can help in creating systematic and structured records leading to a coherent use of the information.

It can later be used in other institutes too.

SHORT OVERVIEW

- **Login Interface:** Interface for students, warden / caretaker, guards / cleaning services to login onto the virtual platform.
- **Dashboard Updates:** Notification window where warden / caretaker / guards can post some hostel regarding updates.
- **Record Storage:** Keeping a centralized database which would be maintained via the online system
 - Managing courier / deliveries
 - Managing room related issues
 - Managing sports equipments
 - Managing hostel cleaning issues
- **Authentication:** Restricted authentication for only institute employees and students
- **Administration Access:** Administration would be able to keep an eye on the records of students

FUNCTIONAL REQUIREMENTS

- Users must have a valid User ID and password to login thus having their individual profiles.
- Wardens can register new students and update any existing records.
- Students can check for the status of their deliveries with hostel guards.
- Students can inform if room cleaning services are required.
- Guards can update the status (receive / submit) of sports equipment.
- Dashboard for any notices or updates from wardens and guards.
- Students can send queries regarding any hostel issues to the associated administration.
- Caretakers can register new guards and cleaning staff.
- Cleaning staff can keep track of the cleaning requests, accept and handle them accordingly.

NON-FUNCTIONAL REQUIREMENTS

Requirements specifying criteria that can be used to judge the operation of a system are called non-functional requirements. For our system, they are as follows:

- Secure access of any confidential data
- Round-the-clock-service availability
- Better component design to get efficiency at peak time
- Flexible service based architecture (MVC pattern) will be highly desirable for future extension.
- CRUD implementation and limited accessibility for users according to their respective roles.

Logic Implementation

ROLE DISTRIBUTION

We basically have 5 different types of users who can log in through our login page and thus get directed to their respective home pages as per the roles they have been assigned.

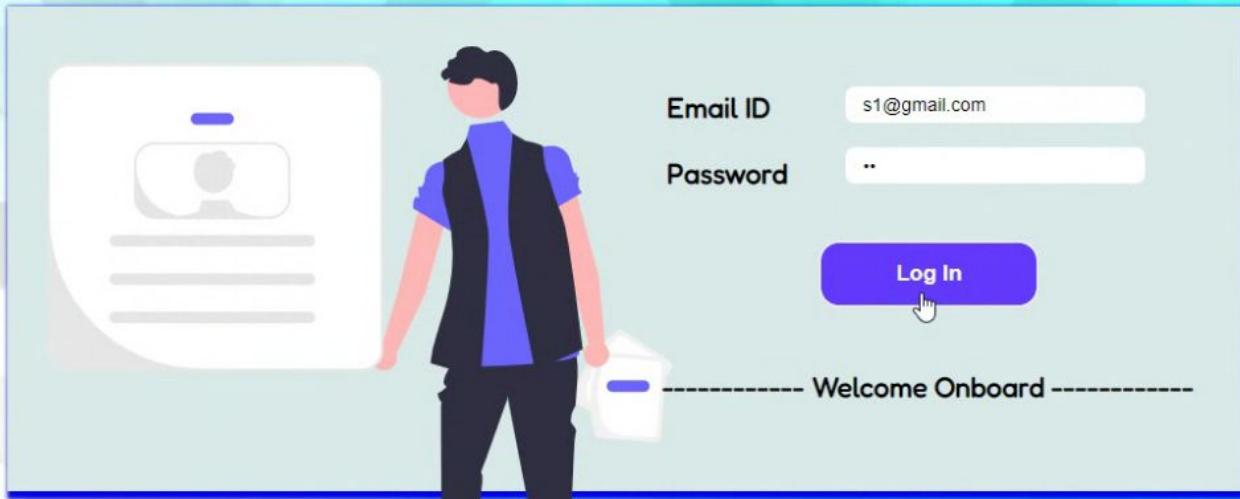
We have divided the roles as :

- Student
- Guard
- Cleaning Service
- Caretaker
- Warden

which can be viewed in the database as numbers from 1-5 respectively to keep the naming of the roles more dynamic and keep the database more independent.

LOGIN PAGE

- To handle the information of every role we have a different collection set in the database (caretakers, employees, guards, students, wardens) so that data about every user would be more easily accessible and cleanly assembled.
- The page fetches UID and role from the documents according to the email and directs them to their respective home pages accordingly.
- The functionalities of each user have been developed according to their respective authority and the work they do.



Email ID

s1@gmail.com

Password

..

Log In

----- Welcome Onboard -----

STUDENT INTERFACE

- **Home Page**

- **Profile** - Every student can view their homepage which holds all the personal information about themselves that the database holds which is fetched via API call to the “students” table in the database with UID as a primary key.
- **Cleaning Request** - A student can request cleaning of their rooms with a click of a button, which sends a direct request to the cleaning service which is handled by sending the UID of the student as the sender’s ID and the room no. of the student to the “requests” table as cleaning request in the DB.

- **Room Query** - Students can also put any other queries or complaints from the homepage which are added to the request table by similarly sending the sender's id and room no. which can be viewed by the guards.
 - **Sports Equipment Request** - Students can request any sports equipment, the POST request with equipment, student UID is sent to guards and saved in the “sports” collection in DB.
-
- **Noticeboard**
 - Students have access to the dashboard where they can get any important notices that might have been posted by Warden or Caretaker which are fetched from the “notifies” collection in DB.

**Hostel life is a great event
that teaches about
difficulties of life.**

Report Hostel Issue

Cleaning Request

Sports Equipment Request

Change Password

(+91)9414253689

CSE

Major

2023

Passing Year

356

Room No



Test Student

1901127

test@gmail.com

Hostel Issue

My Room's door hinge is broken.
Kindly do the needful.

Submit

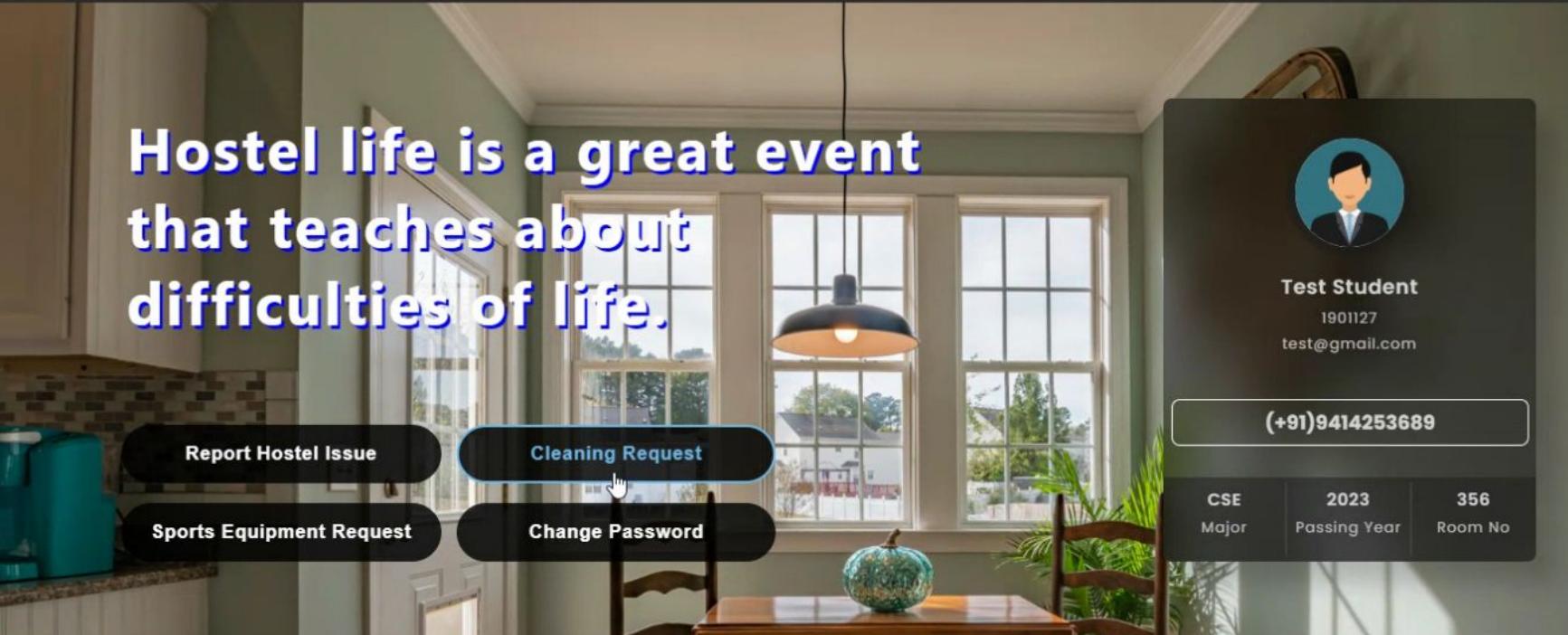
Hostel
that
diffic

Report H

Sports Equip



2023 356
Missing Year Room No



**Hostel life is a great event
that teaches about
difficulties of life.**

Report Hostel Issue

Cleaning Request

Sports Equipment Request

Change Password



Test Student

1901127

test@gmail.com

(+91)9414253689

CSE

Major

2023

Passing Year

356

Room No

localhost:3000 says

Request Sent!!!!

OK

Hostel life is a great event that teaches about difficulties of life.

Report Hostel Issue

Cleaning Request

Sports Equipment Request

Change Password



Test Student

1901127

test@gmail.com

(+91)9414253689

CSE
Major

2023
Passing Year

356
Room No

HMS

localhost:3000 says

Request already sent...Your room will be cleaned soon....

Home

Dashboard

Profile

Logout

OK

Hostel life is a great event that teaches about difficulties of life.

[Report Hostel Issue](#)[Cleaning Request](#)[Sports Equipment Request](#)[Change Password](#)

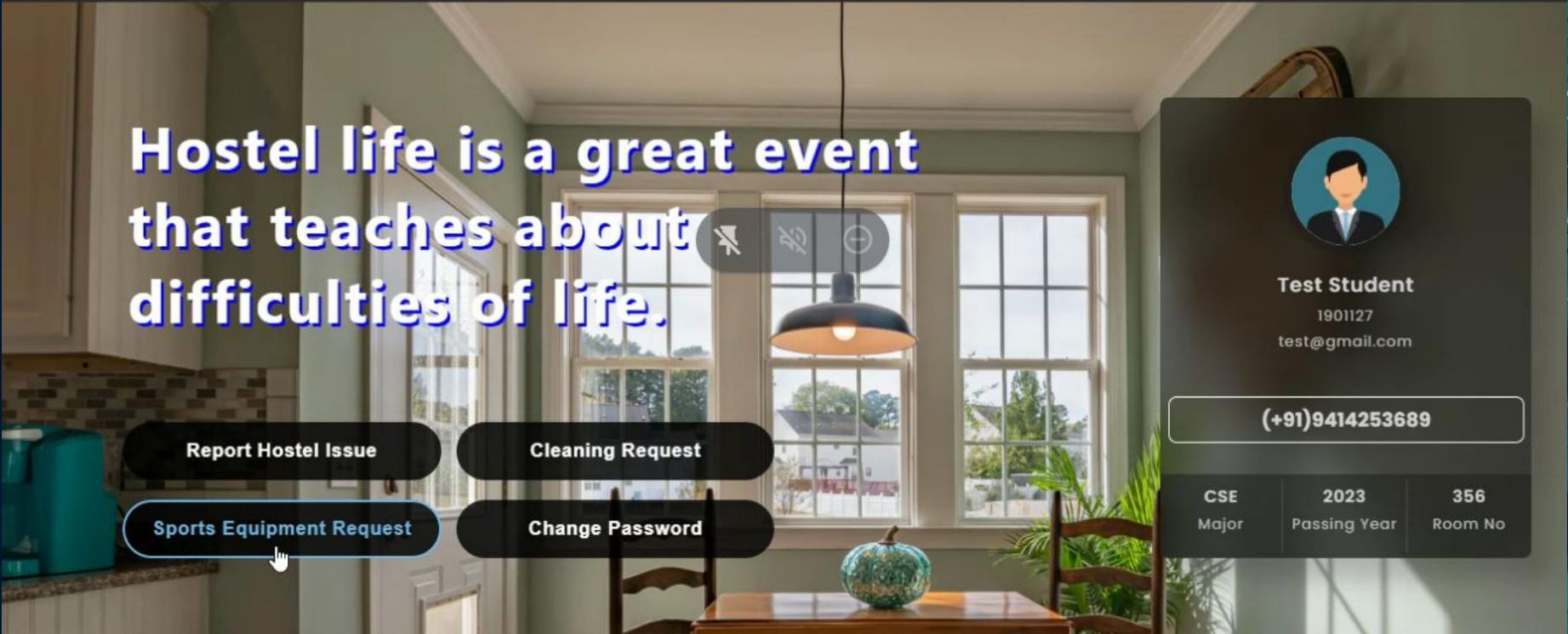
Test Student

1901127

test@gmail.com

(+91)9414253689

CSE
Major2023
Passing Year356
Room No



**Hostel life is a great event
that teaches about
difficulties of life.**

[Report Hostel Issue](#)[Cleaning Request](#)[Sports Equipment Request](#)[Change Password](#)[\(+91\)9414253689](#)

CSE
Major

2023
Passing Year

356
Room No



Test Student

1901127

test@gmail.com

Hostel
that
diffic

Report H

Sports Equipment Request

Change Password

Sports Equipment

- Badminton
- Badminton
- Football
- Cricket Kit
- Table Tennis Bat
- Table Tennis Ball
- Chess
- Carrom Coins

Submit



1st Student

1901127

@gmail.com

9414253689

CSE
Major

2023
Passing Year

356
Room No

Hostel
that
diffic

Report H

Sports Equipment Request

Change Password



Sports Equipment

Badminton

1

Submit



1st Student

1901127

@gmail.com

9414253689

CSE
Major

2023
Passing Year

356
Room No

React App

localhost:3000/student

Gmail YouTube Maps

HMS

Home Dashboard Profile Logout

Hostel life is a great event that teaches about difficulties of life.

Report Hostel Issue

Cleaning Request

Sports Equipment Request

Change Password

Test Student
1901127
test@gmail.com

(+91)9414253689

CSE Major	2023 Passing Year	356 Room No
--------------	----------------------	----------------

This image shows a screenshot of a web application for a Hostel Management System (HMS). The page is titled 'HMS' at the top left. At the top right, there are links for 'Home', 'Dashboard', 'Profile', and a red 'Logout' button. The main content area features a photograph of a bright, airy room with large windows, a dining table, and chairs. Overlaid on this image is a large, semi-transparent text box containing the message: 'Hostel life is a great event that teaches about difficulties of life.' Below this text are several buttons: 'Report Hostel Issue', 'Cleaning Request', 'Sports Equipment Request', and 'Change Password'. On the right side of the screen, there is a dark sidebar containing a user profile picture of a person in a suit, the text 'Test Student', the ID '1901127', the email 'test@gmail.com', and a phone number '(+91)9414253689'. At the bottom, there is a table with three columns: 'CSE Major', '2023 Passing Year', and '356 Room No'.

Hostel life is that teaches difficulties of

[Report Hostel Issue](#)[Sports Equipment Request](#)[Change Password](#)

ENTER NEW PASSWORD

Enter Old Password

Enter New Password

Enter New Password Again

[Change Password](#)

Test Student

1901127

test@gmail.com

(+91)9414253689

CSE

2023

356

Major

Passing Year

Room No

2022-03-27 12:40:12

Dear All,
hi there, free smoothies for all today

Posted By : Vishesh Sharma

2022-03-25 12:40:12

Dear All,
It is to inform you that the GPU server (IP: 172.16.2.17) will be down from 12 pm IST to 2 pm IST today due to some urgent maintenance work.

Posted By : ICT Section, IIITG

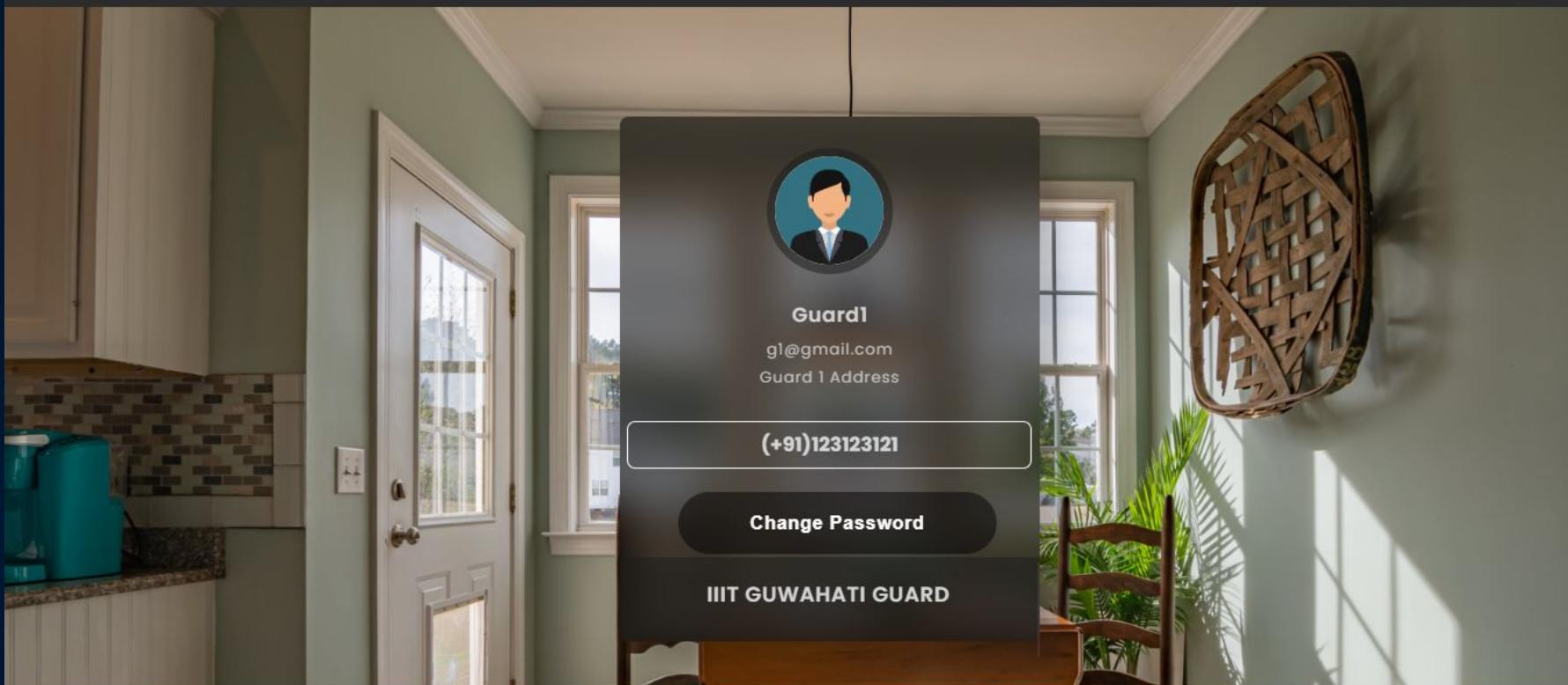
2022-03-25 12:40:12

Dear All,

GUARD INTERFACE

- **Profile**
 - Guard's personal information is fetched from the API call to the "guard" table.
- **Room Cleaning Service**
 - Status of cleaning staff (Assigned/Not-Assigned) is fetched from the "requests" table.
- **Room Queries**
 - Room Queries are fetched from the "requests" table and can be viewed by the guards on the room queries dashboard

- **Delivery Updates**
 - New delivery can be added by the guards by filling up details in the form.
 - It displays the data fetched from the “delivers” collection and guards can change status of delivery to delivered when the package has been taken by the student, which is then reflected in the DB.
- **Sports Equipment Request**
 - All the requests for the sports equipment by the students, fetched from the “sports” collection are displayed in the table.
 - Guards can accept requests when the equipments are issued to the students, which would send a POST request sending student’s UID as the issuer ID.
- **Close Sports Equipment Request**
 - The issued equipments are displayed in the table which are fetched by api call to the sports collection.



2022-03-27 12:40:12

Dear All,
hi there, free smoothies for all today

Posted By : Vishesh Sharma

2022-03-25 12:40:12

Dear All,
It is to inform you that the GPU server (IP: 172.16.2.17) will be down from 12 pm IST to 2 pm IST today due to some urgent maintenance work.

Posted By : ICT Section, IIITG

2022-03-25 12:40:12

Dear All,

Navigation

 Hostel Issues Cleaning Requests Delivery Updates Sports Eqp Request Close Sports Eqp Request

Roll No	Name	Room No	Mobile	Read Issue
1901127	Test Student	356	9414253689	<button>OPEN ISSUE</button>
1901212	Vaibhav Kesarwani	422	9695070039	<button>OPEN ISSUE</button>

Rows per page: 10 ▾ 1-2 of 2 < >

React App

localhost:3000/guards

Gmail YouTube Maps

HMS

Home Dashboard Profile Logout

Navigation

- Hostel Issues**
- Cleaning Requests
- Delivery Updates
- Sports Eqp Request
- Close Sports Eqp Request

Roll No	Name	Room No	Mobile	Read Issue
1901127	Test Student	356	9414253689	<button>OPEN ISSUE</button>
1901212	Vaibhav Kesarwani	422	9695070039	<button>OPEN ISSUE</button>

Rows per page: 10 < 1-2 of 2 >

Navigation

 Hostel Issues Cleaning Requests Delivery Updates Sports Eqp Request Close Sports Eqp Request

X

ISSUE OF "Test Student"

My Room's door hinge is broken. Kindly do the needful.

 CLOSE REQUEST

Navigation

 Hostel Issues

 Cleaning Requests

 Delivery Updates

 Sports Eqp Request

 Close Sports Eqp Request

Roll No	Name	Room No	Mobile	Read Issue
1901212	Vaibhav Kesarwani	422	9695070039	<button>OPEN ISSUE</button>

Rows per page: 10 < 1-1 of 1 >

Navigation

Hostel Issues

Cleaning Requests

Delivery Updates

Sports Eqp Request

Close Sports Eqp Request

Date	Room Number	Name	Phone No
2022-04-15T13:30:48.920Z	422	Employee 2	9585635245
2022-04-17T17:38:46.465Z	356	Not Assigned	Not Assigned

Rows per page: 10 < 1-2 of 2 >

Navigation

Hostel Issues

Cleaning Requests

Delivery Updates

Sports Eqp Request

Close Sports Eqp Request

Add Delivery

Delivery Id	Name	Date	Change Status
865642152	Vaibhav Kesarwani	2022-03-27T18:30:00.000Z	<button>CHANGE STATUS</button>
654564	Rahul Soni	2022-03-27T18:30:00.000Z	<button>CHANGE STATUS</button>
9656455	Omang Rawat	2022-03-27T18:30:00.000Z	DELIVERED
85662	Sarthak Gupta	2022-03-27T18:30:00.000Z	DELIVERED
123123123123	Utkarsh	2022-03-27T21:06:04.077Z	DELIVERED

Rows per page: 10 < > 1-5 of 5

Navigation

Hostel Issues

Cleaning Requests

Delivery Updates

Sports Eqp Request

Close Sports Eqp Request

X

Delivery Id	Name	Date	Change Status
8656			<button>CHANGE STATUS</button>
6545			<button>CHANGE STATUS</button>
9656			DELIVERED
8566			DELIVERED
1231			DELIVERED

Delivery Form

Delivery Id: 789456123

Name: Sample Delivery

Add Delivery

Navigation

Hostel Issues

Cleaning Requests

Delivery Updates

Sports Eqp Request

Close Sports Eqp Request

Add Delivery

Delivery Id	Name	Date	Change Status
865642152	Vaibhav Kesarwani	2022-03-27T18:30:00.000Z	<button>CHANGE STATUS</button>
654564	Rahul Soni	2022-03-27T18:30:00.000Z	<button>CHANGE STATUS</button>
9656455	Omang Rawat	2022-03-27T18:30:00.000Z	DELIVERED
85662	Sarthak Gupta	2022-03-27T18:30:00.000Z	DELIVERED
123123123123	Utkarsh	2022-03-27T21:06:04.077Z	DELIVERED
789456123	Sample Delivery	2022-04-17T18:27:39.669Z	<button>CHANGE STATUS</button>

Rows per page: 10 < 1–6 of 6 >

Navigation

 Hostel Issues Cleaning Requests Delivery Updates Sports Eqp Request Close Sports Eqp Request

Add Delivery

Delivery Id	Name	Date	Change Status
865642152	Vaibhav Kesarwani	2022-03-27T18:30:00.000Z	CHANGE STATUS
654564	Rahul Soni	2022-03-27T18:30:00.000Z	CHANGE STATUS
9656455	Omang Rawat	2022-03-27T18:30:00.000Z	DELIVERED
85662	Sarthak Gupta	2022-03-27T18:30:00.000Z	DELIVERED
123123123123	Utkarsh	2022-03-27T21:06:04.077Z	DELIVERED
789456123	Sample Delivery	2022-04-17T18:27:39.669Z	DELIVERED

Rows per page: 10 < 1-6 of 6 >

Navigation



Hostel Issues



Cleaning Requests



Delivery Updates



Sports Eqp Request



Close Sports Eqp Request

Name	Roll Number	Equipment	Accept Request
Test Student	1901127	Football	<button>ACCEPT</button>

Rows per page: 10 < 1-1 of 1 >

Navigation



Hostel Issues



Cleaning Requests



Delivery Updates



Sports Eqp Request



Close Sports Eqp Request

Name	Roll Number	Equipment	Accept Request
Test Student	1901127	Football	<button>CLOSE REQUEST</button>
Vaibhav Kesarwani	1901212	Cricket Kit	<button>CLOSE REQUEST</button>

Rows per page: 10 < 1-2 of 2 >

Navigation



Hostel Issues



Cleaning Requests



Delivery Updates



Sports Eqp Request



Close Sports Eqp Request

Name	Roll Number	Equipment	Accept Request
Vaibhav Kesarwani	1901212	Cricket Kit	CLOSE REQUEST

Rows per page: 10 < 1-1 of 1 >

CLEANING SERVICE INTERFACE

- **Profile**
 - Cleaning Service's personal information is fetched from the API call to the "employees" table.
- **Cleaning Requests**
 - Cleaning Requests from the "requests" collection is fetched and displayed in the table.
 - Cleaning requests are accepted by the cleaning service according to the floors assigned to each employee.
- **Accepted Requests**
 - Cleaning Requests accepted by the cleaning service is displayed in the table.
 - Accepted cleaning requests can marked as done after the assigned room has been cleaned.

Navigation

 Cleaning Requests Accepted Requests

Room Number	Date	Accept Request
356	2022-04-17T17:38:46.465Z	
422	2022-04-17T18:39:38.961Z	

Rows per page: 10 ▾ 1-2 of 2 < >

Navigation

 Cleaning Requests Accepted Requests

Room Number	Date	Accept Request
422	2022-04-17T18:39:38.961Z	<button>Accept</button>

Rows per page: 10 ▾ 1-1 of 1 < >

React App

localhost:3000/cleaningService/acceptedRequest

Gmail YouTube Maps

HMS

Home Dashboard Profile Logout

Navigation

- Cleaning Requests
- Accepted Requests

Room Number	Date	Mark As Done
356	2022-04-17T17:38:46.465Z	<button>Done</button>

Rows per page: 10 1-1 of 1 < >

The screenshot shows a web-based application interface titled "HMS". At the top, there's a navigation bar with links for "Home", "Dashboard", "Profile", and "Logout". On the left, a sidebar titled "Navigation" contains two items: "Cleaning Requests" and "Accepted Requests", with "Accepted Requests" being the active one. The main content area features a modal dialog with a blue header. Inside the dialog, there are two columns: "Room Number" (containing the value "356") and "Date" (containing the value "2022-04-17T17:38:46.465Z"). To the right of these columns is a button labeled "Mark As Done". Below the dialog, there are pagination controls showing "Rows per page: 10" and "1-1 of 1". The background of the main content area has a blue hexagonal pattern.

Navigation

- Cleaning Requests
- Accepted Requests

Room Number	Date	Mark As Done
		Rows per page: 10 0–0 of 0 < >

WARDEN INTERFACE

- **Profile**

- Warden's personal information is fetched from the API call to the "wardens" table.

- **Student**

- Student details can be added / edited / deleted by making respective POST request to the "students" collection..
 - All the document of the student are fetched from the "students" collection and displayed

Navigation

 Students[Add Student](#)

Roll No	Name	Branch	Room No	Mobile	Delete
1901212	Vaibhav Kesarwani	CSE	422	9695070039	DELETE
1901126	Omang Rawat	CSE	351	8510025035	DELETE
1901127	Test Student	CSE	356	9414253689	DELETE
1901012	Aditya Shara	CSE	451	9898098765	DELETE

Rows per page: 10 < 1-4 of 4 >

Navigation

 Students

Student

Register Form

Email ID	<input type="text" value="abhishek@iitg.ac.in"/>	Password	<input type="password" value="....."/>
Roll No	<input type="text" value="1901125"/>	Name	<input type="text" value="Abhishek Gupta"/>
Phone No	<input type="text" value="9456896525"/>	Graduation Year	<input type="text" value="2021"/>
Branch	<input type="text" value="CSE"/>	Address	<input type="text" value="Kanpur, UP"/>
Room No	<input type="text" value="752"/>		

 Register

Navigation

 Students

Add Student

Roll No	Name	Branch	Room No	Mobile	Delete
1901212	Vaibhav Kesarwani	CSE	422	9695070039	<button>DELETE</button>
1901126	Omang Rawat	CSE	351	8510025035	<button>DELETE</button>
1901127	Test Student	CSE	356	9414253689	<button>DELETE</button>
1901012	Aditya Shara	CSE	451	9898098765	<button>DELETE</button>
1901127	Abhishek Gupta	CSE	752	9586236541	<button>DELETE</button>

Rows per page: 10 < 1-5 of 5 >

Navigation



Students

Add Student

Roll No	Name	Branch	Room No	Mobile	Delete
1901212	Vaibhav Kesarwani	CSE	422	9695070039	<button>DELETE</button>
1901126	Omang Rawat	CSE	351	8510025035	<button>DELETE</button>
1901127	Test Student	CSE	356	9414253689	<button>DELETE</button>
1901012	Aditya Shara	CSE	451	9898098765	<button>DELETE</button>

Rows per page: 10 ▾ 1–4 of 4 < >

Navigation

 Students

Add Student

Roll No	Name	Branch	Room No	Mobile	Delete	Edit
1901212	Vaibhav Kesarwani	CSE	422	9695070039	<button>DELETE</button>	<button>EDIT</button>
1901126	Omang Rawat	CSE	351	8510025035	<button>DELETE</button>	<button>EDIT</button>
1901127	Test Student	CSE	356	9414253689	<button>DELETE</button>	<button>EDIT</button>
1901012	Aditya Shara	CSE	451	9898098765	<button>DELETE</button>	<button>EDIT</button>

Rows per page: 10 ▾ 1-4 of 4 < >

X

Register Form

Email ID	<input type="text" value="test@gmail.com"/>	New Password	<input type="text" value="Enter new password"/>
Roll No	<input type="text" value="1901127"/>	Name	<input type="text" value="Test Student NEW"/>
Phone No	<input type="text" value="9414253689"/>	Graduation Year	<input type="text" value="2023"/>
Branch	<input type="text" value="CSE"/>	Address	<input type="text" value="preet vihar, delhi"/>
Room No	<input type="text" value="356"/>		

[Apply Edits](#)

Student

Edit

EDIT

EDIT

EDIT

EDIT

1 of 4 < >

Navigation

 Students

Add Student

Roll No	Name	Branch	Room No	Mobile	Delete	Edit
1901212	Vaibhav Kesarwani	CSE	422	9695070039	<button>DELETE</button>	<button>EDIT</button>
1901126	Omang Rawat	CSE	351	8510025035	<button>DELETE</button>	<button>EDIT</button>
1901127	Test Student NEW	CSE	356	9414253689	<button>DELETE</button>	<button>EDIT</button>
1901012	Aditya Shara	CSE	451	9898098765	<button>DELETE</button>	<button>EDIT</button>

Rows per page: 10 ▾ 1–4 of 4 < >

[New Notice](#)

2022-03-27 12:40:12

Dear All,
hi there, free smoothies for all today

Posted By : Vishesh Sharma

[DELETE](#)

2022-03-25 12:40:12

Dear All,
It is to inform you that the GPU server (IP: 172.16.2.17) will be down from 12 pm IST to 2 pm IST today due to some urgent maintenance work.

Posted By : ICT Section, IIITG

[DELETE](#)

2022-03-25 12:40:12

Dear All,

2022-03-25 12:40:12

Dear All,

It is to inform you that the GPU server (IP: 172.16.2.17) will be down from 12 pm IST to 2 pm IST today due to some urgent maintenance work.

Posted By : ICT Section, IITG

New Notice**DELETE**

2022-03-25 12:40:12

Dear All,

Before leaving the hostel, make sure to deposit your room key at hostel security.

Posted By : Warden, Boys Hostel

DELETE

2022-03-20 12:40:12

Dear All,

React App

localhost:3000/dashboard

Gmail YouTube Maps

HMS

Home Dashboard Profile Logout

Register Form

Sent By Dr. Sumit Mishra message Submit your room keys before leaving hostel.

Register

Dear All

New Notice

This screenshot shows a web application interface for a Hostel Management System (HMS). At the top, there's a dark header with the title 'HMS'. Below it is a navigation bar with links for 'Home', 'Dashboard', 'Profile', and 'Logout'. A red 'Logout' button is highlighted. The main content area has a blue-to-purple gradient background and contains a 'Register Form'. On the left, under 'Sent By', it says 'Dr. Sumit Mishra'. To the right, there's a 'message' field with the placeholder 'Submit your room keys before leaving hostel.' Below this is a large, rounded rectangular button with the word 'Register' in purple. In the bottom left corner of the main area, there's a small snippet of text: 'Dear All'. To the right of the main form, there's a sidebar with a notice titled 'New Notice' and some other UI elements like trash cans.

Posted By : Warden, Boys Hostel

Home

Dashboard

Profile

Logout

2022-03-19 12:40:12

Dear All,

It is to inform you that due to some maintenance of the water pipeline, the water supply will be temporarily stopped from 9:00 AM - 5:00 PM on 15th-March-2022, especially in Boys Hostel. Therefore everyone is requested to store some water for tomorrow.

Posted By : Warden, Boys Hostel

DELETE

2022-04-17 19:21:56

Dear All,

Submit your room keys before leaving hostel.

Posted By : Dr. Sumit Mishra

DELETE

CARETAKER INTERFACE

- **Profile**
 - Caretaker's personal information is fetched from the API call to the "caretakers" table.
- **Guards**
 - Guard details can be added / edited / deleted by making respective POST request to the "guards" collection.
 - The documents from the "guards" collection are fetched and displayed in the table.
- **Cleaning Staff**
 - Cleaning Staff details can be added / edited / deleted by making respective POST request to the "guards" collection.
 - The documents from the "employees" collection are fetched and displayed on the dashboard.

Navigation

 Guard Staff Cleaning Staff

Add Guard

Job Id	Name	Mobile	Status	Delete	Edit
1	Guard1	123123121	Off Duty	<button>DELETE</button>	<button>EDIT</button>
1	Guard 2	9456896512	Off Duty	<button>DELETE</button>	<button>EDIT</button>

Rows per page: 10 < 1-2 of 2 >

Navigation

 Guard Staff Cleaning Staff

Add Employee

Job Id	Name	Mobile	Status	Delete	Edit
1	Employee 1	123123131	Off Duty	DELETE	EDIT
1	Employee 2	9585635245	Off Duty	DELETE	EDIT

Rows per page: 10 < 1-2 of 2 >

[New Notice](#)

2022-03-25 12:40:12

Dear All,
It is to inform you that the GPU server (IP: 172.16.2.17) will be down from 12 pm IST to 2 pm IST today due to some urgent maintenance work.

Posted By : ICT Section, IIITG

[DELETE](#)

2022-03-25 12:40:12

Dear All,
Before leaving the hostel, make sure to deposit your room key at hostel security.

Posted By : Warden, Boys Hostel

[DELETE](#)

2022-03-20 12:40:12

Dear All,

Validation

HASHING and SALTING

A “Hash” is a one-way function that generates a representation of the password. So when a user signs up for an account and they choose a password, the password is stored as the generated hash, rather than the actual characters that the user typed in.

Unfortunately, hashing a password is not nearly enough

In order to make it more difficult to expose a hash, you also need to salt it. Salting is where you add an extra bit of data to the password before you hash it.

Salting is important because it adds a whole new level of required computational power in order to expose the hash. By adding a salt, you effectively render any lookup table useless. Hashing a password is not 100% secure as hashing alone is not that difficult to break. When you add a salt to the hash, you make it much more difficult to crack.

notifies

requests

sports

students

users

wardens



```
_id: ObjectId("6259537b8aff7d6a4685c671")
phNo: 9414253689
email: "test@gmail.com"
password: "$2b$04$6CV.3W4aXbZr17zpWEYDH0ch7zyN7s6xFQqyQxUof44oIhvdy4iEm"
role: 1
__v: 0
```

logins



notifies

```
_id: ObjectId("6259537a8aff7d6a4685c66f")
name: "Test Student NEW"
phNo: 9414253689
rollNo: 1901127
address: "preet vihar, delhi"
password: "$2b$04$6CV.3W4aXbZr17zpWEYDH0ch7zyN7s6xFQqyQxUof44oIhvdy4iEm"
branch: "CSE"
passingYear: 2023
email: "test@gmail.com"
roomNo: 356
__v: 0
```

requests

sports

students

users

wardens

Code Snippet of hashing and salting

```
3 const Warden=require('../models/warden')
4 const bcrypt= require('bcrypt');
5
6 module.exports.addStudent=async (req,res)=>
7
8     try{
9         console.log('hello');
10        const { name,phNo,rollNo,address,password,branch,passingYear,email,roomNo } = req.body;
11
12        let salt_round=1;
13        let hashed_password= bcrypt.hashSync(password,salt_round);
14        const newStu = new Student({name: name,phNo: phNo,rollNo: rollNo,address:address,password:hashed_password,branch:branch,passingYear:passi
15        await newStu.save();
16        console.log(newStu);
17
18        const newUser = new User({phNo: phNo,email:email,password:hashed_password,role: 1 });
19        await newUser.save();
20
21        console.log(newUser);
22        res.status(200).json(newUser);
23    }
24    catch(e){
25        console.log(e);
26    }
27}
28}
```

ROLES AUTHORIZATION

controllers > JS users > dets > dets > func > bcrypt.compare() callback

```
20  const func = () =>{
21    bcrypt.compare(password, user.password, async function(err, isMatch) {
22      if (err) {
23        throw err
24      } else if (!isMatch) {
25        console.log("Password doesn't match!")
26        sample = {};
27        sample['message'] = '404';
28        res.status(200).json(sample)
29      } else {
30        console.log("Password matches!")
31        sample = {}
32        if(user.role==1){
33          var info= await Student.findOne({email:username});
34          console.log(info)
35          sample['id']= info._id
36          sample['role']= user.role
37          sample['message']= '500'
38          res.status(200).json(sample);
39        }
40        if(user.role==2){
41          var info= await Guard.findOne({email:username});
42          console.log(info);
43          sample['id']= info._id
44          sample['role']= user.role
45          sample['message']= '500'
46          res.status(200).json(sample);
47        }
48        if(user.role==3){
49          var info= await Employee.findOne({email:username});
50          // res.status(200).json(info);
51          sample['id']= info._id
52          sample['role']= user.role
53          sample['message']= '500'
```

controllers > JS users > dets > func > bcrypt.compare() callback

```
46  res.status(200).json(sample);
47  }
48  if(user.role==3){
49    var info= await Employee.findOne({email:username});
50    // res.status(200).json(info);
51    sample['id']= info._id
52    sample['role']= user.role
53    sample['message']= '500'
54    res.status(200).json(sample);
55  }
56  if(user.role==4){
57    console.log("inside")
58    var info= await Warden.findOne({email:username});
59    // res.status(200).json(info);
60    sample['id']= info._id
61    sample['role']= user.role
62    sample['message']= '500'
63    res.status(200).json(sample);
64  }
65  if(user.role==5){
66    console.log("caretaker")
67    var info= await Caretaker.findOne({email:username});
68    // res.status(200).json(info);
69    sample['id']= info._id
70    sample['role']= user.role
71    sample['message']= '500'
72    res.status(200).json(sample);
73  }
74  }
75  }
76  }
77  }
78  }
79  }
80  }

func()
```

MVC Pattern

MVC Pattern stands for Model-View-Controller Pattern. This pattern is used to separate application's concerns.

Model

Represents an object carrying data. It can also have logic to update controller if its data changes

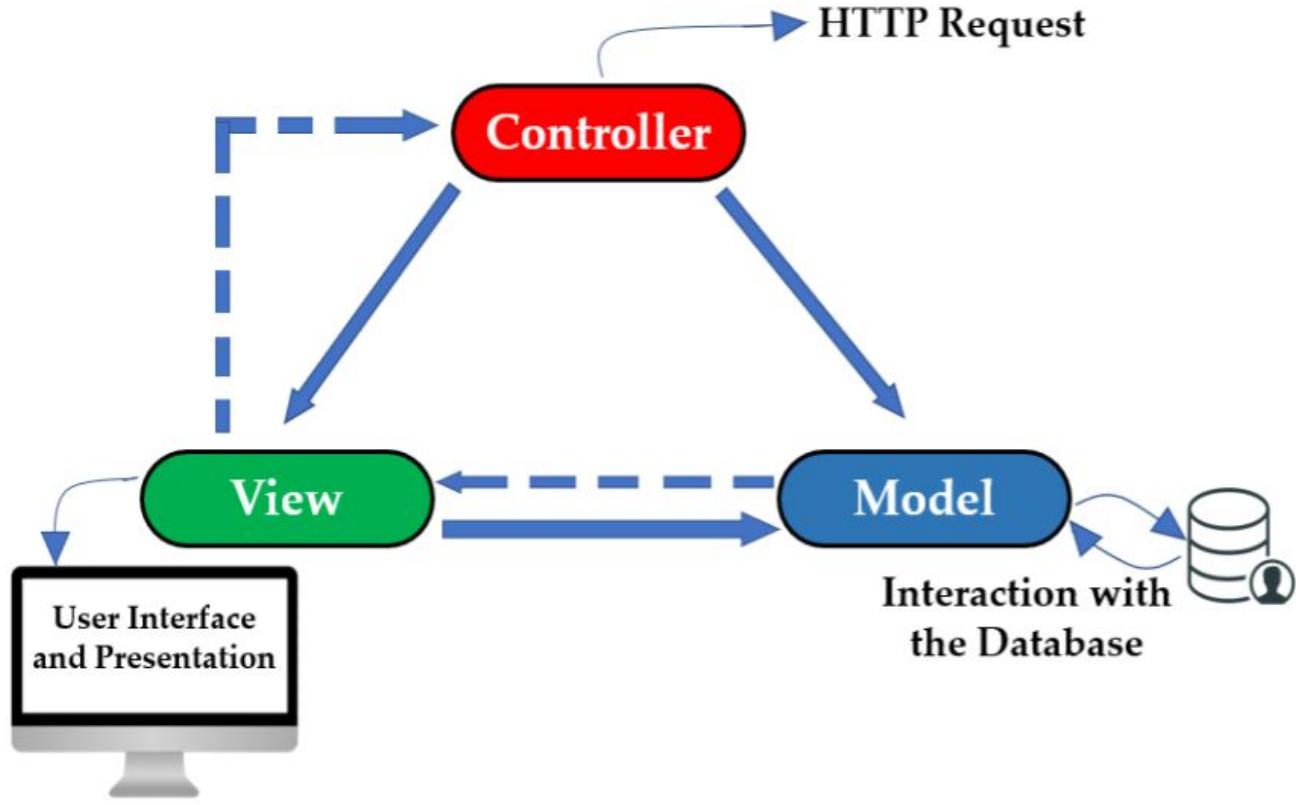
Controller

Acts on both model and view. It controls the data flow into model object and updates the view whenever data changes.
It keeps view and model separate.

View

Represents the visualization of the data that model contains.

Model-View-Controller Architecture Block Diagram



CRUD IMPLEMENTATION

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/dashboard". The page has a dark blue header with the text "HMS" and navigation links for "Home", "Dashboard", "Profile", and "Logout". Below the header, there is a list of three news items, each enclosed in a white box with a blue border. Each item includes a timestamp, the content of the notice, and the name of the poster.

- 2022-03-27 12:40:12
Dear All,
hi there, free smoothies for all today
Posted By : Vishesh Sharma
- 2022-03-25 12:40:12
Dear All,
It is to inform you that the GPU server (IP: 172.16.2.17) will be down from 12 pm IST to 2 pm IST today due to some urgent maintenance work.
Posted By : ICT Section, IIITG
- 2022-03-25 12:40:12
Dear All,

Student, Guard and Cleaning Staff can Read the updates on dashboard.

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/dashboard". The page has a dark blue header with the text "HMS" and navigation links for "Home", "Dashboard", "Profile", and "Logout". Below the header, there is a list of four news items, each enclosed in a white box with a blue border. The first three items are identical to those in the left screenshot. A fourth item is added at the bottom. On the right side of the screen, there is a vertical sidebar with a "New Notice" button at the top and a "DELETE" button at the bottom.

- 2022-03-25 12:40:12
Dear All,
It is to inform you that the GPU server (IP: 172.16.2.17) will be down from 12 pm IST to 2 pm IST today due to some urgent maintenance work.
Posted By : ICT Section, IIITG
- 2022-03-25 12:40:12
Dear All,
Before leaving the hostel, make sure to deposit your room key at hostel security.
Posted By : Warden, Boys Hostel
- 2022-03-20 12:40:12
Dear All,

Warden and Caretaker can Create, Read or Delete updates on dashboard.

SCHEMAS

A schema is a JSON object that defines the the structure and contents of your data.

Schemas represent *types* of data rather than specific values. There are many built-in schema types. These include primitives, like strings and numbers, as well as structural types, like objects and arrays, which you can combine to create schemas that represent custom *object types*.

In our project we have used 11 schemas. Namely: user, student, warden, caretaker, guard, employee, delivery, Sports and notify, request, dashbo

Some of them are shown below

```
const mongoose = require('mongoose');
const {Schema} =mongoose;

const studentSchema= new Schema({
    name : String,
    phNo : Number,
    rollNo : Number,
    address: String,
    password: String,
    branch: String,
    passingYear: Number,
    email: {
        type: String,
        required: true,
        unique:true
    },
    roomNo:Number
});

const Student= mongoose.model('Student',studentSchema);

module.exports=Student;
```

```
const mongoose =require('mongoose');
const {Schema} =mongoose;

const wardenSchema= new Schema({
    name : String,
    phNo : Number,
    jobId : Number,
    address: String,
    password: String,
    onDuty: Boolean,
    email: {
        type: String,
        required: true,
        unique:true
    }
});

var Warden= mongoose.model('Warden',wardenSchema);
module.exports=Warden;
```

CONTRIBUTION

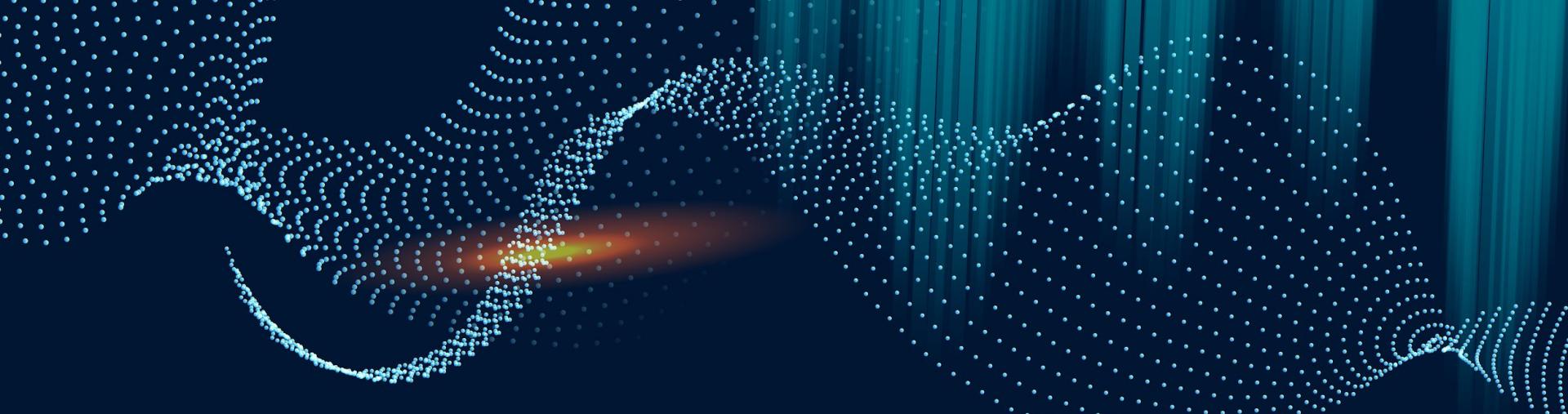
- **Omang Rawat (1901126)**
 - Implemented Login and Dashboard components.
 - Created Student and Caretaker Interface
 - Handled function calls (REST API)
 - Managed dataflow between different components.

- **Rahul Soni (1901154)**
 - Created Guard, Cleaning Service, Warden Interface
 - Adding CRUD implementation. (Frontend)
 - User Authentication and Authorization (Frontend)
 - Hashing and Salting

CONTRIBUTION

- **Vaibhav Kesarwani (1901212)**
 - Implemented Profile Interface
 - MongoDB Setup , Populating and managing data
 - Unit Testing for backend (Using Mocha and chai)
 - Integration Testing and System testing (Black box testing)
 - Finalizing Testing Report

- **Vishesh Sharma (1901217)**
 - Created data models and schemas
 - User Authentication and Authorization (Backend)
 - Created API Routes and Controllers.
 - Adding CRUD implementation. (Backend)



THANK YOU!