# Web Technology

Faculty - Vijay Ukani

November 8, 2019

# Hostel Management System

Aman Yadav (201615007)

Avdhesh Yadav (201651010)

Mayank Pathela (201651029)

Anshuman Verma (201651060)

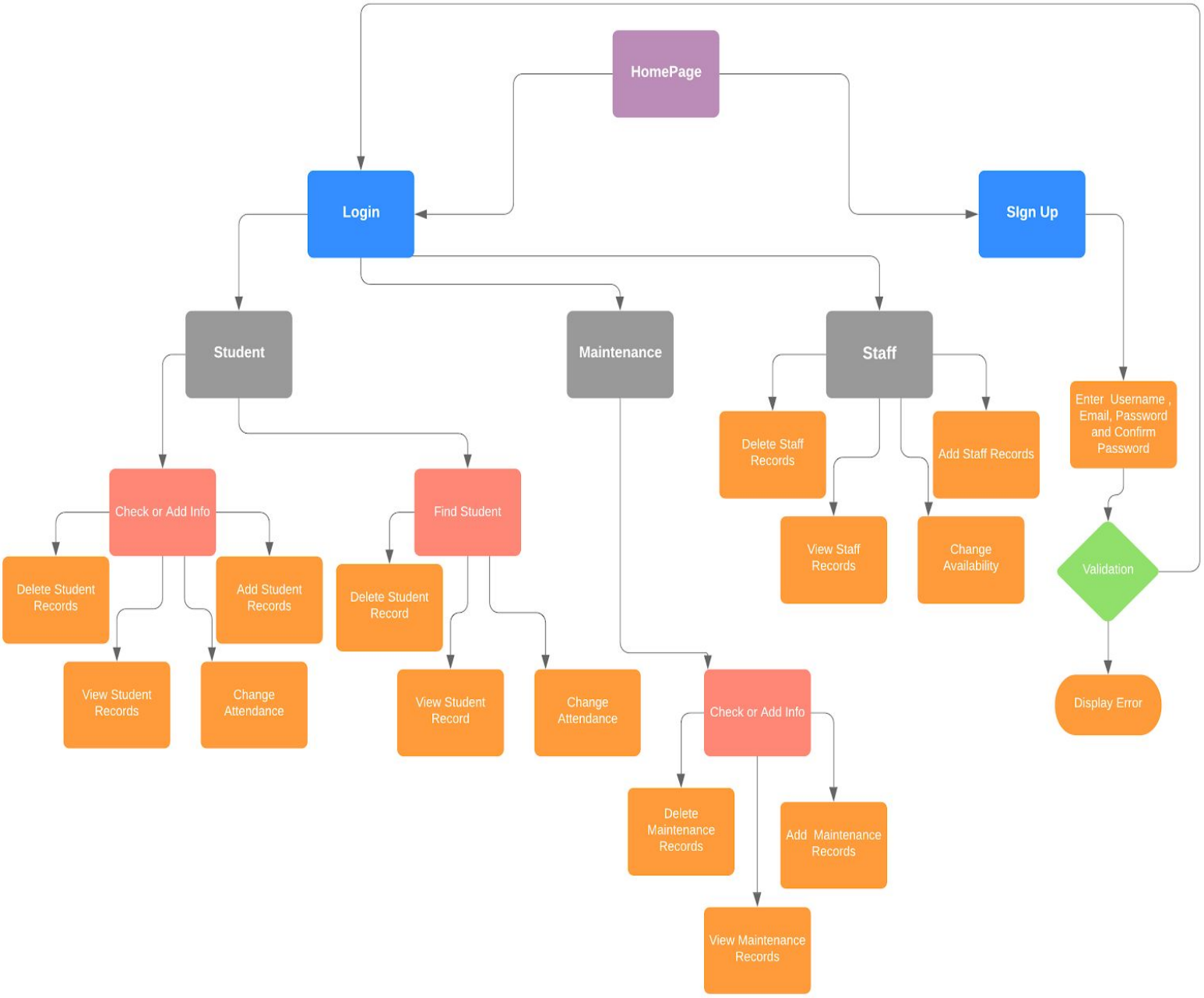
# Overview

**Hostel-Management** is a web application for managing the records of day to day activity at a Hostel. Information regarding students, repairs required and staff is stored systemically. The application lets you maintain and search all these records seamlessly. The current industry choice:- MERN (MongoDB, Express.js, React and Node.js) Stack has been used for development.

The Web Application is divided into four sections as follows:-

1) **Student information:** Under the Student Information Section, you can -
    a) View Records of each student batch-wise.
    b) Add new Student Records.
    c) Delete Student Records.
    d) View and update the attendance status (Absent/Present) of a particular student.
    e) Search Records of a Student with three options for keys:
        i) Student ID
        ii) Room Number
        iii) Attendance Status
2) **Staff information:** Under the Staff Information section, you can -
    a) View Records of all the hostel staff.
    b) Add new Employee Records.
    c) Delete Employee Records.
    d) View and update the availability status (Available/Unavailable) of a particular employee.
3) **Room maintenance records**
    a) View all Maintenance (cleaning/repair) Records block-wise.
    b) Add a new Maintenance Record.
    c) Delete Maintenance Records.
4) **Login/Authentication**
    a) Email, a username, and a password are required to sign up.
    b) All future authentications are done through the email address and password provided at the time of sign up.

# Flowchart


HomePage
Login
Sign Up
Student
Maintenance
Staff
Check or Add Info
Find Student
Delete Staff Records
Add Staff Records
Enter Username , Email, Password and Confirm Password
Delete Student Records
Add Student Records
Delete Student Record
View Staff Records
Change Availability
Validation
View Student Records
Change Attendance
View Student Record
Change Attendance
Check or Add Info
Display Error
Delete Maintenance Records
Add Maintenance Records
View Maintenance Records

# Technical Overview

## Backend

Node.js is used as a server-side technology. Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine).

Here is how Node.js handles a file request:

- Sends the task.
- Ready to handle the next request.
- When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting and simply continues with the next request.

Node.js runs single-threaded, non-blocking, asynchronously, which is very memory efficient.

Creating a basic server using Node.js:

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

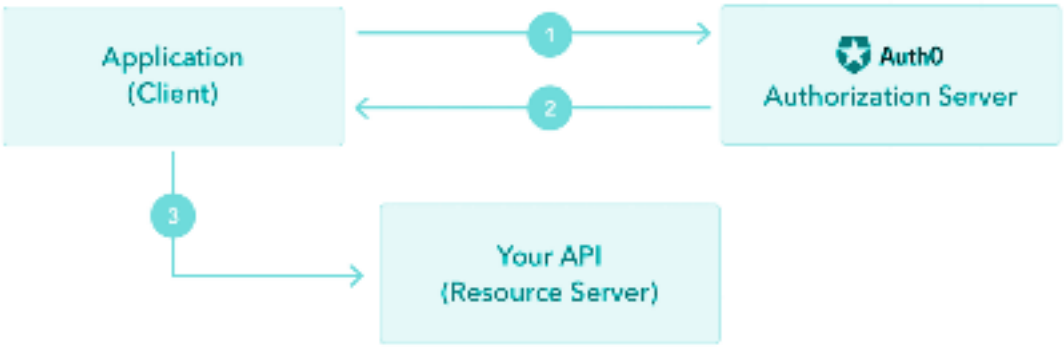
Although Node.js is capable of creating the server and handling requests on itself, Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. provides a thin layer of fundamental web application features. a myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and easy. With a myriad of HTTP utility methods and middleware at disposal, it helps to create a robust API very quickly and easily

The back-end of the Hostel Management System is built using RESTful API architecture using Express with Node.js.

## Authentication

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the **HMAC** algorithm) or a public/private key pair using **RSA** or **ECDSA**.

In authentication, when the user successfully logs in using their credentials, a JSON Web Token will be returned. Whenever the user wants to access a protected route or resource, the user agent should send the JWT. The server's protected routes will check for a valid JWT in the Authorization header, and if it's present, the user will be allowed to access protected resources.




## Database

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with the schema.

MongoDB scales horizontally using sharding.  The user chooses a shard key, which determines how the data in a collection will be distributed. The data is split into ranges (based on the shard key) and distributed across multiple shards. (A shard is a master with one or more replicas.). Alternatively, the shared key can be hashed to map to a shard – enabling an even data distribution. MongoDB can run over multiple servers, balancing the load or duplicating data to keep the system up and running in case of hardware failure.

A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB supports a rich query language to support read and write operations (CRUD).

# Frontend

- UI

React is used to build the User Interface of the Hostel Management System. React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep the state out of the DOM.

A typical component in React:

```
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Hello {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage name="Taylor" />,
  document.getElementById('hello-example')
);
```

- State Management

Redux is used as a state management library for Hostel Management System

Redux is an open-source JavaScript library for managing application state. Redux helps us write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. Redux helps in using the same data at multiple places without having duplicates by maintaining an application-level state from where any component can get the desired data.

When the application level state is updated by redux, the data in the front-end automatically updates as the components are using the data from redux. This helps to keep data persistent throughout the app.

# Testing

Jest is a JavaScript testing framework designed to ensure the correctness of any JavaScript codebase.

Tests are designed to check if the correct data is fetched and correct data is added to the database. Along with checking the correctness, the incorrect data is not added to the database is also tested with authenticated routes included.

# Installation Guide

### Requirements

- Node.js and npm
  - For Windows and macOS - https://nodejs.org/en/
  - For Linux - Follow this guide - https://www.ostechnix.com/install-node-js-linux/
- MongoDB - can be downloaded from https://www.mongodb.com/download-center/community
- Bootstrap - can be downloaded from https://getbootstrap.com/
- Express - can be downloaded from https://expressjs.com/

### Project Setup

- Open the terminal.
- Go to the directory of the project.
- To install backend dependencies, type npm install in the terminal.
- To install frontend dependencies, type npm run client-install in the terminal.
- Now, all the dependencies will be installed in the project folder.

### Executing the project

- Before running the project, run the MongoDB server.
- After running the MongoDB server, open a new terminal and navigate into the directory where the project is located.
- Type npm run dev in the terminal to run the project.
- This will start the project locally in development mode.

### Project Deployment

Heroku is a cloud platform based on a managed container system, with integrated data services and a powerful ecosystem, for deploying and running modern apps.

The project is deployed on Heroku and can be viewed on https://hostel-management01.herokuapp.com/

## Assignment-1 (Information Flow):

### Backend

- **Runtime** - Node.js
- **Framework** - Express
- **Authentication** - JWT

### Database

- MongoDB

### Frontend

- **React** - Frontend Library
- **Redux** - global state management
- **Bootstrap** - UI components
- **Axios** - Promise based HTTP client for the browser and node.js
    - Make XMLHttpRequests from the browser
    - Make Http requests from node.js
    - Supports the Promise API
    - Intercept request and response
    - Transform request and response data
    - Cancel requests
    - Automatic transforms for JSON data
    - Client-side support for protecting against XSRF

## Assignment-2 (Validation Structure):

- **Code Collaboration** - GitHub
- VCS - git
- Backend validations using the **Validator** package.
    - A library of string validators and sanitizers.

      A simple validation look like -


- Frontend validation using native HTML5 tags.
- **Unit Testing** - Jest

## Assignment-3 (Session Management):

- **Passport** is used to manage sessions.

Passport is authentication middleware for Node.js. Extremely flexible and modular, Passport can be unobtrusively dropped into any Express-based web application. A comprehensive set of strategies support authentication using a username and password, Facebook, Twitter, and more.

A basic authentication looks like:

```
app.post('/login',
  passport.authenticate('local'),
  function(req, res) {
    // If this function gets called, authentication was successful.
    // `req.user` contains the authenticated user.
    res.redirect('/users/' + req.user.username);
  });
```

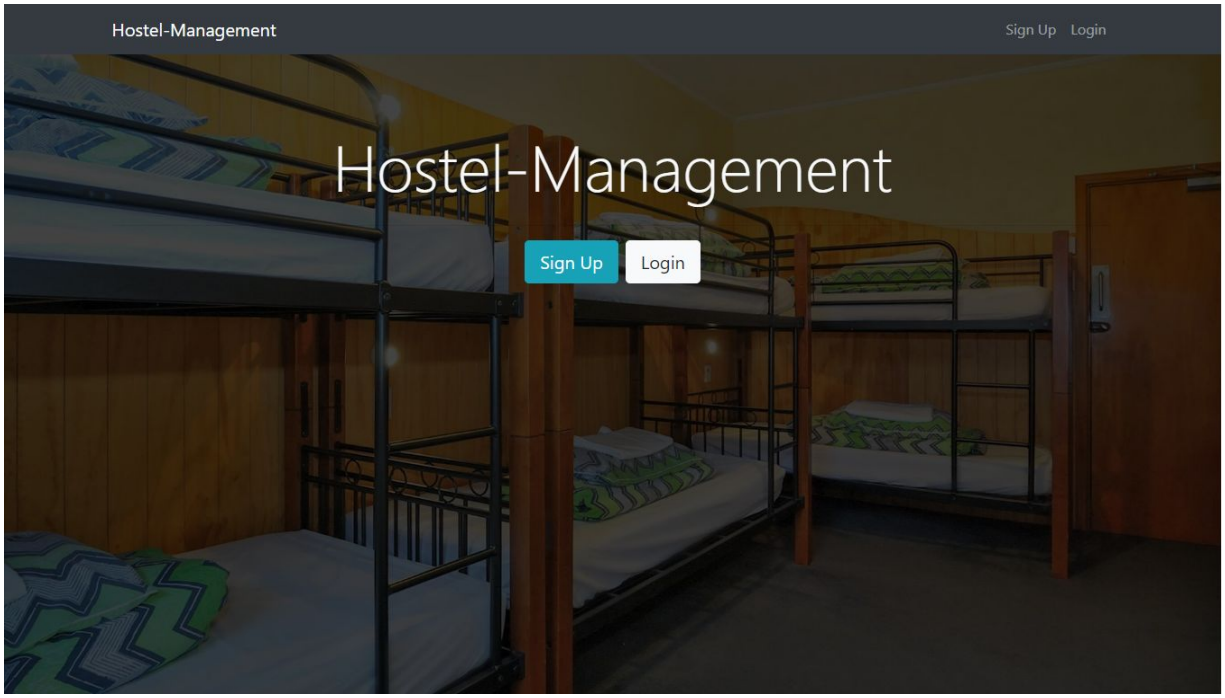
- Bootstrap is an open-source toolkit for developing with HTML, CSS, and JS.

Bootstrap is used as the frontend framework to create UI components and set up a grid system.

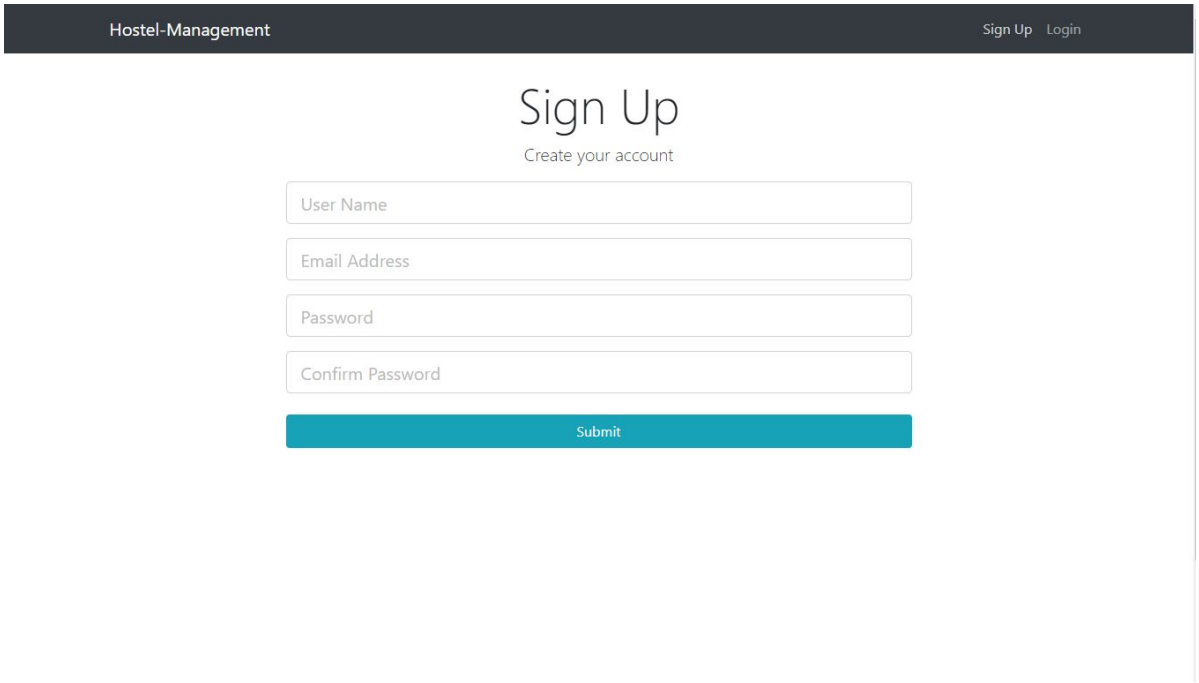
- JWT token is saved on the local storage of the user.

## Styling and Layout

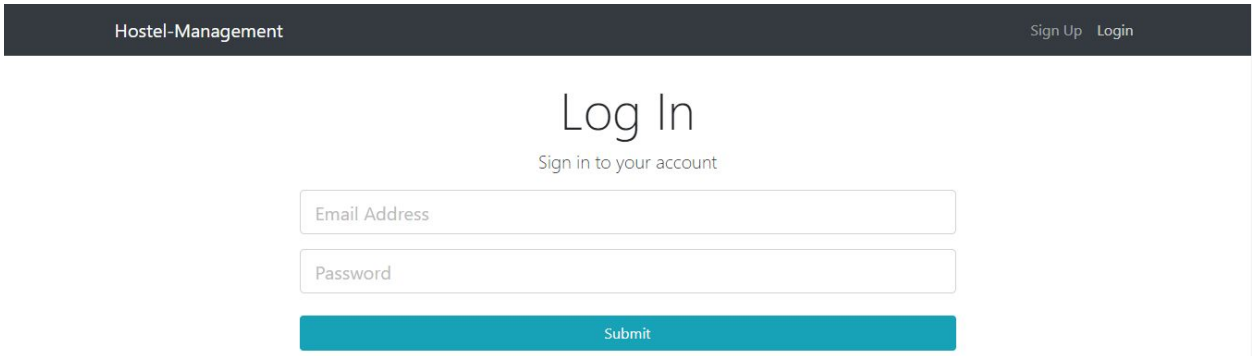
### Home Screen



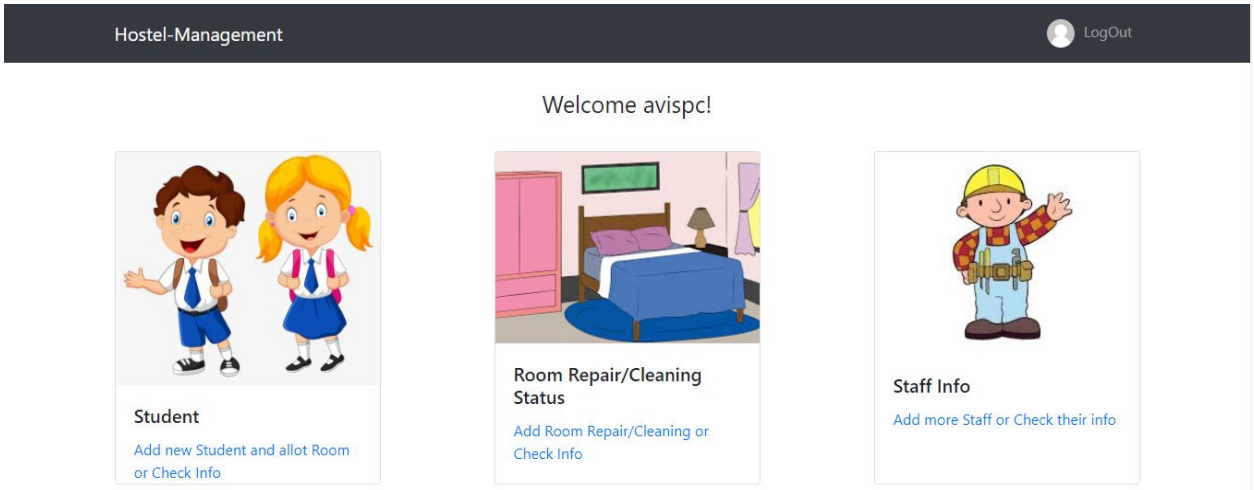

## SignUp



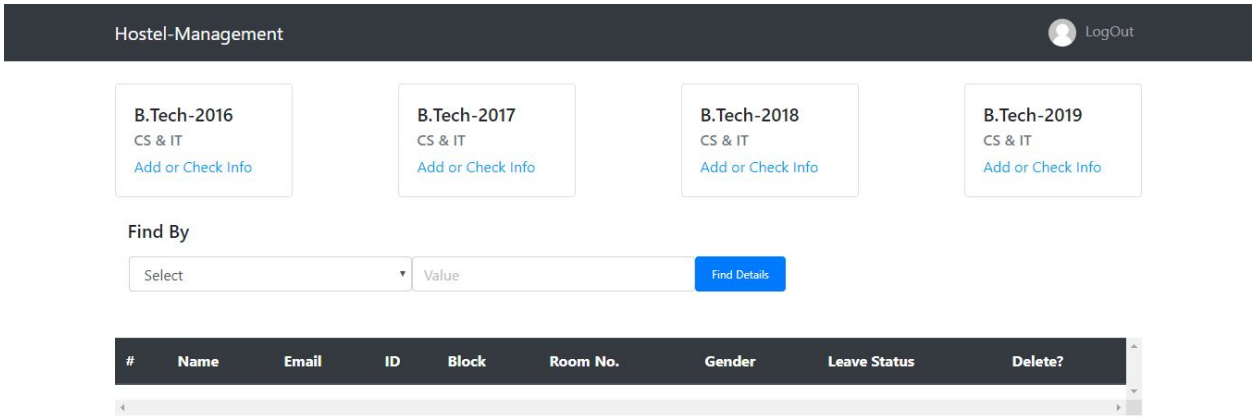
## Login

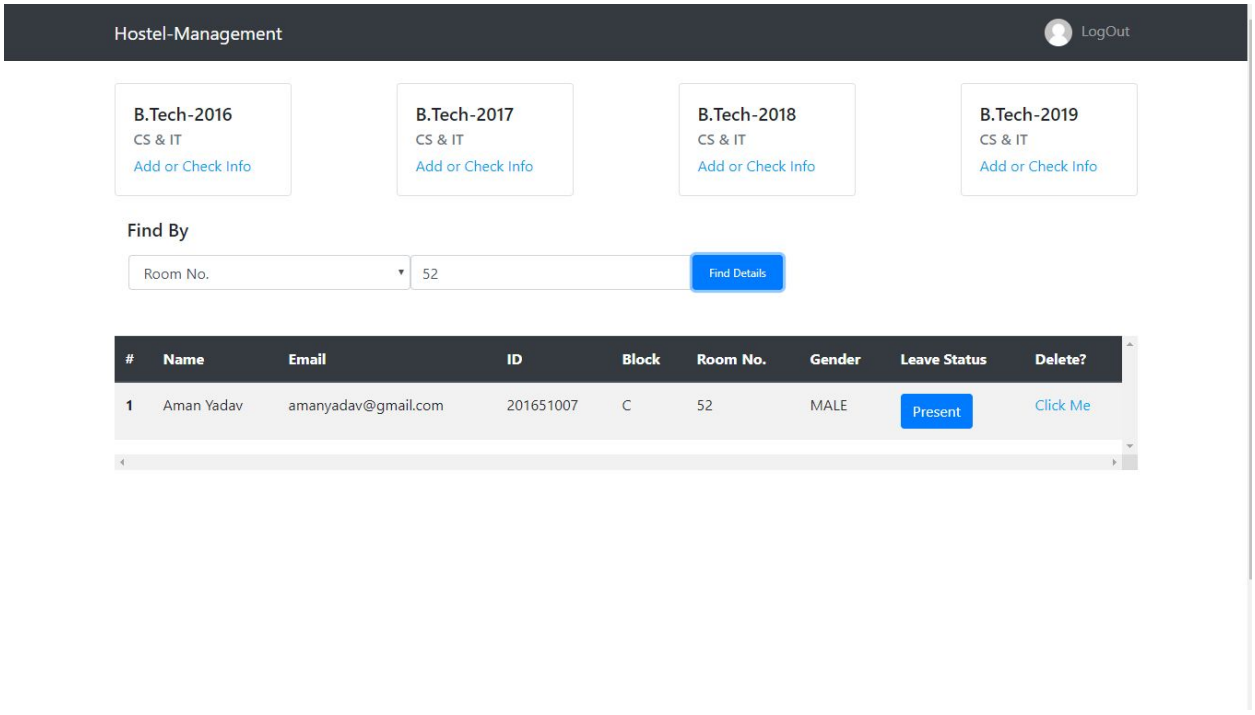


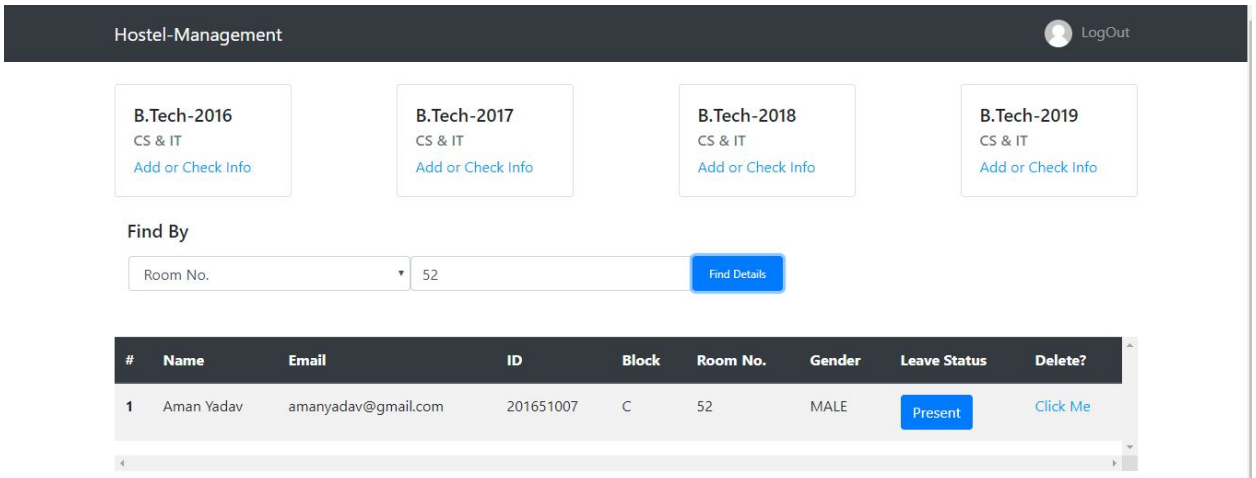

## Main Page



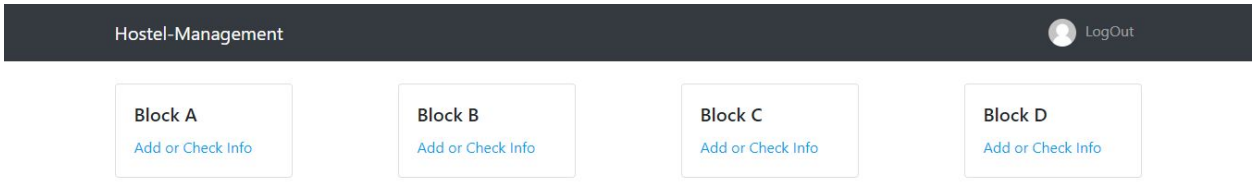
## The Page for adding a new student or finding using a filter



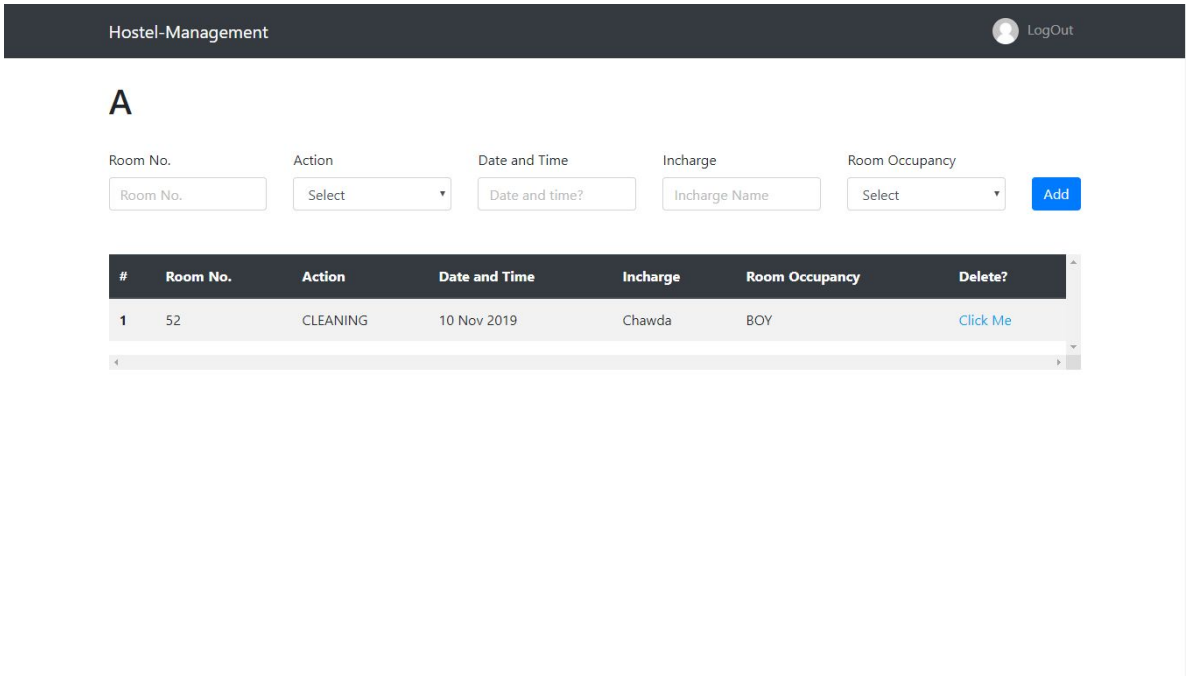





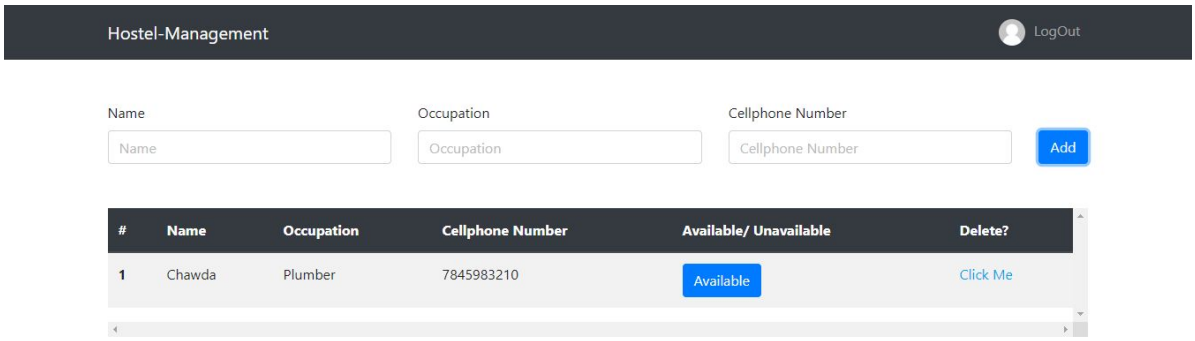

The page for adding a new staff member



## Staff and their duty details page




## The Page for adding new staff members and viewing their details



# Future Enhancements

- Add the day-to-day attendance of students.
- Time and room-based filter and search for cleaning/repair activities.
- Students Complaint Record.