A Project report

on

HOSTEL MANAGEMENT SYSTEM

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

In

Computer Science and Engineering

Submitted by

K Pavan	19H51A0544
P Sai Prasad	19H51A0554
G Vamshi Reddy	19H51A05A1

Under the esteemed guidance of
Mrs. S. Swetha
Assistant Professor CSE Department

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution under UGC & JNTUH, Approved by AICTE, Permanently Affiliated to JNTUH, Accredited by NBA.)
KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.
2022- 2023

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Mini Project-2 report entitled " **HOSTEL MANAGEMENT SYSTEM**" being submitted by **K.PAVAN(19H51A0544), P SAI PRASAD(19H51A0554), G.VAMSHI REDDY(19H51A05A1)** in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mrs .S Swetha
Assistant Professor
Dept of CSE

Dr. S Siva Skandha
Associate Professor and HOD
Dept of CSE

Submitted for viva voice Examination held on _____

External Examiner

Acknowledgment

With great pleasure I want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mrs. S Swetha** (Associate Professor), Dept of Computer Science and Engineering for his valuable suggestions and guidance during the execution of this project work.

We would like to thank **Dr. S Siva Skandha**, Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Vijaya Kumar Koppula**, Dean-Academics, CMR College of Engineering and technology, for his constant support and motivation in carrying out the project work successfully.

I am highly indebted to **Dr. V A Narayana**, Principal CMRCET for giving permission to carry out this project in a successful and fruitful way.

I would like to thank the Teaching & Non- teaching staff of Department of Computer Science and Engineering for their co-operation.

Finally, I express my sincere thanks to **Mr. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care. I sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

SIGNATURE

K PAVAN	19H51A0544
P SAI PRASAD	19H51A0554
G VAMSHI REDDY	19H51A05A1

DECLARATION

We hereby declare that results embodied in this Report of Projection **"HOSTEL MANAGEMENT SYSTEM"** are from the work carried out by using partial fulfillment of the requirements for the award of B. Tech degree. We have not submitted this report to any other university/institute for the award of any other degree.

SIGNATURE:

K PAVAN 19H51A0544

P SAI PRASAD 19H51A0554

G VAMSHI REDDY 19H51A05A1

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	LIST OF FIGURES	**4**
	ABSTRACT	**5**
1	**INTRODUCTION**	**6-8**
	1.1 Problem Definition	
	1.2 User Panel	
	1.2.1 Home Page	
	1.2.2 Registration Page	
	1.2.3 Login page	
	1.3 Admin Panel	
	1.3.1 Home Page	
	1.3.2 Allotment of rooms	
	1.3.3 Vacating the rooms	
2	**BACKGROUND WORK**	**9-10**
	2.1 Existing Solutions	
	2.2 Demerits in existing Solutions	
3	**PROPOSED SYSTEM**	**11-15**
	3.1 Proposed model	
	3.2 Software Requirements	
	3.3 Software Features	
	3.3.1 Html	
	3.3.2 CSS	
	3.3.3 React js	
	3.3.4 Node js	
	3.3.5 Express js	
	3.3.6 Mongo DB	
4	**DESIGNING**	**16-26**
	4. 1 Testing	
	4.1.1 Unit Testing	
	4.1.2 Integration Testing	
	4.1.3 User Acceptance	
	4.1.4 Test Cases	
	4.2 Use Case Diagram	
	4.3 Relational Diagram	
	4.4 Source code	
	4.4.1 Back End	
	4.4.2 Front End	
5	**IMPLEMENTATION**	**27-28**
	5.1 Security and Maintenance	
6	**RESULTS AND DISCUSSION**	**29-33**
7	**CONCLUSON AND FUTURE WORK**	**34-35**
8	**REFERENCES**	**36**

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
4.2	USE CASE DIAGRAM	18
4.3	RELATIONAL DIAGRAM	18
6.1	HOME PAGE	30
6.2	REGISTER PAGE	30
6.3	LOGIN PAGE	31
6.4	UPDATE DETAILS PAGE	31
6.5	RENEWAL PAGE	32
6.6	ROOM DETAILS PAGE	32
6.7	EDIT-USERS PAGE	33
6.8	EDIT-ROOMS PAGE	33

ABSTRACT

As the name specifies "**HOSTEL MANAGEMENT SYSTEM**" is software developed for managing various activities in the hostel. For the number of educational institutions are increasing rapidly .There by the number of hostels are also increasing for the accom modation of the students in this institution. And hence is a lot of strain on the person who are running the hostel and software's are not usually in this context. This project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.

Identification of drawbacks of the existing leads to the designing of computer system that will be compatible to the existing system with the system which is more user friendly and GUI oriented. We can improve the efficiency of the system and thus overcome the draw backs of the existing system.

CHAPTER 1
INTRODUCTION

INTRODUCTION

1.1 PROBLEM DEFINATION:

All the hostels present in the institution are managed manually by the hostel office. The Registration form verification to the different data processing are done manually.

Thus there are a lot of repetitions which can be easily avoided. And hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context.

This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GU oriented. We can improve the efficiency of the system.

1.2 User Panel:

1.2.1 Home Page:

- It contains three popup login, my profile and register.
- This page can access by both user and the admin.
- He can view the Student administration division of the different hostel.

1.2.2 Registration Page:

- It provides application form to the user which should be filled by the user.
- In this, user must fill the details provide in the form.
- It also contains login details.

1.2.3 Login Page:

- In this page user must enter the login details provided in register page.
- After login user can see the full details of the room provided by the admin.
- In this the room details will be updated time to time.

1.3 Admin Panel:

1.3.1 Home Page:

The Administrator can:

1. Allot different students to the different hostels.
2. Vacate the students for the hostels.
3. Control the status of the fee payment.
4. Edit the details of the students & modify the student records.

1.3.2 Allotment of Rooms:

Based on the applications attested from the students, the room is allotted by the admin based on the database of the user. Once the room is allotted the details of the room will be updated automatically. If the rooms are already filled then the application of the user is denied.

1.3.3 Vacating the Rooms:

As the student course is over they will vacate their rooms. So it is required to remove the records of the user by the admin and should update the room details in the home page. So the room can be registered by another user and the details of the user can be removed by the admin from the databases.

CHAPTER 2
BACKGROUND WORK

2. BACKGROUND WORK

2.1 Existing Solution:

For the number of educational institutions are increasing rapidly .There by the number of hostels are also increasing for the accommodation of the students in this institution. And hence is a lot of strain on the person who are running extra the hostel and software's are not usually in this context. This project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.

Hostel facilities in educational institutions are managed with manual paperwork in a labor-intensive environment. This has resulted in poor resource utilization and impacted the overall efficiency of the academic institutions. The system of managing and administering hostels, mess and other facilities in institutions is ineffective due to labor work. It takes more time to manage and more effort while the strength is huge. It requires more labor to work moreover it may be ineffective.

2.2 Demerits in existing Solutions:

- Manual hostel administration
- Manual allocation of facilities
- Wastage of resources
- Increase in staff workload
- Difficulty in retrieving hostel record of assets & inventory
- Tedious and time consuming work
- Inaccuracy of data
- Different data processing for registration form verification
- Data duplication & errors

CHAPTER 3
PROPOSED SYSTEM

3.1 Proposed model:

Hostel facilities which are managed manually can use this application for the data management of hostel students. Not only admin but also user can use the application for the booking of hostel room and can also see the facilities provided by the hostel. Hostel fee can also be paid by the user which is provided in this application. This model is designed by considering all the demerits of existing model. This model is designed by using html, css, bootstrap and React js. Whereas Node js and express js is used a back end.

3.2 Software Requirements:

Operating Systems : Windows 10 and Above

Technology : Web browser (chrome, fire fox)

Front end : Html, css, bootstrap and React js

Back end : Node js and express js

Database : mongoDB Atlas

3.3 Software Features:

3.3.1 About Html:

HTML (Hyper Text Markup Language) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables. It defines the structure of your content. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way. The enclosing tags can make a word or image hyperlink to somewhere else, can italicize words, can make the font bigger or smaller, and so on.

The main parts of our element are as follows:

The opening tag: This consists of the name of the element wrapped in opening and closing angle brackets. This states where the element begins or starts to take effect in this case where the paragraph begins.

The closing tag: This is the same as the opening tag, except that it includes a before the element name. This states where the element ends — in this case where the paragraph ends. Failing to add a closing tag is one of the standard beginner errors and can lead to strange results.

The content: This is the content of the element, which in this case, is just text.

The element: The opening tag, the closing tag, and the content together comprise the element.

3.3.2 About CSS:

A website can run without CSS, but it certainly isn't pretty. CSS makes the front-end of a website shine and it creates a great user experience. Without CSS, websites would be less pleasing to the eye and likely much harder to navigate. In addition to layout and format, CSS is responsible for font color and more.
There are a number of benefits of CSS, including:

Faster Page Speed:
More code means slower page speed. And CSS enables you to use less code. CSS allows you to use one CSS rule and apply it to all occurrences of a certain tag within an HTML document.

Better User Experience:
CSS not only makes web pages easy on the eye, it also allows for user-friendly formatting. When buttons and text are in logical places and well organized, user experience improves.

Quicker Development Time:
With CSS, you can apply specific formatting rules and styles to multiple pages with one string of code. One cascading style sheet can be replicated across several website pages. If, for instance, you have product pages that should all have the same formatting, look, and feel, writing CSS rules for one page will suffice for all pages of that same type.

Easy Formatting Changes:
If you need to change the format of a specific set of pages, it's easy to do so with CSS. There's no need to fix every individual page. Just edit the corresponding CSS style sheet and you'll see changes applied to all the pages that are using that style sheet.

Compatibility Across Devices:
Responsive web design matters. In today's day and age, web pages must be fully visible and easily navigable on all devices. Whether mobile or tablet, desktop, or even smart TV, CSS combines with HTML to make responsive design

3.3.3 About React js:

React, sometimes referred to as a frontend JavaScript framework, is a JavaScript library created by Face book. React is a tool for building UI components. Current version of React.JS is V18.0.0 (April 2022).Initial Release to the Public (V0.3.0) was in July 2013.React.JS was first used in 2011 for Face book's Newsfeed feature.Facebook Software Engineer, Jordan Walke, created it. Current version of create-react-app is v5.0.1 (April 2022).create-react-app includes built tools such as web pack, Babel, and ESLint.

ADVANTAGES OF REACTJS:

Intuitive:
React JS is extremely intuitive to work with and provides interactivity to the layout of any UI. Plus, it enables fast and quality assured application development that in turn saves tome for both - clients and developers.

Declarative:
React JS enables significant data changes that result in automatic alteration in the selected parts of user interfaces. Owing to this progressive functionality, there is no additional function that you need to perform to update your user interface.

Provides Reusable Components:

React JS provides reusable components that developers have the authority to reuse and create a new application. Reusability is exactly like a remedy for developers. This platform gives the developers the authority to reuse the components build for some other application having the same functionality. Thereby, reducing the development effort and ensuring a flawless performance.

3.3.4 About Node js:

The popularity of Node.js, the JavaScript runtime environment, has been steadily growing since its first release in 2009, and it seems that this trend is here to stay. For years, many well-established brands, like PayPal or Yahoo, have favored Node.js for the massive amount of advantages it offers. Node.js seems to be the most obvious choice for startupers, and more and more companies are giving it a try.

- Efficient performance
- Easier development process
- Reusable code
- Ability to handle multiple requests
- Ability to scale smoothly
- Prompt code execution
- Asynchronous and event-driven
- Supported by leading companies
- Top-notch prototyping capabilities
- Vast talent pool

3.3.5 About Express js:

Express JS is a Node.js framework designed to build API's web applications cross-platform mobile apps quickly and make node js easy. Express is a node js web application framework that provides broad features for building web and mobile applications. It is used to build a single page, multipage, and hybrid web application. It's a layer built on the top of the Node js that helps manage servers and routes**.**

Advantages of Using Express With Node.js:

- Express is Un opinionated, and we can customize it.
- For request handling, we can use Middleware.
- A single language is used for frontend and backend development.
- Express is fast to link it with databases like MySQL, MongoDB, etc.
- Express allows dynamic rendering of HTML Pages based on passing arguments to template

3.3.6 About MongoDB:

MongoDB was released on August 27th, 2009. Version 1 was basic, while version 2 introduced features like sharding, usable and special indices, geospatial features, memory, and concurrency improvements, among others. Version 3 came with an aggregation framework mainly as a supplement to the aging MapReduce framework.

MongoDB was created in 2009 as an open-source, highly scalable, robust, and free NoSQL database. It also has a commercial version. You can find MongoDB's source code on GitHub.

MongoDB has cultivated a reputation as a versatile, flexible database and is currently used today as the backend data store of many high-profile businesses and organizations such as Forbes, Facebook, Google, IBM, Twitter, and many more.

MongoDB is a non-relational database system. There are two primary database types: SQL (relational) and NoSQL (non-relational). Relational databases store data in columns and rows. Organizations like Microsoft SQL Server Oracle and Sybase use the relational database management system (RDBMS).

On the other hand, NoSQL databases store schema-less, unstructured data in multiple collections and nodes. Non-relational databases don't need fixed table sachems. NoSQL databases are scaled horizontally and support limited join queries.

Incidentally, NoSQL is short for "Not Only SQL."

CHAPTER-4

DESIGNING

4.1 Testing:

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is the process of executing the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance.

4.1.1. Unit Testing:

The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within each module. This testing includes entering data and ascertaining if the value matches to the type and size supported by java. The various controls are tested to ensure that each performs its action as required.

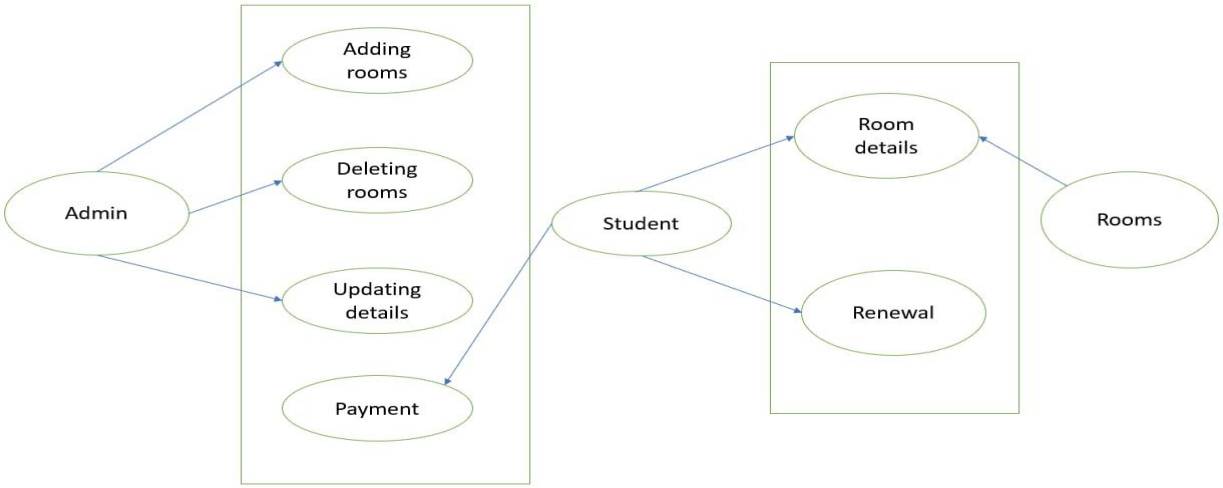
4.1.2. Integration Testing:

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here the Server module and Client module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

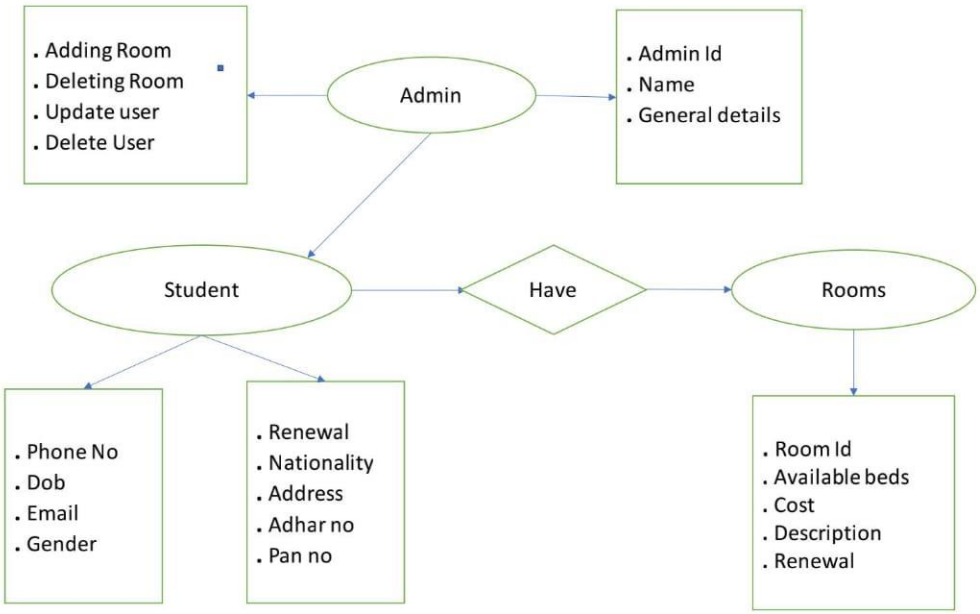
4.1.3 User Acceptance:

Testing User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the system users at time of developing and making changes whenever required.

4.2 Use Case Diagram:



4.3 Relational Diagram:



4.4 Source Code:

4.4.1 Back end:

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

const path=require('path');

const app=express();

const Router=express.Router();

const port=8000;

const validator=require('validator');

 const register= require(path.join(__dirname,"Db/register.js"));

 const rooms=require(path.join(__dirname,"Db/Rooms.js"));

// establishing Database Connection

const mongoose=require('mongoose');

const
ok='mongodb+srv://saiprasad:saiprasad@cluster0.huuoxxb.mongodb.net/mernstack?retryWrites=true&w
=majority';

mongoose.connect(ok).then(()=>{

    console.log("connection Done")

}).catch((error)=>{

    console.log("connection failed")

});

app.use(express.json());

app.use(express.urlencoded({extended:false}));

app.set('view engine', 'hbs')

app.get('/',(req,res)=>{

    res.render("index");
```

```
})

// User registration

app.post('/register', (req,res)=>{

    console.log(req.body);

    const aadharNumber=req.body.aadarCard;

    const panCard=req.body.panCard;

    const email=req.body.email;

    const password=req.body.password;

    const name=req.body.name;

    const mobile=req.body.mobile;

    const AmountPaid=0;

    const BookedRoomNo=0;

    const Active=true;   // will be set to True after fee payment and checkIn

    const TimePeriod=0;

    const checkInDate=new Date();

    const adminApproval=false;  // Admin have to approve Users account


  const reg=new
register({name,email,mobile,aadharNumber,panCard,password,AmountPaid,TimePeriod,BookedRoomNo
,checkInDate,Active,adminApproval});

reg.save().then((obj)=>{console.log("successfully registered");

   res.status(200).send(obj); //redirect to Login Page and dont allow him to book rooms now itself

}).catch((error)=>{

    console.log("couldn't registered");

    res.send(error);
```

```
});

})

// i have to login as Admin and Accept new Registrations by validating, for now im allowing all after wards i set Validation system

app.post('/login',(req,res)=>{

    const email=req.body.email;

    const password=req.body.password;

    if(!validator.isEmail(email)||password==="")

    {

        res.send("Enter Valid Details to Login");

    }

  register.findOne({email:email,password:password}).then((data)=>{

        if(data){

         res.send(data);

        }

         else{

            res.send("No user found with this values");

         }

    }).catch((error)=>{

        res.send(error);

    });

});

// fetching user full details and sending to client

app.post('/userDetail',(req,res)=>{
```

```
  const email=req.body.email;
  if(!validator.isEmail(email))
   {
      res.send("Enter Valid Details to Login");
   }
register.findOne({email:email}).then((data)=>{
      if(data){
          console.log(data)
         res.json({
          name:data.name,
          email:data.email,
          mobile:data.mobile,
          BookedRoom:data.BookedRoomNo,
          AmountPaid:data.AmountPaid,
          TimePeriod:data.TimePeriod,
          checkInDate:data.checkInDate
})
      }
       else{
          res.json({response:"No user found with this values"});
       }
   }).catch((error)=>{
      res.send(error);
   });
```

```
});
// Rooms registration
app.post('/roomregister',(req,res)=>{
 const RoomNumber=req.body.RoomNumber;
    const floor=req.body.floor;
    const roomCapacity=req.body.roomCapacity;
    const FreeRooms=roomCapacity;
    const RoomRent=req.body.RoomRent;
    const roomDescription=req.body.roomDescription;
    const roomRating=req.body.roomRating;
    const roomFeatures=req.body.roomFeatures;
    const Ac=req.body.Ac;

if(RoomNumber===""||floor===""||FreeRooms===""||roomCapacity===""||RoomRent===""||roomDescri
ption===""||roomRating===""||roomFeatures===""||Ac==="")
{
    res.send("Enter Valid Details to register");
}
   const reg=new rooms({RoomNumber:RoomNumber,floor:floor,roomCapacity:roomCapacity,
    FreeRooms:FreeRooms,RoomRent:RoomRent,roomDescription:roomDescription,
    roomRating:roomRating,roomFeatures:roomFeatures,Ac:Ac});
  reg.save().then((obj)=>{console.log("rooms successfully registered");
   res.send(obj); //redirect to Login Page and dont allow him to book rooms now itself
}).catch((error)=>{
    console.log("couldn't registered");
```

```
    res.send(error);
});
})
// Here im importing all routes which i have mentioned in route.js file
app.use(require(path.join(__dirname,"/Routes/route.js")));
app.listen(port,()=>{
    console.log(`listening to port ${port}`);
})
```

4.4.2 Front End:

Home.js:

```
import React from 'react'
import NavBar from './NavBar';
import final1 from './final1.jpg'
function Home() {
  return (
    <div>
      <NavBar/>
      <div class="myClass">
        <img src={final1} alt="Not found" />
        {/* <strong><h1 class="joke" >Hostel Management System</h1></strong> */}
        </div>
    <div class="two">
        <h2>This Module helps the users to get better shelter and great user experience</h2>
        <button><a href='/register'>Get Started</a></button>
```

```
    </div>
    </div>
  )
}
export default Home
```

NavBar.js:

```
import React from 'react'
import { Outlet, Link } from "react-router-dom";
function NavBar() {
  return (
    <>
        <nav class="navbar navbar-expand-lg bg-light" >
      <div class="container-fluid">
        <Link class="navbar-brand" to="/">HostelManager</Link>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav me-auto mb-2 mb-lg-0">
            <li class="nav-item">
            <Link class="nav-link" to="/">Home</Link>
            </li>
```

```
            <li class="nav-item">
            <Link class="nav-link" to="/register">Register</Link>
            </li>


             <li class="nav-item">
            <Link class="nav-link" to="/Login">Login</Link>
              </li>
           </ul>
       </div>
       </div>
      </nav>
      </>
    )
  }
export default NavBar
```

CHAPTER 5
IMPLEMENTATION

5 IMPLEMENTATION:

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involve careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over, an evaluation of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. In network backup system no additional resources are needed.

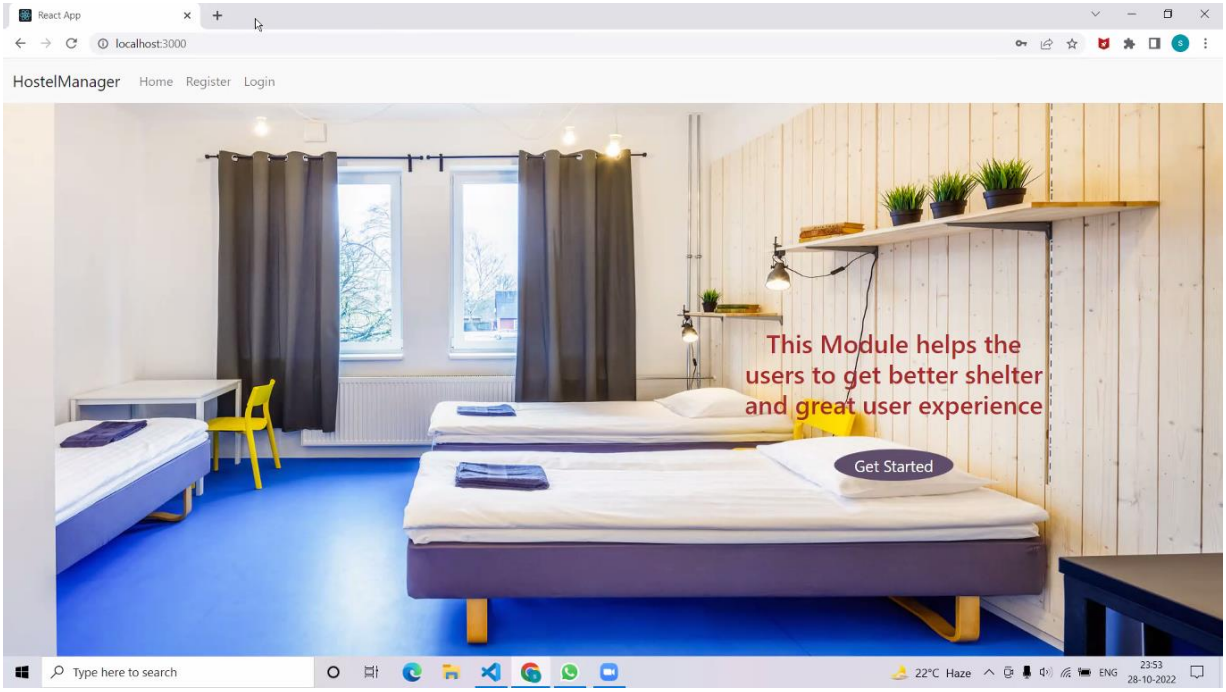
Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

5.1 Security and Maintenance:

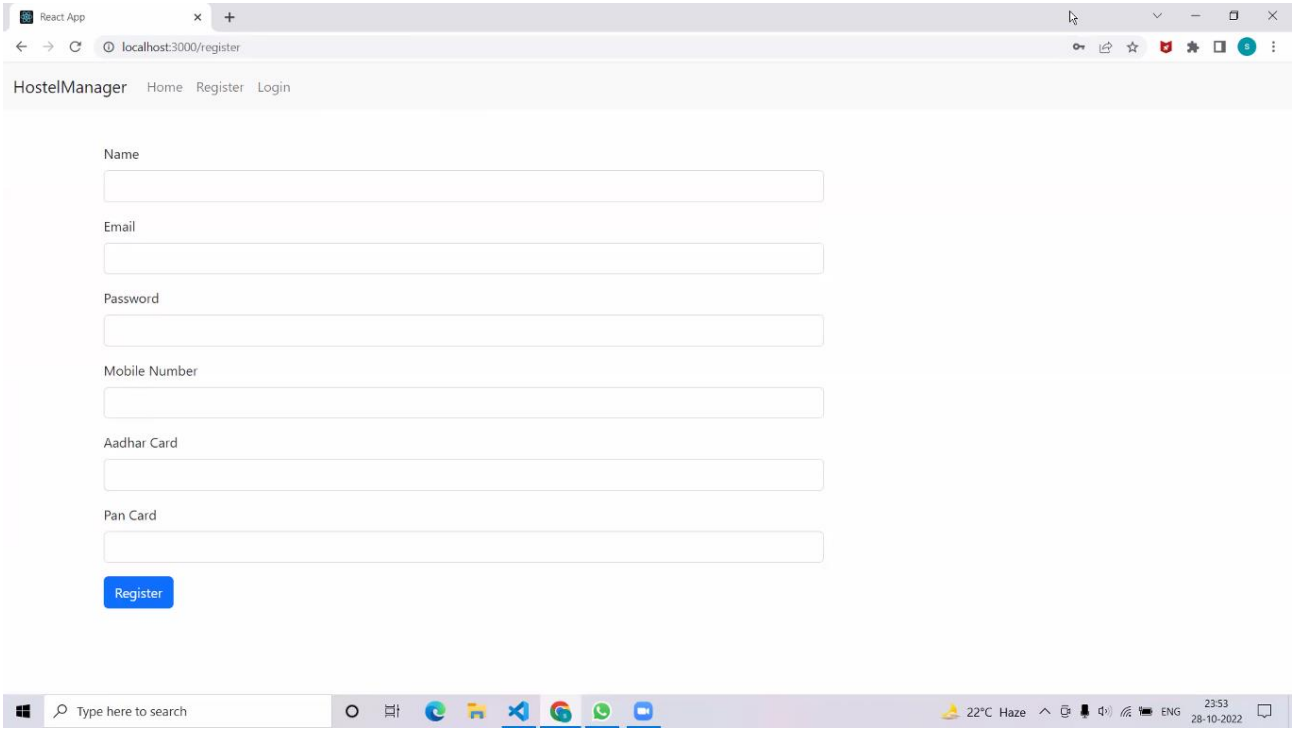
Maintenance involves the software industry captive, typing up system resources .It means restoring something to its original condition. Maintenance follows conversion to the extent that changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software. Any system developed should be secured and protected against possible hazards. Security measures are provided to prevent unauthorized access of the database at various levels. An uninterrupted power supply should be so that the power failure or voltage fluctuations will not erase the data in the files. Password protection and simple procedures to prevent the unauthorized access are provided to the users .The system allows the user to enter the system only through proper user name and password

CHAPTER 6
RESULTS AND DISCUSSION

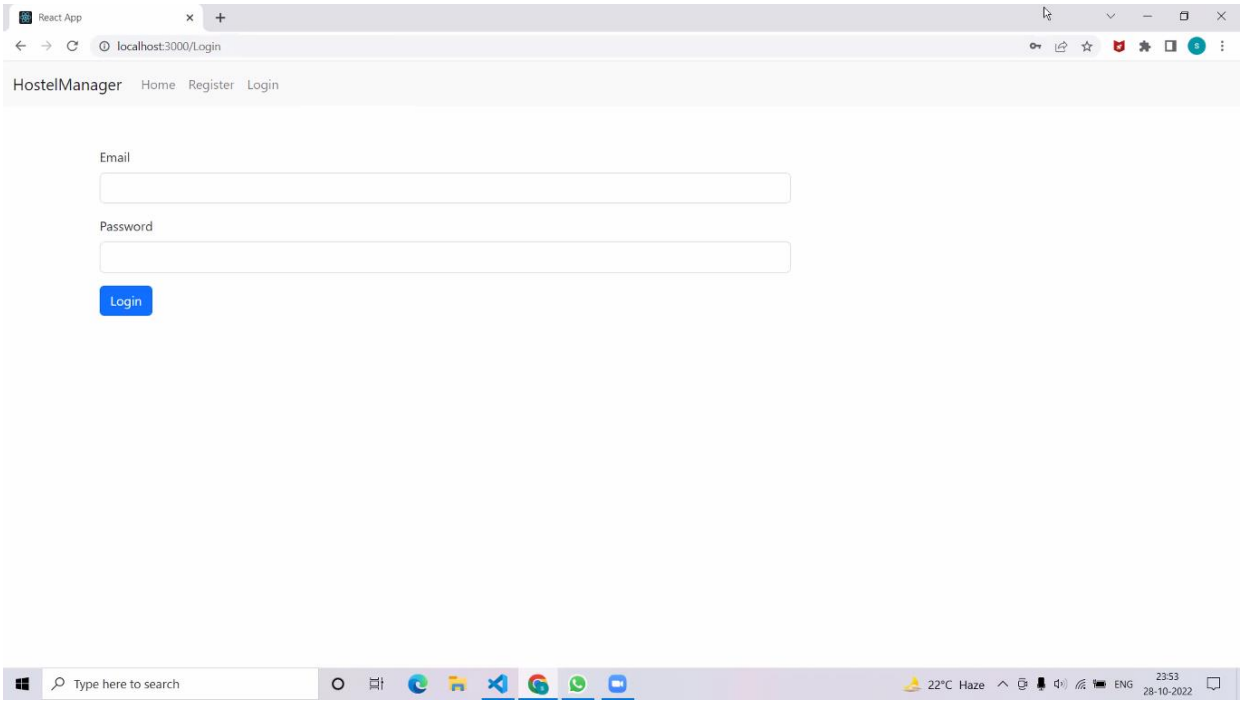
6.1Home page: This page shows us what user can see and access. User can register and login from this page.



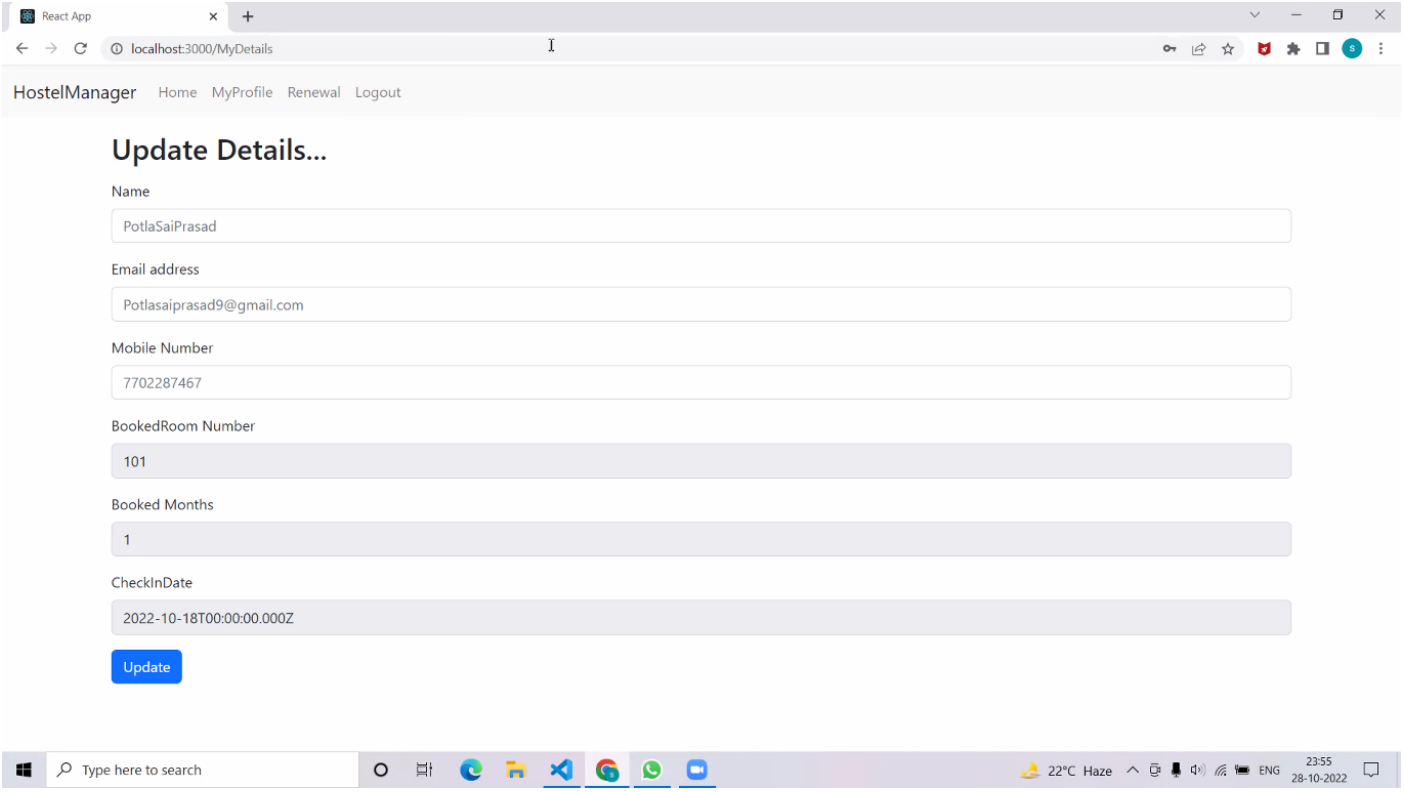
6.2 Register page: In this page user can register and can find login details for further steps. In this form it contains name, email, password, mobile number, Aadhar card number and pan card



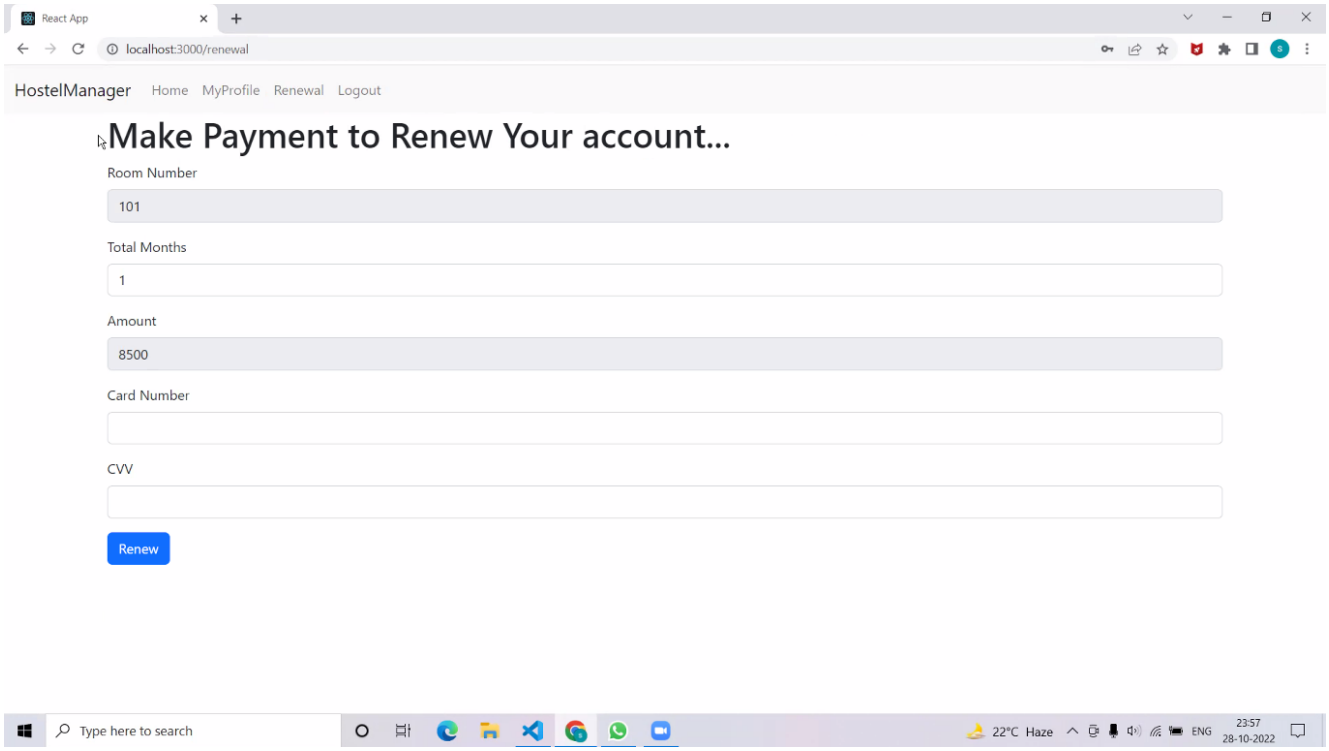
6.3 Login Page: In this page user can login into the website with the credentials provided in register page. It contains email and password.



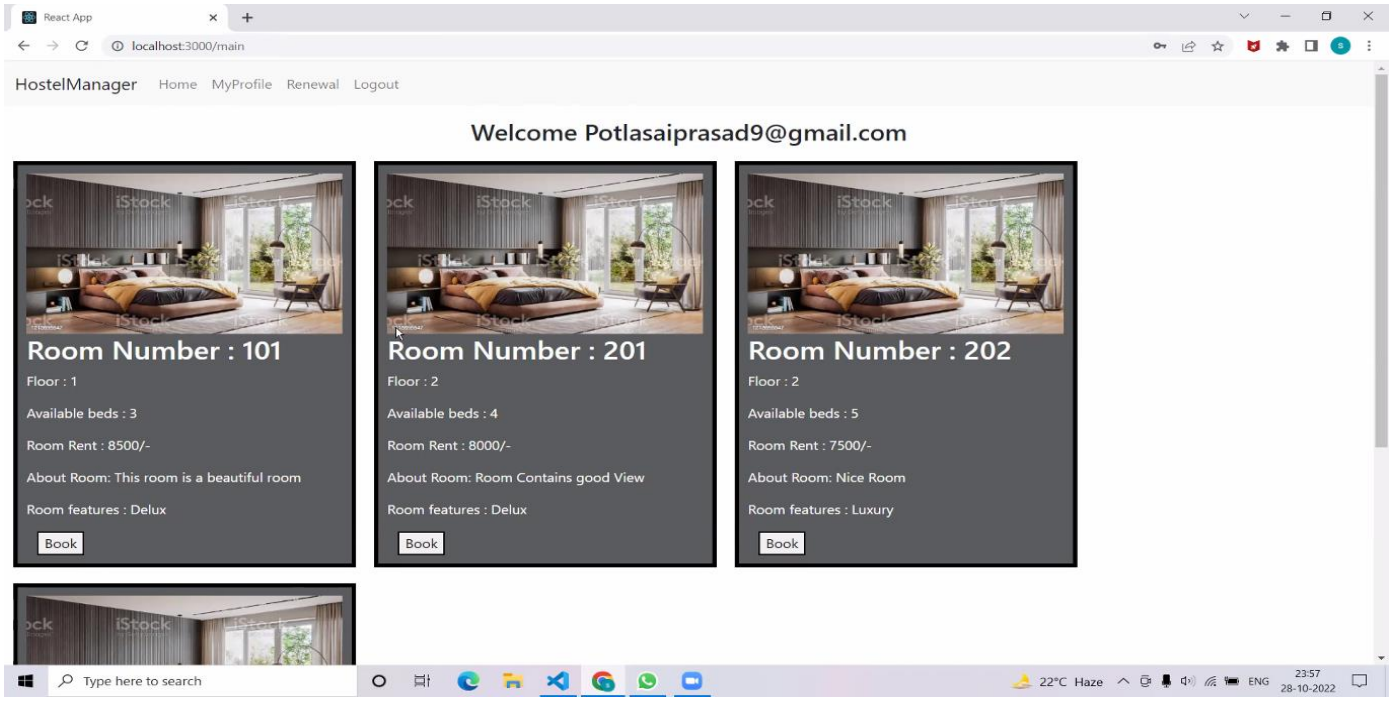
6.4 Update details page: In this page user can update the details if any wrong in the details of the user.



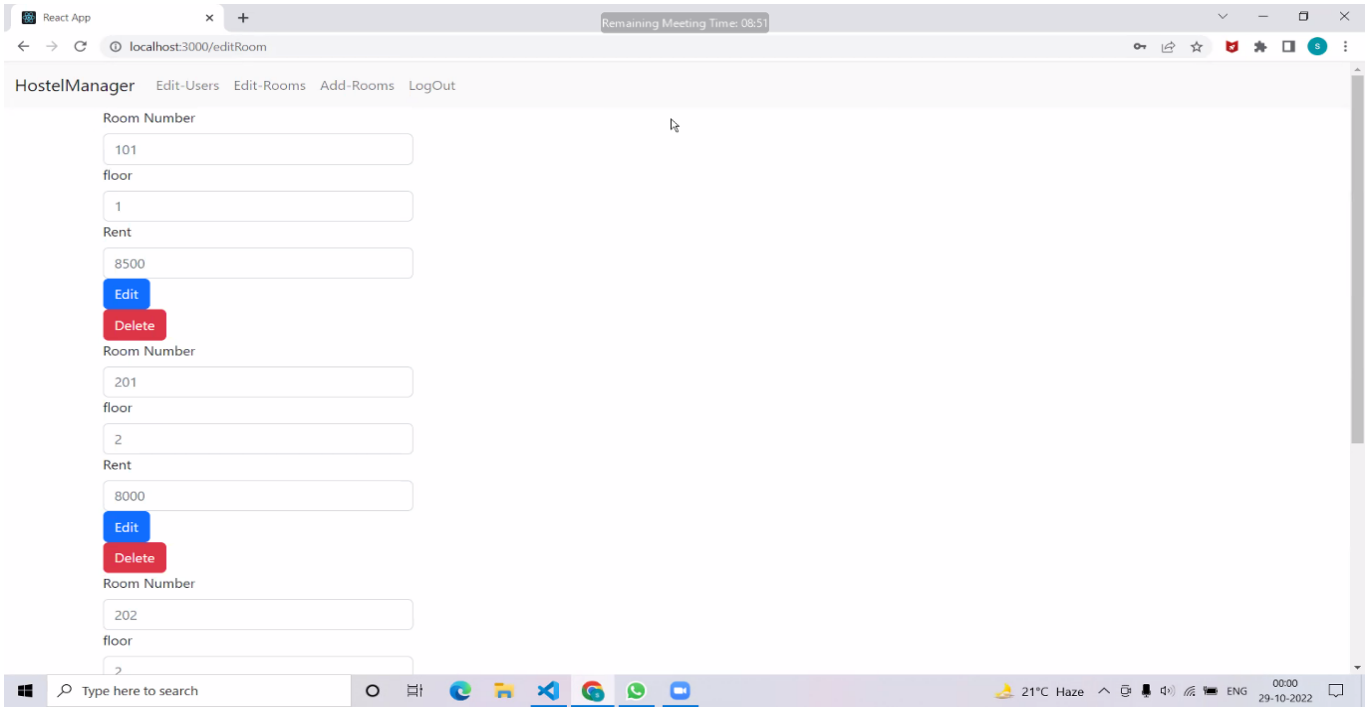
6.5 Renewal page: In this page user can renewal the details about accommodation once he completed his duration time in the hostel.



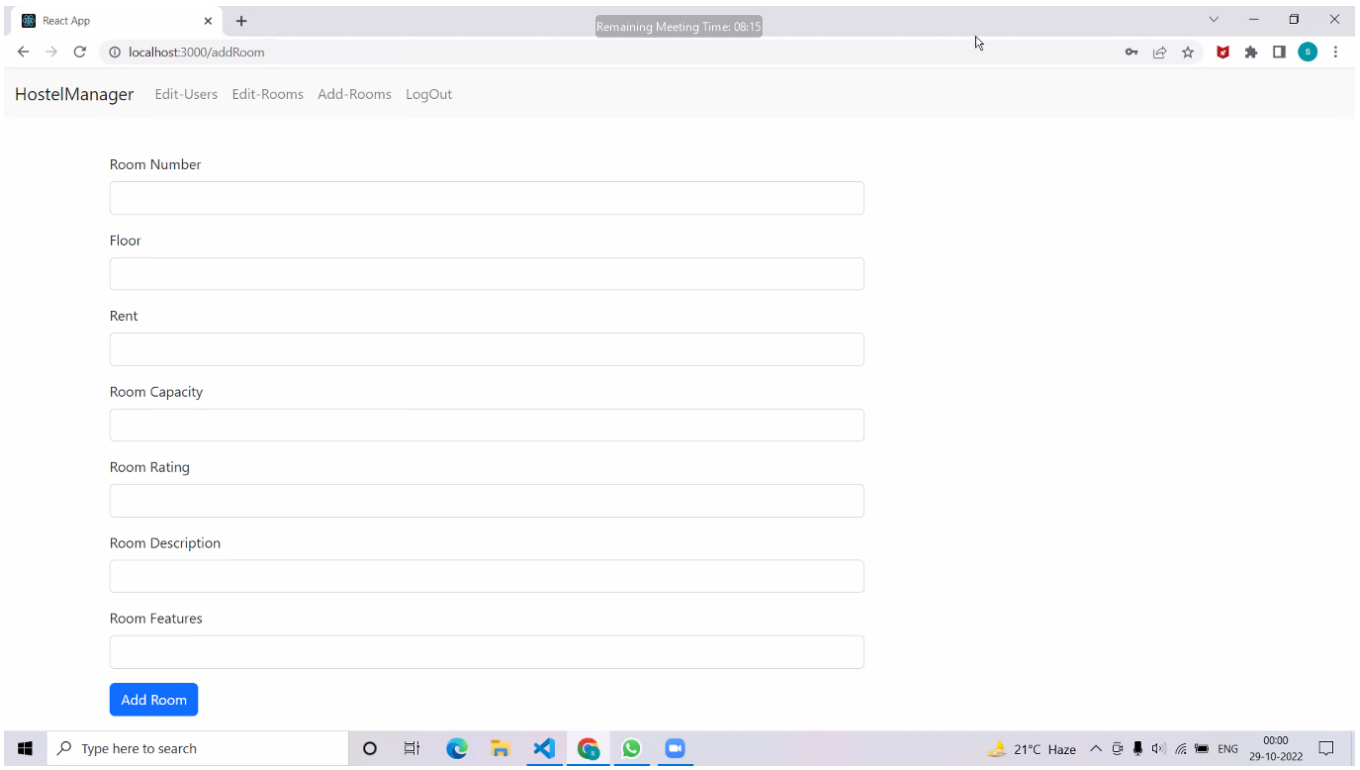
6.6 Room Details: In this page user can see the details of the room and also book the room as per the user requirements provided in the website.



6.7 Edit-User Page: In this page Admin can edit the details of the user as per the requirements. If any user vacates or join new these all details are managed by the admin.



6.8 Edit-Rooms: In this page admin can add rooms once any room is empty. If any user duration is done then that room can be added into the room list.



CHAPTER 7
CONCLUSION AND FUTURE WORK

CONCLUSION

This project developed using HTML and MongoDB is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement.

The expanded functionality of today's software requires an appropriate approach towards software development. This hostel management software is designed for people who want to manage various activities in the hostel. For the past few years the numbers of educational institutions are increasing rapidly. Thereby the numbers of hostels are also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more users friendly and more GUI oriented

REFERENCES

- Internet & World Wide Web: How to Program Deitel, PJ Deitel.
- Code for Interview YouTube Channel.
- Database System Concepts, by Silberschatz, Sudarshan, and Korth.
- https://github.com/topics/hostel-management