Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

B.Tech CE Semester - V

Subject: (CE-515) Advanced Technologies

Project Title:

Hotel Management System

By:

Vagh Divya Rameshbhai

(Roll No: CE136 ID:18CEUSF116)

Guided by:

Prof. Prashant M. Jadhav

Associate Professor Dept. of Comp. Engg.DDU

Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

B.Tech CE Semester – V

Certificate:

This is to certify that Advanced Technologies project entitled

“Hotel Management System”

is the bonafide report of work carried out by

Vagh Divya Rameshbhai

(CE136)(18CEUSF116)

Of B.Tech semester-V in the branch of

Computer Engineering

During the academic year **2020-21**

Guide:

Prof. Prashant M. Jadav

Assistant Professor of Department of Computer Engineering, Dharamsinh Desai University, Nadiad

HOD:

Dr. C. K. Bhensdadia

Head of Department of Computer Computer Engineering, Dharamsinh Desai University, Nadiad

1. Abstract:

The Hotel Management System is a web application for to book a room . The main objective is to book a car without going to hotel it's save a lots of time of user. User can choose a room as per their choices. They can check all room book history in this application.

2. Introduction:

I. Brief Introduction:

The Hotel Management System which is going to be implemented for Hotel Sunshine will automate the major operations of the hotel. The reservation system is to keep track in room reservation and check availability. There is the facility to login, registration, log out etc. User can reserve room from reservation module. Using this system you can manage check in and check out process easily. User can check rooms-type according to their need. User can also give feedback of the hotel.

II. Technologies Used:

- Cascading Style Sheet (CSS 3) for styling the HTML pages.
- Bootstrap4 for styling the HTML pages using predefined classes of bootstrap.
- JavaScript for providing dynamic content for the HTML pages.
- JQuery to provide dynamic data to the request pages.
- Angular for frontend purposes.
- Express.js for routing purposes.
- Node.js for backend purposes.
- Typescript used in Angular 6 for accessing all the components of the same.
- Mongo DB as a database manager to save and fetch data from the database using Node.js mongoose module.

III. Tools Used:

- Visual Studio Code

- MongoDB Atlas

IV. **Platform Used:**

- Localhost/4200 for Angular
- Localhost/3000 for Node.js

3. Software Requirement Specification

➢ Functional Requirements:

Function 1	**Registration**
Input:	First name, Last name, User name, Email address, password
Output:	Successfully login and Display home page Successfully record the value in database

Function 2	**Customer Login**
Input:	Email address, password
Output:	Successfully login and Display home page

Function 3	**Room Book**
Input:	Your destination, check-in date, check-out date, room-type, number of rooms, number of adults, number of child
Output:	Successfully book the room

Function 4	**Contact us**
Input:	First name, Email address, your message
Output:	Submit your message successfully

Function 1	**Feedback**
Input:	First name, last name, Email address, Your experience, Feedback
Output:	Successfully submit the feedback

4. Design:

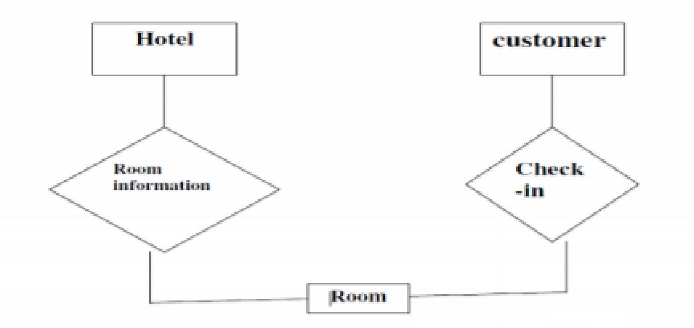
I. XML:

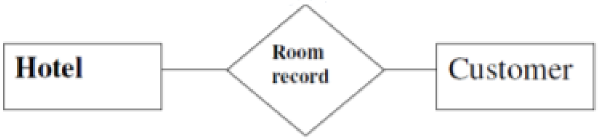
```
<User>
<User id=”U_01”>
<firstname>Divya</firstname>
<lastname>Vagh</lastname>
<username>Divya123</username>
<email>abc123@gmail.com</email>
<password>abc123</password>
</User>
</User>
```

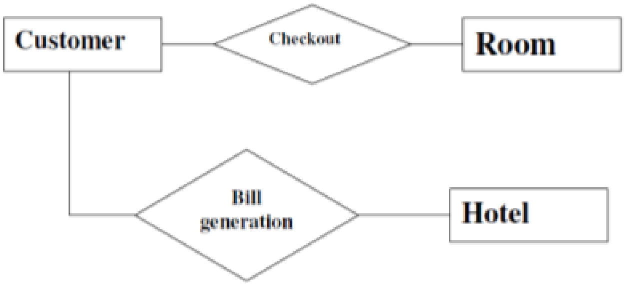
II. XSD:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Created with Liquid Technologies Online Tools 1.0 (https:/
/www.liquid-technologies.com) -->
<xs:schema attributeFormDefault="unqualified" elementFormDefaul
t="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="User">
 <xs:complexType>
  <xs:sequence>
        <xs:element name="User">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="firstname" type="xs:string" />
              <xs:element name="lastname" type="xs:string" />
              <xs:element name="username" type="xs:string" />
              <xs:element name="email" type="xs:string" />
              <xs:element name="password" type="xs:string" />
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" use="requi
red" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

III. E-R Diagram:

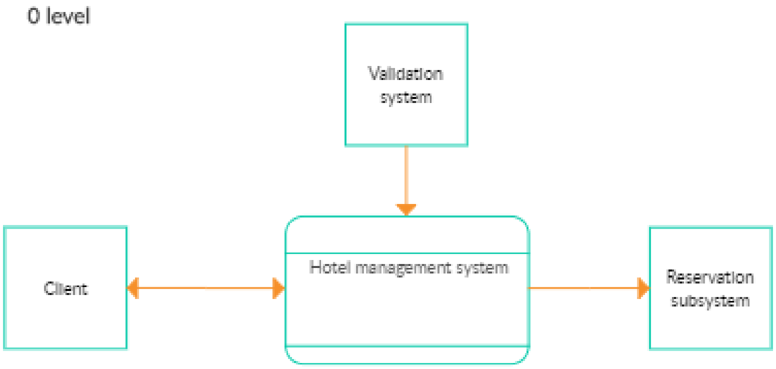




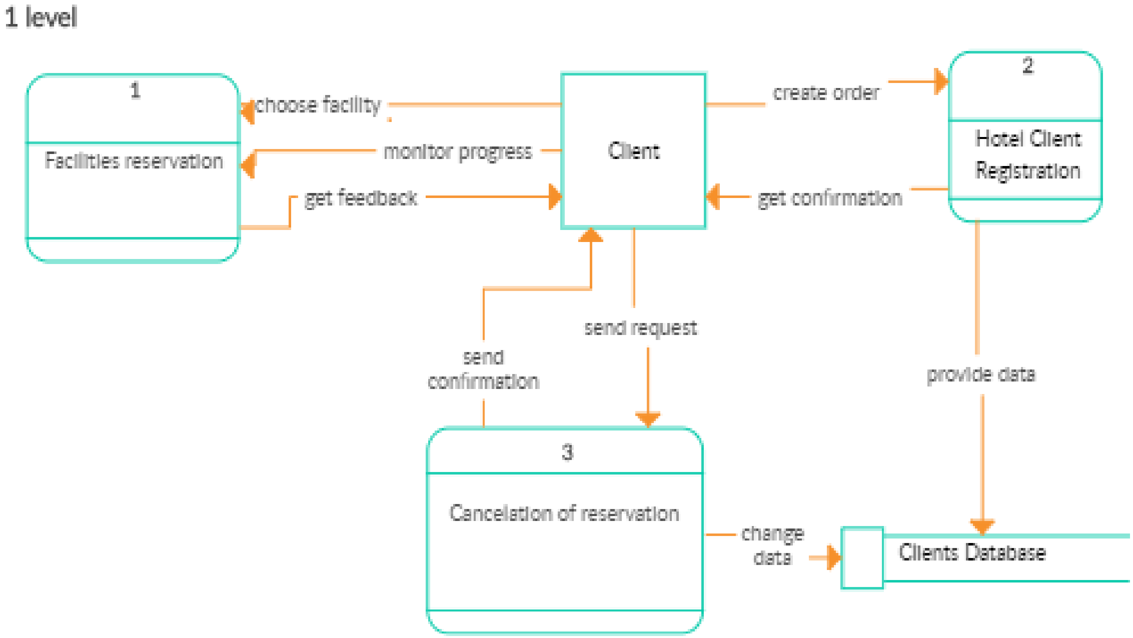


IV. DFD:

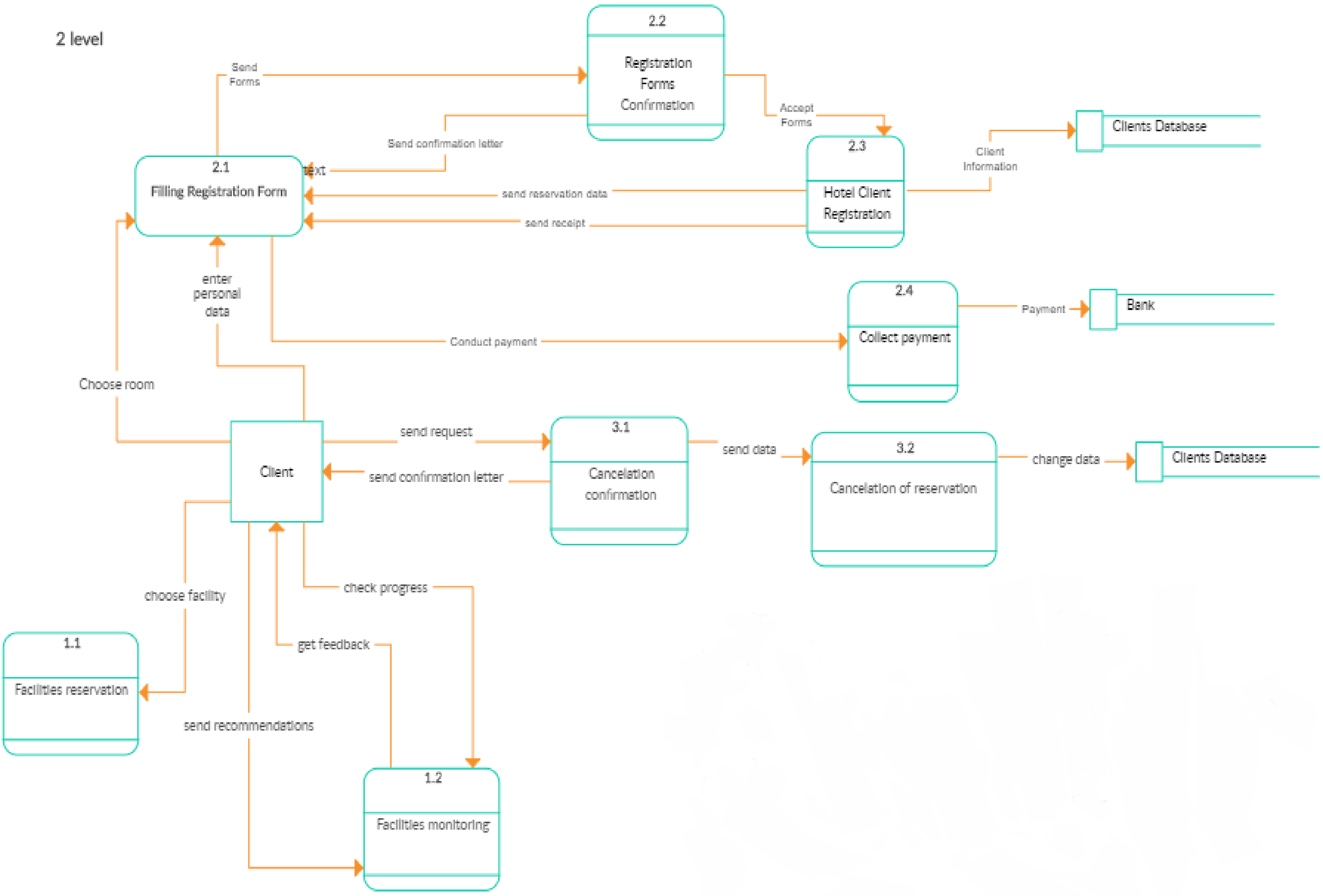
Level 0:



Level 1:


1 level
1
Facilities reservation
choose facility
monitor progress
get feedback
Client
create order
get confirmation
2
Hotel Client Registration
send request
send confirmation
provide data
3
Cancelation of reservation
change data
Clients Database

Level 2:



5. Implementation Details:

I. Modules:

V. Home Module:

In the home page, a navigation bar to go the desired location to go to the other pages like lohin module, room book module, feedback module etc.

VI. Login Module:

In the Login page user must enter a valid his/her email id and password to login in the system.

VII. **Signup Module:**

User must enter his/her first name, last name, username, email id and password to sign up.

VIII. **Room book Module:**

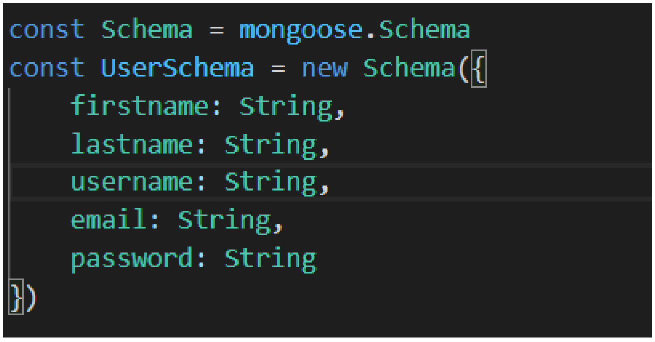
When user click on room book option in the navbar, a room book page open. This module maintains the information of all the booking of the units, as pet the guest requirements like room types, number of rooms etc.

IX. Feedback Module:

When user click on feedback option in the navbar, a feedback page opens. User can add feedbacks in this page.

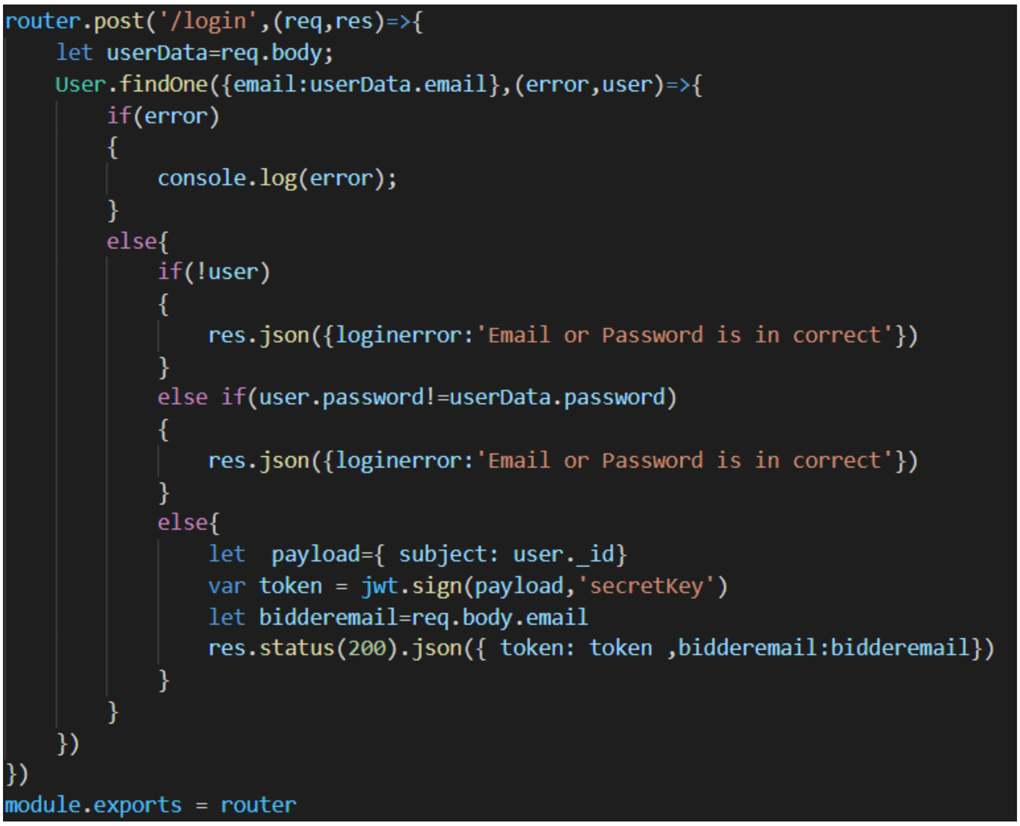
II. Function Prototype:

X. Database Schemas For MongoDB:



```
const Schema = mongoose.Schema
const UserSchema = new Schema({
    firstname: String,
    lastname: String,
    username: String,
    email: String,
    password: String
})
```

XI. Getting User Profile Data:



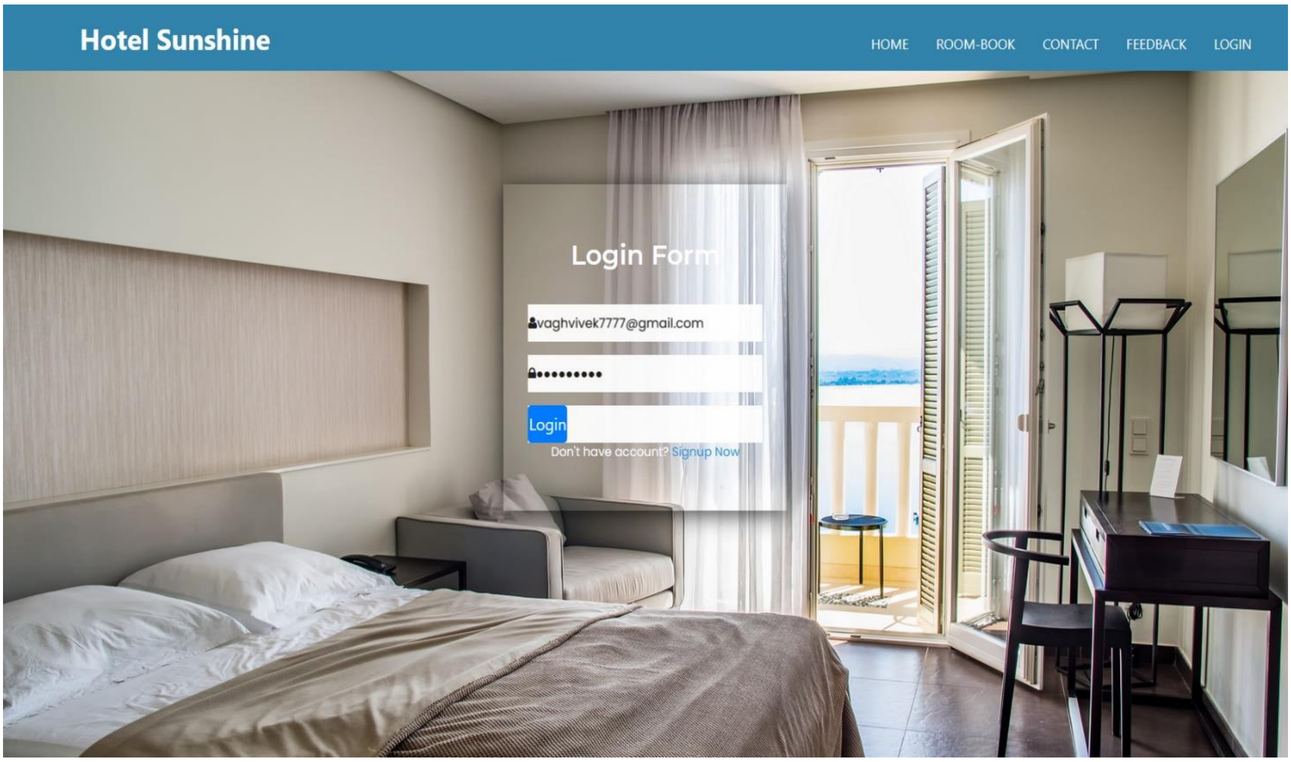
```
router.post('/login',(req,res)=>{
    let userData=req.body;
    User.findOne({email:userData.email},(error,user)=>{
        if(error)
        {
            console.log(error);
        }
        else{
            if(!user)
            {
                res.json({loginerror:'Email or Password is in correct'})
            }
            else if(user.password!=userData.password)
            {
                res.json({loginerror:'Email or Password is in correct'})
            }
            else{
                let  payload={ subject: user._id}
                var token = jwt.sign(payload,'secretKey')
                let bidderemail=req.body.email
                res.status(200).json({ token: token ,bidderemail:bidderemail})
            }
        }
    })
})
module.exports = router
```

6. Testing:

XII. We used Mozilla Firefox to do front-end testing.
XIII. We used postman to test back-end APIs.
XIV. We used ng serve for front end testing and node server for back-end testing.

7. Screen Shots:

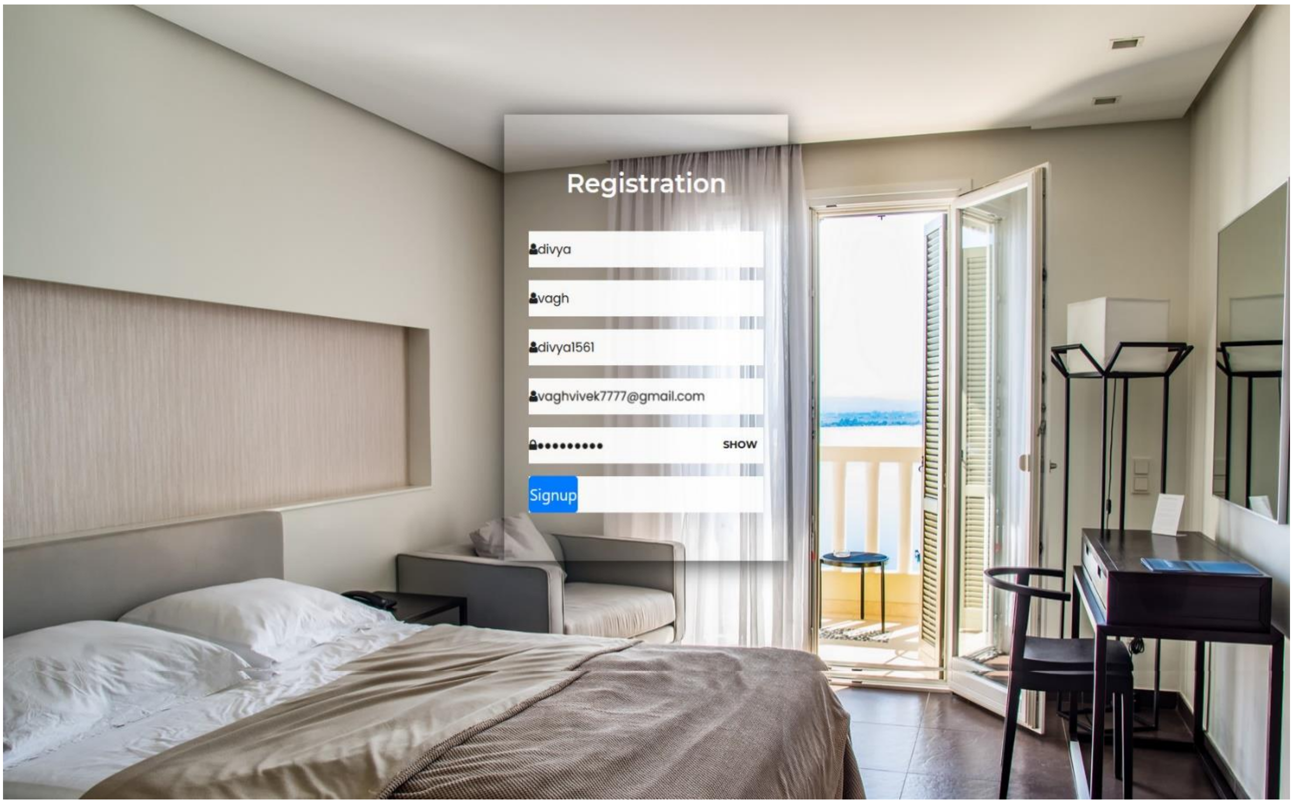
- **Login Page:**



vaghvivek7777@gmail.com

••••••••••

- **Sign-Up Page:**



divya

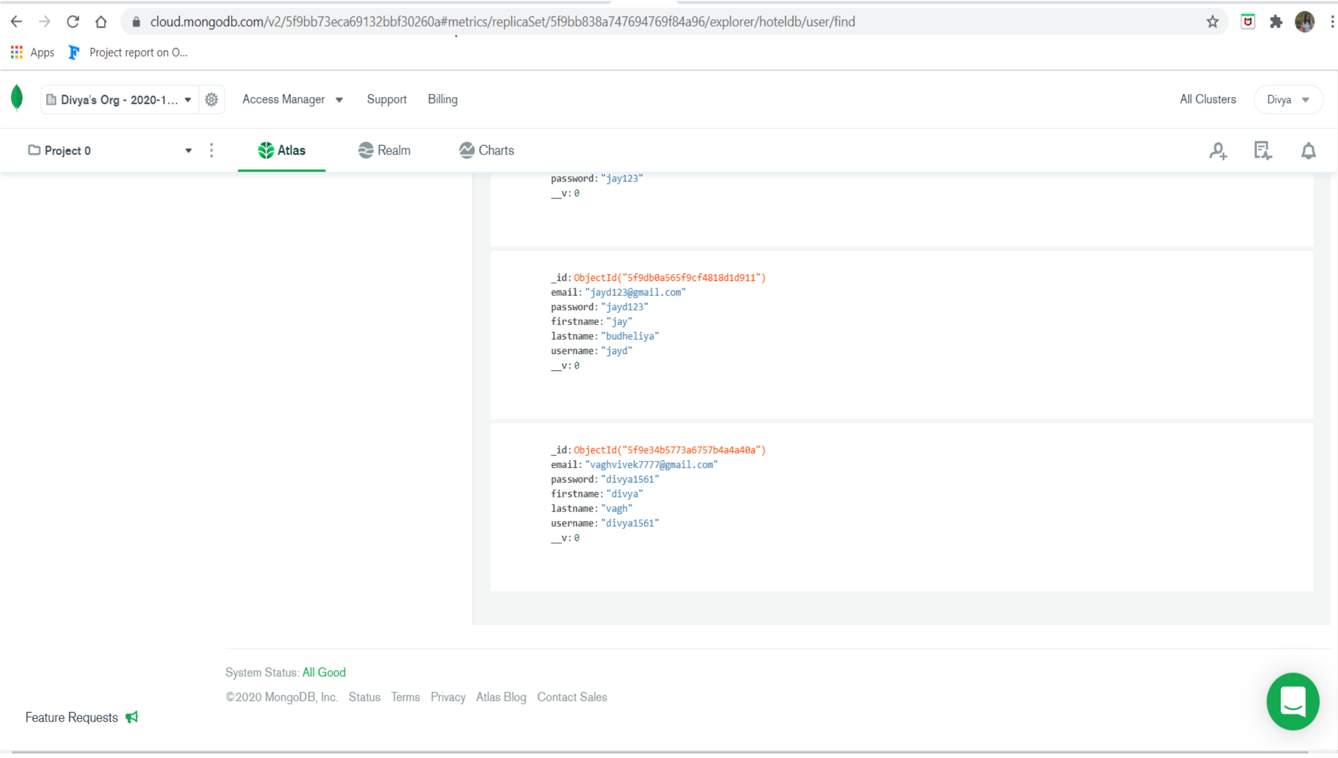
vagh

divya1561

vaghvivek7777@gmail.com

●●●●●●●●●

- **Successfully Stored value in Database:**



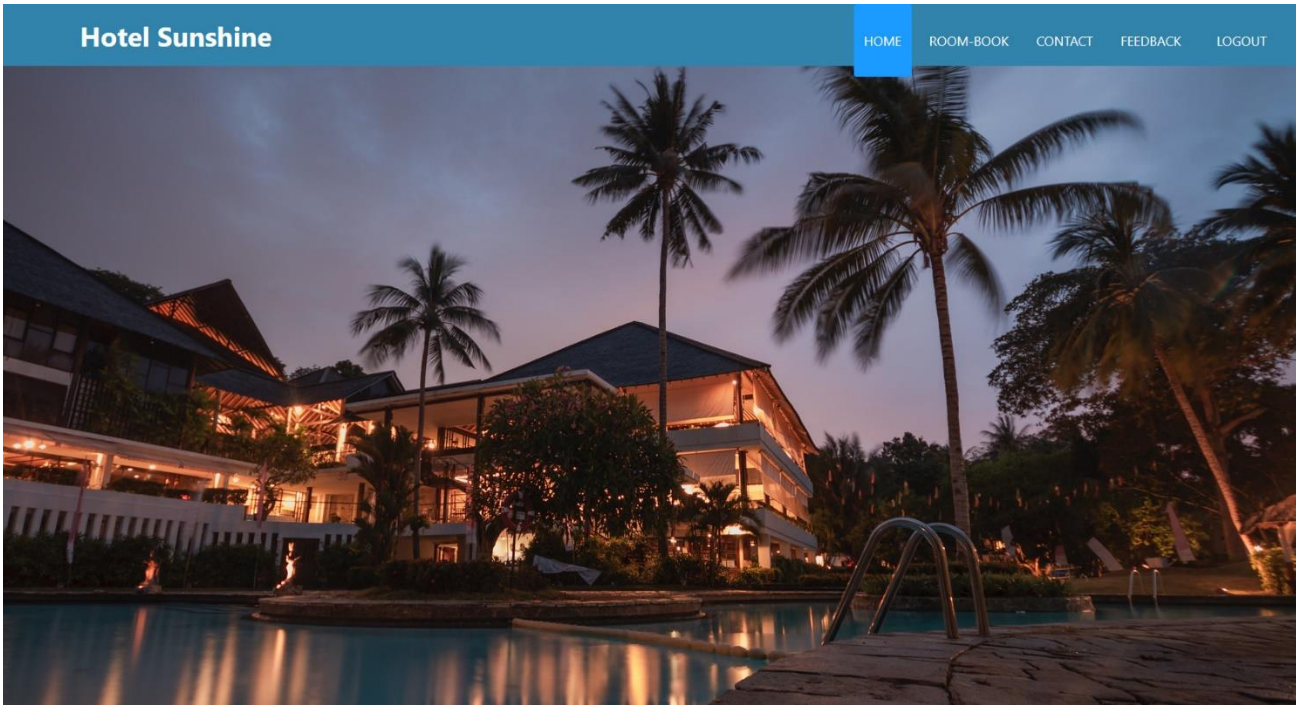
cloud.mongodb.com/v2/5f9bb73eca69132bbf30260a#metrics/replicaSet/5f9bb838a747694769f84a96/explorer/hoteldb/user/find

```
password: "jay123"
__v: 0

_id: ObjectId("5f9db0a565f9cf4818d1d911")
email: "jayd123@gmail.com"
password: "jayd123"
firstname: "jay"
lastname: "budheliya"
username: "jayd"
__v: 0

_id: ObjectId("5f9e34b5773a6757b4a4a40a")
email: "vaghvivek7777@gmail.com"
password: "divya1561"
firstname: "divya"
lastname: "vagh"
username: "divya1561"
__v: 0
```

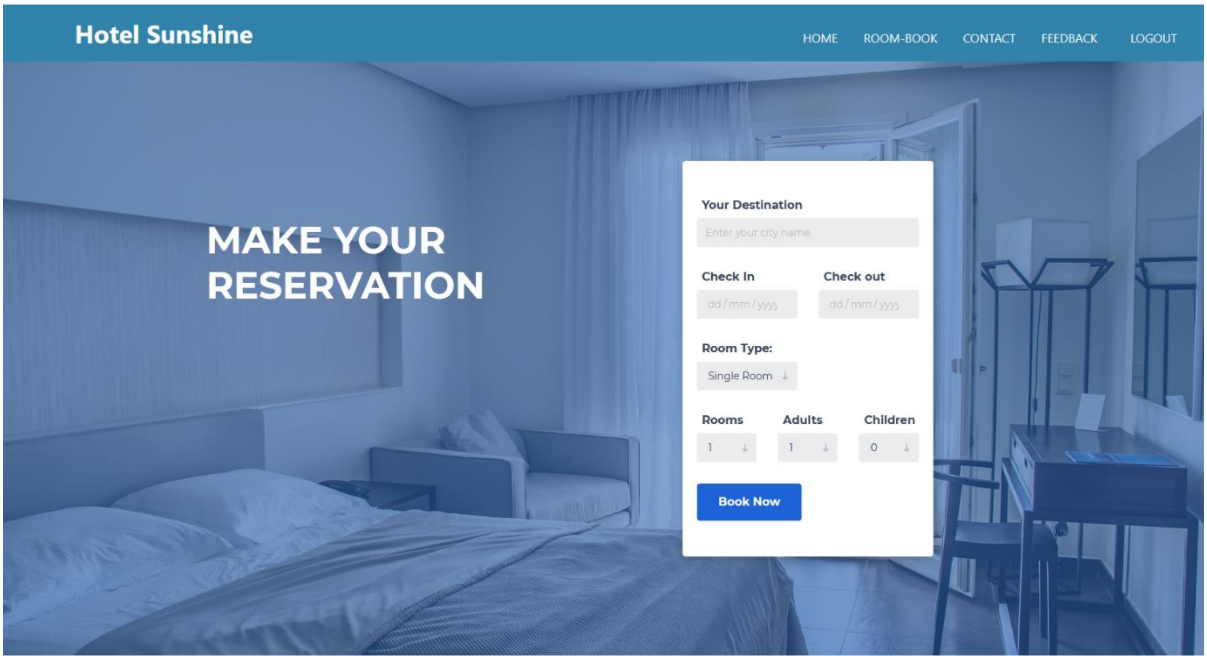
System Status: All Good

©2020 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

- **Home Page:**



- **Room-Booking Page:**



MAKE YOUR RESERVATION

Your Destination: Enter your city name

Check In: dd / mm / yyyy

Check out: dd / mm / yyyy

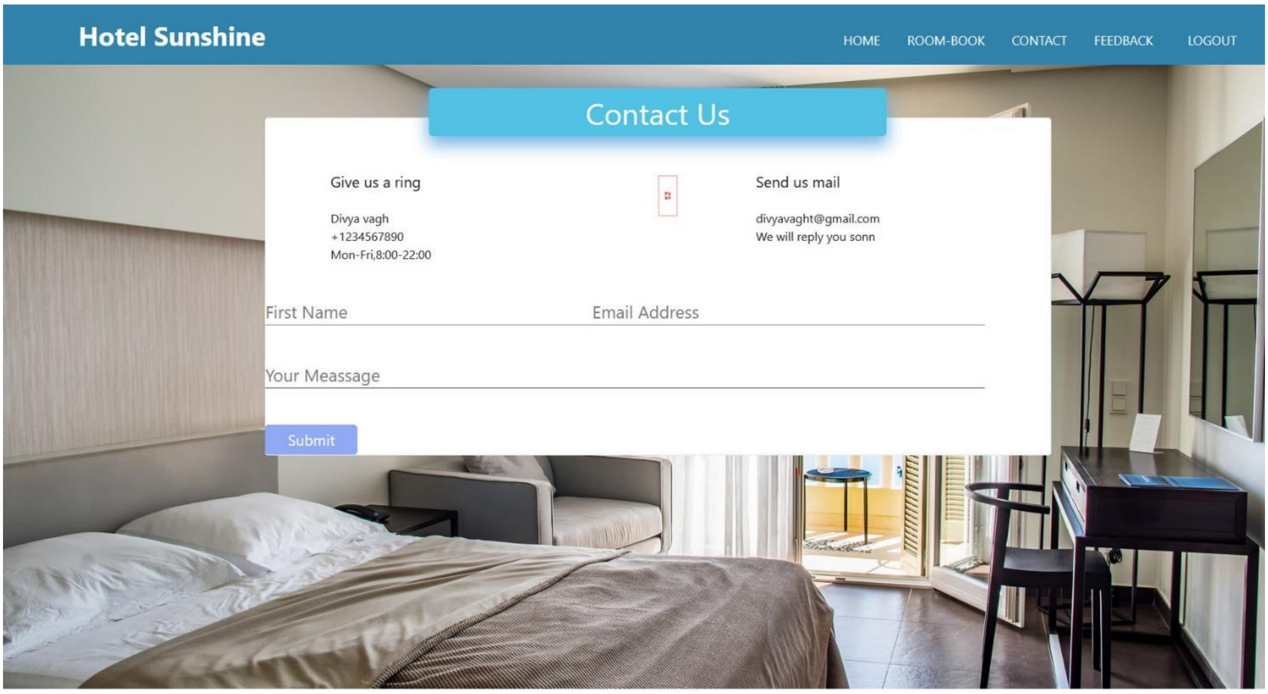
Room Type: Single Room

Rooms: 1

Adults: 1

Children: 0

- **Contact Us Page:**



Contact Us

Give us a ring

Divya vagh
+1234567890
Mon-Fri,8:00-22:00

Send us mail

divyavaght@gmail.com
We will reply you sonn

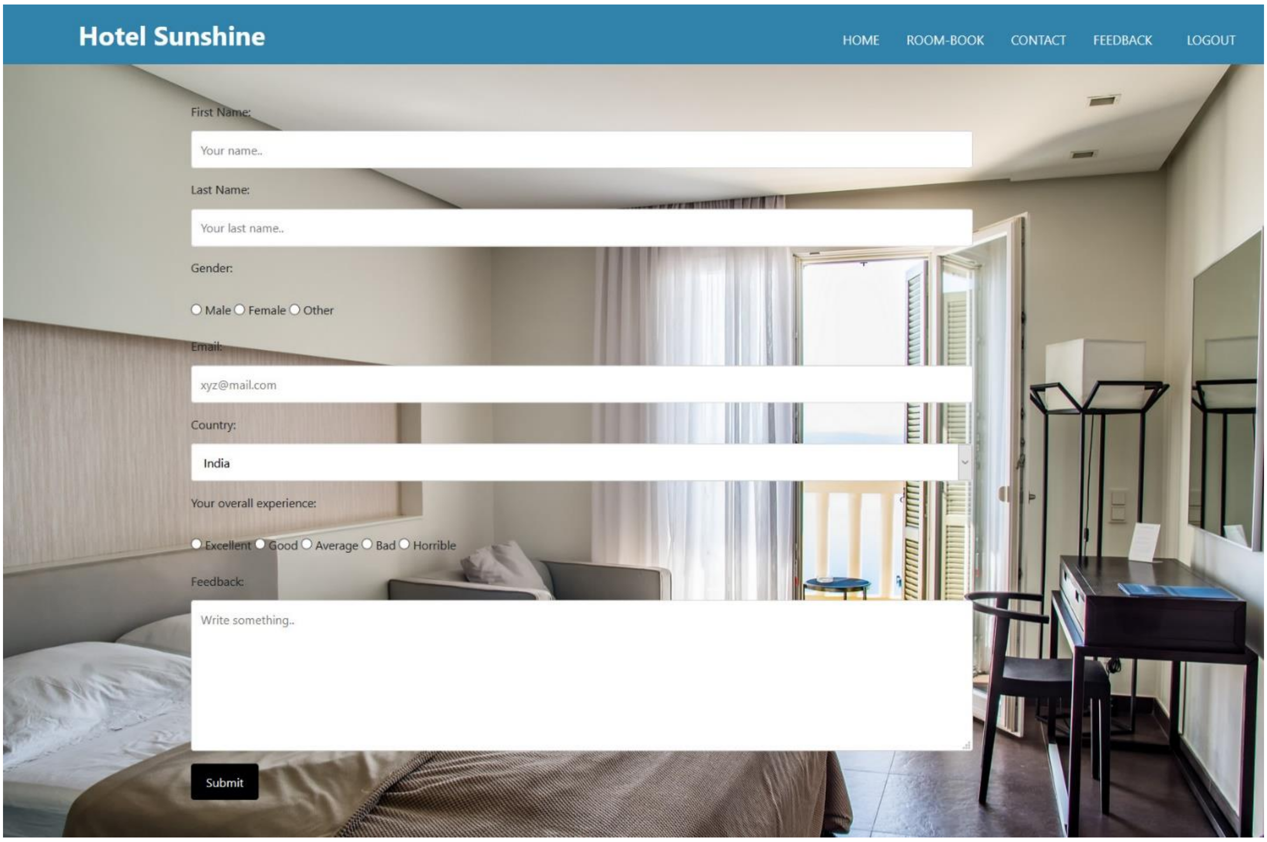
First Name

Email Address

Your Meassage

Submit

- **Feedback Page:**



First Name:

Your name..

Last Name:

Your last name..

Gender:

○ Male ○ Female ○ Other

Email:

xyz@mail.com

Country:

India

Your overall experience:

○ Excellent ○ Good ○ Average ○ Bad ○ Horrible

Feedback:

Write something..

Submit

8. Conclusion:

Login functionality was successfully implemented by the use of angular. Home page Animation, Contact details was successfully implemented using CSS, bootstrap and JavaScript. Sign up details was successfully stored in database. Backend was successfully run in postman. User can successfully book the room.

9. Limitations and Future Extension:

- **Limitations:**

- Payment module is not implemented.
- There is only one hotel for booking room.
- User cannot see their room booking history and cancel the order.

- **Future Extension:**
  - Different cities and different hotels can be included.
  - Admin can be included to add and delete the room.
  - Email verification
  - Add many offers

10. Bibliography:

W3School:

- HTML5: https://www.w3schools.com/html/default.asp
- CSS: https://www.w3schools.com/css/default.adp
- JavaScript: https://www.w3schools.com/js/default.asp
- AJAX: https://www.w3schools.com/js/js_ajax_intro.asp
- jQuery: https://www.w3schools.com/jquery/default.asp
- XML: https://www.w3schools.com/xml/default.asp
- XSLT: https://www.w3schools.com/xml/xsl_intro.asp
- XSD: https://www.w3schools.com/xml/schema_intro.asp
- JSON: https://www.w3schools.com/js/js_json_intro.asp

Bootstrap4: https://getbootstrap.com/docs/4.3/getting-started/introduction/

Fonts: https://fonts.google.com/

Icons: https://fontawesome.com/icons?d=gallery

Express.js: https://expressjs.com/en/api.html

Node.js: https://nodejs.org/en/

Mongo DB: https://www.mongodb.com/

Auth0 API: https://auth0.com/

Other References: https://stackoverflow.com/

https://www.youtube.com/

https://codepen.io/