

# ds

*by minosh imantha*

---

**Submission date:** 19-May-2022 01:07AM (UTC+0530)

**Submission ID:** 1779257270

**File name:** ASSIGNMENT\_DS\_02.pdf (835.37K)

**Word count:** 8007

**Character count:** 57538

# **Assignment 2**

**Hotel Reservation System**



## **Distributed Systems**

Group ID: **2022S1\_REG\_21**

### **Group Members**

IT20132750	Izmie.A.A
IT20158422	Paboda P.D.K
IT20088682	K.D.M.I Jayathilaka
IT20134730	A.D.N Jayathilaka

## Content

1. Introduction

High-Level Architectural Diagram

Tools & Technologies

System Components

Rest API

Web Client

Desktop Application

Email Service

SMS Service

Payment gateway

Database

Sequence of Execution

Authentication and Security

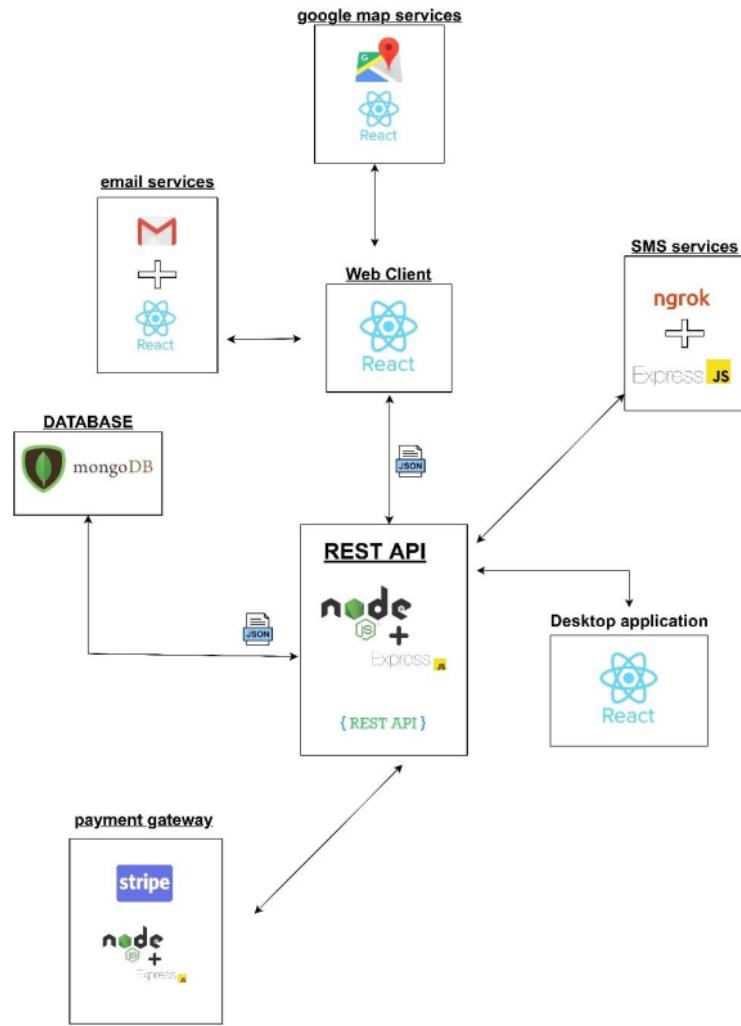
Appendix

## 1. Introduction

The hotel reservation system is implemented using Express.js and node.js for the Rest API,  
ReactJS for Web Client, and react Js for SMS Server and Email Server and the mongo DB as the  
database. This system allows the system admin to manage the reservation information add new  
reservation information and update and delete the information as well in the building.

The travelers first register to the system and create profiles and login into the system. and can  
see their profiles. a traveler may make multiple reservations The travelers can make the  
reservation and travelers can cancel their reservation booking. once the reservation confirms  
the traveler makes the payment and the system sends an SMS to the customer. traveler can see  
their booking information after the booking and login into the account. once the reservation is  
done the system will send an email to the customer. Once a reservation is made, a traveler may  
(optionally) reserve a taxi, where a request may be sent to a taxi service.

## 2. High-Level Architectural Diagram



## Tools & Technologies

- Rest API
  - Node js and express js
  - Postman
  - Vs code
- Web Client
  - VS Code
  - React Js
  - Axios HTTP Client
- Database
  - MongoDB
- Desktop Application (Admin)
  - React js
  - Vs code
- Email Service
  - Gmail
  - Vs code
  - React email
- SMS Service
  - Shoutout
  - Express js
  - Vs code
- Payment Service
  - Node js
  - Express js
  - Vs code
  - React js

## System Components

### Rest API

port 5000. Rest API has all Admin, traveler, system admin, payment options details, traveler profile, hotel location, booking reservation and Rooms CRUD operations, and send-email notification as well. Rest API has access to the MongoDB database. Web clients(react js) could access MongoDB through this Rest API.

### Rest API End Points

endpoint	Description
http://localhost:5000/ api/User/register	Create a new traveler account
http://localhost:5000/ api/User/login	Login traveler account
http://localhost:5000/ api/User/logout	Logout traveler account
http://localhost:5000/ api/User/getAllUsers	Display all traveler account information
http://localhost:5000/ user/login	Admin, system admins login
http://localhost:5000/user/register	Admin register
http://localhost:5000/user/logout	Admin, system admin logout
http://localhost:5000/api/rooms/getAllRooms	Display all rooms
http://localhost:5000/api/rooms/getRoomByid	Get a room by id ta o reservation
http://localhost:5000/api/rooms/getRoombyid	Get a room by id
http://localhost:5000/api/rooms/getallrooms	Display room
http://localhost:5000/api/rooms/addRoom	Add new rooms

http://localhost:5000/api/rooms/update	Update room
http://localhost:5000/api/rooms/id	Delete rooms
http://localhost:5000/api/bookings/bookroom	Make room reservation
http://localhost:5000/api/bookings/cancelbooking	Cancel reservation by id
http://localhost:5000/api/bookings/getuserbookings	Get users booking by id
http://localhost:5000/api/bookings/getallbookings	Get all users booking details
http://localhost:5000/smsroute/sms	Send payment confirm messages to customer
http://localhost:5000/	

## Web Client

Web Client has implemented in ReactJS and redux some part has used. Through Rest API which runs on port 5000. Web clients run on port 3000. Web client gets all room and reservation information data from that rest API. To request HTTP request web client has used Axios HTTP client.

## Desktop Application

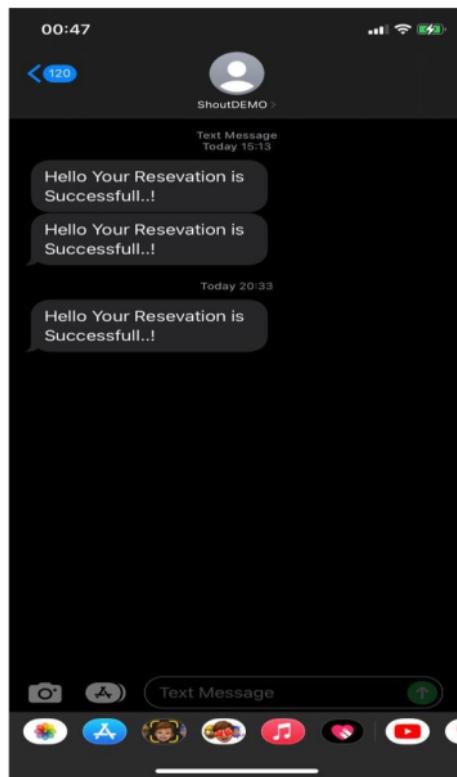
The desktop application has an Admin, and system admin login. So, unauthorized users cannot access the system where it provides security for the application. this also used run on port 3000. To request HTTP requests desktop application has used the Axios HTTP client. This provides a dashboard that displays all booking information and all customer information and room details. the system admin can add and register the admin accounts. admin can manage (update/delete) rooms and add new rooms and reservations.

## Email Service

An email will be sent to the customer's email account. The email service component was developed using react JS and it is connected to the Rest API. Google Gmail service is used to send email alerts to customers when the reservation is done. It's sent by the system admin.

## SMS Service

A web API called used to implement the SMS Service in this system. This service also runs on the react JS which is connected to the REST API. Same as the email service.



## Payment gateway

A web API called used to implement the payment Service in this system. This service also runs on the react JS and node JS and also express JS. which is connected to the REST API. Same as the email service. when the customer selects the room reservation customer must payment there booking rooms. And the customer get a confirmation message via SMS

## Database

we are using the database MongoDB. it is used by the rest of API and access database and manage data

## Authentication and Security

For approved users only, the hotel reservation system application provides an admin login. A different policy is then used to safeguard client calls as well as passport JS. They can only communicate with each other via the port that has been designated.

## Appendix

### Rest API

#### Package.Json (backend)

```
{
  "name": "restfulbackend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\"Error: no test specified\\" && exit 1",
    "start": "nodemon server.js"
  },
  "author": "DSlabAssignment02",
  "license": "ISC",
  "dependencies": {
    "axios": "^0.27.2",
    "bcrypt": "^5.0.1",
    "cookie-parser": "^1.4.6",
    "cors": "^2.8.5",
    "express": "^4.17.3",
    "jsonwebtoken": "^8.5.1",
    "moment": "^2.29.3",
    "mongoose": "^6.3.1",
    "passport": "^0.5.2",
    "passport-jwt": "^4.0.0",
    "passport-local": "^1.0.0",
    "shoutout-sdk": "^3.0.4",
    "stripe": "^9.1.0",
    "uuid": "^8.3.2"
  },
  "devDependencies": {
    "nodemon": "^2.0.15"
  }
}
```

## Model class

41  
Admin.js

```
const mongoose = require('mongoose');
const bcrypt = require('bcrypt');

const UserSchema = new mongoose.Schema({

    username :{
        type : String,
        required : true,
        min : 6,
        max : 65
    },
    51
    password : {
        type : String,
        required : true
    },
    role : {
        type : String,
        enum : ['user','admin','sysadmin'],
        required: true
    },
    email : {
        type : String,
        required : true
    },
});

8
UserSchema.pre('save',function(next){

    if(!this.isModified('password'))
        return next();
```

```
bcrypt.hash(this.password,10,(err,passwordHash)=>{
  if(err)
    return next(err);
  this.password = passwordHash;
  next();
});

UserSchema.methods.comparePassword = function(password,cb){
  bcrypt.compare(password,this.password,(err,isMatch)=>{
    if(err)
      return cb(err);
    else{
      if(!isMatch)
        return cb(null,isMatch);
      return cb(null,this);
    }
  });
}

module.exports = mongoose.model('User',UserSchema);
```

### booking.js

```
const mongoose = require("mongoose");

const bookingSchema = mongoose.Schema({
```

```
room:{type:String, required:true},
roomid:{type:String, required:true},
userid:{type:String , required:true},
fromdate:{type:String, required:true},
todate:{type:String, required:true},
totalDays:{type:String , required:true},
totalAmount:{type:String, required:true},
transactionId:{type:String, required:true},
status:{type:String, required:true , default:'booked'},
}, [
  timestamps:true,
})

const bookingModel=mongoose.model('book' , bookingSchema)

module.exports = bookingModel
```

#### room.js

```
const mongoose = require("mongoose");
const roomSchema = mongoose.Schema({
  name: {
    type: String,
    required: true,
  },
  description: {
    type: String,
    required: true,
  },
  imageUrl: [],
  phoneNumber: { type: String, required: true },
  rentPerDay: { type: String, required: true },
  type: { type: String, required: true },
});

const roomModel = mongoose.model("rooms", roomSchema);

module.exports = roomModel;
```

#### user.js

```
const mongoose = require("mongoose");

const userSchema = mongoose.Schema(
{
  3
  name: { type: String, require },
  email: { type: String, require },
  country: { type: String, require },
  phononenumber: { type: String, require },
  password: { type: String, require },
  isAdmin: { type: Boolean, require, default: false },
},
{
  timestamps: true,
}
);

module.exports = mongoose.model("admin", userSchema);
```

## Routes

bookingRoutes.js

```
const express = require("express");
const router = express.Router();
const mongoose = require("mongoose");
const { v4: uuidv4 } = require("uuid");
const moment = require("moment");

const Booking = require("../model/booking");
const Room = require("../model/room");
const stripe = require("stripe")("sk_test_51KyZL6FeoRdqg6f8CfkSdkB704x3VtsehfnbEosCyzphC6IAMvV3rYUWejnhdEOflv6aEdxTPGMzYJvv1FRMthtc00wtSqLtYz");

router.post("/bookroom", async (req, res) => {
```

```
const { room, fromdate, todate, totalDays, totalAmount, user , token } = req.body;

try {
    const customer = await stripe.customers.create({
        email: token.email,
        source: token.id,
    });

    const payment = await stripe.charges.create(
        {
            amount: totalAmount * 100,
            currency: "lkr",
            customer: customer.id,
            receipt_email: token.email,
        },
        {
            idempotencyKey: uuidv4(),
        }
    );
}

if (payment) {
    try {
        const newbooking = new Booking({
            userid: user._id,
            room: room.name,
            roomid: room._id,
            totalDays: totalDays,
            fromdate: moment(fromdate).format("DD-MM-YYYY"),
            todate: moment(todate).format("DD-MM-YYYY"),
            totalAmount: totalAmount,
            transactionId: "1234",
            status:'booked'
        });

        await newbooking.save(async (err, booking) => {
            const oldroom = await Room.findOne({ _id: room._id });

            oldroom.currentbookings.push({
                bookingid: booking._id,
                fromdate: moment(fromdate).format("DD-MM-YYYY"),
                todate: moment(todate).format("DD-MM-YYYY"),
            });
        });
    }
}
```

```

        userid: user._id,
        status:'booked'
    });
    //console.log(currentbookings)
    await oldroom.save();
});

res.send("Room Booked Successfully");
} catch (error) {
    console.log(error);
    return res.status(400).json({ message: error });
}
} else {
    res.send("Payment failed");
}
} catch (error) {
    return res.status(400).json({ message: " wrong" + error });
}
}

});

router.post("/cancelbooking", async (req, res) => {
    const {bookingid, roomid} = req.body;

    try {

        const bookingitem = await Booking.findOne({_id: bookingid})
        bookingitem.status='cancelled'
        await bookingitem.save();
        const room = await Room.findOne({_id:roomid})
        const bookings = room.currentbookings
        const
temp=bookings.filter(booking=>booking.bookingid.toString()!==bookingid)
        console.log(temp);
        room.currentbookings=temp;
        await room.save()

        res.send('Booking deleted successfully')
    } catch (error) {
        console.log(error);
        return res.status(400).json({ message: "something went wrong" });
    }
}

```

```
}

});

router.post("/getuserbookings", async (req, res) => {
  const { userid } = req.body;
  console.log(userid)
  try {
    const bookings = await Booking.find({ userid: userid }).sort({ _id: -1 });
    res.send(bookings);

  } catch (error) {
    return res.status(400).json({ message: "Something went wrong" });
  }
});

router.get("/getallbookings", async (req, res) => {
  try {
    const bookings = await Booking.find({});
    res.send(bookings);
  } catch (error) {
    return res.status(400).json({ message: error });
  }
});

module.exports = router;
```

## roomRoutes.js

```
const express = require("express");
const Router = express.Router();
const Room = require("../model/room")
const mongoose = require("mongoose");

Router.get("/getAllRooms", async (req, res) => {

    try {
        const rooms = await Room.find()
        res.send(rooms)
    } catch (error) {
        return res.status(400).json({ message: 'wrong' });
    }
});

Router.post("/getRoomByid/", async(req, res) => {
    console.log(req.body);
    try {
        const room = await Room.findOne({_id : req.body.roomid})
        res.send(room)
    } catch (error) {
        return res.status(400).json({ message: error });
    }
});

Router.put("/getRoombyid/", async(req, res) => {
    console.log(req.body);
    const {id} = req.params;
    try {
        const room = await Room.findOne({_id : req.body.id})
        res.send(room)
    } catch (error) {
        return res.status(400).json({ message: error });
    }
});
```

```
Router.get("/getallrooms", async(req, res) => {
    console.log(req.body);
    try {
        const rooms = await Room.find({})
        res.send(rooms)
    } catch (error) {
        return res.status(400).json({ message: error });
    }
});

Router.post("/addRoom", async(req, res) => {
    const { room , name,
        rentPerDay ,description ,phoneNumber ,type ,image1 } = req.body

    const newroom = new Room({
        name : room,
        rentPerDay, name,
        description , phoneNumber , type , imageUrl:[image1 ]
    })
    try {
        await newroom.save()
        res.send(' Room Added Successfully')
    } catch (error) {
        return res.status(400).json({ error });
    }
});

Router.put("/update/:id" , (req,res) =>{
    Room.findById(req.params.id)
    .then(Room =>{
        Room.name=req.body.name,
        Room.rentPerDay=req.body.rentPerDay,
        Room.description=req.body.description,
        Room.phoneNumber=req.body.phoneNumber,
        Room.type=req.body.type,
```

```

Room
  .save()
  .then(() => res.json("Room was updated successfully"))
  .catch(err => res.status(400).json(` ${err} `));
})
.catch(err => res.status(400).json(` ${err} `));
});

2
Router.delete("/:id", (req, res)=>{
  Room.findByIdAndDelete(req.params.id)
  .then(( )=> res.json("THIS Room WAS DELETE SUCCESFULLY "))
  .catch(err=> res.status(400).json(` ${err} `))

});
module.exports = Router

```

## User.js

```

const express = require('express');
const userRouter = express.Router();
const passport = require('passport');
const passportConfig = require('../passport');
const JWT = require('jsonwebtoken');
const User = require('../model/Admin');

const signToken = userID =>{
  return JWT.sign({
    iss : "Minosh",
    sub : userID
  }, "Minosh",{expiresIn : "2h"});
}

```

```
//register
65
userRouter.post('/register',(req,res)=>{
    const { username,password,role ,email} = req.body;
    User.findOne({username},(err,user)=>{
        if(err)
            res.status(500).json({message : {msgBody : "Error has occurred",
msgError: true}});
        if(user)
            res.status(400).json({message : {msgBody : "Username is already taken",
msgError: true}});
        else{
            const newUser = new User({username,password,role,email});
            newUser.save(err=>{
                if(err)
                    res.status(500).json({message : {msgBody : "Error has occured",
msgError: true}});
                else
                    res.status(201).json({message : {msgBody : "Account successfully
created", msgError: false}});
            });
        }
    });
});

//Login
8
userRouter.post('/login',passport.authenticate('local',{session :
false}),(req,res)=>{
    if(req.isAuthenticated()){
        const {_id,username,role} = req.user;
        const token = signToken(_id);
    }
});
```

```
res.cookie('access_token',token,{httpOnly: true, sameSite:true});  
res.status(200).json({isAuthenticated : true,user : {username,role}});  
}  
});  
  
// Logout  
  
61 userRouter.get('/logout',passport.authenticate('jwt',{session :  
false}),(req,res)=>{  
  
    res.clearCookie('access_token');  
    res.json({user:{username : "", role : ""},success : true});  
});  
  
//admin user role  
4 userRouter.get('/admin',passport.authenticate('jwt',{session :  
false}),(req,res)=>{  
    if(req.user.role === 'admin'){  
        res.status(200).json({message : {msgBody : 'You are an admin', msgError :  
false}});  
    }  
    else  
        res.status(403).json({message : {msgBody : "You're not an admin,go away",  
msgError : true}});  
});  
  
//Supervisor user role  
4 userRouter.get('/sysadmin',passport.authenticate('jwt',{session :  
false}),(req,res)=>{  
    if(req.user.role === 'sysadmin'){  
        res.status(200).json({message : {msgBody : 'You are an Supervisor',  
msgError : false}});  
    }  
    else
```

```

        res.status(403).json({message : {msgBody : "You're not an Supervisor,go
away", msgError : true}});
    });

2
userRouter.get("/alldata", (req, res) =>{
    User.find()

    .then(User =>res.json(User))
    .catch(err => res.status(400).json(`${err}`));

});

// all account Login authenticated

69
userRouter.get('/authenticated',passport.authenticate('jwt',{session :
false}),(req,res)=>{
    const {username,role} = req.user;
    res.status(200).json({isAuthenticated : true, user : {username,role}});
});

module.exports = userRouter;

```

### 3 smsRoutes.js

```

const express = require("express");
const router = express.Router();
var ShoutoutClient = require("shoutout-sdk");
var debug = true,
    verifySSL = false;
var apiKey =
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiIzDk5MWJmMC1kNGU0LTExZWMtODY0MS0
zOTQzMUTuZy2I4ZjEiLCJzdWIoIJTSE9VVE9VVF9BUElfVVNFUiIsImlhCI6MTY1MjY4Mzc4MiwiZXhw
IjoxOTY4MzAyOTgyLCJzY29wZXMiOnsiYWN0aXZpdGllcyI6WyJyZWFkIiwid3JpdGUiXSwibWVzc2FnZ
XMiOlsicmVhZCIsIndyaXRlIl0sImNvbnRhY3RzIjpbiInJlYWQiLCJ3cmI0ZSJdfSwic29fdXNlc19pZC
I6IjY4MDE3Iiwiic29fdXNlc19yb2xlijoidXNlcicIsInNvX3Byb2ZpbGUiOijhbGwiLCJzb191c2VyX25
hbWUiOiiLCJzb19hcGlrZXkiOijub25lIn0.Mk21zIIIn-
EmFzwCy3AFM2GZKSuW42pmmqVprx7hI5tw";

```

```

75 router.post("/sms", (req, res) => {
  const data = req.body;
  console.log("Phone number", data.number);
  // Usage

25 var client = new ShoutoutClient(apiKey, debug, verifySSL);
var message = {
  content: { sms: "Hello Your Resevation is Successfull..!" },
  destinations: [data.number],
  26 source: "ShoutDEMO",
  transports: ["SMS"],
};

client.sendMessage(message, (error, result) => {
  if (error) {
    console.error("Error sending message!", error);
  } else {
    console.log("Sending message successful!", result);
  }
});
});

36 module.exports = router;

```

## userRoutes.js

```

const express = require("express");
const router = express.Router();
const User = require("../model/user");

//register

router.post("/register", async (req, res) => {
  const { name, email, password, country, phonenumber } = req.body;

  const newUser = new User({ name, email, country, password, phonenumber });

  try {
    newUser.save();
    res.send("User Registered successfully");
  } catch (error) {
    return res.status(400).json({ message: error });
  }
});

```

```
    }
});

//Login
17
router.post("/login", async (req, res) => {
  const { email, password } = req.body;

  try {
    const user = await User.find({ email, password });

    if (user.length > 0) {
      const currentUser = {
        name: user[0].name,
        email: user[0].email,
        phononenumber: user[0].phononenumber,
        isAdmin: user[0].isAdmin,
        _id: user[0]._id,
      };
      res.send(currentUser);
    } else {
      return res.status(400).json({ message: " Login Failed" });
    }
  } catch (error) {
    return res.status(400).json({ message: "error login" });
  }
});

20
router.get("/getAllUsers", async (req, res) => {
  try {
    const users = await User.find({});
    res.send(users);
  } catch (error) {
    return res.status(400).json({ message: error });
  }
});

module.exports = router;
```

## app.js (server)

```
const express = require("express");
const app = express();

const cors = require("cors");
const mongoose = require("mongoose");
const bodyParser = require("body-parser");
const cookieParser = require("cookie-parser");

app.use(cookieParser());
app.use(express.json());
app.use(cors());

mongoose.connect(
  "mongodb+srv://minosh:minosh@cluster0.u1yxx.mongodb.net/hotelsystem?retryWrites=true&w=majority"
);

mongoose.connection.on("error", (err) => {
  console.log("connection failed ");
});

mongoose.connection.on("connected", (connected) => {
  console.log("connected with database ");
});

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

const admin = require("./routes/User");
app.use("/user", admin);

const roomRoutes = require("./routes/roomRoutes");
app.use("/api/rooms", roomRoutes);

const userRoutes = require("./routes/userRoutes");
app.use("/api/User", userRoutes);
```

```
const bookingsRoute = require("./routes/bookingRoutes");
app.use("/api/bookings", bookingsRoute);

const smsRoutes = require("./routes/smsRoutes");
app.use("/smsroute", smsRoutes);

module.exports = app;
```

## passport.js

```
31
const passport = require('passport');
const LocalStrategy = require('passport-local').Strategy;
const JwtStrategy = require('passport-jwt').Strategy;
const User = require('./model/Admin');

const cookieExtractor = req =>{
    let token = null;
    if(req && req.cookies){
        token = req.cookies["access_token"];
    }
    return token;
}

//authorization
passport.use(new JwtStrategy({
    jwtFromRequest : cookieExtractor,
    secretOrKey : "Minosh"
},(payload,done)=>{

    User.findById({_id : payload.sub},(err,user)=>{
        if(err)
            return done(err,false);
        if(user)
            return done(null,user);
        else
    })
})
```

```
        return done(null, false);
    });
}

//authenticated local strategy using user name and password
63 passport.use(new LocalStrategy((username, password, done)=>{

    User.findOne({username}, (err, user)=>{
        //something went wrong with database
        31 if(err)
            return done(err);

        // if no user exist
        if(!user)
            return done(null, false);

        // check if password is correct
        user.comparePassword(password, done);
    });
});
});
```

### 3 Server.js

```
const http = require('http');
const app = require('./app');

const server = http.createServer(app);

server.listen(5000, console.log('app is running'));
```

# Web Client

## Package.JSON

```
{  
  "name": "frontend",  
  "version": "0.1.0",  
  "private": true,  
  "dependencies": {  
    "@react-google-maps/api": "^2.10.2",  
    "@testing-library/jest-dom": "^5.16.4",  
    "@testing-library/react": "^13.1.1",  
    "@testing-library/user-event": "^13.5.0",  
    "antd": "^4.20.4",  
    "aos": "^2.3.4",  
    "axios": "^0.26.1",  
    "bootstrap": "^4.6.1",  
    "browser-router": "^0.2.0",  
    "cors": "^2.8.5",  
    "date-fns": "^2.28.0",  
    "daterangepicker": "^3.1.0",  
    "dayjs": "^1.10.5",  
    "jquery": "^3.6.0",  
    "moment": "^2.29.3",  
    "popper.js": "^1.16.1",  
    "react": "^18.0.0",  
    "react-bootstrap": "^1.6.5",  
    "react-date-range": "^1.4.0",  
    "react-datepicker": "^4.7.0",  
    "react-dom": "^18.0.0",  
    "react-phone-number-input": "^3.1.52",  
    "react-range-picker": "^1.0.1",  
    "react-redux": "^7.2.4",  
    "react-responsive-carousel": "^3.2.18",  
  }  
}
```

```
    "react-router": "^5.1.2",
    "react-router-dom": "^5.1.2",
    "react-scripts": "5.0.1",
    "react-spinners": "^0.11.0",
    "react-stripe-checkout": "^2.6.3",
    "react-switch": "^6.0.0",
    "redux": "^4.1.0",
    "scrollmagic": "^2.0.8",
    "space": "^0.22.2",
    "sweetalert2": "^11.0.16",
    "sweetalert2-react": "^0.8.3",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "proxy": "http://localhost:5000"
}
```

10

## Index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/logo.png" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo.png" />

    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

    <title>Sri </title>
    <link rel="stylesheet"
        href="https://use.fontawesome.com/releases/v5.11.2/css/all.css" />18
    <link rel="stylesheet"
        href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap" />43
    <link rel="stylesheet" href="node_modules/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="node_modules/bootstrap/css/mdb.min.css">
    <link rel="stylesheet" href="node_modules/bootstrap/css/style.css">
  </head>
  <body>
    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet" integrity="sha384-wvfXpqpZZVQGK6TAh5PVlGofQNHSoD2xbE+QkPxCAF1NEvoEH3Sl0sibVcOQVnN" />19
    <crossorigin="anonymous">
      <noscript>You need to enable JavaScript to run this app.</noscript>
      <div id="root"></div>
      <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>13
      <link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" />20
      <crossorigin="anonymous">
        <script src="https://unpkg.com/gijgo@1.9.13/js/gijgo.min.js" type="text/javascript"></script>
    
```

```
<link href="https://unpkg.com/gijgo@1.9.13/css/gijgo.min.css"
rel="stylesheet" type="text/css" />
<script src="https://kit.fontawesome.com/2f503da49d.js"
crossorigin="anonymous"></script>
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
<link href="https://fonts.googleapis.com/css2?family=Jura:wght@700&family=Source+Sans+Pro:wght@300&display=swap" rel="stylesheet">
<link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"
integrity="sha384-eMNCOe7tC1doHpGoWe/6oMVeMDAVTMs2xqW4mwXrXsw0L84Iytr2wi5v2QjrP/xp"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.min.js"
integrity="sha384-cn7l7gDp0eyniUwwAZgrzD06kc/tftFf19T0As2zVinnD/C7E91j9yyk5//jjpt/"
crossorigin="anonymous"></script>
<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

<script src="https://kit.fontawesome.com/2f503da49d.js"
crossorigin="anonymous"></script>
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
<link href="https://fonts.googleapis.com/css2?family=Jura:wght@700&family=Source+Sans+Pro:wght@300&display=swap" rel="stylesheet">
<link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"
integrity="sha384-eMNCOe7tC1doHpGoWe/6oMVeMDAVTMs2xqW4mwXrXsw0L84Iytr2wi5v2QjrP/xp"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.min.js"
integrity="sha384-cn7l7gDp0eyniUwwAZgrzD06kc/tftFf19T0As2zVinnD/C7E91j9yyk5//jjpt/"
crossorigin="anonymous"></script>
```

```

<link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
<path d="M15.502 1.94a.5.5 0 0 1 0 .706L14.459 3.691-2-2L13.502.646a.5.5 0
0 1 .707 0L1.293 1.293z-1.75 2.456-2-2L4.939 9.21a.5.5 0 0 0-.121.196l-.805
2.414a.25.25 0 0 0 .316.316L2.414-.805a.5.5 0 0 0 .196-.12L6.813-6.814z"/>
<path fill-rule="evenodd" d="M1 13.5A1.5 1.5 0 0 0 2.5 15h11a1.5 1.5 0 0 1
1.5-1.5v-6a.5.5 0 0 0-1 0v6a.5.5 0 0 1-.5.5h-11a.5.5 0 0 1-.5-.5v-11a.5.5 0 0 1
.5-.5H9a.5.5 0 0 0 0-1H2.5A1.5 1.5 0 0 0 1 2.5v11z"/>
</svg>
12
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"
integrity="sha384-
7+zCNj/IqJ95Wo16oMtfSkBZ9ccEh31e0z1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>

34
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
" integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

</body>
</html>

```

## 17 Index.js

```

import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import AuthProvider from './Context/AuthContext';

```

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <AuthProvider>
    <App />
  </AuthProvider>
);
```

## 77 App.js

```
import React, { useState, useEffect } from "react";
import Register from "./Components/Register";
import Sysadmin from "./Components/Sysadmin";
import Admin from "./Components/Admin";
import PrivateRoute from "./hocs/PrivateRoute";
import UnPrivateRoute from "./hocs/UnPrivateRoute";
import Login from "./Components/Login";
import BookingRoom from "./Components/BookingRoom";
import Homescreen from "./Components/MainPage";
import Mainpage from "./Components/Home";
import NavBar from "./Components/NavBar";
import UserLogin from "./Components/UserLogin";
import RegisterUser from "./Components/RegisterUser";
import UserProfile from "./Components/UserProfile";
import AdminNavBar from "./Components/AdminNavBar";
import AllReservationAdmin from "./Components/AllReservationAdmin";
import ManageAllRooms from "./Components/ManageAllRooms";
import AddRooms from "./Components/AddRooms";
import ManageAllUsers from "./Components/ManageAllUsers";
import UpdateRooms from "./Components/UpdateRooms";

import axios from "axios";
import { BrowserRouter as Router, Route } from "react-router-dom";

function App() {
```

```
const [room, setPosts] = useState([ ]);

useEffect(() => {
  axios
    .get("http://localhost:5000/api/rooms/getallrooms/")
    .then((res) => setPosts(res.data))
    .catch((error) => console.log(error));
});

return (
  <div className="App">
    <NavBar />
    <Router>
      <Route path="/" exact component={Homescreen} />
      <Route path="/home" component={Mainpage} />
      <Route path="/register" component={RegisterUser} />
      <Route path="/AdminNarBar" component={AdminNarBar} />
      <Route path="/login" component={UserLogin} />
      <Route path="/adminlogin" component={Login} />
      <Route path="/adminregister" component={Register} />
      <Route path="/profile" component={UserProfile} />
      <Route
        path="/update/:id"
        render={(props) => <UpdateRooms {...props} posts={room} />}
      />16
      <PrivateRoute path="/Admin" roles={[ "admin" ]} component={Admin} />
      <PrivateRoute
        path="/allReservation"
        roles={[ "admin" ]}
        component={AllReservationAdmin}
      />
      <PrivateRoute
        path="/adminroom"
        roles={[ "admin" ]}
        component={ManageAllRooms}
      />16
      <PrivateRoute
        path="/ManageAllUsers"
        roles={[ "admin" ]}
        component={ManageAllUsers}
      />
      <Route path="/AddRooms" component={AddRooms} />
      <Route
```

```
        path="/booking/:roomid/:fromdate/:todate"
        component={BookingRoom}
      />
      <PrivateRoute
        path="/sysadmin"
        roles={[ "sysadmin" ]} 
        component={Sysadmin}
      />
    </Router>
  </div>
);
}

export default App;
```

## Components

### BookingRoom.js

```
import React, { useEffect , useState } from 'react' 40
import axios from "axios";
import Swal from 'sweetalert2';
import Error from "../Components/Error";
import Loader from "../Components/Loader";
import Success from '../Components/Success';
import StripeCheckout from 'react-stripe-checkout'71
import 'react-phone-number-input/style.css'
import PhoneInput from 'react-phone-number-input'
import moment from "moment"
import AOS from 'aos';

AOS.init();
AOS.refresh()

function BookingRoom({match}) {

  const smsfunc= (e) => {
```

```

e.preventDefault();
let number = JSON.parse(localStorage.getItem('currentUser')).phonenumber
console.log("Phone number ",number)
let data = {
    number:number
}
axios.post("http://localhost:5000/smsroute/sms/",data)
.then((data)=>{
    console.log(data);
    alert("Success");
})
.catch((err)=>{
    console.log(err);
})
}

const[loading, setloading]=useState(true);
const[error, seterror]=useState(false)
const[success, setsuccess]=useState(false)
const[room , setroom]=useState()
const roomid=match.params.roomid
const fromdate=moment(match.params.fromdate , 'DD-MM-YYYY')14
const todate=moment(match.params.todate,'DD-MM-YYYY')
const totalDays = moment.duration(todate.diff(fromdate)).asDays()+1
const [totalAmount , settotalAmount]=useState()
const [phonenumber , setphonenumber]=useState()

useEffect(async() => {

try {
    setloading(true);
    const data = await (await
axios.post("http://localhost:5000/api/rooms/getRoomById" , {roomid}).data;
        console.log(data);
        setroom(data);
        setphonenumber(data);
        setloading(false);
        settotalAmount(data.rentPerDay * totalDays)
    } catch (error) {
        console.log(error);
        setloading(false);
    }
}, [])

```

```
async function tokenHander(token) {  
  console.log(token);  
  const bookingDetails ={  
    token ,  
    user : JSON.parse(localStorage.getItem('currentUser')),  
    room ,  
    fromdate,  
    todate,  
    phonenumber,  
    totalDays,  
    totalAmount  
  }  
  
  try {  
    setLoading(true);  
    const result = await axios.post('http://localhost:5000/api/bookings/bookroom' , bookingDetails)  
    setLoading(false)  
    Swal.fire('Congrats' , 'Your Room has booked successfully' ,  
'success').then(result=>{  
      window.location.href='/profile'  
    })  
  } catch (error) {  
    console.log(error);  
    setLoading(false)  
    Swal.fire('Oops' , 'Something went wrong , please try later' ,  
'error')  
  }  
  
  return (  
    <div className='m-5'>  
      {loading ? (<Loader/>) : error ? (<Error/>) : (  
        <div className="row p-3 mb-5 bs" data-aos='flip-right'  
duration='2000'>  
          <div className="col-md-6 my-auto">
```

```
<div>
  <h1> {room.name}</h1>
</div>
<div className="col-md-6 text-right">
  <div>
    <h1><b>Booking Details</b></h1>

    <p><b>Name</b> :<br/>
{JSON.parse(localStorage.getItem('currentUser')).name}</p>
    <p><b>From Date</b> : {match.params.fromdate}</p>
    <p><b>To Date</b> : {match.params.todate}</p>
    <p><b>phonenumber</b> :<br/>
{JSON.parse(localStorage.getItem('currentUser')).phonenumber}</p>
  </div>

  <div className='mt-5'>
    <h1><b>Amount</b></h1>

    <p>Total Days : <b>{totalDays}</b></p>
    <p>Rent Per Day : <b>{room.rentPerDay}</b></p>
    <p>phoneNumber : <b>{room.phoneNumber}</b></p>
    <h1><b>Total Amount : {totalAmount} /-</b></h1>

    <StripeCheckout
      amount={totalAmount*100}
      currency='LKR'
      token={tokenHandler}

      stripeKey='pk_test_51KyZL6FeoRdqg6f816PBNMT9VvvDUCbwgUS6pvLiK2kQ6Mvx3q4B0FTQc9ENZ
      pb0IeCH2a3GABCaxBbTStcyvocg00TK9Bbc8z' >
```

```
    ^
    <button className='btn btn-primary'
onClick={(e)=>smsfunc(e)}>Pay Now</button>

    </StripeCheckout>
^
    </div>

    </div>
)
}

export default BookingRoom;
```

## Home.js

```
import React, { useState, useEffect } from "react";
import { Calendar } from "react-date-range";
import { DateRangePicker } from "react-date-range";
import "react-date-range/dist/styles.css"; // main style file
import "react-date-range/dist/theme/default.css"; // theme css file
import "react-datepicker/dist/react-datepicker.css";

import moment from "moment";
56
import axios from "axios";
import Loader from "../Components/Loader";
import Room from "../Components/DisplayAllRooms";
import { DatePicker, Space } from "antd";

const { RangePicker } = DatePicker;
function Home() {
  const [reservation, setreservation] = useState([]);
```

```
const [hotels, sethotels] = useState([]);
const [fromdate, setfromdate] = useState("");
const [todate, settodate] = useState("");
const [loading, setloading] = useState(false);
const [searchkey, setsearchkey] = useState("");
const [type, setttype] = useState("all");

function filterByDate(dates) {  
    setfromdate(moment(dates[0]).format("DD-MM-YYYY"));  
    settodate(moment(dates[1]).format("DD-MM-YYYY"));  
  
    var temp = [];  
    for (var room of hotels) {  
        var availability = false;  
  
        for (var booking of room.currentbookings) {  
            if (room.currentbookings.length) {  
                if (  
                    !moment(moment(dates[0]).format("DD-MM-YYYY")).isBetween(  
                        booking.fromdate,  
                        booking.todate  
                    ) &&  
                    !moment(moment(dates[1]).format("DD-MM-YYYY")).isBetween(  
                        booking.fromdate,  
                        booking.todate  
                    )  
                ) {  
                    if (  
                        moment(dates[0]).format("DD-MM-YYYY") !== booking.fromdate &&  
                        moment(dates[0]).format("DD-MM-YYYY") !== booking.todate &&  
                        moment(dates[1]).format("DD-MM-YYYY") !== booking.fromdate &&  
                        moment(dates[1]).format("DD-MM-YYYY") !== booking.todate  
                    ) {  
                        availability = true;  
                    }  
                }  
            }  
        }  
        if (availability || room.currentbookings.length == 0) {  
            temp.push(room);  
        }  
        setreservation(temp);  
    }  
}
```

```
28
useEffect(async () => {
  try {
    setLoading(true);
    const rooms = await (
      await axios.get("http://localhost:5000/api/rooms/getAllRooms")
    ).data;
    console.log(rooms);
    setReservation(rooms);
    setHotels(rooms);
    setLoading(false);
  } catch (error) {
    console.log(error);
    setLoading(false);
  }
}, []);

function filterBySearch() {
  const update = hotels.filter((room) =>
    room.name.toLowerCase().includes(searchKey)
  );
  setReservation(update);
}

function filterByType(e) {
  setType(e);
  if (e !== "all") {
    const update = hotels.filter((room) =>
      room.type.toLowerCase().includes(e.toLowerCase())
    );
    setReservation(update);
  } else {
    setReservation(hotels);
  }
}

return (
  <div>
    <div>
      <div>
        <div>
          <RangePicker
            id="datepicker"
            type="date"
            onChange={filterByDate}
            format="DD-MM-YYYY"

```

```
        className="m-2"
    />
</div>

9 <div className="col-md-4">
    <input
        type="text"
        className="form-control i2 m-2"
        placeholder="find Rooms"
        value={searchkey}
        onKeyUp={filterBySearch}
        onChange={(e) => {
            setsearchkey(e.target.value);
        }}
    />
</div>
<div className="col-md-4">
    <select
        className="form-control m-2"
        value={type}
        onChange={(e) => {
            filterByType(e.target.value);
        }}
    >
        84
        <option value="all">All</option>
        <option value="delux">AC rooms</option>
        <option value="non-delux">Non AC Room</option>
        <option value="delux">Luxury Room</option>
        <option value="non-delux">Normal Room</option>
    </select>
</div>
</div>
</div>

<div className="row justify-content-center">
    {loading ? (
        <Loader />
    ) : (
        reservation.map((room) => {
            return (
                <div className="col-md-8" data-aos="zoom-in">
                    <Room room={room} fromdate={fromdate} todate={todate} />
                </div>
            );
        })
    )}

```

```
        )}
    </div>
</div>
);
}

export default Home;
```

## MainPage.js

```
import React from "react";
import Map from "./Map";
import { Link } from "react-router-dom";

function MainPage() {
    return (
        <div className="">
            <div className="landing row justify-content-center text-center">
                <div
                    className="col-md-9 my-auto"
                    style={{ borderRight: "8px solid white" }}
                >
                    <br><br>
                    <h2 style={{ color: "black", fontSize: "90px" }} data-aos="zoom-in">
                        hotel bo0king
                    </h2>

                    <img
                        class="card-img-top"
                        src={require("./qq.jpg")}
                        height={450}
                        width={-10}
                        alt="Card image cap"
                    />

                    <h1 data-aos="zoom-out">"The bestplace to booking your room" </h1>
                    <Link to="/home">
                        <button className="btn btn-primary">Get Started</button>
                    </Link>

                    <br><br>
                </div>
            </div>
        </div>
    );
}

export default MainPage;
```

```

<br></br>
<br></br>
<Link to="/adminlogin">
  <button className="btn btn-primary">Admin login</button>
</Link>

<br></br>

<Map />
</div>
</div>
</div>
);
}

export default MainPage;

```

## Map.js

```

import React from "react";
import { GoogleMap, LoadScript, Marker } from "@react-google-maps/api";

const containerStyle = {
  marginTop: "220px",
  width: "1210px",
  height: "600px",
};

const position = {
  lat: 6.93197,
  lng: 79.85775,
};

const onLoad = (marker) => {
  console.log("marker: ", marker);
};

function Map() {
  return (
    <LoadScript googleMapsApiKey="AIzaSyBAGsTlZT7fLRb_IVRmMFSXf-qbE1nnIKk">
      <GoogleMap mapContainerStyle={containerStyle} center={position} zoom={10}>

```

```
        <Marker onLoad={onLoad} position={position} />
      </GoogleMap>
    </LoadScript>
  );
}

export default React.memo(Map);
```

## DisplayAllRooms.js 9

```
import React, { useState, useEffect } from "react";
import { Modal, Carousel } from "react-bootstrap";
import { Link } from "react-router-dom";

function DisplayAllRooms({ room, fromdate, todate }) {
  const [show, setShow] = useState(false);
  const handleClose = () => setShow(false);
  const handleShow = () => setShow(true);

  return (
    <div className="row m-3 p-3 bs">
      <div className="col-md-4">
        <img src={room.imageUrl[0]} className="img-fluid" />
      </div>
      <div className="col-md-8">
        <b>Room Name</b>
        <h2>{room.name}</h2>
        <b>Hotel facilities</b>
        <p>
          {" "}
          24 hour front desk   ,Business Center,   ,Fitness Center   ,Lounge/Bar   ,Internet Access - Free in room   , Contactless check-in and check-out
        </p>
        <p>
          <b>Type : {room.type}</b>
        </p>
      </div>
    </div>
  );
}

export default DisplayAllRooms;
```

```

<p>
  <b>rent Per Day : {room.rentPerDay}</b>
</p>

<p>
  <b>phoneNumber : {room.phoneNumber}</b>
</p>

<div style={{ float: "right" }}>
  {fromdate && todate && (
    <Link to={`/booking/${room._id}/${fromdate}/${todate}`}>
      <button className="btn btn-dark m-2">Book </button>
    </Link>
  )}
</div>
<Link to="/home">
  <button className="btn btn-primary">view </button>
</Link>
</div>
</div>
);
}
export default DisplayAllRooms;

```

## UserLogin.js

```

import React, { useState, useEffect } from "react";
import { useDispatch, useSelector } from "react-redux";
import axios from "axios";
import Error from "../Components/Error";
import Loader from "../Components/Loader";
import Success from "../Components/Success";

export default function UserLogin() {
  const [email, setemail] = useState("");
  const [password, setpassword] = useState("");
  const [loading, setloading] = useState(false);
  const [error, seterror] = useState(false);
  const [success, setsuccess] = useState(false);

  useEffect(() => {
    if (localStorage.getItem("currentUser")) {

```

```
        window.location.href = "/";
    }
}, []);
```

```
async function login() {
    const user = {
        email,
        password,
    };
    try {
        setloading(true);
        const result = await (
            await axios.post("http://localhost:5000/api/User/login", user)
        ).data;
        localStorage.setItem("currentUser", JSON.stringify(result));
        window.location.href = "/home";
    } catch (error) {
        seterror(true);
        setloading(false);
        console.log(error);
    }
}

return (
    <div className="login">
        <div className="row justify-content-center mt-12">
            <div className="col-md-5 mt-5 text-left shadow-lg p-3 mb-12 ">
                <h1 className="text-center m-2">Customer Login</h1>
                <br><br>
                {loading && <Loader />}
                {error && <Error error="Invalid Credentials" />}
                {success && <Success success="User Login Successfully" />}
                <div>
                    <input
                        required
                        type="text"
                        placeholder=" enter your email"
                        className="form-control mt-1"
                        value={email}
                        onChange={(e) => {
                            setemail(e.target.value);
                        }}
                    />
                    <br><br>
                    <input
```

```

        type="text"
        placeholder="enter your password"
        className="form-control mt-1"
        value={password}
        required
        onChange={(e) => {
          setPassword(e.target.value);
        }}
      />

      <button onClick={login} className="btn btn-success mt-3 mb-6">
        login
      </button>
      <br />
      <a style={{ color: "black" }} href="/register" className="mt-2">
        {" "}
        Register to the systems
      </a>
    </div>
  </div>
</div>
);
}

```

## UserProfile.js

```

import React, { useEffect, useState } from "react";
import { Tabs } from "antd";
import axios from "axios";
import Swal from "sweetalert2";
import Error from "../Components/Error";
import Loader from "../Components/Loader";
import Success from "../Components/Success";
import { Tag, Divider } from "antd";

const { TabPane } = Tabs;

const user = JSON.parse(localStorage.getItem("currentUser"));

```

```
function UserProfile() {
  return (
    <div className="mt-5 ml-3">
      <Tabs defaultActiveKey="1">
        <button className="btn btn-danger" tab="My Account" key="1">
          <div className="row">
            <div className="col-md-6 bs m-2 p-3">
              <h4>my Name : {user.name}</h4>
              <h4>my Email : {user.email}</h4>
            </div>
          </div>
        </button>
        <TabPane tab="Bookings" key="2">
          <h1>
            <MyReservation />
          </h1>
        </TabPane>
      </Tabs>
    </div>
  );
}

export default UserProfile;

export const MyReservation = () => {
  const [mybookings, setmybookings] = useState([]);
  const [loading, setloading] = useState(false);
  const [error, seterror] = useState(false);
  const [success, setsuccess] = useState(false);
  useEffect(async () => {
    try {
      setloading(true);
      const data = await (
        await axios.post("http://localhost:5000/api/bookings/getuserbookings", {
          userid: JSON.parse(localStorage.getItem("currentUser"))._id,
        })
      ).data;
      setmybookings(data);
      setloading(false);
    } catch (error) {
      setloading(false);
      seterror(true);
    }
  }, []);
}
```

```
async function cancelBooking(bookingid, roomid) {
  try {
    setloading(true);
    const result = await axios.post(
      "http://localhost:5000/api/bookings/cancelbooking",
      { bookingid: bookingid, userid: user._id, roomid: roomid }
    );
    setloading(false);
    Swal.fire(
      "Congrats",
      "Your Room has cancelled successfully",
      "success"
    ).then((result) => {
      window.location.href = "/profile";
    });
  } catch (error) {
    Swal.fire(
      "Congrats",
      "Your Room has cancelled successfully",
      "success"
    ).then((result) => {
      window.location.href = "/profile";
    });
    setloading(false);
  }
}

return (
  <div>
    {loading ? (
      <Loader />
    ) : error ? (
      <Error />
    ) : (
      mybookings.map((booking) => {
        return (
          <div className="row">
            <div className="col-md-6 my-auto">
              <div className="bs m-1 p-2">
                <h4>{booking.room}</h4>

                <h6>TransactionId : {booking.transactionId}</h6>
                <h6>
                  <b>Check In : </b>
                
```

```
        {booking.fromdate}
    </h6>
    <h6>
        <b>Check Out : </b>
        {booking.todate}
    </h6>
    <h6>
        <b>Amount : </b> {booking.totalAmount}
    </h6>
    <h6>
        <b>Status</b> :{" "}
        {booking.status == "booked" ? (
            <Tag color="green">Confirmed</Tag>
        ) : (
            <Tag color="red">Cancelled</Tag>
        )}
    </h6>
    <div className="text-right">
        {booking.status == "booked" && (
            <button
                className="btn btn-primary"
                onClick={() =>
                    cancelBooking(booking._id, booking.roomid)
                }
            >
                Cancel Reservations
            </button>
        )}
    </div>
    </div>
    </div>
);
})
);
</div>
);
};
```

## NavBar.js

```
import React from "react";
import { Link } from "react-router-dom";
function NavBar() {

    function logout() {
        localStorage.removeItem('currentUser')
        window.location.href='/login'
    }

    return (
        <div>
            <nav class="navbar bg-primary navbar-expand-lg">
                <a class="navbar-brand">
                    Hotel Reservation
                </a>

                <button
                    class="navbar-toggler"
                    type="button"
                    data-toggle="collapse"
                    data-target="#navbarNav"
                    aria-controls="navbarNav"
                    aria-expanded="false"
                    aria-label="Toggle navigation"
                >
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarNav">

                    <ul class="navbar-nav ml-auto">

{localStorage.getItem('currentUser') ? (
    <div class="dropdown mr-5">
        <button class="btn btn-secondary dropdown-toggle" type="button" aria-
haspopup="true" aria-expanded="false">
            <i class="fa fa-user" aria-hidden="true"></i>
            {JSON.parse(localStorage.getItem('currentUser')).name} </button>
<div id="myDropdown" class="dropdown-content" >

```

```

    <a class="dropdown-item" href="/profile">Profile</a>
    <a class="dropdown-item" href="#" onClick={logout}>Logout</a>

</div>

) : (
  <>

<li class="nav-item active">
  <a class="navbar-item" href="/register">
    Register
  </a>
</li>

<li class="nav-item">
  <a class="navbar-item" href="/login">
    Login As Customer
  </a>
</li>
</>
)>

</ul>
</div>
</nav>
</div>
);

}

export default NavBar;

```

## Loader.js

```

import React, { useState } from "react";
import { css } from "@emotion/react";
import FadeLoader from "react-spinners/FadeLoader";
function Loader() {
  return (

```

```
<div className="sweet-loading text-center" style={{ marginTop: "100px" }}>
  <FadeLoader color="#C43D15" loading={true} css="" size={50} />
</div>
);

export default Loader;
```

## Success.js

```
67
import React from "react";

export default function Success({ success }) {
  return (
    <div>
      <div className="alert alert-success" role="alert">
        {success}
      </div>
    </div>
  );
}
```

## Error.js

```
59
import React from "react";

export default function Error({ error }) {
  return (
    <div>
      <div className="alert alert-danger" role="alert">
        {error}
      </div>
    </div>
  );
}
```

## Message.js

```
import React from "react";

const getClass = (props) => {
  let baseClass = "alert ";
  if (props.message.msgError) baseClass = baseClass + "alert-danger";
  else baseClass = baseClass + "alert-primary";
  return baseClass + " text-center";
};

const Message = (props) => {
  return (
    <div className={getClass(props)} role="alert">
      {props.message.msgBody}
    </div>
  );
};

export default Message;
```

## Database connection

```
7
const mongoose = require('mongoose');

mongoose.connect('mongodb+srv://minosh:minosh@cluster0.u1yxx.mongodb.net/hotelsystem?retryWrites=true&w=majority');

mongoose.connection.on('error', err=>{
  35
  console.log("connection failed ");
});

mongoose.connection.on('connected' , connected=>{
```

```
    console.log('connected with database ');
});
```

## Desktop Application (Admin)

### Admin.js

```
import React, { useEffect, useState } from "react";
import { Link } from "react-router-dom";
import axios from "axios";

const Admin = () => (
  <div className="container">
    <br></br>
    <br></br>
    <h2 style={{ color: "black", fontSize: "70px" }} data-aos="zoom-in">
      Welcome to System Admin Page
    </h2>
    <h1> </h1>
    <br></br>
    <br></br>
    <Link to="/allReservation">
      <button type="button" class="btn btn-primary">
        All Reservation
      </button>
    </Link>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <Link to="/adminroom">
      <button type="button" class="btn btn-primary">
        manage Room
      </button>
    </Link>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <Link to="/ManageAllUsers">
      <button type="button" class="btn btn-primary">
```

```

    Manage All Users
    </button>
</Link>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<Link to="/adminregister">
    <button type="button" class="btn btn-primary">
        Add Admin (Admin register)
    </button>
</Link>
</div>
);

export default Admin;

```

## 1 AdminNarBar.js

```

import React, { useContext } from "react";
import { Link } from "react-router-dom";
import AuthService from "../Services/AuthService";
import { AuthContext } from "../Context/AuthContext";

const AdminNarBar = (props) => {
    const { isAuthenticated, user, setIsAuthenticated, setUser } =
        useContext(AuthContext);

    //Logout button create
    const onClickLogoutHandler = () => {
        AuthService.logout().then((data) => {
            if (data.success) {
                setUser(data.user);
                setIsAuthenticated(false);
            }
        });
    };

    const unauthenticatedNavBar = () => {
        return (
            <>
                <Link to="/">
                    <li className="nav-item nav-link">Home</li>
                </Link>
                <Link to="/adminlogin">
                    <li className="nav-item nav-link">Login</li>
                </Link>

```

```
<Link to="/adminregister">
  <li className="nav-item nav-link">Register</li>
</Link>
</>
);
};

const authenticatedNavBar = () => {
  return (
    <>
      <Link to="/">
        <li className="nav-item nav-link">Home</li>
      </Link>

      <Link to="/to">
        <li className="nav-item nav-link">View Hotel Rooms</li>
      </Link>

      {user.role === "admin" ? (
        <Link to="/admin">
          <li className="nav-item nav-link">Admin</li>
        </Link>
      ) : null}
      {user.role === "sysadmin" ? (
        <Link to="/sysadmin">
          <li className="nav-item nav-link">Admin</li>
        </Link>
      ) : null}

      <button
        type="button"
        className="btn btn-link nav-item nav-link"
        onClick={onClickLogoutHandler}
      >
        Logout
      </button>
    </>
  );
};

return (
  <div>
    <nav className="navbar navbar-expand-lg navbar-dark bg-dark">
      <Link className="navbar-brand" href="#">
        Hotel Reservation{" "}
      </Link>
    </nav>
  </div>
);
```

```

        </Link>

        <div class="collapse navbar-collapse" id="navbarText">
            <ul class="navbar-nav mr-auto">
                {!isAuthenticated ? unauthenticatedNavBar() : authenticatedNavBar()}
            </ul>
        </div>
    </nav>
</div>
);

};

export default AdminNavBar;

```

## AllReservationAdmin.js

```

import React, { useState, useEffect } from "react";
import axios from "axios";
import Swal from "sweetalert2";
import Error from "../Components/Error";
import Loader from "../Components/Loader";
import Success from "../Components/Success";
import { Link } from "react-router-dom";

function AllReservationAdmin() {
    const [bookings, setbookings] = useState([]);
    const [loading, setloading] = useState(false);
    const [error, seterror] = useState(false);
    const [success, setsuccess] = useState(false);
    useEffect(async () => {
        try {
            setloading(true);
            const data = await (
                await axios.get("http://localhost:5000/api/bookings/getallbookings")
            ).data;
            setbookings(data);
            setloading(false);
        } catch (error) {
            setloading(false);
            seterror(true);
        }
    }, []);
    return (

```

```
<div className="col-md-11">
  <h1>All Reservation</h1>
  {loading ? (
    <Loader />
  ) : error ? (
    <Error />
  ) : (
    <div>
      <Link to="/Admin">
        <button type="button" class="btn btn-primary">
          back to admin
        </button>
      </Link>
      <br><br>
```

```
11
<table className="table table-bordered table-dark">
  <thead className="bs">
    <tr>
      <th>Booking Id</th>
      <th>Userid</th>
      <th>Room</th>
      <th>From</th>
      <th>To</th>
      <th>Status</th>
    </tr>
  </thead>
  <tbody>
    {bookings.map((booking) => {
      return (
        <tr>
          <td>{booking._id}</td>
          <td>{booking.userid}</td>
          <td>{booking.room}</td>
          <td>{booking.fromdate}</td>
          <td>{booking.todate}</td>
          <td>{booking.status}</td>
        </tr>
      );
    })}
  </tbody>
</table>
</div>
)}
```

```
</div>
);
}

export default AllReservationAdmin;
```

## UpdateRooms.js 15

```
import React, { useState, useEffect } from "react";
import axios from "axios";
import { Link } from "react-router-dom";

const UpdateRooms = (props) => {
  const [name, setroom] = useState("");
  const [rentPerDay, setrentperday] = useState();
  const [description, setdescription] = useState("");
  const [phoneNumber, setphonenumber] = useState("");
  const [type, settype] = useState("");

  const changeOnClick = (f) => {
    //f.preventDefault();

    const updateroom = {
      name,
      rentPerDay,
      description,
      phoneNumber,
      type,
    };

    setroom("");
    setrentperday("");
    setdescription("");
    setphonenumber("");
    settype("");

    axios
      .put(
        `http://localhost:5000/api/rooms/update/${props.match.params.id}`,
        updateroom
      )
      .then((res) => alert(" Updated!")(res.data))
      .catch((err) => {
```

```
        console.log(err);
    });
}

useEffect(() => {
    axios
        .put(
            `http://localhost:5000/api/rooms/getRoombyid/${props.match.params.id}`)
        )

        .then((res) => [
            setroom(res.data.name),
            setrentperday(res.data.rentPerDay),
            setdescription(res.data.description),
            setphonenumber(res.data.phoneNumber),
            setttype(res.data.type),
        ])
        74
        .catch((error) => console.log(error));
}, []);

return (
    <div className="container">
        <h1>Update rooms </h1>

        <form onSubmit={changeonClick} encType="multipart/form-data">
            <div class="mb-3">
                <label htmlFor="courseID" class="form-label">
                    name room{" "}
                </label>
                <input
                    type=""
                    class="form-control"
                    placeholder=""
                    onChange={(e) => setroom(e.target.value)}
                    value={name}
                />
            </div>

            <div class="mb-3">
                <label htmlFor="courseName" class="form-label">
                    rent per day
                </label>
                <input
                    type=""

```

```
        class="form-control"
        placeholder=""
        onChange={(e) => setrentperday(e.target.value)}
        value={rentPerDay}
      />
    </div>

    <div class="form-outline">53
      <label htmlFor="courseType" class="form-label">
        description
      </label>
      <input
        type=""
        class="form-control"
        placeholder=""
        onChange={(e) => setDescription(e.target.value)}
        value={description}
      />
    </div>

    <div class="form-group">
      <label htmlFor="description">phone number</label>
      <textarea
        className="form-control"
        rows="5"
        onChange={(e) => setPhoneNumber(e.target.value)}
        value={phoneNumber}
      ></textarea>
    </div>

    <div class="row mb-4">
      <div class="col">
        <div class="form-outline">
          <label htmlFor="requirement" class="form-label">
            type
          </label>
          <input
            type="text"
            class="form-control"
            placeholder=""
            onChange={(e) => setType(e.target.value)}
            value={type}
          />
        </div>
      </div>
    </div>
```

```

        </div>

        <div className="container">
            <br><br>
            <button type="submit" class="btn btn-success">
                Update Course
            </button>
            <br><br>
            <br><br>
            <Link to="/Admin">
                <button type="button" class="btn btn-primary">
                    back to admin
                </button>
            </Link>
        </div>
    </form>
    <div align="right">
        <p></p>
    </div>
</div>
);

};

export default UpdateRooms;

```

## ManageAllUsers.js

```

import React, { useState, useEffect } from "react";
import axios from "axios";
import Swal from "sweetalert2";
import Error from "../Components/Error";
import Loader from "../Components/Loader";
import Success from "../Components/Success";
import { Link } from "react-router-dom";

function ManageAllUsers() {
    const [users, setusers] = useState();
    const [loading, setloading] = useState(true);
    useEffect(async () => {
        try {
            const data = await (

```

```
        await axios.get("http://localhost:8290/User/getAllUsers")
    ).data; 5
    setusers(data);
    setloading(false);
} catch (error) {
    console.log(error);
    setloading(false);
}
}, []);
```

**return (**

```
    <div className="row"62>
        {loading && <Loader />}
```

**<div className="col-md-10">**

```
    <br></br>
    <br></br>
    <Link to="/Admin"53>
        <button type="button" class="btn btn-primary">
            back to admin
        </button>
    </Link>
    <br></br>
    <br></br>
    <table className="table table-bordered table-dark">
        <thead className="bs"54>
            <tr>
                <th>Id</th>
                <th>Name</th>
                <th>Email</th>
            </tr>
        </thead>
```

**<tbody>**

```
    {users &&
        users.map((user) => {
            return (
                <tr>
                    <td>{user._id}</td>
                    <td>{user.name}</td>
                    <td>{user.email}</td>
                </tr>
            );
        )});
    </tbody>
```

```
        </table>
      </div>
    </div>
  );
}

4 export default ManageAllUsers;
```

## ManageAllRooms.js

```
import React, { useState, useEffect } from "react";
import axios from "axios";
import Swal from "sweetalert2";
import Error from "../Components/Error";
import Loader from "../Components/Loader";
5 import Success from "../Components/Success";
import { Link } from "react-router-dom";

function ManageAllRooms() {
  const [rooms, setrooms] = useState([]);
  const [loading, setloading] = useState(false);
  const [error, seterror] = useState(false);
  const [success, setsuccess] = useState(false);

  7 const deleteRoom = (id) => {
    axios
      .delete(`http://localhost:5000/api/rooms/${id}`)
      .then((res) => alert(res.data));
  }

  setrooms(rooms.filter((elem) => elem._id !== id));
};

28 useEffect(async () => {
  try {
    setloading(true);
    const data = await (
      await axios.get("http://localhost:5000/api/rooms/getallrooms")
    ).data;
```

```

        setrooms(data);
        setloading(false);
    } catch (error) {
        setloading(false);
        seterror(true);
    }
}, []);
return (
    <div className="col-md-11">
        <br></br>
        <br></br>
        <Link to="/Admin">
            <button type="button" class="btn btn-primary">
                back to admin
            </button>
        </Link>
        <br></br> <br></br> <br></br>
        <Link to="/AddRooms">
            <button type="button" class="btn btn-primary">
                Add Room
            </button>
        </Link>
        <br></br>
        <h1>Rooms</h1>
        {loading ? (
            <Loader />
        ) : error ? (
            <Error />
        ) : (
            <div>
                <table className="table table-bordered table-dark">
                    <thead className="bs">
                        <tr>
                            <th>Room Id</th>
                            <th>Name</th>
                            <th>Type</th>
                            <th>Rent Per day</th>
                            <th>Phone Number</th>
                            <th>delete</th>
                            <th>update</th>
                        </tr>
                    </thead>
                    <tbody>
{rooms.map((room) => {

```

```
        return (
          <tr>
            <td>{room._id}</td>
            <td>{room.name}</td>
            <td>{room.type}</td>
            <td>{room.rentPerDay}</td>
            <td>{room.phoneNumber}</td>
            <td>
              <button
                onClick={() => deleteRoom(room._id)}79
                className="btn btn-danger"
                style={{ marginTop: "10px" }}
              >
                &nbsp;delete
              </button>
            </td>

            <td>
              <Link
                to={`/update/${room._id}`}
                className="btn btn-warning"
                style={{ marginTop: "13px" }}
              >
                <i class="fas fa-edit"></i>&nbsp;Edit{" "}
              </Link>
            </td>
          </tr>
        );
      )})
    </tbody>
  </table>
</div>
);
}

export default ManageAllRooms;
```

## AddRooms.js

```
import axios from "axios";
import React, { useState } from "react";

import Swal from "sweetalert2";
import { Link } from "react-router-dom";

const AddRooms = () => {
  const [name, setroom] = useState("");
  const [rentPerDay, setrentperday] = useState();
  const [description, setdescription] = useState("");
  const [phoneNumber, setphonenumber] = useState("");
  const [type, setttype] = useState("");
  const [image1, setimage1] = useState("");

  const changeOnClick = (f) => {
    //f.preventDefault();

    const addroom = {
      name,
      rentPerDay,
      description,
      phoneNumber,
      type,
      image1,
    };

    axios.post("http://localhost:5000/api/rooms/addRoom", addroom);
    Swal.fire("Congrats", "New Room Added successfully", "success")
      .then((res) => console.log("success"))

      .catch((err) => {
        console.log(err);
      });
  };
  return (
    <div className="container ">
      <br></br> <br></br>
      <center>
        <h1>Add a New room</h1>
        <form onSubmit={changeOnClick} encType="">
          <div className="form-row">
            <div className="form-group col-md-6">
              <label htmlFor="name">room name</label>
              <br />
              <input
```

```
        type="text"
        onChange={(f) => setroom(f.target.value)}
        className="form-control"
        placeholder="name"
      />
    </div>

    <div className="form-group col-md-6">
      <label htmlFor="rentPerDay">rent Per Day</label>
      <br />
      <input
        type="text"
        onChange={(f) => setrentperday(f.target.value)}
        className="form-control"
        placeholder="rent Per Day"
      />
    </div>

    <div className="form-group col-md-6">
      <label htmlFor="description">description</label>
      <br />
      <input
        type="text"
        onChange={(f) => setDescription(f.target.value)}
        className="form-control"
        placeholder="description"
      />
    </div>
  </div>

  <div className="form-group col-md-6">
    <label htmlFor="phoneNumber">phone Number </label>
    <input
      type="text"
      onChange={(f) => setphonenumber(f.target.value)}
      className="form-control"
    />
  </div>

  <div className="form-group col-md-6">
    <label htmlFor="type">Room type</label>
    <br />
    <input
      type="text"
      onChange={(f) => settype(f.target.value)}
    />
  </div>
```

```

21
      className="form-control"
      placeholder="type"
    />
  </div>

  <div className="form-group col-md-6">
    <label htmlFor="image1">image</label>
    <br />
    <input
      type="text"
      onChange={(f) => setImage1(f.target.value)}
      className="form-control"
      placeholder="upload image url"
    />
  </div>

  <br />
  <br />

7
  <button type="submit" className="btn btn-primary">
    Add new room
  </button>
</form>
</center>
<Link to="/Admin">
  <button type="button" className="btn btn-primary">
    back to admin
  </button>
</Link>
<br></br> <br></br>
</div>
);

};

export default AddRooms;

```

## Register.js

```

42
import React, {useState, useRef, useEffect} from 'react';
import AuthService from '../Services/AuthService';
import Message from '../Components/Message';

```

```
const Register = props=>{

    const [user, setUser] = useState({username: "", password : "", email : "", role : "user"});
    const [message, setMessage] = useState(null);
    let timerID = useRef(null);

    useEffect(()=>{

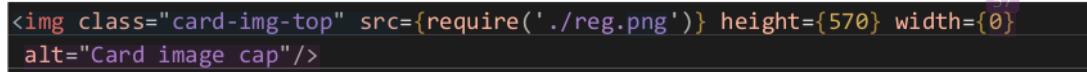
        return ()=>{
            clearTimeout(timerID);
        }
    },[]);

    const 5 onChange = e =>{
        setUser({...user,[e.target.name] : e.target.value});
    }

    const resetForm = ()=>{
        setUser({username : "", password : "", email : "",role : "admin"});
    }

    const onSubmit = e =>{
        e.preventDefault();
        AuthService.register(user).then(data=>{
            const { message } = data;
            setMessage(message);
            resetForm();
            if(!message.msgError){
                timerID = setTimeout(()=>{
                    props.history.push('/adminlogin');
                },2000)
            }
        });
    }

    return(
        <div className="container">
<br></br> <br></br>
<div class="card">
```

```


## Admin Register

Username:  
Password:  
Email:  
Role:Register

{message ? <Message message={message}/> : null}


```

```
      )
}

export default Register;
```

## Sysadmin.js

```
import React, { useEffect, useState } from "react";
import { Link } from "react-router-dom";
import axios from "axios";

const Sysadmin = () => (
  <div className="container">
    <br></br>
    <br></br>
    <h1>Welcome to System Admin page </h1>
    <Link to="/">
      <li className="nav-item nav-link">All Reservations</li>
    </Link>
  </div>
);

export default Sysadmin;
```

## AuthService.js

```
export default {
  login : user =>{
    console.log(user);
    return fetch('/user/login',{
      method : "post",
      body : JSON.stringify(user),
      headers : {
        'Content-Type' : 'application/json'
      }
    }).then(res => {
      if(res.status !== 401)
```

```

        return res.json().then(data => data);
    else
        return { isAuthenticated : false, user : {username : "",role :
""}};
    })
},
register : user =>{
    console.log(user);
    return fetch('/user/register',{
        method : "post",
        body : JSON.stringify(user),
        headers : {
            'Content-Type' : 'application/json'
        }
    }).then(res => res.json())
    .then(data => data);
},
logout : ()=>{
    return fetch('/user/logout')
    .then(res => res.json())
    .then(data => data);
},
isAuthenticated : ()=>{
    return fetch('/user/authenticated')
    .then(res=>{
        if(res.status !== 401)
            return res.json().then(data => data);
        else
            return { isAuthenticated : false, user : {username :
""},role : ""};
    });
}
}

```

## PrivateRoute.js

```

import React, {useContext} from 'react';
import {Route,Redirect} from 'react-router-dom';
import { AuthContext } from '../Context/AuthContext';

```

```
const PrivateRoute = ({component : Component, roles, ...rest})=>{  
  
    const { isAuthenticated, user } = useContext(AuthContext);  
    return(  
        <Route {...rest} render={props =>  
            if(!isAuthenticated)  
                return <Redirect to={{ pathname: '/login',  
                    state : {from : props.location}}}/>  
  
            if(!roles.includes(user.role))  
                return <Redirect to={{ pathname: '/',  
                    state : {from : props.location}}}/>  
            return <Component {...props}/>  
        })/>  
    )  
}  
  
export default PrivateRoute;
```

## UnPrivateRoute.js 27

```
import React, {useContext} from 'react';  
import {Route,Redirect} from 'react-router-dom';  
import { AuthContext } from '../Context/AuthContext';  
  
const UnPrivateRoute = ({component : Component,...rest})=>{  
    const { isAuthenticated } = useContext(AuthContext);  
    return(  
        <Route {...rest} render={props =>  
            if(isAuthenticated)  
                return <Redirect to={{ pathname: '/',  
                    state : {from : props.location}}}/>  
  
            return <Component {...props}/>  
        })/>  
    )  
}  
  
export default UnPrivateRoute;
```

1

## AuthContext.js

```
import React, {createContext, useState, useEffect} from 'react';
import AuthService from '../Services/AuthService';

export const AuthContext = createContext();

export default ({ children })=>{

    const [user, setUser] = useState(null);
    const [isAuthenticated, setIsAuthenticated] = useState(false);
    const [isLoaded, setIsLoaded] = useState(false);

    useEffect(()=>{
        AuthService.isAuthenticated().then(data =>{
            setUser(data.user);
            setIsAuthenticated(data.isAuthenticated);
            setIsLoaded(true);
        });
    }, []);

    return (
        <div>
            {!isLoaded ? <h1>Loading</h1> :
            <AuthContext.Provider
value={{user, setUser, isAuthenticated, setIsAuthenticated}}>
                { children }
            </AuthContext.Provider>}
        </div>
    )
}
```

ds

---

ORIGINALITY REPORT

---

**47** %  
SIMILARITY INDEX

41 %  
INTERNET SOURCES

12 %  
PUBLICATIONS

34 %  
STUDENT PAPERS

---

PRIMARY SOURCES

---

- |   |                                                                                    |            |
|---|------------------------------------------------------------------------------------|------------|
| 1 | <b>stackoverflow.com</b><br>Internet Source                                        | <b>7</b> % |
| 2 | <b>Submitted to University of Ulster</b><br>Student Paper                          | <b>4</b> % |
| 3 | <b>Submitted to Sri Lanka Institute of Information Technology</b><br>Student Paper | <b>3</b> % |
| 4 | <b>Submitted to Coventry University</b><br>Student Paper                           | <b>2</b> % |
| 5 | <b>Submitted to University of Northampton</b><br>Student Paper                     | <b>2</b> % |
| 6 | <b>Submitted to University of Hertfordshire</b><br>Student Paper                   | <b>2</b> % |
| 7 | <b>gitlab.sliit.lk</b><br>Internet Source                                          | <b>2</b> % |
| 8 | <b>Submitted to Rochester Institute of Technology</b><br>Student Paper             | <b>1</b> % |
| 9 | <b>Submitted to University of Westminster</b>                                      |            |

- 
- 10 abbasimusab2000.medium.com 1 %  
Internet Source
- 
- 11 gitlab.devops.ifrn.edu.br 1 %  
Internet Source
- 
- 12 www.angularfix.com 1 %  
Internet Source
- 
- 13 pastebin.com 1 %  
Internet Source
- 
- 14 www.thebankerdatabase.com 1 %  
Internet Source
- 
- 15 Submitted to Technological Institute of the Philippines 1 %  
Student Paper
- 
- 16 jasonwatmore.com 1 %  
Internet Source
- 
- 17 Submitted to Sim University 1 %  
Student Paper
- 
- 18 mdbootstrap.com 1 %  
Internet Source
- 
- 19 Juntao Qiu. "Test-Driven Development with React", Springer Science and Business Media LLC, 2021 1 %  
Publication
-

20	dokumen.pub Internet Source	1 %
21	sebastian.com Internet Source	1 %
22	Submitted to Tarlac State University Student Paper	<1 %
23	www.passportjs.org Internet Source	<1 %
24	Submitted to University of London External System Student Paper	<1 %
25	lite.getshoutout.com Internet Source	<1 %
26	www.178linux.com Internet Source	<1 %
27	Submitted to University of Sydney Student Paper	<1 %
28	hogent-web.github.io Internet Source	<1 %
29	openarchive.nure.ua Internet Source	<1 %
30	www.techiematrix.com Internet Source	<1 %

31	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1 %
32	omi-gitlab.e-technik.uni-ulm.de Internet Source	<1 %
33	www.phpro.org Internet Source	<1 %
34	github.coventry.ac.uk Internet Source	<1 %
35	wb-hk.blogspot.com Internet Source	<1 %
36	Submitted to The Cornell Education Group Student Paper	<1 %
37	gitlab.fdmci.hva.nl Internet Source	<1 %
38	repositorio.ufsc.br Internet Source	<1 %
39	mixtapefranchise.com Internet Source	<1 %
40	gitlab.ecs.vuw.ac.nz Internet Source	<1 %
41	ekmair.ukma.edu.ua Internet Source	<1 %
42	git.kpi.fei.tuke.sk	

Internet Source

<1 %

43 [gitlab.stud.idi.ntnu.no](http://gitlab.stud.idi.ntnu.no)  
Internet Source

<1 %

44 [tufat.com](http://tufat.com)  
Internet Source

<1 %

45 [speakerdeck.com](http://speakerdeck.com)  
Internet Source

<1 %

46 [webhaunt.com](http://webhaunt.com)  
Internet Source

<1 %

47 [www.sparkpeople.com](http://www.sparkpeople.com)  
Internet Source

<1 %

48 Submitted to Middle East College of  
Information Technology  
Student Paper

<1 %

49 [codebun.com](http://codebun.com)  
Internet Source

<1 %

50 [hackerthemes.com](http://hackerthemes.com)  
Internet Source

<1 %

51 Submitted to unibuc  
Student Paper

<1 %

52 [react-bootstrap.netlify.com](http://react-bootstrap.netlify.com)  
Internet Source

<1 %

53 [library.samdu.uz](http://library.samdu.uz)  
Internet Source

<1 %

---

54 tools.ages.pucrs.br <1 %  
Internet Source

55 Adam Freeman. "Pro React 16", Springer  
Science and Business Media LLC, 2019 <1 %  
Publication

56 Submitted to itera <1 %  
Student Paper

57 projectcs.sci.ubu.ac.th <1 %  
Internet Source

58 Submitted to Buckinghamshire Chilterns  
University College <1 %  
Student Paper

59 Nabendu Biswas. "Foundation Gatsby  
Projects", Springer Science and Business  
Media LLC, 2021 <1 %  
Publication

60 www.joshmorony.com <1 %  
Internet Source

61 www.mokujime <1 %  
Internet Source

62 Submitted to AUT University <1 %  
Student Paper

63 codenza.app

Internet Source

<1 %

64 [github.com](https://github.com)  
Internet Source

<1 %

65 [itnext.io](https://itnext.io)  
Internet Source

<1 %

66 [knowledge.cryptokylin.io](https://knowledge.cryptokylin.io)  
Internet Source

<1 %

67 [slides.com](https://slides.com)  
Internet Source

<1 %

68 [webdesignleaves.com](https://webdesignleaves.com)  
Internet Source

<1 %

69 [woutr.me](https://woutr.me)  
Internet Source

<1 %

70 [www.freecodecamp.org](https://www.freecodecamp.org)  
Internet Source

<1 %

71 Mohit Thakkar. "Building React Apps with  
Server-Side Rendering", Springer Science and  
Business Media LLC, 2020  
Publication

<1 %

72 [developer.aliyun.com](https://developer.aliyun.com)  
Internet Source

<1 %

73 [gitlab.uwe.ac.uk](https://gitlab.uwe.ac.uk)  
Internet Source

<1 %

74	soshace.com Internet Source	<1 %
75	transfer.hft-stuttgart.de Internet Source	<1 %
76	www.c-sharpcorner.com Internet Source	<1 %
77	casualdevelopers.com Internet Source	<1 %
78	essuir.sumdu.edu.ua Internet Source	<1 %
79	gitlab.eeecs.qub.ac.uk Internet Source	<1 %
80	inde601-shift.herokuapp.com Internet Source	<1 %
81	khuhub.khu.ac.kr Internet Source	<1 %
82	www.thiscodeworks.com Internet Source	<1 %
83	Azat Mardan. "Practical Node.js", Springer Science and Business Media LLC, 2018 Publication	<1 %
84	Vasan Subramanian. "Pro MERN Stack", Springer Science and Business Media LLC, 2019 Publication	<1 %

85

Nabendu Biswas. "MERN Projects for Beginners", Springer Science and Business Media LLC, 2021

<1 %

Publication

---

86

stripe.com

Internet Source

<1 %

---

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

On

ds

---

PAGE 1

---

PAGE 2

---

PAGE 3

---

PAGE 4

---

PAGE 5

---

PAGE 6

---

PAGE 7

---

PAGE 8

---

PAGE 9

---

PAGE 10

---

PAGE 11

---

PAGE 12

---

PAGE 13

---

PAGE 14

---

PAGE 15

---

PAGE 16

---

PAGE 17

---

PAGE 18

---

PAGE 19

---

PAGE 20

---

PAGE 21

---

PAGE 22

---

PAGE 23

---

PAGE 24

---

PAGE 25

---

---

PAGE 26

---

PAGE 27

---

PAGE 28

---

PAGE 29

---

PAGE 30

---

PAGE 31

---

PAGE 32

---

PAGE 33

---

PAGE 34

---

PAGE 35

---

PAGE 36

---

PAGE 37

---

PAGE 38

---

PAGE 39

---

PAGE 40

---

PAGE 41

---

PAGE 42

---

PAGE 43

---

PAGE 44

---

PAGE 45

---

PAGE 46

---

PAGE 47

---

PAGE 48

---

PAGE 49

---

PAGE 50

---

PAGE 51

---

PAGE 52

---

PAGE 53

---

PAGE 54

---

PAGE 55

---

PAGE 56

---

PAGE 57

---

PAGE 58

---

PAGE 59

---

PAGE 60

---

PAGE 61

---

PAGE 62

---

PAGE 63

---

PAGE 64

---

PAGE 65

---

PAGE 66

---

PAGE 67

---

PAGE 68

---

PAGE 69

---

PAGE 70

---

PAGE 71

---

PAGE 72

---

PAGE 73

---

PAGE 74

---

PAGE 75

---

PAGE 76

---

PAGE 77

---

