



Sri Lanka Institute of Information Technology

## **Distributed Systems**

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

Submitted By:

1. IT20132750– Izmie.A.A
2. IT20088682 – Jayathilaka K.D.M.I
3. IT20158422 – Paboda P.D.K
4. IT20134730– Jayathilaka A.D.N

## Contents

1.	Introduction .....	4
2.	High-Level Architectural Diagram .....	5
3.	Tools & Technologies .....	6
3.1	Rest API .....	6
3.2	Web Client.....	6
3.3	Database .....	6
3.4	Desktop Application (Admin) .....	6
3.5	Email Service .....	6
3.6	SMS Service.....	7
3.7	Payment Service.....	7
4.	System Components .....	7
4.1	Rest API .....	7
4.1.1	Rest API End Points .....	7
4.2	Web Client.....	8
4.3	Desktop Application.....	9
4.4	Email Service .....	9
4.5	SMS Service.....	10
4.6	Payment gateway.....	10
4.7	Database .....	11
4.8	Authentication and Security .....	11
5.	Interconnectivity.....	12
5.1.1	Booking room.....	12
5.1.2	Cancel reservation .....	13
5.1.3	Customer Register.....	14
5.1.4	Customer login .....	15
6.	Appendix .....	16
6.1	Rest API .....	16
6.1.2	Model class .....	17
6.1.3	Routes .....	20
6.2	Web Client.....	34
6.2.2	Components.....	41

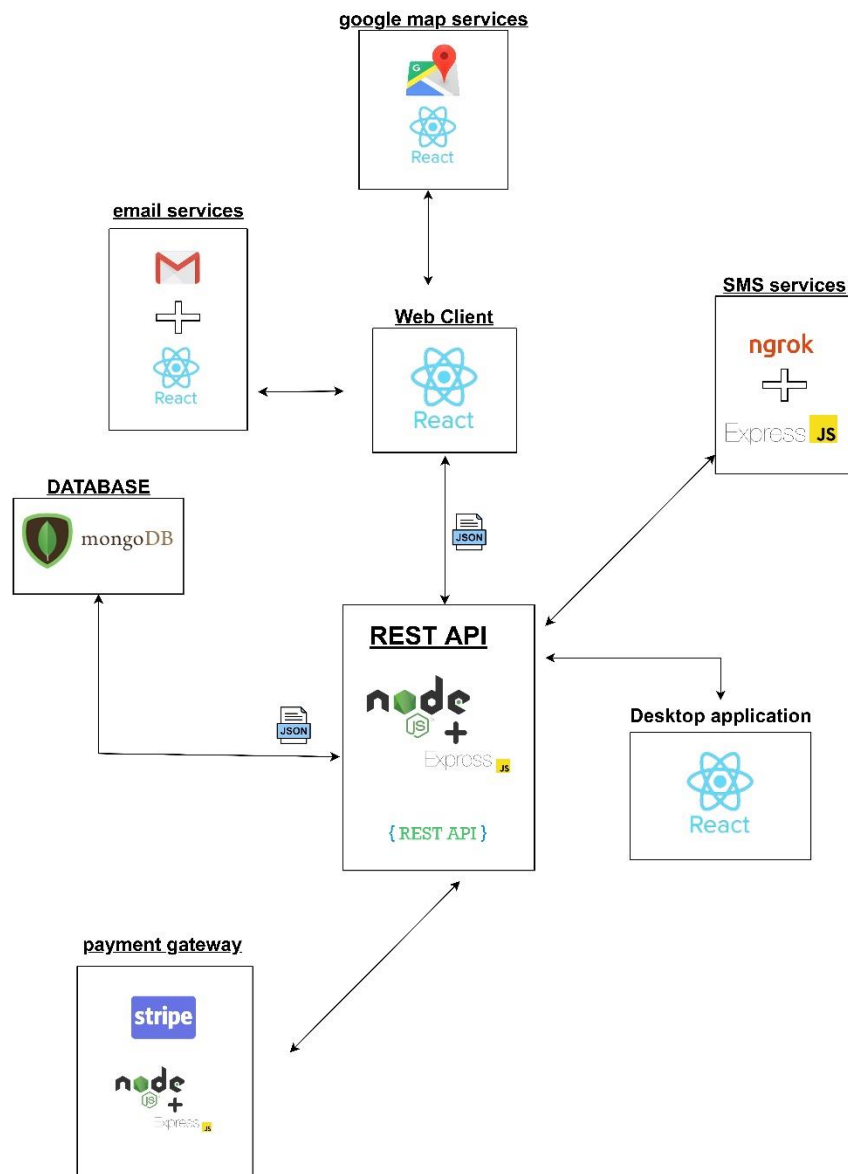
6.2.3	Database connection .....	62
6.3	Desktop Application (Admin) .....	63
6.3.1	Admin.js .....	63
6.3.2	AdminNarBar.js .....	64
6.3.3	AllReservationAdmin.js .....	66
6.3.4	UpdateRooms.js .....	68
6.3.5	ManageAllUsers.js .....	71
6.3.6	ManageAllRooms.js .....	73
6.3.7	AddRooms.js .....	75
6.3.8	Register.js .....	78
6.3.9	Sysadmin.js .....	81
6.3.10	AuthService.js .....	81
6.3.11	PrivateRoute.js .....	82
6.3.12	UnPrivateRoute.js .....	83
6.3.13	AuthContext.js .....	83

# **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



## **3. Tools & Technologies**

### **3.1 Rest API**

- Node js and express js
- Postman
- Vs code

### **3.2 Web Client**

- VS Code
- React Js
- Axios HTTP Client

### **3.3 Database**

- MongoDB

### **3.4 Desktop Application (Admin)**

- React js
- Vs code

### **3.5 Email Service**

- Gmail
- Vs code
- React email

### 3.6 SMS Service

- Shoutout
- Express js
- Vs code

### 3.7 Payment Service

- Node js
- Express js
- Vs code
- React js

## 4. System Components

### 4.1 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.

#### 4.1.1 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/Malier	Send payment confirm email

## 4.2 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.



### **4.3 Desktop Application**

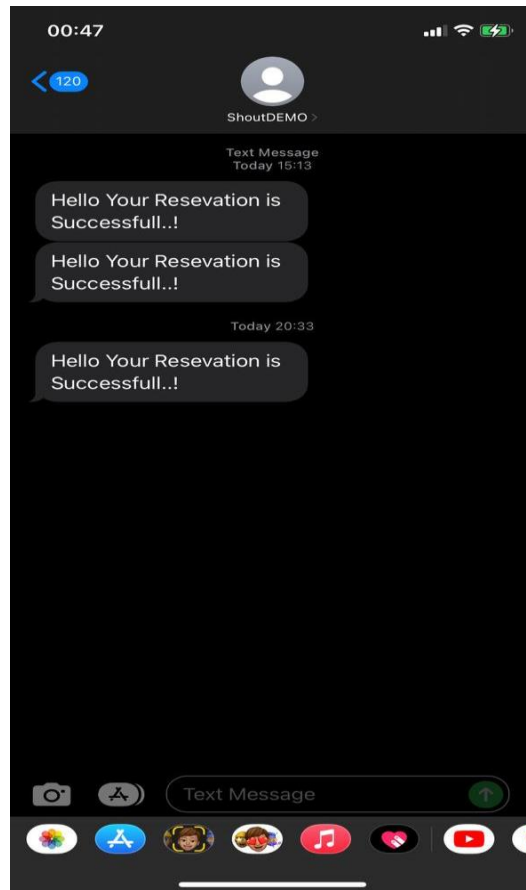
The desktop application has an Admin, and system admin login. So, unauthorized users cannot access the system where it provides security for the application. this also used run on port 3000. To request HTTP requests desktop application has used the Axios HTTP client. This provides a dashboard that displays 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.

### **4.4 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.

## 4.5 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.



## 4.6 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

## **4.7 Database**

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

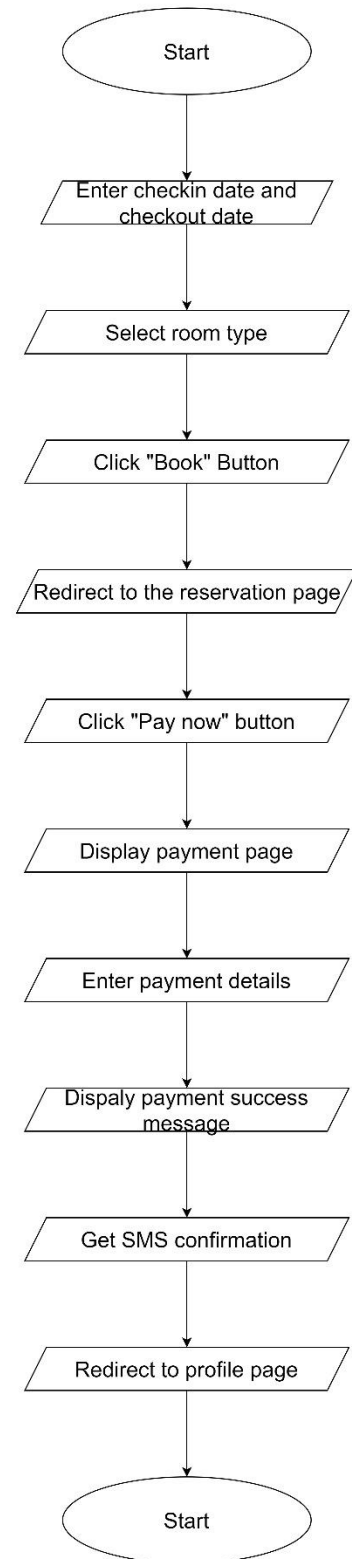
## **4.8 Authentication and Security**

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.

## 5. Interconnectivity

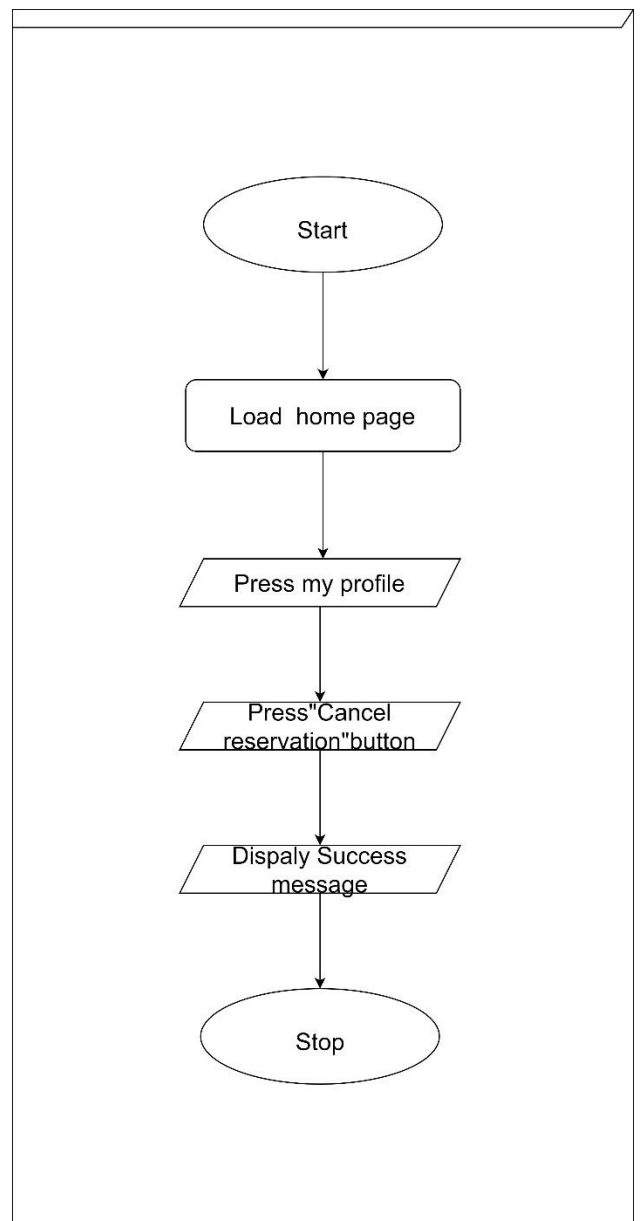
### 5.1.1 Booking room

When the customer register to the system then user can booking reservation option one reservation or more restation can booking one user .



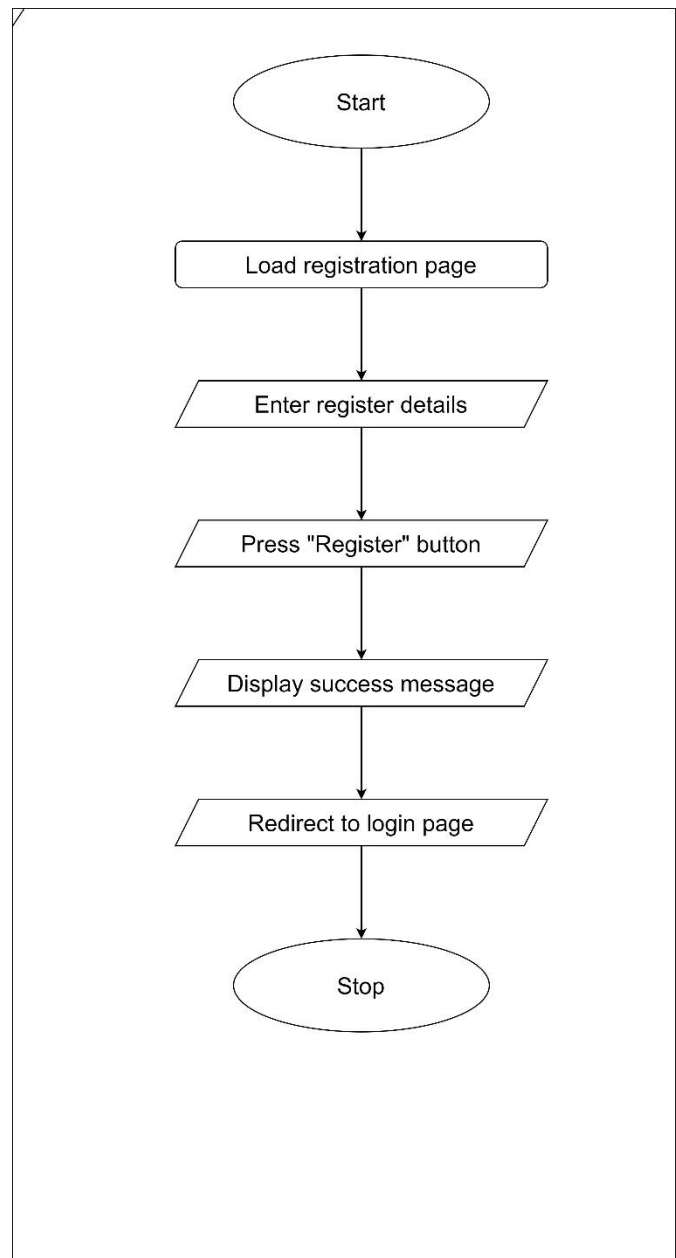
### 5.1.2 Cancel reservation

When the customer needs to cancel there reservation customer can cancel there reservation login to the there profile.



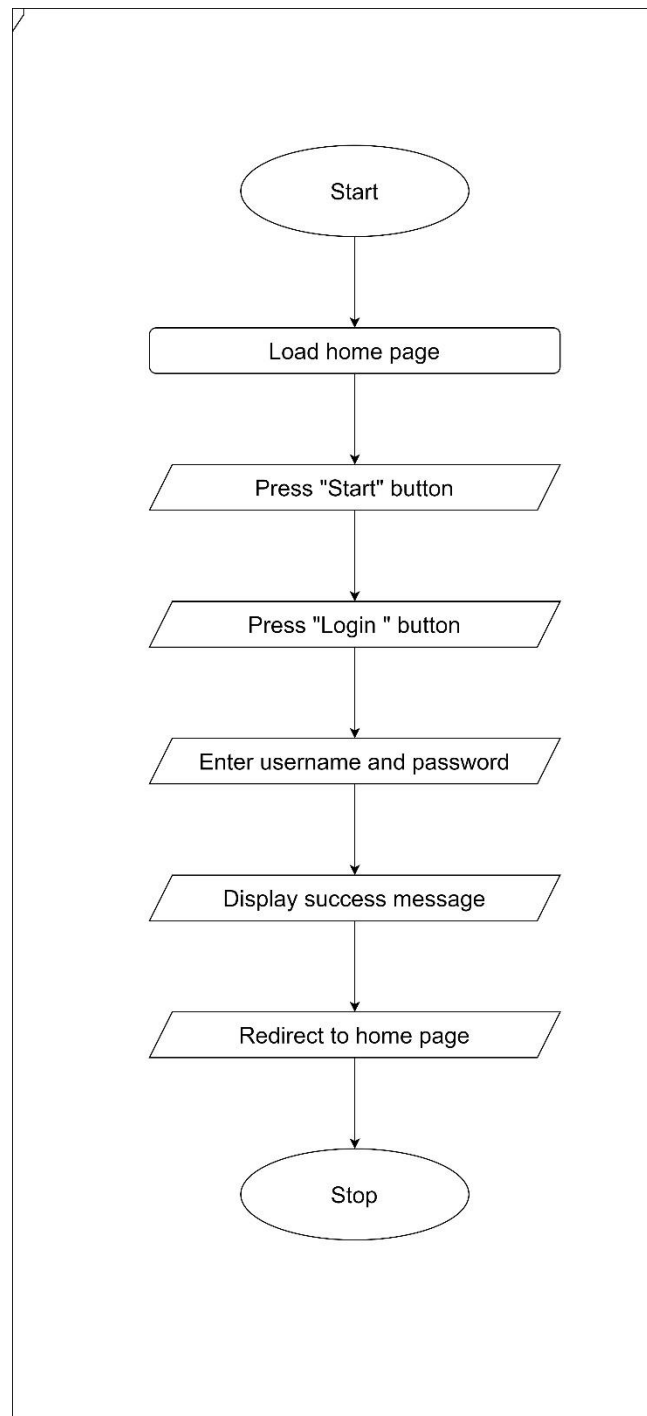
### 5.1.3 Customer Register

If customer need to make the reservations 1<sup>st</sup> customer need to register to the system and after the customer can make reservation



### 5.1.4 Customer login

Customer need to login there profile  
customer must enter there user  
email and password



## 6. Appendix

### 6.1 Rest API

#### 6.1.1.1.1 Package.Json (RESTfulWebServices)

```
{
  "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"
  }
}
```



## 6.1.2 Model class

### 6.1.2.1.1 Admin.js

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

const UserSchema = new mongoose.Schema({

  username :{
    type : String,
    required : true,
    min : 6,
    max : 65
  },

  password : {
    type : String,
    required : true
  },

  role : {
    type : String,
    enum : ['user','admin','sysadmin'],
    required: true
  },

  email : {
    type : String,
    required : true
  },

});

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);

```

#### 6.1.2.1.1.1 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

```

#### 6.1.2.1.1.2 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;

```

#### 6.1.2.1.1.2.1 user.js

```

const mongoose = require("mongoose");

const userSchema = mongoose.Schema(
  {
    name: { type: String, require },
    email: { type: String, require },
    country: { type: String, require },
    phonenumber: { type: String, require },
    password: { type: String, require },
    isAdmin: { type: Boolean, require, default: false },
  },

```

```

    {
      timestamps: true,
    }
  );

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

```

## 6.1.3 Routes

### 6.1.3.1 *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;

```

#### 6.1.3.2 *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}`));
});

```

```

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
```

### 6.1.3.3 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  
  
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 occurred",
msgError: true}}});
        else
            res.status(201).json({message : {msgBody : "Account successfully
created", msgError: false}}});
        });
    });
});
```

*//login*

```
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*

```
userRouter.get('/logout',passport.authenticate('jwt',{session :
false}),(req,res)=>{
```

```
    res.clearCookie('access_token');
    res.json({user:{username : "", role : ""},success : true});
});
```

```

//admin user role
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
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}}});
});

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

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

});

// all account login authenticated

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;
```

#### 6.1.3.4 smsRoutes.js

```
const express = require("express");
const router = express.Router();
var ShoutoutClient = require("shoutout-sdk");
var debug = true,
    verifySSL = false;
var apiKey =
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI1ZDk5MWJmMC1kNGU0LTExZWMTODY0Mz0zOTQzMtUzY2I4ZjEiLCJzZWUiOiJTSE9VVE9VVF9BUeIjVFNlhdCI6MTY1MjY4Mzc4MiwiaXNjaXoxOTY4MzAyOTgyLCJzY29wZXMiOnsiYWN0aXZpdGllcyI6WyJyZWZkIiwid3JpdGUhXSwibWVzc2FnZXMiOnsicmVhZCIsIndyaXRlIj0sImNvbnRhY3RzIjpbInJlYWQiLCJ3cm10ZSJdfSwic29fdXNlc19pZCI6IjY4MDE3Iiwic29fdXNlc19yb2x1IjoiaXNlcisiInNvX3Byb2ZpbGUhOiJhbGwiLCJzb191c2VyX25hbWUiOiIiLCJzb19hcGlrZXkiOiJub251In0.Mk21zIIIn-EmFzwCy3AFM2GZKSUw42pmmqVprx7hI5tw";

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

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

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

module.exports = router;
```

#### 6.1.3.5 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

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,
        phonenumber: user[0].phonenumber,
        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" });
  }
});
```

```

    }
  });

  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;

```

#### 6.1.3.6 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;

```

#### 6.1.3.7 passport.js

```

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

passport.use(new LocalStrategy((username,password,done)=>{

  User.findOne({username}),(err,user)=>{

    //something went wrong with database
    if(err)
      return done(err);

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

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

  });

}));
```

### 6.1.3.8 Server.js

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

const server = http.createServer(app);

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

## 6.2 Web Client

### 6.2.1.1 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"
}

```

### 6.2.1.2 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">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"
">
    <link rel="stylesheet" href="node_modules/mdbootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="node_modules/mdbootstrap/css/mdb.min.css">
    <link rel="stylesheet" href="node_modules/mdbootstrap/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-
wvXpqpZZVQGK6TAh5PV1GOfQNHSoD2xbE+QkPxCAF1NEevoEH3Sl0sibVcOQVnN"
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>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
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+P
ro: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-
eMNC0e7tC1d0HpGoWe/6oMVemdAVTMs2xqW4mwXrXSwOL84Iytr2wi5v2QjrP/xp"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/js/bootstrap.min.js"
integrity="sha384-
cn7l7gDp0eyniUwwAZgrzD06kc/tftFf19TOAs2zVinnD/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+P
ro: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-
eMNC0e7tC1d0HpGoWe/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 1.293zm-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-.121l6.813-6.814z"/>
    <path fill-rule="evenodd" d="M1 13.5A1.5 1.5 0 0 0 2.5 15h11a1.5 1.5 0 0 0
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-6a.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>

    <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/IqJ95wo16oMtfS KbZ9ccEh31e0z1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
QJHtvGhmr9X0IpI6YVutG+2Q0K9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ka7Sk0GlN4gmtz2MlQnikT1wXgYs0g+OMhuP+IlRH9sENB00LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

</body>
</html>

```

### 6.2.1.3 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>
);
```

### 6.2.1.4 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={Homescreeen} />
        <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} />}
        />
        <PrivateRoute path="/Admin" roles={["admin"]} component={Admin} />
        <PrivateRoute
          path="/allReservation"
          roles={["admin"]}
          component={AllReservationAdmin}
        />
        <PrivateRoute
          path="/adminroom"
          roles={["admin"]}
          component={ManageAllRooms}
        />
        <PrivateRoute
          path="/ManageAllUsers"
          roles={["admin"]}
        />
      </Router>
    </div>
  );
}

```



```

        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;

```

## 6.2.2 Components

### 6.2.2.1 BookingRoom.js

```

import React, { useEffect, useState } 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 StripeCheckout from 'react-stripe-checkout'

import moment from "moment"
import AOS from 'aos';

AOS.init();
AOS.refresh()

```

```

function BookingRoom({match}) {

  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')
  const todate=moment(match.params.todate,'DD-MM-YYYY')
  const totalDays = moment.duration(todate.diff(fromdate)).asDays()+1
  const [totalAmount , settotalAmount]=useState()
  const [phonenummer , setphonenummer]=useState()

  const smsfunc= (e) => {
    e.preventDefault();
    let number = JSON.parse(localStorage.getItem('currentUser')).phonenummer
    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);
      })
  }

  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,
        phonenumber,
        todate,
        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 suceessfully' ,
'success').then(result=>{
            window.location.href='/profile'
        })
    } catch (error) {
        console.log(error);
        setloading(false)
    }
}

```



```

        <br></br>
        <br></br>
        <br></br>
        <StripeCheckout

        token={tokenHandler}
        stripeKey='pk_test_51KyZL6FeoRdqg6f816PBNMT9VvvDUCbwgUS6pvLiK2kQ6Mvx3
q4BOFTQc9ENZpb0IeCH2a3GABcaxBbTStcyvocg00TK9Bbc8z' >

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

        </StripeCheckout>

        </div>

    </div>

</div>

    )
}

export default BookingRoom;

```

### 6.2.2.2 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";

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, settype] = 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);
}
}

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 dupdate = hotels.filter((room) =>
    room.name.toLowerCase().includes(searchkey)
  );
  setreservation(dupdate);
}

function filterByType(e) {
  settype(e);
  if (e !== "all") {
    const dupdate = hotels.filter((room) =>
      room.type.toLowerCase().includes(e.toLowerCase())
    );
    setreservation(dupdate);
  } else {
    setreservation(hotels);
  }
}

```

```

return (
  <div className="mt-5">
    <div className="container">
      <div className="row bs p-3 m-5">
        <div className="col-md-3">
          <RangePicker
            id="datepicker"
            type="date"
            onChange={filterByDate}
            format="DD-MM-YYYY"
            className="m-2"
          />
        </div>

        <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);
            }}
          >
            <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>

  <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;

```

### 6.2.2.3 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 booking
          </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>
    <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;

```

#### 6.2.2.4 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);

```

mailer.js

```

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

const Mailer = () => {

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

  function sendEmail(e) {

    e.preventDefault();

    emailjs.sendForm(
      "service_24jbmdk",
      "template_dg25svx",
      e.target,
      "15NUKPpbvRhbn3ZL1"
    ).then(res=>{
      alert("email send success ")
      console.log(res);
    }).catch(err=> console.log(err));
  }

  return (

```

```

<div
  className="container border"
  style={{
    marginTop: "50px",
    width: "50%",
    backgroundImage: `url('${}')`,
    backgroundPosition: "center",
    backgroundSize: "cover",
  }}
>
  <h1 style={{ marginTop: "25px" }}>Contact Form</h1>
  <form
    className="row"
    style={{ margin: "25px 85px 75px 100px" }}
    onSubmit={sendEmail}
  >
    <label>name</label>
    <input type="text" name="name" value={user.name} className="form-
control" />

    <label>Email</label>
    <input type="email" name="user_email" value={user.email} className="form-
control" />

    <label>Message</label>
    <textarea name="message" rows="4" value="hotel reservation successfully "
className="form-control" />
    <input
      type="submit"
      value="Send"
      className="form-control btn btn-primary"
      style={{ marginTop: "30px" }}
    />
  </form>
</div>
);
};

export default Mailer;

```

#### 6.2.2.5 DisplayAllRooms.js

```

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 &nbsp; &nbsp; ,Business Center, &nbsp; &nbsp;
          Fitness Center &nbsp; &nbsp; ,Lounge/Bar &nbsp; &nbsp; ,Internet
          Access - Free in room &nbsp; &nbsp; , Contactless check-in and
          check-out
        </p>

        <p>
          <b>Type : {room.type}</b>
        </p>
        <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>
          )}

          <Link to="/home">
            <button className="btn btn-primary">view </button>
          </Link>
        </div>
      </div>
    </div>
  );
}

```

```

        </div>
      </div>
    </div>
  );
}
export default DisplayAllRooms;

```

#### 6.2.2.6 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);
    }
  }

```



```

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

```

#### 6.2.2.7 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";
import { Link } from "react-router-dom";
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>
              <h4>my id : {user._id}</h4>

            </div>
          </div>
        </button>
        <TabPane tab="Bookings" key="2">
          <h1>
            <MyReservation />
          </h1>
        </TabPane>
      </Tabs>
    </div>
  )
}

```



```

    </Tabs>

    <Link to="/Mailer">
      <button type="button" class="btn btn-primary">
        get email
      </button>
    </Link>

  </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 suceessfully' ,
'success').then(result=>{
            window.location.href='/profile'
        })
    } catch (error) {
        Swal.fire('Congrats' , 'Your Room has cancelled suceessfully' ,
'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>
            )
        )
    )

```

```

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

```

#### 6.2.2.8 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} </but
ton>

    <div id="myDropdown" class="dropdown-content" >

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

    </div>

  </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;

```

#### 6.2.2.9 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;
```

#### 6.2.2.10 Success.js

```
import React from "react";

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

#### 6.2.2.11 Error.js

```
import React from "react";

export default function Error({ error }) {
  return (
    <div>
```

```

    <div className="alert alert-danger" role="alert">
      {error}
    </div>
  </div>
);
}

```

### 6.2.2.12 Message.js

```

import React from "react";

const getStyle = (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={getStyle(props)} role="alert">
      {props.message.msgBody}
    </div>
  );
};

export default Message;

```

### 6.2.3 Database connection

```

const mongoose = require('mongoose');

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 ');  
});
```

### 6.3 Desktop Application (Admin)

### 6.3.1 Admin.js

[illegible]





```

        <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 class="navbar navbar-expand-lg navbar-dark bg-dark">

```

```

    <Link class="navbar-brand" href="#">
      Hotel Reservation{" "}
    </Link>

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

export default AdminNarBar;

```

### 6.3.3 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>

        <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;

```

### 6.3.4 UpdateRooms.js

```

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),
                settype(res.data.type),
            ])

            .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">
    <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>
    </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;

```

### 6.3.5 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;
setusers(data);
setloading(false);
} catch (error) {
  console.log(error);
  setloading(false);
}
}, []));

```

```

return (
  <div className="row">
    {loading && <Loader />}

    <div className="col-md-10">
      <br></br>
      <br></br>
      <Link to="/Admin">
        <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">
          <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>
);

```



```

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

export default ManageAllUsers;

```

### 6.3.6 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";
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);

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

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

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

```



```

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

export default ManageAllRooms;

```

### 6.3.7 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, setphonenummer] = useState("");
  const [type, settype] = 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)}
      className="form-control"
    />
  </div>

```

```

        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 />

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

export default AddRooms;

```

### 6.3.8 Register.js

```

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 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">
<img class="card-img-top" src={require('./reg.png')} height={570}
width={0} alt="Card image cap"/>
<div class="card-body">

```

```

<h2 class="card-title">Admin Register</h2>
<form onSubmit={onSubmit}>
  <label htmlFor="username" className="sr-only">Username: </label>
    <input type="text"
      name="username"
      value={user.username}
      onChange={onChange}
      className="form-control"
      placeholder="Enter Username"/>
    <br></br>
    <label htmlFor="password" className="sr-only">Password: </label>
    <input type="password"
      name="password"
      value={user.password}
      onChange={onChange}
      className="form-control"
      placeholder="Enter Password"/>
    <br></br>
    <label htmlFor="email" className="sr-only">email: </label>
    <input type="email"
      name="email"
      value={user.email}
      onChange={onChange}
      className="form-control"
      placeholder="Enter email"/>

  <br></br>

    <label htmlFor="role" className="sr-only">Role: </label>
    <input type="text"
      name="role"
      value={user.role}
      onChange={onChange}
      className="form-control"
      placeholder=""/>
    <br></br>

    <button className="btn btn-lg btn-primary btn-block"
      type="submit">Register</button>

  </form>
  {message ? <Message message={message}/> : null}
</div>
</div>
</div>
  )
}

```



```
export default Register;
```

### 6.3.9 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;
```

### 6.3.10 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 : ""}};
      });
  }
}

```

### 6.3.11 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;

```

### 6.3.12 UnPrivateRoute.js

```

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;

```

### 6.3.13 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 [isLoading, setIsLoaded] = useState(false);

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

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