

# **Sri Lanka Institute of Information Technology**



## **Assignment (II)**

**SE3020 – Distributed System**

**REST API Project**

## **Online Movie Reservation Platform**

<b>Student Name</b>	<b>IT Number</b>
<b>Senevirathna D.K.C.</b>	<b>IT20009540</b>
<b>Nadula J.K.D.P.</b>	<b>IT20168568</b>
<b>Mapatuna R.M.</b>	<b>IT20072674</b>
<b>Kulathunga H.M.C.K.</b>	<b>IT20274498</b>

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****Content**

Figures .....	3
Tables.....	4
1.0 Introduction .....	5
2.0 Tools and Technologies used.....	6
3.0 Distributed System Components .....	7
3.1 REST API.....	7
3.1.1 Models .....	7
3.1.2 Endpoints.....	8
3.2 Email and QR Component.....	9
3.2.1 Email .....	9
.....	9
3.2.2 QR .....	10
3.3 Security and Authentication .....	12
3.4 Validations .....	12
3.5 WSO2 Enterprise Integrator .....	12
4.0 High-level Architectural diagram .....	13
5.0 UseCase Diagram .....	14
6.0 Sequence Diagram .....	15
7.0 Class Diagrams .....	16
8.0 Appendix .....	17
8.1 BackEnd .....	17
8.1.1 Models .....	17
8.1.2 Routes .....	22
8.1.3 Controllers .....	26
8.1.4 Middleware.....	45
8.1.5 Server.js .....	48
8.1.6 package.json .....	49

## Assignment II – REST API Project

### SE3020 – Distributed Systems

### Semester 1

8.2 FrontEnd .....	50
8.2.1 Components.....	50
8.2.2 Layouts.....	175
8.2.3 Pages.....	175
8.2.4 Reducers .....	179
8.2.5 App.js .....	182
8.2.6 index.js.....	185
8.2.7 package.json .....	186

## Figures

Figure 1: Booking Confirmation Email.....	9
Figure 2:QR Code Generator.....	10
Figure 3: QR Code Scanner (Using Camera) .....	11
Figure 4:QR Code Scanner (Using Image).....	11
Figure 5:High-level Architecture.....	13
Figure 6: UseCase Diagram.....	14
Figure 7:Sequence Diagram .....	15
Figure 8:Class Diagram .....	16

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****Tables**

Table 1: BackEnd Components .....	7
Table 2: Endpoints Summary .....	8

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****1.0 Introduction**

Nowadays, the IT sector is growing rapidly. So, with the advancement of the IT sector, all other fields are automatically evolving because these days all the fields are using ICT for their functionality to work effectively and easily.

The purpose of our project is to develop a collaborative online movie reservation platform for the requirements given by the client using the latest technology. The purpose of building this project is to ease the online movie management and all other activities related to the system. Fans still have to go to many cinemas and book a movie ticket, because of that those people have to face many issues.

By developing this project we try to solve the issues that fans were faced and provide an easy, efficient way to manage the online movie booking.

Our system has three main roles: Customer, Movie Administrator and System Administrator. The production crew can add, update, and delete movies under multiple theatres using our online movie reservation platform. In all of the films that are shown, Our website includes the show time, description, cast, and other pertinent information so that the user may easily find its details. Users can use this service to search/book and cancel movie reservations. Also they can book numerous tickets at several cinemas at various times. Once a ticket or multiple tickets are added to cart, the user can select the theater option, where a request may be sent to the relevant movie theater. The payment for the tickets can be made using credit cards.

Once you made the payments successfully system issue a unique QR code which contains necessary details about your reservation. You can scan the code to check the details. Also within 8 hours you can cancel your reservation and get your money back.

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****2.0 Tools and Technologies used.**

1. Web Client.
  - React JS
  - VS code
  - Bulma.io
2. REST API
  - Node JS
  - Express JS
  - Mongo DB Atlas
  - Postman
3. API Integration
  - WSO2 Enterprise Integrator
4. SMS gateway
  -
5. Email
  - NodeMailer
6. JSON Web Token

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

## 3.0 Distributed System Components

### 3.1 REST API

Rest API was developed using NodeJS, ExpressJS, and MongoDB Atlas as the database.

Section	Model	Controller	Route
User	User.js	user.js	users.js
Movie	Movie.js	Movie.js	Movie.js
Category	Category.js	Category.js	Category.js
Movie Hall	MovieHall.js	MovieHall.js	MovieHall.js
Show	Show.js	Show.js	Show.js
Booking	Booking.js	Booking.js	Booking.js

*Table 1: BackEnd Components*

#### 3.1.1 Models

- The mongoose schema for User has 5 attributes which are name, email, userName, password and type. Type attribute is not required and other 4 attributes are required.
- The mongoose schema for Movie has 9 attributes which are name, categoryId, description, image, startDate, endDate, area, rate, hallid. name is required. hallid attribute is the ObjectId of MovieHall and area attribute is the ObjectId of Category.
- The mongoose schema for Category has 1 attribute which is name. name is a required attribute and it has a 32 max length and should be unique.
- The mongoose schema for MovieHall has 5 attributes which are name, rows, cols, seats, rate. name attribute is required.
- The mongoose schema for Show has 6 attributes which are mid, hid, date, time, seatbook, price. mid attribute is a Objectid which refers to “movie” and hid attribute also is a objectId which refers to “hall”.
- The mongoose schema for Booking has 8 attributes which are uid, sid, tickets, price, total, mid, seats, token. uid, sid and mid are ObjectIds which refers to “user”, “show” and “movie”.

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

#### 3.1.2 Endpoints

The following table will show the endpoints created for sending and retrieving data

Method	End Point	Purpose
POST	http://localhost:8070/user /login	Login User
PUT	http://localhost:8070/user/update/:userId	Update a User Profile by ID
POST	http://localhost:8070/user/signup	Register User
GET	http://localhost:8070/user/getuser/:userId	Get a User by ID
GET	http://localhost:8070/user/all	Get all Users
DELETE	http://localhost:8070/user/delete/:id	Delete a User by ID
POST	http://localhost:8070/movies/add	Add a Movie
GET	http://localhost:8070/movies/all	Admin get all Movies
GET	http://localhost:8070/movies/list	Get all Movies
POST	http://localhost:8070/movies/by/search	Search Movies
GET	http://localhost:8070/movies/getone/:id	Get a Movie by ID
PUT	http://localhost:8070/movies/update/:id	Update a Movie by ID
DELETE	http://localhost:8070/movies/delete/:id	Delete a Movie by ID
GET	http://localhost:8070/categories/list	Get all Categories
POST	http://localhost:8070/categories/create	Create Category
POST	http://localhost:8070/hall/	Add a Movie Hall
GET	http://localhost:8070/hall/list	Get all Movie Halls
GET	http://localhost:8070/hall/getone/:id	Get a Movie Hall by ID
PUT	http://localhost:8070/hall/update/:id	Update Movie Hall by ID
DELETE	http://localhost:8070/hall/delete/:id	Delete Movie Hall by ID
POST	http://localhost:8070/show/	Create a Show
GET	http://localhost:8070/show/all	Get all Shows
GET	http://localhost:8070/show/all/:id	Get specific Shows by ID
GET	http://localhost:8070/show/getone/:id	Get one Show by ID
PUT	http://localhost:8070/show/update/:id	Update seat of the show by ID
PUT	http://localhost:8070/show/updateone/:id	Update a Show by ID
DELETE	http://localhost:8070/show/delete/:id	Delete a Show by ID
PUT	http://localhost:8070/show/updatecart	Update Cart
GET	http://localhost:8070/show/analytic	Get Analytics
POST	http://localhost:8070/booking/	Create a Booking
GET	http://localhost:8070/booking/getbooking/:id	Get a Booking by ID
DELETE	http://localhost:8070/booking/:id	Delete a Booking by ID
GET	http://localhost:8070/booking/all	Get all Bookings

*Table 2: Endpoints Summary*



### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

## 3.2 Email and QR Component

### 3.2.1 Email

Nodemailer is a module for Node.js applications to allow you to send emails from your server with ease. By Using the email accounts registered with EmailEngine, you can receive and send emails. EmailEngine supports OAuth2, delayed sends, opens and clicks tracking, bounce detection, etc. All on top of regular email accounts without an external MTA service.

We used 'Nodemailer' to implement the email function in the system. In our system we send email about reservation details to our customers when they have made a successful payment.

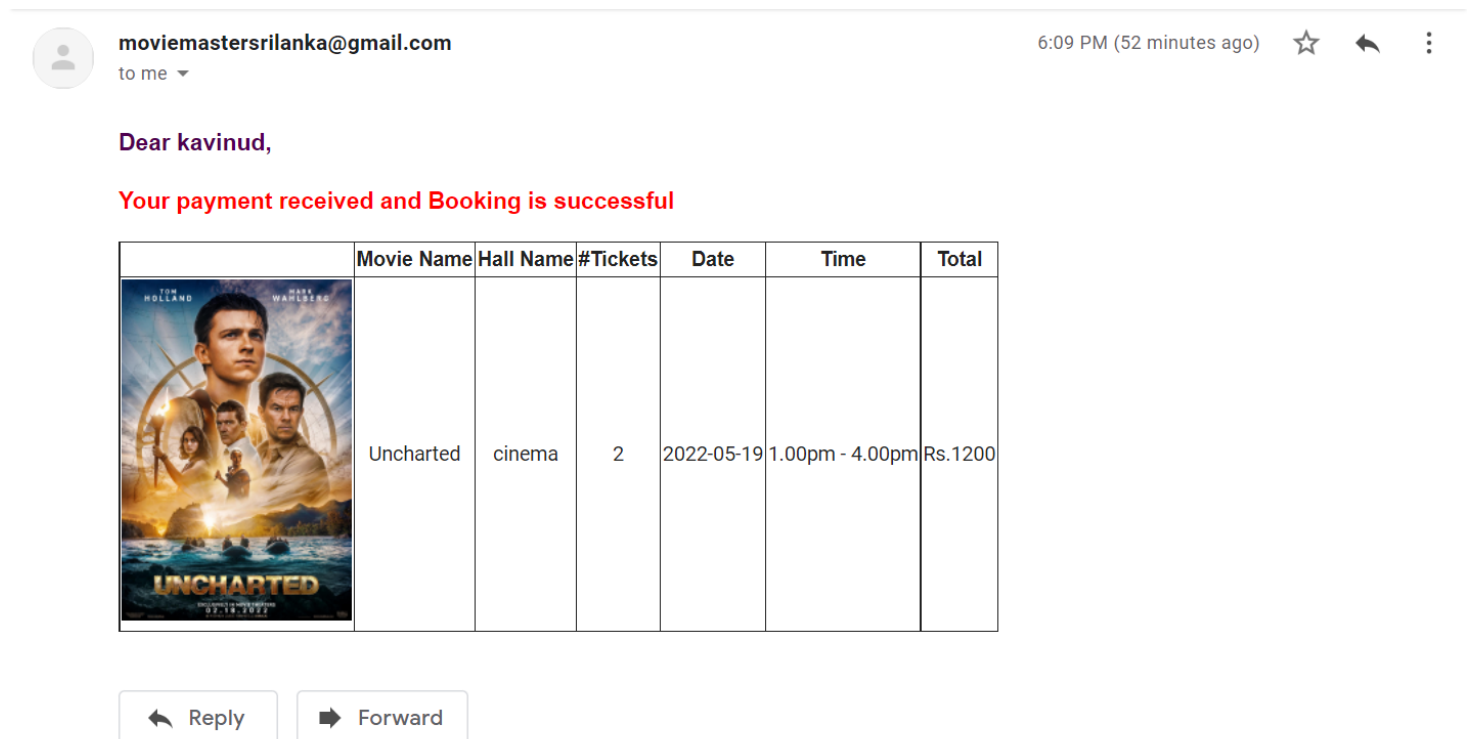


Figure 1: Booking Confirmation Email

## Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

### 3.2.2 QR

QR stands for "Quick Response". While they may look simple, QR codes are capable of storing lots of data. But no matter how much they contain, when scanned, the QR code should allow the user to access information instantly. When scanned, the unique pattern on the barcode translates into human-readable data. This transaction happens in seconds.

We used 'qrcode' and 'react-qrcode' to implement the QR code generating and scanning functionalities in our system. In our system after customers made their reservations, system issue a QR code which contains the details about the reservation. Then customers can scan the QR code to check their reservation details.

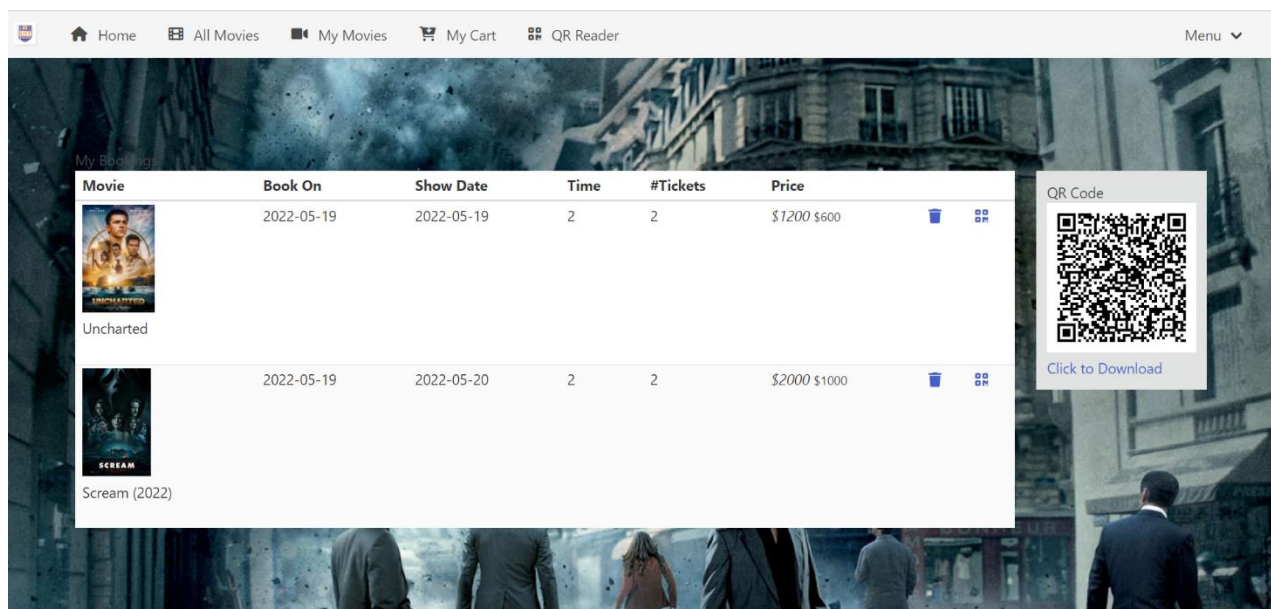


Figure 2:QR Code Generator

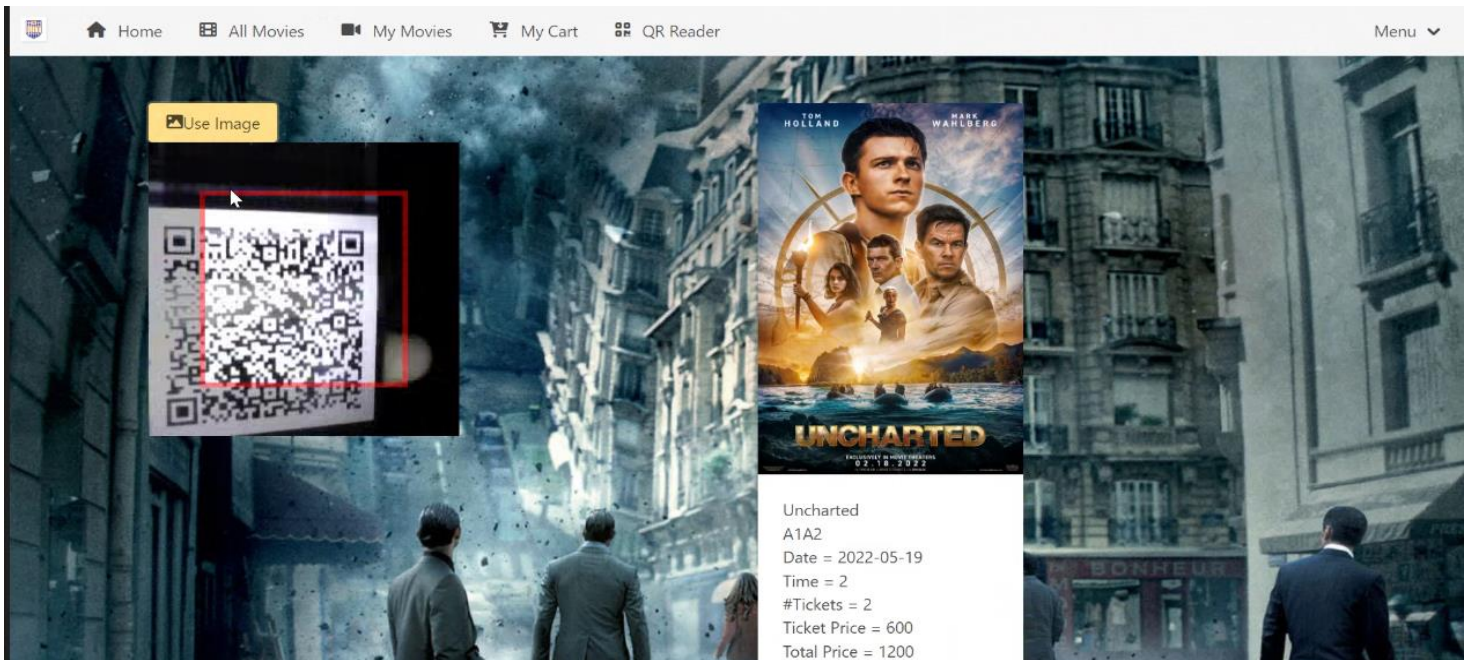


Figure 3: QR Code Scanner (Using Camera)

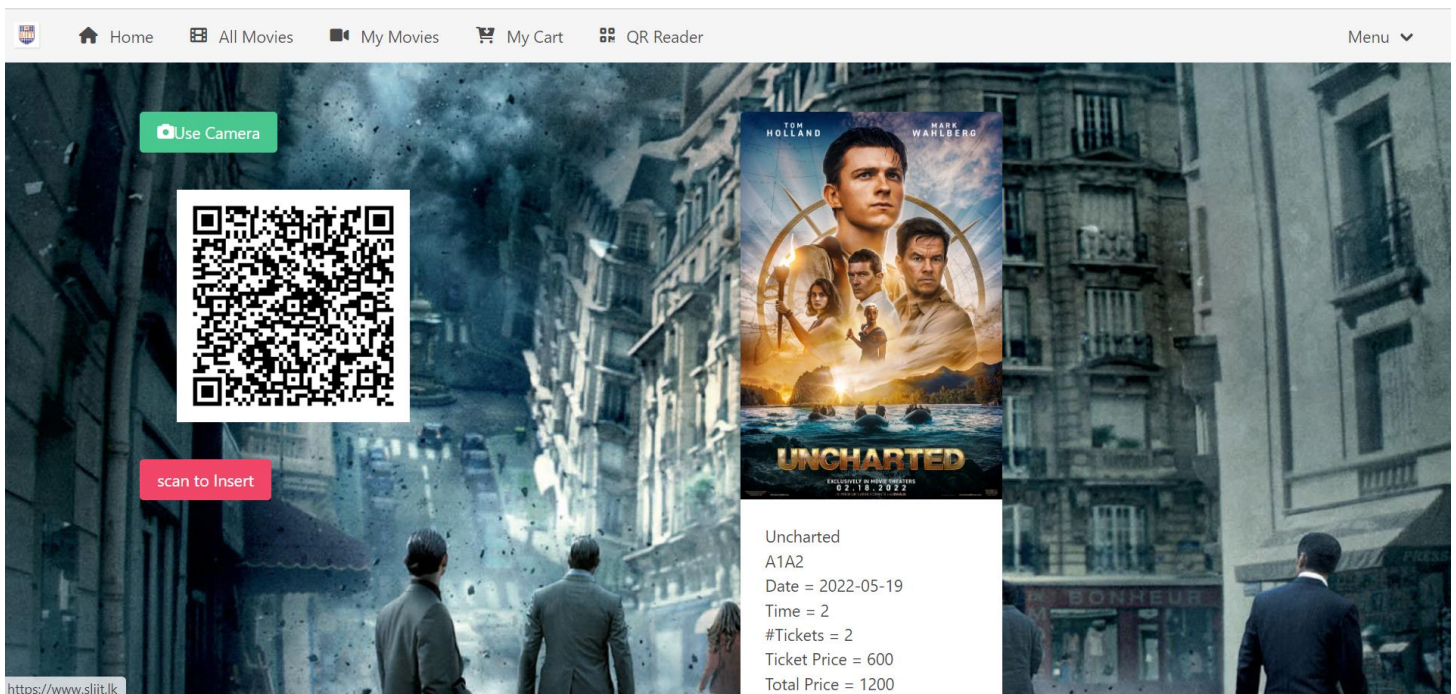


Figure 4: QR Code Scanner (Using Image)



**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****3.3 Security and Authentication**

User can only view movies without login into the site. Before being able to use other operations, user must register to the system giving email, password, and other details. When assigning a password, it must contain Upper case letter, Lowercase letter, special character and must be longer than 8 characters in total. As the security aspect of this part, the provided password will be hashed before storing it to the database by using a library called “bcrypt.js”. Thereby the actual password will be stored as an encrypted password. System has 2 levels when register, admin, and customer. Admins can access everything, but Customers have restricted access.

When a user logs into the system JSON web token are used to authenticate them. “JSON web token (jwt) is an open, industry-standard RFC 7519 method for representing claim securely between two parties”. Here the web token contains all the necessary information which is used when authenticating the user. This information will be attached to the request header before passing them to the backend.

**3.4 Validations**

When we are viewing list of shows related to the movie it will only display the shows that are scheduled for today or upcoming days. Shows that have already been finished are not going to display.

When users book tickets for a show, they can only cancel the order within 24 hours from the time that user made the payments.

**3.5 WSO2 Enterprise Integrator**

WSO2 Enterprise Integrator (WSO2 EI) is an open-source hybrid integration platform providing graphical and CLI tooling, integration runtimes, and monitoring with a variety of deployment options. The integration runtime engine is capable of playing multiple roles in your enterprise architecture. It can act as an ESB, a streaming data processor, and a microservices integrator. Deployment options include on-premises, cloud, hybrid, or a container orchestration platform of choice.

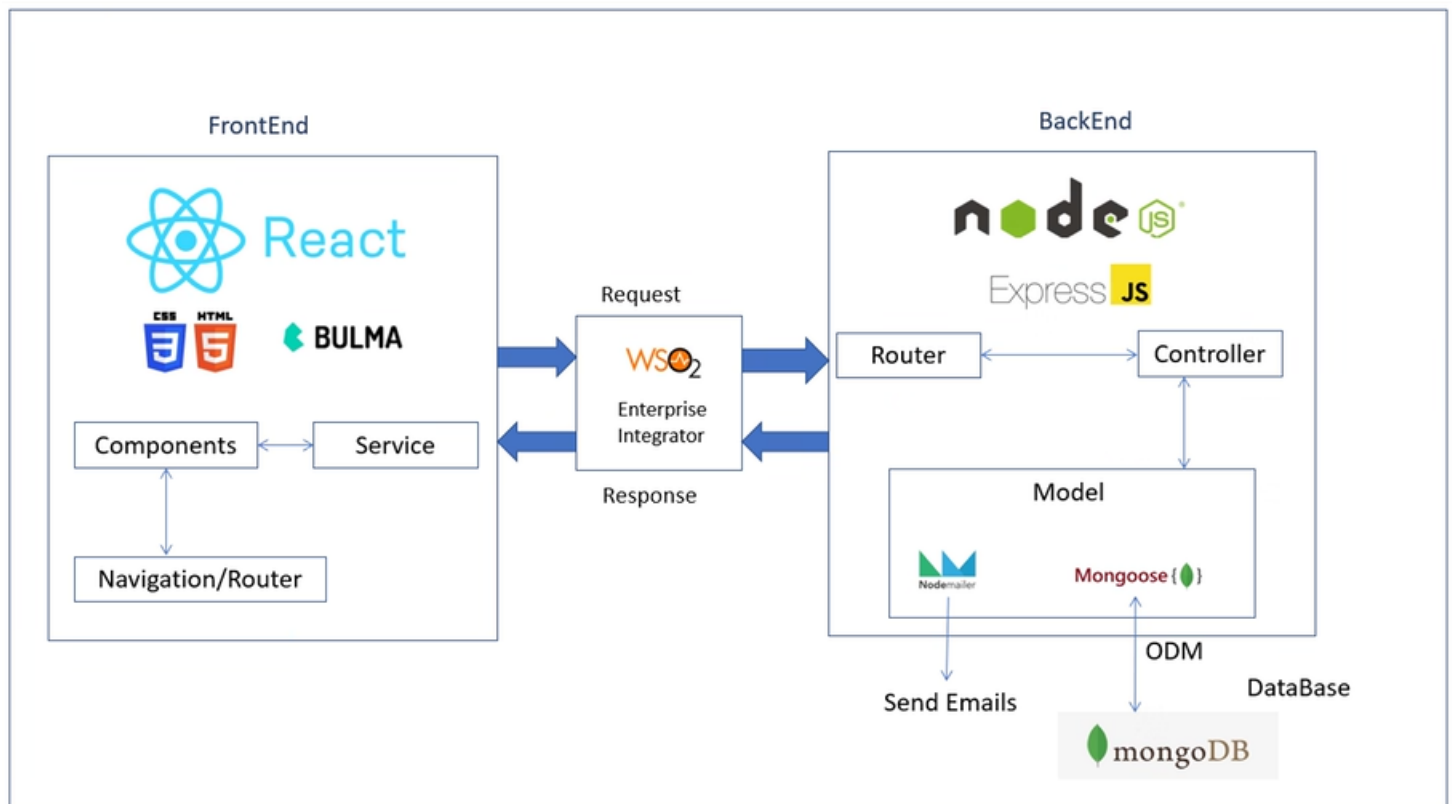
In our project we used WSO2 Enterprise Integrator manage the REST APIs.

**Assignment II – REST API Project**

**SE3020 – Distributed Systems**

**Semester 1**

**4.0 High-level Architectural diagram**



*Figure 5:High-level Architecture*

## Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

### 5.0 UseCase Diagram

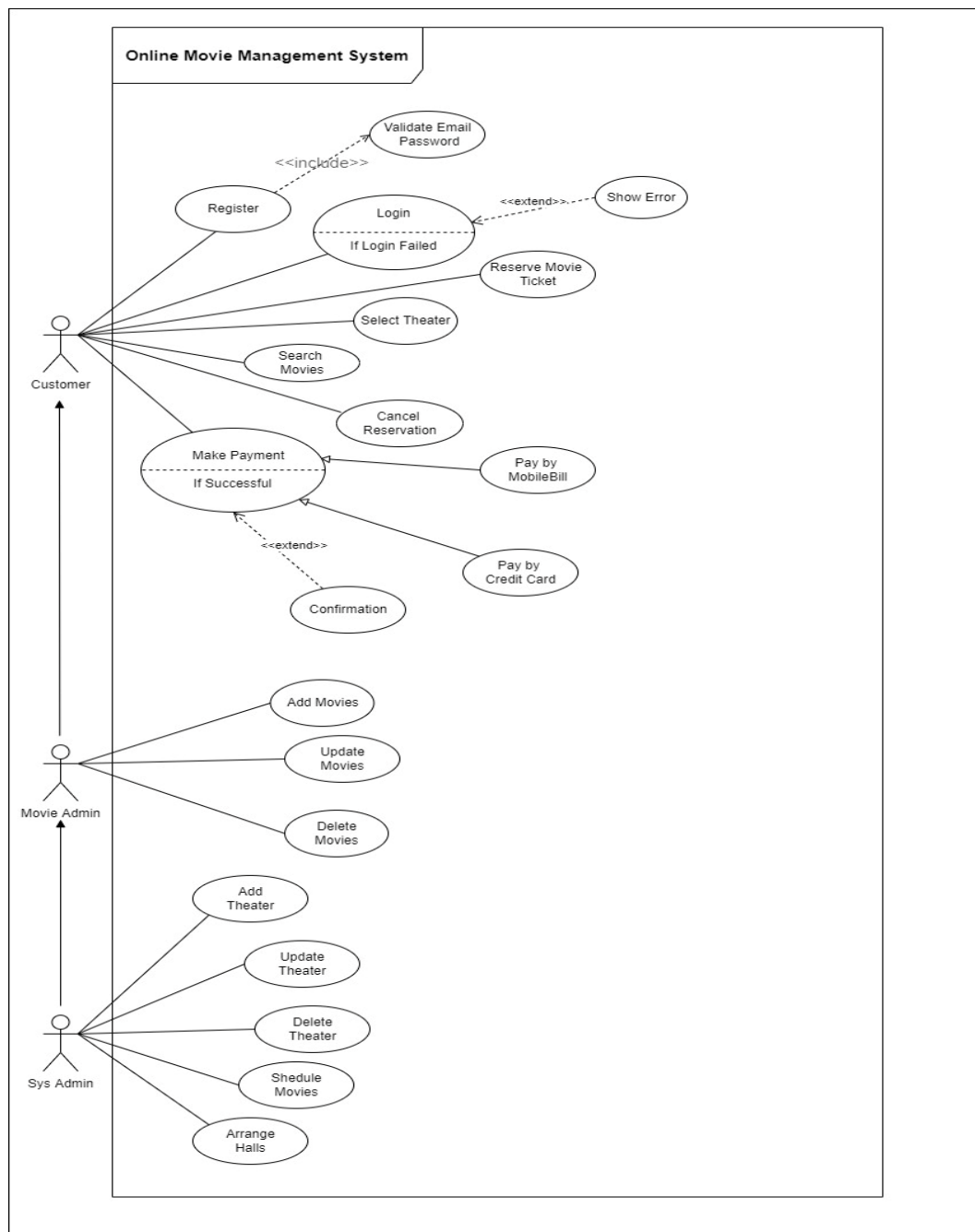


Figure 6: UseCase Diagram

## Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

### 6.0 Sequence Diagram

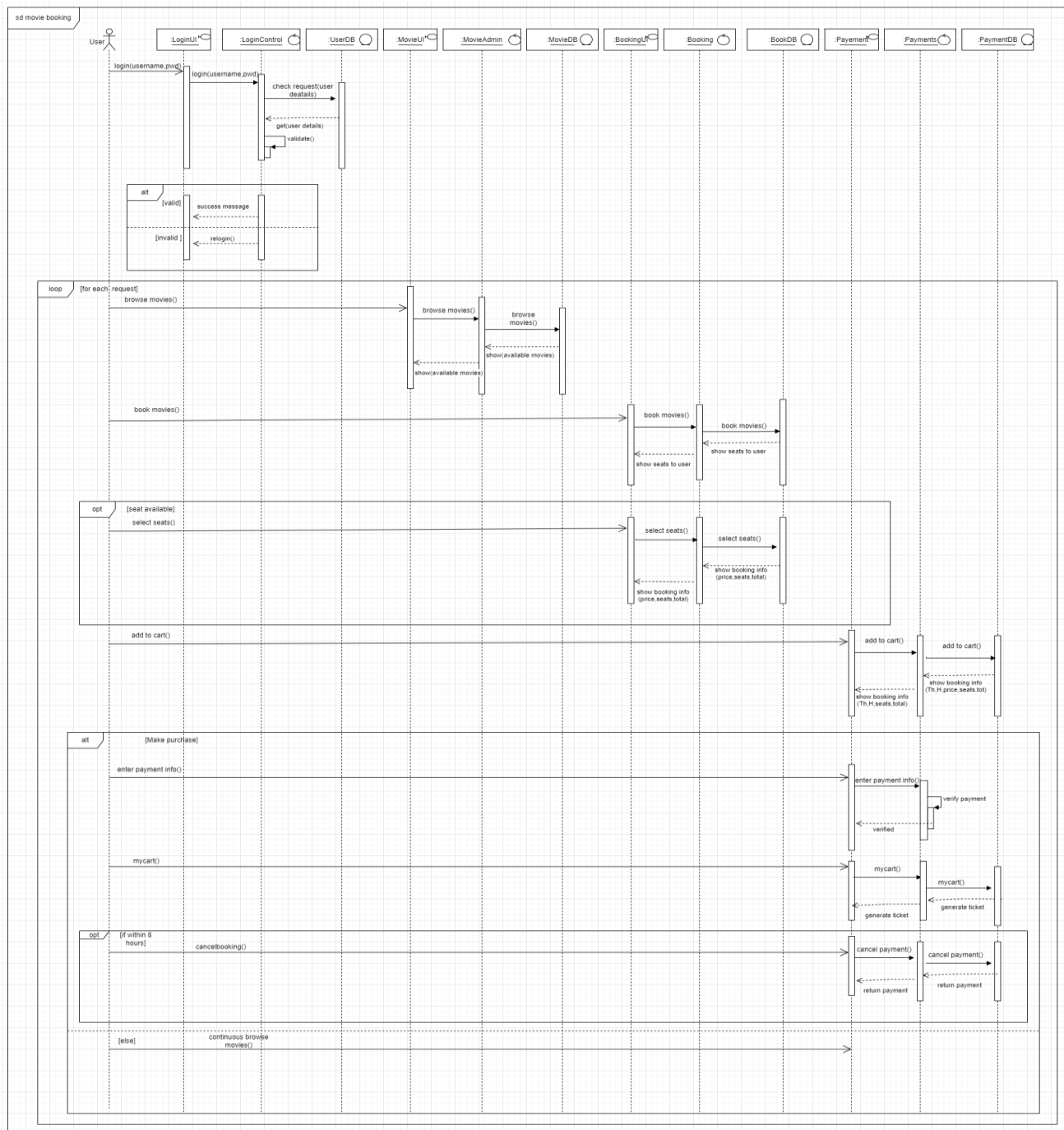


Figure 7:Sequence Diagram

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

## 7.0 Class Diagrams

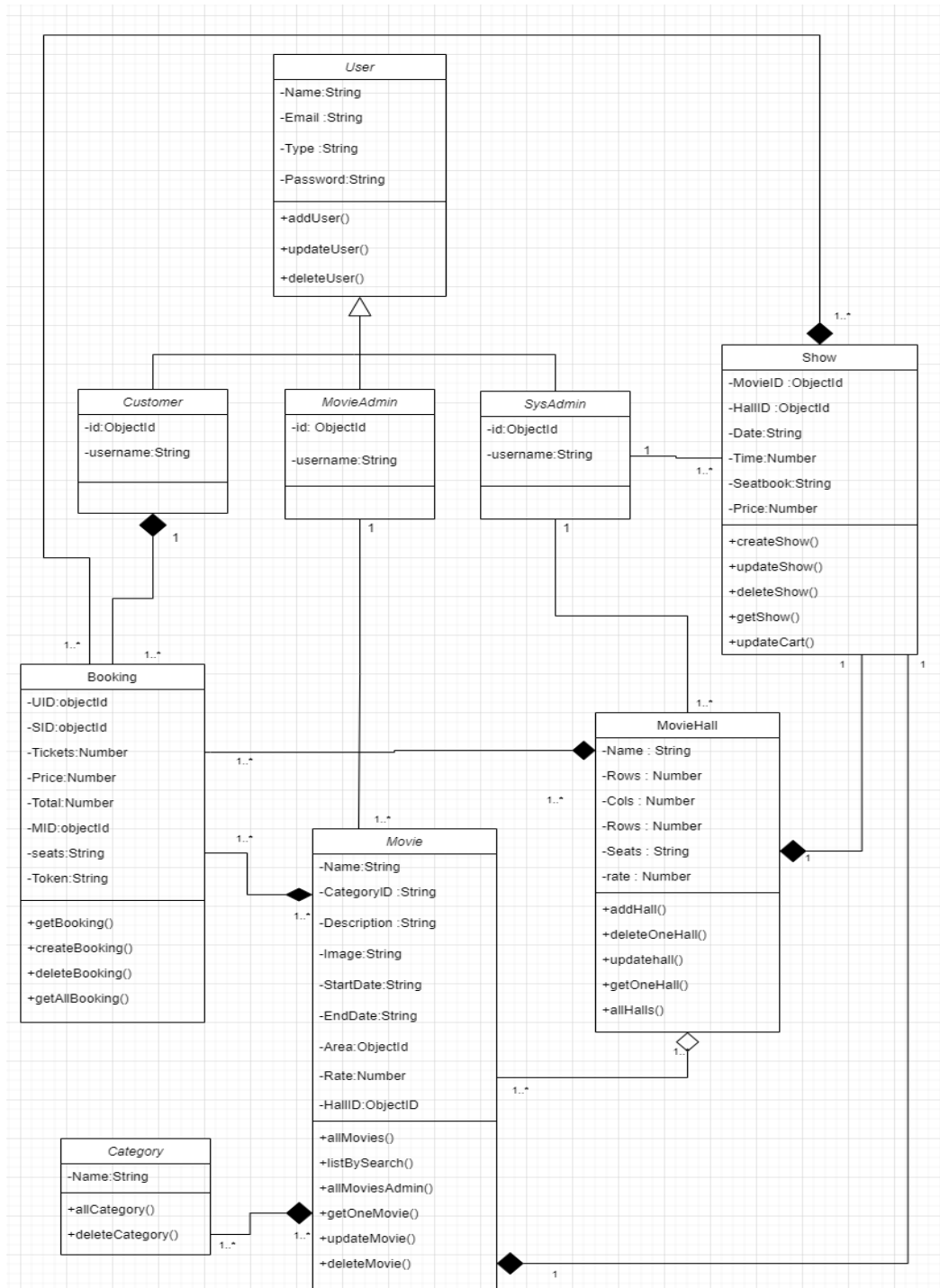


Figure 8:Class Diagram



**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.0 Appendix****8.1 BackEnd****8.1.1 Models****8.1.1.1 Booking.js**

```
const mongoose = require('mongoose')
var ObjectId = require('bson').ObjectId;
const BookingSchema = new mongoose.Schema({
  uid: {
    type: ObjectId,
    ref: "user"
  },
  sid: {
    type: ObjectId,
    ref: "show"
  },
  tickets: {
    type: Number
  },
  price: {
    type: Number
  },
  total: {
    type: Number
  },
  mid: {
    type: ObjectId,
    ref: "movie"
  },
  seats: [{ type: String }],
  token: {
    type: String
  }
}, { timestamps: true })

const Booking = mongoose.model("booking", BookingSchema);
module.exports = Booking;
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.1.1.2 Category.js**

```
const mongoose = require('mongoose')
const CategorySchema = new mongoose.Schema({
  name: {
    type: String,
    trim: true,
    required: true,
    maxlength: 32,
    unique: true
  }
}, { timestamps: true })

const Category = mongoose.model("category", CategorySchema);
module.exports = Category;
```

**8.1.1.3 Movie.js**

```
const mongoose = require('mongoose');
const schema = mongoose.Schema;
var ObjectId = require('bson').ObjectId;
const movieShema = new schema({

  name: {
    type: String,
    required: true
  },
  categoryId: {
    type: String,
  },
  description: {
    type: String
  },
  image: {
    type: String
  },
  startDate: {
    type: String
  },
},
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
endDate: {
  type: String
},
area: [{ type: ObjectId, ref: 'category' }],
rate: {
  type: Number
},
hallid: {
  type: ObjectId
}
})
const Movie = mongoose.model("movie", movieSchema);
module.exports = Movie;
```

**8.1.1.4 MovieHall.js**

```
const mongoose = require('mongoose');
const schema = mongoose.Schema;
var ObjectId = require('bson').ObjectId;
const HallSchema = new schema({

  name: {
    type: String,
    required: true
  },
  rows: {
    type: Number,

  },
  cols: {
    type: Number
  },
  seats: [{ type: String }],
  rate: {
    type: Number
  }

})

const Hall = mongoose.model("hall", HallSchema);
module.exports = Hall;
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.1.1.5 Show.js**

```
const mongoose = require('mongoose')
var ObjectId = require('bson').ObjectId;
const ShowSchema = new mongoose.Schema({
  mid: {
    type: ObjectId,
    ref: "movie"
  },
  hid: {
    type: ObjectId,
    ref: "hall"
  },
  date: {
    type: String
  },
  time: {
    type: Number
  },
  seatbook: [{ type: String }],
  price: {
    type: Number
  }
}, { timestamps: true })

const Show = mongoose.model("show", ShowSchema);
module.exports = Show;
```

**8.1.1.6 User.js**

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const UserSchema = new Schema({
  name: {
    type: String,
    required: true,
  },
  email: {
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    type: String,  
    required: true,  
  },  
  userName: {  
    type: String,  
    required: true,  
  },  
  password: {  
    type: String,  
    required: true,  
  },  
  type: {  
    type: String  
  }  
});  
  
const User = mongoose.model('user', UserSchema);  
  
module.exports = User;
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.1.2 Routes****8.1.2.1 Booking.js**

```
const { createBooking, getbookings, delbookings, getallbookings } =  
require('../Controllers/Booking');
```

```
const router = require('express').Router();
```

```
router.post('/', createBooking)  
router.get('/getbookings/:id', getbookings)  
router.delete('/:id', delbookings)  
router.get('/all', getallbookings)
```

```
module.exports = router;
```

**8.1.2.2 Category.js**

```
const { allcategory, createCategory } = require('../Controllers/Category');
```

```
const router = require('express').Router();
```

```
router.get('/list', allcategory)  
router.post("/create", createCategory)
```

```
module.exports = router;
```

**8.1.2.3 Movie.js**

```
const { addMovie, allmovies, listbysearch, getone, allmoviesadmin, updatemovie, deletemovie }  
= require("../Controllers/Movie");  
const multer = require('multer')  
const router = require("express").Router();  
const Movie = require('../Models/Movie')  
const path = require("path");  
const fs = require('fs')
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
const mongoose = require('mongoose')
const storage = multer.diskStorage({

  destination: function (req, file, cb) {
    cb(null, './uploads/');
  },
  filename: function (req, file, cb) {
    cb(null, new Date().toISOString().replace(/:/g, '-') + file.originalname);
  }
});
const filefilter = (req, file, cb) => {
  if (file.mimetype === 'image/jpeg' || file.mimetype === 'image/png') {
    cb(null, true);
  } else {
    cb(null, false)
  }
};

const upload = multer({ storage: storage, fileFilter: filefilter });

router.route("/add").post(upload.single('studentImage'), (req, res) => {
  const name = req.body.name;
  const rate = req.body.rates;
  const description = req.body.description;
  const startDate = req.body.from;
  const endDate = req.body.to;
  let areas = req.body.arr
  areas = JSON.parse(areas)
  console.log(areas)
  const area = areas.map(s => mongoose.Types.ObjectId(s));
  // const hallid = mongoose.Types.ObjectId(req.body.hid)
  const image = req.file?.path;

  const newmovie = new Movie({
    name,
    rate,
    description,
    startDate,
    endDate,
    image,
    area,
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    })

    newmovie.save().then(() => {
      res.json("student added");
    }).catch(function (err) {
      console.log(err);
    })
  })
})

router.get('/list', allmovies)
router.get('/all', allmoviesadmin)
router.post('/by/search', listbysearch)
router.get('/getone/:id', getone)
router.put('/update/:id', updatemovie)
router.delete('/delete/:id', deletemovie)

module.exports = router;
```

**8.1.2.4 MovieHall.js**

```
const { addhall, allHalls, updateHall, deleteHall, getonehall } =
require('../Controllers/MovieHall');

const router = require('express').Router();

router.post('/', addhall)
router.get('/list', allHalls)
router.get('/getone/:id', getonehall)
router.put('/update/:id', updateHall)
router.delete('/delete/:id', deleteHall)

module.exports = router;
```



**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.1.2.5 Show.js**

```
const { allshows, createShow, specificshows, getoneshow, updateshow, updatecart, analytics, deleteshow, updateoneshow } = require('../Controllers/Show');
```

```
const router = require('express').Router();
```

```
router.post('/', createShow)
router.get("/all", allshows)
router.get("/all/:id", specificshows)
router.get('/getone/:id', getoneshow)
router.put('/update/:id', updateshow)
router.put('/updateone/:id', updateoneshow)
router.delete('/delete/:id', deleteshow)
router.put('/updatecart', updatecart)
router.get('/analytic', analytics)
```

```
module.exports = router;
```

**8.1.2.6 users.js**

```
const router = require('express').Router();
const authController = require("../Controllers/user");
const isAuth = require("../middleware/auth");
```

```
router.post("/login", authController.login);
```

```
router.put("/update/:userId", isAuth, authController.update);
```

```
router.post("/signup", authController.signup,);
```

```
// get one user for update
```

```
router.get("/getuser/:userId", isAuth, authController.getUser);
router.get("/all", authController.getallusers);
router.delete("/delete/:id", authController.deleteuser);
```

```
module.exports = router;
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.1.3 Controllers****8.1.3.1 Booking.js**

```
const Booking = require('../Models/Booking')
const Show = require('../Models/Show')
const handlebars = require('handlebars')
const fs = require('fs')
const path = require('path')

var nodemailer = require('nodemailer');
const User = require('../Models/User');
exports.createBooking = async (req, res) => {
  const { cart, uid } = req.body
  const user = await User.findById({ _id: uid })
  setdata(cart, user.email, user.name);
  sendSMS({ to: "0740849017" })

  for (var i = 0; i < cart.length; i++) {
    var book = new Booking();
    book.uid = uid;
    book.sid = cart[i].sid
    book.tickets = cart[i].tickets
    book.price = cart[i].show.price
    book.mid = cart[i].mov._id
    book.seats = cart[i].seatbook

    book.save((err, data) => {
      if (err) {
        return res.status(400).json({
          error: "booking already Exists"
        })
      }
      else {
        res.json({ data })
      }
    })
  }
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    })
  }

}

exports.getbookings = (req, res) => {
  let mid = req.params.id;
  const book = Booking.find({ uid: mid }).populate("sid").populate("mid").then((sh) => {
    res.json(sh);
  }).catch((err) => {
    console.log(err);
    res.status(500).send({ status: "error in fetching", error: err.message });
  })
}

exports.delbookings = async (req, res) => {
  let mid = req.params.id;
  const booking = await Booking.findById(mid);

  const show = await Show.findById(booking.sid)
  var arr = show.seatbook
  for (var i = 0; i < booking.seats.length; i++) {
    arr = arr.filter(item => item !== booking.seats[i])
  }
  show.seatbook = arr;
  console.log(arr)
  show.save()

  const bo = await Booking.findByIdAndDelete(mid)

}

exports.getallbookings = async (req, res) => {
  try {

    const books = await Booking.find().populate("mid").populate("uid")
    res.json(books)
  }
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
} catch (error) {  
  res.status(400).json({  
    error: String(err)  
  })  
}  
}  
  
const setdata = (cart, email, name) => {  
  
  for (var i = 0; i < cart.length; i++) {  
    let item = {  
      mname: cart[i].mov.name,  
      mimage: cart[i].mov.image,  
      seats: cart[i].seatbook,  
      hall: cart[i].show.hid.name,  
      date: cart[i].show.date,  
      time: cart[i].show.time,  
      price: cart[i].show.price  
    }  
    sendmail(item, email, name)  
  }  
}  
  
const gettime = (id) => {  
  if (id === 1) {  
    return "10.00am - 1.00pm"  
  }  
  else if (id === 2) {  
    return "1.00pm - 4.00pm"  
  }  
  else {  
    return "4.00pm - 7.00pm"  
  }  
}  
  
const sendmail = (ob, email, name) => {  
  const filePath = path.join(__dirname, '../middleware/template.html');  
  const source = fs.readFileSync(filePath, 'utf-8').toString();  
  const template = handlebars.compile(source);  
  const replacements = {
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    uname: name,
    mname: ob.mname,
    hname: ob.hall,
    tic: ob.seats.length,
    date: ob.date,
    time: gettime(ob.time),
    Total: (ob.seats.length * ob.price)

};
const htmlToSend = template(replacements);

var transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: 'moviemastersrilanka@gmail.com',
    pass: 'movie22@'
  }
});

var mailOptions = {
  from: 'moviemastersrilanka@gmail.com',
  to: email,
  subject: 'Sending Email using Node.js',
  text: 'That was easy!',
  attachments: [{
    filename: ob.mimage,
    path: __dirname + '/../' + ob.mimage,
    cid: 'imagenamename'
  }],
  html: htmlToSend
};

transporter.sendMail(mailOptions, function (error, info) {
  if (error) {
    console.log(error);
  } else {
    console.log('Email sent: ' + info.response);
  }
});
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

}

```
const sendSMS = (smsObj) => {  
  // Your Account Sid and Auth Token from twilio.com/console  
  const accountSid = "AC765e6daabdbbee2af8e1b7b07309a4b5d";  
  const authToken = "b018f6ebf4cbad8f2f2fb6009ea5bfdf";  
  const client = require("twilio")(accountSid, authToken);  
  
  client.messages  
    .create({  
      body:  
        "Dear Customer,Your payment is succesfully received... Your Booking ID is  
1159075. Hope you enjoy this service.",  
      from: "+13393090187",  
      to: smsObj.to.replace("0", "+94")  
    })  
    .then(message =>  
      console.log(message.sid + " Message has been successfully send")  
    ).catch((err) => {  
      console.log(err)  
    });  
}
```

**8.1.3.2 Category.js**

```
const Category = require('../Models/Category')  
  
exports.allcategory = (req, res) => {  
  Category.find().exec((err, result) => {  
    if (err) {  
      return res.status(400).json({  
        error: String(err)  
      })  
    }  
    res.json(result)  
  })  
}  
  
exports.createCategory = (req, res) => {
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
const category = new Category(req.body)
category.save((err, data) => {
  if (err) {
    return res.status(400).json({
      error: "category already Exists"
    })
  }
  else {
    res.json({ data })
  }
})
}
```

**8.1.3.3 Movie.js**

```
const Movie = require('../Models/Movie')
const Booking = require('../Models/Booking')
const Show = require('../Models/Show')
var ObjectId = require('bson').ObjectId;
exports.allmovies = (req, res) => {
  Movie.find().exec((err, result) => {
    if (err) {
      return res.status(400).json({
        error: String(err)
      })
    }
    res.json(result)
  })
}

exports.allmoviesadmin = (req, res) => {
  Movie.find().populate("area").exec((err, result) => {
    if (err) {
      return res.status(400).json({
        error: String(err)
      })
    }
    res.json(result)
  })
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
//method that used to filter movies according to user selection
exports.listbysearch = (req, res) => {
  let order = req.query.order ? req.query.order : 'desc'
  let sortBy = req.query.sortBy ? req.query.sortBy : '_id'
  var limit
  if (req.body.limit) {
    limit = parseInt(req.body.limit)
  } else {
    limit = 100
  }
  //const isSupervisor = req.body.isSupervisor;
  let skip = parseInt(req.body.skip)
  var ind = 0;
  let findArgs = {}
  // findArgs["isSupervisor"] = isSupervisor;

  for (let key in req.body.filters) {
    if (req.body.filters[key].length > 0) {
      if (key === 'groups') {

        findArgs["rate"] = {
          $gt: req.body.filters["groups"][0],
          $lte: req.body.filters["groups"][1]
        };
      } else {
        let arr = []
        arr = req.body.filters["area"];

        findArgs["area"] = {
          $in: arr
        }
      }

      // findArgs1 = {
```



**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
//      area: req.body.filters[key]
// }
    }
  }
}

Movie.find(findArgs)

    .sort([[sortBy, order]])
    .skip(skip)
    .limit(limit)
    .exec((err, data) => {
      if (err) {
        return res.status(400).json({
          error: String(err)
        })
      }
      res.json({
        size: data.length,
        data
      })
    })
}

exports.getone = (req, res) => {
  let mid = req.params.id;
  const movie = Movie.findById(mid).populate("area").then((mov) => {
    res.json(mov);
  }).catch((err) => {
    console.log(err);
    res.status(500).send({ status: "error in fetching", error: err.message });
  })
}

exports.updatemovie = async (req, res) => {
  let mid = req.params.id;
  const { description, rate } = req.body;
  try {
    await Movie.findOneAndUpdate({ _id: mid }, { description, rate }).then(data => {
      res.send({ status: "movie updated" })
    })
  }
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
}).catch(err => {

})

} catch (error) {
  res.status(500).send({ status: "error in updating data", error: error.message });
}
}

exports.deletemovie = async (req, res) => {
  let delid = req.params.id;
  try {
    await Movie.findByIdAndDelete(delid);
    await Booking.deleteMany({ mid: delid })
    await Show.deleteMany({ mid: delid })
    res.send("deleted")
  } catch (err) {
    res.status(500).send({ status: "error in deleting data", error: err.message });
  }
}
```

**8.1.3.4 MovieHall.js**

```
const Hall = require('../Models/MovieHall')
const Show = require('../Models/Show')

exports.addhall = (req, res) => {
  const hall = new Hall(req.body)

  let seats = []
  let alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';

  for (var i = 0; i < hall.rows; i++) {
    for (var j = 0; j < hall.cols; j++) {
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
        let se = alphabet.charAt(i) + String(j + 1)
        seats.push(se)
    }
}
hall.seats = seats;
hall.save((err, data) => {
    if (err) {
        return res.status(400).json({
            error: "hall already Exists"
        })
    }
    else {
        res.json({ data })
    }
})
}

exports.allHalls = (req, res) => {
    Hall.find().exec((err, result) => {
        if (err) {
            return res.status(400).json({
                error: String(err)
            })
        }
        res.json(result)
    })
}

exports.updateHall = async (req, res) => {
    let mid = req.params.id;
    const { rate, name } = req.body;
    try {
        await Hall.findOneAndUpdate({ _id: mid }, { rate, name }).then(data => {
            res.send({ status: "hall updated" })
        }).catch(err => {

        })
    }

} catch (error) {
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    res.status(500).send({ status: "error in updating data", error: error.message });
  }
}

exports.deleteHall = async (req, res) => {
  let delid = req.params.id;
  try {
    await Hall.findByIdAndDelete(delid);
    await Show.deleteMany({ hid: delid })
    res.send("deleted")
  } catch (err) {
    res.status(500).send({ status: "error in deleting data", error: err.message });
  }
}

exports.getonehall = (req, res) => {
  let mid = req.params.id;
  const movie = Hall.findById(mid).then((mov) => {
    res.json(mov);
  }).catch((err) => {
    console.log(err);
    res.status(500).send({ status: "error in fetching", error: err.message });
  })
}
```

**8.1.3.5 Show.js**

```
const Show = require('../Models/Show')
const moment = require('moment')
const Movie = require('../Models/Movie')
const User = require('../Models/User')
const Hall = require('../Models/MovieHall')
const Booking = require('../Models/Booking')
const Category = require('../Models/Category')
exports.allshows = (req, res) => {
  Show.find().populate("mid").populate("hid").exec((err, result) => {
    if (err) {
      return res.status(400).json({
        error: String(err)
      })
    }
  })
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    })
  }
  res.json(result)
})
}

exports.createShow = (req, res) => {
  const show = new Show(req.body)

  show.save((err, data) => {
    if (err) {
      return res.status(400).json({
        error: "show already Exists"
      })
    }
    else {
      res.json({ data })
    }
  })
}

exports.specificshows = (req, res) => {
  let mid = req.params.id;
  var resu = []
  const shows = Show.find({ mid: mid }).then((movs) => {
    for (var i = 0; i < movs.length; i++) {
      const d1 = moment(movs[i].date, "YYYY MM DD")
      var current = moment().startOf('day');
      var dif = moment.duration(d1.diff(current)).asDays()
      if (dif >= 0) {
        resu.push(movs[i])
      }
    }
    res.json(resu);
  }).catch((err) => {
    console.log(err);
    res.status(500).send({ status: "error in fetching", error: err.message });
  })
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
exports.getoneshow = (req, res) => {
  let mid = req.params.id;
  const show = Show.findById(mid).populate("hid").populate("mid").then((sh) => {
    res.json(sh);
  }).catch((err) => {
    console.log(err);
    res.status(500).send({ status: "error in fetching", error: err.message });
  })
}

exports.updateshow = (req, res) => {
  let showid = req.params.id;
  const { seatbook } = req.body;
  const updateshow = {
    seatbook
  }

  const update = Show.findByIdAndUpdate(showid, updateshow).then(() => {
    res.status(200).send({ status: "show updated" })
  }).catch((err) => {
    console.log(err);
    res.status(500).send({ status: "error in updating data", error: err.message });
  })
}

exports.updatecart = (req, res) => {
  const { cart } = req.body;

  for (var i = 0; i < cart.length; i++) {
    const update = Show.findByIdAndUpdate({ _id: cart[i].sid }, {
      $push: { seatbook: cart[i].seatbook }
    }).then(() => {
      res.status(200).send({ status: "show updated" })
    }).catch((err) => {
      console.log(err);
      res.status(500).send({ status: "error in updating data", error: err.message });
    })
  }
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
}

exports.analytics = async (req, res) => {
  const scount = await Show.count();
  const hcount = await Hall.count();
  const ucount = await User.count();
  const mcount = await Movie.count();
  const bcount = await Booking.count();
  const ccount = await Category.count();

  const analytic = {
    scount,
    hcount,
    ucount,
    mcount,
    bcount,
    ccount
  }

  res.status(200).send(analytic)
}

exports.deleteshow = async (req, res) => {
  let delid = req.params.id;
  try {
    await Show.findByIdAndDelete(delid);
    await Booking.deleteMany({ sid: delid })
  } catch (err) {
    res.status(500).send({ status: "error in deleting data", error: err.message });
  }
}

exports.updateoneshow = async (req, res) => {
  let showid = req.params.id;
  const { date, time, price } = req.body;
  try {
    await Show.findOneAndUpdate({ _id: showid }, { date, time, price }).then(data => {
      res.send({ status: "show updated" })
    })
  }
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
}).catch(err => {
```

```
})
```

```
} catch (error) {  
  res.status(500).send({ status: "error in updating data", error: error.message });  
}  
}
```

**8.1.3.6 user.js**

```
const bcrypt = require("bcryptjs");  
const jwt = require("jsonwebtoken");  
const Booking = require('../Models/Booking')  
const User = require("../Models/User");  
  
//add  
exports.signup = async (req, res, next) => {  
  
  const email = req.body.email;  
  const userName = req.body.userName;  
  const name = req.body.name;  
  const password = req.body.password;  
  const type = req.body.type;  
  try {  
    const checkuser = await User.findOne({ email: email });  
    if (checkuser) {  
      const error = new Error("Email exsisted");  
      error.statusCode = 500;  
      throw error;  
    }  
    const hashedPw = await bcrypt.hash(password, 12);  
    const user = new User({  
      email: email,  
      password: hashedPw,  
      userName: userName,
```



**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    name: name,
    type: type
  });
  const result = await user.save();
  res.status(201).json({ message: "User created!", userId: result._id });
} catch (err) {
  if (!err.statusCode) {
    err.statusCode = 500;
  }
  next(err);
}
};
```

//update

```
exports.update = async (req, res, next) => {

  let userID = req.params.userId;
  const { name, email, userName, password } = req.body;
  const hashedPw = await bcrypt.hash(password, 12);

  const updateUser = {
    name,
    email,
    userName,
    password: hashedPw,
  };

  const update = await User.findByIdAndUpdate(userID, updateUser)
    .then(() => {
      res.status(200).send({ status: 'User Updated' });
    })
    .catch((err) => {
      console.log(err);
      res
        .status(500)
        .send({ status: 'Error with updating data', error: "error" });
    });
};
```

//get one user

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
exports.getUser = async (req, res, next) => {  
  const userId = req.params.userId;  
  try {  
    const user = await User.findById(userId);  
    if (!user) {  
      const error = new Error("Could not find user.");  
      error.statusCode = 404;  
      throw error;  
    }  
    res.status(200).json({ message: "user fetched.", user: user });  
  } catch (err) {  
    if (!err.statusCode) {  
      err.statusCode = 500;  
    }  
    next(err);  
  }  
};
```

```
//login check validations  
exports.login = async (req, res, next) => {  
  //const email = req.body.email;  
  let filter  
  if (req.body.email) {  
    if (req.body.type) {  
      filter = {  
        email: req.body.email,  
        type: req.body.type  
      }  
    }  
    else {  
      filter = {  
        email: req.body.email  
      }  
    }  
  }  
  else {  
    filter = { userName: req.body.userName }  
  }  
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
const password = req.body.password;
try {
  const user = await User.findOne(filter);
  if (!user) {
    const error = new Error("A user with this email could not be found.");
    error.statusCode = 401;
    throw error;
  }
  const isEqual = await bcrypt.compare(password, user.password);
  if (!isEqual) {
    const error = new Error("Wrong password!");
    error.statusCode = 401;
    throw error;
  }

  const token = jwt.sign(
    {
      email: user.email,
      name: user.name,
      userId: user._id.toString(),
    },
    "somesupersecretsecret",
    { expiresIn: "1h" },
  );
  res.status(200).json({ token: token, user: user });
} catch (err) {
  if (!err.statusCode) {
    err.statusCode = 500;
  }
  next(err);
}
};

exports.getallusers = async (req, res) => {
  try {
    const nusers = await User.find({ type: "" })
    const ausers = await User.find({ type: "ma" })
    const users = {
      nusers,
      ausers
    }
  }
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
res.json(users)

} catch (error) {
  res.status(400).json({
    error: String(err)
  })
}

}

exports.deleteuser = async (req, res) => {
  let delid = req.params.id;
  try {
    await User.findByIdAndDelete(delid);
    await Booking.deleteMany({ uid: delid })
  } catch (err) {
    res.status(500).send({ status: "error in deleting data", error: err.message });
  }

}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.1.4 Middleware****8.1.4.1 auth.js**

```
const jwt = require("jsonwebtoken");

module.exports = (req, res, next) => {
  const authHeader = req.get("Authorization");
  if (!authHeader) {
    const error = new Error("Not authenticated.");
    error.statusCode = 401;
    throw error;
  }
  const token = authHeader.split(" ")[1];
  let decodedToken;
  try {
    decodedToken = jwt.verify(token, "somesupersecretsecret");
  } catch (err) {
    err.statusCode = 500;
    throw err;
  }
  if (!decodedToken) {
    const error = new Error("Not authenticated.");
    error.statusCode = 401;
    throw error;
  }
  req.userId = decodedToken;
  next();
};
```

**8.1.4.2 template.html**

```
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
<title>Document</title>
<style>
  img {
    width: 150px;
  }

  table,
  th,
  td {
    border: 1px solid;
    text-align: center
  }

  table {
    border-collapse: collapse;
  }

  p {
    color: red;
  }
</style>
</head>

<body>
  <h3>Dear {{uname}},<p> Your payment received and Booking is successful</p>
</h3>
  <table>
    <tr>
      <th></th>
      <th>Movie Name</th>
      <th>Hall Name</th>
      <th>#Tickets</th>
      <th>Date</th>
      <th>Time</th>
      <th>Total</th>
    </tr>
    <tr>
      <td>
      </td>
      <td>{{mname}}</td>
      <td>{{hname}}</td>
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
<td>{{tic}}</td>
<td>{{date}}</td>
<td>{{time}}</td>
<td>Rs.{{Total}}</td>
</tr>
</table>

</body>

</html>
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.1.5 Server.js**

```
////Declare the variables
const express = require('express');
const mongoose = require('mongoose');
const dotenv = require('dotenv');
const app = express();
require('dotenv').config();
const PORT = process.env.PORT || 8070;
const URL = process.env.MONGODB_URL;
app.use('/uploads', express.static('uploads'));
const MovieRouter = require('./Routes/Movie')
const CategoryRouter = require('./Routes/Category')
const HallRouter = require('./Routes/MovieHall')
const ShowRouter = require('./Routes/Show')
const BookingRouter = require('./Routes/Booking')
const UserRouter = require('./Routes/users')
const cors = require('cors');
const bodyParser = require('body-parser');
app.use(cors());
app.use(bodyParser.json());

////connect to mongoDB
mongoose.connect(URL, (err) => {
  if (err) throw err;
  console.log('connected to MongoDB');
});

app.use('/movies', MovieRouter)
app.use('/categories', CategoryRouter)
app.use('/hall', HallRouter)
app.use('/show', ShowRouter)
app.use('/booking', BookingRouter)
app.use('/user', UserRouter);
////create server with port numebr
app.listen(PORT, () => {
  console.log(`service is up and running on port ${PORT}`);
});
```



**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.1.6 package.json**

```
{  
  "name": "backend",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "start": "node server.js",  
    "dev": "nodemon server.js"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC",  
  "dependencies": {  
    "bcryptjs": "^2.4.3",  
    "body-parser": "^1.20.0",  
    "cors": "^2.8.5",  
    "dotenv": "^16.0.0",  
    "express": "^4.17.3",  
    "handlebars": "^4.7.7",  
    "jsonwebtoken": "^8.5.1",  
    "moment": "^2.29.3",  
    "mongoose": "^6.2.9",  
    "multer": "^1.4.4",  
    "nodemailer": "^6.7.5",  
    "nodemon": "^2.0.15",  
    "twilio": "^3.77.0"  
  }  
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.2 FrontEnd****8.2.1 Components****8.2.1.1 Admin****8.2.1.1.1 Addadmin.js**

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

function Addadmin(props) {
  const [name, setname] = useState("")
  const [email, setemail] = useState("")
  const [userName, setuserName] = useState("")
  const [password, setpassword] = useState("")
  const [confpassword, setconfpassword] = useState("")
  const [users, setUsers] = useState([])
  useEffect(() => {

  }, [])

  function sendData(e) {
    e.preventDefault();

    if (confpassword === password) {
    }
    else {
      return (alert("Password Doesn't match"))
    }
    const type = "ma";
    const newUser = {
      name,
      email,
      userName,
```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

        password,
        type
    };

    axios
    .post('http://localhost:8070/user/signup', newUser)
    .then(() => {
        alert('admin Create Successfully ');
        setemail('');
        setname('');
        setUsername('');
        setPassword('');
        setconfpassword('');
        window.location.href = '/users'
    })
    .catch((err) => {
        alert("Email Already Exists");
        console.log(err)
    });
}

return (
    <>
    <div className='home-div'> </div>

    <section className="hero is-fullheight-with-navbar all-home-flexs">

        <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
            <div className="container is-widescreen">
                <div className='content '>
                    <div className='columns is-justify-content-center'>
                        <div className='column is-3'>
                            <h2 className="subtitle is-2 is-semibold has-text-white
has-text-centered">Create Movie Admin</h2>
                            <form onSubmit={sendData}>
                                <div className="mb-3">
                                    <input type="text" className="input is-primary"
placeholder="Full Name" onChange={(e) => { setname(e.target.value) }} aria-describedby="basic-
addon1" required />

                                </div>
                                <div className="mb-3">

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        <input type="email" className="input is-primary"
placeholder="Email" onChange={(e) => { setemail(e.target.value) }} aria-describedby="basic-
addon1" required />

    </div>
    <div className="mb-3">
        <input type="text" className="input is-primary"
placeholder="Username" onChange={(e) => { setUsername(e.target.value) }} aria-
describedby="basic-addon1" required />
    </div>
    <div className="mb-3">
        <input type="password" pattern="(?=.*\d)(?=.*[a-
z])(?=.*[A-Z]).{8,}" className="input is-primary" placeholder="Password" onChange={(e) => {
setpassword(e.target.value) }} aria-describedby="basic-addon1" required />
    </div>
    <div className="mb-3">
        <input type="password" pattern="(?=.*\d)(?=.*[a-
z])(?=.*[A-Z]).{8,}" className="input is-primary" placeholder="Confirm Password" onChange={(e)
=> { setconfpassword(e.target.value) }} aria-describedby="basic-addon1" required />
    </div>
    {/ * <PasswordChecklist
    rules=[["minLength", "specialChar", "number", "capital", "match"]]
    minLength={5}
    value={password}
    valueAgain={confpassword}
    onChange={({isValid) => { }}
    /> */}

    <div>
        <input type="submit" value="Create Admin"
className="button is-link is-fullwidth" />

        <Link to="/users"> <button type="submit"
className="button is-success is-fullwidth mt-2">Back to List</button></Link>
    </div>

    </form>
</div>
</div>
</div>

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
        </div>
      </div>
    </section>

  </>
);
}
```

```
export default Addadmin;
```

**8.2.1.1.2 Addcategory.js**

```
import axios from 'axios';
import React, { useState } from 'react';
import { Link } from 'react-router-dom';
import NavBar_Admin from '../Home/NavBar_Admin';
//import { message } from 'antd'

function Addcategories(props) {
  const [name, setname] = useState("")

  const createcate = (e) => {
    e.preventDefault()

    const cate = {
      name,
    }

    axios.post('http://localhost:8070/categories/create', cate).then((data) => {
      console.log("created")
      alert("category Added")
      window.location = "/allcategory"
    }).catch((err) => {
      console.log(err)
    })
  }

  return (
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```
<>
{localStorage.getItem('atoken') ? (
  <>
    <NavBar_Admin />

    <section className="hero is-fullheight-with-navbar admin-div">
      <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
        <div className="container is-widescreen">
          <div className='columns is-justify-content-center'>
            <div className='column is-6'>
              <div className="card" style={{ backgroundColor:
'#fff6' }}>

                <div className='title is-2 has-text-danger-dark
has-text-centered has-background-danger-light pb-3 pt-2'>
                  <b>Adding Category</b>
                </div>
                <form onSubmit={createcate}>
                  <div className="field has-addons m-3 pt-3">
                    <div className="control is-expanded">
                      <div className="is-fullwidth">
                        <input type="text"
className="input" onChange={(e) => { setname(e.target.value) }} placeholder="Name" required />
                      </div>
                    </div>
                    <div className="button control is-static">
                      <span className="icon is-left">
                        <i className="fa fa-film"></i>
                      </span>
                      <div type="submit" className=" is-
primary is-static mr-6 pr-3">Name</div>
                    </div>
                  </div>
                </div>

                <div className=" has-background-danger-light
pt-1 pb-2">
                  <div className=" mt-5 mr-3 ml-3 pb-3 ">
                    <Link to='/allcategory'><button
className="button is-link is-fullwidth mb-2" type='submit' value="Create" >Back to
categories</button></Link>
```

## Assignment II – REST API Project

## SE3020 – Distributed Systems

## Semester 1

```

        <button className="button is-danger
is-fullwidth " type="submit" value="create" >Create</button>
      </div>
    </div>
  </form>
</div>
</div>
</div>
</div>
</div>
</section>
</>
) : (
  window.location = "/"
)}
</>
);
}

export default Addcategories;
```

#### 8.1.1.1.3 Adduser.js

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

function Adduser(props) {
  const [name, setname] = useState("")
  const [email, setemail] = useState("")
  const [userName, setuserName] = useState("")
  const [password, setpassword] = useState("")
  const [confpassword, setconfpassword] = useState("")
  const [users, setUsers] = useState([])
  useEffect(() => {
    }, [])
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
function sendData(e) {
  e.preventDefault();

  if (confpassword === password) {

  }
  else {
    return (alert("Password Doesn't match"))
  }
  const newUser = {
    name,
    email,
    userName,
    password,
    type: ""
  };

  axios
    .post('http://localhost:8070/user/signup', newUser)
    .then(() => {
      alert('User Create Successfully ');
      setemail('');
      setname('');
      setUsername('');
      setPassword('');
      setconfpassword('');
      window.location.href = '/users'
    })
    .catch((err) => {
      alert("Email Already Exists");
      console.log(err)
    });
}
return (
  <>
  <div className='home-div'> </div>

  <section className="hero is-fullheight-with-navbar all-home-flexs">

    <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
      <div className="container is-widescreen">
```



### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    <div className='content '>
      <div className='columns is-justify-content-center'>
        <div className='column is-3'>
          <h2 className="subtitle is-2 is-semibold has-text-white
has-text-centered">Create User</h2>
          <form onSubmit={sendData}>
            <div className="mb-3">
              <input type="text" className="input is-primary"
placeholder="Full Name" onChange={(e) => { setname(e.target.value) }} aria-describedby="basic-
addon1" required />
            </div>
            <div className="mb-3">
              <input type="email" className="input is-primary"
placeholder="Email" onChange={(e) => { setemail(e.target.value) }} aria-describedby="basic-
addon1" required />
            </div>
            <div className="mb-3">
              <input type="text" className="input is-primary"
placeholder="Username" onChange={(e) => { setUsername(e.target.value) }} aria-
describedby="basic-addon1" required />
            </div>
            <div className="mb-3">
              <input type="password" pattern="(?=.*\d)(?=.*[a-
z])(?=.*[A-Z]).{8,}" className="input is-primary" placeholder="Password" onChange={(e) => {
setpassword(e.target.value) }} aria-describedby="basic-addon1" required />
            </div>
            <div className="mb-3">
              <input type="password" pattern="(?=.*\d)(?=.*[a-
z])(?=.*[A-Z]).{8,}" className="input is-primary" placeholder="Confirm Password" onChange={(e)
=> { setconfpassword(e.target.value) }} aria-describedby="basic-addon1" required />
            </div>
            /* <PasswordChecklist
rules=[["minLength", "specialChar", "number", "capital", "match"]]
minLength={5}
value={password}
valueAgain={confpassword}
onChange={(isValid) => { }}
/> */
          </form>
        </div>
      </div>
    </div>
  
```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

        <input type="submit" value="Create User"
className="button is-link is-fullwidth" />

        <Link to='/users'> <button type="submit"
className="button is-success is-fullwidth mt-2">Back List</button></Link>
        </div>

    </form>
  </div>
</div>
</div>
</div>
</div>
</div>
</section>

</>
);
}

export default Adduser;

```

#### 8.1.1.1.4 AdminDashboard.js

```

import axios from 'axios';
import React, { useEffect, useState } from 'react';
import { Link } from 'react-router-dom';
import '../Assets/Styles/admindash.css'

import cinemahalls from '../Assets/Images/cinemahalls.jpg';
import movie from '../Assets/Images/movie.jpg';
import showing from '../Assets/Images/showing.jpg';
import tickes from '../Assets/Images/tickes.jpg';
import users from '../Assets/Images/users.jpg';
import cate from '../Assets/Images/cate.jpg';
import NavBar_Admin from '../Home/NavBar_Admin';
function AdminDashboard(props) {
  const admin = JSON.parse(localStorage.getItem("admin"))
  const [count, setcount] = useState()

  useEffect(() => {

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

    axios.get('http://localhost:8280/shows/analysis').then((data) => {
      setcount(data.data)
    }).catch((err) => {
      console.log(err)
    })
  }, [])
  return (
    <>
    {localStorage.getItem('atoken') ? (
      <>
      <NavBar_Admin />

      <section className="hero is-fullheight-with-navbar admin-div-home ">
        <div className="hero-body">
          <div className="container is-fluid">
            <div className='title is-2 has-text-white has-text-centered'>Admin Dashboard</div>
            <div className='columns addash is-multiline is-justify-content-center'>
              <div class="column is-4">
                <div className='card m-5'>
                  <Link to='/allshow'>
                    <div class="">
                      <div class="p-6 is-flex is-justify-content-space-between">
                        <h6 class="m-b-20">Shows Ongoing</h6>
                        <p class="m-b-0"><span class="float-right">{count?.scount}</span></p>
                        <span><i class="fa fa-ticket float-left"></i></span>
                      </div>
                    </div>
                  </div>
                <div>
                  <img src={showing}></img>
                </div>
              </Link>
            </div>
          </div>
        </div>
      </div>
      <div class="column is-4" >

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

<div className='card m-5' style={{ background:
'linear-gradient(45deg, #4099ff, #73b4ff)' }}>
  <Link to='/allmovies'>
    <div class="" >
      <div class="p-6 is-flex is-justify-
content-space-between">
        <h6 class="m-b-20">Movies</h6>
        <p class="m-b-0"><span class="f-
right">{count?.mcount}</span></p>
        <span><i class="fa fa-film f-
left"></i></span>
      </div>
    </div>
  </div>
</div>

<div class="column is-4">
  <div className='card m-5' style={{ background:
'linear-gradient(45deg, #2ed8b6, #59e0c5)' }} >
    <Link to="/allhalls">
      <div class="">
        <div class="p-6 is-flex is-justify-
content-space-between">
          <h6 class="m-b-20">Movie Halls</h6>
          <p class="m-b-0"><span class="f-
right">{count?.hcount}</span></p>
          <span><i class="fa fa-video-camera f-
left"></i></span>
        </div>
      </div>
    </div>
  </div>
</div>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

<div class="column is-4">
  <div className='card m-5' style={{ background:
'linear-gradient(45deg, #2ed8b6, #59e0c5)' }} >
    <Link to="/allcategory">
      <div class="">
        <div class="p-6 is-flex is-justify-
content-space-between">

          <h6 class="m-b-20">Categories</h6>
          <p class="m-b-0"><span class="f-
right">{count?.ccount}</span></p>

          <span><i class="fa fa-video-camera f-
left"></i></span>

        </div>
      </div>
    </div>
    <div>
      <img src={cate}></img>
    </div>
  </Link>
</div>
</div>

{admin.type === "sa" ? <>
  <div className='column is-4'>
    <div className='card m-5' style={{ background:
'linear-gradient(45deg, #FFB64D, #ffcb80)' }}>
      <Link to="/users">
        <div class="">
          <div class="p-6 is-flex is-justify-
content-space-between">

            <h6 class="m-b-20">Users</h6>
            <p class="m-b-0"><span class="f-
right">{count?.ucount - 1}</span></p>

            <span><i class="fa fa-user f-
left"></i></span>

          </div>
        </div>
      </div>
    </div>
    <div>
      <img src={users}></img>
    </div>
  </Link>
</div>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    </div>

    </> : <></>

    {admin.type === "sa" ? <>
      <div className='column is-4'>
        <div className='card m-5' style={{ background:
'linear-gradient(45deg, #FF5370, #ff869a)' }}>
          <Link to="/bookings">
            <div class="">
              <div class="p-6 is-flex is-justify-
content-space-between">
                <div class="flex-grow-1">
                  <h6 class="m-b-20">Bookings</h6>
                  <p class="m-b-0"><span class="f-
right">{count?.bcount}</span></p>
                  <span><i class="fa fa-cart-arrow-
down f-left"></i></span>
                </div>
              </div>
            </div>
          <div>
            <img src={tickes}></img>
          </div>
        </Link>
      </div>
    </div> : <></>

    </div>
  </div >
</div>
</section>
</>
) : (
  window.location = "/"
)
</>
);
}

export default AdminDashboard;

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

## 8.1.1.1.5 AdminEdit.js

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';
import { Link, useParams } from 'react-router-dom';
import NavBar_Admin from '../Home/NavBar_Admin';

function AdminEdit(props) {
  const [name, setname] = useState("")
  const [email, setemail] = useState("")
  const [userName, setUsername] = useState("")
  const [password, setPassword] = useState("")
  const [confpassword, setconfpassword] = useState("")
  const params = useParams()
  const token = localStorage.getItem("atoken");
  const user = localStorage.getItem("admin");
  const userId = user._id;
  useEffect(() => {
    getData()
  }, [])
  //get data from profile page
  function getData() {
    axios({
      method: "get",
      baseURL: `http://localhost:8070/user/getuser/${props.match.params.id}`,
      headers: {
        Authorization: "Bearer " + token
      }
    })
    .then((res) => {
      setname(res.data.user.name)
      setemail(res.data.user.email)
      setUsername(res.data.user.userName)
      setPassword(res.data.user.password)
      setconfpassword(res.data.user.password)
    }).catch((err) => {
      alert(err.message);
    })
    console.log(params.id)
  }
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
}
```

```
//send to updated data to
function sendData(e) {
  e.preventDefault();
  if (confpassword === password) {

  }
  else {
    return (alert("Password Doesn't match"))
  }
  const newUser = {
    name,
    email,
    userName,
    password
  };

  axios({
    method: "put",
    baseUrl: `http://localhost:8070/user/update/${props.match.params.id}`,
    data: newUser,
    headers: {
      Authorization: "Bearer " + token
    }
  })

  .then(() => {
    alert('User Updated, Click ok');
    window.location.href = '/aprofile'

  })
  .catch((err) => {
    alert(err);
  });
}

return (
```



### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```
<>
{localStorage.getItem('atoken') ? (
  <>
    <NavBar_Admin />

    <section className="hero is-fullheight-with-navbar movie-div ">
      <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
        <div className="container is-widescreen">
          <div className='columns is-justify-content-center'>
            <div className='column is-6'>
              <div className="card" style={{ backgroundColor:
'#fff6', borderRadius: '10px' }}>
                <div className='title is-3 has-text-warning-dark
has-text-centered has-background-warning-light pb-3 pt-2' style={{ borderRadius: '10px' }}>
                  <b>Edit Profile</b>
                </div>
                <form onSubmit={sendData} >
                  <div className="field has-addons m-3 pt-3">
                    <div className="control is-expanded">
                      <div className="is-fullwidth">
                        <input type="text"
className="input" value={name} placeholder="name" onChange={(e) => { setname(e.target.value)
}} />
                      </div>
                    </div>
                    <div className="button control is-static">
                      <span className="icon is-left">
                        <i className="fa fa-user-circle-
o"></i>
                      </span>
                      <div type="submit" className=" is-
primary is-static mr-6 pr-1">Name</div>
                    </div>
                  </div>
                  <div className="field has-addons m-3 pt-3">
                    <div className="control is-expanded">
                      <div className="is-fullwidth">
                        <input type="email"
className="input" value={email} placeholder="Email" onChange={(e) => {
setemail(e.target.value) }} />
                      </div>
                    </div>
                  </div>
                </form>
              </div>
            </div>
          </div>
        </div>
      </div>
    </section>
  </div>
) : (
  <div className="hero is-fullheight-with-navbar movie-div ">
    <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
      <div className="container is-widescreen">
        <div className='columns is-justify-content-center'>
          <div className='column is-6'>
            <div className="card" style={{ backgroundColor:
'#fff6', borderRadius: '10px' }}>
              <div className='title is-3 has-text-warning-dark
has-text-centered has-background-warning-light pb-3 pt-2' style={{ borderRadius: '10px' }}>
                <b>Edit Profile</b>
              </div>
              <form onSubmit={sendData} >
                <div className="field has-addons m-3 pt-3">
                  <div className="control is-expanded">
                    <div className="is-fullwidth">
                      <input type="text"
className="input" value={name} placeholder="name" onChange={(e) => { setname(e.target.value)
}} />
                    </div>
                  </div>
                  <div className="button control is-static">
                    <span className="icon is-left">
                      <i className="fa fa-user-circle-
o"></i>
                    </span>
                    <div type="submit" className=" is-
primary is-static mr-6 pr-1">Name</div>
                  </div>
                </div>
                <div className="field has-addons m-3 pt-3">
                  <div className="control is-expanded">
                    <div className="is-fullwidth">
                      <input type="email"
className="input" value={email} placeholder="Email" onChange={(e) => {
setemail(e.target.value) }} />
                    </div>
                  </div>
                </div>
              </form>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
)
}
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

primary is-static mr-6 pr-2">Email</div>

</div>
</div>
<div className="field has-addons m-3 pt-3">
  <div className="control is-expanded">
    <div className="is-fullwidth">
      <input type="text"
className="input" value={userName} placeholder="User Name" onChange={(e) => {
setuserName(e.target.value) }} />

      </div>
    </div>
    <div className="button control is-static">
      <span className="icon is-left">
        <i className="fa fa-user"></i>
      </span>
      <div type="submit" className=" is-
primary is-static mr-3 pr-1">User Name</div>
    </div>
  </div>
  <div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
      <div className="is-fullwidth">
        <input type="password"
className="input" value={password} placeholder="Password" onChange={(e) => {
setpassword(e.target.value) }} />

        </div>
      </div>
      <div className="button control is-static">
        <span className="icon is-left">
          <i className="fa fa-lock"></i>
        </span>
        <div type="submit" className=" is-
primary is-static mr-5 pr-1">Password</div>
      </div>
    </div>
  </div>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        <div className="field has-addons m-3 pt-3">
            <div className="control is-expanded">
                <div className="is-fullwidth">
                    <input type="password"
className="input" value={confpassword} placeholder="Password" onChange={(e) => {
setconfpassword(e.target.value) }} />
                </div>
            </div>
            <div className="button control is-static">
                <span className="icon is-left">
                    <i className="fa fa-lock"></i>
                </span>
                <div type="submit" className=" is-
primary is-static mr-5 pr-1">Confirm Password</div>
            </div>
        </div>
        <div className=" has-background-info-light pt-
1 pb-2">
            <div className=" mt-5 mr-3 ml-3 pb-3 ">
                <Link to="/aprofile"><button
className="button is-link is-light is-fullwidth mb-2" type='submit' value="Create" >Back to
profile</button></Link>
                <button className="button is-warning
is-fullwidth " type='submit' value="Create" >Save Profile</button>
            </div>
        </div>
    </form>
</div>
</div>
</div>
</div>
</div>
</section>
</>
) : (
    window.location = "/"
)
</>
);

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
}
```

```
export default AdminEdit;
```

**8.1.1.1.6 AdminLogin.js**

```
import React, { useState } from "react";  
import axios from 'axios';  
import '../Assets/Styles/user.css'  
import { Link } from "react-router-dom";
```

```
const AdminLogin = () => {
```

```
  const [userName, setUser] = useState('')  
  const [email, setEmail] = useState('')  
  const [password, setPassword] = useState('')  
  const [test, setTest] = useState(false)  
  const [type, setType] = useState('')  
  const [userId, setUserId] = useState('')
```

```
  function sendData(e) {  
    e.preventDefault();  
    const newUser = {  
      //userName:userName,  
      email: email,  
      password,  
      type  
    };  
  
    axios  
      .post('http://localhost:8070/user/login', newUser)  
      .then((res) => {  
        if (res.status !== 200) {  
          console.log(res)  
        }  
      })  
  }  
}
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    }
    localStorage.setItem("admin", JSON.stringify(res.data.user))
    localStorage.setItem("atoken", res.data.token)

    console.log(res.data.token)
    setUserId(res.data.user._id)
    console.log(userId)
    setTest(!test)

    window.location.href = '/dashboard'

  })
  .catch((err) => {
    alert("Email or Password is worng");
    console.log(err)
  });
}

return (
  <>
  <div className='home-div'> </div>

  <section className="hero is-fullheight-with-navbar all-home-flexs">

    <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
      <div className="container is-widescreen">
        <div className='content'>
          <div className="columns is-justify-content-center">
            <div className="column is-3">
              <h2 className="subtitle is-2 is-semibold has-text-white
has-text-centered">Admin Login</h2>
              <form onSubmit={sendData}>
                <div className="control has-text-white has-text-
centered mb-4">
                  <label className="radio" >
                    <input type="radio" required value="ma"
name="rad" onChange={(e) => { settype(e.target.value) }} />
                    Movie Admin

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        </label>
        <label className="radio">
            <input type="radio" required value="sa"
name="rad" onChange={(e) => { settype(e.target.value) }} />
            System Admin
        </label>
    </div>
    <div className="mb-3">
        <input type="email" className="input is-primary"
placeholder="Email Address" onChange={(e) => { setemail(e.target.value) }} aria-
label="Username" aria-describedby="basic-addon1" required />
    </div>
    <div className="mb-3">
        <input type="password" className="input is-
primary" placeholder="Password" onChange={(e) => { setpassword(e.target.value) }} aria-
label="Password" aria-describedby="basic-addon1" required />
    </div>
    <div className="">
        <button type="submit" className="button is-link
is-fullwidth">Login</button>
        <Link to="/"> <button type="submit"
className="button is-success is-fullwidth mt-2">Back to Home Page</button></Link>
    </div>
</form>
</div>
</div>
</div>
</div>
</div>
</section>
</>
);
}
export default AdminLogin;

```

#### 8.1.1.1.7 Adminprofile.js

```

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

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
import NavBar_Admin from '../Home/NavBar_Admin';

function Adminprofile(props) {

  const [currentUser, setCurrentUser] = useState("")

  const token = localStorage.getItem("atoken");
  const users = JSON.parse(localStorage.getItem("admin"));
  console.log(users)
  const userId = users._id;

  useEffect(() => {
    const token = localStorage.getItem("atoken");
    try {
      axios({
        method: "get",
        baseURL: `http://localhost:8070/user/getuser/${userId}`,
        headers: {
          Authorization: "Bearer " + token
        },
      }).then(res => {
        setCurrentUser(res.data.user)
        console.log(res.data)
      })
    } catch (err) {
      throw err;
    }
  }, [])

  return (
    <>
    {localStorage.getItem('atoken') ? (
      <>
      <NavBar_Admin />

      <section className="hero is-fullheight-with-navbar movie-div ">
        <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
          <div className="container is-widescreen">
            <div className='columns is-justify-content-center'>
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

<div className='column is-6'>
  <div className="card" style={{ backgroundColor:
' #fff6', borderRadius: '10px' }}>
    <div className='title is-3 has-text-info-dark has-
text-centered has-background-info-light pb-3 pt-2' style={{ borderRadius: '10px' }}>
      <b>My Profile</b>
    </div>
    <form >
      <div className="field has-addons m-3 pt-3">
        <div className="control is-expanded">
          <div className="is-fullwidth">
            <input type="text"
className="input" value={currentUser.name} placeholder="name" disabled />
          </div>
        </div>
        <div className="button control is-static">
          <span className="icon is-left">
            <i className="fa fa-user-circle-
o"></i>

          </span>
          <div type="submit" className=" is-
primary is-static mr-6 pr-1">Name</div>
        </div>
      </div>
      <div className="field has-addons m-3 pt-3">
        <div className="control is-expanded">
          <div className="is-fullwidth">
            <input type="email"
className="input" value={currentUser.email} placeholder="Email" disabled />
          </div>
        </div>
        <div className="button control is-static">
          <span className="icon is-left">
            <i className="fa fa-envelope"></i>
          </span>
          <div type="submit" className=" is-
primary is-static mr-6 pr-2">Email</div>
        </div>
      </div>
      <div className="field has-addons m-3 pt-3">
        <div className="control is-expanded">

```



### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        <div className="is-fullwidth">
            <input type="text"
className="input" value={currentUser.userName} placeholder="User Name" disabled />
        </div>
    </div>
    <div className="button control is-static">
        <span className="icon is-left">
            <i className="fa fa-user"></i>
        </span>
        <div type="submit" className=" is-
primary is-static mr-3 pr-1">User Name</div>
    </div>
</div>
<div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
        <div className="is-fullwidth">
            <input type="password"
className="input" value={currentUser.password} placeholder="Password" disabled />
        </div>
    </div>
    <div className="button control is-static">
        <span className="icon is-left">
            <i className="fa fa-lock"></i>
        </span>
        <div type="submit" className=" is-
primary is-static mr-5 pr-1">Password</div>
    </div>
</div>
<div className=" has-background-info-light pt-
1 pb-2">
    <div className=" mt-5 mr-3 ml-3 pb-3 ">
        <Link to='/dashboard'><button
className="button is-link is-light is-fullwidth mb-2" type='submit' value="Create" >Back to
Dashboard</button></Link>
        <Link to={"/adedit/" +
currentUser._id}><button className="button is-info is-fullwidth " type='submit' value="Create"
>Edit Profile</button></Link>
    </div>
</div>
</form>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        </div>
      </div>
    </div>
  </div>
</section>
</>
) : (
  window.location = "/"
)}
</>
);
}

```

`export default Adminprofile;`

#### 8.1.1.1.8 AllBooking.js

```

import React, { useEffect, useState } from 'react';
import axios from 'axios';
import NavBar_Admin from '../Home/NavBar_Admin';

function AllBookings(props) {
  const [bookings, setbookings] = useState([])

  useEffect(() => {
    axios.get('http://localhost:8280/bookings/all').then((data) => {
      setbookings(data.data)

    }).catch((err) => {

    })
  }, [])
  return (
    <>
    {localStorage.getItem('atoken') ? (
      <>
      <NavBar_Admin />
      <section className="hero is-fullheight-with-navbar admin-div-home ">

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    <div className="hero-body">
      <div className="container is-fluid">
        <div className='title is-2 has-text-white has-text-
centered'>All Bookings</div><br />

        <table className='table is-bordered is-fullwidth '>
          <tr>
            <th>user name</th>
            <th>Movie name</th>
            <th>#Tickets</th>
            <th>Ticket Price</th>
            <th>Total</th>
          </tr>

          {bookings?.map((bo, i) => (
            <tr>
              <td>{bo.uid.name}</td>
              <td>{bo.mid.name}</td>
              <td>{bo.tickets}</td>
              <td>{bo.price}</td>
              <td>{(bo.tickets * bo.price)}</td>
            </tr>
          ))}
        </table>
      </div>
    </div>
  </section>

  </>
) : (
  window.location = "/"
)
</>
);
}

export default AllBookings;

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

#### 8.1.1.1.9 AllCategory.js

```
import axios from 'axios';
import React, { useEffect, useState } from 'react';
import NavBar_Admin from '../Home/NavBar_Admin';
import { Link } from 'react-router-dom'

function AllCategory(props) {
  const [cate, setcate] = useState([])

  useEffect(() => {
    axios.get('http://localhost:8280/categories/list').then((data) => {
      setcate(data.data)
    }).catch((err) => {

    })
  }, [cate])
  return (
    <>
      {localStorage.getItem('atoken') ? (
        <>

          <NavBar_Admin />
          <section className="hero is-fullheight-with-navbar admin-div-home ">
            <div className="hero-body">
              <div className='container is-fluid'>
                <div className='title is-2 has-text-white has-text-centered'>All Categories</div>
                <div className='is-flex is-right is-justify-content-end mb-3'>
                  <Link to="/addcategory"><a className='button is-info'>Add
Category<i class="fa fa-video-camera ml-2" aria-hidden="true"></i></a></Link>
                </div>
                <div className='columns'>
                  <div className='column'>
                    <table className="table is-bordered is-fullwidth">
                      <thead>
                        <tr>
                          <th scope="col">#</th>
                          <th scope="col">Name</th>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    </tr>
  </thead>
  <tbody>
    {cate.map((mov, i) => (
      <tr>
        <th scope="row">{(i + 1)}</th>
        <td>{mov.name}</td>

      </tr>
    ))}
  </tbody>
</table>
</div>
</div>
</div>
</div>
</section>
</>
) : (
  window.location = "/"
)}
</>
);
}

```

export default AllCategory;

8.1.1.1.10 AllHalls.js

```

import React, { useEffect } from 'react';
import { useSelector, useDispatch } from 'react-redux'
import { Link } from 'react-router-dom';
import { getHalls } from '../../Actions/Hall';
import axios from 'axios';
import NavBar_Admin from '../Home/NavBar_Admin';
function AllHalls(props) {
  const { halls } = useSelector(state => state.halls);
  const dispatch = useDispatch()

  useEffect(() => {

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

    dispatch(getHalls())
  }, [halls])

const deleteconfirm = (id) => {

  const confirmBox = window.confirm(
    "All The shows related to that Hall will be Deleted"
  )
  if (confirmBox === true) {
    deletefunction(id)
  }
}

const deletefunction = (id) => {
  axios.delete(`http://localhost:8070/hall/delete/${id}`).then((dat) => {

  }).
    catch((err) => {
      console.log(err)
    })
}

return (
  <>
    {localStorage.getItem('atoken') ? (
      <>

      <NavBar_Admin />
      <section className="hero is-fullheight-with-navbar admin-div-home ">
        <div className="hero-body">
          <div className='container is-fluid'>
            <div className='title is-2 has-text-white has-text-
centered'>All Halls</div>
            <div className='is-flex is-right is-justify-content-end mb-3'>
              <Link to="/addhall"><a className='button is-info'>Add
Halls<i class="fa fa-video-camera ml-2" aria-hidden="true"></i></a></Link>
            </div>
            <div className='columns'>
              <div className='column'>
                <table className="table is-bordered is-fullwidth">
                  <thead>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        <tr>
          <th scope="col">#</th>
          <th scope="col">Name</th>
          <th scope="col">Seat Rows</th>
          <th scope="col">Seats for one column</th>
          <th scope="col">Rate</th>
          <th style={{ width: '10%' }}>Action</th>
        </tr>
      </thead>
      <tbody>
        {halls.map((mov, i) => (
          <tr>
            <th scope="row">{(i + 1)}</th>
            <td>{mov.name}</td>
            <td>{mov.rows}</td>
            <td>{mov.cols}</td>
            <td>{mov.rate}</td>

            <td><Link to={'/updatehall/' +
mov._id}><a className='button is-warning mr-2'><i class="fa fa-pencil " aria-
hidden="true"></i></a></Link>

              <a className='button is-danger'
onClick={ (e) => { deleteconfirm(mov._id) } }><i class="fa fa-trash-o" aria-
hidden="true"></i></a></td>

            </tr>

          <td>{mov.name}</td>
          <td>{mov.rows}</td>
          <td>{mov.cols}</td>
          <td>{mov.rate}</td>

          <td><Link to={'/updatehall/' +
mov._id}><a className='button is-warning mr-2'><i class="fa fa-pencil " aria-
hidden="true"></i></a></Link>

            <a className='button is-danger'
onClick={ (e) => { deleteconfirm(mov._id) } }><i class="fa fa-trash-o" aria-
hidden="true"></i></a></td>

          </tr>

        </tbody>
      </table>
    </div>
  </div>
</div>
</div>
</section>
</>
) : (
  window.location = "/"
)}
</>
);
}

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
export default AllHalls;
```

## 8.1.1.1.11 AllMovies.js

```
import React, { useEffect } from 'react';
import { useSelector, useDispatch } from 'react-redux'
import { Link } from 'react-router-dom';
import { getmovies } from '../../Actions/MovieActions';
import '../../Assets/Styles/user.css'
import axios from 'axios';
import NavBar_Admin from '../Home/NavBar_Admin';

function AllMovies(props) {
  const { movies } = useSelector(state => state.movies);
  const dispatch = useDispatch()
  useEffect(() => {

    dispatch(getmovies())
  }, [movies])

  const deleteconfirm = (id) => {

    const confirmBox = window.confirm(
      "All The bookings and shows related to that movie will be Deleted"
    )
    if (confirmBox === true) {
      deletefunction(id)
    }
  }

  const deletefunction = (id) => {
    axios.delete(`http://localhost:8070/movies/delete/${id}`).then((dat) => {

    }).
    catch((err) => {
      console.log(err)
    })
  }
}
```



### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

return (
  <>
    {localStorage.getItem('atoken') ? (
      <>

        <NavBar_Admin />
        <section className="hero is-fullheight-with-navbar admin-div-home ">
          <div className="hero-body">
            <div className='container is-fluid'>
              <div className='title is-2 has-text-white has-text-
centered'>All Movies</div>
              <div className='is-flex is-right is-justify-content-end mb-3'>
                <Link to="/addmovie"><a className='button is-info'>Add
Movies<i class="fa fa-film ml-2" aria-hidden="true"></i></a></Link>
              </div>
              <div className='columns'>
                <div className='column'>
                  <table className="table is-bordered is-fullwidth">
                    <thead>
                      <tr>
                        <th scope="col">#</th>
                        <th scope="col" style={{ width: '100px' }}>
                          <Image/>
                        <th scope="col">Name</th>
                        <th scope="col">Description</th>
                        <th scope="col">Rate</th>
                        <th style={{ width: '10%' }}>Action</th>
                      </tr>
                    </thead>
                    <tbody>
                      {movies.map((mov, i) => (
                        <tr>
                          <th scope="row">{(i + 1)}</th>
                          <td> <img
src={"http://localhost:8070/" + mov.image} alt="movie poster" className="pro_user" style={{
height: '100px' }} /></td>
                          <td>{mov.name}</td>
                          <td>{mov.description}</td>
                          <td>{mov.rate}</td>

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

mov._id}><a className='button is-warning mr-2'><i class="fa fa-pencil" aria-
hidden="true"></i></a></Link>
                                <a className='button is-danger'
onClick={ (e) => { deleteconfirm(mov._id) } }><i class="fa fa-trash-o" aria-
hidden="true"></i></a></td>
                                </tr>
                                </tbody>
                                </table>
                                </div>
                                </div>
                                </div>
                                </div>
                                </section>
                                </>
                                ) : (
                                window.location = "/"
                                )}
                                </>
                                );
                                }

export default AllMovies;

```

#### 8.1.1.1.12 AllShows.js

```

import axios from 'axios';
import React, { useEffect, useState } from 'react';
import { Link } from 'react-router-dom';
import NavBar_Admin from '../Home/NavBar_Admin';

function AllShows(props) {

  const [shows, setshows] = useState([])

  const gettime = (id) => {
    if (id === 1) {
      return "10.00am - 1.00pm"
    }
  }

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

    }
    else if (id === 2) {
        return "1.00pm - 4.00pm"
    }
    else {
        return "4.00pm - 7.00pm"
    }
}

useEffect(() => {
    axios.get('http://localhost:8280/shows/list').then((data) => {
        setshows(data.data)
    }).catch((err) => {

    })
}, [shows])

const deleteconfirm = (id) => {

    const confirmBox = window.confirm(
        "All The bookings related to that show will be Deleted"
    )
    if (confirmBox === true) {
        deletefunction(id)
    }
}

const deletefunction = (id) => {
    axios.delete(`http://localhost:8070/show/delete/${id}`).then((dat) => {

    }).
        catch((err) => {
            console.log(err)
        })
}
return (
    <>
    {localStorage.getItem('atoken') ? (
        <>
        <NavBar_Admin />
        <section className="hero is-fullheight-with-navbar admin-div-home ">

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

<div className="hero-body">
  <div className="container is-fluid">
    <div className='title is-2 has-text-white has-text-
centered'>All Shows Outgoing</div>
    <div className='is-flex is-right is-justify-content-end mb-3'>
      <Link to="/addshow"><a className='button is-info'>Add
Shows<i class="fa fa-ticket ml-2" aria-hidden="true"></i></a></Link>
    </div>
    <div className='columns'>
      <div className='column'>

        <table className='table is-bordered is-fullwidth '>
          <tr>
            <th>Movie Name</th>
            <th>Hall Name</th>
            <th>Date</th>
            <th>Time</th>
            <th>Ticket Price</th>
            <th style={{ width: '10%' }}>Action</th>
          </tr>
          {shows?.map((m, i) => (
            <tr>
              <td>{m.mid.name}</td>
              <td>{m.hid.name}</td>
              <td>{m.date}</td>
              <td>{gettime(m.time)}</td>
              <td>{m.price}</td>
              <td><Link to={'/updateshow/' + m._id}><a
className='button is-warning mr-2'><i class="fa fa-pencil " aria-hidden="true"></i></a></Link>
                <a className='button is-danger'
onClick={e => { deleteconfirm(m._id) }}><i class="fa fa-trash-o" aria-hidden="true"></i></a>
              </td>
            </tr>
          ))}
        </table>
      </div>
    </div>
  </div>
</div>
</section>
</>

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    ) : (  
      window.location = "/"  
    )}  
  </>  
);  
}
```

```
export default AllShows;
```

8.1.1.1.13Allusers.js

```
import React, { useEffect, useState } from 'react';  
import axios from 'axios';  
import NavBar_Admin from '../Home/NavBar_Admin';  
import { Link } from 'react-router-dom';  
function Allusers(props) {  
  const [nusers, setnusers] = useState([])  
  const [ausers, setausers] = useState([])  
  
  useEffect(() => {  
    axios.get('http://localhost:8070/user/all').then((data) => {  
      setnusers(data.data.nusers)  
      setausers(data.data.ausers)  
    }).catch((err) => {  
  
    })  
  }, [nusers, ausers])  
  
  const deleteconfirm = (id) => {  
  
    const confirmBox = window.confirm(  
      "All The bookings related to that User will be Deleted"  
    )  
    if (confirmBox === true) {  
      deletefunction(id)  
    }  
  }  
  const deleteconfirmadmin = (id) => {  
  
    const confirmBox = window.confirm(  

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    "Are you sure"
  )
  if (confirmBox === true) {
    deletefunction(id)
  }
}

const deletefunction = (id) => {
  axios.delete(`http://localhost:8070/user/delete/${id}`).then((dat) => {

  })
  .catch((err) => {
    console.log(err)
  })
}
return (
  <>
  {localStorage.getItem('atoken') ? (

    <NavBar_Admin />
    <section className="hero is-fullheight-with-navbar admin-div-home ">
      <div className="hero-body">
        <div className='container is-fluid'>
          <div className='title is-2 has-text-white has-text-
centered'>All Users</div><br />

          <div className=''>
            <div className=''>
              <h2 className='title is-4 has-text-
centered'>Customers</h2>

              <div className='is-flex is-right is-justify-content-
end mb-3'>

                <Link to="/adduser"><a className='button is-
danger'>Add User<i class="fa fa-user ml-2" aria-hidden="true"></i></a></Link>
              </div>
              <table className="table is-bordered is-fullwidth">
                <tr>
                  <th>Name</th>
                  <th>Email</th>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        <th>UserName</th>
        <th style={{ width: '05%' }}>Action</th>

    </tr>

    {users?.map((us, i) => (
        <tr>
            <td>{us.name}</td>
            <td>{us.email}</td>
            <td>{us.userName}</td>
            <th><a className='button is-danger'
onClick={e => { deleteconfirm(us._id) }}><i class="fa fa-trash-o" aria-
hidden="true"></i></a></th>

            </tr>
        ))}
    </table>

    <h2 className='title is-4 has-text-centered'>Movie
Admins</h2>

    <div className='is-flex is-right is-justify-content-
end mb-3'>
        <Link to="/addadmin"><a className='button is-
danger'>Add Movie Admin<i class="fa fa-user ml-2" aria-hidden="true"></i></a></Link>
    </div>

    <table className="table is-bordered is-fullwidth">
        <tr>
            <th>Name</th>
            <th>Email</th>
            <th>UserName</th>
            <th style={{ width: '05%' }}>Action</th>
        </tr>

        {users?.map((us, i) => (
            <tr>
                <td>{us.name}</td>
                <td>{us.email}</td>
                <td>{us.userName}</td>

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

        <th><a className='button is-danger'
onClick={ (e) => { deleteconfirmadmin(us._id) } }><i class="fa fa-trash-o" aria-
hidden="true"></i></a></th>
        </tr>
      </table>
    </div>
  </div>
</section>
</>
) : (
  window.location = "/"
)
</>
);
}

export default Allusers;

```

#### 8.1.1.14 EditHall.js

```

import React, { useEffect, useState } from 'react';
import axios from 'axios';
import NavBar_Admin from '../Home/NavBar_Admin';
import { Link } from 'react-router-dom';

function EditHall(props) {

  const [name, setname] = useState()
  const [rows, setrows] = useState()
  const [rate, setrate] = useState()
  const [cols, setcols] = useState()
  useEffect(() => {
    axios.get(`http://localhost:8280/halls/getone/${props.match.params.id}`).then((data)
=> {
      setname(data.data.name)
      setrows(data.data.rows)

```



### Assignment II – REST API Project

#### SE3020 – Distributed Systems

#### Semester 1

```

    setrate(data.data.rate)
    setcols(data.data.cols)

  }).catch((err) => {
    console.log(err)
    alert("Hall Updated")
    window.location.href = "/allhalls"
  })
}, [])

const updateHall = (e) => {
  e.preventDefault()

  const mov = {
    rate,
    name
  }

  axios.put(`http://localhost:8070/hall/update/${props.match.params.id}`, mov).then((da)
=> {
    alert("Hall Upadated")
    window.location.href = "/allhalls"

  }).catch((err) => {
    console.log(err)
  })
}

return (
  <>
    {localStorage.getItem('atoken') ? (
      <>
        <NavBar_Admin />

        <section className="hero is-fullheight-with-navbar admin-div">
          <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
            <div className="container is-widescreen">
              <div className='columns is-justify-content-center'>
                <div className='column is-6'>
                  <div className="card" style={{ backgroundColor:
'rgba(255, 255, 255, 0.5)' }}>

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```


90


```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        <input type="number"
className="input" value={cols} readOnly disabled onChange={(e) => { setcols(e.target.value) }}
placeholder="name" aria-label="Username" />

        </div>
    </div>
    <div className="button control is-static">
        <span className="icon is-left">
            <i className="fa fa-film"></i>
        </span>
        <div type="submit" className=" is-
primary is-static">Seats for a row</div>

        </div>
    </div>
    <div className="field has-addons m-3 pt-3">
        <div className="control is-expanded">
            <div className="is-fullwidth">
                <input type="number"
className="input" value={rate} onChange={(e) => { setrate(e.target.value) }}
placeholder="Rate" required />

                </div>
            </div>
            <div className="button control is-static">
                <span className="icon is-left">
                    <i className="fa fa-film"></i>
                </span>
                <div type="submit" className=" is-
primary is-static mr-6 pr-5">Rate</div>

                </div>
            </div>

            <div className=" has-background-danger-light
pt-1 pb-2">

                <div className=" mt-5 mr-3 ml-3 pb-3 ">
                    <Link to="/allhalls"><button
className="button is-link is-fullwidth mb-2" type='submit' value="Create" >Back to
Shows</button></Link>

                    <button className="button is-danger
is-fullwidth " type="submit" value="create" >Create</button>
                </div>
            </div>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    </form>

    </div>
  </div>
</div>
</div>
</div>
</section>
</>
) : (
  window.location = "/"
)}
</>
);
}

export default EditHall;

```

#### 8.1.1.1.15 EditMovie.js

```

import React, { useEffect, useState } from 'react';
import axios from 'axios';
import NavBar_Admin from '../Home/NavBar_Admin';
import { Link } from 'react-router-dom';
function EditMovie(props) {
  const [name, setname] = useState()
  const [description, setdescription] = useState()
  const [rate, setrate] = useState()
  const [image, setimage] = useState()
  useEffect(() => {
    axios.get(`http://localhost:8280/movies/getmovie/${props.match.params.id}`).then((data) => {
      setname(data.data.name)
      setdescription(data.data.description)
      setrate(data.data.rate)
      setimage(data.data.image)
    }).catch((err) => {
      console.log(err)
    })
  })
}

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

}, [])

const updatemovie = (e) => {
  e.preventDefault()

  const mov = {
    rate,
    description
  }

  axios.put(`http://localhost:8070/movies/update/${props.match.params.id}`,
mov).then((da) => {
  alert("Movie Updated")
  window.location.href = "/allmovies"

}).catch((err) => {
  console.log(err)
})

}
return (
  <>
    {localStorage.getItem('atoken') ? (
      <>
        <NavBar_Admin />
        <section className="hero is-fullheight-with-navbar admin-div">
          <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
            <div className="container is-widescreen">
              <div className='columns is-justify-content-center'>
                <div className='column is-6'>
                  <div className="card" style={{ backgroundColor:
'#fff6' }}>
                    <div className='title is-2 has-text-danger-dark
has-text-centered has-background-danger-light pb-3 pt-2'>
                      <b>Edit Movies</b>
                    </div>
                    <form onSubmit={updatemovie}>
                      <div className=" is-flex is-justify-content-
center">

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

style={{ width: "215px" }}>
src={"http://localhost:8070/" + image} alt="movie poster" className="mt-1" /><br></br>
</div>
</div>

<div className="field has-addons m-3 pt-3">
  <div className="control is-expanded">
    <div className="is-fullwidth">
      <input type="text"
className="input" value={name} readOnly disabled onChange={(e) => { setname(e.target.value) }}
placeholder="name" aria-label="Username" />
    </div>
  </div>
  <div className="button control is-static">
    <span className="icon is-left">
      <i className="fa fa-film"></i>
    </span>
    <div type="submit" className=" is-
primary is-static mr-3 pr-1">Movie Name</div>
  </div>
</div>

<div className="field has-addons m-3 pt-3">
  <div className="control is-expanded">
    <div className="button control is-
static is-fullwidth">
      <span className="icon is-left">
        <i className="fa fa-pencil-
square-o"></i>
      </span>
      <div type="submit" className=" is-
primary is-static mr-3 pr-1">Movie Description</div>
    </div>
  <div className="is-fullwidth">
    <textarea class="textarea"
value={description} onChange={(e) => { setdescription(e.target.value) }}
placeholder="Textarea" required></textarea>
  </div>
</div>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    </div>

    <div className="field has-addons m-3 pt-3">
      <div className="control is-expanded">
        <div className="is-fullwidth">
          <input type="number"
className="input" value={rate} onChange={(e) => { setrate(e.target.value) }}
placeholder="Movie Rate" min="0" max="5" required />
        </div>
      </div>
      <div className="button control is-static">
        <span className="icon is-left">
          <i className="fa fa-star"></i>
        </span>
        <div type="submit" className=" is-
primary is-static mr-3 pr-1">Ratings</div>
      </div>
    </div>
    <div className=" has-background-danger-light
pt-1 pb-2">
      <div className=" mt-5 mr-3 ml-3 pb-3 ">
        <Link to='/allmovies'><button
className="button is-link is-fullwidth mb-2" type='submit' value="Create" >Back to
Movies</button></Link>
        <button className="button is-danger
is-fullwidth " type='submit' value="save" >Save</button>
      </div>
    </div>
  </form>
</div>
</div>
</div >

</div >
</div>
</section>
</>
) : (
  window.location = "/"
)}

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
</>
);
}

export default EditMovie;
```

**8.1.1.1.16 EditShow.js**

```
import axios from 'axios';
import React, { useEffect, useState } from 'react';
import { Link } from 'react-router-dom';
import NavBar_Admin from '../Home/NavBar_Admin';

function EditShow(props) {

  const [date, setdate] = useState('')
  const [time, settime] = useState('')
  const [price, setprice] = useState('')
  const [hall, sethall] = useState('')
  const [mov, setmov] = useState('')

  useEffect(() => {
    axios.get(`http://localhost:8280/shows/getone/${props.match.params.id}`).then((data)
=> {
      setmov(data.data.mid.name)
      sethall(data.data.hid.name)
      settime(data.data.time)
      setprice(data.data.price)
      setdate(data.data.date)

    }).catch((err) => {
      console.log(err)
    })
  }, [])

  const updateshow = (e) => {
    e.preventDefault()
    const show = {
      date,
      time,
```



### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

    price
  }
  axios.put(`http://localhost:8070/show/updateone/${props.match.params.id}`,
show).then((da) => {
  alert("Show Updated")
  window.location.href = "/allshow"

}).catch((err) => {
  console.log(err)
})

}

return (
  <>
    {localStorage.getItem('atoken') ? (
      <>
        <NavBar_Admin />

        <section className="hero is-fullheight-with-navbar admin-div">
          <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
            <div className="container is-widescreen">
              <div className='columns is-justify-content-center'>
                <div className='column is-6'>
                  <div className="card" style={{ backgroundColor:
'#fff6' }}>

                    <div className='title is-2 has-text-danger-dark
has-text-centered has-background-danger-light pb-3 pt-2'>
                      <b>Edit Shows</b>
                    </div>
                    <form onSubmit={updateshow}>

                      <div className="field has-addons m-3 pt-3">
                        <div className="control is-expanded">
                          <div className="select is-fullwidth">
                            <input type='text'
className="input" value={mov} readOnly disabled onChange={(e) => { setmov(e.target.value) }}
/>

                            </div>
                          </div>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

primary is-static mr-5 pr-2">Movie Name</div>

</div>
</div>

<div className="field has-addons m-3 pt-3">
  <div className="control is-expanded">
    <div className="select is-fullwidth">
      <input type='text'
className="input" value={hall} readOnly disabled onChange={(e) => { sethall(e.target.value) }}
/>

      </div>
    </div>
    <div className="button control is-static">
      <span className="icon is-left">
        <i className="fa fa-building"></i>
      </span>
      <div type="submit" className=" is-
primary is-static mr-6">Hall Name</div>

    </div>
  </div>

  <div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
      <div className="is-fullwidth">
        <input type="date"
className="input" value={date} onChange={(e) => { setdate(e.target.value) }}
placeholder="Date" required />

      </div>
    </div>
    <div className="button control is-static">
      <span className="icon is-left">
        <i className="fa fa-calendar"></i>
      </span>
      <div type="submit" className=" is-
primary is-static mr-6">Show date</div>

    </div>
  </div>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

settime(e.target.value) }} value={time}>
slot</option>
1.00 pm</option>
4.00 pm</option>
7.00 pm</option>

primary is-static mr-3">Select Time slot</div>

</div>
</div>

<div className="field has-addons m-3 pt-3">
  <div className="control is-expanded">
    <div className="select is-fullwidth">
      <select onChange={(e) => {
        <option>Select Time
        <option value="1">10.00am -
        <option value="2">1.00pm -
        <option value="3">4.00pm -
      </select>
    </div>
  </div>
</div>
<div className="button control is-static">
  <span className="icon is-left">
    <i className="fa fa-clock-o"></i>
  </span>
  <div type="submit" className=" is-
primary is-static mr-3">Select Time slot</div>
</div>
</div>

<div className="field has-addons m-3 pt-3">
  <div className="control is-expanded">
    <div className=" is-fullwidth">
      <input min='0' type='number'
className="input" value={price} onChange={(e) => { setprice(e.target.value) }} required />
    </div>
  </div>
<div className="button control is-static">
  <span className="icon is-left">
    <i className="fa fa-ticket"></i>
  </span>
  <div type="submit" className="is-
primary is-static ">Ticket Show Price</div>
</div>
</div>

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

pt-1 pb-2">
    <div className=" has-background-danger-light
        <div className=" mt-5 mr-3 ml-3 pb-3 ">
            <Link to='/allshow'><button
                className="button is-link is-fullwidth mb-2" type='submit' value="Create" >Back to
Shows</button></Link>
                <button className="button is-danger
is-fullwidth " type='submit' value="save" >Save</button>
            </div>
        </div>
    </div>
</div>
    </div >
</div>
    </div>
</section>
</>
) : (
    window.location = "/"
)}
</>
);
}

export default EditShow;

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.2.1.2 Hall**

## 8.2.1.2.1 AddHall.js

```
import axios from 'axios';
import React, { useState } from 'react';
import { Link } from 'react-router-dom';
import NavBar_Admin from '../Home/NavBar_Admin';
//import { message } from 'antd'

function AddHall(props) {
  const [name, setname] = useState("")
  const [cols, setcols] = useState("")
  const [rows, setrows] = useState("")
  const [rate, setrate] = useState("")

  const createHall = (e) => {
    e.preventDefault()

    const hall = {
      name,
      cols,
      rows,
      rate
    }

    axios.post('http://localhost:8070/hall', hall).then((data) => {
      console.log("created")
      alert("Hall Added")
      window.location = "/allhalls"
    }).catch((err) => {
      console.log(err)
    })
  }

  return (
    <>
    {localStorage.getItem('atoken') ? (
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```
<>
  <NavBar_Admin />

  <section className="hero is-fullheight-with-navbar admin-div">
    <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
      <div className="container is-widescreen">
        <div className='columns is-justify-content-center'>
          <div className='column is-6'>
            <div className="card" style={{ backgroundColor:
'#fff6' }}>
              <div className='title is-2 has-text-danger-dark
has-text-centered has-background-danger-light pb-3 pt-2'>
                <b>Adding Halls</b>
              </div>
              <form onSubmit={createHall}>
                <div className="field has-addons m-3 pt-3">
                  <div className="control is-expanded">
                    <div className="is-fullwidth">
                      <input type="text"
className="input" onChange={(e) => { setname(e.target.value) }} placeholder="Name" required />
                    </div>
                  </div>
                  <div className="button control is-static">
                    <span className="icon is-left">
                      <i className="fa fa-film"></i>
                    </span>
                    <div type="submit" className=" is-
primary is-static mr-6 pr-3">Name</div>
                  </div>
                </div>
                <div className="field has-addons m-3 pt-3">
                  <div className="control is-expanded">
                    <div className="is-fullwidth">
                      <input type="text"
className="input" onChange={(e) => { setrows(e.target.value) }} placeholder="Seat rows"
required />
                    </div>
                  </div>
                  <div className="button control is-static">
                    <span className="icon is-left">
                      <i className="fa fa-film"></i>
                    </span>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

primary is-static mr-5 pr-3">Seat rows</div>

</div>
</div>
<div className="field has-addons m-3 pt-3">
  <div className="control is-expanded">
    <div className="is-fullwidth">
      <input type="text"
className="input" onChange={(e) => { setcols(e.target.value) }} placeholder="Seats for a row"
required />

    </div>
  </div>
  <div className="button control is-static">
    <span className="icon is-left">
      <i className="fa fa-film"></i>
    </span>
    <div type="submit" className=" is-
primary is-static">Seats for a row</div>

  </div>
</div>
<div className="field has-addons m-3 pt-3">
  <div className="control is-expanded">
    <div className="is-fullwidth">
      <input type="text"
className="input" onChange={(e) => { setrate(e.target.value) }} placeholder="Rate" required />
    </div>
  </div>
  <div className="button control is-static">
    <span className="icon is-left">
      <i className="fa fa-film"></i>
    </span>
    <div type="submit" className=" is-
primary is-static mr-6 pr-5">Rate</div>

  </div>
</div>
<div className=" has-background-danger-light

  <div className=" mt-5 mr-3 ml-3 pb-3 ">

```

## Assignment II – REST API Project

## SE3020 – Distributed Systems

## Semester 1

```

        <Link to='/allhalls'><button
className="button is-link is-fullwidth mb-2" type='submit' value="Create" >Back to
Shows</button></Link>

                                <button className="button is-danger
is-fullwidth " type="submit" value="create" >Create</button>
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </section>
</>
) : (
    window.location = "/"
)}
</>
);
}

export default AddHall;
```



### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

#### 8.2.1.3 Home

##### 8.2.1.3.1 NavBar\_Admin.js

```
import React from 'react'
import { Link } from 'react-router-dom'

function NavBar_Admin() {
  const admin = JSON.parse(localStorage.getItem("admin"))
  // let [user, setUser] = useState();

  // useEffect(() => {
  //   const user = JSON.parse(localStorage.getItem("user"));
  //   setUser(user);
  //   console.log(user);

  // }, []);
  return (
    <nav className="navbar is-light" role="navigation" aria-label="main navigation">
      <div className="navbar-brand">
        <a className="navbar-item" target="_blank" href="https://www.sliit.lk/">
          
        </a>

        <a role="button" className="navbar-burger" aria-label="menu" aria-expanded="false"
data-target="navbarBasicExample">
          <span aria-hidden="true"></span>
          <span aria-hidden="true"></span>
          <span aria-hidden="true"></span>
        </a>
      </div>

      <div id="navbarBasicExample" className="navbar-menu">
        <div className="navbar-start">

          <Link to="/" className="navbar-item"> <i className="navbar-item fas fa-home"></i>
Home </Link>
          <Link to="/all" className="navbar-item"><i className="navbar-item fa fa-film"></i>
All Movies </Link>
```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```
{(localStorage.getItem('token')) && (
  <>
    <Link to="/mybookings" className="navbar-item"><i className="navbar-item fa fa-
video-camera"></i> My Movies </Link>
    <Link to="/cart" className="navbar-item"><i className="navbar-item fa fa-cart-
arrow-down"></i>My Cart </Link>
  </>
)}
{(localStorage.getItem('atoken')) && (
  <>
    <Link to="/dashboard" className="navbar-item"><i className="navbar-item fa fa-
tachometer"></i> Dashboard </Link>
    <Link to="/allshow" className="navbar-item"><i className="navbar-item fa fa-
ticket"></i> Shows </Link>
    <Link to="/allmovies" className="navbar-item"><i className="navbar-item fa fa-
film"></i> Movies </Link>
    <Link to="/allhalls" className="navbar-item"><i className="navbar-item fa fa-
video-camera"></i> Movies Halls </Link>
    <Link to="/allcategory" className="navbar-item"><i className="navbar-item fa fa-
folder-o"></i>Categories </Link>
    {(admin.type === "sa")&&(
      <>
        <Link to="/users" className="navbar-item"><i className="navbar-item fa fa-
user"></i> Users</Link>
        <Link to="/bookings" className="navbar-item"><i className="navbar-item fa fa-
cart-arrow-down"></i>Booking </Link>
      </>
    )}
  </>
)}
<Link to="/qrread" className="navbar-item"> <i className="navbar-item fa fa-qr-code"
aria-hidden="true"></i>QR Reader </Link>

</div>

<div className="navbar-end">

  {(localStorage.getItem('atoken')) && (
    <div className="navbar-item has-dropdown is-hoverable">
      <a className="navbar-link">
```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

Menu

```

</a>
<div className="navbar-dropdown">
  {(localStorage.getItem('atoken')) && (<
    <Link to="/aprofile" className="navbar-item"> <i className="fa fa-user mr-
2"></i>My Account </Link>
    </>
  )}
  {(localStorage.getItem('token')) && (<
    <Link to="/profile" className="navbar-item"> <i className="fa fa-user mr-
2"></i>My Account </Link>
    </>
  )}

  <hr className="navbar-divider" />
  <a className="navbar-item" onClick={(e) => {
    localStorage.clear();
    window.location.href = '/admin'
  }}>

    <b> Sign Out </b>
  </a>
</div>
</div>
))

<div className="navbar-item">
  <div className="buttons">

    { /* {!(localStorage.getItem('token')) && (
    <>
    <Link to="/login"><button className="button is-success mr-
3">Login</button></Link>
    <Link to="/register"><button className="button is-
danger">Register</button></Link>

    </>
  )} */}

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

        </div>
      </div>
    </div>
  </div>
</nav>
)
}

export default NavBar_Admin

```

#### 8.2.1.3.2 NavBar\_Home.js

```

import React from 'react'
import { Link } from 'react-router-dom'

function NavBar_Home() {

  // let [user, setUser] = useState();

  // useEffect(() => {
  //   const user = JSON.parse(localStorage.getItem("user"));
  //   setUser(user);
  //   console.log(user);

  // }, []);
  return (
    <nav className="navbar is-light" role="navigation" aria-label="main navigation">
      <div className="navbar-brand">
        <a className="navbar-item" target="_blank" href="https://www.sliit.lk/">
          
        </a>

        <a role="button" className="navbar-burger" aria-label="menu" aria-expanded="false"
data-target="navbarBasicExample">
          <span aria-hidden="true"></span>
          <span aria-hidden="true"></span>
          <span aria-hidden="true"></span>
        </a>
      </div>

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```
<div id="navbarBasicExample" className="navbar-menu">
  <div className="navbar-start">

    <Link to="/" className="navbar-item"> <i className="navbar-item fas fa-home"></i>
Home </Link>
    <Link to="/all" className="navbar-item"><i className="navbar-item fa fa-film"></i>
All Movies </Link>
    {(localStorage.getItem('token')) && (
      <>
        <Link to="/mybookings" className="navbar-item"><i className="navbar-item fa fa-
video-camera"></i> My Movies </Link>
        <Link to="/cart" className="navbar-item"><i className="navbar-item fa fa-cart-
arrow-down"></i>My Cart </Link>
      </>
    )}
    {(localStorage.getItem('atoken')) && (
      <>
        <Link to="/dashboard" className="navbar-item"><i className="navbar-item fa fa-
tachometer"></i> Dashboard </Link>
        <Link to="/allshow" className="navbar-item"><i className="navbar-item fa fa-
ticket"></i> Shows </Link>
        <Link to="/allmovies" className="navbar-item"><i className="navbar-item fa fa-
film"></i> Movies </Link>
        <Link to="/allhalls" className="navbar-item"><i className="navbar-item fa fa-
video-camera"></i> Movies Halls </Link>
        <Link to="/users" className="navbar-item"><i className="navbar-item fa fa-
user"></i> Users</Link>
        <Link to="/bookings" className="navbar-item"><i className="navbar-item fa fa-
cart-arrow-down"></i>Booking </Link>
      </>
    )}
    <Link to="/qrread" className="navbar-item"> <i className="navbar-item fa fa-qr-code"
aria-hidden="true"></i>QR Reader </Link>

  </div>

  <div className="navbar-end">

    {(localStorage.getItem('token') || localStorage.getItem('atoken')) && (
      <div className="navbar-item has-dropdown is-hoverable">
        <a className="navbar-link">
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

Menu

</a>

<div className="navbar-dropdown">

{(localStorage.getItem('atoken')) && (<

<Link to="/" className="navbar-item"> <i className="fa fa-user mr-2"></i>My

Account </Link>

</>

)}>

{(localStorage.getItem('token')) && (<

<Link to="/profile" className="navbar-item"> <i className="fa fa-user mr-

2"></i>My Account </Link>

</>

)}>

<hr className="navbar-divider" />

<a className="navbar-item" onClick={(e) => {

localStorage.clear();

window.location.href = '/'

}}>

<b> Sign Out </b>

</a>

</div>

</div>

)}>

<div className="navbar-item">

<div className="buttons">

{!(localStorage.getItem('token') || localStorage.getItem('atoken')) && (

<

<Link to="/login"><button className="button is-success mr-

3">Login</button></Link>

<Link to="/register"><button className="button is-

danger">Register</button></Link>

</>

)}>

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
        </div>
      </div>
    </div>
  </div>
</nav>
)
}

export default NavBar_Home
```

**8.2.1.4 QR****8.2.1.4.1 QR.js**

```
import React, { useState, useRef } from 'react';
import QRCode from 'qrcode';
import { Link } from 'react-router-dom'
function Qr() {
  //useStates
  const [text, setText] = useState('');
  const [imageUrl, setImageUrl] = useState('');
  // const [scanResultFile, setScanResultFile] = useState('');
  // const [scanResultWebCam, setScanResultWebCam] = useState('');

  // const qrRef = useRef(null);

  //function for Qr Code Generate
  const generateQrCode = async () => {
    try {
      //create object and passing into data
      const response = await QRCode.toDataURL(text);
      //set image url in to setImageUrl
      setImageUrl(response);
    } catch (error) {
      console.log(error);
    }
  }
  // //function fro qr reader
```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```
// const handleErrorFile = (error) => {
//   console.log(error);
// }
// const handleScanFile = (result) => {
//   if (result) {
//     setScanResultFile(result);
//   }
// }
// //upload image
// const onScanFile = () => {
//   qrRef.current.openImageDialog();
// }
// const handleErrorWebCam = (error) => {
//   console.log(error);
// }
// const handleScanWebCam = (result) => {
//   if (result) {
//     setScanResultWebCam(result);
//   }
// }
// }
return (
  <div className='container is-max-widescreen'>
    <h2><b>QR Code</b></h2>
    <Link to="/qrread"> <a>Scan Page</a></Link>
    <br /><br />
    <div className="columns">
      <div className="column is-half">
        <b>Text : </b><input type="text" onChange={(e) => setText(e.target.value)} required
      />
      <br /><br />
      <button className="button is-danger" onClick={(e) => { generateQrCode()
    }}>Generate</button>
      <br /><br />
      {imageUrl ? (<a href={imageUrl} download> <img src={imageUrl} alt="img" /><br
    />Click to Download</a>) : null}
      <br />
    </div>

    </* <div className="column">
      <button onClick={(e) => { onScanFile() }}>scan to Insert</button>
      <QrReader
```



### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

        ref={qrRef}
        delay={300}
        style={{ width: '100%' }}
        onError={handleErrorFile}
        onScan={handleScanFile}
        legacyMode
      />
      <h3>Scanned Code: {scanResultFile}</h3>
    </div>
    <div className="column">
      <h3>Qr Code Scan by Web Cam</h3>
      <QrReader
        delay={300}
        style={{ width: '100%' }}
        onError={handleErrorWebCam}
        onScan={handleScanWebCam}
      />
      <h3>Scanned By WebCam Code: {scanResultWebCam}</h3>
    </div> */}
  </div>
</div>
)
}

export default Qr

```

#### 8.2.1.4.2 QRRead.js

```

import React, { useState, useRef, useEffect } from 'react';
import QrReader from 'react-qr-reader';
import NavBar_Home from '../Home/NavBar_Home';
function QrRead() {
  //useStates
  const [scanResultFile, setScanResultFile] = useState();
  const [scanResultWebCam, setScanResultWebCam] = useState('');
  const [scanitems, setscanitems] = useState([])
  const [type, settype] = useState(true)
  const [movie, setmovie] = useState()

  const qrRef = useRef(null);

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
useEffect(() => {
  setmovie(null)
}, [type])

//function fro qr reader
const handleErrorFile = (error) => {
  console.log(error);
}
const handleScanFile = (result) => {
  if (result) {
    // setScanResultFile(result);
    // console.log(result);

    setmovie(JSON.parse(result))
  }
}
//upload image
const onScanFile = () => {
  qrRef.current.openImageDialog();
}
const handleErrorWebCam = (error) => {
  console.log(error);
}
const handleScanWebCam = (result) => {
  if (result) {

    setmovie(JSON.parse(result))
  }
}
let range
const getTime = (ind) => {

  if (ind === 1) {
    range = "10.00am - 1.00pm"
  } else if (ind === 2) {
    range = "1.00pm - 4.00pm"
  } else if (ind === 3) {
    range = "4.00pm - 7.00pm"
  }
}
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```
return (
  <div>
    <NavBar_Home />
    <section className="hero is-fullheight-with-navbar movie-div all-home-flexs">

      <div className="hero-body">
        <div className='container is-max-widescreen'>

          <div className="columns card">
            <div className="column is-half">

              {type ? <>
                <button className="button is-success" onClick={(e) => { settype(!type) }}><i
className="fa fa-camera" aria-hidden="true"></i>Use Camera</button>
                <QrReader
                  ref={qrRef}
                  delay={300}
                  style={{ width: '525px', backgroundColor: '#fff' }}
                  onError={handleErrorFile}
                  onScan={handleScanFile}
                  legacyMode
                />
                <div className="has-text-centered">
                  <button className="button is-danger " onClick={(e) => { onScanFile()
}}>scan to Insert</button>
                </div>
              </> : <>
                <div className="has-text-centered">
                  <button className="button is-warning " onClick={(e) => { settype(!type)
}}><i className="fa fa-picture-o" aria-hidden="true"></i>Use Image</button>
                </div>
                <QrReader
                  delay={300}
                  style={{ width: '580px', backgroundColor: '#fff' }}
                  onError={handleErrorWebCam}
                  onScan={handleScanWebCam}
                />

              </>
            </div>
            <div className="column">
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

<div>
  {movie ? <>
    <div className="card" style={{ width: '16rem' }}>
      <div className="card-image">
        <figure className="image ">
          <img src={"http://localhost:8070/" + movie.image} className="card-
img-top" alt="..." />
        </figure>
      </div>
      <div className="card-content">
        <div className="">
          <div className="card-body">
            <h2 className="card-title"><b>{movie.name}</b></h2>
            <p className="card-text">{movie.seats+', '}</p>
          </div>
          <ul className="">
            <li className="list-group-item">{"Date = " + movie.date}</li>
            { getTime(movie.time)}
            <li className="list-group-item">{"Time = " + range}</li>
            <li className="list-group-item">{"#Tickets = " +
movie.seats.length}</li>
            <li className="list-group-item">{"Ticket Price = " +
movie.price}</li>
            <li className="list-group-item">{"Total Price = " + (movie.price *
movie.seats.length)}</li>
          </ul>
        </div>
      </div>
    </div>
  </>
}

</div>

</div>
</div>
</div>

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    </section>
  </div>
)
}

export default QrRead
import React, { useState, useRef, useEffect } from 'react';
import QrReader from 'react-qr-reader';
import NavBar_Home from '../Home/NavBar_Home';
function QrRead() {
  //useStates
  const [scanResultFile, setScanResultFile] = useState();
  const [scanResultWebCam, setScanResultWebCam] = useState('');
  const [scanitems, setscanitems] = useState([])
  const [type, settype] = useState(true)
  const [movie, setmovie] = useState()

  const qrRef = useRef(null);

  useEffect(() => {
    setmovie(null)
  }, [type])

  //function fro qr reader
  const handleErrorFile = (error) => {
    console.log(error);
  }
  const handleScanFile = (result) => {
    if (result) {
      // setScanResultFile(result);
      // console.log(result);

      setmovie(JSON.parse(result))
    }
  }
  //upload image
  const onScanFile = () => {
    qrRef.current.openImageDialog();
  }
  const handleErrorWebCam = (error) => {
    console.log(error);
  }
}
```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

}
const handleScanWebCam = (result) => {
  if (result) {

    setmovie(JSON.parse(result))
  }
}
let range
const getTime = (ind) => {

  if (ind === 1) {
    range = "10.00am - 1.00pm"
  } else if (ind === 2) {
    range = "1.00pm - 4.00pm"
  } else if (ind === 3) {
    range = "4.00pm - 7.00pm"
  }
}
return (
  <div>
    <NavBar_Home />
    <section className="hero is-fullheight-with-navbar movie-div all-home-flexs">

      <div className="hero-body">
        <div className='container is-max-widescreen'>

          <div className="columns card">
            <div className="column is-half">

              {type ? <>
                <button className="button is-success" onClick={ (e) => { settype(!type) } }><i
className="fa fa-camera" aria-hidden="true"></i>Use Camera</button>
                <QrReader
                  ref={qrRef}
                  delay={300}
                  style={{ width: '525px', backgroundColor: '#fff' }}
                  onError={handleErrorFile}
                  onScan={handleScanFile}
                  legacyMode
                />
                <div className="has-text-centered">

```

### Assignment II – REST API Project

#### SE3020 – Distributed Systems

#### Semester 1

```

    <button className="button is-danger " onClick={(e) => { onScanFile()
}}>scan to Insert</button>
  </div>
</> : <>
  <div className="has-text-centered">
    <button className="button is-warning " onClick={(e) => { setType(!type)
}}><i className="fa fa-picture-o" aria-hidden="true"></i>Use Image</button>
  </div>
  <QrReader
    delay={300}
    style={{ width: '580px', backgroundColor: '#fff' }}
    onError={handleErrorWebCam}
    onScan={handleScanWebCam}
  />

</>
</div>
<div className="column">
  <div>
    {movie ? <>
      <div className="card" style={{ width: '16rem' }}>
        <div className="card-image">
          <figure className="image ">
            <img src={"http://localhost:8070/" + movie.image} className="card-
img-top" alt="..." />
          </figure>
        </div>
        <div className="card-content">
          <div className="">
            <div className="card-body">
              <h2 className="card-title"><b>{movie.name}</b></h2>
              <p className="card-text">{movie.seats+', '}</p>
            </div>
            <ul className="">
              <li className="list-group-item">{"Date = " + movie.date}</li>
              { getTime(movie.time)}
              <li className="list-group-item">{"Time = " + range}</li>
              <li className="list-group-item">{"#Tickets = " +
movie.seats.length}</li>
              <li className="list-group-item">{"Ticket Price = " +
movie.price}</li>

```

## Assignment II – REST API Project

## SE3020 – Distributed Systems

## Semester 1

```

        <li className="list-group-item">{"Total Price = " + (movie.price *
movie.seats.length)}</li>
      </ul>
    </div>
  </div>
</div>
</> : <>

</>}

</div>

</div>

</div>
</div>
</section>
</div>
)
}

export default QrRead

```

### 8.1.2.5 Show

#### 8.1.2.5.1 AddShow.js

```
import axios from 'axios';
import React, { useEffect, useState } from 'react';
//import { message } from 'antd'
import { getHalls } from '../Actions/Hall';
import { getmovies } from '../Actions/MovieActions';
import { useSelector, useDispatch } from 'react-redux'
import NavBar_Admin from '../Home/NavBar_Admin';
import { Link } from 'react-router-dom';
function Addshow(props) {
  const [mid, setmid] = useState("")
```



**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
const [hid, sethid] = useState("")
const [date, setdate] = useState("")
const [time, settime] = useState("")
const [price, setprice] = useState()
const { halls } = useSelector(state => state.halls);
const { movies } = useSelector(state => state.movies);

const dispatch = useDispatch()
useEffect(() => {
  dispatch(getHalls())
  dispatch(getmovies())
}, [])

const createshow = (e) => {
  e.preventDefault()

  const show = {
    hid,
    mid,
    date,
    time,
    price
  }
  console.log(show)

  axios.post('http://localhost:8070/show', show).then((data) => {
    console.log("created")
    alert("show added");
    window.location = "/allshow"
  }).catch((err) => {
    console.log(err)
  })
}

return (
  <>
    {localStorage.getItem('atoken') ? (
      <>
        <NavBar_Admin />
      </>
    ) : (
      <NavBar />
    )}
  </>
)
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

<section className="hero is-fullheight-with-navbar admin-div">
  <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
    <div className="container is-widescreen">
      <div className='columns is-justify-content-center'>
        <div className='column is-6'>
          <div className="card" style={{ backgroundColor:
'#fff6' }}>

                                <div className='title is-2 has-text-danger-dark
has-text-centered has-background-danger-light pb-3 pt-2'>
                                  <b>Adding Shows</b>
                                </div>
                                <form onSubmit={createshow}>

                                <div className="field has-addons m-3 pt-3">
                                  <div className="control is-expanded">
                                    <div className="select is-fullwidth">
                                      <select onChange={(e) => {
setmid(e.target.value) }} >

                                                                                   <option>Select Movie</option>
                                                                                   {movies.map(m => (
                                                                                     <option
value={m._id}>{m.name}</option>

                                                                                   )))}
                                                                                   </select>
                                    </div>
                                  </div>
                                  <div className="button control is-static">
                                    <span className="icon is-left">
                                      <i className="fa fa-film"></i>
                                    </span>
                                    <div type="submit" className=" is-
primary is-static mr-5 pr-2">Select Movie</div>
                                  </div>
                                </div>
                                <div className="field has-addons m-3 pt-3">
                                  <div className="control is-expanded">
                                    <div className="select is-fullwidth">
                                      <select onChange={(e) => {
sethid(e.target.value) }}>

                                                                                   <option>Select Hall</option>
                                                                                   {halls.map(m => (

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

value={m._id}>{m.name}</option>
    </select>
  </div>
</div>
<div className="button control is-static">
  <span className="icon is-left">
    <i className="fa fa-building"></i>
  </span>
  <div type="submit" className=" is-
primary is-static mr-6">Select Hall</div>
</div>
</div>
<div className="field has-addons m-3 pt-3">
  <div className="control is-expanded">
    <div className="is-fullwidth">
      <input type="date"
className="input" onChange={(e) => { setdate(e.target.value) }} required/>
    </div>
</div>
<div className="button control is-static">
  <span className="icon is-left">
    <i className="fa fa-calendar"></i>
  </span>
  <div type="submit" className=" is-
primary is-static mr-3 pr-1">Select the date</div>
</div>
</div>
<div className="field has-addons m-3 pt-3">
  <div className="control is-expanded">
    <div className=" is-fullwidth">
      <input min='0' type='number'
className="input" onChange={(e) => { setprice(e.target.value) }} placeholder="Ticket Price"
required/>
    </div>
</div>
<div className="button control is-static">
  <span className="icon is-left">
    <i className="fa fa-ticket"></i>
  </span>

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

primary is-static ">Enter Ticket price</div>

settime(e.target.value) }>

slot</option>

1.00 pm</option>

4.00 pm</option>

7.00 pm</option>

<div type="submit" className="is-
    </div>
</div>
<div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
        <div className="select is-fullwidth">
            <select onChange={e => {
                <option>Select Time
                <option value="1">10.00am -
                <option value="2">1.00pm -
                <option value="3">4.00pm -
            </select>
        </div>
    </div>
</div>
<div className="button control is-static">
    <span className="icon is-left">
        <i className="fa fa-clock-o"></i>
    </span>
    <div type="submit" className=" is-
primary is-static mr-3">Select Time slot</div>
    </div>
</div>
<div className=" has-background-danger-light
pt-1 pb-2">
    <div className=" mt-5 mr-3 ml-3 pb-3 ">
        <Link to='/allshow'><button
className="button is-link is-fullwidth mb-2" type='submit' value="Create" >Back to
Shows</button></Link>
        <button className="button is-danger
is-fullwidth " type="submit" value="create" >Create</button>
    </div>
</div>
</form>
</div>
</div>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        </div >
      </div>
    </div>
  </section>
</>
) : (
  window.location = "/"
)}
</>
);
}

```

export default Addshow;

#### 8.1.2.5.2 Booking.js

```

import axios from 'axios';
import React, { useEffect, useState } from 'react';
import '../Assets/Styles/book.css'
import NavBar_Home from '../Home/NavBar_Home';

// import { Tabs, Radio, Space } from 'antd';
// const { TabPane } = Tabs;

function Booking(props) {

  const [show, setshow] = useState()
  const [row, setrow] = useState()
  const [col, setcol] = useState()
  const [data, setdata] = useState(0)
  const [seats, setseats] = useState([])
  const [total, settotal] = useState(0)
  const [mov, setmov] = useState()

  const [savedata, setsavedata] = useState([])
  var cartitems = JSON.parse(localStorage.getItem("cart")) || []

  let divide = []
  let selectedseats = []
  useEffect(() => {

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
axios.get(`http://localhost:8070/show/getone/${props.match.params.id}`).then((data) =>
{
    setshow(data.data)
    setrow(data.data.hid.rows)
    setcol(data.data.hid.cols)
    setseats(data.data.hid.seats)
    setmov(data.data.mid)
    selectedseats = data.data.seatbook
    setsavedata(selectedseats)

}).catch((err) => {
    console.log(err)
})

// Get all "navbar-burger" elements
const $navbarBurgers = document.querySelectorAll('.navbar-burger')

// Add a click event on each of them
$navbarBurgers.forEach(el => {
    el.addEventListener('click', () => {

        // Get the target from the "data-target" attribute
        const target = el.dataset.target;
        const $target = document.getElementById(target);

        // Toggle the "is-active" class on both the "navbar-burger" and the "navbar-
menu"
        el.classList.toggle('is-active');
        $target.classList.toggle('is-active');

    });
});
}, [])

const dividefunc = () => {
    var dis = []
    for (var i = 0; i < seats.length; i++) {

        dis.push(seats[i])
    }
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
        if ((i + 1) % col === 0) {
            divide.push(dis)
            dis = []
        }
    }
}

dividefunc()

const clickfunc = (id) => {

    const ch = document.getElementById("ch")
    const tota = document.getElementById("total")
    var ds = document.getElementById(id)
    if (ds.classList.contains("ocupied")) {
        return
    }
    if (ds.classList.contains("selected")) {
        ds.classList.remove("selected")
        ds.innerHTML = ""
    } else {
        ds.classList.add("selected")
        ds.innerHTML = id
    }
    if (selectedseats.includes(id)) {
        selectedseats = selectedseats.filter(item => item !== id)
        ch.innerHTML = selectedseats
        tota.innerHTML = (selectedseats.length * show.price)
    } else {

        selectedseats.push(id)
        ch.innerHTML = selectedseats
        tota.innerHTML = (selectedseats.length * show.price)

    }

    console.log(selectedseats)
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
const booktickets = () => {
  // const upshow = {
  //   seatbook: [...savedata, ...selectedseats]
  // }
  // axios.put(`http://localhost:8070/show/update/${props.match.params.id}`,
upshow).then((data) => {
  //   alert("booked")
  // }).catch((err) => {
  //   console.log(err)
  // })
  if (selectedseats.length === 0) {
    alert("please select seats")
    return
  }

  const cartitem = {
    sid: props.match.params.id,
    tickets: selectedseats.length,
    show: show,
    total: (selectedseats.length * show.price),
    mov: mov,
    num: cartitems.length,
    seatbook: [...savedata, ...selectedseats],
    sseats: selectedseats
  }
  cartitems.push(cartitem)
  localStorage.setItem("cart", JSON.stringify(cartitems))
  window.location.href = '/cart';
}
let selected = false;
const findselected = (id) => {
  if (savedata.includes(id)) {
    document.getElementById(id).classList.add("ocupied")
  }
  else {

  }
}
return (
```



### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```
<>
<NavBar_Home />
<section className="hero is-fullheight-with-navbar movie-div">

  <div className="hero-body">
    <div className="container is-fluid">

      <div className='container is-fluid'>
        <div className='rows is-justify-content-center'>

          <div className=''>
            <div className='rows is-justify-content-center mt-5'>
              <div className='col-xl-8 col-lg-8 col-md-8 col-sm-10
col-12'>

                <ul className='showcase'>
                  <li>
                    <div className='seat'></div>
                    <small>NA</small>
                  </li>
                  <li>
                    <div className='seat selected'></div>
                    <small>selected</small>
                  </li>
                  <li>
                    <div className='seat occupied'></div>
                    <small>ocupied</small>
                  </li>
                </ul>
              </div>
            </div>

            <div className='columns'>
              <div className='column is-10'>
                <div className='con'>
                  <div className='screen'></div>
                  <div className='columns is-justify-content-
center'>

                    <div className='col-xl-10 col-lg-10 col-
md-10 col-sm-12 col-12'>

                      <table>
                        {divide.map((s, i) => (
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        <tr>
            {s.map(d => (
                <>
                <td> <div
onClick={ (e) => { clickfunc(d) }} id={d} className="seat" ></div></td>
                    {findselected(d)}
                </>
            ) )}
        </tr>
    </table>
</div>
</div>
</div>
</div >
<div className='column mt-6 title is-4 has-text-
white'>
    <div className='mt-2 '>
        <h4 className='subtitle is-4 has-text-
white'>Ticket Price :<b><p className='subtitle is-6 has-text-danger-dark'>
{show?.price}</p></b></h4>
        <h4 className='subtitle is-4 has-text-
white'>Booked seats :<b><p className='subtitle is-6 has-text-danger-dark'
id='ch'></p></b></h4>
        <h4 className='subtitle is-4 has-text-white'>Total
price : <b><p className='subtitle is-6 has-text-danger-dark' id='total'></p></b></h4>
        <button onClick={ (e) => { booktickets() }}
className='button is-warning'>Add to cart</button>
    </div>
</div>
</div>
</div>
</div>
</div >
</div>
</div>
</section>
</>
);
}

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
export default Booking;
```

## 8.1.2.5.3 Cart.js

```
import axios from 'axios';
import StripeCheckout from 'react-stripe-checkout';
import React, { useEffect, useState } from 'react';
import '../Assets/Styles/cart.css'
import NavBar_Home from '../Home/NavBar_Home';
import { Link } from 'react-router-dom';
function Cart(props) {
  const [total, settotal] = useState(0)
  var t = 0;
  var sum = 0;
  var it = JSON.parse(localStorage.getItem("cart"))

  var user = JSON.parse(localStorage.getItem("user"))
  const [cartitems, setcartitems] = useState(it)

  useEffect(() => {

    // Get all "navbar-burger" elements
    const $navbarBurgers = document.querySelectorAll('.navbar-burger')

    // Add a click event on each of them
    $navbarBurgers.forEach(el => {
      el.addEventListener('click', () => {

        // Get the target from the "data-target" attribute
        const target = el.dataset.target;
        const $target = document.getElementById(target);

        // Toggle the "is-active" class on both the "navbar-burger" and the "navbar-
menu"
        el.classList.toggle('is-active');
        $target.classList.toggle('is-active');
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    });  
  });  
  
  }, [cartitems])  
  useEffect(() => {  
  
    localStorage.setItem("cart", JSON.stringify(cartitems))  
  
  
  }, [cartitems])  
  
  const calctotal = (tot) => {  
    sum = sum + tot;  
  }  
  
  
  const removecart = (id) => {  
    setcartitems(cartitems.filter(item => item.num !== id))  
    calctotal()  
  }  
  
  const bookmovies = () => {  
    const data = {  
      cart: cartitems,  
      uid: user._id,  
  
    }  
    axios.post('http://localhost:8070/booking', data).then((resu) => {  
  
    }).catch((err) => {  
      console.log(err)  
    })  
  
    axios.put(`http://localhost:8070/show/updatecart`, data).then((da) => {  
      alert("booked")  
    }).catch((err) => {  
      console.log(err)  
    })  
  }  
}
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

localStorage.setItem("cart", null)

window.location.href = '/all';
}
return (

  <>
    {localStorage.getItem('token') ? (
      <>
        <NavBar_Home />
        <section className="hero is-fullheight-with-navbar movie-div all-home-
flexs">
          <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
            <div className="container is-widescreen">
              <div className="columns is-justify-content-center">
                <div className="column is-9">
                  <div className="card">
                    <div className="card p-1">
                      <table className="table is-striped is-narrow
is-hoverable is-fullwidth ">
                        <thead>
                          <tr className="is-uppercase">
                            <th scope="col"><div
className="">Movie</div></th>
                            <th scope="col" ><div
className="">Quantity</div></th>
                            <th scope="col" ><div
className="">Price</div></th>
                            <th scope="col" ><div
className="" style={{ float: 'right' }}>Action</div></th>
                        </tr>
                      </thead>
                      <tbody>
                        {cartitems?.map((m, i) => (
                          <tr key={i}>
                            <td>

                                <figure className="is-flex
is-align-items-center">

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

src={"http://localhost:8070/" + m.mov.image} style={{ height: '120px' }} />
paddingLeft: '0.25rem' }} className="pl-2">

<h3><b>{m.mov.name}</b></h3>

text-grey">{m.show.date}<br /> 11.00pm</p>

has-text-danger"><b>{m.tickets}</b> </div></td>

wrap">

fontWeight: '600', color: '#212529' }} >

<div>${m.total}</div>

text-grey">

${m.show.price}</div>

float: 'right' }}>

light " > <i className="fa fa-heart " /></a>

danger" onClick={(e) => { removecart(m.num) }} > <i className="fa fa-trash" /></a>

float: 'right' }}>

</tbody>
  
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    </table>
  </div>
</div>
</div>

<div className="column">
  <div className="card">
    {/* <div className="card-content">

      <form>
        <div className="form-group"> <label>Have
coupon?</label>
          <div className="input-group"> <input
type="text" className="input" name placeholder="Coupon code" />
            <span className="input-group-
append">
              <button className="button is-
danger is-outlined is-fullwidth ">Apply</button>
            </span>
          </div>
        </div>
      </form>

    </div> */}
  </div>

  <div className="card mt-3">
    <div className="card-content">
      <dl className="is-flex">
        {/* <dt>Total price:</dt>
        <dd className="text-right ml-
3">$69.97</dd> */}

        </dl>
        <dl className="is-flex">
          <dt>Discount:</dt>
          <dd className="text-right text-danger ml-
3">- $0.00</dd>

          </dl>
          <dl className="is-flex">
            <dt>Total:</dt>

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

3"><strong>${sum ? sum : 0}</strong></dd>

</dl>
<hr />
<StripeCheckout
  shippingAddress
  currency='LKR'
  amount={sum * 100}
  token={bookmovies}

stripeKey="pk_test_51KON7QSGc2uzmcTNMsY4QEFqEOPT7kUQaFthMpzSvbbeDYNxBvvPTkiZDnQhMMuuLadaLvFR36
OxyQBbVKmXkYnT000ZDxnzBd"

  >
  <a href="#" className="button is-primary
is-outlined is-fullwidth" data-abc="true"> Make Purchase </a>
</StripeCheckout>
  <a href="#" className="button is-info is-
outlined mt-2 is-fullwidth" data-abc="true"><Link to="/all">Continue Shopping</Link></a>
</div>
</div>
</div>
</div>
</div>
</section>
</>
) : (
  window.location = "/"
)
}
</>

);
}

export default Cart;

```



**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.1.2.6 User**

## 8.1.2.6.1 login.js

```
import React, { useEffect, useState } from "react";
import axios from 'axios';
import '../Assets/Styles/user.css'
import { Link } from 'react-router-dom';

const Login = () => {

  const [userName, setUser] = useState("")
  const [email, setEmail] = useState("kavinduchamith01@gmail.com")
  const [password, setPassword] = useState("Donkavi2@")
  const [test, setTest] = useState(false)
  const [userId, setUserId] = useState('')

  function sendData(e) {
    e.preventDefault();
    const newUser = {
      //userName:userName,
      email: email,
      password
    };

    axios
      .post('http://localhost:8070/user/login', newUser)
      .then((res) => {
        if (res.status !== 200) {
          console.log(res)
        }

        localStorage.setItem("user", JSON.stringify(res.data.user))
        localStorage.setItem("token", res.data.token)

        console.log(res.data.token)
        setUserId(res.data.user._id)
```

## Assignment II – REST API Project

## SE3020 – Distributed Systems

## Semester 1

```
setTest(!test)

window.location.href = '/all';

})
.catch((err) => {
  alert("Email or Password is worng");
  console.log(err)
});
}

return (
  <>
    <div className='home-div'> </div>

    <section className="hero is-fullheight-with-navbar all-home-flexs">

      <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
        <div className="container is-widescreen">
          <div className='content '>
            <div className="columns is-justify-content-center">
              <div className="column is-3">
                <h2 className="subtitle is-2 is-semibold has-text-white has-text-centered">Login</h2>

                <form onSubmit={sendData}>

                  <div className="mb-3">
                    <input type="email" className="input is-primary" value="123@gmail.com"
placeholder="Email Address" onChange={(e) => { setemail(e.target.value) }} aria-
label="Username" aria-describedby="basic-addon1" required />
                  </div>
                  <div className="mb-3">
                    <input type="password" className="input is-primary" value="Donkavi2@"
placeholder="Password" onChange={(e) => { setpassword(e.target.value) }} aria-label="Password"
aria-describedby="basic-addon1" required />
                  </div>
                  <div>
```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

        <button type="submit" className="button is-link is-
fullwidth">Login</button>
        <Link to="/register"><button className="button is-danger is-fullwidth
mt-2">Register</button></Link>
        <Link to="/"> <button type="submit" className="button is-success is-
fullwidth mt-2">Back to Home Page</button></Link>
    </div>
</form>
</div>
</div>
</div>
</div>
</div>
</section>
</>
);
}
export default Login;

```

#### 8.1.2.6.2 Mybookings.js

```

import axios from 'axios';
import React, { useEffect, useState } from 'react';
import moment from 'moment';
import QRCode from 'qrcode';
import '../Assets/Styles/mybook.css'
import NavBar_Home from '../Home/NavBar_Home';
function Mybookings(props) {
    const [imageUrl, setImageUrl] = useState('');
    const [items, setitems] = useState([])
    const [check, setcheck] = useState(false)
    var user = JSON.parse(localStorage.getItem("user"))
    useEffect(() => {
        axios.get(`http://localhost:8280/bookings/getone/${user._id}`).then((data) => {
            setitems(data.data)

        }).catch((err) => {
            console.log(err)
        })
    }, [items])
}

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
const generateQrCode = async (book) => {
  try {
    //create object and passing into data

    const ob = {
      name: book.mid.name,
      image: book.mid.image,
      seats: book.seats,
      date: book.sid.date,
      time: book.sid.time,
      price: book.sid.price,
    }
    const response = await QRCode.toDataURL(JSON.stringify(ob));
    //set image url in to setImageUr1
    setImageUr1(response);
  } catch (error) {
    console.log(error);
  }
}

const checkavailable = (dat) => {
  var d1 = moment(dat, "YYYY MM DD")
  var current = moment().startOf('day');
  var dif = moment.duration(current.diff(d1)).asDays()

  if (dif > 1) {
    return false
  } else {
    return true
  }
}

const delbooking = (id) => {
  axios.delete(`http://localhost:8070/booking/${id}`).then((dat) => {

  }).catch((er) => {
    console.log(er)
  })
}
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

return (
  <>
    {localStorage.getItem('token') ? (
      <>
        <NavBar_Home />

        <section className="hero is-fullheight-with-navbar movie-div all-home-
flexs">

          <div className="hero-body">
            <div className="container is-fluid">
              <div className=''>
                <h2 className=''>My Bookings</h2>
              </div>
              <div className='columns is-justify-content-center'>
                <div className='column is-10'>
                  <div className="">
                    <table className="table is-striped is-narrow is-
hoverable is-fullwidth ">

                      <thead className="text-muted">
                        <tr className="small text-uppercase">
                          <th scope="col">Movie</th>
                          <th scope="col">Book On</th>
                          <th scope="col">Show Date</th>
                          <th scope="col">Time</th>
                          <th scope="col">#Tickets</th>
                          <th scope="col">Price</th>
                          <th scope="col"></th>
                          <th scope="col"></th>
                        </tr>
                      </thead>
                      <tbody>
                        {items?.map((m, i) => (
                          <tr className='table_row'>

                            <td>
                              <figure className="itemside
align-items-center">

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        <div
className="aside"><img src={"http://localhost:8070/" + m.mid?.image} style={{ height: '120px'
}} /></div>

        <figcaption

        <h3

className='txt'>{m.mid.name}</h3>

        <p className="text-
muted small txt2"><br /></p>

        </figcaption>
        </figure>
        </td>

        <td>{moment(m.createdAt).format("YYYY-MM-DD")}</td>

        <td>{m.sid.date}</td>
        <td>{m.sid.time}</td>
        <td>{m.tickets} </td>
        <td>
            <div className="price-wrap">
                <var className="price">${m.sid.price * m.tickets}</var> <small className="text-muted">
                    ${m.sid.price} </small> </div>
            </td>

            <td className="text-right d-md-
block"> {checkavailable(m.createdAt) ? <a href className="btn btn-light" data-abc="true"
onClick={ (e) => { delbooking(m._id) } } ><i className="fa fa-trash" /></a> : <></> } </td>
            <td><a onClick={ (e) => {
generateQrCode(m) } } className="btn btn-light"><i className="fa fa-qrcode" aria-
hidden="true"></i></a></td>

        </tr>
    </tbody>
</table>

</div>

</div>
<div className='column is-2' >
    <div className=' justify-content-center'>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

src={imageUrl} alt="img" /><br /><h4 className='text-center'>Click to Download</h4></a> :
null}

</div>
</div>

</div>
</div>
</div>
</div>
</section>

</>
) : (
  window.location = "/"
)}
</>

);
}

```

export default Mybookings;

#### 8.1.2.6.3 profile.js

```

import React, { useEffect, useState } from 'react';
import { Link } from 'react-router-dom';
import axios from 'axios';
import NavBar_Home from '../Home/NavBar_Home'
function Profile({ setTest }) {
  const user = (localStorage.getItem("user"))

  const [currentUser, setcurrentUser] = useState("")

  const token = localStorage.getItem("token");
  const users = JSON.parse(localStorage.getItem("user"));
  console.log(users)
  const userId = users._id;

```

## Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```
useEffect(() => {
  const token = localStorage.getItem("token");
  try {
    axios({
      method: "get",
      baseURL: `http://localhost:8070/user/getuser/${userId}`,
      headers: {
        Authorization: "Bearer " + token
      },
    }).then(res => {
      setCurrentUser(res.data.user)
      console.log(res.data)
    })
  } catch (err) {
    throw err;
  }
}, [])

return (
  <>
    {localStorage.getItem('token') ? (
      <>
        <NavBar_Home />

        <section className="hero is-fullheight-with-navbar movie-div ">
          <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
            <div className="container is-widescreen">
              <div className='columns is-justify-content-center'>
                <div className='column is-6'>
                  <div className="card" style={{ backgroundColor: '#fff6', borderRadius:
'10px' }}>
                    <div className='title is-3 has-text-info-dark has-text-centered has-
background-info-light pb-3 pt-2' style={{ borderRadius: '10px' }}>
                      <b>My Profile</b>
                    </div>
                    <form >
                      <div className="field has-addons m-3 pt-3">
```



### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

1" >Name</div>
    <div className="control is-expanded">
      <div className="is-fullwidth">
        <input type="text" className="input" value={currentUser.name}
placeholder="name" disabled />
      </div>
    </div>
    <div className="button control is-static">
      <span className="icon is-left">
        <i className="fa fa-user-circle-o"></i>
      </span>
      <div type="submit" className=" is-primary is-static mr-6 pr-
2">Email</div>
    </div>
  </div>
  <div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
      <div className="is-fullwidth">
        <input type="email" className="input" value={currentUser.email}
placeholder="Email" disabled />
      </div>
    </div>
    <div className="button control is-static">
      <span className="icon is-left">
        <i className="fa fa-envelope"></i>
      </span>
      <div type="submit" className=" is-primary is-static mr-6 pr-
2">Email</div>
    </div>
  </div>
  <div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
      <div className="is-fullwidth">
        <input type="text" className="input"
value={currentUser.userName} placeholder="User Name" disabled />
      </div>
    </div>
    <div className="button control is-static">
      <span className="icon is-left">
        <i className="fa fa-user"></i>
      </span>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

1">User Name</div>
    </div>
  </div>
  <div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
      <div className="is-fullwidth">
        <input type="password" className="input"
value={currentUser.password} placeholder="Password" disabled />
      </div>
    </div>
    <div className="button control is-static">
      <span className="icon is-left">
        <i className="fa fa-lock"></i>
      </span>
      <div type="submit" className=" is-primary is-static mr-5 pr-
1">Password</div>
    </div>
  </div>
  <div className=" has-background-info-light pt-1 pb-2">
    <div className=" mt-5 mr-3 ml-3 pb-3 ">
      <Link to='/all'><button className="button is-link is-light is-
fullwidth mb-2" type='submit' value="Create" >Back to Movies</button></Link>
      <Link to={"/updateuser/" + currentUser._id}><button
className="button is-info is-fullwidth " type='submit' value="Create" >Edit
Profile</button></Link>
    </div>
  </div>
</form>

</div>
</div>
</div>
</div>
</div>
</section>
</>
) : (
  window.location = "/"
)}

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

&lt;/&gt;

)

}

`export default Profile;`

## 8.1.2.6.4 register.js

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

// import PasswordChecklist from "react-password-checklist"

function Register() {

  const [name, setname] = useState("")
  const [email, setemail] = useState("")
  const [userName, setUsername] = useState("")
  const [password, setPassword] = useState("")
  const [confpassword, setconfpassword] = useState("")
  const [users, setUsers] = useState([])
  useEffect(() => {

  }, [])

  function sendData(e) {
    e.preventDefault();

    if (confpassword === password) {

    }
    else {
      return (alert("Password Doesn't match"))
    }
    const newUser = {
```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

name,
email,
userName,
password,
type: ""
};

axios
.post('http://localhost:8070/user/signup', newUser)
.then(() => {
  alert('User Create Successfully ');
  setemail('');
  setname('');
  setUser('');
  setPassword('');
  setconfirmPassword('');
  window.location.href = '/login'
})
.catch((err) => {
  alert("Email Already Exists");
  console.log(err)
});
}

return (
  <>
  <div className='home-div'> </div>

  <section className="hero is-fullheight-with-navbar all-home-flexs">

    <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
      <div className="container is-widescreen">
        <div className='content '>
          <div className='columns is-justify-content-center'>
            <div className='column is-3'>
              <h2 className="subtitle is-2 is-semibold has-text-white has-text-centered">Registration</h2>
              <form onSubmit={sendData}>
                <div className="mb-3">
                  <input type="text" className="input is-primary" placeholder="Full Name"
                    onChange={(e) => { setName(e.target.value) }} aria-describedby="basic-addon1" required />

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    </div>
    <div className="mb-3">
      <input type="email" className="input is-primary" placeholder="Email"
onChange={(e) => { setEmail(e.target.value) }} aria-describedby="basic-addon1" required />
    </div>
    <div className="mb-3">
      <input type="text" className="input is-primary" placeholder="Username"
onChange={(e) => { setUsername(e.target.value) }} aria-describedby="basic-addon1" required />
    </div>
    <div className="mb-3">
      <input type="password" pattern="(?!.*\d)(?!.*[a-z])(?!.*[A-Z]).{8,}"
className="input is-primary" placeholder="Password" onChange={(e) => {
setpassword(e.target.value) }} aria-describedby="basic-addon1" required />
    </div>
    <div className="mb-3">
      <input type="password" pattern="(?!.*\d)(?!.*[a-z])(?!.*[A-Z]).{8,}"
className="input is-primary" placeholder="Confirm Password" onChange={(e) => {
setconfpassword(e.target.value) }} aria-describedby="basic-addon1" required />
    </div>
    {/ * <PasswordChecklist
rules=[["minLength", "specialChar", "number", "capital", "match"]]
minLength={5}
value={password}
valueAgain={confpassword}
onChange={({isValid) => { }}
/ * */}

    <div>
      <input type="submit" value="Register" className="button is-link is-
fullwidth" />
      <Link to="/login"><button className="button is-danger is-fullwidth mt-
2">Login</button></Link>
      <Link to="/"> <button type="submit" className="button is-success is-
fullwidth mt-2">Back to Home Page</button></Link>
    </div>

  </form>
</div>
</div>

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
        </div>
      </div>
    </div>
  </section>

</>
)

}
export default Register;
```

## 8.1.2.6.5 update.js

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';
import { Link, useParams } from 'react-router-dom';
import NavBar_Home from '../Home/NavBar_Home';
```

```
function Update(props) {
```

```
  const [name, setname] = useState("")
  const [email, setemail] = useState("")
  const [userName, setuserName] = useState("")
  const [password, setpassword] = useState("")
  const [confpassword, setconfpassword] = useState("")
  const params = useParams()
  const token = localStorage.getItem("token");
  const user = localStorage.getItem("user");
  const userId = user._id;
  useEffect(() => {
    getData()
  }, [])
  //get data from profile page
  function getData() {
    axios({
      method: "get",
      baseURL: `http://localhost:8070/user/getuser/${props.match.params.id}`,
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
headers: {
  Authorization: "Bearer " + token
}
})
.then((res) => {
  setname(res.data.user.name)
  setemail(res.data.user.email)
  setUsername(res.data.user.userName)
  setPassword(res.data.user.password)
  setconfpassword(res.data.user.password)
}).catch((err) => {
  alert(err.message);
})
console.log(params.id)
}

//send to updated data to
function sendData(e) {
  e.preventDefault();
  if (confpassword === password) {

  }
  else {
    return (alert("Password Doesn't match"))
  }
  const newUser = {
    name,
    email,
    userName,
    password
  };

  axios({
    method: "put",
    baseURL: `http://localhost:8070/user/update/${props.match.params.id}`,
    data: newUser,
```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

headers: {
  Authorization: "Bearer " + token
}
})

.then(() => {
  alert('User Updated, Click ok');
  window.location.href = '/profile'

})
.catch((err) => {
  alert(err);
});
}

return (
  <>
  {localStorage.getItem('token') ? (
    <>
    <NavBar_Home />

    <section className="hero is-fullheight-with-navbar movie-div ">
      <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
        <div className="container is-widescreen">
          <div className='columns is-justify-content-center'>
            <div className='column is-6'>
              <div className="card" style={{ backgroundColor: '#fff6', borderRadius:
'10px' }}>
                <div className='title is-3 has-text-warning-dark has-text-centered has-
background-warning-light pb-3 pt-2' style={{ borderRadius: '10px' }}>
                  <b>Edit Profile</b>
                </div>
                <form onSubmit={sendData} >
                  <div className="field has-addons m-3 pt-3">
                    <div className="control is-expanded">
                      <div className="is-fullwidth">
                        <input type="text" className="input" value={name}
placeholder="name" onChange={(e) => { setname(e.target.value) }} />
                      </div>
                    </div>
                    <div className="button control is-static">

```



### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

1">Name</div>
    <span className="icon is-left">
        <i className="fa fa-user-circle-o"></i>
    </span>
    <div type="submit" className=" is-primary is-static mr-6 pr-
1">Name</div>
    </div>
</div>
<div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
        <div className="is-fullwidth">
            <input type="email" className="input" value={email}
placeholder="Email" onChange={(e) => { setemail(e.target.value) }} />
        </div>
    </div>
    <div className="button control is-static">
        <span className="icon is-left">
            <i className="fa fa-envelope"></i>
        </span>
        <div type="submit" className=" is-primary is-static mr-6 pr-
2">Email</div>
    </div>
</div>
<div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
        <div className="is-fullwidth">
            <input type="text" className="input" value={userName}
placeholder="User Name" onChange={(e) => { setUserName(e.target.value) }} />
        </div>
    </div>
    <div className="button control is-static">
        <span className="icon is-left">
            <i className="fa fa-user"></i>
        </span>
        <div type="submit" className=" is-primary is-static mr-3 pr-
1">User Name</div>
    </div>
</div>
<div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
        <div className="is-fullwidth">

```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

        <input type="password" className="input" value={password}
placeholder="Password" onChange={(e) => { setpassword(e.target.value) }} />
    </div>
</div>
<div className="button control is-static">
    <span className="icon is-left">
        <i className="fa fa-lock"></i>
    </span>
    <div type="submit" className=" is-primary is-static mr-5 pr-
1">Password</div>
    </div>
</div>
<div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
        <div className="is-fullwidth">
            <input type="password" className="input" value={confpassword}
placeholder="Password" onChange={(e) => { setconfpassword(e.target.value) }} />
        </div>
    </div>
    <div className="button control is-static">
        <span className="icon is-left">
            <i className="fa fa-lock"></i>
        </span>
        <div type="submit" className=" is-primary is-static mr-5 pr-
1">Confirm Password</div>
    </div>
</div>
<div className=" has-background-info-light pt-1 pb-2">
    <div className=" mt-5 mr-3 ml-3 pb-3 ">
        <Link to='/profile'><button className="button is-link is-light is-
fullwidth mb-2" type='submit' value="Create" >Back to profile</button></Link>
        <button className="button is-warning is-fullwidth " type='submit'
value="Create" >Save Profile</button>
    </div>
</div>

</form>

</div>
</div>
</div>

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
        </div>
      </div>
    </section>
  </>
) : (
  window.location = "/"
)}
</>

)

}
export default Update;
```

**8.2.1.7 Movie****8.2.1.7.1 Addmovie.js**

```
import React, { useState, useEffect } from 'react';
import { useSelector, useDispatch } from 'react-redux'
import axios from 'axios'
//import { Card } from 'antd';
import { getcategories } from '../Actions/CategoryActions';
import Checkbox from './Checkbox';
import { getHalls } from '../Actions/Hall';
import NavBar_Admin from '../Home/NavBar_Admin';
import { Link } from 'react-router-dom';
//const { Meta } = Card;
function Addmovie(props) {

  const [name, setName] = useState("");
  const [description, setdescription] = useState("");
  const [to, setto] = useState("");
  const [from, setfrom] = useState("");
  const [rates, setrate] = useState("");
  const [formdata, setformdata] = useState(null);
  const [ida, setida] = useState([])
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
let array = []
let data = new FormData();

const { categories } = useSelector(state => state.categories);
const { halls } = useSelector(state => state.halls);

const dispatch = useDispatch()

useEffect(() => {
  dispatch(getcategories())
  dispatch(getHalls())

}, [])

const upload = ({ target: { files } }) => {

  data.append("studentImage", files[0]);

  data.append("imgname", files[0].name);

}
data.append("studentImage", formdata);
data.append("name", name);

data.append("description", description);
data.append("to", to);
data.append("from", from);
//data.append("rates", rates);

function sendData(e) {

  data.append("arr", JSON.stringify(ida))
  data.append("rates", rates);

  console.log(ida)
  e.preventDefault();
```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

console.log(data.name)

axios.post("http://localhost:8070/movies/add", data).then(() => {
  alert("movie added");
  window.location = "/allmovies"

}).catch((err) => {
  alert(err);
})
}
const handleFilters = (id) => {

  array = id;
  setida(array)
}

return (
  <>
    {localStorage.getItem('atoken') ? (
      <>
        <NavBar_Admin />
        <section className="hero is-fullheight-with-navbar admin-div">
          <div className="hero-body" style={{ padding: '3rem 0.5rem' }}>
            <div className="container is-widescreen">
              <div className='columns is-justify-content-center'>
                <div className='column is-6'>
                  <div className="card" style={{ backgroundColor:
'rgba(255, 255, 255, 0.5)' }}>
                    <div className='title is-2 has-text-danger-dark
has-text-centered has-background-danger-light pb-3 pt-2'>
                      <b>Adding Movies</b>
                    </div>
                    <form onSubmit={sendData}>
                      <div className="field has-addons m-3 pt-3">
                        <div className="control is-expanded">
                          <div className="is-fullwidth">
                            <input type="text"
className="input" onChange={(e) => { setName(e.target.value) }} placeholder="name" required />

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        </div>
    </div>
    <div className="button control is-static">
        <span className="icon is-left">
            <i className="fa fa-film"></i>
        </span>
        <div type="submit" className=" is-
primary is-static mr-3 pr-1">Movie Name</div>
    </div>
</div>
<div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
        <div className="button control is-
static is-fullwidth">
            <span className="icon is-left">
                <i className="fa fa-pencil-
square-o"></i>
            </span>
            <div type="submit" className=" is-
primary is-static mr-3 pr-1">Movie Description</div>
        </div>
        <div className="is-fullwidth">
            <textarea class="textarea"
onChange={(e) => { setDescription(e.target.value) }} placeholder="Textarea"
required></textarea>
        </div>
    </div>
</div>
<div className="field has-addons m-3 pt-3">
    <div className="control is-expanded">
        <div className="is-fullwidth">
            <input type="date"
className="input" onChange={(e) => { setfrom(e.target.value) }} placeholder="name" required />
        </div>
    </div>
    <div className="button control is-static">
        <span className="icon is-left">
            <i className="fa fa-calendar"></i>
        </span>
        <div type="submit" className=" is-
primary is-static mr-3 ">Movie Start Date</div>
    </div>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        </div>
    </div>
    <div className="field has-addons m-3 pt-3">
        <div className="control is-expanded">
            <div className="is-fullwidth">
                <input type="date"
className="input" onChange={(e) => { setto(e.target.value) }} placeholder="name" required />
            </div>
        </div>
        <div className="button control is-static">
            <span className="icon is-left">
                <i className="fa fa-calendar"></i>
            </span>
            <div type="submit" className=" is-
primary is-static mr-3 pr-1">Movie End Date</div>
        </div>
    </div>

    <div className="field has-addons m-3 pt-3">
        <div className="control is-expanded">
            <div className="button control is-
static is-fullwidth">

                <span className="icon is-left">
                    <i className="fa fa-
check"></i>

                </span>
                <div type="submit" className=" is-
primary is-static mr-3 pr-1">Movie Catagory</div>
            </div>
            <div className="" style={{ display:
'contents !important' }}>
                <Checkbox className=""
                    handleFilters={filters =>
                        handleFilters(filters, 'area')} />
            </div>
        </div>
    </div>

    <div className="field has-addons m-3 pt-3">
        <div className="control is-expanded">

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

className="input" onChange={(e) => { setrate(e.target.value) }} placeholder="Movie Rate"
min="0" max="5" aria-label="Username" />

</div>
</div>
<div className="button control is-static">
  <span className="icon is-left">
    <i className="fa fa fa-star"></i>
  </span>
  <div type="submit" className=" is-
primary is-static mr-5 pr-4">Movie Rate</div>
</div>
</div>

<div className="field has-addons m-3 pt-3">
  <div className="control is-expanded">
    <div className="is-fullwidth">
      <input className="input"
type="file" name="resume" onClick={(e) => { console.log(ida) }} onChange={upload}
id="inputGroupFile04" required />
    </div>
  </div>
  <div className="button control is-static">
    <span className="icon is-left">
      <i className="fas fa-upload"></i>
    </span>
    <div type="submit" className=" is-
primary is-static mr-3 pr-1">Choose a file...</div>
  </div>
</div>

<div className=" has-background-danger-light
pt-1 pb-2">

  <div className=" mt-5 mr-3 ml-3 pb-3 ">
    <Link to='/allmovies'><button
className="button is-link is-fullwidth mb-2" type='submit' value="Create" >Back to
Movies</button></Link>

    <button className="button is-danger
is-fullwidth " type='submit' value="Create" >Create a Movie</button>
  </div>

```



## Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    </div>

    </form>
  </div>
</div>

</div >

</div >
</div>
</section>
</>
) : (
  window.location = "/"
)}
</>
);
}

export default Addmovie;

```

### 8.2.1.7.2 Allmovies.js

```

import React, { useState, useEffect } from 'react';
import { useSelector, useDispatch } from 'react-redux'
import { getfiltermovies } from '../Actions/MovieActions';
import "../Assets/Styles/home.css"
import { groups } from './GroupNum'
import { getcategories } from '../Actions/CategoryActions';
import Checkbox from './Checkbox';
import Radiobox from './Radiobox';
//import { Radio } from 'antd';
import MovieCard from './MovieCard';
import { Link } from 'react-router-dom'
import NavBar_Home from '../Components/Home/NavBar_Home';

// import Paper from '@mui/material/Paper';
// import InputBase from '@mui/material/InputBase';
// import Divider from '@mui/material/Divider';
// import IconButton from '@mui/material/IconButton';

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
// import MenuIcon from '@mui/icons-material/Menu';
// import SearchIcon from '@mui/icons-material/Search';
function Allmovies(props) {

  const { categories } = useSelector(state => state.categories);
  const dispatch = useDispatch()

  useEffect(() => {
    dispatch(getcategories())
    LoadfilterResults(myFilters.filters)

    // Get all "navbar-burger" elements
    const $navbarBurgers = document.querySelectorAll('.navbar-burger')

    // Add a click event on each of them
    $navbarBurgers.forEach(el => {
      el.addEventListener('click', () => {

        // Get the target from the "data-target" attribute
        const target = el.dataset.target;
        const $target = document.getElementById(target);

        // Toggle the "is-active" class on both the "navbar-burger" and the "navbar-
menu"
        el.classList.toggle('is-active');
        $target.classList.toggle('is-active');

      });
    });

  }, [])

  const [limit, setlimit] = useState(12)
  const [skip, setskip] = useState(0)
  const [type, settype] = useState(false);
  const [size, setsize] = useState(0)
  const [filterResults, setfilterResults] = useState(0)
  const [myFilters, setmyFilters] = useState({
    filters: { area: [], groups: [] }
  })
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
const handleFilters = (filters, filterby) => {
  const newfilter = { ...myFilters }
  newfilter.filters[filterby] = filters
  if (filterby == "groups") {
    let pricevalues = handleprice(filters)
    newfilter.filters[filterby] = pricevalues
  }
  LoadfilterResults(myFilters.filters)
  setmyFilters(newfilter)
}

function handleprice(values) {
  const data = groups
  let array = []
  for (let key in data) {
    if (data[key]._id === parseInt(values)) {
      array = data[key].array
    }
  }
  return array;
}

function LoadfilterResults(newfilter) {

  getfiltermovies(skip, limit, newfilter).then(data => {
    if (data.error) {
      console.log(data.error)
    }
    else {
      setfilterResults(data.data)
      setsize(data.size)
      setskip(0)
    }
  })
}

const loadmorebutton = () => {
  return (
    size > 0 && size >= limit && (
      <button className='button is-warning' onClick={loadmore}>
        loadMore
      </button>
    )
  )
}
```

### Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

```

    )
  )
}

// const searchbar = () => {
//   return (
//     <Paper

//       component="form"
//       sx={{ p: '2px 4px', display: 'flex', alignItems: 'center' }}
//       >

//         <InputBase
//           sx={{ ml: 1, flex: 1 }}
//           placeholder="dd"
//           inputProps={{ 'aria-label': 'search google maps' }}
//         />
//         <IconButton type="submit" sx={{ p: '10px' }} aria-label="search">
//           <SearchIcon />
//         </IconButton>
//         <Divider sx={{ height: 28, m: 0.5 }} orientation="vertical" />

//       </Paper>
//     )
//   }

function loadmore() {
  let toskip = skip + limit
  getfiltermovies(toskip, limit, myFilters.filters).then(data => {
    if (data.error) {
      console.log(data.error)
    }
    else {
      setfilterResults([...filterResults, ...data.data])
      setsize(data.size)
      setskip(toskip)
    }
  })
}

return (

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```
<>
  <NavBar_Home />

  <section className="hero is-fullheight-with-navbar movie-div all-home-flexs">

    <div className="hero-body">
      <div className="container is-fluid">

        <div className='title is-2 has-text-white'>All Movies</div><br />
        <div className="columns">
          <div className="column is-2">

            <div className='container'>
              <div className=''>
                <div className='title is-4 has-text-
white'>Category</div>

                <Checkbox areas={categories}
                  handleFilters={filters => handleFilters(filters,
'area')}} />

              </div>
            </div>

            <div className='container'>
              <div Group className='radio'>
                <br /><h3 className='title is-4 has-text-
white'>Ratings</h3>

                <Radiobox groups={groups}
                  handleFilters={filters => handleFilters(filters,
'groups')}} />

              </div>
            </div>

          </div>
          <div className="column">
            <div className='columns is-multiline '>
              {filterResults ? (filterResults.map((mov, i) => (
                <div key={i} className='column is-2 '>
                  <Link to={` /view/${mov._id}`} ><MovieCard
movie={mov} /></Link>

                </div>
              )) : null}
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>
</div>
```

## Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

        ))) : ''}
      </div>
      {loadmorebutton()}
    </div>
  </div>

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

export default Allmovies;

```

8.2.1.7.3 Checkbox.js

```

import React, { useState } from "react";

function Checkbox({ areas, handleFilters }) {
  const [checked, setchecked] = useState([])

  const handleToggle = c => {
    const currentcategoryId = checked.indexOf(c)
    const newcheckedcategoryid = [...checked]

    if (currentcategoryId === -1) {
      newcheckedcategoryid.push(c)
    }
    else {
      newcheckedcategoryid.splice(currentcategoryId, 1)
    }

    setchecked(newcheckedcategoryid)
    handleFilters(newcheckedcategoryid)
  }
  return areas.map((c, i) => (

```

### Assignment II – REST API Project

#### SE3020 – Distributed Systems

#### Semester 1

```

<div className='title is-6 has-text-white form-inline d-flex align-items-center py-1'
key={i}>

    <input type='checkbox' className='checkbox' onChange={(e) => { handleToggle(c._id)
}} value={checked.indexOf(c._id === -1)} />
    <label className='tick ml-5'>{c.name}</label>
    <span className="check" />

</div>
))
}
export default Checkbox;
import React, { useState } from "react";

function Checkbox({ areas, handleFilters }) {
  const [checked, setchecked] = useState([])

  const handleToggle = c => {
    const currentcategoryId = checked.indexOf(c)
    const newcheckedcategoryid = [...checked]

    if (currentcategoryId === -1) {
      newcheckedcategoryid.push(c)
    }
    else {
      newcheckedcategoryid.splice(currentcategoryId, 1)
    }

    setchecked(newcheckedcategoryid)
    handleFilters(newcheckedcategoryid)
  }
  return areas.map((c, i) => (
    <div className='title is-6 has-text-white form-inline d-flex align-items-center py-1'
key={i}>

        <input type='checkbox' className='checkbox' onChange={(e) => { handleToggle(c._id)
}} value={checked.indexOf(c._id === -1)} />
        <label className='tick ml-5'>{c.name}</label>
        <span className="check" />

    </div>

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    ))  
  }  
  export default Checkbox;
```

**8.2.1.7.4 GroupNum.js**

```
export const groups = [  
  {  
    _id: 0,  
    name: "Any",  
    array: []  
  },  
  {  
    _id: 1,  
    name: "1",  
    array: [0, 1]  
  },  
  {  
    _id: 2,  
    name: "2",  
    array: [1, 2]  
  },  
  {  
    _id: 3,  
    name: "3",  
    array: [2, 3]  
  },  
  {  
    _id: 4,  
    name: "4",  
    array: [3, 4]  
  }  
]
```



**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

## 8.2.1.7.5 MovieCard.js

```
import React from 'react';
import '../Assets/Styles/card.css'
function MovieCard({ movie }) {
  return (
    <div className="movie-card">
      <img src={"http://localhost:8070/" + movie.image} alt="movie poster"
className="movie-poster" />
      <div className="movie-info">
        /* <h1 className="movie-title">I am not okay with this</h1> */
        /* <button className="btn-watch">Watch Now</button> */
      </div>
    </div>

  );
}

export default MovieCard;
```

## 8.2.1.7.6 Radiobox.js

```
import React, { useState } from 'react';
//import { Radio } from 'antd';

function Radiobox({ groups, handleFilters }) {

  const [Radio, setRadio] = useState('')

  const [value, setvalue] = useState(0)
  function handlechange(e) {
    handleFilters(e.target.value)
    setvalue(e.target.value)
  }
  return groups.map((p, i) => (
    <div key={i}>
      <input type="radio" value={` ${p._id}`} onChange={(e) => { handlechange(e) }}
name="rad" />
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
        <label className=' ml-5 title is-6 has-text-white'>{p.name}</label><br /> <br />
    </div >
  ))
  //
}

export default Radiobox;
```

## 8.2.1.7.7 ViewOne.js

```
import axios from 'axios';
import React, { useEffect, useState } from 'react';
import { useSelector, useDispatch } from 'react-redux'
import { Link } from 'react-router-dom';
import { getShows } from '../../../Actions/ShowActions';
import '../../../Assets/Styles/one.css'
import NavBar_Home from '../../../Components/Home/NavBar_Home';

function ViewOne(props) {
  const [movie, setmovie] = useState({});
  const [area, setarea] = useState([]);
  const { shows } = useSelector(state => state.shows);

  const dispatch = useDispatch()

  useEffect(() => {

    axios.get(`http://localhost:8280/movies/getmovie/${props.match.params.id}`).then((data) => {

      setmovie(data.data)
      setarea(data.data.area)

    }).catch((err) => {
      console.log(err)
    })

    // Get all "navbar-burger" elements
    const $navbarBurgers = document.querySelectorAll('.navbar-burger')
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
// Add a click event on each of them
$navbarBurgers.forEach(el => {
  el.addEventListener('click', () => {

    // Get the target from the "data-target" attribute
    const target = el.dataset.target;
    const $target = document.getElementById(target);

    // Toggle the "is-active" class on both the "navbar-burger" and the "navbar-
menu"

    el.classList.toggle('is-active');
    $target.classList.toggle('is-active');

  });
});

}, [])

useEffect(() => {
  dispatch(getShows(props.match.params.id))
}, [])
let range
const getTime = (ind) => {

  if (ind === 1) {
    range = "10.00am - 1.00pm"
  } else if (ind === 2) {
    range = "1.00pm - 4.00pm"
  } else if (ind === 3) {
    range = "4.00pm - 7.00pm"
  }
}
return (
  <>
    <NavBar_Home />
    <section className="hero is-fullheight-with-navbar movie-div all-home-flex">

      <div className="hero-body">
        <div className="container is-fluid">

          <div className="">
```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    <div className="" >
      <div className="columns one-movie" >
        <div className="column" style={{ backgroundColor: 'rgb(255
255 255 / 70%)', borderRadius: '20px' }}>
          <div className='columns' >
            <div className='column is-2'>
              <a href="#"><img src={"http://localhost:8070/"
+ movie?.image} alt="cover" style={{ height: '290px', borderRadius: '20px' }} /></a>
              <div>
                {area.map(s => (
                  <span className="tag ml-
1">{s.name}</span>
                )))}
              </div>
            </div>
            <div className='column'>
              <div className="title is-2 has-text-danger-
dark " >{movie?.name}</div>
            </div>
          </div>
          <div className='columns'>
            <div className='column'>
              <div className="description">
                <div className="column2">
                  <div className='title is-5 has-text-
justified'>{movie.description}</div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
    {shows.length > 0 ? <>

      <div className='columns'>
        <div className='column'>
          <div className='container'>
            <table className='table is-hoverable is-
fullwidth'>

              <tr>

```

### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    <th>No</th>
    <th>Hall Name</th>
    <th>Date</th>
    <th>Time</th>
    <th style={{ width: '120px'

}}>Action</th>

|| (localStorage.getItem('atoken')) ? (
    <td><Link
to={`/booking/${s._id}`}><button className='button is-success is-fullwidth'
>Book</button></Link></td>
    ) : (
    <td ><Link
to={`/login`} ><button className='button is-success is-fullwidth'
disabled={!localStorage.getItem('token') || localStorage.getItem('atoken')}>Please Login to
Book</button></Link></td>

    )}
  </tr>
)}}
</table>
</div>
</div>
</div>
</> : <>
  <div className='columns'>
    <div className='column'>
      <div className="container">
        <div className="card">
          <article className="message is-success">
            <div className="message-body">

```

## Assignment II – REST API Project

**SE3020 – Distributed Systems**

**Semester 1**

Allocated Yet !!!</strong>

<strong>There are No shows

</div>

</article>

</div>

</div>

</div>

</div>

</>}

</div>

</div>

</div>

</div>

</section>

</>

);

}

export default ViewOne;

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.2.2 Layouts****8.2.2.1 Layout1.js**

```
import React from 'react'

function Layout1() {
  return (
    <div>Layout1</div>
  )
}

export default Layout1
```

**8.2.3 Pages****8.2.3.1 Home.js**

```
import React, { useEffect } from 'react'
import { Link } from 'react-router-dom'
import "../Assets/Styles/home.css"
import NavBar_Home from '../Components/Home/NavBar_Home';
import { Slide } from 'react-slideshow-image';
import 'react-slideshow-image/dist/styles.css'
import '../Assets/Styles/Slider.css'

function Home() {
  useEffect(() => {
    // Get all "navbar-burger" elements
    const $navbarBurgers = document.querySelectorAll('.navbar-burger')

    // Add a click event on each of them
    $navbarBurgers.forEach(el => {
      el.addEventListener('click', () => {

        // Get the target from the "data-target" attribute
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
const target = el.dataset.target;
const $target = document.getElementById(target);

// Toggle the "is-active" class on both the "navbar-burger" and the "navbar-menu"
el.classList.toggle('is-active');
$target.classList.toggle('is-active');

});
});

}, [])

const slideImages = [
  {
    url: 'https://raw.githubusercontent.com/brixiobodino/coffeholic/main/banner.jpg',
    // caption: 'Slide 1'
  },
  {
    url: 'https://i.ytimg.com/vi/586CpeypGkQ/maxresdefault.jpg',
    // caption: 'Spider Man'
  },
  {
    url: 'https://images.indianexpress.com/2022/05/doctor-strange-madness-multiverse-review-
.jpg',
    // caption: 'Doctor Strange'
  },
];

return (

<>
<NavBar_Home />

<div className='home-div'> </div>

<section className="hero is-fullheight-with-navbar all-home-flexs">
  {/* <div className="slide-container">
    <Slide>
      {slideImages.map((slideImage, index) => (
        <div className="each-slide" key={index}>
          <div style={{ 'backgroundImage': `url(${slideImage.url})` }}>
```



### Assignment II – REST API Project

SE3020 – Distributed Systems

Semester 1

```

    <span>{slideImage.caption}</span>
  </div>
</div>
  )})
</Slide>
</div> */}
<div className="hero-body ">
  <div className="container ">
    <div className='is-vcentered '>
      <div className='content '>
        <div className='columns'>

          {/* Landing page Title */}
          <div className="column is-5 is-align-self-center">
            <h1 className="title is-1 is-light is-semibold is-spaced main-title has-
text-white has-text-centered">Get Your All Movies </h1>
            <h2 className="subtitle is-5 is-light is-thin has-text-white has-text-
centered">

              Netflix, Amazon Prime, Hindi, English , Korean
            </h2>
            {/* CTA */}
            <div className="buttons is-justify-content-center">
              <Link to="/all"><button className="button is-success">Go to All
Movies</button></Link>
            </div>
          </div>
          {/* Hero image */}
          <div className="column is-7">
            {/* <figure className="image" style={{ width: '70%' }}>
              <img src={netflix} alt="Pipeline SVG" />
            </figure> */}
            <div className="slide-container">
<Slide>
  {slideImages.map((slideImage, index) => (
    <div className="each-slide" key={index}>
      <div style={{ 'backgroundImage': `url(${slideImage.url})` }}>
        {/* <span>{slideImage.caption}</span> */}
      </div>
    </div>
  )})
</Slide>

```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
</div>
    </div>
  </div>
</div>
</div>
</div>
</div>
</section>
```

```
</>
```

```
)
}
```

```
export default Home
```

**8.2.3.2 Home2.js**

```
import React from 'react'
```

```
function Home2() {
  return (
    <div>Home2</div>
  )
}
```

```
export default Home2
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.2.4 Reducers****8.2.4.1 CategoryReducer.js**

```
const initialState = {
  categories: []
};

const CategoryReducer = (state = initialState, action) => {
  switch (action.type) {
    case 'GET_ALL_CATE': {
      return {
        ...state,
        categories: action.payload
      }
    }

    default:
      return state;
  }
}

export default CategoryReducer;
```

**8.2.4.2 Halls.js**

```
const initialState = {
  halls: []
};

const HallReducer = (state = initialState, action) => {
  switch (action.type) {
    case 'GET_ALL_HALL': {
      return {
        ...state,
        halls: action.payload
      }
    }

    default:
      return state;
  }
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
}  
}  
export default HallReducer;
```

**8.2.4.3 index.js**

```
import MovieReducer from "../MovieReducer";  
import { createStore, combineReducers } from "redux";  
import HallReducer from "../Halls";  
import CategoryReducer from "../CategoryReducer";  
import ShowReducer from "../Show";  
  
const allreducers = new combineReducers({  
  movies: MovieReducer,  
  halls: HallReducer,  
  categories: CategoryReducer,  
  shows: ShowReducer  
});  
  
export default allreducers;  
import MovieReducer from "../MovieReducer";  
import { createStore, combineReducers } from "redux";  
import HallReducer from "../Halls";  
import CategoryReducer from "../CategoryReducer";  
import ShowReducer from "../Show";  
  
const allreducers = new combineReducers({  
  movies: MovieReducer,  
  halls: HallReducer,  
  categories: CategoryReducer,  
  shows: ShowReducer  
});  
  
export default allreducers;
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.2.4.4 MovieReducer.js**

```
const initialState = {
  movies: []
};

const MovieReducer = (state = initialState, action) => {
  switch (action.type) {
    case 'GET_ALL_MOVIES': {
      return {
        ...state,
        movies: action.payload
      }
    }

    default:
      return state;
  }
}

export default MovieReducer;
```

**8.2.4.5 Show.js**

```
const initialState = {
  shows: []
};

const ShowReducer = (state = initialState, action) => {
  switch (action.type) {
    case 'GET_ALL_SHOW': {
      return {
        ...state,
        shows: action.payload
      }
    }
  }
}
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
    default:
      return state;
  }
}
export default ShowReducer;
```

**8.2.5 App.js**

```
import '../src/Assets/Styles/App.css';
import { BrowserRouter, Route } from 'react-router-dom';
import Home from './Pages/Home';
import Home2 from './Pages/Home2';
import Addmovie from './Components/Movie/Addmovie';
import Allmovies from './Components/Movie/Allmovies';
import AddHall from './Components/Hall/AddHall';
import ViewOne from './Components/Movie/ViewOne';
import Addshow from './Components/Show/Addshow';
import Booking from './Components/Show/Booking';
import Qr from './Components/Qr/Qr';
import QrRead from './Components/Qr/QrRead';
import Cart from './Components/Show/Cart';
import Login from './Components/User/login';
import Register from './Components/User/register';
import Profile from './Components/User/profile';
import Mybookings from './Components/User/Mybookings';
import AdminLogin from './Components/Admin/AdminLogin';
import AdminDashboard from './Components/Admin/AdminDashboard';
import AllMovies from './Components/Admin/AllMovies';
import AllHalls from './Components/Admin/AllHalls';
import AllShows from './Components/Admin/AllShows';
import EditShow from './Components/Admin/EditShow';
import EditMovie from './Components/Admin/EditMovie';
import EditHall from './Components/Admin/EditHall';
import Allusers from './Components/Admin/Allusers';
import AllBookings from './Components/Admin/AllBookings';
import Update from './Components/User/update';
import AddCategory from './Components/Admin/AllCategory';
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
import AllCategory from './Components/Admin/AllCategory';
import Addcategories from './Components/Admin/Addcategories';
import Addadmin from './Components/Admin/Addadmin';
import Adduser from './Components/Admin/Adduser';
import Adminprofile from './Components/Admin/Adminprofile';
import AdminEdit from './Components/Admin/AdminEdit';

function App() {
  return (

    <BrowserRouter>
      <Route path="/" exact component={Home} />
      <Route path="/home" exact component={Home2} />
      <Route path="/addmovie" exact component={Addmovie} />
      <Route path="/all" exact component={Allmovies} />
      <Route path="/addhall" exact component={AddHall} />
      <Route path="/view/:id" exact component={ViewOne} />
      <Route path="/addshow" exact component={Addshow} />
      <Route path="/booking/:id" exact component={Booking} />
      <Route path="/qr" exact component={Qr} />
      <Route path="/qrread" exact component={QrRead} />
      <Route path="/cart" exact component={Cart} />
      <Route path="/login" exact component={Login} />
      <Route path="/register" exact component={Register} />
      <Route path="/profile" exact component={Profile} />
      <Route path="/mybookings" exact component={Mybookings} />
      <Route path="/updateuser/:id" exact component={Update} />

      <Route path="/admin" exact component={AdminLogin} />
      <Route path="/dashboard" exact component={AdminDashboard} />
      <Route path="/allmovies" exact component={AllMovies} />
      <Route path="/allhalls" exact component={AllHalls} />
      <Route path="/allshow" exact component={AllShows} />
      <Route path="/updateshow/:id" exact component={EditShow} />
      <Route path="/updatemovie/:id" exact component={EditMovie} />
      <Route path="/updatehall/:id" exact component={EditHall} />
      <Route path="/users" exact component={Allusers} />
      <Route path="/bookings" exact component={AllBookings} />
      <Route path="/addcategory" exact component={Addcategories} />
      <Route path="/allcategory" exact component={AllCategory} />
      <Route path="/addadmin" exact component={Addadmin} />
    </BrowserRouter>
  );
}
```

## Assignment II – REST API Project

### SE3020 – Distributed Systems

### Semester 1

```
<Route path='/adduser' exact component={Adduser} />  
<Route path='/aprofile' exact component={Adminprofile} />  
<Route path='/adedit/:id' exact component={AdminEdit} />
```

```
</BrowserRouter>
```

```
);  
}
```

```
export default App;
```



**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.2.6 index.js**

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

import { composeWithDevTools } from 'redux-devtools-extension';
import { createStore, applyMiddleware, compose } from 'redux';
import allreducers from './Reducers';
import { Provider } from 'react-redux';
import thunk from 'redux-thunk';
//STORE

const store = createStore(
  allreducers,
  composeWithDevTools(
    applyMiddleware(thunk)
  )
);

ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>,
  document.getElementById('root')
);
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1****8.2.7 package.json**

```
{
  "name": "react_qrcode_generation",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.11.9",
    "@testing-library/react": "^11.2.3",
    "@testing-library/user-event": "^12.6.2",
    "axios": "^0.27.2",
    "moment": "^2.29.3",
    "qrcode": "^1.4.4",
    "react": "^17.0.1",
    "react-dom": "^17.0.1",
    "react-qrcode-reader": "^2.2.1",
    "react-redux": "^7.2.8",
    "react-router-dom": "^5.3.0",
    "react-scripts": "4.0.1",
    "react-slideshow-image": "^3.7.5",
    "react-stripe-checkout": "^2.6.3",
    "redux": "^4.2.0",
    "redux-devtools-extension": "^2.13.9",
    "redux-thunk": "^2.4.1",
    "web-vitals": "^0.2.4"
  },
  "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": {
```

**Assignment II – REST API Project****SE3020 – Distributed Systems****Semester 1**

```
"production": [  
  ">0.2%",  
  "not dead",  
  "not op_mini all"  
],  
"development": [  
  "last 1 chrome version",  
  "last 1 firefox version",  
  "last 1 safari version"  
]  
}  
}
```