

**Sri Lanka Institute of Information Technology**

**Distributed Systems (SE3020)**

**Assignment 02**

**Train ticket reservation Application**

**Assignment Report**

Submitted By: IT17155380 (M.R.M. Rushaid)

**Table of Contents**

[**Introduction** 2](#_Toc142933)

[**High Level Architectural Diagram** 4](#_Toc142934)

[**System Workflow** 5](#_Toc142935)

[**System Workflow Scenario Execution** 6](#_Toc142936)

[**Scenario** 6](#_Toc142937)

[**Authentication and Security Mechanism** 10](#_Toc142938)

[**Appendix** 13](#_Toc142939)

[**Frontend**  13](#_Toc142940)

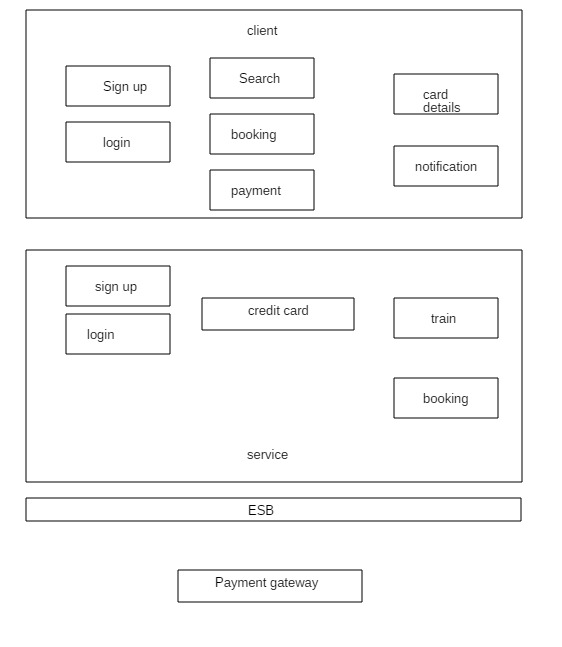
[**Backend**  31](#_Toc142941)

Introduction

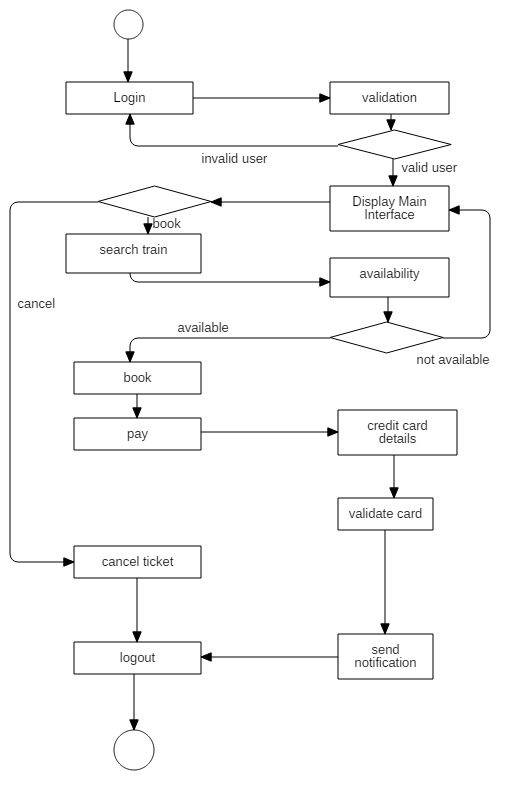
This web application is developed using react libraries for web client. MongoDB, Express and Node.js were used for REST API, backend. WSO2 EI was used to route the payment. Passport authentication and JWT Json webtoken dependancies were used for security mechanism of users. This online train ticket reservation application is a online ticket booking website. It is capble of booking ticket according to the available trains. This website has following requirements fullfilled and it is comprised of,

1. A central service that will store all the information
2. A client that will provide information on ticket price
3. Registered users able to view booking
4. Search trains
5. Users get notification on booking
6. Admin user make changes on train details
7. User simply log and book ticket

High level Architectural Diagram



System Workflow



System workflow scenario execution

User has to sign up and store his details.

Then he can login to the main page.

When user has logged in , his dashbord will be displayed.

He can select train information there and then select go to payment.

Next his booking and cost will be displayed in payment interface.

He can confirm the payment if his given details are correcr.

Then he will be directed to credit card information page.

User has to give card information such as name on card, card number etc.

Then he can confirm pay.

A notification of booking will be sent to user email.

Scenario

Front end of the application has the files app.js, payment.js and creditcard.js. as components. Back end is divided into 3 main folders as api’s, model’s, and validation files. Model folder has schemas which connect with ESB. ESB then connect with api’s. React uses this api routes and display the contents to user in the UI.

Authentication / Security Mechanism

Validator dependancy and Passport dependancy are used for validation and authentication purposes. There are validation files as follow,

Creditcard.js – validate the credit card details

Register.js – validate user login and sign up forms . Uses JWT token to return the logged in user.

Search.js – validate the train search form

is-empty.js – validate form fields

Appendix

This web app which allow people to search by places of departure and destination to whatever place you want to go is single page. This was done using MERN stack method. MERN is Mongoose, Express, React and node. Mongoose is using MongoDB and it can be connected to MLAB or MongoDB atlas. Node runs the back end of the project.

For extra development of the system following suggestions are made.

You can add a google maps api to display a map and you can get google api places to search by address and display.

The informations are free and everything is legal.

Mongodb/mongoose Schema on mlab for database, React-Redux Bootstrap, reactstrap for front-end, expressJs/NodeJs ecosystem for backend,.

Furthermore it can be used Heroku for hosting. Amazon Route 53 and Amazon S3 Bucket for domain name and routing to heroku.

Some part of the project was developed with the help of MERN tutorial guides.

This report is submitted to the lecture-in-charge of Distributed Systems module, 3rd year, Software Engineering, Sri lanka Institute of Management and Technology.

Front End

App.js

import React, { Component } from "react";

import logo from './logo.svg';

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

import './App.css';

class App extends Component{

constructor(props) {

super(props);

this.handlechangefrom = this.handlechangefrom.bind(this);

this.handlechangetogo = this.handlechangetogo.bind(this);

this.handlechangequantity = this.handlechangequantity.bind(this);

this.handleSubmit = this.handleSubmit.bind(this);

this.state = {

from: " ",

togo: " ",

quantity: " ",

formErrors:{

}

}

}

handlechangefrom = e =>{

// e.preventDefault();

this.setState({

from : e.target.value

})

}

handlechangeto = e =>{

// e.preventDefault();

this.setState({

togo: e.target.value

})

}

handlechangequantity = e =>{

this.setState({

quantity: e.target.value

})

}

handleSubmit = e => {

e.preventDefault();

// console.log(this.state)

const newData = {

from : this.state.from,

togo : this.state.togo,

quantity: this.state.quantity

};

axios.post("http://localhost:3000/api/booking", newData)

.then(res => {

console.log(res.data)

})

this.setState({

from : " ",

to : " ",

quantity: ""

})

}

render() {

return(

   <div className="row " style={{marginTop:"20px",marginLeft:"10px"}}>

<form>

<i className="form-horizontal">Train Ticket Reservation</i>

<input id=""

type="text"

id="from"

className="form-control "

placeholder="From"

style={{ width: "200px" }}

value={this.state.from}

onChange={this.handlechangefrom}

/>

<span> to </span>

<input

id=""

type="text"

id="to"

className="form-control "

placeholder="To"

style={{ width: "200px" }}

value={this.state.togo}

onChange={this.handlechangetogo}

/>

<span>Number of Tickets</span>

<input

id=""

type="text"

id="quantity"

className="form-control "

placeholder="Number Of Tickets"

style={{ width: "100px" }}

value={this.state.quantity}

onChange={this.handlechangequantity}

/>

<Link to="/admin" >

     <button className="btn btn-box-tool" type="submit" style = {{marginBottom:"10px"}}>

     Search

</button>

</Link>

</form>

</div>

)

}

}

export default App

Payment.js

import React, { Component } from "react";

import axios from "axios";

import { Link,BrowserRouter,Route } from 'react-router-dom';

import payment from "./payment";

const Todo = props => (

<tr>

<td>{props.todo.name}</td>

<td>{props.todo.email}</td>

<td>{props.todo.number}</td>

<td>{props.todo.password}</td>

<td>

<Link to={"/user/edit/"+props.todo.\_id} className=" " > <i className="material-icons">edit</i> </Link>

</td>

</tr>

)

class Useredit extends Component{

constructor(props) {

super(props);

this.state = {todos: []};

}

componentDidMount() {

axios.get("http://localhost:3000/api/user/edit")

.then(response => {

this.setState({ todos: response.data });

})

.catch(function (error){

console.log(error);

})

}

componentDidUpdate(){

axios.get("http://localhost:3000/api/user/edit")

.then(response => {

this.setState({ todos: response.data });

})

.catch(function (error){

console.log(error);

})

}

todoList() {

return this.state.todos.map(function(currentTodo, i){

return <Todo todo={currentTodo} key={i} />;

})

}

    render() {

return(

   <div className="row " style={{marginTop:"20px",marginLeft:"10px"}}>

<form>

<i className="form-horizontal">Train Ticket Reservation</i>

         <table className="" >

<thead>

<tr>

</tr>

</thead>

<tbody>

<tr>

<td> From : Colobo-fort</td>

<td> To : Kandy </td>

<td> Cost :158</td>

</tr>

<Link to="/admin" >

                <button className="btn btn-box-tool" type="submit" style = {{marginBottom:"10px"}}>

                Confrim payment

                </button>

                </Link>

</tbody>

</table>

</form>

</div>

)

}

}

export default Useredit

creditcard.js

import React , {Component} from "react";

import axios from "axios";

class Creditcard extends Component{

constructor(props) {

super(props);

this.handlechangenameOnCard = this.handlechangenameOnCard.bind(this);

this.handlechangecardNumber = this.handlechangecardNumber.bind(this);

this.handlechangeAmount = this.handlechangeAmount.bind(this);

this.handlechangecvcNumber = this.handlechangecvcNumber.bind(this);

this.handleSubmit = this.handleSubmit.bind(this);

this.state = {

nameOnCard: " ",

cardNumber: " ",

Amount: " ",

cvcNumber:"",

formErrors:{

}

}

}

handlechangenamenameOnCard = e =>{

// e.preventDefault();

this.setState({

nameOnCard: e.target.value

})

}

handlechangcardNumber = e =>{

// e.preventDefault();

this.setState({

cardNumber: e.target.value

})

}

handlechangeAmount = e =>{

Amount: " ",

cvcNumber: ""

})

}

render(){

return(

<div>

<div className="row " style={{marginTop:"30px"}}>

<form className="form-horizontal" onSubmit={this.handleSubmit}>

< input

id=""

type="text"

id="from"

className="form-control "

placeholder="Name on card"

style={{ width: "200px" }}

value={this.state.from}

onChange={this.handlechangefrom}

/>

<span> to </span>

<input

id=""

type="text"

id="to"

className="Card Number"

placeholder="To"

style={{ width: "200px" }}

value={this.state.togo}

onChange={this.handlechangetogo}

/>

<span>Number of Tickets</span>

<input

id=""

type="text"

id="quantity"

className="CVC"

placeholder="Number Of Tickets"

style={{ width: "100px" }}

value={this.state.quantity}

// e.preventDefault();

this.setState({

Amount: e.target.value

})

}

handlechangecvcNumber= e => {

// e.preventDefault();

this.setState({

cvcNumber: e.target.value

})

}

handleSubmit = e => {

e.preventDefault();

// console.log(this.state)

console.log(`Form submitted:`);

console.log(`Todo Description: ${this.state.nameOnCard}`);

console.log(`Todo Responsible: ${this.state.cardNumber}`);

console.log(`Todo Priority: ${this.state. Amount}`);

console.log(`Todo Priority: ${this.state. cvcNumber}`);

const newData = {

nameOnCard : this.state.nameOnCard,

mail : this.state.cardNumber,

number: this.state.Amount,

password:this.state.cvcNumber

};

axios.post("http://localhost:3000/api/card/add", newData)

.then(res => {

console.log(res.data)

})

this.setState({

nameOnCard: " ",

cardNumber: " ",

onChange={this.handlechangequantity }

     />

<input

id=""

type="text"

id="quantity"

className="CVC"

placeholder="Amount"

style={{ width: "100px" }}

value={this.state.quantity}

onChange={this.handlechangequantity}

/>

<Link to="/admin" >

     <button className="btn btn-box-tool" type="submit" style = {{marginBottom:"10px"}}>

     Pay

</button>

</Link>

</form>

</div>

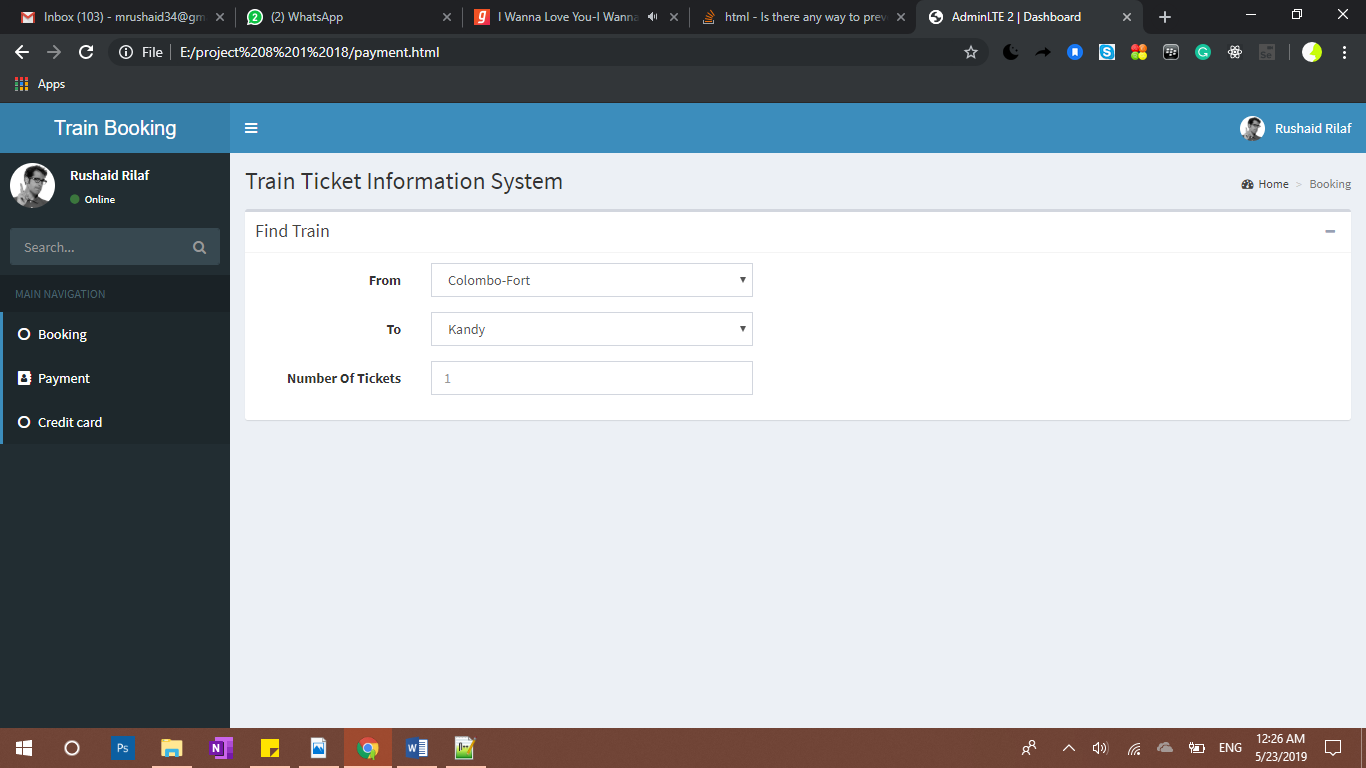
</div>

)

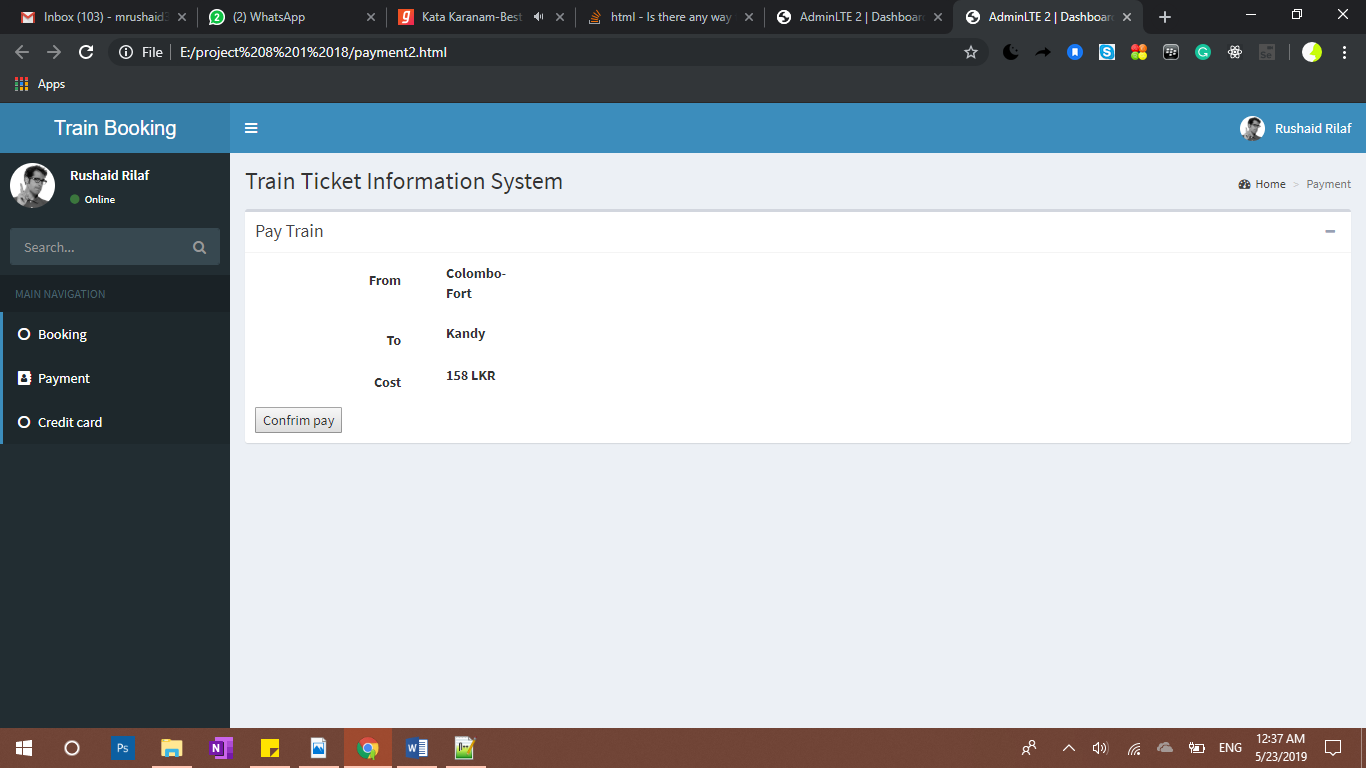
}

}

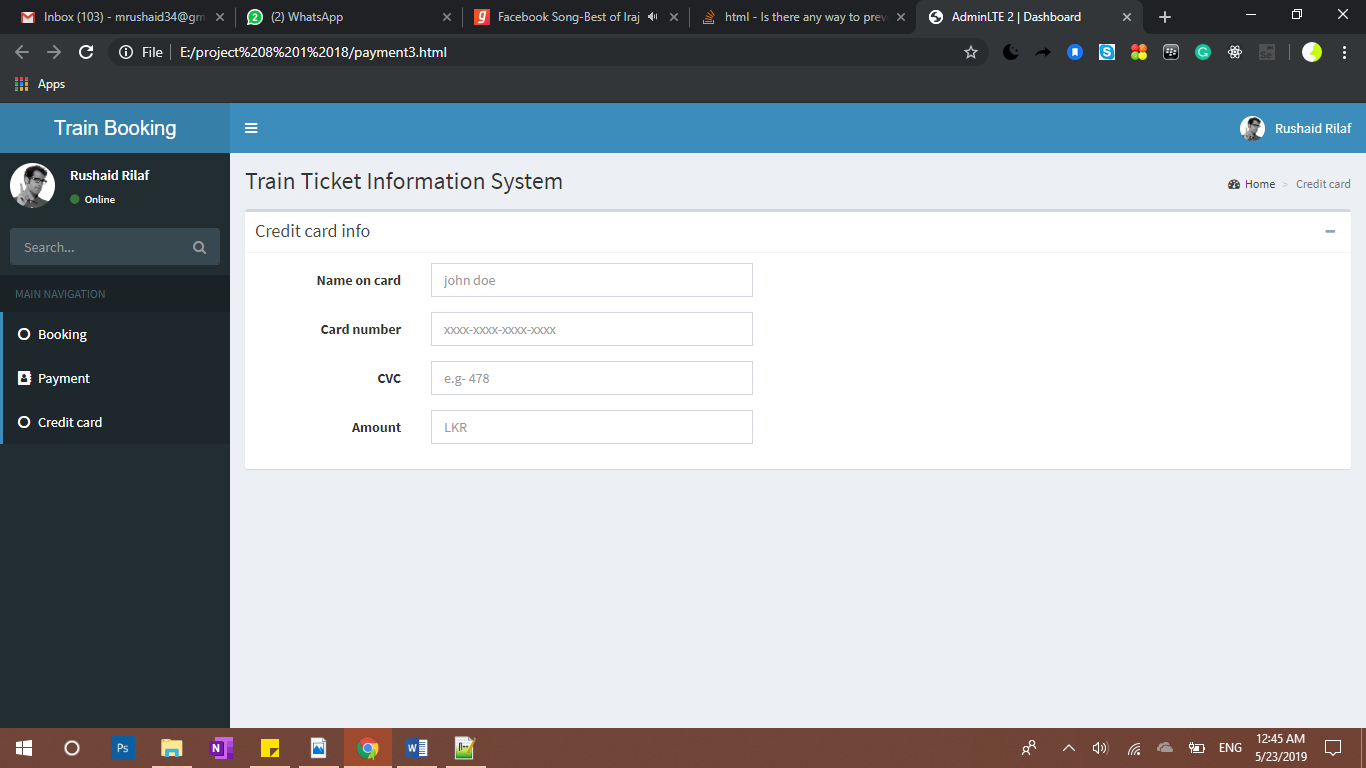
export default Creditcard;

Find train ( App.js front )

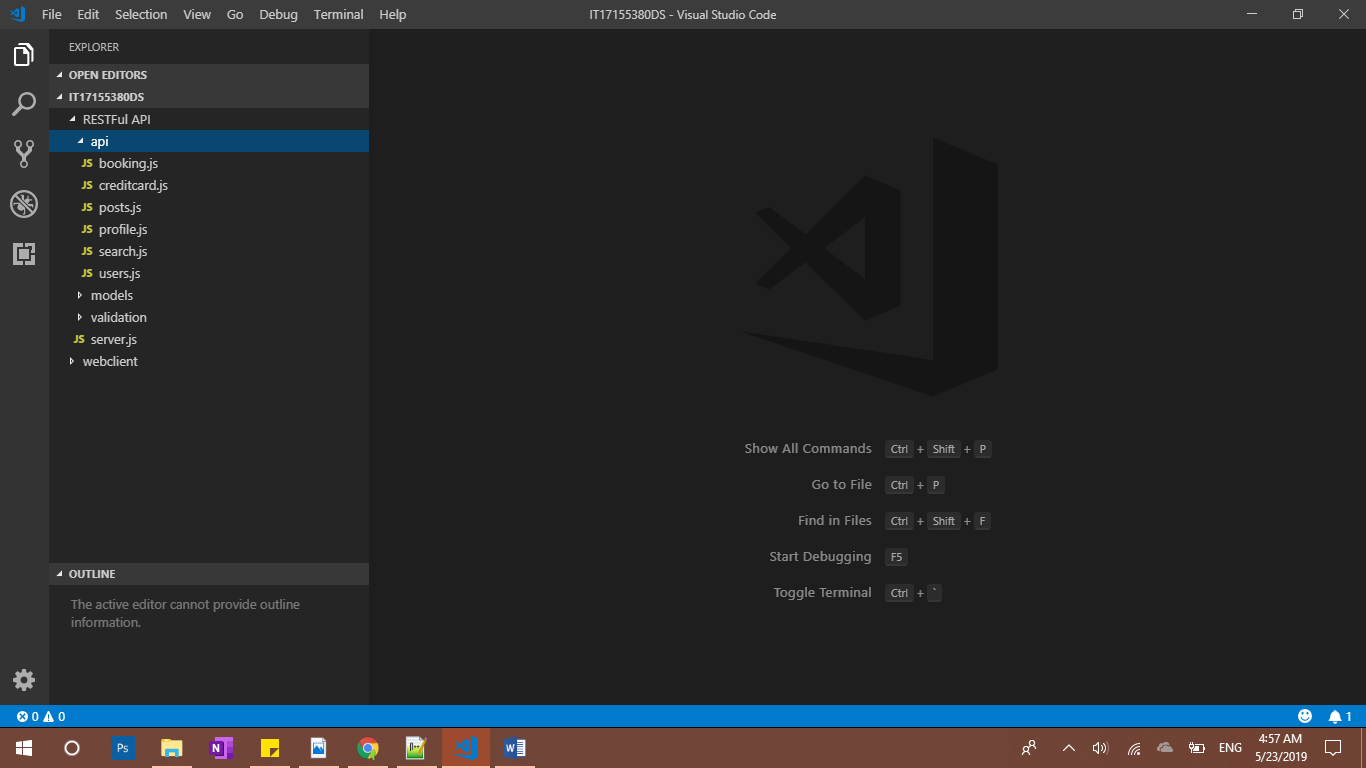
Pay train (payment.js front)

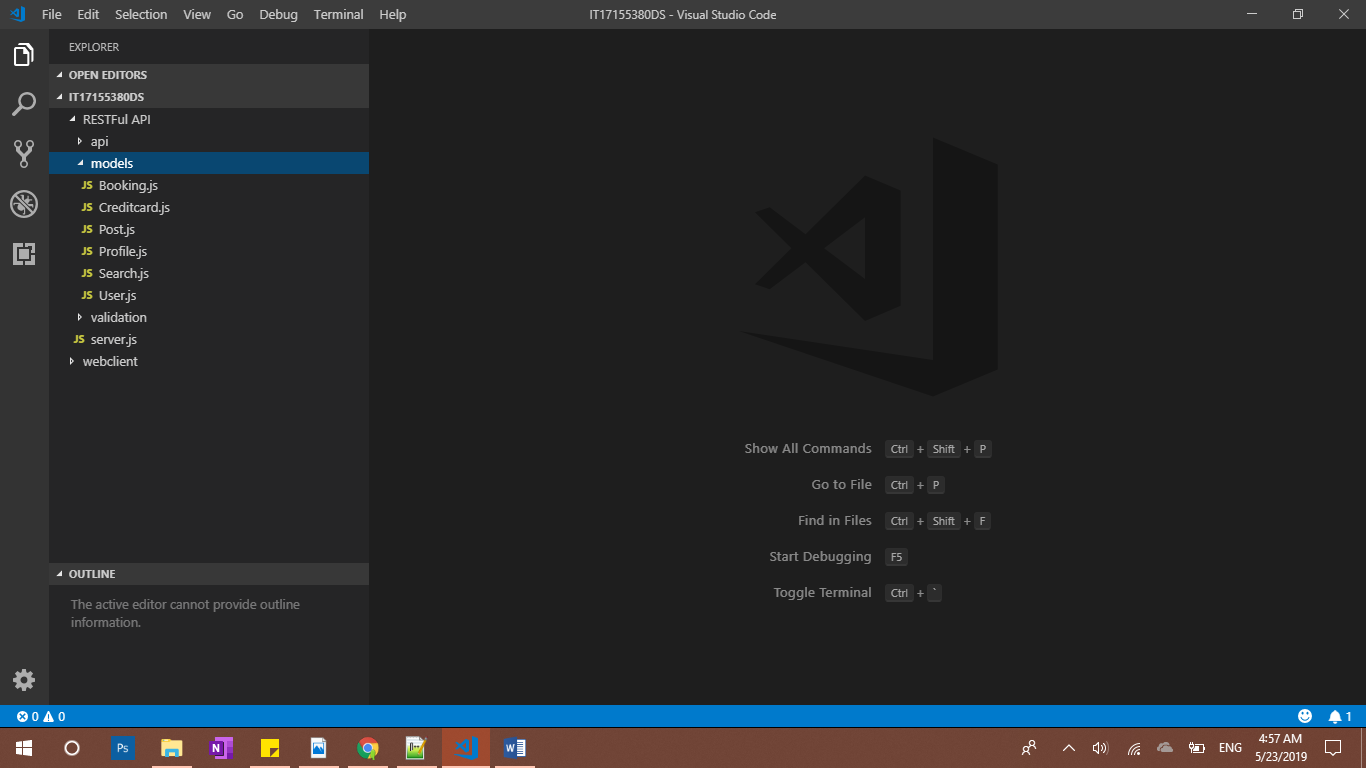


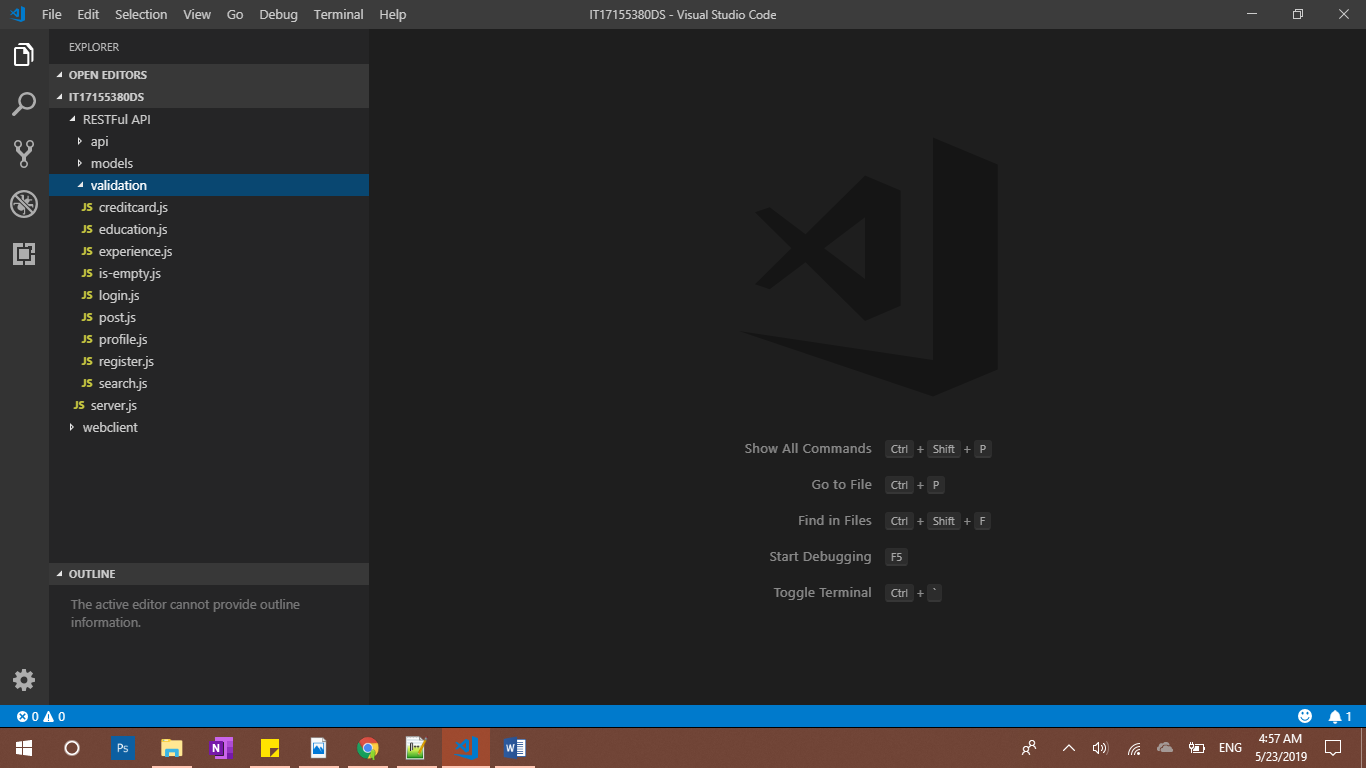
credit card ( creditcard.js front )



Backend

Api files.

model files ( contains schema )

validation files

*End*