



Distributed Systems - SE3020

Assignment 02 – Rest API

Train Ticket reservation Application Report

Submitted By:

IT17051026 | J.M.T. Jayasekara

On: 2019-05-21

01.Introduction

1.1 System Overview

This system is a web application that allows the user to login and reserve train tickets and make the relevant payments either using credit cards or using the mobile phone number which would credit the user's mobile bill. The system can connect to Sampath Bank payment gateway for credit card transactions and Dialog service to credit the mobile bill. Furthermore, the system allows a discount of 10% if the user is a government employee. The government employee status can be verified using a government dept's service that verifies whether a person is a government employee, given the NIC no. The system send an automated email and SMS confirming the payment.

1.2 Technologies Used

As the requirement of the application is to be a REST API, and the usage of SOA principles it is a combination of a Node.js and Express.js which is accompanied by a MongoDB database. Some other resources too were integrated for security, authentication, password encryption, bcrypt protection and sending confirmation emails and SMS.

Technologies used for the Frontend:

- JavaScript.
- Handlebars.js - templating engine that can separate the generation of HTML from the rest of JavaScript.
- jQuery - jQuery library for dynamically changing the content of the web page.
- HTML/CSS/Bootstrap - For styling the web pages

Technologies used for the Backend:

The backend restful services have been implemented using MEN Stack.

- Mongo DB and Mongoose
- Express.js
- Node.js

Technologies used for Authentication:

- Passport - an authentication middleware for Node.js. used for authenticating the user login and signup.
- CSURF - Protects Node.js app from Cross-Site Request Forgery

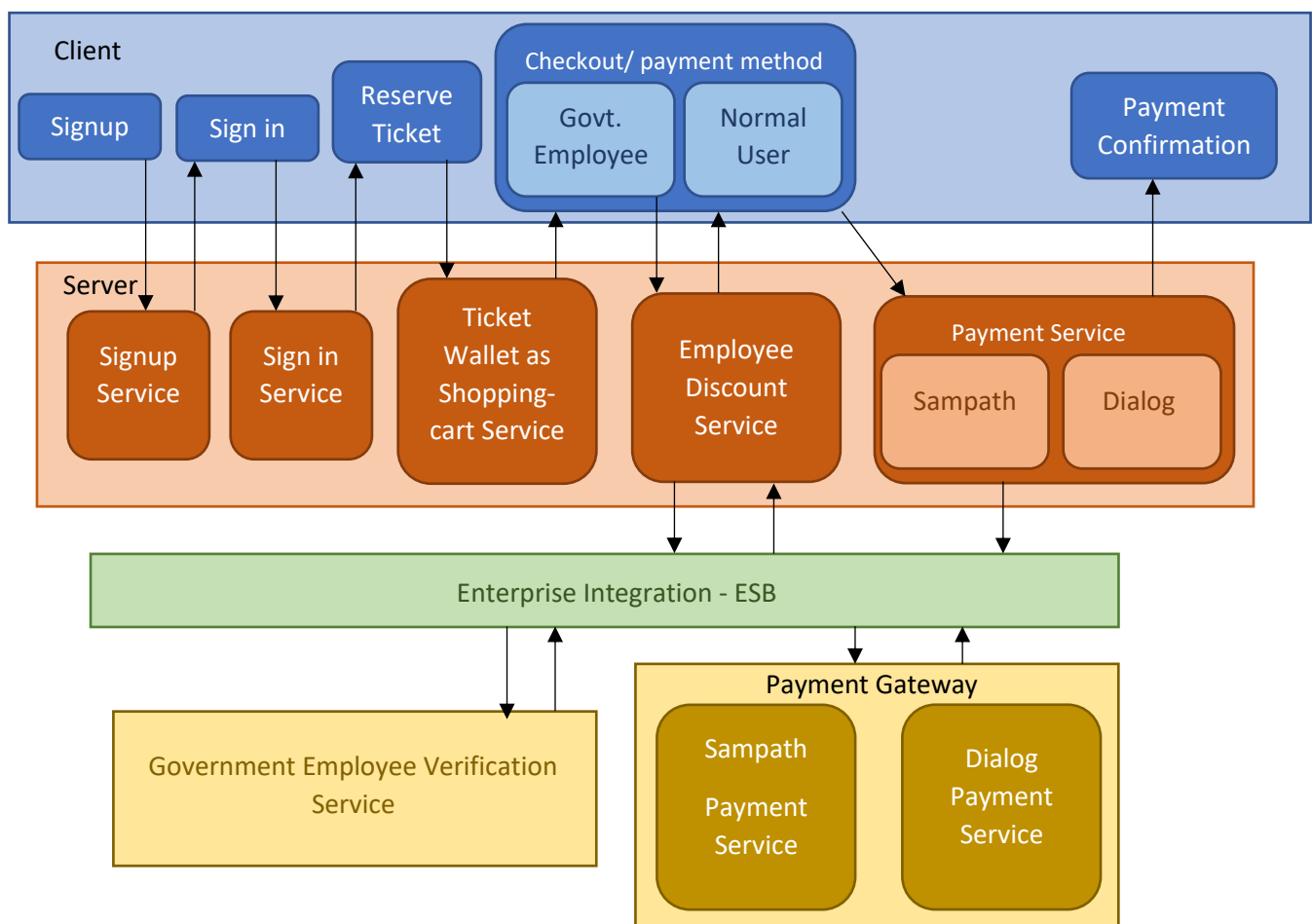
Technologies used for Online Payments:

- **Stripe** - a third-party resource which eliminate needless complexity and details when implementing online payments making it easy.

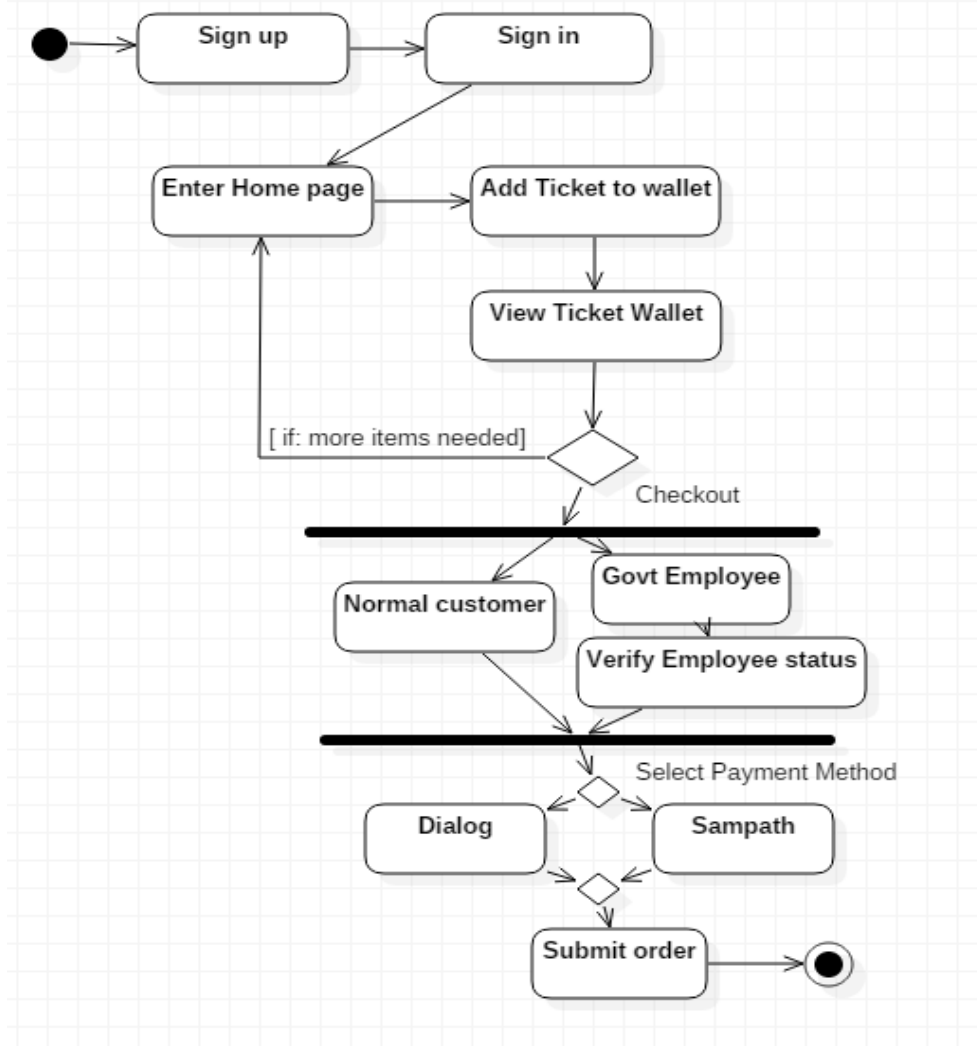
Technologies used to send confirmation messages:

- **Nodemailer** - a module for Node.js applications to allow email sending. Here it is used to send confirmation email to customer confirming the payment. As it uses SMTP service the emails are not actually delivered and are captured halfway; they can be viewed from my Ethereum email account.
 - Preview URL: <https://ethereal.email/message>
 - Username: mustafa66@ethereal.email
 - Password: nHTk7j1EJDn2Eunn
- **Nexmo** – facilitates sending and receiving messages on WhatsApp, Facebook Messenger, Viber, MMS, and SMS -- all with the Messages API. Create your own failover messaging flow to ensure delivery with the Dispatch API. Here messages are actually sent in real life as the number is verified for testing purposes in my Nexmo account.

1.3 High Level Architecture Diagram



1.4 Activity Diagram



02. System workflow scenario execution

Here the scenario of a verified government employee reserving tickets is shown step by step in both Sampath and Dialog Payment Gateways.

01. User Visits the site and SignUp

Train Ticket Reservation System

Sign Up

E-Mail

Password

[Sign Up](#)

2. User Signs in

Train Ticket Reservation System

Sign In

E-Mail

Password


[Sign In](#)

Don't have an account? [Sign up instead!](#)


03. User is redirected to home page

Train Ticket Reservation System


Ticket Selection Ticket Wallet Sign up Sign in User



Colombo-Jaffna
4077 - Yaldevi
Pick a date: Rs950
Reserve Ticket
Click multiple times to increase the number of tickets




Colombo-Batticaloa
6011 - Udaya Devi
Pick a date: Rs800
Reserve Ticket
Click multiple times to increase the number of tickets



Colombo-Kandy
1030 - Intercity Express
Pick a date: Rs700
Reserve Ticket
Click multiple times to increase the number of tickets

04. Initially Wallet is empty


Wallet Empty!

06. User can either opt for Govt. Emp Discount or Proceed to Check out!

05. User is Reserves 3 tickets (1- Kandy and 2- Batticaloa)

Ticket Wallet 3

Colombo-Kandy Rs700	Qty 1	<button>Reduce by 1</button>	<button>Remove All</button>
Colombo-Batticaloa Rs1600	Qty 2	<button>Reduce by 1</button>	<button>Remove All</button>

Total: Rs 2300

Checkout
Get Govt Employee Discount

7.1. User needs discount (has to submit NIC to verify Employment status)

Employment Verification

Enter NIC Number: Submit

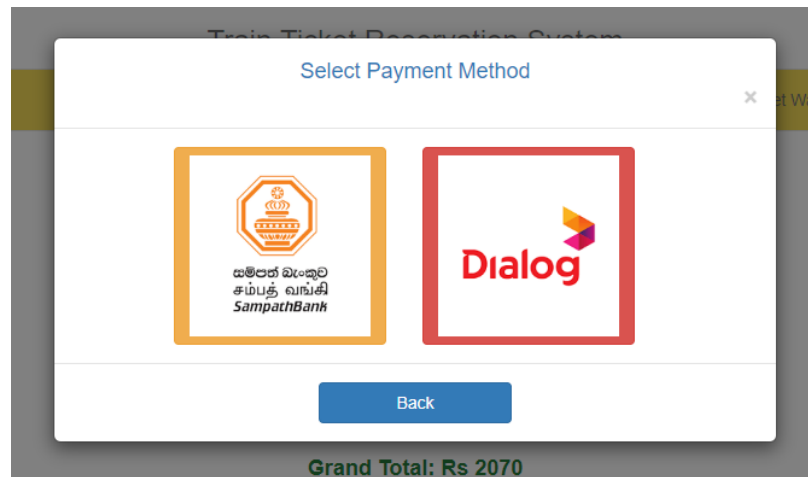
7.1. User is directed to the ticket wallet page again BUT with a discount!

Verified Government Employee!
You are now eligible for a discount of 10%


Total: Rs 2300
Discount: Rs 230
Grand Total: Rs 2070

Checkout




7.2. Now again both Govt. Employee and normal users have to choose payment method in a popup.



8.1.1. User selects Sampath Credit Card option



සමපත් බැංකුව
சம்பத் வங்கி
Sampath Bank

Checkout   

Verified Government Employee! ×

Your Grand Total: Rs2070

Name

Thilini Jayasekara

Address

237, Henpitagedara, Marandagahamula

Card Holder Name

Thilini Jayasekara

Credit Card Number

4242424242424242

Expiration Month

02

Expiration Year


2021




CVC

123

Pay now

8.2.1. User selects Dialog mobile credit option



Checkout   

Verified Government Employee! ×

Your Grand Total: Rs2070

Continuing with this payment using Dialog service will be credited to your mobile bill!!

Dialog Mobile Number:

0766446337

Pin Number:

1456

Pay now

8.1.2 Sampath payment completed notice displayed

Payment Successful!!! A confirmation SMS will be sent to your mobile shortly...

d

8.2.2 Dialog payment completed notice displayed

Payment Successful!!! A confirmation email will be sent to your registered email address shortly...

9. Sampath card details are validated and the payment is done through Stripe.

<https://dashboard.stripe.com/test/dashboard>

Filter

TEST DATA

+ New

Export


AMOUNT	DESCRIPTION		CUSTOMER	DATE	
\$23.00 USD	Succeeded ✓	Test Charge – ch_1EcbeeFfJ8FjZzgkN4tve4Ui	Thilini Jayasekara	2019/05/21 22:51:36	...

Payment method

Number	**** 4242	Owner	Thilini Jayasekara
Fingerprint	Q02JSN544m8eU3Jm	Address	No address
Expires	02 / 2021	Origin	United States 🇺🇸
Type	Visa credit card	CVC check	Passed ✓
ID	card_1EcbeeFfJ8FjZzgkN4tve4Ui		

9. Sampath payment verification Automated Email is sent to user via Nodemail

- As it uses SMTP service the emails are not actually delivered and are captured halfway; they can be viewed from my Ethereum email account. You can observe that the email and the payment has happened at the same time

 **Ethereal** Home FAQ Help Messages

Headers Envelope Source

[Public URL of this message](#)


Subject: Payment Confirmed
From: Sampath Payment Gateway - Train Ticket <reservetrainticket@sampath.lk>
To: <thilini1997@gmail.com>
Time: Today at 22:51
Message-ID: <10fbd0a6-1ae4-7615-f221-8367055a8ea9@sampath.lk>


☒ HTML ☐ Plaintext

Payment Confirmed!!!The tickets have been reserved successfully, the amount has been credited to your account


9. Dialog payment verification Automated Email is sent to user via Nexmo

- Here messages are actually sent in real life as the number “94766446337” is verified for testing purposes in my Nexmo account.

 NEXMO code: 0733. Valid for 5 minutes. 8:21 am

 Hello from Nexmo?<FREE SMS DEMO, TEST MESSAGE?> 8:22 am

Tuesday, 21 May 2019

 Ticket Reservation Payment Confirmed!<FREE SMS DEMO, TEST MESSAGE?> 10:58 pm

03. Authentication and security

Sign Up

Email is already in use.

E-Mail

Password

Sign Up

Sign In

Wrong password.

E-Mail

Password

Sign In

[Don't have an account? Sign up instead!](#)

Employment Verification

Enter NIC Number:

789456123

Submit

Please match the requested format.

Your card's expiration year is invalid.

Name

Thilini Jayasekara

Address

237, Henpitagedara, Marandagahamula

Card Holder Name

Thilini Jayasekara

Credit Card Number

4242424242424242

Expiration Month

02

Expiration Year

2015

Your card number is incorrect.

Name

Thilini Jayasekara

Address

237, Henpitagedara, Marandagahamula

Card Holder Name

Thilini Jayasekara

Credit Card Number

4242424242424|

Dialog Mobile Number:

54564|

Please match the requested format.

Pin Number

1234

Pay now

04.Appendix

- Front End
Layout.hbs

```
<!--Contains all the scripts and links needed in the app including Bootstrap CDN,
Jquery, and CSS Style sheets-->
<!DOCTYPE html>
<html>
<head>
  <title>{{title}}</title>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
  integrity="sha384-
1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDDdjZlpLegxhjVME1fgjWPGmkzs7"
crossorigin="anonymous">
  <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.8.2/css/all.css"
  integrity="sha384-
oS3vJWv+0UjzBfQzYUhtDYW+Pj2yciDJxpsK1OYPAYjqT085Qq/1cq5FLXAZQ7Ay"
crossorigin="anonymous">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/4.6.1/css/font-awesome.min.css">
  <link rel='stylesheet' href='/stylesheets/style.css' />
</head>
<body>
{{> header }}
<script src="https://code.jquery.com/jquery-1.12.3.min.js" integrity="sha256-
aaODHAgvwQWlbfOGXMeX+pc4PZIPsvn2hlsArYOhgXQ=" crossorigin="anonymous"></script>
<div class="container">
  {{{body}}}
</div>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"
integrity="sha384-0mSbJDEHialfmuBBQP6A4Qrprq5OVfW3PRR3j5ELqxs1yVqOtnepnHVP9aJ7xS"
crossorigin="anonymous"></script>
</body>
</html>
```

Header.hbs

```
<!-- This is the navigation bar of the app which contains the links to sign up,
sign in, view ticket wallet -->

<div class="text-dark bg-secondary text-center" style="color:rgba(30,30,30,0.65)">
  <h3>Train Ticket Reservation System</h3>
</div>
<nav class="navbar navbar-default" style="background-color:rgba(246,212,36,0.76)">
  <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
```

```

<a class="navbar-brand" href="/">    <i class="fas fa-ticket-alt" aria-
hidden="true"></i>    Ticket Selection</a>
</div>

<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
    <ul class="nav navbar-nav navbar-right">
        <li>
            <a href="/shopping-cart/">
                <i class="fas fa-wallet" aria-hidden="true"></i> Ticket
Wallet
                <span class="badge">{{session.cart.totalQty}}</span>
            </a>
        </li>
        <li>
            <a href="/user/signup"><i class="fas fa-user-plus"></i> Sign
up</a>
        </li>
        <li>
            <a href="/user/signin"><i class="fas fa-sign-in-alt"></i> Sign
in</a>
        </li>
        <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false"><i class="fa fa-user"
aria-hidden="true"></i> User <span class="caret"></span></a>
            <ul class="dropdown-menu">
                {{# if login }}
                <li><a href="/user/profile">User Account</a></li>
                <li role="separator" class="divider"></li>
                <li><a href="/user/logout">Logout</a></li>
                {{ else }}
                <li><a href="/user/signup">Sign up</a></li>
                <li><a href="/user/signin">Sign in</a></li>
                {{/if}}
            </ul>
        </li>
    </ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

```

dialogCheckout.hbs

```

<!--The Checkout page that is for the normal customers who do not have a discount,
to make payments through Dialog Mobile Number-->

<div class="row">
    <div class="col-sm-6 col-md-4 col-md-offset-4 col-sm-offset-3">
        <div class="text-center"></div>
        <h1 class="text-center">Checkout <i class="fas fa-mobile-alt"></i> <i
class="fas fa-sms"></i> <i class="fas fa-phone-square"></i></h1>
        <h4 class="text-center">Your Total: Rs{{total}}</h4>
        <div id="charge-error" class="alert alert-danger {{#if
noError}}hidden{{/if}}">
            {{errMsg}}
        </div>
        <form action="/sms" method="post" id="sms-form">
            <div class="row">
                <h4 class="text-center">Continuing with this payment using Dialog
service will be credited to your mobile bill!!</h4>
                <div class="col-xs-12">
                    <div class="form-group">

```

```

        <label for="name">Dialog Mobile Number:</label>
        <input type="text" id="mobile" class="form-control"
pattern="[9][4][0-9]{9}" required>
    </div>
</div>
<div class="col-xs-12">
    <div class="form-group">
        <label for="address">Pin Number:</label>
        <input type="text" id="pin" class="form-control"
pattern="[0-9]{4}" required>
    </div>
</div>
</div>
<button type="submit" class="btn btn-success btn-block">Pay
now</button>
</form>
</div>
</div>
</div>

<script type="text/javascript" src="javascripts/sms.js"></script>

```

discountCheckout.hbs

```

<!--The Checkout page that is for the Government Employees with discount , to make
payments through Sampath payment portal-->

<div class="row">
    <div class="col-sm-6 col-md-4 col-md-offset-4 col-sm-offset-3">
        <div class="text-center"></div>
        <h1 class="text-center">Checkout <i class="fab fa-cc-visa"></i> <i
class="fab fa-cc-mastercard"></i> <i class="fab fa-cc-stripe"></i></h1>
        <div class="alert alert-info alert-dismissible text-center">
            <button type="button" class="close" data-
dismiss="alert">&times;</button>
            <strong>Verified Government Employee!</strong><br>
        </div>
        <h4 class="text-center">Your Grand Total: Rs{{grandTotal}}</h4>
        <div id="charge-error" class="alert alert-danger {{#if
noError}}hidden{{/if}}">
            {{errMsg}}
        </div>
        <form action="/checkout" method="post" id="checkout-form">
            <div class="row">
                <div class="col-xs-12">
                    <div class="form-group">
                        <label for="name">Name</label>
                        <input type="text" id="name" class="form-control" required
name="name">
                    </div>
                </div>
                <div class="col-xs-12">
                    <div class="form-group">
                        <label for="address">Address</label>
                        <input type="text" id="address" class="form-control"
required name="address">
                    </div>
                </div>
            </div>
            <hr>
            <div class="col-xs-12">
                <div class="form-group">
                    <label for="card-name">Card Holder Name</label>
                    <input type="text" id="card-name" class="form-control"
required>
                </div>
            </div>
        </form>
    </div>
</div>

```

```

        </div>
    </div>
    <div class="col-xs-12">
        <div class="form-group">
            <label for="card-number">Credit Card Number</label>
            <input type="text" id="card-number" class="form-control"
required>
        </div>
    </div>
    <div class="col-xs-12">
        <div class="row">
            <div class="col-xs-6">
                <div class="form-group">
                    <label for="card-expiry-month">Expiration
Month</label>
                    <input type="text" id="card-expiry-month"
class="form-control" required>
                </div>
            </div>
            <div class="col-xs-6">
                <div class="form-group">
                    <label for="card-expiry-year">Expiration
Year</label>
                    <input type="text" id="card-expiry-year"
class="form-control" required>
                </div>
            </div>
        </div>
    </div>
    <div class="col-xs-12">
        <div class="form-group">
            <label for="card-cvc">CVC</label>
            <input type="text" id="card-cvc" class="form-control"
required>
        </div>
    </div>
    </div>
    <button type="submit" class="btn btn-success btn-block">Pay
now</button>
</form>
</div>
</div>

<script type="text/javascript" src="https://js.stripe.com/v2/"></script>
<script type="text/javascript" src="javascripts/checkout.js"></script>

```

discountDialogCheckout.hbs

```

<!--The Checkout page is for the Government Employees with discount , to make
payments through Dialog Mobile Number-->

<div class="row">
    <div class="col-sm-6 col-md-4 col-md-offset-4 col-sm-offset-3">
        <div class="text-center"></div>
        <h1 class="text-center">Checkout <i class="fas fa-mobile-alt"></i> <i
class="fas fa-sms"></i> <i class="fas fa-phone-square"></i></h1>
        <div class="alert alert-info alert-dismissible text-center">
            <button type="button" class="close" data-
dismiss="alert">&times;</button>
            <strong>Verified Government Employee!</strong><br>
        </div>
        <h4 class="text-center">Your Grand Total: Rs{{grandTotal}}</h4>
        <div id="charge-error" class="alert alert-danger {{#if

```

```

noError}}hidden{{/if}}">
    {{errMsg}}
</div>
<form action="/sms" method="post" id="sms-form">
    <div class="row">
        <h4 class="text-center">Continuing with this payment using Dialog
service will be credited to your mobile bill!!</h4>
        <div class="col-xs-12">
            <div class="form-group">
                <label for="name">Dialog Mobile Number:</label>
                <input type="text" id="mobile" class="form-control"
pattern="[9][4][0-9]{9}" required>
            </div>
            <!--The mobile
number shoud start with 94 followed by 9 other digits-->
            </div>
            <div class="col-xs-12">
                <div class="form-group">
                    <label for="address">Pin Number:</label>
                    <input type="text" id="pin" class="form-control"
pattern="[0-9]{4}" required>
                </div>
                <!-- Pin should be a
4 digit number-->
                </div>
            </div>
            <button type="submit" class="btn btn-success btn-block">Pay
now</button>
        </form>
    </div>
</div>

<script type="text/javascript" src="javascripts/sms.js"></script>

```

discountShoppingCart.hbs

```

<!--The Ticket wallet/ Shopping cart page that is for the Govt Employee who have a
discount,
After being verified that they are government employees by checking their NIC
number-->

{{# if products }}
    <div class="row">
        <div class="col-sm-6 col-md-6 col-md-offset-3 col-sm-offset-3">
            <ul class="list-group">
                {{# each products }}
                    <li class="list-group-item">
                        <span class="badge">{{ this.qty }}</span>
                        <strong>{{ this.item.title }}</strong>
                        <span class="label label-success">Rs{{ this.price }}</span>
                        <div class="btn-group">
                            <button class="btn btn-primary btn-xs dropdown-toggle"
type="button" data-toggle="dropdown">Action <span class="caret"></span></button>
                            <ul class="dropdown-menu">
                                <li><a href="/reduce/{{this.item._id}}">Reduce by
1</a></li>
                                <li><a href="/remove/{{this.item._id}}">Remove
All</a></li>
                            </ul>
                        </div>
                    </li>
                {{/each}}
            </ul>
        </div>
        <div class="col-sm-6 col-md-6 col-md-offset-3 col-sm-offset-3 text-center">
            <div class="alert alert-success alert-dismissible">

```

```

        <button type="button" class="close" data-dismiss="alert">&times;</button>
        <strong>Verified Government Employee!</strong><br> You are now eligible for
a discount of 10%
    </div>
</div>
<div class="row">
    <div class="col-sm-6 col-md-6 col-md-offset-3 col-sm-offset-3 text-center">
        <strong>Total: Rs {{ totalPrice }} </strong><br>
        <strong>Discount: Rs {{ discount }} </strong><br>
        <h4><strong><span style="color: #237c39; "> Grand Total: Rs {{
discountedPrice }} </span></strong></h4>
    </div>
</div>
<hr>
<div class="row">
    <div class="col-sm-6 col-md-6 col-md-offset-3 col-sm-offset-3">
        <button type="button" class="btn btn-primary btn-block" data-
toggle="modal" data-target="#myModal">
            Checkout
        </button>
    </div>
</div>

<div class="modal" id="myModal">
    <div class="modal-dialog">
        <div class="modal-content">

            <!-- Modal Header -->
            <div class="modal-header">
                <h4 class="modal-title text-center text-primary">Select Payment
Method</h4>

                <button type="button" class="close" data-
dismiss="modal">&times;</button>
            </div>

            <!-- Modal body -->
            <div class="modal-body">
                <table>
                    <tr>
                        <td>
                            <div class="col-sm-4 col-md-4 col-md-offset-4 col-
sm-offset-4">
                                <a href="/discountCheckout" type="button"
class="btn btn-warning"></a>
                            </div>
                        </td>
                        <td>
                            <div class="col-sm-4 col-md-4 col-md-offset-4 col-
sm-offset-4">
                                <a href="/discountDialogCheckout" type="button"
class="btn btn-danger"></a>

</div>
</td>
</tr>
</table>

</div>

<!-- Modal footer -->
<div class="modal-footer">
  <div class="col-sm-4 col-md-4 col-md-offset-4 col-sm-offset-4">
    <button type="button" class="btn btn-primary btn-block"
data-dismiss="modal">Back</button>
  </div>
</div>

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

{{ else }}
  <div class="row">
    <div class="col-sm-6 col-md-6 col-md-offset-3 col-sm-offset-3">
      <h2>No Items in Cart</h2>
    </div>
  </div>
</div>
{{ /if }}
```

index.hbs

```
<!--The Homepage is generated here, the ticket details are displayed by getting
them from the db-->

<div class="row" id="head">
  <div class="col-sm-6 col-md-4 col-md-offset-4 col-sm-offset-3">
    <div id="success" class="alert alert-success" {{#if
noMessages}}hidden{{/if}}>
      {{ successMsg }}
    </div>
  </div>
</div>

{{# each products }}
  <div class="row">
    {{# each this }}
      <div class="col-sm-6 col-md-4 bg-dark">
        <div class="thumbnail">
          
          <div class="caption">
            <h3 class="text-center">{{ this.title }}</h3>
            <p class="description text-center">{{ this.description
}}</p>
            <div class="price pull-right">Rs{{ this.price }}</div>
            <label>Pick a date: </label>&nbsp; &nbsp;<input type="date"
value="2019-06-21">
            <div class="clearfix">
              <table>
                <tr><label style="color: white"> space
</label></tr>
                <tr>
                  <td>
                    <label style="color: white"> Used as aspace
</label>
                  </td>
                  <td>
                    <a href="/add-to-cart/{{this._id}}"
class="btn btn-success btn-block pull-right" role="button">Reserve
Ticket</a><br/><br/>
                  </td>
                </tr>
              </table>
              <h5 class="text-center text-success"><i class="far fa-
lightbulb bg-warning"></i> Click multiple times to increase the number of
tickets</h5>
            </div>
          </div>
        </div>
      </div>
    {{/each}}
  </div>
{{/each}}
```


SampathCheckout.hbs

```
<!--The Checkout page that is for normal customers without discount , to make
payments through Sampath payment portal-->

<div class="row">
  <div class="col-sm-6 col-md-4 col-md-offset-4 col-sm-offset-3">
    <div class="text-center"></div>
    <h1 class="text-center">Checkout <i class="fab fa-cc-visa"></i> <i
class="fab fa-cc-mastercard"></i> <i class="fab fa-cc-stripe"></i></h1>
    <h4 class="text-center">Your Total: Rs{{total}}</h4>
    <div id="charge-error" class="alert alert-danger {{#if
noError}}hidden{{/if}}">
      {{errMsg}}
    </div>
    <form action="/checkout" method="post" id="checkout-form">
      <div class="row">
        <div class="col-xs-12">
          <div class="form-group">
            <label for="name">Name</label>
            <input type="text" id="name" class="form-control" required
name="name">
          </div>
        </div>
        <div class="col-xs-12">
          <div class="form-group">
            <label for="address">Address</label>
            <input type="text" id="address" class="form-control"
required name="address">
          </div>
        </div>
        <hr>
        <div class="col-xs-12">
          <div class="form-group">
            <label for="card-name">Card Holder Name</label>
            <input type="text" id="card-name" class="form-control"
required>
          </div>
        </div>
        <div class="col-xs-12">
          <div class="form-group">
            <label for="card-number">Credit Card Number</label>
            <input type="text" id="card-number" class="form-control"
required>
          </div>
        </div>
        <div class="col-xs-12">
          <div class="row">
            <div class="col-xs-6">
              <div class="form-group">
                <label for="card-expiry-month">Expiration
Month</label>
                <input type="text" id="card-expiry-month"
class="form-control" required>
              </div>
            </div>
            <div class="col-xs-6">
              <div class="form-group">
                <label for="card-expiry-year">Expiration
Year</label>
                <input type="text" id="card-expiry-year"
class="form-control" required>
              </div>
            </div>
          </div>
        </div>
      </div>
    </form>
  </div>
</div>
```

```

        <div class="col-xs-12">
            <div class="form-group">
                <label for="card-cvc">CVC</label>
                <input type="text" id="card-cvc" class="form-control"
required>
            </div>
        </div>
    </div>
    <button type="submit" class="btn btn-success btn-block">Pay
now</button>
</form>
</div>
</div>

<script type="text/javascript" src="https://js.stripe.com/v2/"></script>
<script type="text/javascript" src="javascripts/checkout.js"></script>

```

shopping-cart.hbs

```

<!--The Ticket wallet/ Shopping cart page that is for normal customers who ar enot
Govt Employeess-->

{{# if products }}
    <div class="row">
        <div class="col-sm-6 col-md-6 col-md-offset-3 col-sm-offset-3">
            <ul class="list-group">
                {{# each products }}
                    <li class="list-group-item">
                        <h4><strong>{{ this.item.title }}</strong>
                        <label style="color: darkblue">Rs{{ this.price }}</label>
                        <a href="/remove/{{this.item._id}}" class="btn btn-danger
pull-right">Remove All</a>
                        <label class="pull-right" style="color: white">ce
</label>
                        <a href="/reduce/{{this.item._id}}" class="btn btn-
primary pull-right">Reduce by 1</a>
                        <label class="pull-right" style="color: white">ce
</label>
                        <span class="badge pull-right"><h6>Qty {{ this.qty
}}</h6></span>
                    </li>
                {{/each}}
            </ul>
        </div>
    </div>
    <div class="row">
        <div class="col-sm-6 col-md-6 col-md-offset-3 col-sm-offset-3">
            <strong><h3 class="text-center">Total: Rs {{ totalPrice
}}</h3></strong>
        </div>
    </div>
    <hr>
    <div class="row">
        <div class="col-sm-6 col-md-6 col-md-offset-3 col-sm-offset-3">
            <button type="button" class="btn btn-primary btn-block"
data-toggle="modal" data-target="#myModal">
                Checkout
            </button>
            <button type="button" class="btn btn-success btn-block"
data-toggle="modal" data-target="#myModal2">
                Get Govt Employee Discount
            </button>
        </div>
    </div>

```

```
</div>

<div class="modal" id="myModal">
  <div class="modal-dialog">
    <div class="modal-content">

      <!-- Modal Header -->
      <div class="modal-header">
        <h4 class="modal-title text-center text-primary">Select
Payment Method</h4>
        <button type="button" class="close" data-
dismiss="modal">&times;</button>
      </div>

      <!-- Modal body -->
      <div class="modal-body">
        <table>
          <tr>
            <td>
              <div class="col-sm-4 col-md-4 col-md-offset-4
col-sm-offset-4">
                <a href="/sampathCheckout" type="button"
class="btn btn-warning"></a>
              </div>
            </td>
            <td>
              <div class="col-sm-4 col-md-4 col-md-offset-4
col-sm-offset-4">
                <a href="/dialogCheckout" type="button"
class="btn btn-danger">
      <div class="col-sm-6 col-md-6 col-md-offset-3 col-sm-offset-3">
        <h1 class="text-center"><i class="fas fa-wallet"></i></h1>
        <h2 class="text-center">Wallet Empty!</h2>
      </div>
    </div>
  {{/if}}

```

- Backend

Passport.js

```

/*Passport is authentication middleware for Node.js*/
/*Here passport is used for authenticating the user login and signup*/

var passport = require('passport');
var User = require('../models/user');
var LocalStrategy = require('passport-local').Strategy;

passport.serializeUser(function (user, done) {
  done(null, user.id);
});

passport.deserializeUser(function (id, done) {
  User.findById(id, function (err, user) {
    done(err, user);
  });
});

/*validation and authentication of User signup*/

passport.use('local.signup', new LocalStrategy({
  usernameField: 'email',
  passwordField: 'password',
  passReqToCallback: true
}, function (req, email, password, done) {
  req.checkBody('email', 'Invalid email').notEmpty().isEmail();
  req.checkBody('password', 'Invalid password').notEmpty().isLength({min:4});
  var errors = req.validationErrors();
  if (errors) {
    var messages = [];
    errors.forEach(function(error) {
      messages.push(error.msg);
    });
    return done(null, false, req.flash('error', messages));
  }
  User.findOne({'email': email}, function (err, user) {
    if (err) {
      return done(err);
    }
    if (user) {
      return done(null, false, {message: 'Email is already in use.'});
    }
    var newUser = new User();
    newUser.email = email;
    newUser.password = newUser.encryptPassword(password);
    newUser.save(function(err, result) {

```

```

        if (err) {
            return done(err);
        }
        return done(null, newUser);
    });
});
});

/*validation and authentication of User signup*/
passport.use('local.signin', new LocalStrategy({
    usernameField: 'email',
    passwordField: 'password',
    passReqToCallback: true
}, function(req, email, password, done) {
    req.checkBody('email', 'Invalid email').notEmpty().isEmail();
    req.checkBody('password', 'Invalid password').notEmpty();
    var errors = req.validationErrors();
    if (errors) {
        var messages = [];
        errors.forEach(function(error) {
            messages.push(error.msg);
        });

        /*Redirects are often combined with flash messages in order to display
        status information to the user.*/
        return done(null, false, req.flash('error', messages));
    }

    /*Error handling in the signup process*/
    User.findOne({'email': email}, function (err, user) {
        if (err) {
            return done(err);
        }
        if (!user) {
            return done(null, false, {message: 'No user found.'});
        }
        if (!user.validPassword(password)) {
            return done(null, false, {message: 'Wrong password.'});
        }
        return done(null, user);
    });
});
});

```

cart.js

```

/*The cart contains the selected tickets,
here the processes of adding tickets to cart,
reducing number of tickets, removing all tickets like functions are implemented*/

module.exports = function Cart(oldCart) {
    this.items = oldCart.items || {};
    this.totalQty = oldCart.totalQty || 0;
    this.totalPrice = oldCart.totalPrice || 0;

    /*Add Items to cart*/
    this.add = function(item, id) {
        var storedItem = this.items[id];
        if (!storedItem) {
            storedItem = this.items[id] = {item: item, qty: 0, price: 0};
        }
        storedItem.qty++;
        storedItem.price = storedItem.item.price * storedItem.qty;
    };
};

```

```

        this.totalQty++;
        this.totalPrice += storedItem.item.price;
    });
    /*Reduce items from cart one at a time at the click of a button*/
    this.reduceByOne = function(id) {
        this.items[id].qty--;
        this.items[id].price -= this.items[id].item.price;
        this.totalQty--;
        this.totalPrice -= this.items[id].item.price;

        if (this.items[id].qty <= 0) {
            delete this.items[id];
        }
    };

    /*Remove the ticket altogether from the cart at once*/
    this.removeItem = function(id) {
        this.totalQty -= this.items[id].qty;
        this.totalPrice -= this.items[id].price;
        delete this.items[id];
    };

    this.generateArray = function() {
        var arr = [];
        for (var id in this.items) {
            arr.push(this.items[id]);
        }
        return arr;
    };
};

```

Order.js

```

/*Mongoose is an Object Data Modeling library for MongoDB and Node.js*/

var mongoose = require('mongoose');
var Schema = mongoose.Schema;

/*Setting up the schema for the order details to be saved in the db*/

var schema = new Schema({
    user: {type: Schema.Types.ObjectId, ref: 'User'},
    cart: {type: Object, required: true},
    address: {type: String, required: true},
    name: {type: String, required: true},
    paymentId: {type: String, required: true}
});

module.exports = mongoose.model('Order', schema);

```

Product.js

```

/*Mongoose is an Object Data Modeling library for MongoDB and Node.js*/

var mongoose = require('mongoose');
var Schema = mongoose.Schema;

/*Setting up the schema for the ticket details to be saved in the db*/

var schema = new Schema({
    imagePath: {type: String, required: true},
    title: {type: String, required: true},
    description: {type: String, required: true},

```

```

    price: {type: Number, required: true}
  });

module.exports = mongoose.model('Product', schema);

```

Model/User.js

```

/*Mongoose is an Object Data Modeling library for MongoDB and Node.js*/
/*Bcrypt is a secured way to store passwords.*/

var mongoose = require('mongoose');
var Schema = mongoose.Schema;
var bcrypt = require('bcrypt-nodejs');

/*Setting up the schema for the User details to be saved in the db*/

var userSchema = new Schema({
  email: {type: String, required: true},
  password: {type: String, required: true}
});

/*Using bcrypt for password encryption and storing in db for more security*/

userSchema.methods.encryptPassword = function(password) {
  return bcrypt.hashSync(password, bcrypt.genSaltSync(5), null);
};

userSchema.methods.validPassword = function(password) {
  return bcrypt.compareSync(password, this.password);
};

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

```

Checkout.js

```

/*Stripe is a third party resource which eliminate needless complexity and details
 * when conducting safer online payments*/

Stripe.setPublishableKey('pk_test_MZCsVXhAX2ujPC8ih9Vaqvrj00t6wiHd6d'); //My
personal Stripe API key for testing purposes

var $form = $('#checkout-form');

/*the jquery that gets the inputs from the submitted form and validates it with the
use of STRIPE */
$form.submit(function (event) {
  $('#charge-error').addClass('hidden');
  $form.find('button').prop('disabled', true);
  Stripe.card.createToken({
    number: $('#card-number').val(),
    cvc: $('#card-cvc').val(),
    exp_month: $('#card-expiry-month').val(),
    exp_year: $('#card-expiry-year').val(),
    name: $('#card-name').val()
  }, stripeResponseHandler);
  return false;
});

function stripeResponseHandler(status, response) {
  if (response.error) { // Problem!

    // Show the errors on the form
    $('#charge-error').text(response.error.message);
    $('#charge-error').removeClass('hidden');
  }
}

```



```

    $form.find('button').prop('disabled', false); // Re-enable submission
  } else { // Token was created!

    // Get the token ID:
    var token = response.id;

    // Insert the token into the form so it gets submitted to the server:
    $form.append($('

```

Index.js

```

var express = require('express');
var router = express.Router();
var Cart = require('../models/cart');

var Product = require('../models/product');
var Order = require('../models/order');

/* GET home page. */
router.get('/', function (req, res, next) {
  var successMsg = req.flash('success')[0];
  Product.find(function (err, docs) {
    var productChunks = [];
    var chunkSize = 3;
    for (var i = 0; i < docs.length; i += chunkSize) {
      productChunks.push(docs.slice(i, i + chunkSize));
    }
    res.render('shop/index', {title: 'Shopping Cart', products: productChunks,
      successMsg: successMsg, noMessages: !successMsg});
  });
});

//The function adds tickets to the cart by identifying using the product/ticket ID
according to the GET request

router.get('/add-to-cart/:id', function (req, res, next) {
  var productId = req.params.id;
  var cart = new Cart(req.session.cart ? req.session.cart : {});

  Product.findById(productId, function (err, product) {
    if (err) {
      return res.redirect('/');
    }
    cart.add(product, product.id);
    req.session.cart = cart;
    console.log(req.session.cart);
    res.redirect('/');
  });
});

//The function reduce a certain type of ticket one by one from the cart by
identifying using the product/ticket ID according to the GET request
router.get('/reduce/:id', function (req, res, next) {
  var productId = req.params.id;
  var cart = new Cart(req.session.cart ? req.session.cart : {});

  cart.reduceByOne(productId);
  req.session.cart = cart;

```

```

    res.redirect('/shopping-cart');
});

//The function removes that certain ticket type all at once from the cart by
identifying using the product/ticket ID according to the GET request
router.get('/remove/:id', function(req, res, next) {
    var productId = req.params.id;
    var cart = new Cart(req.session.cart ? req.session.cart : {});

    cart.removeItem(productId);
    req.session.cart = cart;
    res.redirect('/shopping-cart');
});

//The function generates the ticket wallet or more specifically the shopping cart
by listing down the chosen tickets
router.get('/shopping-cart', function(req, res, next) {
    if (!req.session.cart) {
        return res.render('shop/shopping-cart', {products: null});
    }
    var cart = new Cart(req.session.cart);
    res.render('shop/shopping-cart', {products: cart.generateArray(), totalPrice:
cart.totalPrice});
});

//The function generates the ticket wallet which gives a discount of 10% for
government employees by listing down the chosen tickets
router.get('/discountShoppingCart', function(req, res, next) {
    if (!req.session.cart) {
        return res.render('shop/discountShoppingCart', {products: null});
    }
    var cart = new Cart(req.session.cart);
    res.render('shop/discountShoppingCart', {products:
cart.generateArray(), discount: cart.totalPrice * 0.1, discountedPrice:
cart.totalPrice * 0.9, totalPrice: cart.totalPrice});
});

//The function that generates the Sampath payment portal without the govt employee
discount
router.get('/sampathCheckout', isLoggedIn, function(req, res, next) {
    if (!req.session.cart) {
        return res.redirect('/shopping-cart');
    }
    var cart = new Cart(req.session.cart);
    var errMsg = req.flash('error')[0];
    res.render('shop/sampathCheckout', {total: cart.totalPrice, errMsg: errMsg,
noError: !errMsg});
});

//The function that generates the Sampath payment portal with the govt employee
discount
router.get('/discountCheckout', isLoggedIn, function(req, res, next) {
    if (!req.session.cart) {
        return res.redirect('/discountShoppingCart');
    }
    var cart = new Cart(req.session.cart);
    var errMsg = req.flash('error')[0];
    res.render('shop/discountCheckout', {grandTotal: cart.totalPrice * 0.9, errMsg:
errMsg, noError: !errMsg});
});

//The function that generates the Dialog payment portal without the govt employee
discount
router.get('/dialogCheckout', isLoggedIn, function(req, res, next) {
    if (!req.session.cart) {
        return res.redirect('/shopping-cart');
    }

```

```

    var cart = new Cart(req.session.cart);
    var errMsg = req.flash('error')[0];
    res.render('shop/dialogCheckout', {total: cart.totalPrice, errMsg: errMsg,
noError: !errMsg});
});

//The function that generates the Dialog payment portal with the govt employee
discount
router.get('/discountDialogCheckout', isLoggedIn, function(req, res, next) {
    if (!req.session.cart) {
        return res.redirect('/discountShoppingCart');
    }
    var cart = new Cart(req.session.cart);
    var errMsg = req.flash('error')[0];
    res.render('shop/discountDialogCheckout', {grandTotal: cart.totalPrice * 0.9,
errMsg: errMsg, noError: !errMsg});
});

/*Stripe is a third party resource which eliminate needless complexity and details
* when conducting safer online payments*/

/*The function that gets the inputs from the submitted form and validates it with
the use of STRIPE */
router.post('/checkout', isLoggedIn, function(req, res, next) {
    if (!req.session.cart) {
        return res.redirect('/shopping-cart');
    }
    var cart = new Cart(req.session.cart);

    var stripe = require("stripe")(
        "sk_test_slbTl0wyvclcnwFIXcr8kjRH00pO3xUgeF" //My personal Stripe SECRET
API key for testing purposes
    );

    stripe.charges.create({
        amount: cart.totalPrice,
        currency: "usd",
        source: req.body.stripeToken, // obtained with Stripe.js
        description: "Test Charge"
    }, function(err, charge) {
        if (err) {
            req.flash('error', err.message);
            return res.redirect('/sampathCheckout');
        }
        var order = new Order({
            user: req.user,
            cart: cart,
            address: req.body.address,
            name: req.body.name,
            paymentId: charge.id
        });
        order.save(function(err, result) {

            /* Nodemailer is a module for Node.js applications to allow email
sending.
            Here it is used to send confirmation email to customer confirming the
payment
            As it uses SMTP service the emails are not actually delivered and are
captured halfway

            They can be viewed from my ethereal.email account
            Preview URL: https://ethereal.email/message
            Username: mustafa66@ethereal.email
            Password: nHTk7j1EJDn2Eunn */

            const nodemailer = require("nodemailer");

```

```

    async function main() {

        // Generate test SMTP service account from ethereal.email
        let testAccount = await nodemailer.createTestAccount();

        // create reusable transporter object using the default SMTP
transport
        const transporter = nodemailer.createTransport({
            host: 'smtp.ethereal.email',
            port: 587,
            auth: {
                user: 'mustafa66@ethereal.email',
                pass: 'nHTk7jJlEJDJn2Eunn'
            }
        });

        // send mail with defined transport object
        let info = await transporter.sendMail({
            from: '"Sampath Payment Gateway - Train Ticket"
<reservetraiticket@sampath.lk>', // sender address
            to: "thilini1997@gmail.com, pop3.ethereal.email", // list of
receivers
            subject: "Payment Confirmed", // Subject line
            text: "Test mail", // plain text body
            html: "<b>Payment Confirmed!!!</b>The tickets have been
reserved successfully, the amount has been credited to your account"
        });

        console.log("Message sent: %s", info.messageId);
        // Message sent:

        // Preview only available when sending through an Ethereal account
        console.log("Preview URL: %s", nodemailer.getTestMessageUrl(info));
        // Preview URL: https://ethereal.email/message
        // Username: mustafa66@ethereal.email
        // Password: nHTk7jJlEJDn2Eunn
    }
    main().catch(console.error);

    req.flash('success', 'Payment Successful!!! A confirmation email will
be sent to your registered email address shortly... ');
    req.session.cart = null;
    res.redirect('/');
  });
});
});

//The function that generates the SMS fro dialog payment portal using thrid part
resource NEXMO
//NEXMO allows you to Send and receive messages on WhatsApp, Facebook Messenger,
Viber, MMS, and SMS -- all with the Messages API.
// Create your own failover messaging flow to ensure delivery with the Dispatch API

router.post('/sms', isLoggedIn, function(req, res, next) {
  if (!req.session.cart) {
    return res.redirect('/shopping-cart');
  }

  const Nexmo = require('nexmo');
  const nexmo = new Nexmo({
    apiKey: '13578082', // My personal NEXMO account API KEY
    apiSecret: 'bLfyl2EopI9lTwUs' // My personal NEXMO account SECRET API KEY
  }, { debug: true });

  nexmo.message.sendSms(
    'YOUR_VIRTUAL_NUMBER', '94766446337', 'Ticket Reservation Payment
Confirmed!',

```

```

        (err, responseData) => {
            if (err) {
                console.log(err);
            } else {
                console.dir(responseData);
            }
        }
    ); //This Msg is actually sent in real life as the number is verified for
testing purposes

    req.flash('success', 'Payment Successful!!! A confirmation SMS will be sent to
your mobile shortly... ');
    req.session.cart = null;
    res.redirect('/');
});

module.exports = router;

//Checks whether the user is logged in before making any purchases
function isLoggedIn(req, res, next) {
    if (req.isAuthenticated()) {
        return next();
    }
    req.session.oldUrl = req.url;
    res.redirect('/user/signin');
}

```

route/user.js

```

var express = require('express');
var router = express.Router();
var csrf = require('csrf');
var passport = require('passport');

var Order = require('../models/order');
var Cart = require('../models/cart');

var csrfProtection = csrf(); //Protect app from Cross-Site Request Forgery
router.use(csrfProtection);

//Handles the user profile by listing down all the purchases upto date
router.get('/profile', isLoggedIn, function (req, res, next) {
    Order.find({user: req.user}, function(err, orders) {
        if (err) {
            return res.write('Error!');
        }
        var cart;
        orders.forEach(function(order) {
            cart = new Cart(order.cart);
            order.items = cart.generateArray();
        });
        res.render('user/profile', { orders: orders });
    });
});

router.get('/logout', isLoggedIn, function (req, res, next) {
    req.logout();
    res.redirect('/');
});

router.use('/', notLoggedIn, function (req, res, next) {
    next();
});

```

```

router.get('/signup', function (req, res, next) {
  var messages = req.flash('error');
  res.render('user/signup', {csrfToken: req.csrfToken(), messages: messages,
hasErrors: messages.length > 0});
});

router.post('/signup', passport.authenticate('local.signup', {
  failureRedirect: '/user/signup',
  failureFlash: true
}), function (req, res, next) {
  if (req.session.oldUrl) {
    var oldUrl = req.session.oldUrl;
    req.session.oldUrl = null;
    res.redirect(oldUrl);
  } else {
    res.redirect('/user/profile');
  }
});

router.get('/signin', function (req, res, next) {
  var messages = req.flash('error');
  res.render('user/signin', {csrfToken: req.csrfToken(), messages: messages,
hasErrors: messages.length > 0});
});

router.post('/signin', passport.authenticate('local.signin', {
  failureRedirect: '/user/signin',
  failureFlash: true
}), function (req, res, next) {
  if (req.session.oldUrl) {
    var oldUrl = req.session.oldUrl;
    req.session.oldUrl = null;
    res.redirect(oldUrl);
  } else {
    res.redirect('/user/profile');
  }
});

module.exports = router;

function isLoggedIn(req, res, next) {
  if (req.isAuthenticated()) {
    return next();
  }
  res.redirect('/');
}

function notLoggedIn(req, res, next) {
  if (!req.isAuthenticated()) {
    return next();
  }
  res.redirect('/');
}

```

product-seeder.js

```

/*Here the tickets are seeded in to the database using mongoose*/

var Product = require('../models/product');

var mongoose = require('mongoose');

mongoose.connect('localhost:27017/shop'); //db connection

//Ticket details storing in db

```

```

var products = [
  new Product({
    imagePath: 'http://www.bestoflanka.com/images/dont-miss-when-in-sri-
lanka/visit-to-kandy-sacred-city/08.jpg',
    title: 'Colombo-Kandy',
    description: '1030 - Intercity Express',
    price: 700
  }),
  new Product({
    imagePath:
'https://www.asianmirror.lk/media/k2/items/cache/610ffdc3f8bd82cb781ff35b8cc7fce3_X
L.jpg',
    title: 'Colombo-Badulla',
    description: '1015 - Udarata Menike',
    price: 'Rs.850'
  }),
  new Product({
    imagePath:
'http://srilankanexpeditions.com/location_img/1455075583khguyf.jpg',
    title: 'Colombo-Batticalo',
    description: '6011 - Udaya Devi',
    price: 800
  }),
  new Product({
    imagePath: 'https://worldtravelfamily.com/wp-content/uploads/2016/05/Lets-
go-to-Jaffna.jpg',
    title: 'Colombo-Jaffna',
    description: '4077 - Yaldevi',
    price: 950
  }),
  new Product({
    imagePath: 'https://t-
ec.bstatic.com/images/hotel/max1280x900/591/59178304.jpg',
    title: 'Colombo-Matara',
    description: '50 - 8003 - Express',
    price: 650
  }),
  new Product({
    imagePath: 'https://media.holidayme.com/wp-
content/uploads/2016/03/23153835/Koneswaram-Temple-also-use-as-banner.jpg',
    title: 'Colombo-Trinco',
    description: '4077 - Yaldevi',
    price: 850
  }),
  new Product({
    imagePath:
'https://cdn2.veltra.com/ptr/20180608033151_1411982644_13948_0.jpg?imwidth=550&impo
licy=custom',
    title: 'Colombo-Anuradapura',
    description: '4018 - Uththara Devi',
    price: 650
  }),
  new Product({
    imagePath: 'https://photos.wikimapia.org/p/00/03/22/09/60_big.jpg',
    title: 'Colombo- Kankesanthura',
    description: '4021 - AC Intercity',
    price: 1500
  })
];

var done = 0;
for (var i = 0; i < products.length; i++) {
  products[i].save(function(err, result) {
    done++;
    if (done === products.length) {
      exit();
    }
  });
};

```

```

}

function exit() {
  mongoose.disconnect(); //disconnecting db
}

```

App.js

```

var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');
var expressHbs = require('express-handlebars');
var mongoose = require('mongoose');
var session = require('express-session');
var passport = require('passport');
var flash = require('connect-flash');
var validator = require('express-validator');
var MongoStore = require('connect-mongo')(session);

var routes = require('./routes/index'); //setting routes
var userRoutes = require('./routes/user');

var app = express();

mongoose.connect('localhost:27017/shop'); //db connection
require('./config/passport');

// view engine setup
app.engine('.hbs', expressHbs({defaultLayout: 'layout', extname: '.hbs'}));
app.set('view engine', '.hbs');

app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(validator());
app.use(cookieParser());
app.use(session({
  secret: 'mysupersecret',
  resave: false,
  saveUninitialized: false,
  store: new MongoStore({ mongooseConnection: mongoose.connection }),
  cookie: { maxAge: 180 * 60 * 1000 }
}));
app.use(flash()); //flash message
app.use(passport.initialize()); //Passport Authentication
app.use(passport.session());
app.use(express.static(path.join(__dirname, 'public')));

app.use(function(req, res, next) {
  res.locals.login = req.isAuthenticated();
  res.locals.session = req.session;
  next();
});

app.use('/user', userRoutes);
app.use('/', routes);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  var err = new Error('Not Found');
  err.status = 404;
  next(err);
}

```



```
});  
  
// error handlers  
  
// development error handler  
// will print stacktrace  
if (app.get('env') === 'development') {  
  app.use(function(err, req, res, next) {  
    res.status(err.status || 500);  
    res.render('error', {  
      message: err.message,  
      error: err  
    });  
  });  
}  
  
// production error handler  
// no stacktraces leaked to user  
app.use(function(err, req, res, next) {  
  res.status(err.status || 500);  
  res.render('error', {  
    message: err.message,  
    error: {}  
  });  
});  
  
module.exports = app;
```

End of Report
