Sri Lanka Institute of Information Technology

Distributed Systems (SE3020)

Assignment 2
Rest API – Loops' Train Reservation

2019

Name	**N.Shriram**
ID Number	IT 17 7064 38
Specialization	Software Engineering
Year/Semester	3rd Year 1st Semester

14TH JANUARY 2019

Table of Contents

1. Introduction

This Loop's Train Reservation is an online train reservation system that is developed by integrating several independent service together to form a well build.

The List of key Services that made-up this System are as follows

- Authentication Service
- Government ID verification Service
- Payment Gateway Service
- Email Service
- Mobile OTP Service

The Technologies used for build up this system are,

- Front End Libraries and dependency modules
  - React Js
  - Reactstrap
  - React-icon
  - Axios

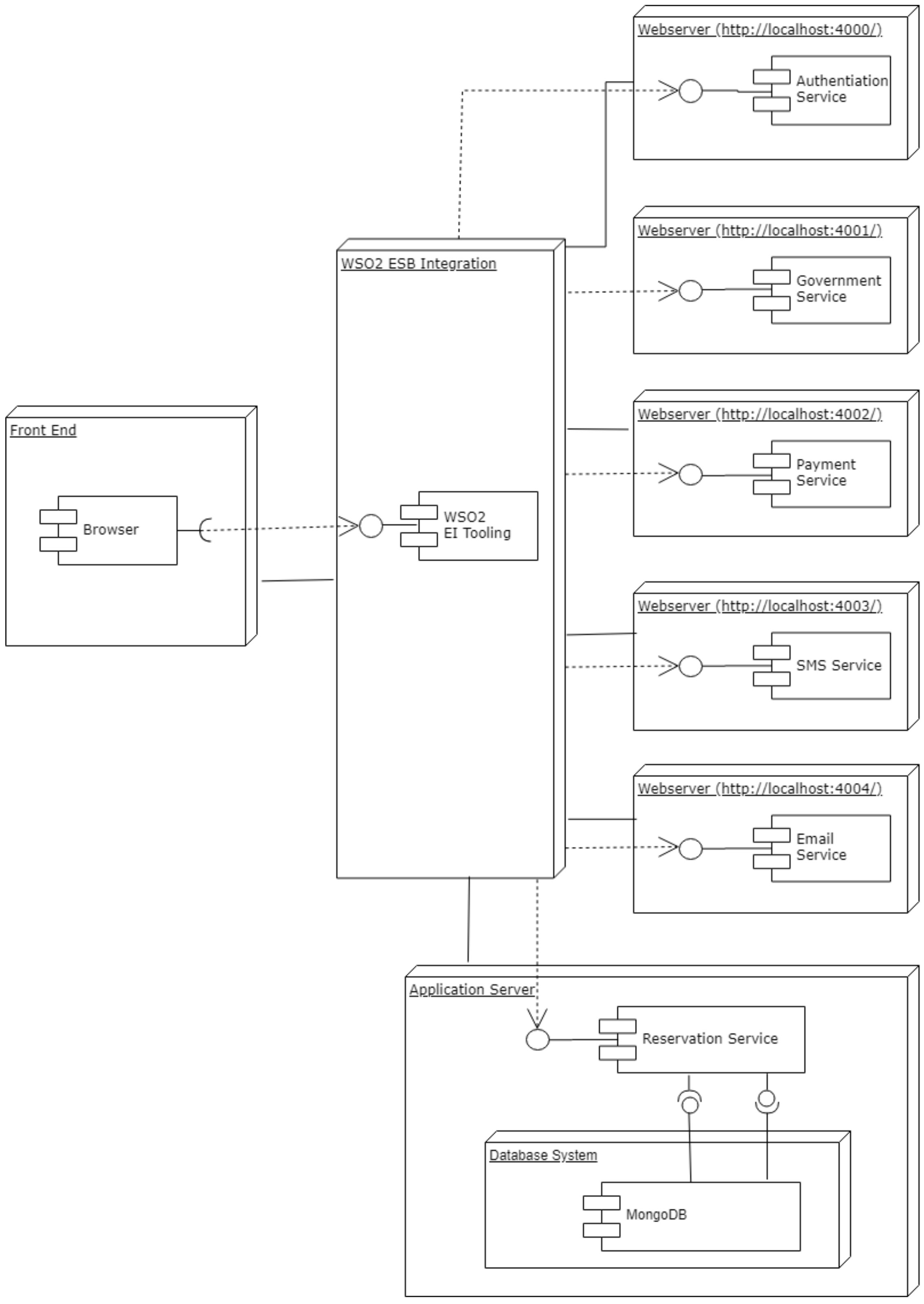
- Back End Libraries and dependency modules
  - Node JS
  - Express Js
  - Axios
  - Body-Parser
  - Cors
  - Nodemailer
  - Speakeasy
  - Ejs template engine
  - Paypal-rest-sdk
  - Twilio

2.0. High Level System Diagram

This System is implemented using Service Oriented Architecture where we use WSO2 ESN Integration to integrate those service together and render them to the user based on the request made. All the service functionalities are implemented as set of REST API calls.

Data between Front end and backend are passed using JSON data format, over the HTTP network communication Protocol. System uses JSON as the only data communication format since its light weight, and both front end and back end are implemented using JavaScript framework so it will be easy to manipulate and process data using JSON.

2.1. High Level System Architecture Design



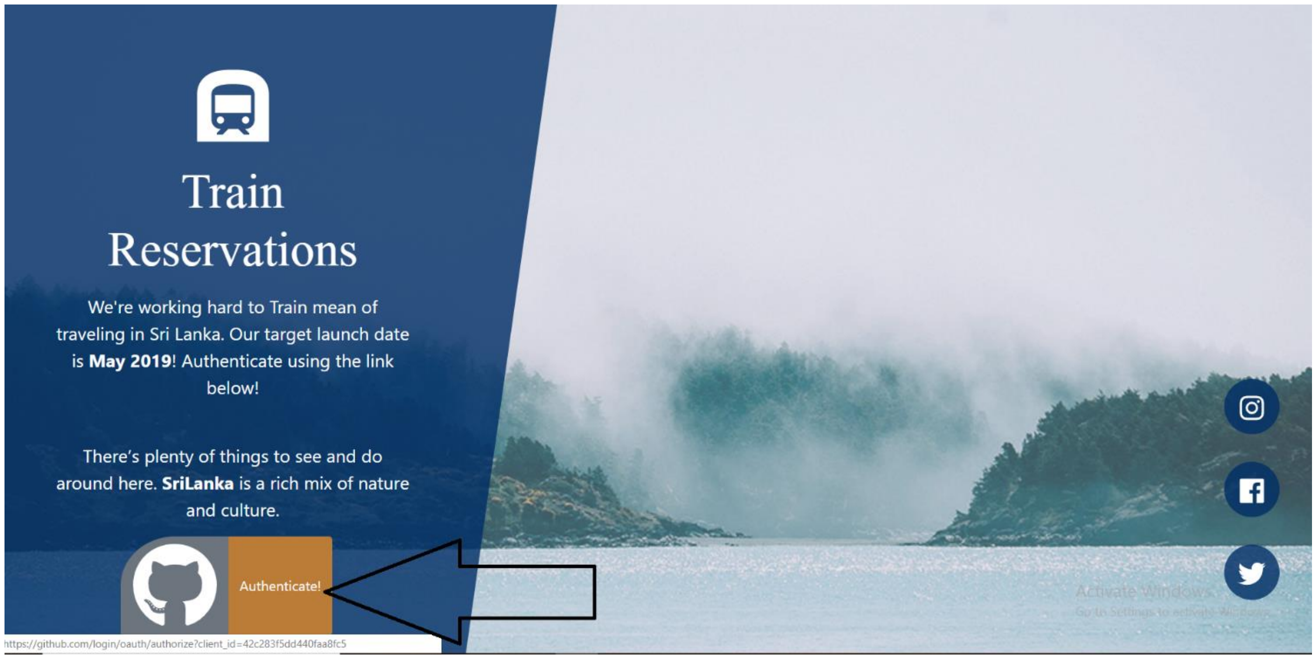
3. System Work Flow

Basic workflow of the System is as follows,

1. User Authenticate into the system using the Git OAuth which will provide a token to application that ensure user is the one who he purport to be
2. Application will redirect to Home page, and user selects Reservation button in order to carry on with Train reservation
3. Application displays an Reservation Form including the abstract information needed to book a train
4. User input relevant information according to the reservation requirement.
5. As the User select the Train service, Train class and number of seat, dynamically at real time application display the cost for the booking
6. User proceeds to Payment option
7. Application request to input NIC number to provide a 10% percent discount for Government employees
8. System verifies the NIC number and if valid dynamically provide discount and displays Net amount for the train booking.
9. User selects mode of payment for booking either add amount to dialog bill or pay by credit card through PayPal payment gateway
10. If user selects pay via Mobile, and OTP number will be send to user's contact number and user need to input with the OTP to confirm transaction
11. When mobile transaction successful, user would be redirected in to success page and a confirmation mail including booking details would be proceeded and sent to user
12. Else if user selects Credit card option user would be redirected in to PayPal payment gateway
13. User input appropriate Credit card details and proceed payment
14. Payment gateway verify the Credit card confidential and when it's valid user would be redirected in to success page and a confirmation mail including booking details will be sent to user

4. System Interface Implementation Work Flow

Fig 1: User access the URL and directed to Landing page, where they need to Authenticate



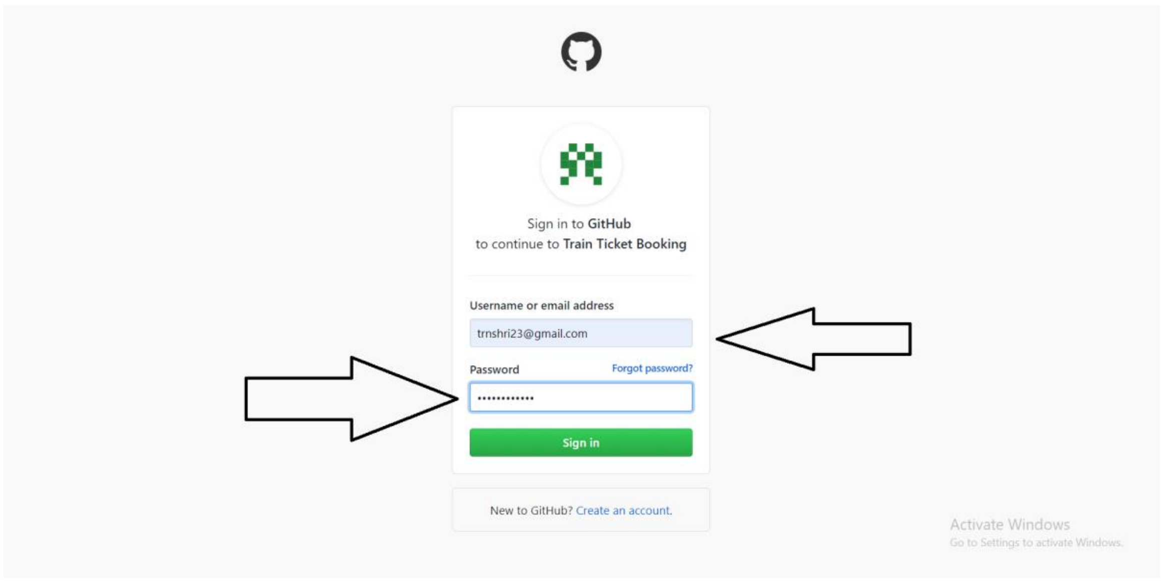
Train Reservations

We're working hard to Train mean of traveling in Sri Lanka. Our target launch date is **May 2019**! Authenticate using the link below!

There's plenty of things to see and do around here. **SriLanka** is a rich mix of nature and culture.

Authenticate!
https://github.com/login/oauth/authorize?client_id=42c283f5dd440faa8fc5
Activate Windows
Go to Settings to activate Windows.

Fig 2: User will be redirected to GitHub Authentication page which is linked with Loop's Reservation App



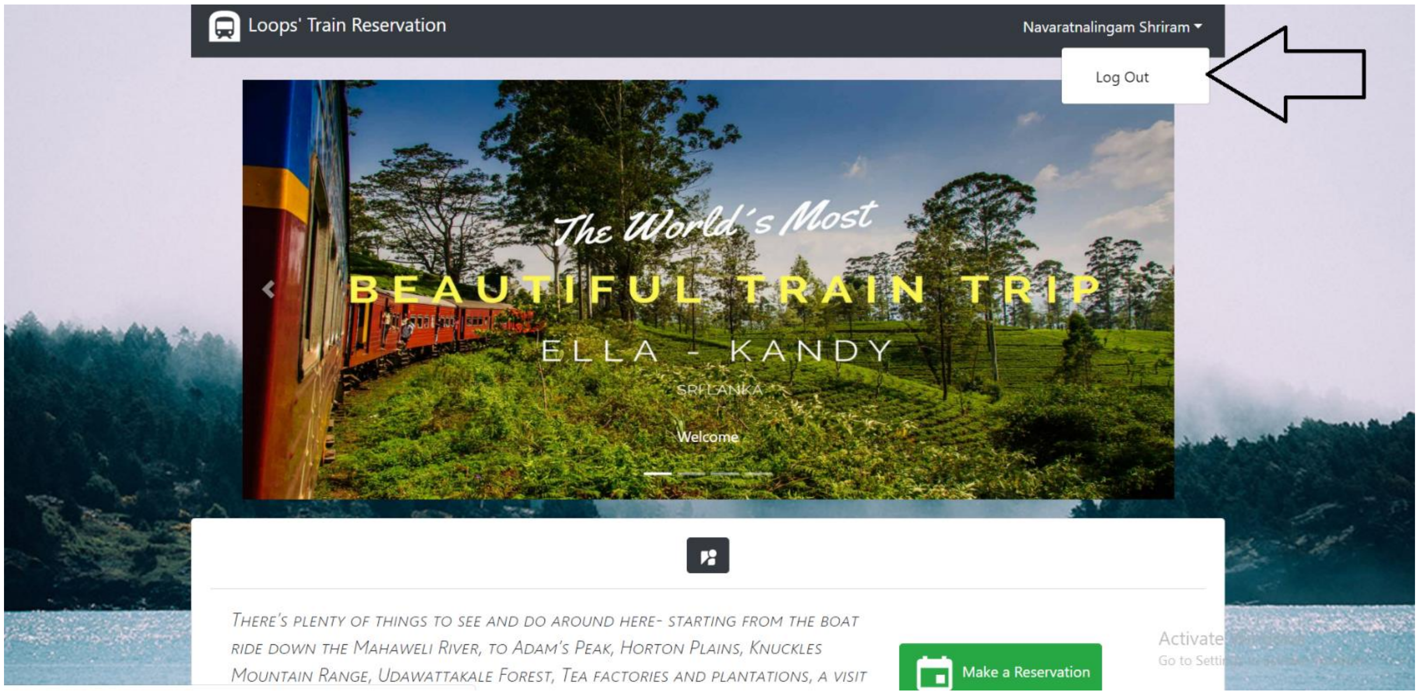
Sign in to GitHub
to continue to Train Ticket Booking

Username or email address: trnshri23@gmail.com

Password: ••••••••••••

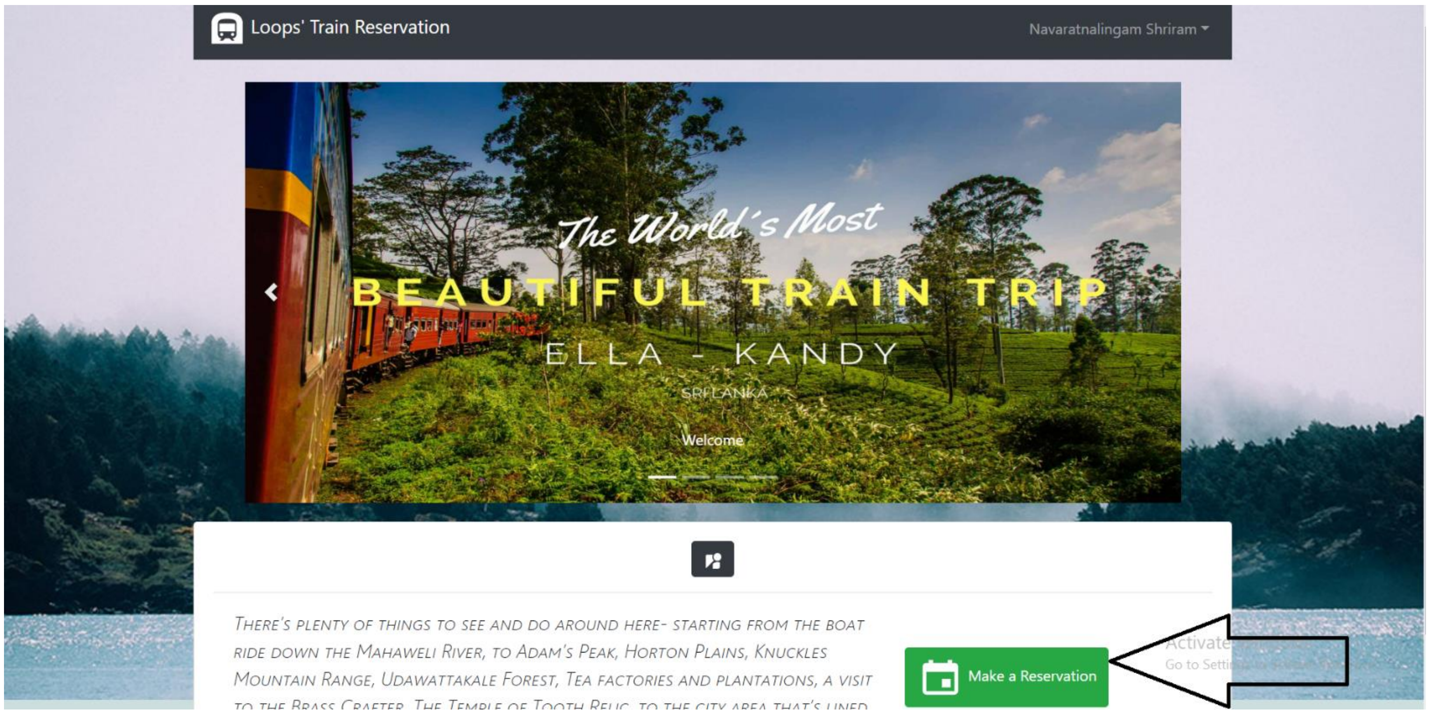
Forgot password?
Sign in
New to GitHub? Create an account.
Activate Windows
Go to Settings to activate Windows.

Fig 4: When Authentication successful, User will be directed to Home page where it indicate user logged in



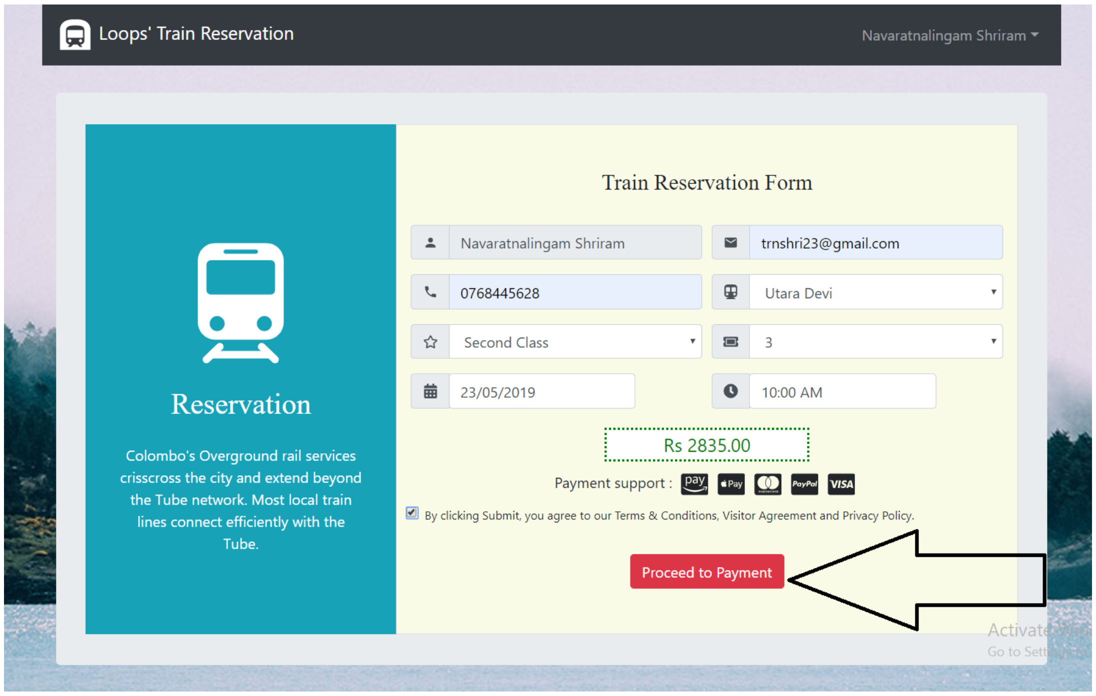
There's plenty of things to see and do around here- starting from the boat ride down the Mahaweli River, to Adam's Peak, Horton Plains, Knuckles Mountain Range, Udawattakale Forest, Tea factories and plantations, a visit

Fig 5: User selects "Make a Reservation" to proceed to Booking process



There's plenty of things to see and do around here- starting from the boat ride down the Mahaweli River, to Adam's Peak, Horton Plains, Knuckles Mountain Range, Udawattakale Forest, Tea factories and plantations, a visit to the Brass Crafter, The Temple of Tooth Relic, to the city area that's lined

Fig 6: User input appropriate reservation details and click “Proceed to payment”


Loops' Train Reservation

Navaratnalingam Shriram

Reservation

Colombo's Overground rail services crisscross the city and extend beyond the Tube network. Most local train lines connect efficiently with the Tube.

Train Reservation Form

Navaratnalingam Shriram	trnshri23@gmail.com
0768445628	Utara Devi
Second Class	3
23/05/2019	10:00 AM

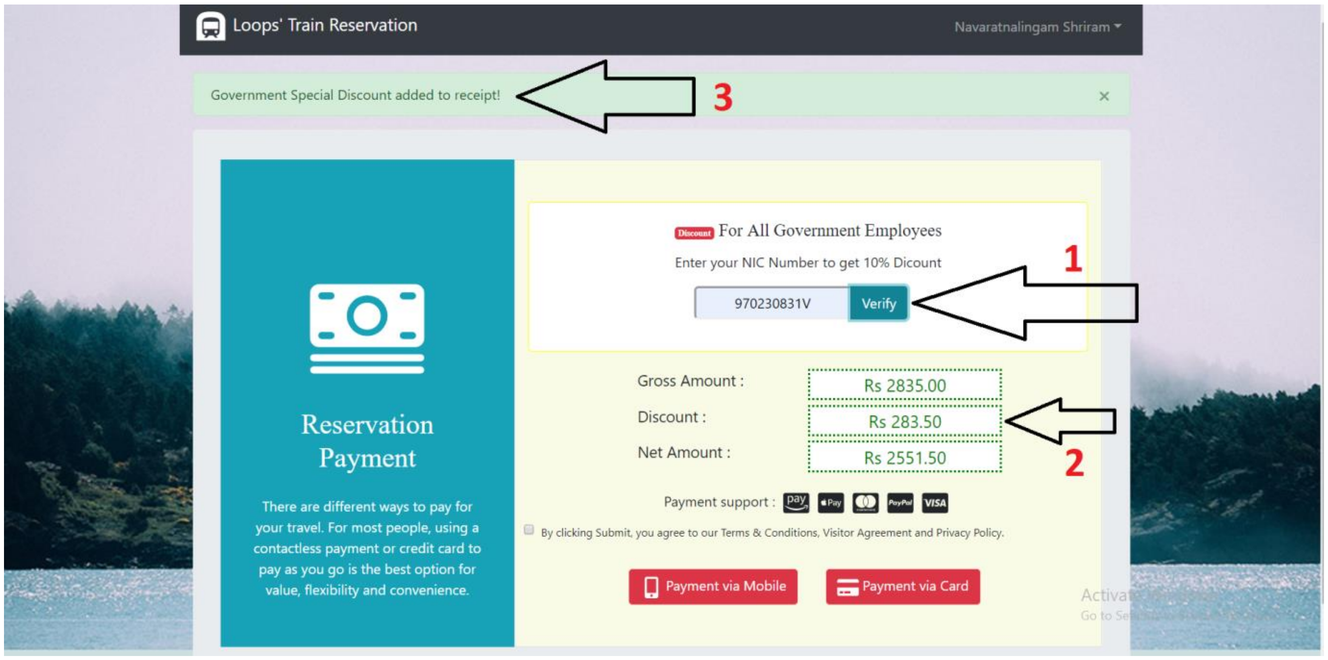
Rs 2835.00

Payment support :

☑ By clicking Submit, you agree to our Terms & Conditions, Visitor Agreement and Privacy Policy.

Proceed to Payment

Fig 7: User input correct NIC, and if valid dyanmically discount will be given to user


Loops' Train Reservation

Navaratnalingam Shriram

Government Special Discount added to receipt!

3

Reservation Payment

There are different ways to pay for your travel. For most people, using a contactless payment or credit card to pay as you go is the best option for value, flexibility and convenience.

Discount For All Government Employees

Enter your NIC Number to get 10% Dicount

970230831V Verify

1

Gross Amount :	Rs 2835.00
Discount :	Rs 283.50
Net Amount :	Rs 2551.50

2

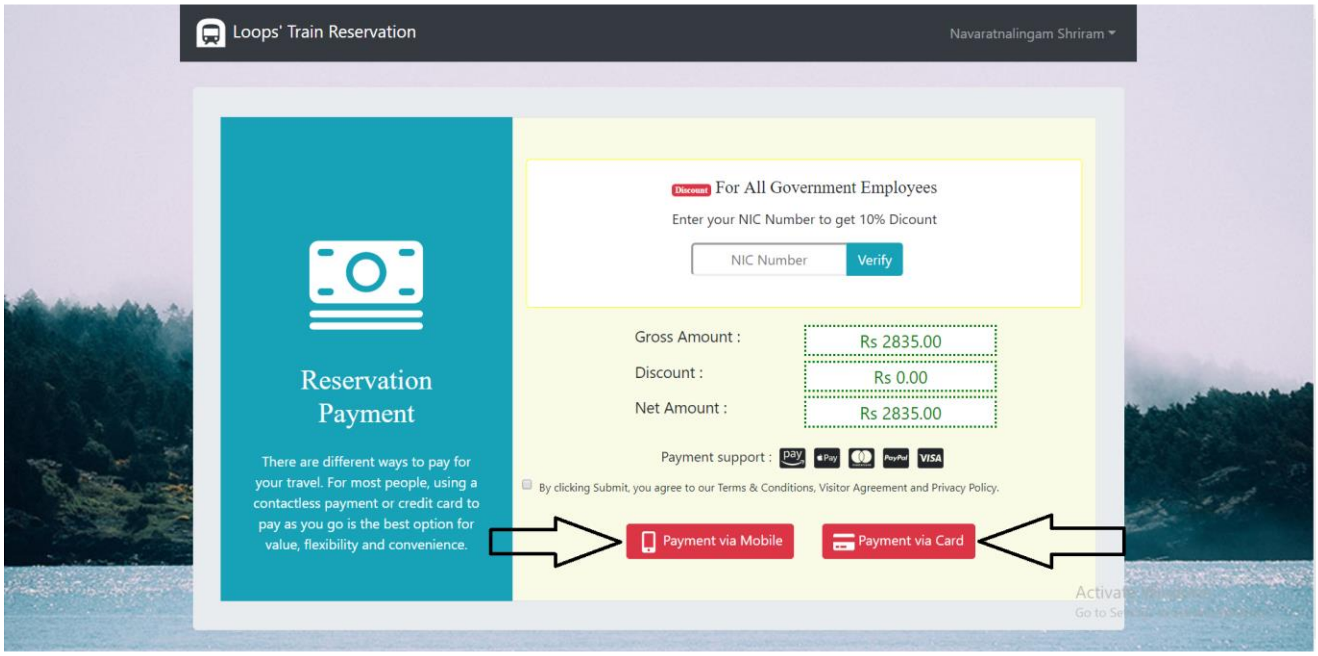
Payment support :

By clicking Submit, you agree to our Terms & Conditions, Visitor Agreement and Privacy Policy.

Payment via Mobile

Payment via Card

Fig 8: User can select “Payment via Card” to pay through payment Gateway


Loops' Train Reservation

Navaratnalingam Shriram

Reservation Payment

There are different ways to pay for your travel. For most people, using a contactless payment or credit card to pay as you go is the best option for value, flexibility and convenience.

Discount For All Government Employees

Enter your NIC Number to get 10% Dicount

NIC Number | Verify

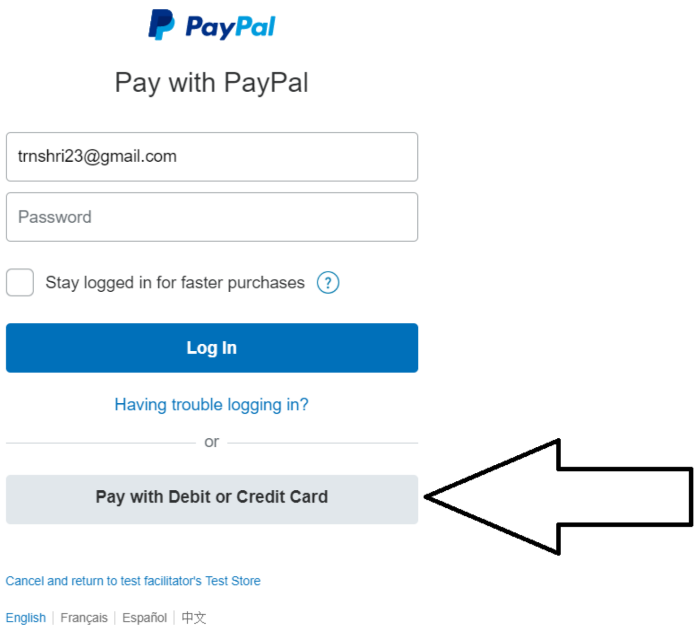
Gross Amount :	Rs 2835.00
Discount :	Rs 0.00
Net Amount :	Rs 2835.00

Payment support :

By clicking Submit, you agree to our Terms & Conditions, Visitor Agreement and Privacy Policy.

Payment via Mobile | Payment via Card

Fig 9: User directed to Payment gateway home page and user should select “Pay with Debit or Credit Card”


PayPal

Pay with PayPal

trnshri23@gmail.com

Password

Stay logged in for faster purchases

Log In

Having trouble logging in?

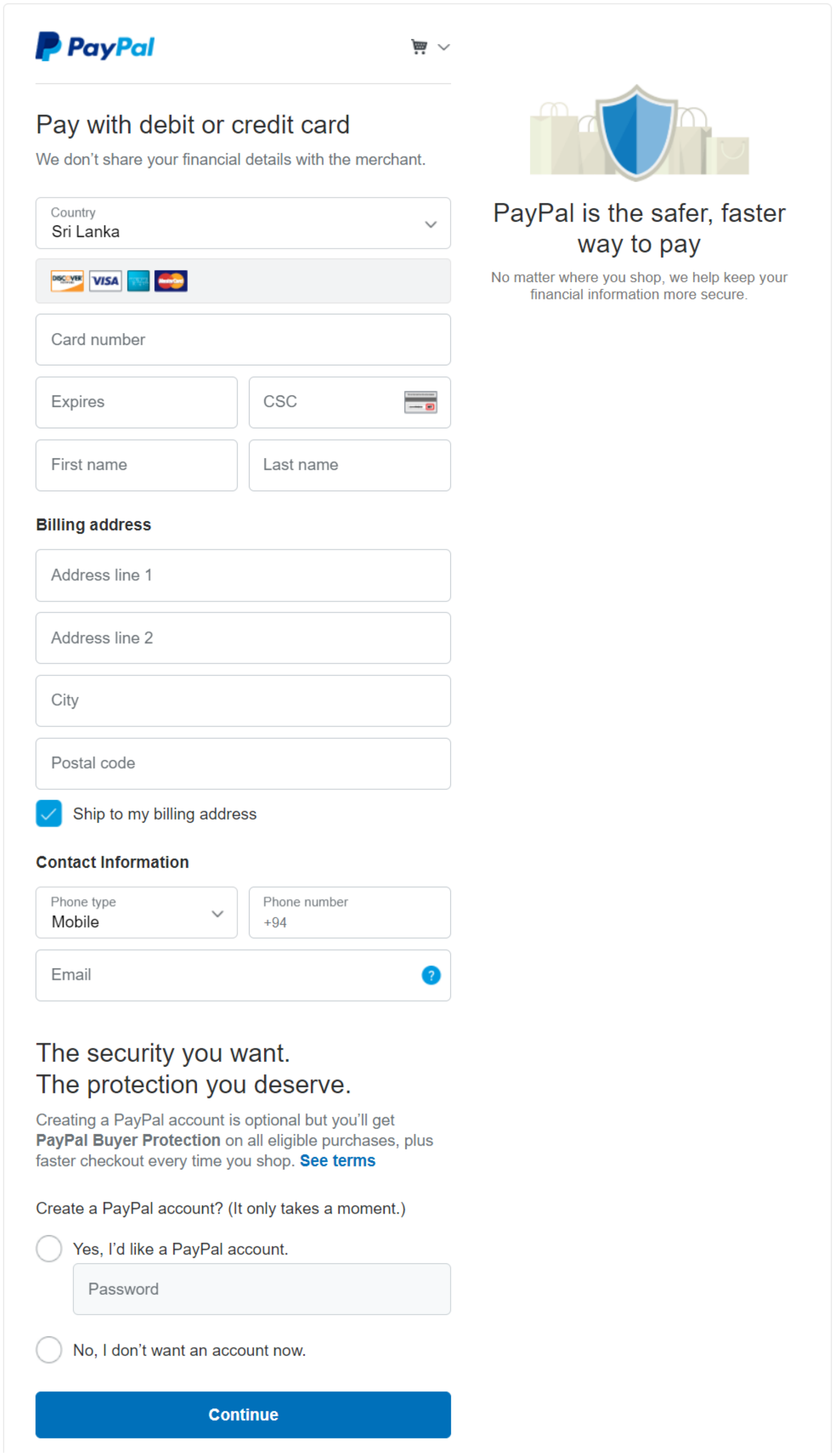
or

Pay with Debit or Credit Card

Cancel and return to test facilitator's Test Store

English | Français | Español | 中文

Fig 11: Sample Credit card details requesting form that will be loaded to user



Pay with debit or credit card

We don't share your financial details with the merchant.

Country: Sri Lanka

Card number

Expires

CSC

First name

Last name

Billing address

Address line 1

Address line 2

City

Postal code

[✓] Ship to my billing address

Contact Information

Phone type: Mobile

Phone number: +94

Email

The security you want.
The protection you deserve.

Creating a PayPal account is optional but you'll get **PayPal Buyer Protection** on all eligible purchases, plus faster checkout every time you shop. **See terms**

Create a PayPal account? (It only takes a moment.)

Yes, I'd like a PayPal account.

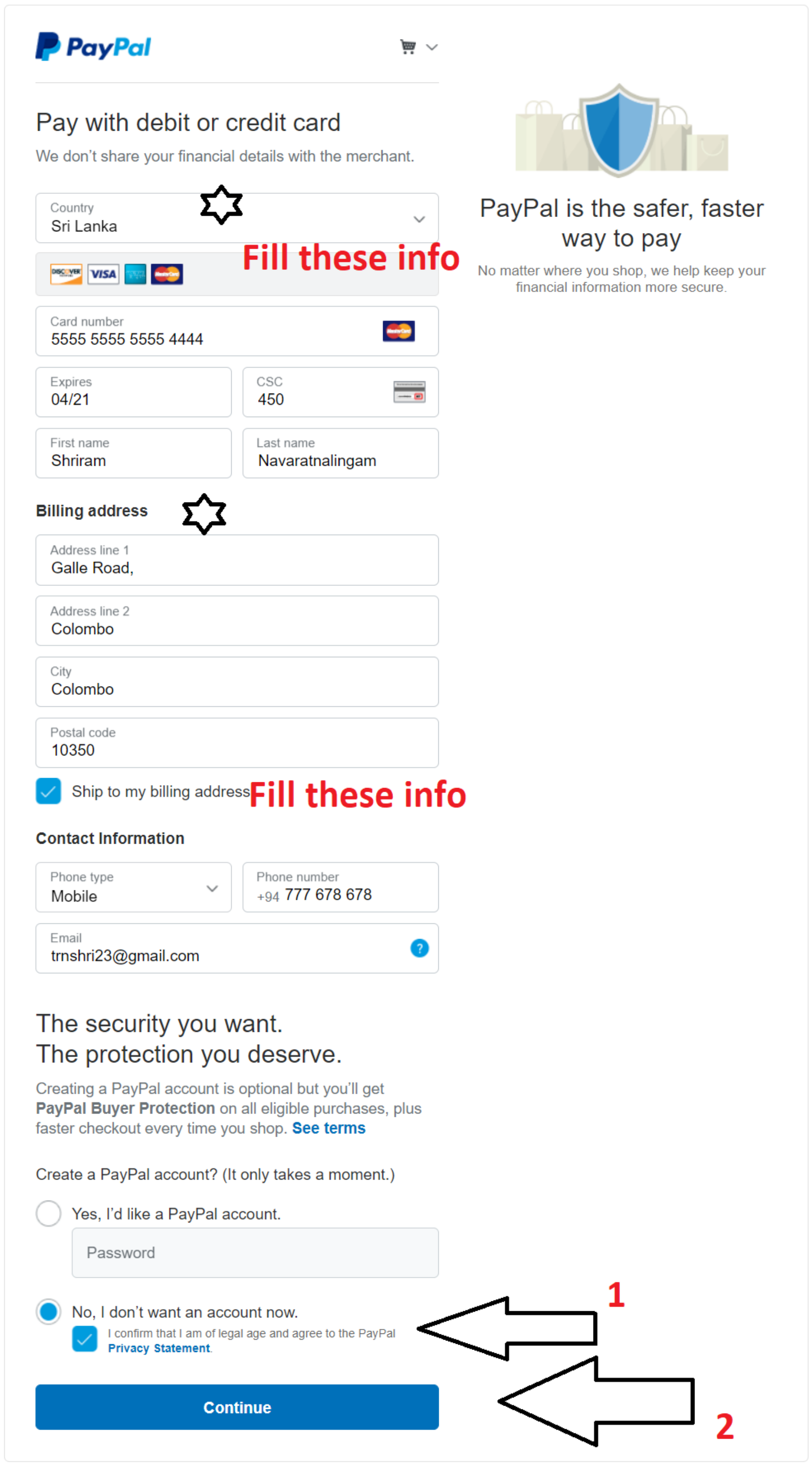
Password

No, I don't want an account now.

PayPal is the safer, faster way to pay

No matter where you shop, we help keep your financial information more secure.

Fig 12: User need to fill appropriate input as specified in the following diagram and finally select the continue button to proceed payment



Pay with debit or credit card

We don't share your financial details with the merchant.

Country
Sri Lanka

Card number
5555 5555 5555 4444

Expires
04/21

CSC
450

First name
Shriram

Last name
Navaratnalingam

Billing address

Address line 1
Galle Road,

Address line 2
Colombo

City
Colombo

Postal code
10350

[✓] Ship to my billing address

Contact Information

Phone type
Mobile

Phone number
+94 777 678 678

Email
trnshri23@gmail.com

PayPal is the safer, faster way to pay

No matter where you shop, we help keep your financial information more secure.

The security you want.
The protection you deserve.

Creating a PayPal account is optional but you'll get **PayPal Buyer Protection** on all eligible purchases, plus faster checkout every time you shop. **See terms**

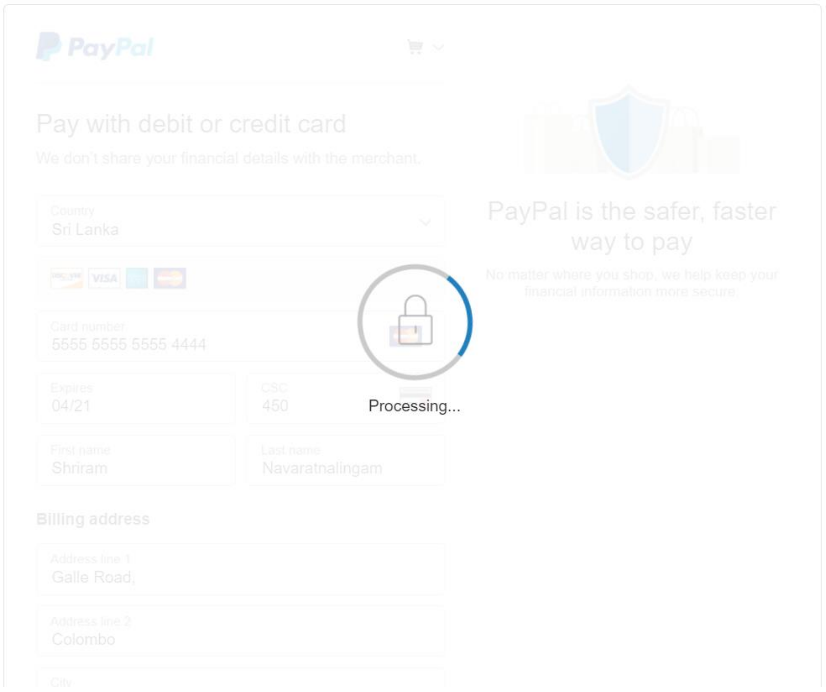
Create a PayPal account? (It only takes a moment.)

Yes, I'd like a PayPal account. [✓ No, I don't want an account now.]

Password

[✓] I confirm that I am of legal age and agree to the PayPal **Privacy Statement**.

Fig 13: Credit card details are processed by payment gateway



PayPal

Pay with debit or credit card

We don't share your financial details with the merchant.

Country: Sri Lanka

Card number: 5555 5555 5555 4444

Expires: 04/21

CSC: 450

First name: Shriram

Last name: Navaratnalingam

Billing address

Address line 1: Galle Road,

Address line 2: Colombo

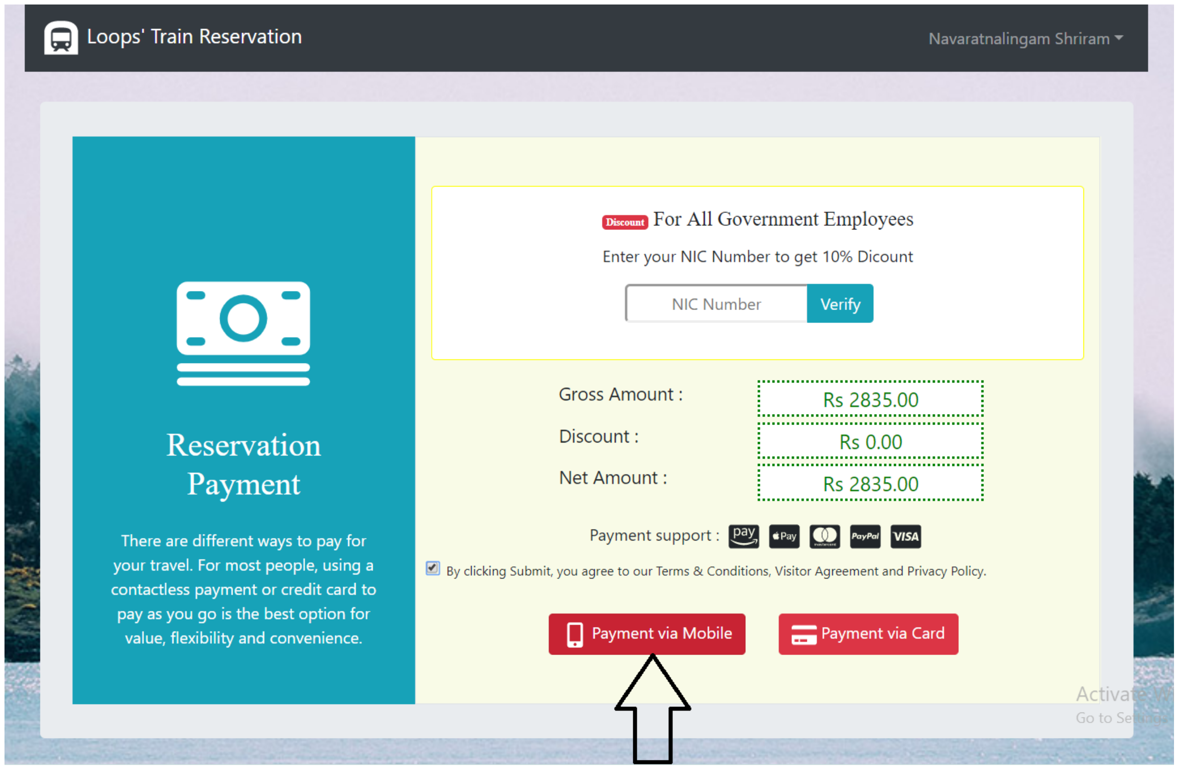
City

Processing...

PayPal is the safer, faster way to pay

No matter where you shop, we help keep your financial information more secure.

Fig 14: Else instead of Credit card user can select "Payment via Mobile"



Loops' Train Reservation

Navaratnalingam Shriram

Reservation Payment

There are different ways to pay for your travel. For most people, using a contactless payment or credit card to pay as you go is the best option for value, flexibility and convenience.

Discount For All Government Employees

Enter your NIC Number to get 10% Dicount

NIC Number | Verify

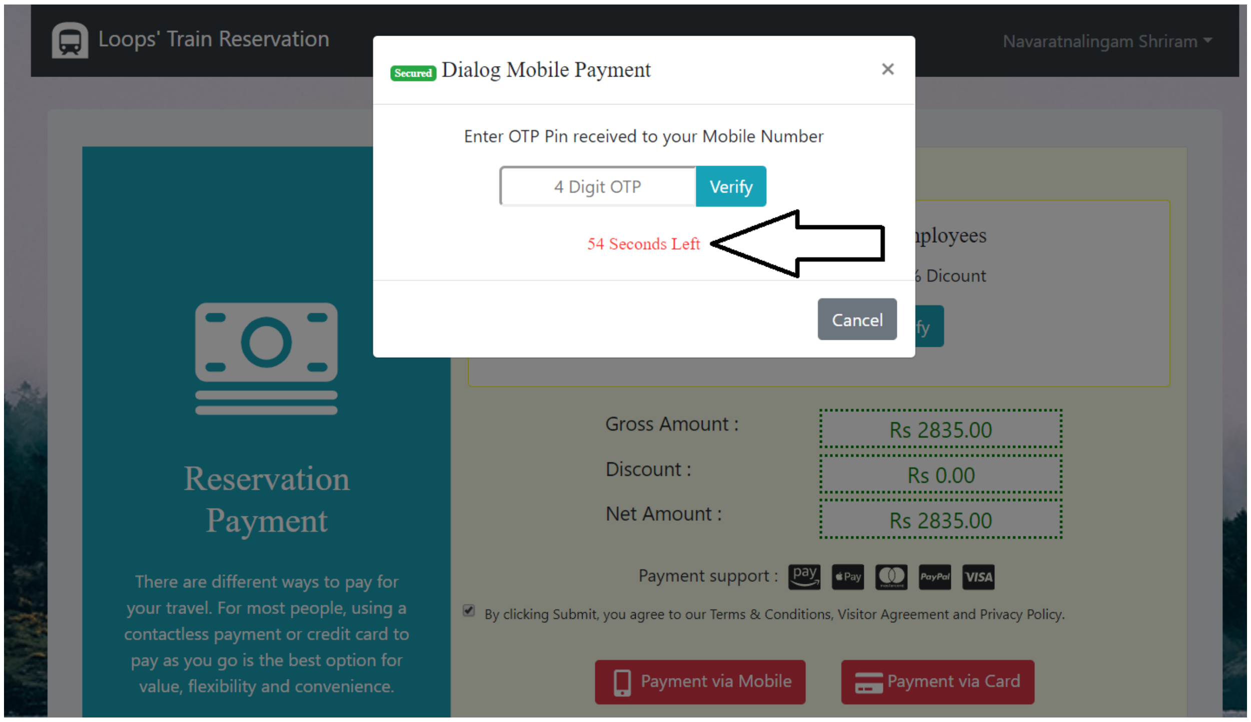
Gross Amount :	Rs 2835.00
Discount :	Rs 0.00
Net Amount :	Rs 2835.00

Payment support :

By clicking Submit, you agree to our Terms & Conditions, Visitor Agreement and Privacy Policy.

Payment via Mobile | Payment via Card

Fig 15: A Dialog message box will popup requesting 4 digit OTP, where that OTP also will be valid for specified number of time period in seconds that is running out.


Loops' Train Reservation

Navaratnalingam Shriram

Secured Dialog Mobile Payment

Enter OTP Pin received to your Mobile Number

4 Digit OTP | Verify

54 Seconds Left

Cancel

Reservation Payment

There are different ways to pay for your travel. For most people, using a contactless payment or credit card to pay as you go is the best option for value, flexibility and convenience.

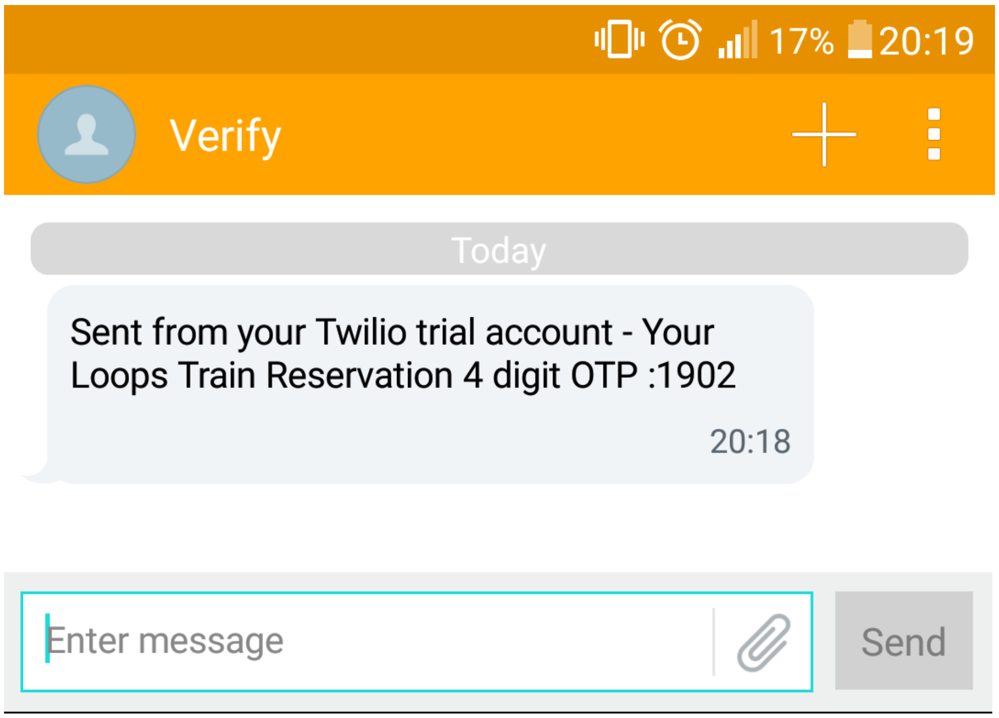
Gross Amount :	Rs 2835.00
Discount :	Rs 0.00
Net Amount :	Rs 2835.00

Payment support :

By clicking Submit, you agree to our Terms & Conditions, Visitor Agreement and Privacy Policy.

Payment via Mobile | Payment via Card

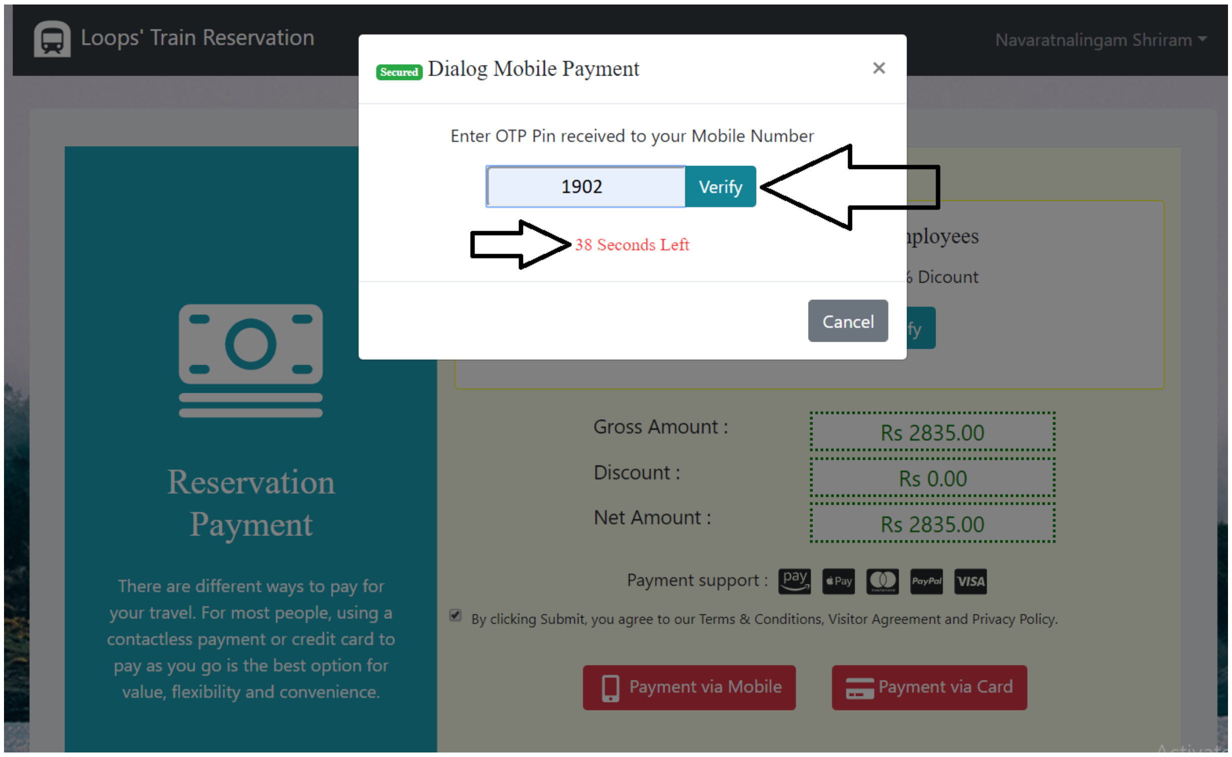
Fig 16: A 4 Digit OTP is sent to contact number to proceed transaction



Sent from your Twilio trial account - Your Loops Train Reservation 4 digit OTP :1902

20:18

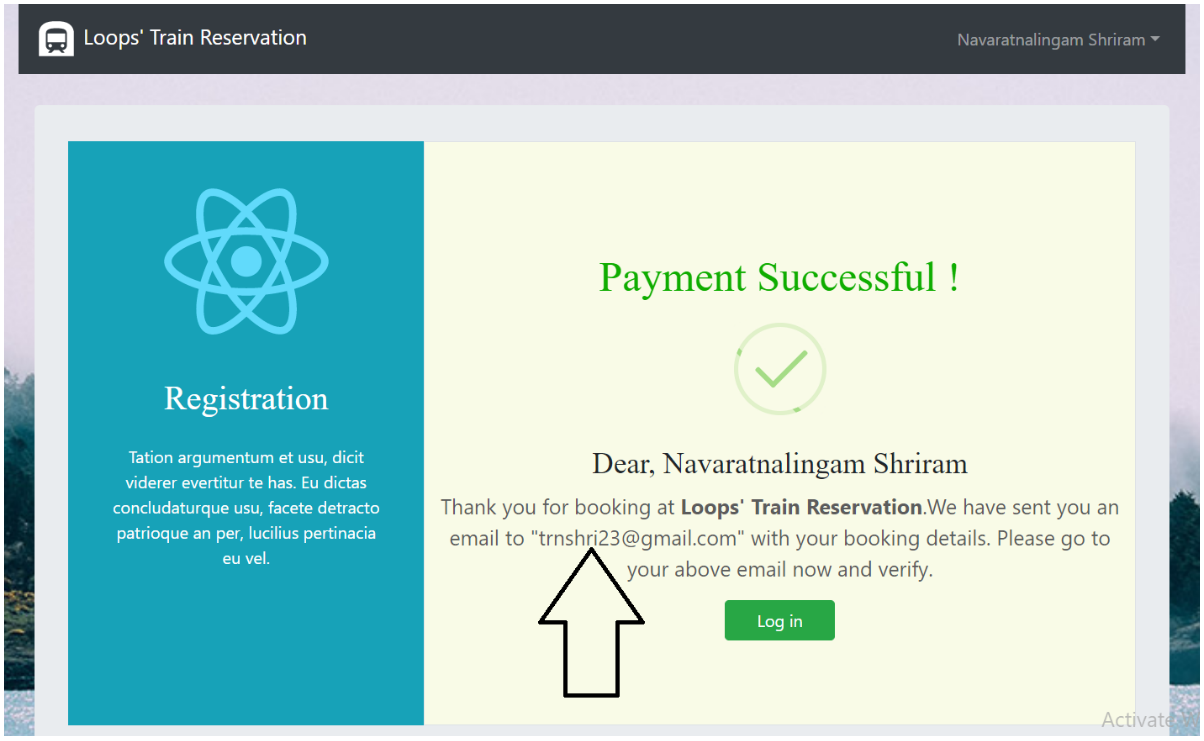
Fig 17: Input the OTP received over the SMS and select verify



There are different ways to pay for your travel. For most people, using a contactless payment or credit card to pay as you go is the best option for value, flexibility and convenience.

By clicking Submit, you agree to our Terms & Conditions, Visitor Agreement and Privacy Policy.

Fig 18: Successful reservation page after the mobile/credit card transaction

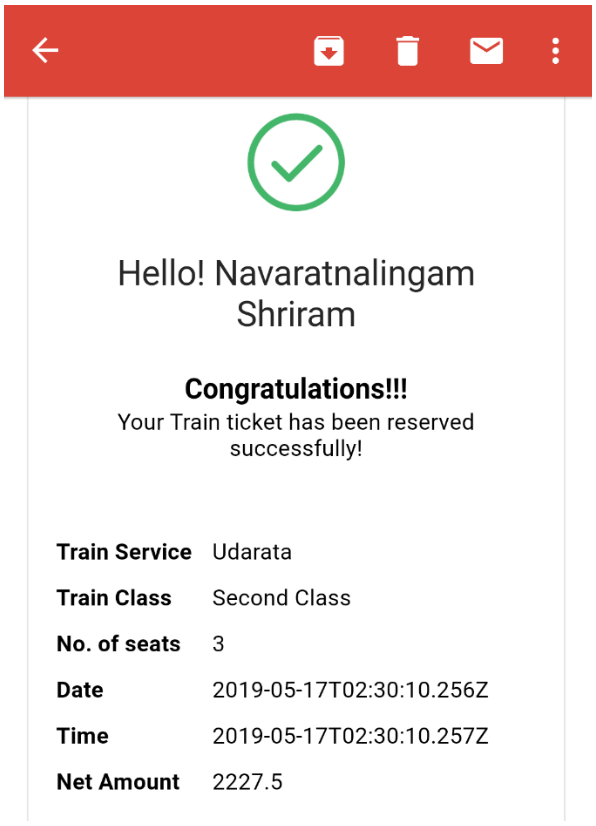


Tation argumentum et usu, dicit viderer evertitur te has. Eu dictas concludaturque usu, facete detracto patrioque an per, lucilius pertinacia eu vel.

Dear, Navaratnalingam Shriram

Thank you for booking at **Loops' Train Reservation**.We have sent you an email to "trnshri23@gmail.com" with your booking details. Please go to your above email now and verify.

Fig 19: Email received with the Train reservation details



Hello! Navaratnalingam Shriram

Congratulations!!!
Your Train ticket has been reserved successfully!

Train Service	Udarata
Train Class	Second Class
No. of seats	3
Date	2019-05-17T02:30:10.256Z
Time	2019-05-17T02:30:10.257Z
Net Amount	2227.5

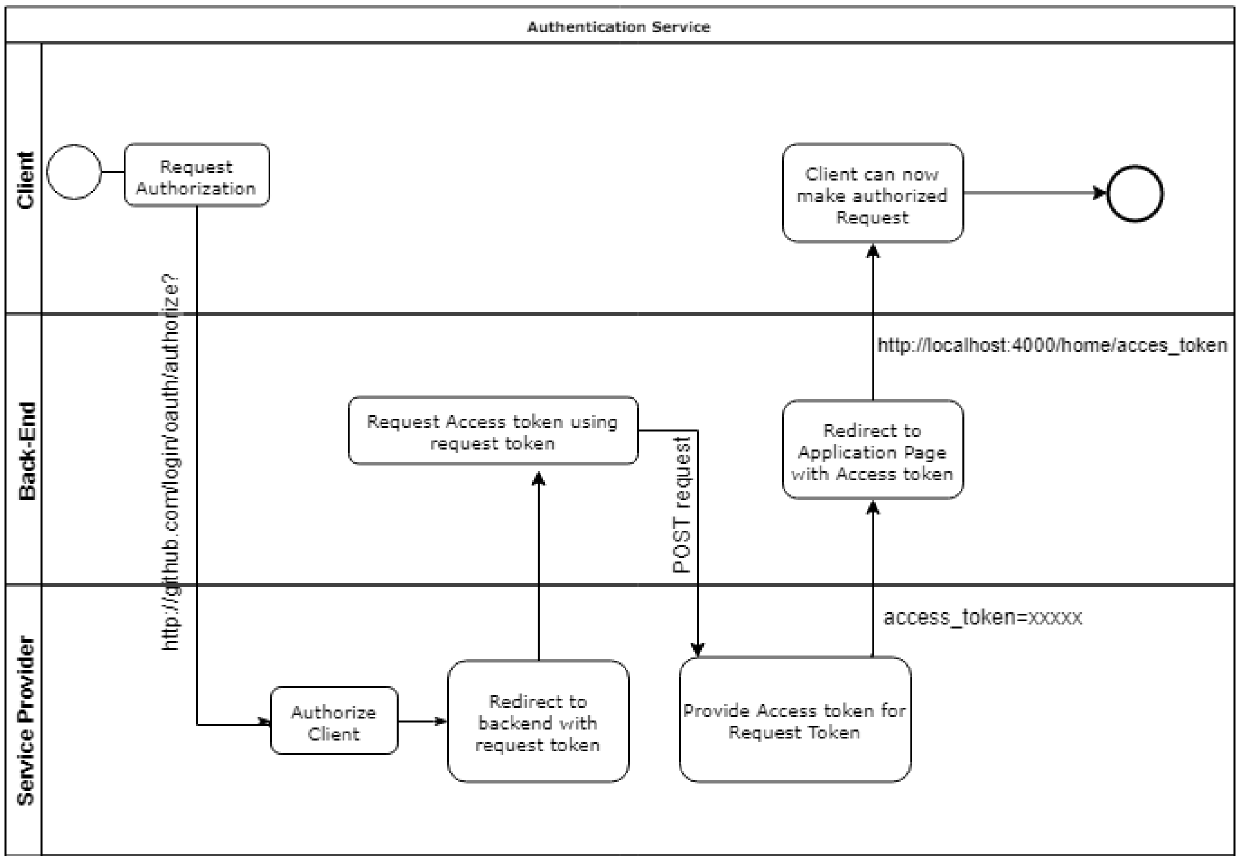
5. Back-End Service Implementation

5.1 Authentication Service

According to our system authentication process there are main three participants

- The End user – Person who uses the system and need to login
- Back End – The application end user required to sign in to application
- Service Provider –It's the third party service which is used to authenticate

In this application users are authenticated using GitHub OAuth2 API and build a separate backend service using node, which run in the port 4000. So in our case, the front end will be a web interface build using react and it would be running on localhost: 3000. Following activity diagram show the process of Authentication performed by the Application:

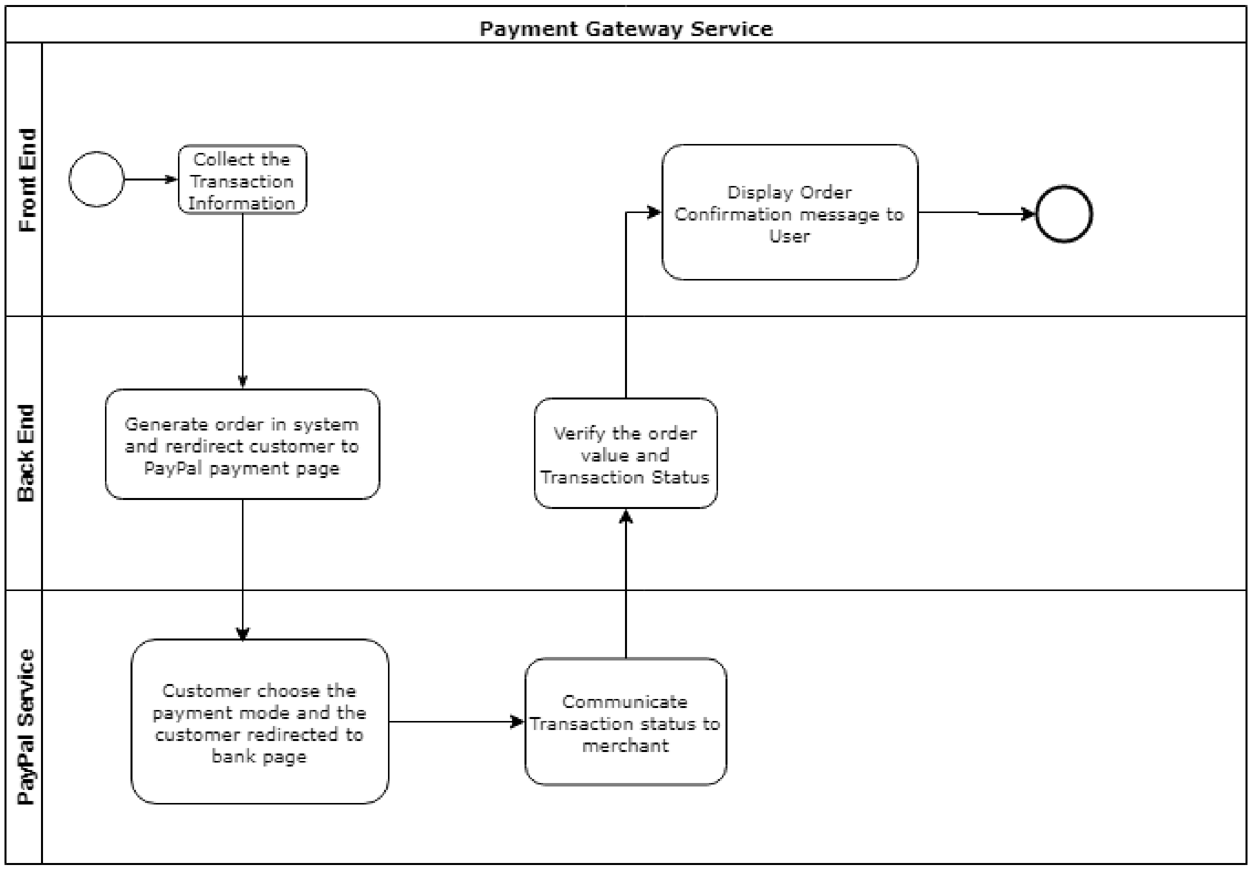

Authentication Service
Client
Back-End
Service Provider
Request Authorization
http://github.com/login/oauth/authorize?
Authorize Client
Redirect to backend with request token
Request Access token using request token
POST request
Provide Access token for Request Token
access_token=xxxxx
Redirect to Application Page with Access token
http://localhost:4000/home/acces_token
Client can now make authorized Request

Following code snippet does the process of getting Access token for the request token already received and then redirect to application page with access token:

```
const requestToken = req.query.code
  console.log(requestToken)
  axios({

    // make a POST request
    method: 'post',

    // to the Github authentication API, with the client ID, client secret
    // and request token
    url:
`https://github.com/login/oauth/access_token?client_id=${clientID}&client_
secret=${clientSecret}&code=${requestToken}`,

    // Set the content type header, so that we get the response in JSOn
    headers: {
         accept: 'application/json'
    }

  }).then((response) => {

    // Once we get the response, extract the access token from
    // the response body
    const accessToken = response.data.access_token
    console.log(response.data)

    // redirect the user to the welcome page, along with the access token
    res.redirect(`http://localhost:3000/home/${accessToken}`)

  })
})
```

For detailed description of building Authentication service using this Github OAuth2 API, could refer to my blog in medium:
https://medium.com/shriram-navaratnalingam/authentication-using-github-oauth-2-0-with-nodejs-be1091ce10a7

5.2 Payment Gateway service


Payment Gateway Service
Front End
Back End
PayPal Service
Collect the Transaction Information
Display Order Confirmation message to User
Generate order in system and rerdirect customer to PayPal payment page
Verify the order value and Transaction Status
Customer choose the payment mode and the customer redirected to bank page
Communicate Transaction status to merchant

Following code snippet show the configuration of PayPal sandbox and how to create the Payment object include the transaction details

```
// configure paypal with the credentials

    paypal.configure({
      //sandbox
     'mode': 'sandbox',

      //client id of the Paypal account
     'client_id': <YOUR CLIENT ID HERE>',

      // client secret of the Paypal account
     'client_secret': '<YOUR CLIENT SECRET HERE>'
    });
    // create payment object
    var payment = {
        "intent": "authorize",
        "payer": {
            "payment_method": "paypal"
        },
        "redirect_urls": {
          "return_url":
            "http://localhost:3000/home/"+reservation.token+"/success",
           "cancel_url":
                "http://localhost:4002/"
        },
        "transactions": [{
            "amount": {
                "total": reservation.netAmount,
                "currency": "USD"
            },
            "description": " a book on mean stack "
        }]
    }
```

Now following snippet show how transaction is done using the previously create Payment object,

```
var createPay = ( create_payment_json ) => {
    return new Promise( ( resolve , reject ) => {
    paypal.payment.create( create_payment_json,function(err, payment )
     {
            if ( err )
                reject(err);

            else
                resolve(payment);

        });
    });
}
```

```
// call the create Pay method
    createPay( payment )
        .then( ( transaction ) => {
            var id = transaction.id;
            var links = transaction.links;
            var counter = links.length;
            while( counter -- ) {
                if ( links[counter].method == 'REDIRECT') {
                    // redirect to paypal payment page
                   res.send(links[counter].href)
                }
            }
        })
        .catch( ( err ) => {
            console.log( err );
            res.redirect('/err');
        });
```

5.3 SMS Service

In this Application apart from the payment via Payment Gateway using the credit card or Debit card we also allows the user to pay for reservation by adding the booking sum to Mobile bill.

5.3.1 SMS Process flow

Here I have developed a Dummy payment when user selects “Payment via Mobile”, the back end service will be called where it generates 4 digit OTP number that is unique to user based on his authentication access token.

In the parallel time in order to promote secure transaction service being implemented in such a way OTP number is valid only for 60 seconds after that it will be disabled.

Once the appropriate OTP number is verified, user would be redirected in to success page.

Following code snippet shows how a unique 4 digit OTP will be generated based on user access token

```
const token = Speakeasy.totp({
        secret: secret,
        encoding: "base32",
        step: 60, // specified in seconds
        digits :4
    });

    res.send({
        "token": token,
        "remaining":(60-Math.floor((new Date()).getTime() /1000.0 % 0))
    });
```

Following code snippet shows how OTP digit is verified against the user input

```
Router.route('/validate').post(function(req,res){
    const pass = req.body.pass;
    console.log(pass.secret)
    console.log(pass.otp)

    res.send({
        "valid": Speakeasy.totp.verify({
            secret: pass.secret,
            encoding: "base32",
            token: pass.otp,
            step: 60,
            digits :4
        })
    });
})
```

Following code snippet shows the way SMS is sent including the 4 digit OTP and the time remaining in seconds for the OTP validity

```
//Sending otp through SMS
    const accountSid = '<YOUR TWILIO ACCOUNT ID>';
    const authToken = '<YOUR TWILIO ACCOUNT AUTH>';
    const client = require('twilio')(accountSid, authToken);

client.messages
        .create({
            body: 'Your Loops Train Reservation 4 digit OTP :'+token,
            from: '+12029028295',
            to: '+94777304722'
        })
        .then(message => console.log(message.sid));
```

5.4 Email Service

In the System implementation after every successful train seat reservation whether transaction carried out through credit/Debit card or mobile payment, user will be sent an email ensuring the transaction and including the reservation details.

I have implemented this service using “node mailer” which look after the email sending process and for creating email template “EJS” template engine has been used

Following code snippet show the configuration of credential for nodemailer:

```
module.exports = {
    USER: <YOUR EMAIL ID>',
    PASS: '<YOUR EMAIL PASSWORD>'
}
```

Following code snippet shows how Nodemailer transport instance is created

```
   //Dependiencies needed for Email service
    const nodemailer = require('nodemailer');
    const ejs = require("ejs");
    const creds = require('./config/credential.js')




//Creating transport instance
    var transport = {
        host: 'smtp.gmail.com',
        auth: {
        user: creds.USER,
        pass: creds.PASS
        }
    }

    //Creating a Nodemailer Transport instance
    var transporter = nodemailer.createTransport(transport)

    //Verifying the Nodemailer Transport instance
    transporter.verify((error, success) => {
        if (error) {
            console.log(error);
        } else {
            console.log('Server is ready to take messages');
        }
    });
```

Following code snippet shows how EJS template engine is integrated to the node mailer with appropriate parameter

```
    //Manipulating data to ejs mail template
    ejs.renderFile(__dirname +"/template/Hello.ejs", {},function(err,data)
{
        if (err) {
            console.log(err);
        } else {
            var mainOptions = {
                from: '"FindMyTrip" findmytrip2017@gmail.com',
                to: reservation.email,
                subject: 'Train Ticket Receipt',
                html: data
            };

            transporter.sendMail(mainOptions, function (err, info) {
              if (err) {
                res.status(200).json({"MAIL":"Not Sent"})
              } else {
                res.status(200).json({"MAIL":"Successfully Sent"})
              }
          });
        }
    });
```

For further clarification on the Email service, could refer to my blog on Medium: https://medium.com/shriram-navaratnalingam/send-emails-with-ejs-template-using-nodemailer-800b8f1a012d

For further clear view on implementation and complete code base could refer to my GitHub repos :

https://github.com/ShriLingam23/ESB-ServiceIntegration

6.0 WSO2 ESB Integration

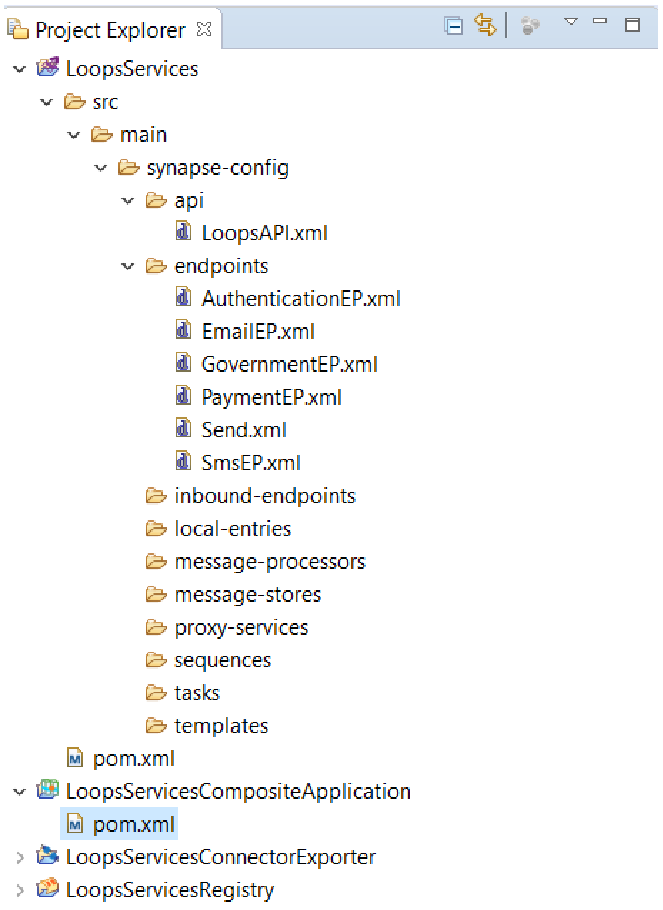
This system is implemented using SOA Architecture where we use WSO2 EI which is an abbreviation of WSO2 Enterprise Integrator. It's a complete supportive integration solution that promotes interaction and communication among diverse, and independent applications.

So Instead of allowing the applications to interoperate with each other which uses various formats to communicate, we make use of WSO2 EI to communicate. ESB handle the routing messages to appropriate endpoint according to the configurations.

In this System implementation, I have developed to system to request the backend services based on the URL routing. So when routing to appropriate Service using "Switch" for the value Source XPath property I made use of

get-property('uri.var.<parameter>')

Following diagram shows the Project Structure of the WSO2 EI Tooling,

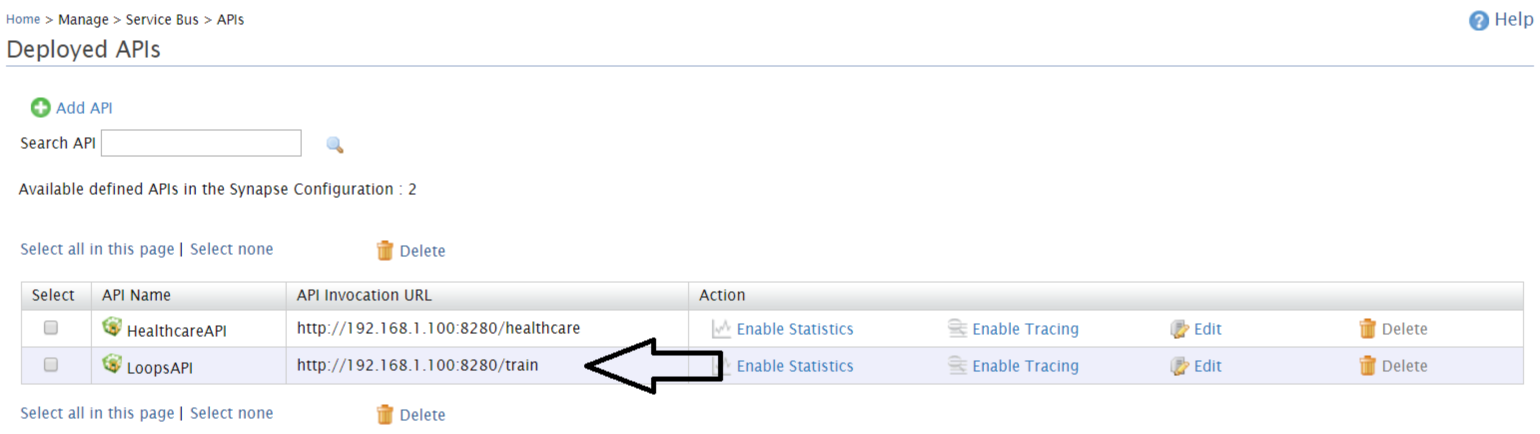


```
LoopsServices
  src
    main
      synapse-config
        api
          LoopsAPI.xml
        endpoints
          AuthenticationEP.xml
          EmailEP.xml
          GovernmentEP.xml
          PaymentEP.xml
          Send.xml
          SmsEP.xml
        inbound-endpoints
        local-entries
        message-processors
        message-stores
        proxy-services
        sequences
        tasks
        templates
  pom.xml
LoopsServicesCompositeApplication
  pom.xml
LoopsServicesConnectorExporter
LoopsServicesRegistry
```

Following code snippet shows the LoopsAPI.xml “Switch” configuration,

```
<switch source="get-property('uri.var.category')">
    <case regex="government">
        <log level="custom">
            <property name="message" value="&quot;Government Service&quot;"/>
        </log>
        <send>
            <endpoint key="GovernmentEP"/>
        </send>
    </case>
    <case regex="paypal">
        <log level="custom">
            <property name="message" value="&quot;Payment Service&quot;"/>
        </log>
        <send>
            <endpoint key="PaymentEP"/>
        </send>
    </case>
    <case regex="email">
        <log level="custom">
            <property name="message" value="&quot; Email Service&quot;"/>
        </log>
        <send>
            <endpoint key="EmailEP"/>
        </send>
    </case>
    <case regex="sms">
        <log level="custom">
            <property name="message" value="&quot; SMS Service&quot;"/>
        </log>
        <send>
            <endpoint key="SmsEP"/>
        </send>
    </case>

</switch>
```

Following figure shows the successfully deployed REST API to WSO2 carbon server


Home > Manage > Service Bus > APIs

Deployed APIs

Available defined APIs in the Synapse Configuration : 2

Select	API Name	API Invocation URL	Action			
☐	HealthcareAPI	http://192.168.1.100:8280/healthcare	Enable Statistics	Enable Tracing	Edit	Delete
☐	LoopsAPI	http://192.168.1.100:8280/train	Enable Statistics	Enable Tracing	Edit	Delete

7.0 Appendix

7.1 Front End (react)

Landing.js

```
class Landing extends Component{

    render(){

        return(
            <div className="App">

                <div className="overlay"><img src={Img} width='100%'
/></div>

                <div className="masthead">
                    <div className="masthead-bg"></div>
                    <div className="container h-300">
                    <div className="row h-100">
                        <div className="col-12 my-auto">
                        <div className="masthead-content text-white py-5
py-md-0" style={{marginTop:'100px'}}>
                            <h1 className="mb-3">

                            <MdSubway size='100px'/>
                            </h1>
                            <h1 className="mb-3">Train Reservations</h1>
                            <div className="input-group input-group-
newsletter">

                            <div className="input-group-append"
style={{marginLeft:'75px'}}>
                                <a className="btn btn-secondary"
href="http://github.com" target='_blank'
style={{borderTopLeftRadius:'50%'}} >
                                <FaGithub size='100px' onClick='#'/>
                                </a>
                                <a

href="https://github.com/login/oauth/authorize?client_id=42c283f5dd440faa8
fc5"
style={{textDecoration:'none',color:'#FFF',alignItems:'center',display:'fl
ex'}}
                                    className="btn btn-secondary"
                                >
                                    Authenticate!
                                </a>

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

                <div className="social-icons">
                    <ul className="list-unstyled text-center mb-0">
                    <li className="list-unstyled-item">
                        <a href="http://instagram.com" target='_blank'>
                        <FaInstagram style={{marginBottom:'7px'}}/>
                        </a>
                    </li>
                    <li className="list-unstyled-item ">
                        <a href="http://fb.com" target='_blank'>
                        <FaFacebook style={{marginBottom:'7px'}} />
                        </a>
                    </li>
                    <li className="list-unstyled-item">
                        <a href="http://twitter.com" target='_blank'>
                        <FaTwitter style={{marginBottom:'7px'}}/>
                        </a>
                    </li>
                    </ul>
                </div>

            </div>
        )
    }
}

export default Landing;
```

Home.js

```
class Home extends Component{

    constructor(props) {
        super(props);

        this.userLogout = this.userLogout.bind(this);
        this.state = {
                collapse: false,
                token:props.match.params.id,
                user:''
            };
    }

    componentDidMount(){

        // Calling the end-user information using the fetch call
        fetch('https://api.github.com/user', {
            headers: {
                // Including the token to the Authorization header
                Authorization: 'token ' + this.state.token
            }
        })

        // Parsing the response as JSON object
        .then(res => res.json())
        .then(res => {
            // Once response is received store it in state
            this.setState({user:res.name});
            console.log(this.state.user)
        })

    }

    userLogout(){
        this.setState({user:''});
        this.props.history.push('/')
    }

    render(){
        return (

            <div className="parallax">
```

```
                    <div className="container" >
                    <header>
                    <Navbar color="dark" dark expand="md">
                        <NavbarBrand><Link to={'/home/'+this.state.token}
style={{textDecoration:'none',color:'#fff'}}><MdSubway size='40px'/>{ }
Loops' Train Reservation</Link></NavbarBrand>
                        <NavbarToggler onClick={this.toggle} />
                        <Collapse isOpen={this.state.isOpen} navbar>
                            <Nav className="ml-auto" navbar>
                            <UncontrolledDropdown nav inNavbar>
                                <DropdownToggle nav caret>
                                {this.state.user}
                                </DropdownToggle>
                                <DropdownMenu right>
                                    <DropdownItem>
                                        <button className='btn'
onClick={this.userLogout} style={{color:'#000',border:'none'}}>Log
Out</button>
                                    </DropdownItem>
                                </DropdownMenu>
                            </UncontrolledDropdown>
                            </Nav>
                        </Collapse>
                        </Navbar>

                    </header>
                    {/* <!--Body--> */}
                    <Route exact path='/home/:id' component={Home_Content}
/>
                    <Route exact path='/home/:id/booking'
component={Reservation} />
                    <Route exact path='/home/:id/payment'
component={Payment} />
                    <Route path='/home/:id/success' component={Success} />

                </div>

            </div>
        )
    }
}

export default Home;
```

HomeContent.js

```
class Home_Content extends Component{

    constructor(props) {
        super(props);
        this.toggle = this.toggle.bind(this);
        this.state = {
                collapse: false,
                token:props.match.params.id
            };
    }

    toggle() {
        this.setState(state => ({ collapse: !state.collapse }));
    }

    render(){
        return(
            <main role="main" >

                <section className=" text-center"  >
                    <br/>
                    <Slide />

                    <p className="card lead"
style={{marginTop:'20px',padding:'20px'}}>

                    <div>
                    <Button color="dark" onClick={this.toggle}
><MdStreetview size="20px"/></Button>
                        <Collapse isOpen={this.state.collapse}>
                            <div className="container">
                                <div className="row">
                                    <div className="col-sm-8 col-md-7 py-
4">
                                        <h4 className="text-
white">About</h4>

</div>
                                    <div className="col-sm-4 offset-md-1
py-4">
                                        <h4 className="text-
black">Contact</h4>
```

```
                                        <ul className="list-unstyled">
                                            <li><a href="#"
className="text-black" style={{textDecoration:'none'}}><FaTwitter/> Follow
on Twitter</a></li>
                                            <li><a href="#"
className="text-black" style={{textDecoration:'none'}}><FaFacebook/> Like
on Facebook</a></li>
                                            <li><a href="#"
className="text-black" style={{textDecoration:'none'}}><FaInstagram/>
Email me</a></li>
                                        </ul>
                                    </div>
                                </div>
                            </div>
                        </Collapse>
                        </div>
                    <hr/>
                    <Row style={{marginLeft:'7px',textAlign:'left'}}>
                        <Col xs="8"
style={{fontStyle:'italic',fontVariant:'small-caps'}}>
                        </Col>
                        <Col xs="4"
style={{display:'flex',alignItems:'center',paddingLeft:'15px'}}>
                            <Link
to={'/home/'+this.state.token+'/booking'} className="btn btn-success
"><IoMdCalendar size="50px" /> Make a Reservation</Link>
                        </Col>

                    </Row>
                    </p>

                </section>

            </main>
        )
    }
}
export default Home_Content;
```

Reservation.js

```
class Reservation extends Component{

    constructor(props) {
        super(props);
        this.state = {
            visible: false,
            pending: false,
            token:props.match.params.id,
            user:'',
            fullName:'',
            email:'',
            contactNum:'',
            trainService:'',
            trainClass:'',
            numTickets:'',
            totalPrice:'',
            scheduleDate:setHours(setMinutes(new Date(), 0), 8),
            scheduleTime:setHours(setMinutes(new Date(), 0), 8)

        };

        this.onFormSubmit= this.onFormSubmit.bind(this);
        this.onValueChange = this.onValueChange.bind(this);

        this.onDismiss = this.onDismiss.bind(this);
        this.newlyAdded = this.newlyAdded.bind(this);
        this.checkPending = this.checkPending.bind(this);
        this.checkTotal= this.checkTotal.bind(this);
        this.handleChangeDate = this.handleChangeDate.bind(this);
        this.handleChangeTime = this.handleChangeTime.bind(this);
    }

    componentDidMount(){

        fetch('https://api.github.com/user', {
            headers: {
                Authorization: 'token ' + this.state.token
            }
        })
        .then(res => res.json())
        .then(res => {
            this.setState({user:res.name});
```

```
                console.log(this.state.user)
            })

    }

    handleChangeDate(date) {
        this.setState({
          scheduleDate: date
        });
    }

    handleChangeTime(date) {
        this.setState({
          scheduleTime: date
        });
    }

    onDismiss() {
        this.setState({ visible: false });
    }

    newlyAdded(){
        if(this.state.visible){
            return (
                <div>
                    <Alert color="success" isOpen={this.state.visible}
toggle={this.onDismiss} fade={false}>
                        Staff details successfully added and a
Confirmation mail has been sent!
                    </Alert>
                </div>
            );
        }
    }

    checkPending(){
        if(this.state.pending){

            return (
                <div className="col-md-8 py-5 border" >
                    <Spinner style={{ width: '10rem', height:
'10rem',paddingTop:'50px'}} type="grow" color="warning" />
                </div>
            )
        }
        else{
            return(
                <div className="col-md-8 py-5 border">
                    <h4 className="pb-4">Train Reservation Form</h4>
                    <form id='staffForm' onSubmit={this.onFormSubmit}>
                        <div className="form-row">
                            <div className="input-group form-group col-md-
6">
                                <div className="input-group-prepend">
                                    <div className="input-group-
text"><MdPerson/></div>
                                </div>
                                <input
                                    name="fullName"
                                    placeholder="Full Name"
                                    className="form-control"
                                    type="text"
                                    onChange={this.onValueChange}
                                    value={this.state.user} readOnly/>
                            </div>
                            <div className="input-group form-group col-md-
6">
                                <div className="input-group-prepend">
                                    <div className="input-group-
text"><MdEmail/></div>
                                </div>
                                <input
                                    name="email"
                                    placeholder="Email"
                                    className="form-control"
                                    type="email"
                                    onChange={this.onValueChange}
                                    value={this.state.email} />
                            </div>
                        </div>
                        <div className="form-row">
                            <div className="input-group form-group col-md-
6">
                                <div className="input-group-prepend">
                                    <div className="input-group-
text"><MdPhone/></div>
                                </div>
```

```
                                            <input
                                                name="contactNum"
                                                placeholder="Contact No."
                                                className="form-control"
                                                required="required"
                                                type="tel"
                                                onChange={this.onValueChange}
                                                value={this.state.contactNum}/>
                                        </div>
                                        <div className="input-group form-group col-md-
6">
                                            <div className="input-group-prepend">
                                                <div className="input-group-
text"><FaSubway/></div>
                                            </div>
                                            <select name="trainService"
className="form-control" onChange={this.onValueChange}>
                                                <option selected>Choose the Train
service ...</option>
                                                <option>Udarata</option>
                                                <option>Utara Devi</option>
                                                <option>Yal Devi</option>
                                                <option>Southern Express</option>
                                            </select>
                                        </div>
                                    </div>
                                    <div className="form-row">
                                        <div className="input-group form-group col-md-
6">
                                            <div className="input-group-prepend">
                                                <div className="input-group-
text"><FaRegStar/></div>
                                            </div>
                                            <select name="trainClass" className="
form-control" onChange={this.onValueChange}>
                                                <option selected>Choose Train Class
...</option>
                                                <option>First Class</option>
                                                <option>Second Class</option>
                                                <option>Third Class</option>
                                            </select>
                                        </div>
                                        <div className="input-group form-group col-md-
6">
```

```
                                <div className=" input-group-prepend">
                                    <div className="input-group-
text"><FaTicketAlt/></div>
                                </div>
                            <select name="numTickets" className=" form-
control" onChange={this.onValueChange}>
                                <option selected>Choose Number of Tickets
...</option>
                                <option>1</option>
                                <option>2</option>
                                <option>3</option>
                                <option>4</option>
                                <option>5</option>
                                <option>6</option>
                                <option>7</option>
                                <option>8</option>
                                <option>9</option>
                            </select>
                            </div>
                        </div>
                        <div className="form-row">
                            <div className="input-group form-group col-md-
6">
                                <div className=" input-group-prepend">
                                    <div className="input-group-
text"><FaCalendarAlt/></div>
                                </div>
                                <DatePicker
                                    selected={this.state.scheduleDate}
                                    onChange={this.handleChangeDate}
                                    todayButton={"Today"}
                                    dateFormat="dd/MM/yyyy"
                                    minDate={new Date()}
                                    maxDate={addDays(new Date(), 5)}
                                    placeholderText="Select a date between
today and 5 days in the future"
                                    className="form-control"
                                    style={{marginRight:'50px'}}
                                />
                            </div>
                            <div className="input-group form-group col-md-
6">
                                <div className=" input-group-prepend">
```

```
                                    <div className="input-group-
text"><FaClock/></div>
                                </div>
                                <DatePicker
                                    selected={this.state.scheduleTime}
                                    onChange={this.handleChangeTime}
                                    showTimeSelect
                                    showTimeSelectOnly
                                    timeIntervals={120}
                                    minTime={setHours(setMinutes(new
Date(), 0), 8)}

                                    maxTime={setHours(setMinutes(new
Date(), 0), 20)}

                                    dateFormat="h:mm aa"
                                    timeCaption="Time"
                                    className="form-control"
                                />
                            </div>
                        </div>
                        <div style={{margin:'5px'}}>
                            <input

style={{fontSize:'20px',color:'green',border:'dotted',textAlign:'center'}}
                                placeholder='Total Amount'
                                readOnly
                                value={'Rs
'+parseFloat(Math.round(this.state.totalPrice * 100) / 100).toFixed(2)}/>
                        </div>
                        <div>
                            Payment support : { }<FaCcAmazonPay
size='40px'style={{padding:'5px'}}/>
                            <FaCcApplePay
size='40px'style={{padding:'5px'}}/>
                            <FaCcMastercard
size='40px'style={{padding:'5px'}}/>
                            <FaCcPaypal
size='40px'style={{padding:'5px'}}/>
                            <FaCcVisa size='40px'style={{padding:'5px'}}/>
                        </div>
                        <div className="form-row">
                            <div className="form-group">
                                <div className="form-group">
                                    <div className="form-check">
                                    <input className="form-check-input"
type="checkbox" value="" id="invalidCheck2" required/>
                                    </div>
                                </div>

                            </div>
                        </div>

                        <div className="form-row"
style={{display:'flex',justifyContent:'center'}}>
                            <button type='submit' className="btn btn-
danger" >Proceed to Payment</button>
                        </div>

                    </form>
                </div>
            )
        }
    }

    onValueChange(e){
        this.setState({
            [e.target.name]:e.target.value
        },()=>{
            this.checkTotal()
        })

    }

    checkTotal(){

        if(this.state.trainClass!=='')
            console.log(this.state.trainClass)

        if(this.state.trainService!=='')
            console.log(this.state.trainService)

        if(this.state.numTickets!=='')
            console.log(this.state.numTickets)

        if(this.state.trainClass!=='' && this.state.trainService!=='' &&
this.state.numTickets!==''){

            let total=0.0;
```

```
 if(this.state.trainClass==='First Class'){
     console.log(true);
     switch(this.state.trainService){
         case 'Udarata':total=1075.00
             break;
         case 'Utara Devi': total=1150.00
             break;
         case 'Yal Devi': total=975.00
             break;
         case 'Southern Express': total=840.00
             break;
         default  : total=0.00
     }
     console.log(total)
 }
 else if(this.state.trainClass==='Second Class'){

     switch(this.state.trainService){
         case 'Udarata': total=825.00
             break;
         case 'Utara Devi': total=945.00
             break;
         case 'Yal Devi': total=765.00
             break;
         case 'Southern Express': total=600.00
             break;
         default  : total=0.00

     }

 }
 else if(this.state.trainClass==='Third Class'){

     switch(this.state.trainService){
         case 'Udarata': total=650.00
             break;
         case 'Utara Devi': total=685.00
             break;
         case 'Yal Devi': total=615.00
             break;
         case 'Southern Express': total=400.00
             break;
         default  : total=0.00
}  }
```

```
            else
                total=0.0;

            // Manipulate with Number of seats
            switch(this.state.numTickets){
                case '1': total *=1;
                    break;
                case '2': total *=2;
                    break;
                case '3': total *=3;
                    break;
                case '4': total *=4;
                    break;
                case '5': total *=5;
                    break;
                case '6': total *=6;
                    break;
                case '7': total *=7;
                    break;
                case '8': total *=8;
                    break;
                case '9': total *=9;
                    break;
                default : total= 0.00;
            }

            console.log(total)
            this.setState({totalPrice:total})
        }

    }

    onFormSubmit(e){
        this.setState({pending:true})
        e.preventDefault();

        const token = this.state.token;
        const user = this.state.user;
        const fullName = this.state.fullName;
        const email = this.state.email;
        const contactNum = this.state.contactNum;
        const trainService = this.state.trainService;
        const trainClass = this.state.trainClass;
        const numTickets = this.state.numTickets;
        const totalPrice = this.state.totalPrice;
        const scheduleDate = this.state.scheduleDate;
        const scheduleTime = this.state.scheduleTime;


        const Reservation={
            token,
            user,
            fullName,
            email,
            contactNum,
            trainService,
            trainClass,
            numTickets,
            totalPrice,
            scheduleDate,
            scheduleTime

        }

        //Setting up sessionStorage
        localStorage.setItem('reservation', JSON.stringify(Reservation));

        this.props.history.push('/home/'+token+'/payment')

    }

    render(){
        return(
            <div className="container" style={{paddingTop:'20px'}}>

                {this.newlyAdded()}
                {/* <!--Body--> */}

                <main role="main" style={{marginTop:'10px'}}>

                    <section className="jumbotron text-center" >
                        <div className="container"
style={{backgroundColor:'#f9fbe7',marginTop:'-30px',marginBottom:'-
30px'}}>

                            <div className='row' >
                                <div className='col-md-4 bg-info text-
white text-center'>
```

```
                                    <div className="card-body" >
                                        <IoIosSubway
size='150px'style={{marginTop:'100px'}}/>
                                        <h2 className="py-
3">Reservation</h2>
                                    </div>
                                </div>

                                {/* form section */}
                                {this.checkPending()}
                            </div>
                        </div>

                    </section>
                </main>

            </div>
        );
    }
}

export default Reservation;
```

Payment.js

```
class Payment extends Component{

    constructor(props) {
        super(props);

        this.state = {
            visible: false,
            pending: false,
            token:props.match.params.id,
            user:'',
            grossAmount:0.00,
            discount:0.00,
            netAmount:0.00,
            reservation:{},
            id:'',
            typeOfSubmit:'card',
            modal: false,
            value:0,
            otp:''
        };

        this.onFormSubmit= this.onFormSubmit.bind(this);
        this.onVerify = this.onVerify.bind(this);
        this.onIdChange = this.onIdChange.bind(this);

        this.onDismiss = this.onDismiss.bind(this);
        this.newlyAdded = this.newlyAdded.bind(this);
        this.checkPending = this.checkPending.bind(this);
        this.checkTotal= this.checkTotal.bind(this);
        this.onMobilePay= this.onMobilePay.bind(this);
        this.onCardPay= this.onCardPay.bind(this);

        //For Modal
        this.toggle = this.toggle.bind(this);

        //Timer
        this._decrease= this._decrease.bind(this);
        this.controlTimer= this.controlTimer.bind(this);

        //Otp
        this.onOtpChange=this.onOtpChange.bind(this);
        this.onVerifyOtp=this.onVerifyOtp.bind(this);
    }

    componentDidMount(){

        const reservation = 
JSON.parse(localStorage.getItem('reservation'));
        //console.log(reservation)
        this.setState({reservation:reservation})

this.setState({grossAmount:reservation.totalPrice,netAmount:reservation.to
talPrice})
        this.setState({user:reservation.user,token:reservation.token});

        if("discount" in reservation){
            this.setState({discount:reservation.discount})
        }

    }

    toggle() {
        this.setState(prevState => ({
          modal: !prevState.modal
        }),()=>this.controlTimer());

    }

    controlTimer(){
        if(this.state.modal===false)
            this.setState({value:-1},)


        if(this.state.modal===true){

            //sending new otp to user

axios.post('http://192.168.1.100:8280/train/reserve/sms/otp',{'secret':thi
s.state.token})
            .then(
                res =>{
                    console.log(res.data)

this.setState({value:res.data.remaining},()=>this._decrease())
                }
            )
        }

    }

    _decrease() {
        this.setState({ value: this.state.value - 1 });

        if(this.state.value>=1){
            setTimeout(()=>{
                this._decrease()
            }, 1000);
        }
        else if(this.state.value==0)
            this.toggle()


    }

    onDismiss() {
        this.setState({ visible: false });
    }

    newlyAdded(){
        if(this.state.visible){
            return (
                <div>
                    <Alert color="success" isOpen={this.state.visible}
toggle={this.onDismiss} fade={false}>
                        Government Special Discount added to receipt!
                    </Alert>
                </div>
            );
        }
    }

    checkPending(){
        if(this.state.pending){

            return (
                <div className="col-md-8 py-5 border" >
                    <Spinner style={{ width: '10rem', height:
'10rem',paddingTop:'50px'}} type="grow" color="warning" />
                </div>
            )

        }
        else{
            return(
                <div className="col-md-8 py-5 border">
                    <form id='staffForm'>
                        <div className="row">
                            <div className="col-sm-12"
style={{marginBottom:'15px'}}>
                                <div className="card "
style={{borderColor:'yellow'}}>
                                <div className="card-body">
                                    <h5 className="card-title"><Badge
style={{fontSize:'10px'}} color="danger">Discount</Badge> For All
Government Employees</h5>
                                    <p className="card-text">Enter your
NIC Number to get 10% Discount</p>
                                    <div className="input-group mb-3"
style={{marginLeft:'170px'}}>
                                        <input
                                            type="text"
                                            placeholder="NIC Number"

style={{borderTopLeftRadius:'5px',borderBottomLeftRadius:'5px',borderStyle
:'inset',textAlign:'center'}}
                                            name='id'
                                            value={this.state.id}
                                            onChange={this.onIdChange}/>
                                        <div className="input-group-
append">
                                            <button className="btn btn-
info" type="button" onClick={this.onVerify}>Verify</button>
                                        </div>
                                    </div>
                                </div>
                                </div>
                            </div>
                        </div>

                        {/* //Model */}
                        <Modal isOpen={this.state.modal}
toggle={this.toggle} className={this.props.className}>
```

```
                            <ModalHeader toggle={this.toggle}><Badge
style={{fontSize:'10px'}} color="success">Secured</Badge>{' '} Dialog
Mobile Payment</ModalHeader>
                            <ModalBody>
                                <p className="text-center">Enter OTP Pin
received to your Mobile Number {this.state.reservation.cotactNum}</p>
                                <div className="input-group"
style={{marginLeft:'100px'}}>
                                    <input
                                        type="text"
                                        placeholder="4 Digit OTP"

style={{borderTopLeftRadius:'5px',borderBottomLeftRadius:'5px',borderStyle
:'inset',textAlign:'center'}}
                                        name='otp'
                                        value={this.state.otp}
                                        onChange={this.onOtpChange}/>
                                    <div className="input-group-append">
                                        <button className="btn btn-info"
type="button" onClick={this.onVerifyOtp}>Verify</button>
                                    </div>
                                </div> <br/>
                                <h6
style={{textAlign:'center',color:'#FF3333'}}>{this.state.value} {" "}
Seconds Left</h6>
                            </ModalBody>
                            <ModalFooter>
                                <Button color="secondary"
onClick={this.toggle}>Cancel</Button>
                            </ModalFooter>
                        </Modal>

                        <div className='row' style={{margin:'5px'}}>
                            <span className='col-md-2'></span>
                            <label className='col-4 text-left'
style={{textAlign:'',fontSize:'18px'}}>Gross Amount :</label> { }
                            <input

style={{fontSize:'20px',color:'green',border:'dotted',textAlign:'center'}}
                                placeholder='Total Amount'
                                readOnly
                                value={'Rs
'+parseFloat(Math.round(this.state.grossAmount * 100) / 100).toFixed(2)}/>
                        </div>
                        <div className='row' style={{margin:'5px'}}>
                            <span className='col-md-2'></span>
                            <label className='col-md-4 text-left'
style={{fontSize:'18px'}}>Discount :</label> { }
                            <input

style={{fontSize:'20px',color:'green',border:'dotted',textAlign:'center'}}
                                placeholder='Total Amount'
                                readOnly
                                value={'Rs
'+parseFloat(Math.round(this.state.discount * 100) / 100).toFixed(2)}/>
                        </div>
                        <div className='row' style={{margin:'5px'}}>
                            <span className='col-md-2'></span>
                            <label className='col-4 text-left'
style={{fontSize:'18px'}}>Net Amount :</label> { }
                            <input

style={{fontSize:'20px',color:'green',border:'dotted',textAlign:'center'}}
                                placeholder='Total Amount'
                                readOnly
                                value={'Rs
'+parseFloat(Math.round(this.state.netAmount * 100) / 100).toFixed(2)}/>
                        </div>
                        <div style={{marginTop:'15px'}}>
                            Payment support : { }<FaCcAmazonPay
size='40px'style={{padding:'5px'}}/>
                            <FaCcApplePay
size='40px'style={{padding:'5px'}}/>
                            <FaCcMastercard
size='40px'style={{padding:'5px'}}/>
                            <FaCcPaypal
size='40px'style={{padding:'5px'}}/>
                            <FaCcVisa size='40px'style={{padding:'5px'}}/>
                        </div>
                        <div className="form-row">
                            <div className="form-group">
                                <div className="form-group">
                                    <div className="form-check">
                                    <input className="form-check-input"
type="checkbox" value="" id="invalidCheck2" required/>
                                    </div>
                                </div>
```

```
                                        </div>
                                    </div>

                                    <div className="form-row"
style={{display:'flex',justifyContent:'center'}}>
                                        <div className='col-4'><button className="btn
btn-danger" onClick={this.onMobilePay}><FaMobileAlt size='25px'/> Payment
via Mobile</button></div>
                                        <div className='col-4'><button className="btn
btn-danger" onClick={this.onCardPay}><FaCreditCard size='25px'/> Payment
via Card</button></div>
                                    </div>

                                </form>
                            </div>
                        )
                    }
                }

onVerify(e){
axios.post('http://192.168.1.100:8280/train/reserve/government/verify',{'i
d':this.state.id})
            .then(
                res =>{
                    if(this.state.discount===0.00){
                        if(res.data.valid){
                            this.setState({
                                discount:(this.state.netAmount*10/100),
                                visible:true
                            },()=>this.checkTotal())
                        }
                    }
                }
            )
    }

    onVerifyOtp(){

        const pass = {
                        'secret':this.state.token,
                        'otp':this.state.otp
                    }
```

```
axios.post('http://192.168.1.100:8280/train/reserve/sms/validate',{'pass':
pass})
            .then(
                res =>{
                    console.log(res.data);
                    if(res.data.valid===true)

this.props.history.push('/home/'+this.state.token+'/success')
                }
            )
    }

    onIdChange(e){
        this.setState({id:e.target.value});
    }

    onOtpChange(e){
        this.setState({otp:e.target.value});
    }

    checkTotal(){
        this.setState({
            netAmount:this.state.grossAmount-this.state.discount
        })

    }

    onMobilePay(e){
        e.preventDefault();
        this.setState({
            typeOfSubmit:'mobile'
        },()=>{
            this.onFormSubmit()
        })

    }
    onCardPay(e){
        e.preventDefault()
        this.setState({
            typeOfSubmit:'card'
        },()=>{
            this.onFormSubmit()
        })   }
    onFormSubmit(){

        var reservation = JSON.parse(localStorage.getItem('reservation'));
            reservation ={
                            ...reservation,

discount:parseFloat(Math.round(this.state.discount * 100) /
100).toFixed(2),

netAmount:parseFloat(Math.round(this.state.netAmount * 100) /
100).toFixed(2)};

            console.log(reservation);
            localStorage.setItem('reservation',
JSON.stringify(reservation));

        if(this.state.typeOfSubmit==='card'){
            console.log("Card Pay")
            this.setState({pending:true})



axios.post('http://192.168.1.100:8280/train/reserve/paypal/pay',{'reservat
ion':reservation})
                .then(
                    (res)=>{
                        console.log(res)
                        window.location.replace(res.data);
                    }
            )
        }
        else{
            console.log("Mobile Pay")
            this.toggle()
        }
    }
```

```
render(){
        return(
            <div className="container" style={{paddingTop:'20px'}}>

                {this.newlyAdded()}
                {/* <!--Body--> */}

                <main role="main" style={{marginTop:'10px'}}>

                    <section className="jumbotron text-center" >
                        <div className="container"
style={{backgroundColor:'#f9fbe7',marginTop:'-30px',marginBottom:'-
30px'}}>

                            <div className='row' >
                                <div className='col-md-4 bg-info text-
white text-center'>
                                    <div className="card-body" >
                                        <IoIosCash
size='150px'style={{marginTop:'100px'}}/>
                                        <h2 className="py-3">Reservation
Payment</h2>
                                    </div>
                                </div>

                                {/* form section */}
                                {this.checkPending()}
                            </div>
                        </div>

                    </section>
                </main>

            </div>
        );
    }
}

export default Payment;
```

PaymentSuccess.js

```
class Payment extends Component{

    constructor(props) {
        super(props);

        this.state = {
            pending: false,
            token:props.match.params.id,
            user:'',
            reservation:{}
        };

        this.onFormSubmit= this.onFormSubmit.bind(this);
        this.checkPending = this.checkPending.bind(this);
        this.checkTotal= this.checkTotal.bind(this);

    }

    componentDidMount(){

        const reservation =
JSON.parse(localStorage.getItem('reservation'));
        //console.log(reservation)
        this.setState({user:reservation.user,reservation:reservation});

        //Sending email

axios.post('http://192.168.1.100:8280/train/reserve/email/confirm',{'reser
vation':reservation})
            .then(
                (res)=>{
                    console.log(res.data);
                }
            )
    }

    checkPending(){
        if(this.state.pending){

            return (
                <div className="col-md-8 py-5 border" >
```

```
                    <Spinner style={{ width: '10rem', height:
'10rem',paddingTop:'50px'}} type="grow" color="warning" />
                </div>
            )

        }
        else{
            return(
                <div className="col-md-8 py-5 border">

                    <div style={{padding:'8px'}}>
                        <div className="row text-center">
                            <div >
                            <br/><br/> <h1
style={{color:'#0fad00'}}>Payment Successful !</h1>

                            <div className="swal2-icon swal2-success
swal2-animate-success-icon" style={{display: 'flex'}}>
                                <div className="swal2-success-circular-
line-left" style={{backgroundColor: '#f9fbe7'}}></div>
                                    <span className="swal2-success-line-
tip"></span>

                                    <span className="swal2-success-line-
long"></span>

                                    <div className="swal2-success-
ring"></div>

                                    <div className="swal2-success-fix"
></div>
                                <div className="swal2-success-circular-
line-right" style={{backgroundColor: '#f9fbe7'}}></div>
                            </div>



                            <h3>Dear, {this.state.reservation.user}</h3>
                            <p style={{fontSize:'20px',color:'#5C5C5C'}}>
                                Thank you for booking at <b>Loops' Train
Reservation</b>.We have sent you an email to
"{this.state.reservation.email}" with your booking details.
                                Please go to your above email now and
verify.
                            </p>
                            <a href="http://www.gmail.com" className="btn
btn-success">     Log in     </a>
                            <br/><br/>
                            </div>

                        </div>
                    </div>

                </div>
            )
        }
    }

    checkTotal(){
        this.setState({
            netAmount:this.state.grossAmount-this.state.discount
        })

    }

    onFormSubmit(e){
        this.setState({pending:true})
        e.preventDefault();
    }

    render(){
        return(
            <div className="container" style={{paddingTop:'20px'}}>

                {/* <!--Body--> */}

                <main role="main" style={{marginTop:'10px'}}>

                    <section className="jumbotron text-center" >
                        <div className="container"
style={{backgroundColor:'#f9fbe7',marginTop:'-30px',marginBottom:'-
30px'}}>

                            <div className='row' >
                                <div className='col-md-4 bg-info text-
white text-center'>
                                    <div className="card-body" >
                                        <img src={logo} />
                                        <h2 className="py-
3">Registration</h2>
                                    </div>
                                </div>
                                {/* form section */}
                                {this.checkPending()}
                            </div>
                        </div>

                    </section>
                </main>

            </div>
        );
    }
}

export default Payment;
```

7.2 Back End

Authentication/server.js

```
// This is the client ID and client secret that you obtained
// while registering the application
const clientID = '<Confidential Client ID>'
const clientSecret = '<Confidential Client Secret>'

// Declare the redirect route
app.get('/home', (req, res) => {

  // The req.query contains the query params that were sent to this route.
  const requestToken = req.query.code

  axios({
    // Making a POST request
    method: 'post',

    // To Github auth with the client ID, secret and request token
    url:
`https://github.com/login/oauth/access_token?client_id=${clientID}&client_
secret=${clientSecret}&code=${requestToken}`,

    // Setting the content type header to receive JSON as response
    headers: {
         accept: 'application/json'
    }
  }).then((response) => {

    // Extracting access token from response body
    const accessToken = response.data.access_token

    // Redirecting to Home page along with the access token
    res.redirect(`http://localhost:3000/home/${accessToken}`)

  })
})

app.listen(4000,()=>{
    console.log("Authentication Server listening on port : 4000")
})
```

Government/server.js

```
const PORT = 4001;
const Router = express.Router();

Router.route('/verify').post(function(req,res){
    let idNum = req.body.id;

    idNum  = idNum.substring(0,idNum.length-1)
    console.log(idNum);

    if(idNum%2!=0)
        res.status(200).json({'valid':true})
    else
        res.status(200).json({'valid':false})

})

app.use('/government',Router);


app.listen(PORT,(err)=>{
    console.log("Government Server running on Port : ",PORT)
})
```

Payment/server.js

```
const PORT = 4002;
const Router = express.Router();

const Router = express.Router();
const paypal = require('paypal-rest-sdk');

Router.route('/pay').post(function(req,res){
    const reservation = req.body.reservation;
    console.log(reservation);

    // configuring paypal with the credentials
    paypal.configure({
        'mode': 'sandbox', //Specifying as Test
        'client_id': < Confidential ID > ', // confidential Client ID
        'client_secret': '< Confidential ID >'//confidential Client secret
    });
    // Creating the Payment Object
    var payment = {
        "intent": "authorize",
        "payer": {
            "payment_method": "paypal"
        },
        "redirect_urls": {
            "return_url":
"http://localhost:3000/home/"+reservation.token+"/success",
            "cancel_url": "http://localhost:4002/"
        },
        "transactions": [{
            "amount": {
                "total": reservation.netAmount,
                "currency": "USD"
            },
            "description": " a book on mean stack "
        }]
    }



    // Calling the Customer Payment method
    createPay( payment )
        .then( ( transaction ) => {
            var id = transaction.id;
            var links = transaction.links;
            var counter = links.length;
            while( counter -- ) {
                if ( links[counter].method == 'REDIRECT') {
                    res.send(links[counter].href)
                }
            }
        })
        .catch( ( err ) => {
            console.log( err );
        });


})
```

```
// Custom functions
var createPay = ( create_payment_json ) => {
    return new Promise( ( resolve , reject ) => {
        paypal.payment.create( create_payment_json , function( err ,
payment ) {
            if ( err ) {
                reject(err);
            }
            else {
                resolve(payment);
            }
        });
    });
}

app.use('/paypal',Router);


app.listen(PORT,(err)=>{
    console.log("Payment Server running on Port : ",PORT)
})
```

Email/server.js

```
const PORT = 4003;
Router.route('/confirm').post(function(req,res){
    const reservation =req.body.reservation;

    //Dependiencies needed for Email service
    const nodemailer = require('nodemailer');
    const ejs = require("ejs");
    const creds = require('./config/credential.js')

    //Creating transport instance
    var transport = {
        host: 'smtp.gmail.com',
        auth: {
        user: creds.USER,
        pass: creds.PASS
        }
    }
    //Creating a Nodemailer Transport instance
    var transporter = nodemailer.createTransport(transport)

    //Verifying the Nodemailer Transport instance
    transporter.verify((error, success) => {
        if (error) {
            console.log(error);
        } else {
            console.log('Server is ready to take messages');
        }
    });

    //Manipulating data to ejs mail template
    ejs.renderFile(__dirname+"/template/Hello.ejs",{ },function(err,data)
{
        if (err) {
            console.log(err);
        } else {
            var mainOptions = {
                from: '"FindMyTrip" findmytrip2017@gmail.com',
                to: reservation.email,
                subject: 'Train Ticket Receipt',
                html: data
            };

            transporter.sendMail(mainOptions, function (err, info) {
              if (err) {
                res.status(200).json({"MAIL":"Not Sent"})
              } else {
                res.status(200).json({"MAIL":"Successfully Sent"})
              }
          });
        }
    });

})

app.use('/email',Router);


app.listen(PORT,(err)=>{
    console.log("Email Server running on Port : ",PORT)
})
```

SMS/server.js

```
const PORT = 4004;
const Router = express.Router();

//For otp generation
const Speakeasy = require("speakeasy");

Router.route('/otp').post(function(req,res){

    const secret = req.body.secret;

    //Sending otp through SMS
    const accountSid = '<My Account ID>';
    const authToken = '<My Account Auth Token>';
    const client = require('twilio')(accountSid, authToken);

    const token = Speakeasy.totp({
        secret: secret,
        encoding: "base32",
        step: 60, // specified in seconds
        digits :4
    });

    client.messages
        .create({
            body: 'Your Loops Train Reservation 4 digit OTP :'+token,
            from: '+12029028295',
            to: '<Receiver's Number>'
        })
        .then(message => console.log(message.sid));

    res.send({
        "token": token,
        "remaining": (60 - Math.floor((new Date()).getTime() / 1000.0 %
60))
    });

})

Router.route('/validate').post(function(req,res){
    const pass = req.body.pass;
    console.log(pass.secret)
    console.log(pass.otp)
    res.send({
        "valid": Speakeasy.totp.verify({
            secret: pass.secret,
            encoding: "base32",
            token: pass.otp,
            step: 60,
            digits :4
        })
    });
})

app.use('/sms',Router);


app.listen(PORT,(err)=>{
    console.log("SMS Server running on Port : ",PORT)
})
```