



DS Assignment 2 - Train Ticket Reservation

Distributed Systems

Assignment 2

Online Train Ticket Reservation System

Assignment Report

Software Engineering Weekday Batch

IT17008006	Ranasinghe A.D.R.N.
------------	---------------------

1 Introduction	3
2 Architecture	4
3 Workflow (Frontdnd)	6
4 Appendix 11	
FrontEnd	12
BackEnd	31

Introduction

This report is Train Ticket Reservation System based on Rest API(backend), website(frontend) and MongoDB database. This is 2nd assignment of DS module. This is an online ticket reservation web application including payment. Passengers can pay using credit/debit or Dialog bill. After card payment is proceed successfully email will be sent to confirm that payment. In dialog bill passengers can pay bill in end of the month.

1.Website: This is a collection basic HTML + CSS + JavaScript and Bootstrap. And also JQuery library and HTTP requests to the REST service are used for backend.

2.REST API: This is implemented based on SOA principles by using Spring Boot with Java and tomcat server. That manifests a number of services.

The user should sign in to the services to buy tickets. After successful sign in, all trains will be displayed real-time. User can search and filter trains according to his/her preference. After the user is done selecting user can input no of tickets and class. Then tickets can be purchased using a credit card or a dialog number. To do payment using dialog services user should verify mobile number using pin code. After that all details will be displayed and add to the database. The user will also get an email about the confirmation.

Architecture

The backend is designed with the all of the functionality including set of a API request and responses. Because of that internal mechanisms can be modified without changing frontend too much. All of the functions are divided into 4 categories.

- 1.Tickets
- 2.Payment
- 3.User
- 4.Trains

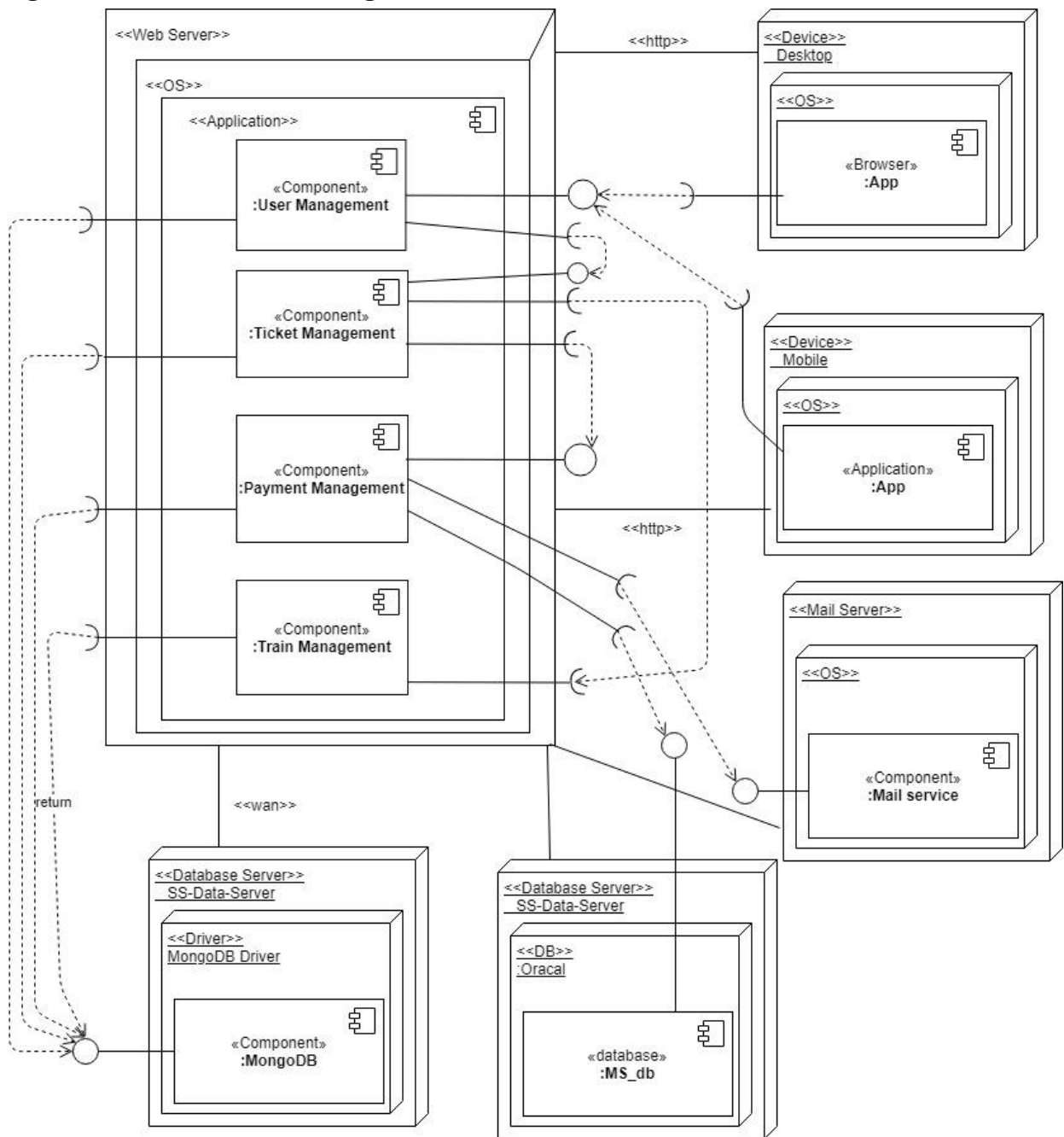
Those categories can run independently without the existence of others. It improves scalability and functional cohesiveness while decreasing coupling nature. But No SQL database is used by all of these services.

For designing the backend, API is used for functioning the major functions of the web service. The data is being passed through HTTP protocols and authentication key is used for API calls. JSON is used for communication between frontend, backend and MongoDB. The system consists of 3 major services. Each services provide set of sub functionalities. Clint can access these services using REST. JOSN is selected as communication medium because of some advanteages.it is light and weight, easy to coding in JavaScript which is the only language all browsers can understand quickly.

Backend

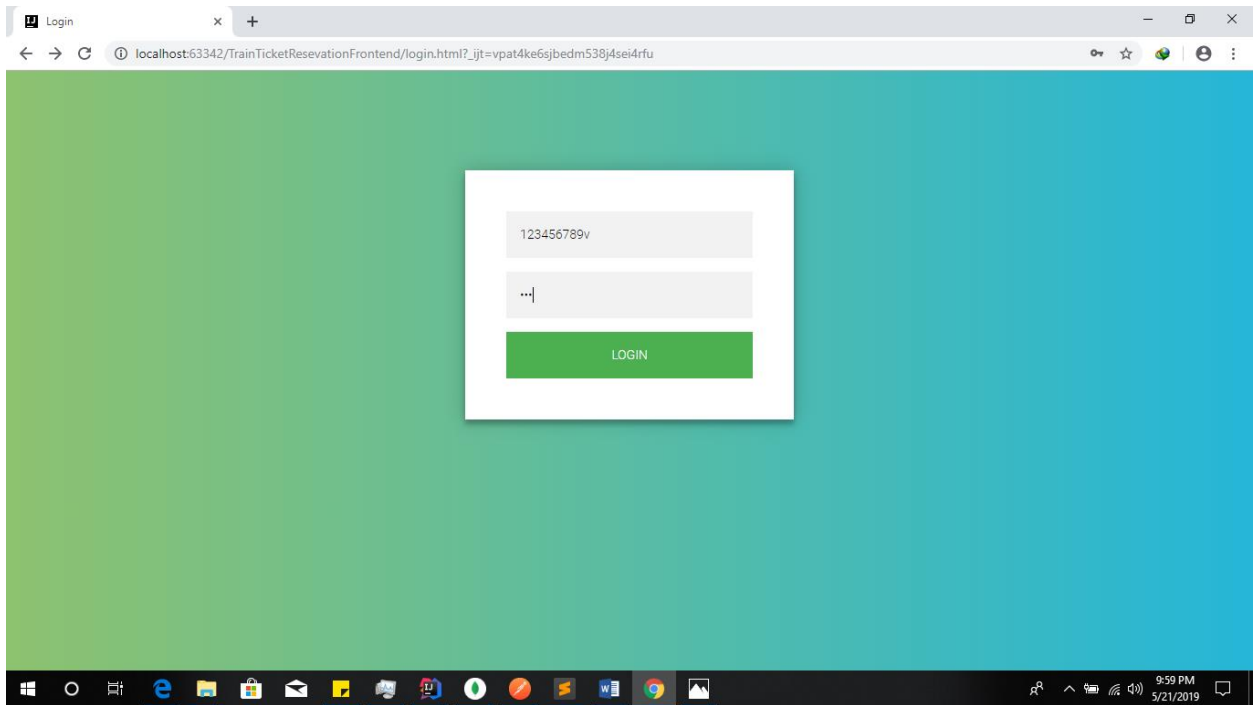
The controller class of each services are director point of all backend tasks. For example, Ticket services' all request is mapped for '/ticket'. That will be handled by TicketController class. Using URL controller will be called suitable method. For an Example, if GET request is called all the tickets list will be returned. Inside that process, TicketRepository is used to deal with MongoDB. Here ,the TicketRepository is the only interface which we do not implement but using MongoRepository interface extends it. That has many methods which was already defined. When some data is fetched from database then use model class to map that details of each.

High Level Architectural Diagram

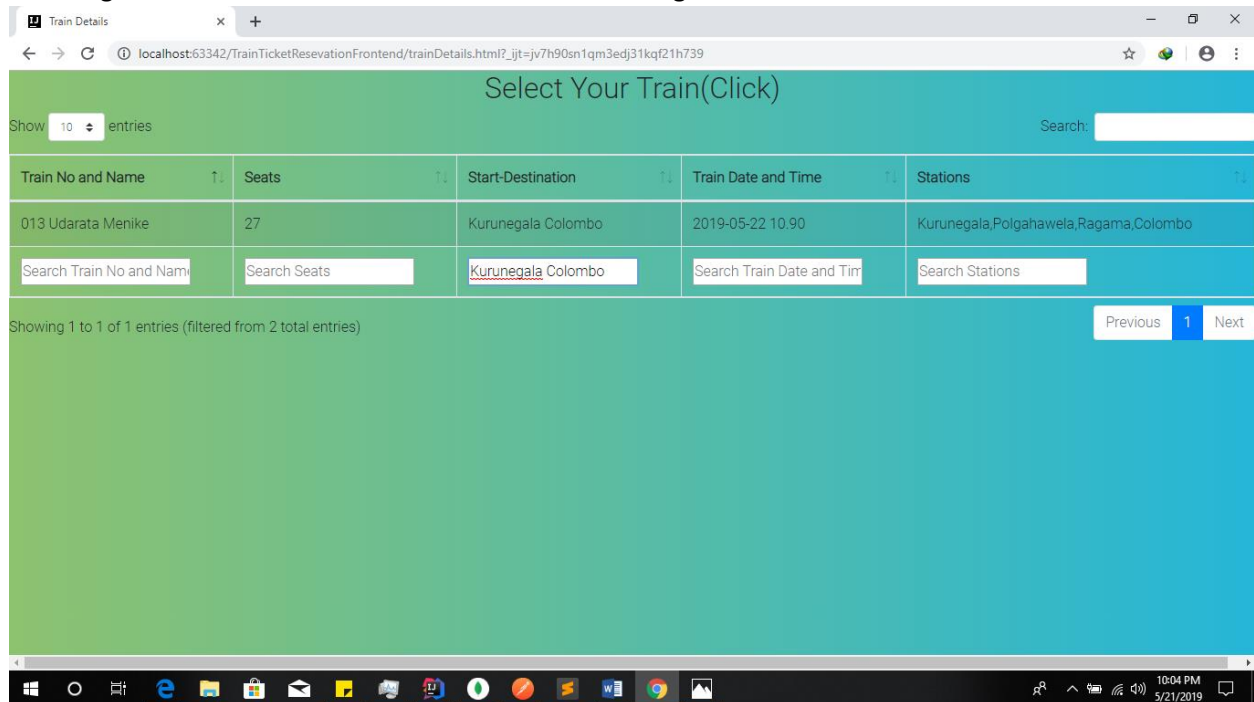


Workflow

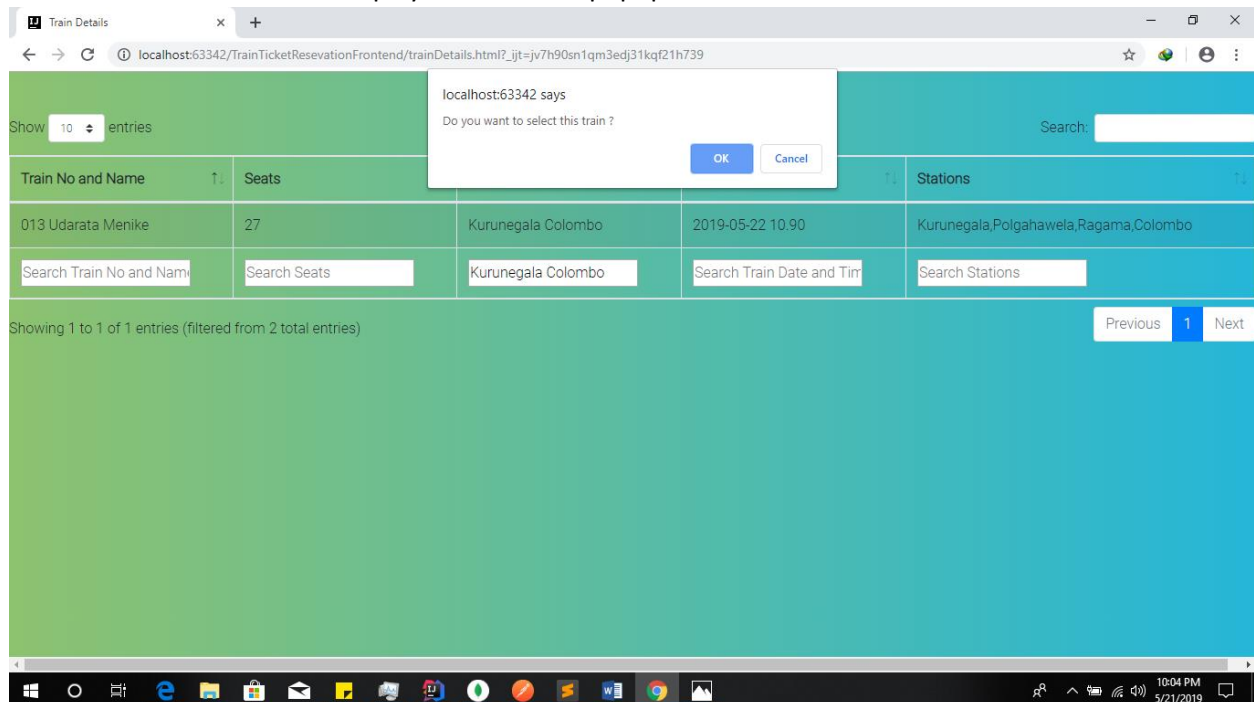
This is the login page. User should give his/her NIC number and password



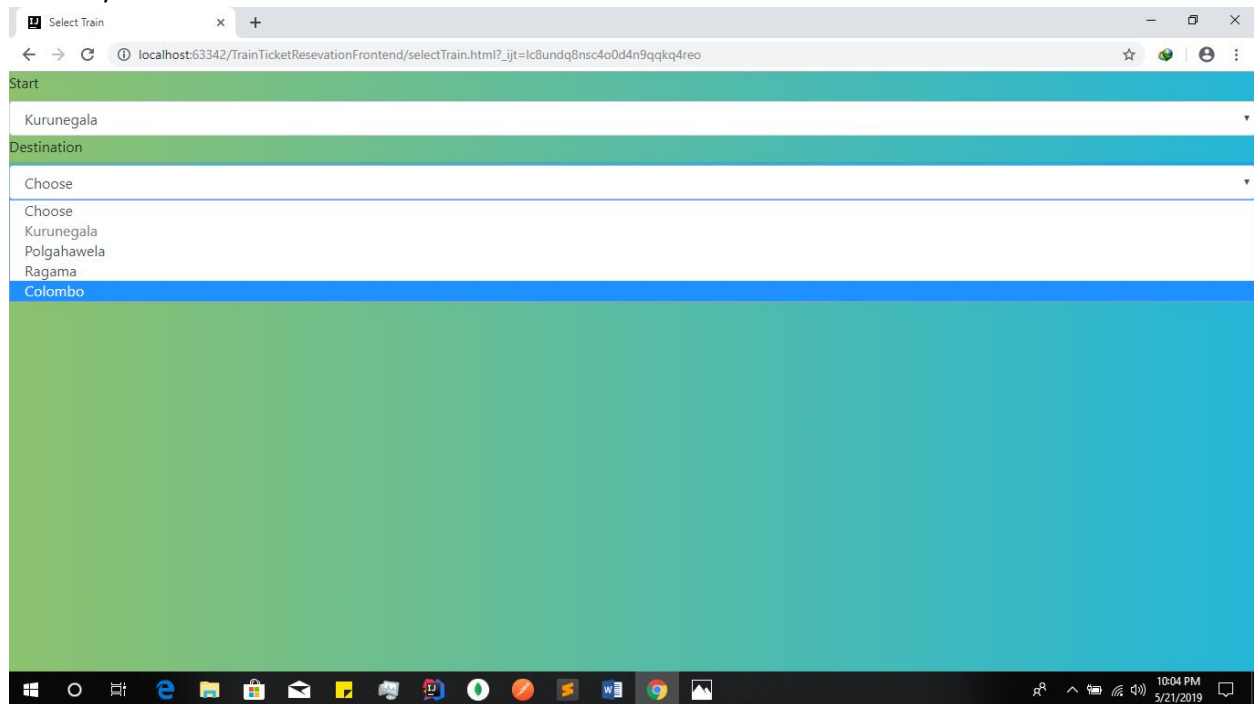
After give correct NIC and password trains details will have displayed. You can choose any train by checking stations. And also user can filter that data using search fields.



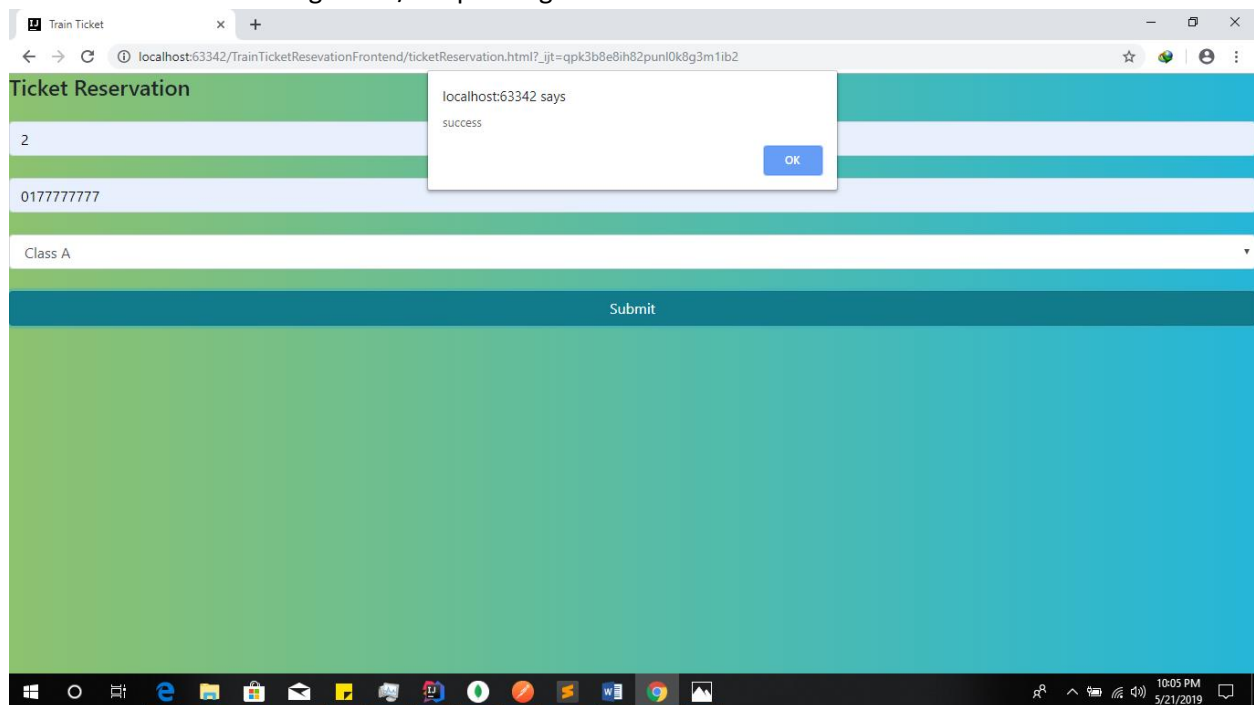
After select one train it will display confirmation popup box.



In here you can choose start station and destination. You cannot select reverse. It is disabled.



In this interface user can give his/her passenger details with class.



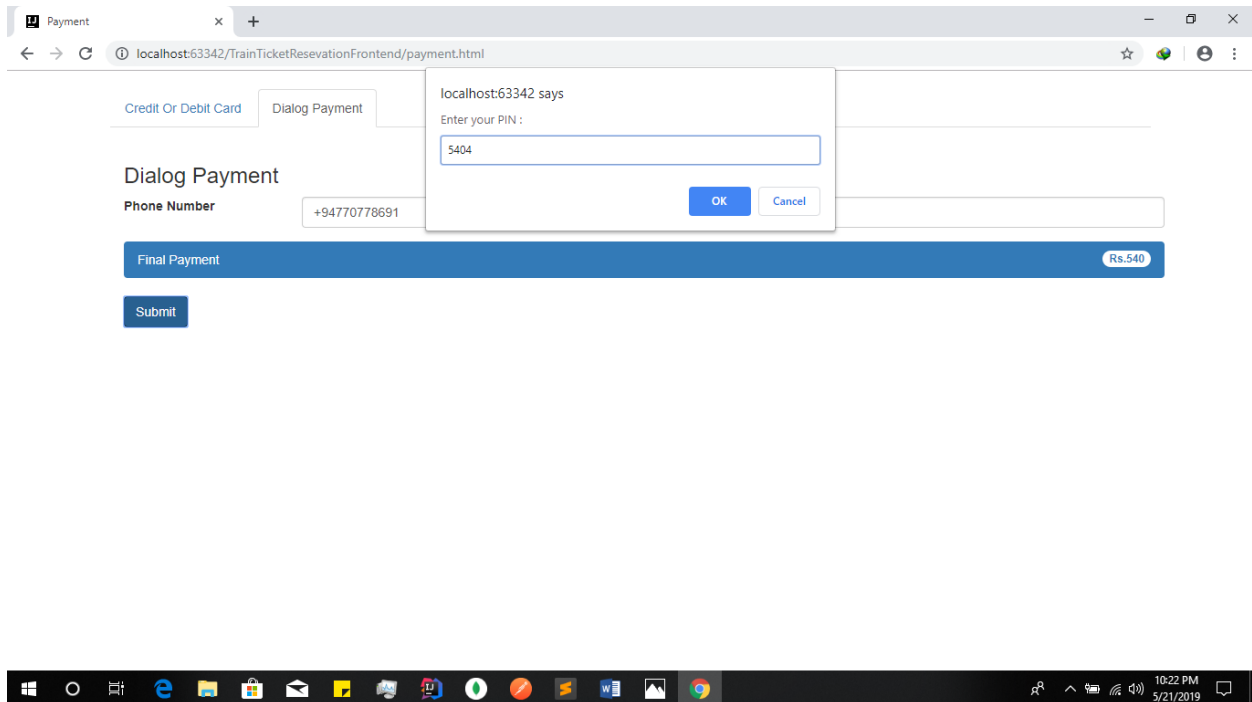
In here user can choose any type of payment method visa or Dialog service. It is validated using database values. If it is wrong user cannot proceed the process.

The screenshot shows a web browser window with the address bar displaying 'localhost:63342/TrainTicketResevationFrontend/payment.html'. The page has two tabs: 'Payment' and '+'. Below the tabs, there are two buttons: 'Credit Or Debit Card' and 'Dialog Payment'. A modal dialog box is open, displaying the text 'localhost:63342 says' and 'Payment Success' with an 'OK' button. Below the dialog, the 'Payment Details' form is visible. It contains the following fields: 'Full Name' (aaaaa), 'NIC' (123456789v), 'CARD NUMBER' (4321432112345678), 'EXPIRY DATE' (12/23), and 'CV CODE' (***). At the bottom of the form, there is a blue button labeled 'Final Payment' with 'Rs.810' next to it, and a green button labeled 'Submit'.

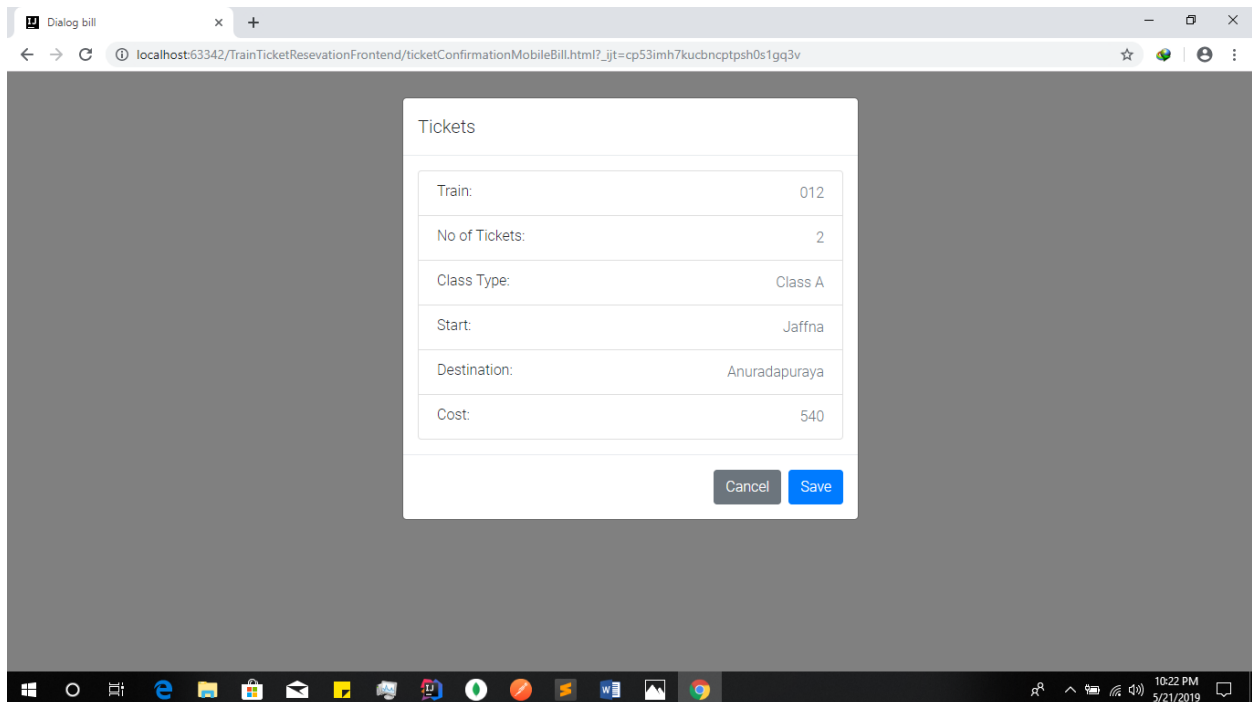
This is the dialog service. Mobile number is verified using pin code which is reserved. If pin number is invalid user cannot proceed the payment.

The screenshot shows the same web browser window, but the 'Dialog Payment' tab is selected. The 'Dialog Payment' form is displayed, featuring a 'Phone Number' field with the value '+94770778691'. Below this field, there is a blue button labeled 'Final Payment' with 'Rs.540' next to it, and a blue button labeled 'Submit'.

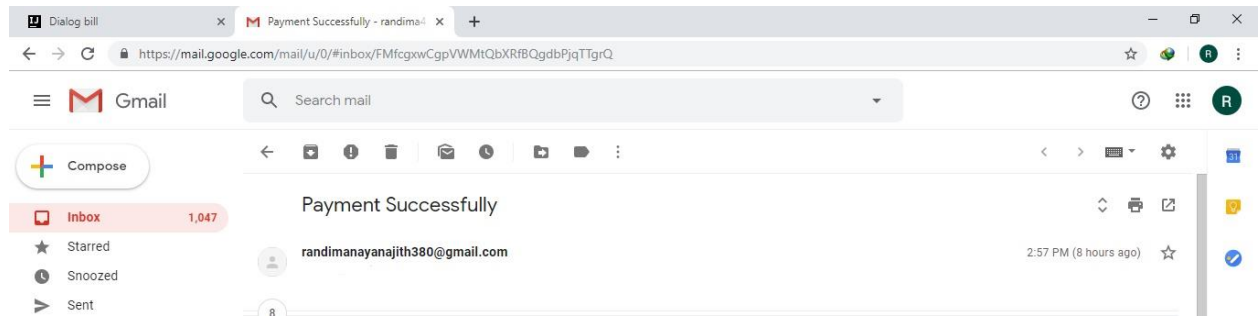
This is the place to type pin number. it is 4 digit number



After complete payment final tickets details will be displayed. After that you can submit.



If you submit that details email will be received.



Login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Login</title>

  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <link rel="stylesheet" type="text/css" href="style.css">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
```

```

integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/njGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
</head>
<body>
<div class="login-page">
  <div class="form">
    <form class="login-form" id="validateUser">
      <input type="text" id="nic" placeholder="Nic"/>
      <input type="password" id="userPassword" placeholder="password"/>
      <button type="submit">login</button>
    </form>
  </div>
</div>
<script src="validation.js"></script>
</body>
</html>

```

trainDetails.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Train Details</title>
</head>
<link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/css/bootstrap.css">
<link rel="stylesheet" type="text/css"
ref="https://cdn.datatables.net/1.10.19/css/dataTables.bootstrap4.min.css">
<link rel="stylesheet" type="text/css" href="style.css">
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
<body>
<h2 style="text-align: center;">Select Your Train(Click)</h2>
<table id="example2" class="table table-striped table-bordered" style="width:100%">
  <thead>
    <tr>
      <th>Train No and Name</th>
      <th>Seats</th>
      <th>Start-Destination</th>
      <th>Train Date and Time</th>
      <th>Stations</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th>Train No and Name</th>
      <th>Seats</th>
      <th>Start-Destination</th>
      <th>Train Date and Time</th>
      <th>Stations</th>
    </tr>
  </tfoot>
</table>
<script src="https://code.jquery.com/jquery-3.3.1.js"></script>
<script src="https://cdn.datatables.net/1.10.19/js/jquery.dataTables.min.js"></script>
<script

```

```

src="https://cdn.datatables.net/1.10.19/js/dataTables.bootstrap4.min.js"></script>
<script src="validation.js"></script>
<script>
    $.ajax({
        type: "GET",
        url: 'http://localhost:8080/train/',
        dataType: 'json',
        success: function (obj, textStatus) {
            $('#example2 tfoot th').each( function () {
                var title = $(this).text();
                $(this).html( '<input type="text" placeholder="Search '+title+'" />'
);
            } );
            var table = $('#example2').DataTable({
                data: obj,
                columns: [
                    {
                        data: null,
                        render: function ( data, type, row ) {
                            // Combine the first and last names into a single table
                            field
                            return data.trainNo+' '+data.trainName;
                        },
                        editField: ['trainNo', 'trainName']
                    },
                    { data: 'seats' },
                    {
                        data: null,
                        render: function ( data, type, row ) {
                            // Combine the first and last names into a single table
                            field
                            return data.start+' '+data.destination;
                        },
                        editField: ['start', 'destination']
                    },
                    {
                        data: null,
                        render: function ( data, type, row ) {
                            // Combine the first and last names into a single table
                            field
                            return data.trainDate+' '+data.startTime;
                        },
                        editField: ['trainDate', 'startTime']
                    },
                    { data: 'stopStations[].station' },
                ]
            });

            table.columns().every( function () {
                var that = this;

                $( 'input', this.footer() ).on( 'keyup change', function () {
                    if ( that.search() !== this.value ) {
                        that
                            .search( this.value )
                            .draw();
                    }
                } );
            } );

            $('#example2 tbody').on('click', 'tr', function () {

```

```

        var data = table.row( this ).data();

        if(confirm("Do you want to select this train ?")){
            console.log("pressed");
            localStorage["trainDetails"] =JSON.stringify(data);
            console.log(data._id);
            window.location.href =
"http://localhost:63342/TrainTicketResevationFrontend/selectTrain.html?_ijt=lc8undq8ns
c4o0d4n9qqkq4reo";
        }
    } );
},
error: function (obj, textstatus) {
    alert(obj.msg);
}
});
</script>
</body>
</html>

```

selectTrain.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Select Train</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="style.css">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
</script>
</script>
</head>
<body>
<form>
    <div class="form-group">
        <label for="startSelect">Start</label>
        <select class="form-control" id="startSelect">

```

```

        <option>Choose</option>
    </select>

    <label for="endSelect">Destination</label>
    <select class="form-control" id="endSelect">
        <option>Choose</option>
    </select>
</div>
<button type="button" class="btn btn-info"
onclick="lordTicketReservation()">Submit</button>
</form>
<script>
    var employees = JSON.parse(localStorage["trainDetails"]);
    console.log(employees);
    var selectStart = document.getElementById("startSelect");
    var selectEnd = document.getElementById("endSelect");
    var options = employees.stopStations;
    console.log(options);
    function getStationDetails(select){
        for(var i = 0; i < options.length; i++) {
            var opt = options[i].station;
            var el = document.createElement("option");
            el.textContent = opt;
            el.value = i;

            select.appendChild(el);
        }
    }
    getStationDetails(selectStart);
    getStationDetails(selectEnd);
    $('#startSelect').on('change', function (event) {
        var minVal = $('#startSelect').val();
        console.log(minVal);
        $('#endSelect option').each(function () {
            if($(this).val() <= minVal)
            {
                $(this).prop('disabled', true);
            }else{
                $(this).prop('disabled', false);
            }
        });
    });

    $('#endSelect').on('change', function (event) {
        var maxVal = $('#endSelect').val();
        $('#startSelect option').each(function () {
            if($(this).val() >= maxVal)
            {
                $(this).prop('disabled', true);
            }else{
                $(this).prop('disabled', false);
            }
        });
    });

    function lordTicketReservation(){
        console.log("Test ticket reservation");
        localStorage["trainStartPlace"] = $('#startSelect :selected').text();
        localStorage["trainDestinationPlace"] = $('#endSelect :selected').text();
        window.location.href =
"http://localhost:63342/TrainTicketResevationFrontend/ticketReservation.html?_ijt=qpk3
b8e8ih82punl0k8g3m1ib2";
    }

```



```

</script>
</body>
</html>

```

ticketReservation.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Train Ticket</title>

  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <link rel="stylesheet" type="text/css" href="style.css">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1cllHTMga3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaEfF/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

  <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
  <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/prettify/r298/prettify.min.js"></script>
  <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
</head>
<body>
  <p class="h4 mb-4">Ticket Reservation</p>
  <form id="selectReservation">
    <input type="text" id="noOfTickets" class="form-control" placeholder="No of
Tickets" pattern="(?!<=\\s|^)\\d+(?!<=\\s|$)" minlength="1" maxlength="3" required><br>
    <input type="text" id="phoneNumber" class="form-control" placeholder="Phone
number" pattern="(?!<=\\s|^)\\d+(?!<=\\s|$)" minlength="10" maxlength="10" required><br>
    <div class="form-group">
      <select class="form-control" id="selectedClass">
        <option>Class B</option>
        <option>Class A</option>
      </select>
    </div>
    <button class="btn btn-info my-4 btn-block" type="submit">Submit</button>
  </form>
  <script src="validation.js"></script>
</body>
</html>

```

Payment.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Payment</title>

  <link href="https://netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
  <script
src="https://netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/lodash@4.17.10/lodash.min.js"></script>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
  <script>
    var trainStartPlace = localStorage["trainStartPlace"];
    var trainDestinationPlace = localStorage["trainDestinationPlace"];
    var selectedTrain = JSON.parse(localStorage["trainDetails"]).stopStations;
    var ticketDetails = JSON.parse(localStorage["ticketDetails"]);
    var i = 0;
    var ticketPrice = 0;
    while(i<selectedTrain.length) {
      if (JSON.stringify(selectedTrain[i].station) ===
JSON.stringify(trainStartPlace)) {
        i++;
        break;
      }
      i++;
    }
    while(i<selectedTrain.length) {
      ticketPrice = ticketPrice + selectedTrain[i].price;
      if (JSON.stringify(selectedTrain[i].station) ===
JSON.stringify(trainDestinationPlace)) {
        break;
      }
      i++;
    }
    $(document).ready(function() {
      $("#myModal").modal({
        keyboard: false
      });
    });
  </script>
</head>
<body>
<div class="container" style="align-content: center">
  <br>
  <!-- Nav tabs -->
  <ul class="nav nav-tabs" role="tablist">
    <li class="nav-item">
      <a class="nav-link active" data-toggle="tab" href="#home">Credit Or Debit
Card</a>
    </li>
  </ul>
</div>
```

```

        <li class="nav-item">
            <a class="nav-link" data-toggle="tab" href="#menu1">Dialog Payment</a>
        </li>
    </ul>

    <!-- Tab panes -->
    <div class="tab-content">
        <div id="home" class="container tab-pane active" style="align-content:
center"><br>
            <h3 style="text-align: center">Credit Or Debit Card</h3>

            <div class="container">
                <form id="paymentForm">
                    <div class="row">
                        <div class="col-xs-12 col-md-4">
                            <div class="panel panel-default">
                                <div class="panel-heading">
                                    <h3 class="panel-title">
                                        Payment Details
                                    </h3>

                                </div>
                                <div class="panel-body">
                                    <form role="form">
                                        <div class="form-group">
                                            <label for="cardNumber">
                                                Full Name</label>
                                            <div class="input-group">
                                                <input type="text" class="form-
control" id="name" placeholder="Full Name" pattern="^[a-zA-Z ]+$" minlength="5"
required autofocus />
                                                <span class="input-group-addon"><span
class="glyphicon glyphicon-user"></span></span>
                                            </div>
                                        </div>

                                        <div class="form-group">
                                            <label for="cardNumber">
                                                NIC</label>
                                            <div class="input-group">
                                                <input type="text" class="form-
control" id="nic" placeholder="NIC" pattern="^[0-9]{9}[x|X|v|V]|[0-9]{12})$"
minlength="10" maxlength="10"
required autofocus />
                                                <span class="input-group-addon"><span
class="glyphicon glyphicon-ok"></span></span>
                                            </div>
                                        </div>

                                        <div class="form-group">
                                            <label for="cardNumber">
                                                CARD NUMBER</label>
                                            <div class="input-group">
                                                <input type="text" class="form-
control" id="cardNumber" placeholder="Valid Card Number (Ex:4*****)" pattern="^(?:4[0-
9]{12})(?:[0-9]{3})?" minlength="16" maxlength="16"
required autofocus />
                                                <span class="input-group-addon"><span
class="glyphicon glyphicon-lock"></span></span>
                                            </div>
                                        </div>
                                    </form>
                                </div>
                            </div>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>

```

```

<div class="col-xs-7 col-md-7">
  <div class="form-group">
    <label for="ex_Month">
      EXPIRY DATE</label>
    <div class="col-xs-6 col-lg-6 pl-
zero">
      <input type="text"
class="form-control" id="ex_Month" placeholder="MM" pattern="^[0-1]?[1-2]$"
minlength="2" maxlength="2" required />
    </div>
    <div class="col-xs-6 col-lg-6 pl-
zero">
      <input type="text"
class="form-control" id="ex_Year" placeholder="YY" pattern="^[1-2]?[9-5]$"
minlength="2" maxlength="2" required /></div>
    </div>
  </div>
  <div class="col-xs-5 col-md-5 pull-right">
    <div class="form-group">
      <label for="cvcCode">
        CV CODE</label>
      <input type="password"
class="form-control" id="cvcCode" placeholder="CV" pattern="(?!=\s|^)\d+(?!=\s|$)"
minlength="3" maxlength="3" required />
    </div>
  </div>
</div>
</form>
</div>
<div class="nav nav-pills nav-stacked">
  <li class="active"><a href="#"><span class="badge
pull-right">Rs.<span class="" id="ticketPriceSum"></span></span> Final Payment</a>
    </li>
</ul>
<br/>
  <button class="btn btn-success btn-lg btn-block"
id="payment-submit" type="submit">Submit</button>
</div>
</div>
</form>
</div>
<div id="menu1" class="container tab-pane fade"><br>
  <h3>Dialog Payment</h3>
  <form id="dialogPayment">
    <div class="form-group row">
      <label for="phoneNumber" class="col-sm-2 col-form-label">Phone
Number</label>
      <div class="col-sm-10">
        <input type="text" class="form-control" id="phoneNumber"
placeholder="Phone Number (+9477*****)" minlength="12" maxlength="12">
      </div>
    </div>
    <ul class="nav nav-pills nav-stacked">
      <li class="active"><a href="#"><span class="badge pull-
right">Rs.<span class="" id="ticketPriceSum2"></span></span> Final Payment</a>
    </li>
  </ul><br>

```

```

        <div class="form-group row">
            <div class="col-sm-10">
                <button type="button" class="btn btn-primary" onclick =
"dialogPaymentService();" data-toggle="modal" data-target="#exampleModal">
                    Submit
                </button>
            </div>
        </div>
    </form>
</div>
</div>
</div>
<script src="validation.js"></script>
<script>
    $('#myTab a').on('click', function (e) {
        e.preventDefault()
        $(this).tab('show')
    })

    if((JSON.stringify(ticketDetails.trainClass)) === (JSON.stringify("Class A"))){
        ticketPrice = ticketPrice*2*ticketDetails.noOfTickets;
    }else{
        ticketPrice = ticketPrice*ticketDetails.noOfTickets;
    }
    var discount = parseFloat(getUserDetails());
    console.log(discount);
    ticketPrice = ticketPrice * discount;
    localStorage["ticketPrice"] =JSON.stringify(ticketPrice);
    $('#ticketPriceSum').text(ticketPrice);
    $('#ticketPriceSum2').text(ticketPrice);
</script>
</body>
</html>

```

ticketConfirmationPage.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Ticket</title>
    <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/css/bootstrap.css">
    <link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.19/css/dataTables.bootstrap4.min.css">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>

    <link rel="stylesheet" type="text/css" href="style.css">
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></scri
pt>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

```

```

<script src="validation.js"></script>
<script>

    $(document).ready(function() {
        $("#myModal").modal({
            keyboard: false
        });
    });

    var trainStartPlace = localStorage["trainStartPlace"];
    var trainDestinationPlace = localStorage["trainDestinationPlace"];
    var paymentDetails = JSON.parse(localStorage["paymentDetails"]);
    var ticketDetails = JSON.parse(localStorage["ticketDetails"]);
    var ticketPrice = JSON.parse(localStorage["ticketPrice"]);
    var trainDetails = JSON.parse(localStorage["trainDetails"]);
    console.log(ticketDetails);
</script>
</head>
<body>
<div class="bs-example">
    <div id="myModal" class="modal fade" tabindex="-1">
        <div class="modal-dialog">

            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title">Tickets</h5>
                </div>
                <div class="modal-body">
                    <li class="list-group-item d-flex justify-content-between lh-
condensed">
                        <div>
                            <h6 class="my-0">Name:</h6>
                        </div>
                        <span class="text-muted" id="name"></span>
                    </li>
                    <li class="list-group-item d-flex justify-content-between lh-
condensed">
                        <div>
                            <h6 class="my-0">NIC:</h6>
                        </div>
                        <span class="text-muted" id="nic"></span>
                    </li>
                    <li class="list-group-item d-flex justify-content-between lh-
condensed">
                        <div>
                            <h6 class="my-0">Train:</h6>
                        </div>
                        <span class="text-muted" id="train"></span>
                    </li>
                    <li class="list-group-item d-flex justify-content-between lh-
condensed">
                        <div>
                            <h6 class="my-0">No of Tickets:</h6>
                        </div>
                        <span class="text-muted" id="noOfTickets"></span>
                    </li>
                    <li class="list-group-item d-flex justify-content-between lh-
condensed">
                        <div>
                            <h6 class="my-0">Class Type:</h6>
                        </div>
                        <span class="text-muted" id="clsType"></span>
                    </li>
                </div>
            </div>
        </div>
    </div>
</div>

```

```
<li class="list-group-item d-flex justify-content-between lh-condensed">  
    <div>  
        <h6 class="my-0">Start:</h6>  
    </div>  
    <span class="text-muted" id="start"></span>  
</li>  
<li class="list-group-item d-flex justify-content-between lh-condensed">  
    <div>  
        <h6 class="my-0">Destination:</h6>  
    </div>  
    <span class="text-muted" id="destination"></span>  
</li>  
<li class="list-group-item d-flex justify-content-between lh-condensed">  
    <div>  
        <h6 class="my-0">Cost:</h6>  
    </div>  
    <span class="text-muted" id="cost"></span>  
</li>  
</div>  
<div class="modal-footer">  
    <button type="button" class="btn btn-secondary" onclick=submitTicket() " data-dismiss="modal">Cancel</button>  
    <button type="button" class="btn btn-primary" onclick=createTrainReservation() ">Save</button>  
</div>  
</div>  
</div>  
</div>  
  
<script>  
    $("#name").text(paymentDetails.name);  
    $("#nic").text(paymentDetails.nic);  
    $("#noOfTickets").text(ticketDetails.noOfTickets);  
    $("#clsType").text(ticketDetails.selectedClass);  
    $("#start").text(trainStartPlace);  
    $("#destination").text(trainDestinationPlace);  
    $("#cost").text(ticketPrice);  
    $("#train").text(trainDetails.trainNo);  
    function submitTicket(){  
        //window.location.href =  
        "http://localhost:63342/TrainTicketResevationFrontend/trainDetails.html?_ijt=jv7h90snlqm3edj31kqf21h739";  
    }  
</script>  
</body>  
</html>
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <title>Dialog bill</title>  
</head>  
<body>
```

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Ticket</title>
  <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/css/bootstrap.css">
  <link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.19/css/dataTables.bootstrap4.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
  <link rel="stylesheet" type="text/css" href="style.css">
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></scri
pt>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
  <script src="validation.js"></script>
  <script>
    $(document).ready(function() {
      $("#myModal").modal({
        keyboard: false
      });
    });
    var trainStartPlace = localStorage["trainStartPlace"];
    var trainDestinationPlace = localStorage["trainDestinationPlace"];
    var ticketDetails = JSON.parse(localStorage["ticketDetails"]);
    var ticketPrice = JSON.parse(localStorage["ticketPrice"]);
    var trainDetails = JSON.parse(localStorage["trainDetails"]);
  </script>
</head>
<body>
<div class="bs-example">
  <div id="myModal" class="modal fade" tabindex="-1">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title">Tickets</h5>
        </div>
        <div class="modal-body">
          <li class="list-group-item d-flex justify-content-between lh-
condensed">
            <div>
              <h6 class="my-0">Train:</h6>
            </div>
            <span class="text-muted" id="train"></span>
          </li>
          <li class="list-group-item d-flex justify-content-between lh-
condensed">
            <div>
              <h6 class="my-0">No of Tickets:</h6>
            </div>
            <span class="text-muted" id="noOfTickets"></span>
          </li>
          <li class="list-group-item d-flex justify-content-between lh-
condensed">
            <div>
              <h6 class="my-0">Class Type:</h6>
            </div>
            <span class="text-muted" id="clsType"></span>
          </li>
          <li class="list-group-item d-flex justify-content-between lh-

```



```

condensed">
    <div>
        <h6 class="my-0">Start:</h6>
    </div>
    <span class="text-muted" id="start"></span>
</li>
<li class="list-group-item d-flex justify-content-between lh-
condensed">
    <div>
        <h6 class="my-0">Destination:</h6>
    </div>
    <span class="text-muted" id="destination"></span>
</li>
<li class="list-group-item d-flex justify-content-between lh-
condensed">
    <div>
        <h6 class="my-0">Cost:</h6>
    </div>
    <span class="text-muted" id="cost"></span>
</li>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary"
onclick="submitTicket()" data-dismiss="modal">Cancel</button>
    <button type="button" class="btn btn-primary"
onclick="sendEmail()">Save</button>
</div>
</div>
</div>
</div>

<script>
    $("#noOfTickets").text(ticketDetails.noOfTickets);
    $("#clsstype").text(ticketDetails.selectedClass);
    $("#start").text(trainStartPlace);
    $("#destination").text(trainDestinationPlace);
    $("#cost").text(ticketPrice);
    $("#train").text(trainDetails.trainNo);
    function submitTicket(){
        window.location.href =
"http://localhost:63342/TrainTicketResevationFrontend/trainDetails.html?_ijt=jv7h90sn1
qm3edj31kqf21h739";
    }
</script>
</body>
</html>
</body>
</html>

```

style.css

```

@import url(https://fonts.googleapis.com/css?family=Roboto:300);
.login-page {
    width: 360px;
    padding: 8% 0 0;
    margin: auto;
}

```

```

.form {
  position: relative;
  z-index: 1;
  background: #FFFFFF;
  max-width: 660px;
  margin: 0 auto 100px;
  padding: 45px;
  text-align: center;
  box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}
.form input {
  font-family: "Roboto", sans-serif;
  outline: 0;
  background: #f2f2f2;
  width: 100%;
  border: 0;
  margin: 0 0 15px;
  padding: 15px;
  box-sizing: border-box;
  font-size: 14px;
}
.form button {
  font-family: "Roboto", sans-serif;
  text-transform: uppercase;
  outline: 0;
  background: #4CAF50;
  width: 100%;
  border: 0;
  padding: 15px;
  color: #FFFFFF;
  font-size: 14px;
  -webkit-transition: all 0.3 ease;
  transition: all 0.3 ease;
  cursor: pointer;
}
.form button:hover, .form button:active, .form button:focus {
  background: #43A047;
}
.form .message {
  margin: 15px 0 0;
  color: #b3b3b3;
  font-size: 12px;
}
.form .message a {
  color: #4CAF50;
  text-decoration: none;
}
.form .register-form {
  display: none;
}
.container {
  position: relative;
  z-index: 1;
  max-width: 300px;
  margin: 0 auto;
}
.container:before, .container:after {
  content: "";
  display: block;
  clear: both;
}
.container .info {
  margin: 50px auto;
}

```

```

        text-align: center;
    }
    .container .info h1 {
        margin: 0 0 15px;
        padding: 0;
        font-size: 36px;
        font-weight: 300;
        color: #1a1a1a;
    }
    .container .info span {
        color: #4d4d4d;
        font-size: 12px;
    }
    .container .info span a {
        color: #000000;
        text-decoration: none;
    }
    .container .info span .fa {
        color: #EF3B3A;
    }
    body {
        background: #76b852; /* fallback for old browsers */
        background: -webkit-linear-gradient(right, #76b852, #8DC26F);
        background: -moz-linear-gradient(right, #76b852, #8DC26F);
        background: -o-linear-gradient(right, #76b852, #8DC26F);
        background: linear-gradient(to left, #25b6d8, #8DC26F);
        font-family: "Roboto", sans-serif;
        -webkit-font-smoothing: antialiased;
        -moz-osx-font-smoothing: grayscale;
    }

```

Validation.js

```

function getUserDetails(){
    var nic;
    var databaseNic;
    $.ajax({
        url: 'http://localhost:8080/card/' + $('#nic').val(),
        type: 'get',
        contentType: 'application/json; charset=utf-8',
        dataType: 'json',
        success: function(data){
            nic= $('#nic').val();
            databaseNic = data.nic;
            console.log(data);
            console.log(nic);
        },
        error: function(){
            return 1;
        }
    });
    if(!_isEqual(JSON.stringify(nic), JSON.stringify(databaseNic))){
        return 0.9;
    }else{
        return 1;
    }
}
var updateTrainDetails =function (){
    var ticketDetails = JSON.parse(localStorage["ticketDetails"]);
    var trainDetails = JSON.parse(localStorage["trainDetails"]);

```

```

$.ajax({
    url: 'http://localhost:8080/train/'+trainDetails.trainNo,
    type: 'put',
    data: JSON.stringify({
        "trainNo":trainDetails.trainNo,
        "trainName":trainDetails.trainName,
        "seats":(trainDetails.seats-ticketDetails.noOfTickets),
        "start":trainDetails.start,
        "destination":trainDetails.destination,
        "trainDate":trainDetails.trainDate,
        "startTime":trainDetails.startTime,

"stopStations": [{ "station":trainDetails.stopStations[0].station, "price":trainDetails.s
topStations[0].price}, {"station":trainDetails.stopStations[1].station, "price":trainDet
ails.stopStations[1].price}, {"station":trainDetails.stopStations[2].station, "price":tr
ainDetails.stopStations[2].price}, {"station":trainDetails.stopStations[3].station, "pri
ce":trainDetails.stopStations[3].price}]
    }),
    contentType: 'application/json; charset=utf-8',
    success: function(data){
        console.log(trainDetails.seats-ticketDetails.noOfTickets);
    },
    error: function(){
        console.log("fail");
    }
});
}

function createTrainReservation(){
    var trainStartPlace = localStorage["trainStartPlace"];
    var trainDestinationPlace = localStorage["trainDestinationPlace"];
    var paymentDetails = JSON.parse(localStorage["paymentDetails"]);
    var ticketDetails = JSON.parse(localStorage["ticketDetails"]);
    var ticketPrice = JSON.parse(localStorage["ticketPrice"]);
    var trainDetails = JSON.parse(localStorage["trainDetails"]);
    $.ajax({
        url: 'http://localhost:8080/ticket/',
        type: 'post',
        data: JSON.stringify({
            "nic": paymentDetails.nic,
            "name": paymentDetails.name,
            "train": trainDetails.trainNo,
            "noOfTickets": ticketDetails.noOfTickets,
            "classType": ticketDetails.selectedClass,
            "start": trainStartPlace,
            "destination": trainDestinationPlace,
            "cost": ticketPrice
        }),
        contentType: 'application/json; charset=utf-8',
        success: function(data){
            console.log("Test success");
            updateTrainDetails.call();
            alert('Success');
            sendEmail();
            localStorage.clear();
            window.location.href =
"http://localhost:63342/TrainTicketResevationFrontend/trainDetails.html?_ijt=jv7h90sn1
qm3edj31kqf21h739";

        },
        error: function(){
            alert('error');
            localStorage.clear();
            window.location.href =

```

```

"http://localhost:63342/TrainTicketResevationFrontend/trainDetails.html?_ijt=jv7h90sn1
qm3edj31kqf21h739";
    }
    });
}
function dialogPaymentService( e ){
    $.ajax({
        url: 'http://localhost:8080/sms/'+ $("#phoneNumber").val(),
        type: 'get',
        contentType: 'application/json; charset=utf-8',
        dataType: 'json',
        success: function(data){
            console.log(data);
            var pin =prompt("Enter your PIN :");
            console.log(pin)
            if(JSON.stringify(pin).localeCompare(JSON.stringify(data))) {
                window.location.href =
'http://localhost:63342/TrainTicketResevationFrontend/ticketConfirmationMobileBill.htm
l?_ijt=cp53imh7kucbncptsh0s1gq3v';
            }else{
                alert("Invalid Pin");
            }
        },
        error: function(){
            alert('Invalid Mobile Number');
        }
    });
    e.preventDefault();
}
function sendEmail(){
    var user = JSON.parse(localStorage["userDetails"]);
    console.log("test");
    $.ajax({
        url: 'http://localhost:8080/mail/'+user.email,
        type: 'get',
        contentType: 'application/json; charset=utf-8',
        dataType: 'json',
        success: function(data){
            console.log("test");
            alert("Success. Confirmation email will be received")
            window.location.href =
'http://localhost:63342/TrainTicketResevationFrontend/trainDetails.html?_ijt=cp53imh7k
ucbncptsh0s1gq3v';
        },
        error: function(){
            alert('');
        }
    });
    localStorage.clear();
}
(function($){
    function redirectToPayment(e){
        console.log("test")
        var data={
            "noOfTickets": $('#noOfTickets').val(),
            "phoneNumber": $('#phoneNumber').val(),
            "selectedClass": $('#selectedClass :selected').text()
        };
        console.log(data);
        localStorage["ticketDetails"] = JSON.stringify(data);
        window.location.href = 'payment.html';
        alert("success");
    }
}

```

```

        e.preventDefault();
    }

    function insertPaymentDetails( e ){
        console.log($("#cardNumber").val());
        console.log("Test amount");
        console.log($("#ticketPriceSum").text());
        $.ajax({
            url: 'http://localhost:8080/card/' + $("#cardNumber").val(),
            type: 'get',
            contentType: 'application/json; charset=utf-8',
            dataType: 'json',
            success: function(data) {

                var insertedPaymentDetails={
                    "name": $('#name').val(),
                    "nic": $('#nic').val(),
                    "cardNumber": $('#cardNumber').val(),
                    "expireMonth": $('#ex_Month').val(),
                    "expireYear": $('#ex_Year').val(),
                    "cvcCode": parseFloat($('#cvcCode').val())
                }
                console.log("database data");
                var databaseData = {
                    "name": data.name,
                    "nic": data.nic,
                    "cardNumber": data.cardNumber,
                    "expireMonth": data.expireMonth,
                    "expireYear": data.expireYear,
                    "cvcCode": data.cvcCode
                }

                console.log(insertedPaymentDetails);
                console.log(databaseData);

                if(_.isEqual(JSON.stringify(insertedPaymentDetails),JSON.stringify(databaseData))){
                    if($("#ticketPriceSum").text()<=data.amount){
                        localStorage["paymentDetails"] =JSON.stringify(databaseData);
                        window.location.href = 'ticketConfirmationPage.html';
                        alert('Payment Success');
                    }else{
                        alert('Low Balance');
                    }
                }else{
                    console.log("Payment error Re-enter!");
                }
            },
            error: function(){
                alert('Payment error');
            }
        });

        e.preventDefault();
    }

    function validate( e ){
        $.ajax({
            url: 'http://localhost:8080/user/' + $("#nic").val(),
            type: 'get',
            contentType: 'application/json; charset=utf-8',
            dataType: 'json',
            success: function(data) {
                console.log(data);
                localStorage["userDetails"] =JSON.stringify(data);
            }
        });
    }

```

```

        if (data.password === ($("#userPassword").val())) {
            window.location.href =
"http://localhost:63342/TrainTicketResevationFrontend/trainDetails.html?_ijt=jv7h90sn1
qm3edj31kqf21h739";
        } else {
            window.location.href =
"http://localhost:63342/TrainTicketResevationFrontend/login.html?_ijt=vpat4ke6sjbedm53
8j4sei4rfu";
        }
    },
    error: function() {
        alert('error');
    }
});

e.preventDefault();
}
$("#validateUser").submit(validate);
$("#selectReservation").submit(redirectToPayment);
$("#paymentForm").submit(insertPaymentDetails);
})(jQuery);

```

Backend(Controllers)

CardController.class

```

package com.example.ticket.controllers;

import com.example.ticket.Config;
import com.example.ticket.Validations;
import com.example.ticket.model.Card;
import com.example.ticket.model.Ticket;
import com.example.ticket.repository.CardRepository;
import com.example.ticket.repository.TicketRepository;
import org.bson.types.ObjectId;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import javax.validation.Valid;

@RestController
@RequestMapping("card")
@CrossOrigin(origins = Config.a_Origin)
public class CardController {

    @Autowired
    private CardRepository cardRepository;
    Validations validate;

    @RequestMapping(value =("/{cardNumber}", method = RequestMethod.GET)

```

```

    public Card getCardById(@PathVariable("cardNumber") String _id) {

        return cardRepository.findBycardNumber(_id);
    }

    @RequestMapping(value = "/", method = RequestMethod.POST)
    public Card createCard(@Valid @RequestBody Card card) {
        System.out.println(card.getNic());
        cardRepository.save(card);
        System.out.println("POST TEST IN SPRING BOOT"+card.toString());
        return card;
    }

}

```

SimpleMailController.class

```

package com.example.ticket.controllers;

import javax.mail.MessagingException;
import javax.mail.internet.MimeMessage;
import javax.validation.Valid;

import com.example.ticket.Config;
import com.example.ticket.model.Card;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.mail.javamail.MimeMessageHelper;
import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("mail")
@CrossOrigin(origins = Config.a_Origin)
public class SimpleMailController {

    @Autowired
    private JavaMailSender sender;

    @RequestMapping(value = "/{email}", method = RequestMethod.GET)
    public String sendMail(@PathVariable("email") String email) {
        MimeMessage message = sender.createMimeMessage();
        MimeMessageHelper helper = new MimeMessageHelper(message);

        try {
            helper.setTo(email);
            helper.setText("Pin Number:");
            helper.setSubject("Payment Successfully");
        } catch (MessagingException e) {
            e.printStackTrace();
            return "Error while sending mail ..";
        }
        sender.send(message);

        return "success";
    }

}

```


sms.class

```
package com.example.ticket.controllers;

import com.example.ticket.Config;
import com.example.ticket.model.Card;
import com.twilio.Twilio;
import com.twilio.rest.api.v2010.account.Message;
import com.twilio.type.PhoneNumber;
import org.springframework.boot.ApplicationArguments;
import org.springframework.boot.ApplicationRunner;
import org.springframework.web.bind.annotation.*;

import javax.validation.Valid;

@RestController
@RequestMapping("sms")
@CrossOrigin(origins = Config.a_Origin)
public class SMS {
    private final static String ACCOUNT_SID = "ACd8f1640006e766095a43ca1e6f4312b8";
    private final static String AUTH_ID = "b2572f7f6f6a9987fb08ec0d413a8176";
    static {
        Twilio.init(ACCOUNT_SID, AUTH_ID);
    }

    @RequestMapping(value = "/{smsNumber}", method = RequestMethod.GET)
    public String sendSms(@PathVariable("smsNumber") String smsNumber) throws
Exception {
        String pinNumber = generateRandomNumber();
        Message.creator(new PhoneNumber(smsNumber), new PhoneNumber("+12242078592"),
            "Your Verification Code is: "+pinNumber).create();
        return pinNumber;
    }

    private String generateRandomNumber() {
        int randomPIN = (int) (Math.random()*9000)+1000;
        String pin = String.valueOf(randomPIN);
        return pin;
    }
}
```

TicketController.class

```
package com.example.ticket.controllers;

import com.example.ticket.Config;
import com.example.ticket.model.Ticket;
import com.example.ticket.repository.TicketRepository;
```

```

import org.bson.types.ObjectId;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import javax.validation.Valid;
import java.util.List;

@RestController
@RequestMapping("ticket")
@CrossOrigin(origins = Config.a_Origin)
public class TicketController {

    @Autowired
    private TicketRepository ticketRepository;

    @RequestMapping(value = "/", method = RequestMethod.GET)
    public List<Ticket> getAllTicket() {

        return ticketRepository.findAll();
    }

    @RequestMapping(value = "/{_id}", method = RequestMethod.GET)
    public Ticket getTicketById(@PathVariable("_id") ObjectId ticketId) {

        return ticketRepository.findBy_id(ticketId);
    }

    @RequestMapping(value = "/", method = RequestMethod.POST)
    public Ticket createTicket(@Valid @RequestBody Ticket ticket) {
        ticket.set_id(ObjectId.get());
        ticketRepository.save(ticket);
        System.out.println("POST TEST IN SPRING BOOT"+ticket.toString());

        return ticket;
    }

    @RequestMapping(value = "/{_id}", method = RequestMethod.PUT)
    public void modifyTicketById(@PathVariable("_id") ObjectId _id, @Valid
    @RequestBody Ticket ticket) {
        ticket.set_id(_id);
        ticketRepository.save(ticket);
    }

    @RequestMapping(value = "/{_id}", method = RequestMethod.DELETE)
    public void deleteTicket(@PathVariable ObjectId _id) {

        ticketRepository.delete(ticketRepository.findBy_id(_id));
    }
}

```

TrainController.class

```

package com.example.ticket.controllers;

import com.example.ticket.Config;
import com.example.ticket.model.Ticket;
import com.example.ticket.model.Train;

```

```

import com.example.ticket.model.User;
import com.example.ticket.repository.TrainRepository;
import com.example.ticket.repository.UserRepository;
import org.bson.types.ObjectId;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import javax.validation.Valid;
import java.util.List;

@RestController
@RequestMapping("train")
@CrossOrigin(origins = Config.a_Origin)
public class TrainController {

    @Autowired
    private TrainRepository repository;

    @RequestMapping(value = "/", method = RequestMethod.GET)
    public List<Train> getAllTrains() {

        return repository.findAll();
    }

    @RequestMapping(value = "/{_id}", method = RequestMethod.PUT)
    public Train modifyTrain(@PathVariable("_id") String trainNo, @Valid @RequestBody
Train train) {
        Train train1 = repository.findById(trainNo).orElseThrow(() -> new
ResourceNotFoundException());

        train1.setTrainName(train.getTrainName());
        train1.setSeats(train.getSeats());
        train1.setStart(train.getStart());
        train1.setDestination(train.getDestination());
        train1.setTrainDate(train.getTrainDate());
        train1.setStartTime(train.getStartTime());
        train1.setStopStations(train.getStopStations());

        return repository.save(train1);
    }

    @RequestMapping(value = "/", method = RequestMethod.POST)
    public Train createTrain(@Valid @RequestBody Train tr) {

        repository.save(tr);
        return tr;
    }

    private class ResourceNotFoundException extends RuntimeException {
        public ResourceNotFoundException() {
        }
    }
}

```

UserController.class

```

package com.example.ticket.controllers;

import com.example.ticket.Config;
import com.example.ticket.model.User;
import com.example.ticket.repository.UserRepository;
import org.bson.types.ObjectId;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import javax.validation.Valid;
import java.util.List;
@RestController
@RequestMapping("user")
@CrossOrigin(origins = Config.a_Origin)
public class UserController {

    @Autowired
    private UserRepository repository;

    @RequestMapping(value = "/", method = RequestMethod.GET)
    public List<User> getAllUsers() {

        return repository.findAll();
    }

    @RequestMapping(value =("/{nic}", method = RequestMethod.GET)
    public User getUserById(@PathVariable("nic") String nic) {

        return repository.findBynic(nic);
    }

    @RequestMapping(value = "/", method = RequestMethod.POST)
    public User createUser(@Valid @RequestBody User us) {
        repository.save(us);
        return us;
    }

    @RequestMapping(value =("/{id}", method = RequestMethod.PUT)
    public void modifyUserById(@PathVariable("id") ObjectId id, @Valid @RequestBody
User us) {

        repository.save(us);
    }

    @RequestMapping(value =("/{nic}", method = RequestMethod.DELETE)
    public void deleteUser(@PathVariable String nic) {

        repository.delete(repository.findBynic(nic));
    }
}

```

Models

Card.class

```

package com.example.ticket.model;

import org.springframework.data.annotation.Id;

import javax.validation.constraints.NotEmpty;

public class Card {
    @Id
    private String cardNumber;
    @NotEmpty
    private String name;
    @NotEmpty
    private String nic;
    @NotEmpty
    private String expireMonth;
    @NotEmpty
    private String expireYear;
    @NotEmpty
    private int cvcCode;

    private double amount;

    public Card(String cardNumber, String name, String nic, String expireMonth, String
expireYear, int cvcCode, double amount) {
        this.cardNumber = cardNumber;
        this.name = name;
        this.nic = nic;
        this.expireMonth = expireMonth;
        this.expireYear = expireYear;
        this.cvcCode = cvcCode;
        this.amount = amount;
    }

    public String getCardNumber() {
        return cardNumber;
    }

    public void setCardNumber(String cardNumber) {
        this.cardNumber = cardNumber;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getNic() {
        return nic;
    }

    public void setNic(String nic) {
        this.nic = nic;
    }

    public String getExpireMonth() {
        return expireMonth;
    }

    public void setExpireMonth(String expireMonth) {
        this.expireMonth = expireMonth;
    }

```

```

    }

    public String getExpireYear() {
        return expireYear;
    }

    public void setExpireYear(String expireYear) {
        this.expireYear = expireYear;
    }

    public int getCvcCode() {
        return cvcCode;
    }

    public void setCvcCode(int cvcCode) {
        this.cvcCode = cvcCode;
    }

    public double getAmount() {
        return amount;
    }

    public void setAmount(double amount) {
        this.amount = amount;
    }
}

```

Station.class

```

package com.example.ticket.model;

import javax.validation.constraints.NotEmpty;

public class Station {
    @NotEmpty
    private String station;
    @NotEmpty
    private double price;

    public Station(String station, double price) {
        this.station = station;
        this.price = price;
    }

    public String getStation() {
        return station;
    }

    public void setStation(String station) {
        this.station = station;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }
}

```

Ticket.class

```
package com.example.ticket.model;

import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;

import javax.validation.constraints.NotEmpty;

public class Ticket {
    @Id
    private ObjectId _id;

    private String nic;

    private String name;

    private String train;

    private int noOfTickets;

    private String classType;

    private String start;

    private String destination;

    private double cost;

    public Ticket(ObjectId _id, String nic, String name, String train, int
noOfTickets, String classType, String start, String destination, double cost) {
        this._id = _id;
        this.nic = nic;
        this.name = name;
        this.train = train;
        this.noOfTickets = noOfTickets;
        this.classType = classType;
        this.start = start;
        this.destination = destination;
        this.cost = cost;
    }

    public ObjectId get_id() {
        return _id;
    }

    public void set_id(ObjectId _id) {
        this._id = _id;
    }

    public String getNic() {
        return nic;
    }

    public void setNic(String nic) {
        this.nic = nic;
    }

    public String getName() {
```

```

        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getTrain() {
        return train;
    }

    public void setTrain(String train) {
        this.train = train;
    }

    public int getNoOfTickets() {
        return noOfTickets;
    }

    public void setNoOfTickets(int noOfTickets) {
        this.noOfTickets = noOfTickets;
    }

    public String getClassType() {
        return classType;
    }

    public void setClassType(String classType) {
        this.classType = classType;
    }

    public String getStart() {
        return start;
    }

    public void setStart(String start) {
        this.start = start;
    }

    public String getDestination() {
        return destination;
    }

    public void setDestination(String destination) {
        this.destination = destination;
    }

    public double getCost() {
        return cost;
    }

    public void setCost(double cost) {
        this.cost = cost;
    }
}

```

Train.class


```

package com.example.ticket.model;

import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;

import javax.persistence.GeneratedValue;
import javax.validation.constraints.NotEmpty;
import java.util.List;

public class Train {
    /*      @Id
           @GeneratedValue
           private String _id;*/
    @Id
    private String trainNo;
    @NotEmpty
    private String trainName;
    @NotEmpty
    private int seats;
    @NotEmpty
    private String start;
    @NotEmpty
    private String destination;
    @NotEmpty
    private String trainDate;
    @NotEmpty
    private String startTime;
    private List<Station> stopStations;

    public Train(String trainNo, String trainName, int seats, String start, String
destination, String trainDate, String startTime, List<Station> stopStations) {

        this.trainNo = trainNo;
        this.trainName = trainName;
        this.seats = seats;
        this.start = start;
        this.destination = destination;
        this.trainDate = trainDate;
        this.startTime = startTime;
        this.stopStations = stopStations;
    }

    public String getTrainNo() {
        return trainNo;
    }

    public void setTrainNo(String trainNo) {
        this.trainNo = trainNo;
    }

    public String getTrainName() {
        return trainName;
    }

    public void setTrainName(String trainName) {
        this.trainName = trainName;
    }

    public int getSeats() {
        return seats;
    }
}

```

```

    public void setSeats(int seats) {
        this.seats = seats;
    }

    public String getStart() {
        return start;
    }

    public void setStart(String start) {
        this.start = start;
    }

    public String getDestination() {
        return destination;
    }

    public void setDestination(String destination) {
        this.destination = destination;
    }

    public String getTrainDate() {
        return trainDate;
    }

    public void setTrainDate(String trainDate) {
        this.trainDate = trainDate;
    }

    public String getStartTime() {
        return startTime;
    }

    public void setStartTime(String startTime) {
        this.startTime = startTime;
    }

    public List<Station> getStopStations() {
        return stopStations;
    }

    public void setStopStations(List<Station> stopStations) {
        this.stopStations = stopStations;
    }
}

```

User.class

```

package com.example.ticket.model;

import org.bson.types.ObjectId;
import org.springframework.data.annotation.Id;

public class User {
    @Id
    private String nic;
    private String firstname;
    private String lastname;
    private String email;
    private String password;
}

```

```

    private int tel;

    public User(String nic, String firstname, String lastname, String email, String
password, int tel) {
        this.nic = nic;
        this.firstname = firstname;
        this.lastname = lastname;
        this.email = email;
        this.password = password;
        this.tel = tel;
    }

    public String getNic() {
        return nic;
    }

    public void setNic(String nic) {
        this.nic = nic;
    }

    public String getFirstname() {
        return firstname;
    }

    public void setFirstname(String firstname) {
        this.firstname = firstname;
    }

    public String getLastname() {
        return lastname;
    }

    public void setLastname(String lastname) {
        this.lastname = lastname;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public int getTel() {
        return tel;
    }

    public void setTel(int tel) {
        this.tel = tel;
    }
}

```

CardRepository (Interface)

```

package com.example.ticket.repository;

import com.example.ticket.model.Card;
import org.springframework.data.mongodb.repository.MongoRepository;

public interface CardRepository extends MongoRepository<Card, String> {
    Card findBycardNumber(String cardNumber);
}

```

TicketRepository(Interface)

```

package com.example.ticket.repository;

import com.example.ticket.model.Ticket;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;

public interface TicketRepository extends MongoRepository<Ticket, String> {
    Ticket findBy_id(ObjectId _id);
}

```

TrainRepository(Interface)

```

package com.example.ticket.repository;

import com.example.ticket.model.Train;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;

public interface TrainRepository extends MongoRepository<Train, String> {
    Train findBytrainNo(String trainNo);
}

```

UserRepository(Interface)

```

package com.example.ticket.repository;

import com.example.ticket.model.User;
import org.bson.types.ObjectId;
import org.springframework.data.mongodb.repository.MongoRepository;

public interface UserRepository extends MongoRepository <User, String>{
    User findBynic(String nic);
}

```

Config.class

```
package com.example.ticket;

public class Config {

    public static final String a_Origin = "http://localhost:63342";

    public static String getAllowedOrigin() {
        return a_Origin;
    }

}
```