Experiment No. 07

Title: Design a web page using Angular JS.

Batch: B2 **Roll No: 1914078** **Experiment No:7**

Aim: To design a web page using Angular JS.

Resources needed: Notepad, any Web Browser and Internet.

Theory:

AngularJS is a JavaScript framework written in JavaScript. It is distributed as a JavaScript file, and can be added to a web page with a script tag:

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">

</script>

It can be added to an HTML page with a <script> tag.

Angular JS with Directive

It extends HTML attributes with Directives, and binds data to HTML with Expressions. It also extends HTML with **ng-directives**.

The **ng-app** directive defines an AngularJS application. The **ng-model** directive binds the value of HTML controls (input, select, text area) to application data. The **ng-bind** directive binds application data to the HTML view.

AngularJS starts automatically when the web page has loaded.

The **ng-app** directive tells AngularJS that the <div> element is the "owner" of an AngularJS **application**.

The **ng-model** directive binds the value of the input field to the application variable **name**.

The **ng-bind** directive binds the content of the <p> element to the application

variable **name**.

AngularJS Modules:

An AngularJS module defines an application. The module is a container for the different parts of an application. The module is a container for the application controllers. Controllers always belong to a module.

Angular JS Application:

Applications in AngularJS enable the creation of real-time Applications. There are four primary steps involved in creation of Applications in AngularJS:

1. Creation of List for an Application.
2. Adding elements in the List.

3. Removing elements from the List.
4. Error Handling

Step 1: Creation of List for an Applications.

To start with, choose the list which you want to create. Then using, controller and ng-repeat directive display the elements of the array as a list.

Step 2: Adding elements in the List.

Use the text field, in your application with the help of the *ng-model* directive. In the controller, make a function named addNewSubject, and use the value of the addSubject input field to add a subject to the 'name' array. Add a button, to add a new subject using an ng-click directive.

Step 3: Removing elements from the List.

To remove a subject, make a remove function with the index as it's a parameter. For each subject, make a span item and give them an ng-click directive to call the remove function.

Step 4: Error Handling

Errors need to be carefully handled. **For example:** If the same subject is added twice in the list, it displays an error message.

Sample Angular JS Application:

Add remove Error Handling with Subject of the Course.

```
<!DOCTYPE html>
<html>
<script src=
"https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">
</script>
<h1 style="color: green">
   KJSCE DISCIPLINES
</h1>
<body>

<script>
var app = angular.module("Subjects", []);
app.controller("my_Ctrl", function($scope) {
   $scope.name =  ["English", "Maths", "Economics"];

    $scope.addingNewSubject = function () {
       $scope.errortext = "";
```

```
        if (!$scope.addSubject) {return;}
        if ($scope.name.indexOf($scope.addSubject) == -1) {
            $scope.name.push($scope.addSubject);
        } else {
            $scope.errortext =
              "This subject is already in the list.";
        }
    }
    $scope.remove = function (x) {
        $scope.errortext = "";
        $scope.name.splice(x, 1);
    }
});
</script>
  <div ng-app="Subjects" ng-controller="my_Ctrl">
  <ul>
   <li ng-repeat="x in name">
     {{x}}<span ng-click="remove($index)">×</span>
    </li>
  </ul>
  <input ng-model="addSubject">
  <button ng-click="addingNewSubject()">Add</button>
   <p>{{errortext}}</p>
</div>
<p>Use cross icon for removing subjects.</p>
</body>
</html>
```

Activities:

□ To design a web page using Angular JS on your theme to showcase the usage of list, tables, forms and its validation, animation.

Results: (Document printout as per the format discussed by the faculty *t*)

Display of the designed webpage along with the code.

CODE:

Reserve.html (form, form validations and animations):-

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"
crossorigin="anonymous">
<link rel="stylesheet" href="styles.css">

<title>Cafe Paradise</title>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-
animate.js"></script>
<style>
li a:hover {
  background-color: lightgrey;
}
@keyframes myChange {
  from {
      height: 200px;
  } to {
      height: 0;
  }
}
</style>

</head>
<body bgcolor="black" text="white" >
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <img align="left"
src="https://www.brandcrowd.com/gallery/brands/pictures/picture13897453145721.jpg"
width="100px" height="50px">
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
          <li class="nav-item">
            <a class="nav-link" href="Restaurant_html.html">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="Gallery.html">Gallery</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="Reserve.html">Reserve</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="Order_Online.html">Order Online</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Location</a>
          </li>
          <li class="nav-item">
            <a class="nav-link"  href="#about">Login</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="Signup.html">Sign Up</a>
          </li>

        </ul>

      </div>
    </nav>
<br>
<h1 style="text-align: center; margin-top:20px;"> Reserve Your Table</h1>
<br>

<div style="text-align: center; border:solid 2px rgb(0, 0, 0); margin:10px 350px; padding-
bottom:20px ;">

<form ng-app="myApp" ng-controller="validateCtrl" name="myForm" novalidate>

  <div ng-show="myForm.$dirty" class="details" style="border: 2px solid red; width: 60%;
margin: 0 auto; text-align: left; padding-left: 20px;">
      <h3>Booking Details:</h3>
      <p>Reservation by: {{firstName + ' ' + lastName}}</p>
      <p>Number of Guests: {{guests}}</p>
      <p>Email Address: {{email}}</p>
      <p>Phone Number: {{tel}}</p>
      <p>Booking Date: {{date | date:"dd/MM/yyyy"}}</p>
      <p>Booking Time: {{time| date:"h:mma"}}</p>
  </div>

  <div>
    <img src="https://media.istockphoto.com/photos/reserved-sign-on-restaurant-table-with-
bar-background-picture-
id1129866096?k=6&m=1129866096&s=612x612&w=0&h=jlgQBWvqDziPF0HNEb0EA9t
2haGZ1UHkGcNQRpkzR1o=" onabort="abort()" height="200" width="500" style="margin-
top: 20px;" ng-hide="myForm.$dirty"><br>
  </div>

  <div>
  <p>First Name:<br>
      <input type="text" name="firstName" ng-model="firstName" required>
      <div style="color:red" ng-show="myForm.firstName.$dirty &&
myForm.firstName.$invalid">
          <span ng-show="myForm.firstName.$error.required">First Name is required.</span>
      </div>
  </p>
</div>

  <p>Last Name:<br>
```

```
    <input type="text" name="lastName" ng-model="lastName" required>
    <div style="color:red" ng-show="myForm.lastName.$dirty &&
myForm.lastName.$invalid">
        <span ng-show="myForm.lastName.$error.required">Last Name is required.</span>
    </div>
 </p>

 <p>Email:<br>
    <input type="email" name="email" ng-model="email" required>
    <div style="color:red" ng-show="myForm.email.$dirty && myForm.email.$invalid">
        <span ng-show="myForm.email.$error.required">Email is required.</span>
        <span ng-show="myForm.email.$error.email">Invalid email address.</span>
    </div>
 </p>

 <p>Phone Number<br>
    <input type="tel" name="tel" ng-model="tel" ng-pattern="/^[7-9][0-9]{9}$/" required>
    <div style="color:red" ng-show="myForm.tel.$dirty && myForm.tel.$invalid">
        <span ng-show="myForm.tel.$error.required">Phone Number is required.</span>
        <span ng-show="myForm.tel.$error.pattern">Invalid Phone Number.</span>
    </div>
 </p>


 <p>Number of Guests:<br>
    <select ng-model="guests" ng-options="num for num in nums"></select>
 </p>

 <p>Booking Date:<br>
    <input type="date" name="date" ng-model="date" required min="2021-04-25">
    <div style="color:red" ng-show="myForm.date.$dirty && myForm.date.$invalid">
        <span ng-show="myForm.date.$error.invalid">Date is required.</span>
    </div>
 </p>

 <p>Booking Time:<br>
    <input type="time" name="time" ng-model="time" required>
    <div style="color:red" ng-show="myForm.time.$dirty && myForm.time.$invalid">
        <span ng-show="myForm.time.$error.required">Time is required.</span>
    </div>
 </p>

 <input type="checkbox" ng-model="accepted">
 <span> I accept the terms and conditions of the hotel.</span><br>

 <input type="submit" value="Book My Table" ng-disabled="myForm.$pristine
    || !accepted
    || myForm.firstName.$dirty && myForm.firstName.$invalid
    || myForm.lastName.$dirty && myForm.lastName.$invalid
    || myForm.email.$dirty && myForm.email.$invalid
    || myForm.tel.$dirty && myForm.tel.$invalid
    || myForm.guests.$pristine
    || myForm.date.$pristine
    || myForm.time.$pristine
 " />
```

```
</form>

<script src="app.js"></script>
</div>
</body>

</html>
```

app.js for reserve.html:

```
var app = angular.module('myApp', ['ngAnimate']);

app.controller('validateCtrl', function ($scope) {
    $scope.firstName = '';
    $scope.lastName = '';
    $scope.email = '';
    $scope.nums = [1,2,3,4,5,6,7,8,9,'10 Or More'];
    $scope.reject = !$scope.accept;
});
```

IOrderonline.html (tables and lists):-

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"
crossorigin="anonymous">
<link rel="stylesheet" href="styleorder.css">

<title>Order Online</title>
<style>
li a:hover {
  background-color: lightgrey;
}
</style>
</head>
<body text="white" >
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <img align="left"
src="https://www.brandcrowd.com/gallery/brands/pictures/picture13897453145721.jpg"
width="100px" height="50px">
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-
expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
          <li class="nav-item">
            <a class="nav-link" href="Restaurant_html.html">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="Gallery.html">Gallery</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="Reserve.html">Reserve</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="Order_Online.html">Order Online</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Location</a>
          </li>
          <li class="nav-item">
            <a class="nav-link"  href="#about">Login</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="Signup.html">Sign Up</a>
          </li>

        </ul>

      </div>
    </nav>
<h1 style="text-align: center; margin-top:50px;">Order Online</h1>

<div ng-app ng-controller="OrderFormController">


  <form>
    <table>
      <tr>
          <td style="color: yellow;">ITEM</td>
          <td style="color: yellow;">PRICE</td>
          <td style="color: yellow;">QUANTITY</td>
      </tr>
      <tr ng-repeat="service in services" ng-click="toggleActive(service)" ng-class="{active:service.active}">
          <td ng-click="toggleActive(service)" ng-class="{active:service.active}">{{service.name}}</td>
          <td ng-click="toggleActive(service)" ng-class="{active:service.active}">Rs.{{service.price}}</td>
          <td ng-click="toggleActive(service)" ng-class="{active:service.active}">
              <button class="btn btn-light" ng-click="increaseItemAmount(service)">+</button>
              {{service.quantity}}
              <button class="btn btn-light" ng-click="decreaseItemAmount(service)">-</button>
          </td>
      </tr>
  </table>

    <div class="total">
      <h4 style="display: inline; margin-left:80px; font-weight:bold;">TOTAL:</h4> <h4 style="display: inline; margin-left: 145px; font-weight:bold;">Rs.{{total()}}</h4>
    </div>

  </form>

  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js"></script>
  <script src="script.js"></script>
</div>
```

```
<div style="text-align:center; margin-bottom: 50px;">
<input type="submit" class="btn btn-dark" id="submitbtn" value="Confirm Order">
</div>
</body>
<script>
document.getElementById("submitbtn").onclick = function formSubmit(){
        alert("Your Order Was Confirmed!!");
};


document.getElementById("submitbtn").onmousedown = function formSubmit(e){
        e.target.style.background = "rgb(255, 184, 184)";
        e.target.style.border="3px solid grey";
};
document.getElementById("submitbtn").onmouse = function formSubmit(e){
        e.target.style.removeProperty('border');
        e.target.style.removeProperty('background');
};
</script>
</html>



<!-- <ul>
    <div>
      <h5 style="margin-left: 10px; display: inline;">ITEM</h5>
      <h5 style="margin-left: 410px; display:inline;">PRICE</h5>
      <h5 style="margin-left: 10px;display:inline;">QUANTITY</h5>
    </div>
    <li ng-repeat="service in services" ng-click="toggleActive(service)" ng-
class="{active:service.active}">
      {{service.name}} <span>Rs.{{service.price}}
      <button class="btn btn-light" ng-click="increaseItemAmount(service)">+</button>
      {{service.quantity}}
      <button class="btn btn-light" ng-click="decreaseItemAmount(service)">-</button>
    </span>
    </li>
  </ul> -->
```

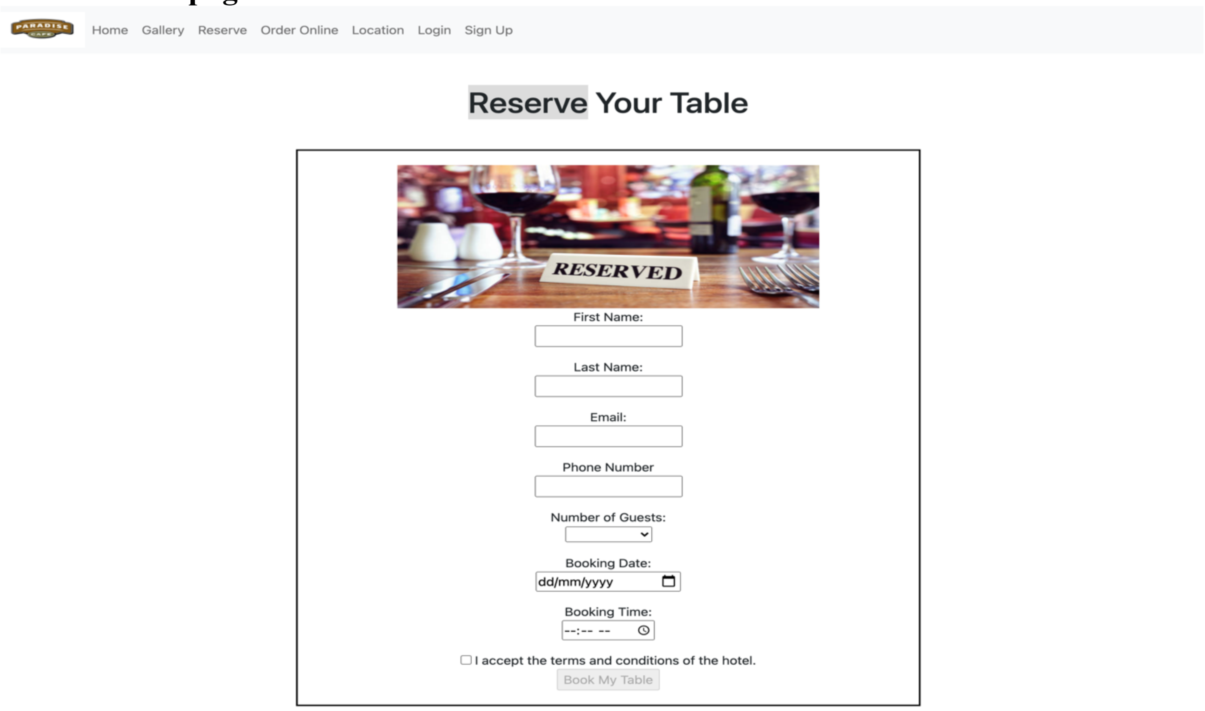
script.js for orderonline.html:

```
function OrderFormController($scope){

        $scope.services = [
                {
                        name: 'Soup',
        price: 100,
        quantity:0,
                        active:false
                },{
                        name: 'Hakka Noodles ',
        price: 300,
        quantity:0,
                        active:false
                },{
                        name: 'Fried Rice',
        price: 250,
        quantity:0,
                        active:false
                },{
                        name: 'Pizza',
        price: 220,
        quantity:0,
                        active:false
                },{
                        name: 'Pasta',
        price: 200,
        quantity:0,
                        active:false
                },{
                        name: 'Biryani',
        price: 320,
        quantity:0,
                        active:false
                },{
                        name: 'Paneer Chilli',
        price: 220,
        quantity:0,
                        active:false
                }
];

        // $scope.toggleActive = function(s){
        //          s.active = !s.active;
        // };


$scope.increaseItemAmount = function(item) {
   item.quantity++;
   item.active= true;
}

$scope.decreaseItemAmount = function(item) {
   item.quantity--;
   if (item.quantity <= 0) {
      item.quantity = 0;
      item.active = false;
   } else {
      item.active = true;
   }
}


        $scope.total = function(){

                var total = 0;

                angular.forEach($scope.services, function(s){
                        if (s.active){
                                total+= s.price*s.quantity;
                        }
                });

                return total;
```

```
    };

}
```

OUTPUT (Reserve.html -USE OF FORMS,VALIDATIONS,ANIMATIONS):

1. **Initial Webpage:-**



Reserve Your Table

First Name:

Last Name:

Email:

Phone Number

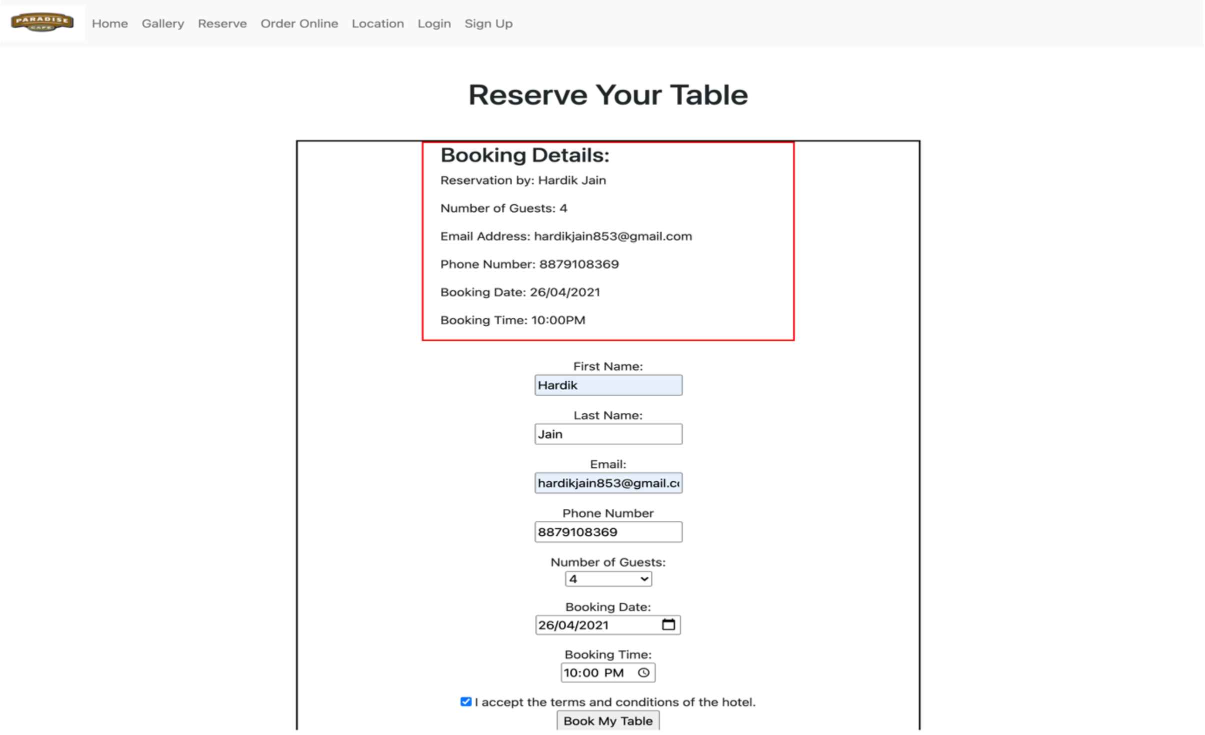
Number of Guests:

Booking Date: dd/mm/yyyy

Booking Time: --:-- --

□ I accept the terms and conditions of the hotel.

2. **Animation as image fades away:-**



Reserve Your Table

Booking Details:

Reservation by: Hardik Jain

Number of Guests: 4

Email Address: hardikjain853@gmail.com

Phone Number: 8879108369

Booking Date: 26/04/2021

Booking Time: 10:00PM

First Name: Hardik

Last Name: Jain

Email: hardikjain853@gmail.c

Phone Number 8879108369

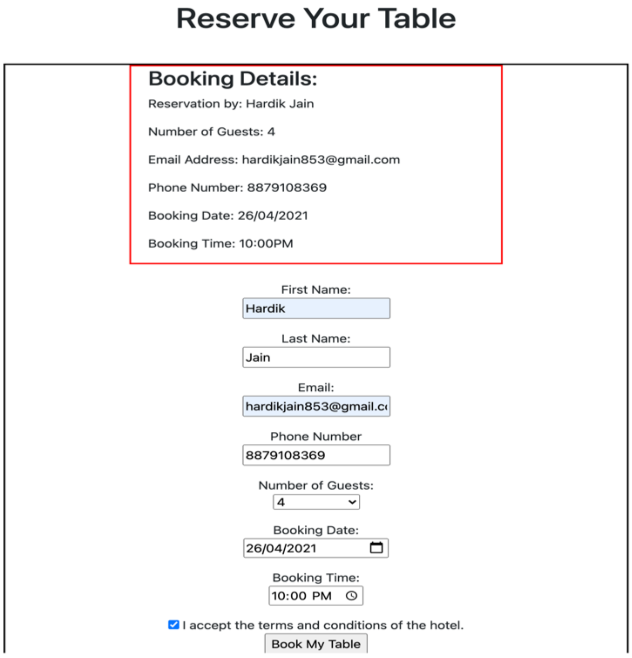
Number of Guests: 4

Booking Date: 26/04/2021

Booking Time: 10:00 PM

☑ I accept the terms and conditions of the hotel.

Image is gone:-



Reserve Your Table

Booking Details:

Reservation by: Hardik Jain

Number of Guests: 4

Email Address: hardikjain853@gmail.com

Phone Number: 8879108369

Booking Date: 26/04/2021

Booking Time: 10:00PM

First Name: Hardik

Last Name: Jain

Email: hardikjain853@gmail.c

Phone Number 8879108369

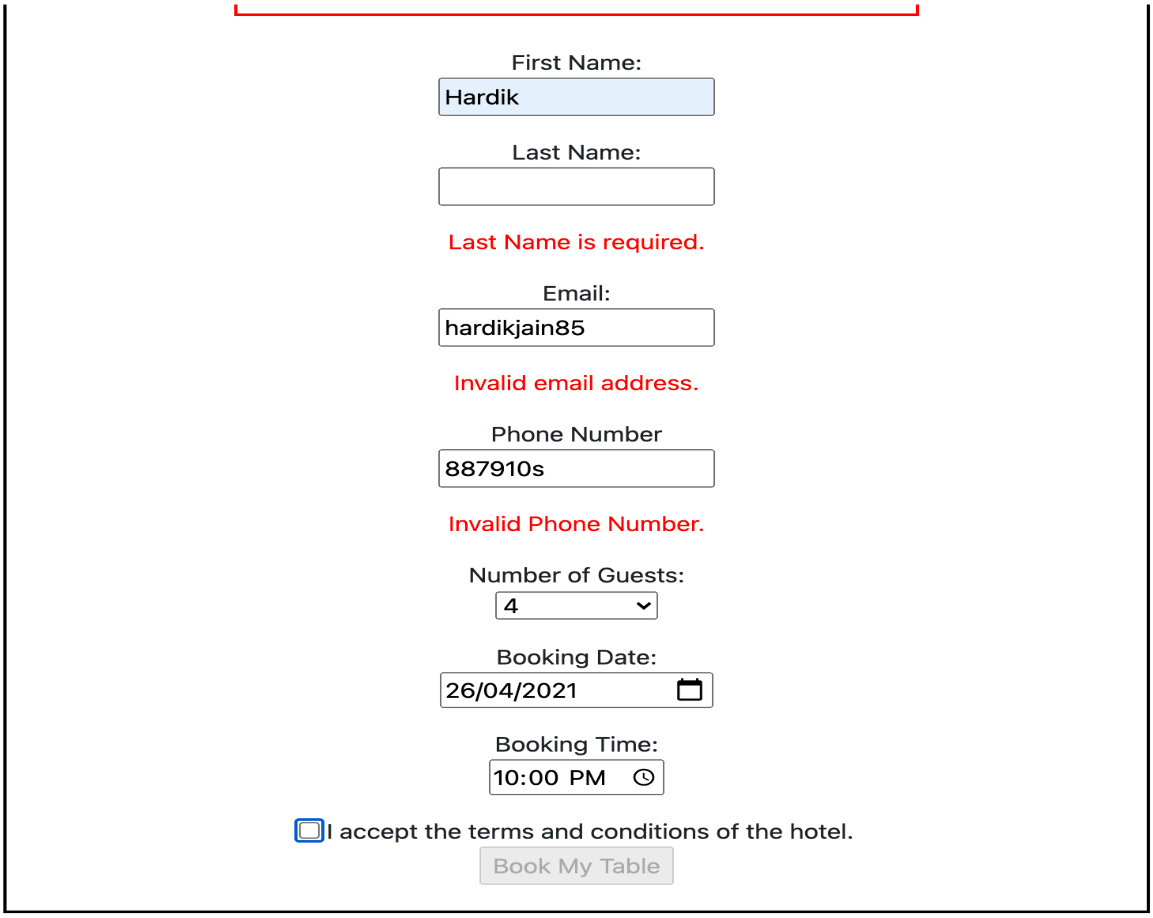
Number of Guests: 4

Booking Date: 26/04/2021

Booking Time: 10:00 PM

☑ I accept the terms and conditions of the hotel.

3. **Errors are displayed:-**



First Name: Hardik

Last Name:

Last Name is required.

Email: hardikjain85

Invalid email address.

Phone Number 887910s

Invalid Phone Number.

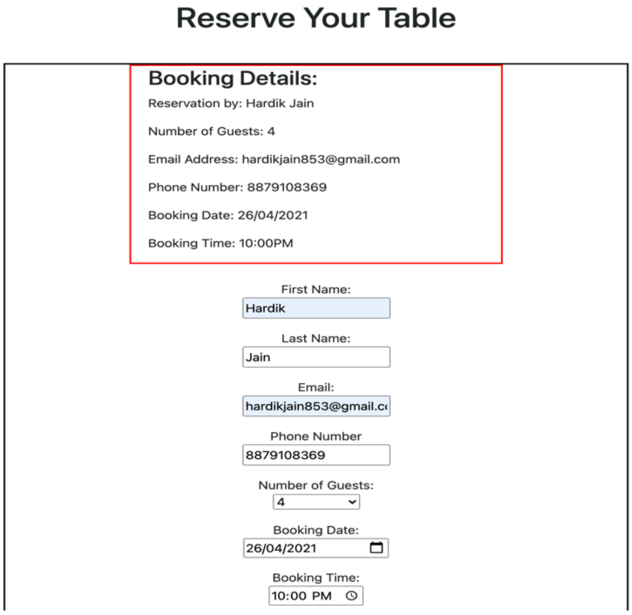
Number of Guests: 4

Booking Date: 26/04/2021

Booking Time: 10:00 PM

□ I accept the terms and conditions of the hotel.

4. **All information is displayed in booking details section:**



Reserve Your Table

Booking Details:

Reservation by: Hardik Jain

Number of Guests: 4

Email Address: hardikjain853@gmail.com

Phone Number: 8879108369

Booking Date: 26/04/2021

Booking Time: 10:00PM

First Name: Hardik

Last Name: Jain

Email: hardikjain853@gmail.co

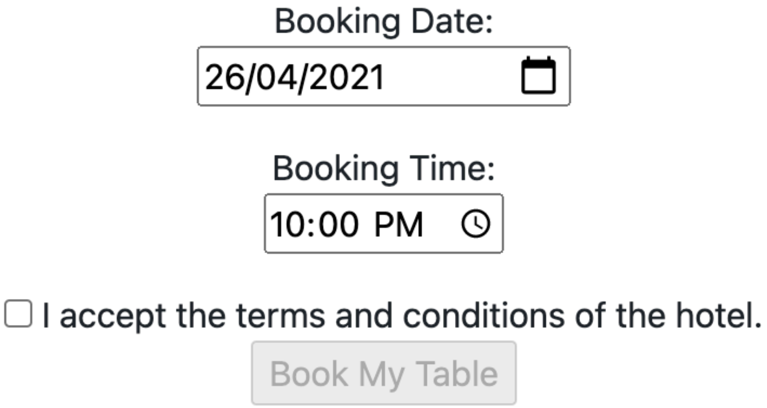
Phone Number 8879108369

Number of Guests: 4

Booking Date: 26/04/2021

Booking Time: 10:00 PM

5. **Book My Table button is disabled till form is invalid:**

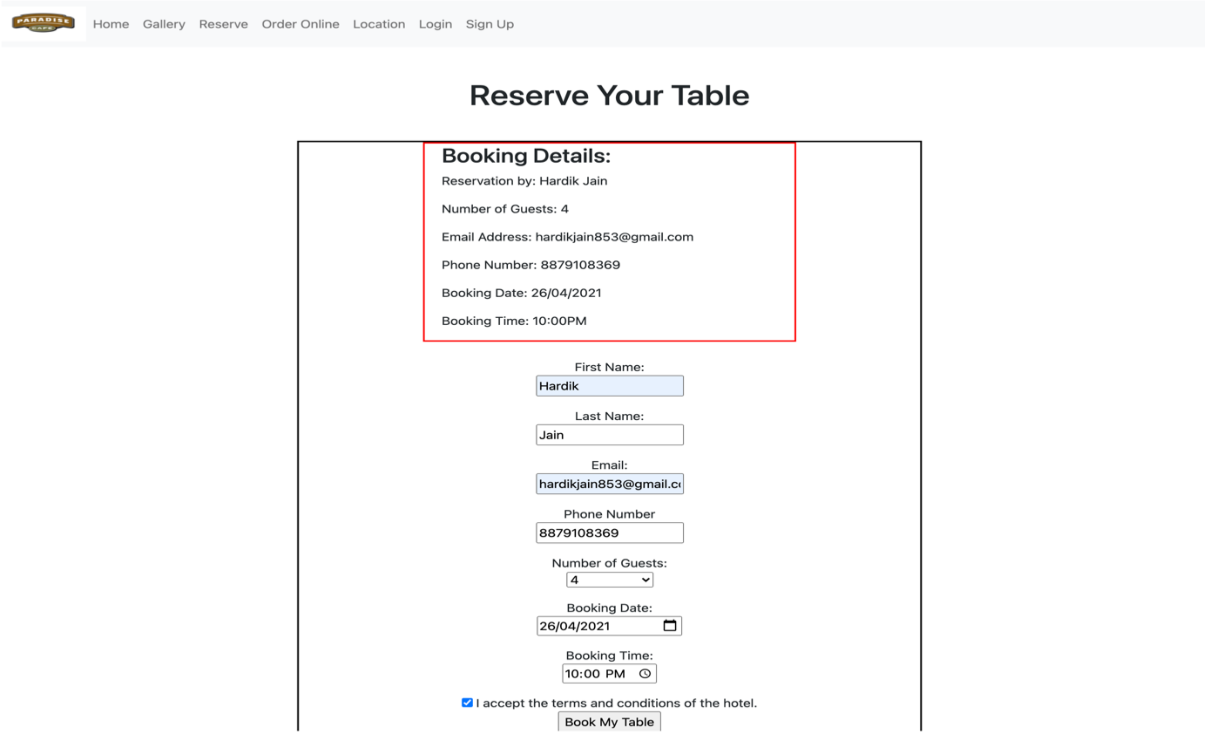


Booking Date: 26/04/2021

Booking Time: 10:00 PM

☐ I accept the terms and conditions of the hotel.

6. Book My Table button is enabled when form is valid:



Reserve Your Table

Booking Details:

Reservation by: Hardik Jain

Number of Guests: 4

Email Address: hardikjain853@gmail.com

Phone Number: 8879108369

Booking Date: 26/04/2021

Booking Time: 10:00PM

First Name: Hardik

Last Name: Jain

Email: hardikjain853@gmail.c

Phone Number 8879108369

Number of Guests: 4

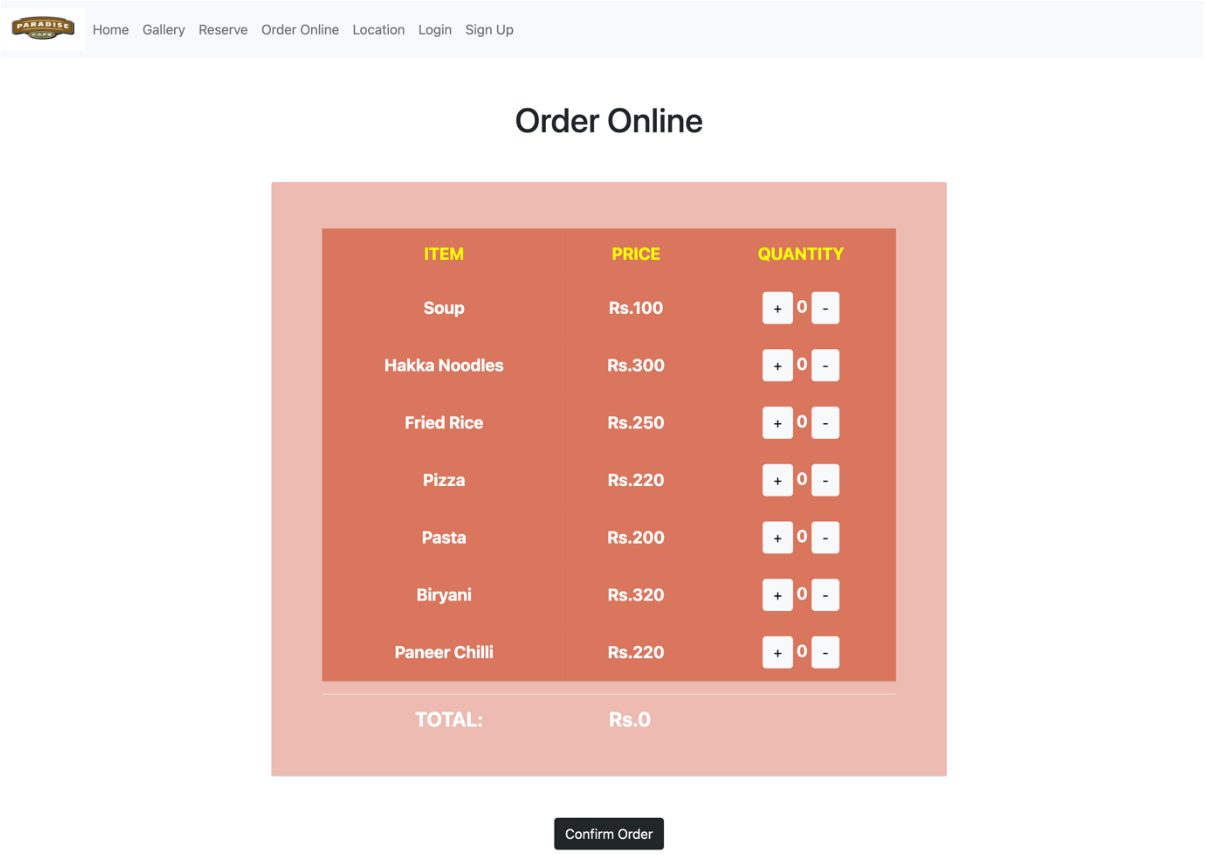
Booking Date: 26/04/2021

Booking Time: 10:00 PM

☑ I accept the terms and conditions of the hotel.

OUTPUT (Order_Online.html- USE OF TABLES IN ANGULAR):

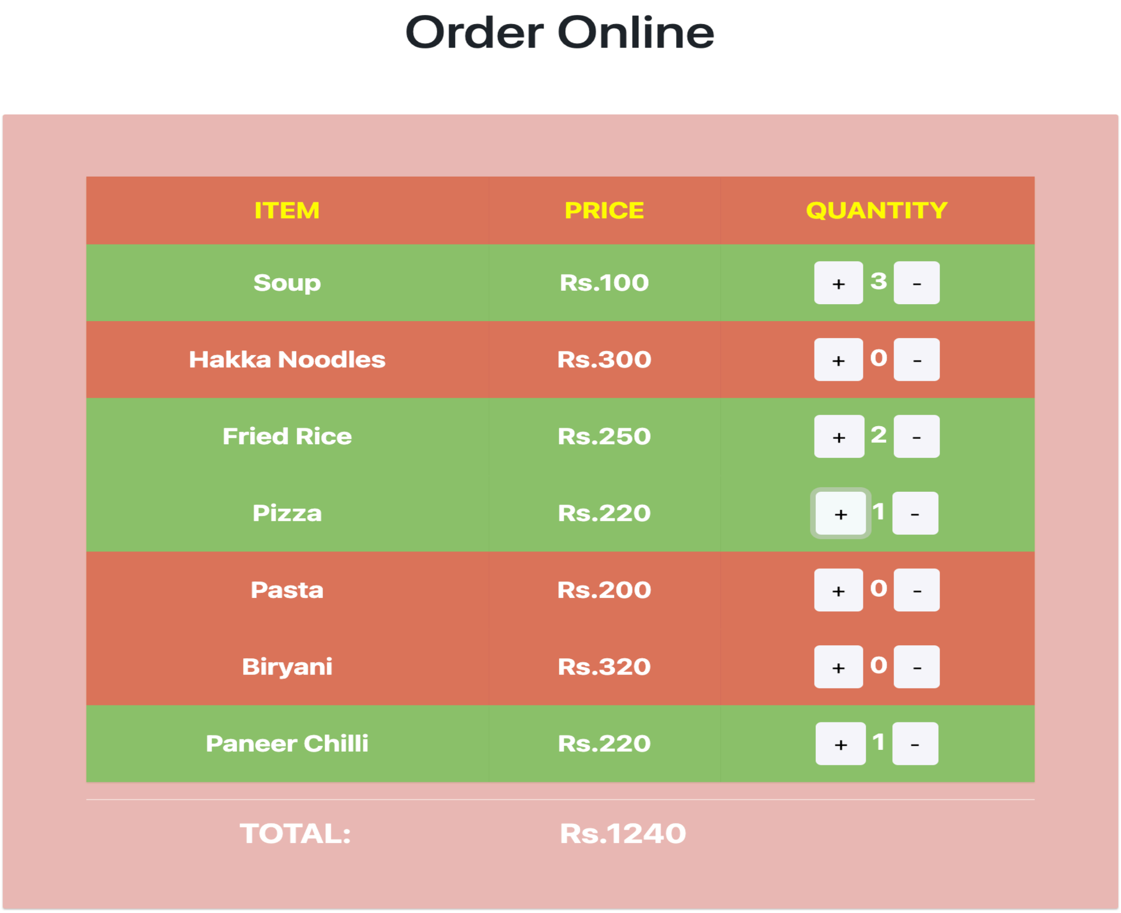
1. Initial Webpage:-



Order Online

ITEM	PRICE	QUANTITY
Soup	Rs.100	+ 0 -
Hakka Noodles	Rs.300	+ 0 -
Fried Rice	Rs.250	+ 0 -
Pizza	Rs.220	+ 0 -
Pasta	Rs.200	+ 0 -
Biryani	Rs.320	+ 0 -
Paneer Chilli	Rs.220	+ 0 -
TOTAL:	Rs.0	

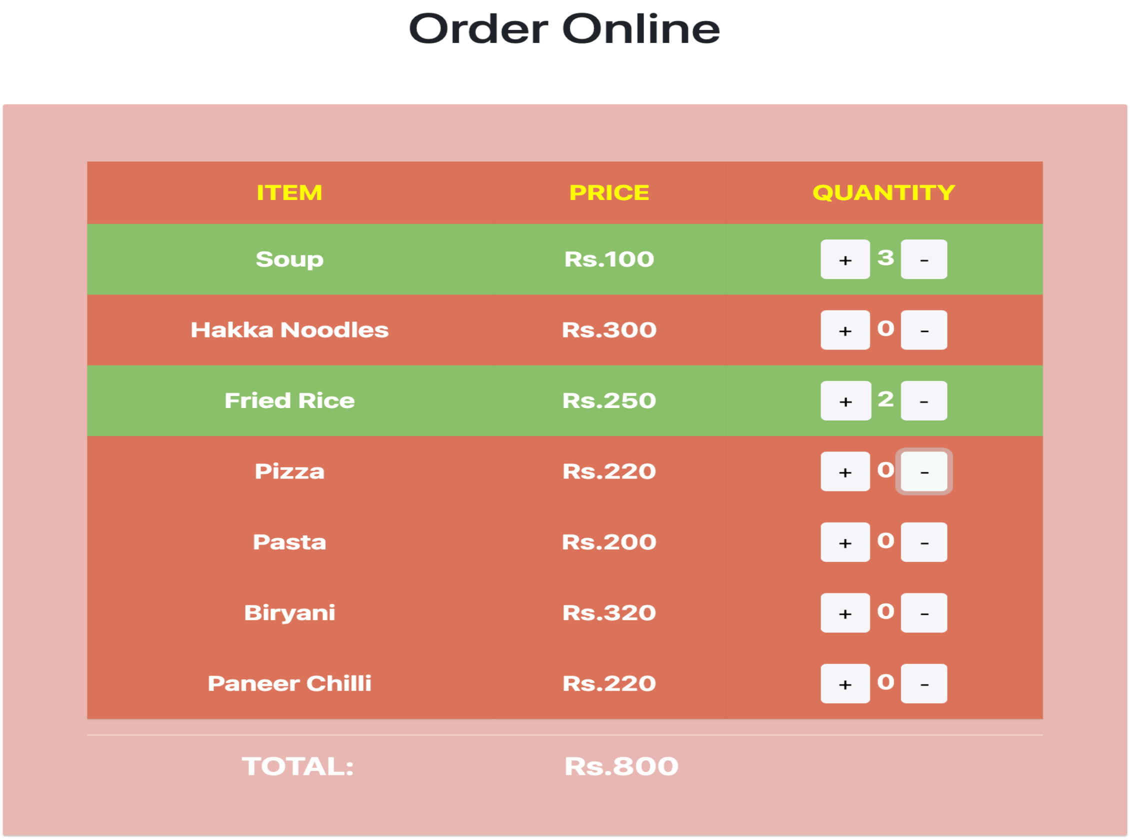
2. **Clicking on + and – changes the quantity, colour and updates the total:**



Order Online

ITEM	PRICE	QUANTITY
Soup	Rs.100	+ 3 -
Hakka Noodles	Rs.300	+ 0 -
Fried Rice	Rs.250	+ 2 -
Pizza	Rs.220	+ 1 -
Pasta	Rs.200	+ 0 -
Biryani	Rs.320	+ 0 -
Paneer Chilli	Rs.220	+ 1 -

TOTAL: Rs.1240

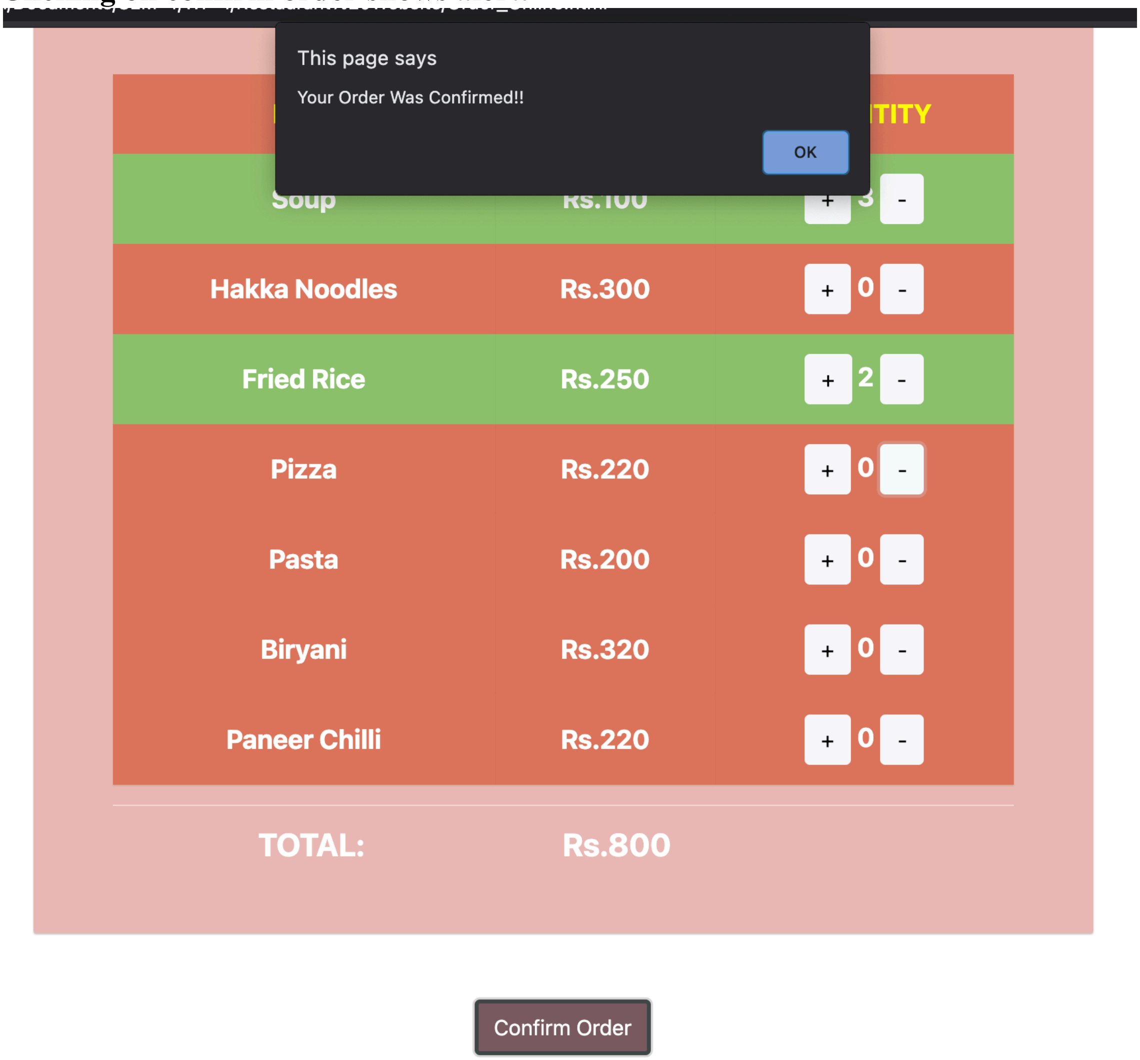


Order Online

ITEM	PRICE	QUANTITY
Soup	Rs.100	+ 3 -
Hakka Noodles	Rs.300	+ 0 -
Fried Rice	Rs.250	+ 2 -
Pizza	Rs.220	+ 0 -
Pasta	Rs.200	+ 0 -
Biryani	Rs.320	+ 0 -
Paneer Chilli	Rs.220	+ 0 -

TOTAL: Rs.800

3. **Clicking on confirm order shows alert:**



This page says

Your Order Was Confirmed!!

		TITY
Soup	Rs.100	+ 3 -
Hakka Noodles	Rs.300	+ 0 -
Fried Rice	Rs.250	+ 2 -
Pizza	Rs.220	+ 0 -
Pasta	Rs.200	+ 0 -
Biryani	Rs.320	+ 0 -
Paneer Chilli	Rs.220	+ 0 -

TOTAL: Rs.800

Questions:

1. What are the different features of angular JS you know? Which features attracts you toward the AJS? What are the disadvantage? How one can overcome it?

Ans . MVC is a software design pattern for developing web applications. It is made up of:

Model: the first level of the pattern which is responsible for maintaining data. It is similar to primitive data types like booleans, numbers, strings or objects. It is the simplest script without any getter and sorter methods.

View: responsible for showing portions of data to the user. They present the data in a particular format triggered by the controller's action.

Controller: controls the interaction between the Model and the View. It responds to user input and interacts with the data model objects. The controller receives the input, validates it, and then conducts the operations.

- Scope − These are objects that refer to the model. They act as a glue between controller and view.
- Data-binding − It is the automatic synchronization of data between model and view components.

The feature that attracts the most is MVC architecture as it makes developing application easy as you just have to split your application into a model, it's view and have a controller to it.

Disadvantages of AJS are

- Performance Tradeoff : Due to two-way data binding especially on old devices, angular dom-manipulation can also lead to a performance tradeoff since the browser will be super busy manipulating the dom elements and will take a lot of the compute resources.
- It works on the principle of a digest-cycle, which is a recursive call to check the scope of each $scope variable or watcher in the HTML (dom tree).In the case of huge nesting in the dom tree, this can have performance issues and developers should be careful with it. Also, recursion has issues like consuming a huge amount of memory since the stack keeps getting filled up and can.

These disadvantages can be overcome

- Using latest version of laptop's/PC since the performance tradeoff is problem for PC's with slow speed.
- Using less than 2000 watches per page can. This is the limit until which performance remains consistent/good.

2. Which are the different applications are exist by using AngularJS? Which application do you feal is most relevant to AngularJS and Why?

Ans. Applications that exist using AngularJS are

- youtube for PS5
- Netflix
- Upwork
- Lego
- Weather
- Gmail
- Paypal

Weather website is the most relevant application according to me as weather may change from time to time and needs to be updated i.e. changes in the DOM can be done dynamically by angular Js and is very quick compared to other javascript frameworks.

3. What do you understand by Dependency Injection in AngularJS?

Ans. Dependency Injection is a software design in which components are given their dependencies instead of hard coding them within the component. It relieves a component from locating the dependency and makes dependencies configurable. It also helps in making components reusable, maintainable and testable.

AngularJS provides a supreme Dependency Injection mechanism. It provides following core components which can be injected into each other as dependencies.

- Value
- Factory
- Service
- Provider
- Constant

4. What are the different directives are there in angular JS? Explain with suitable code.

Ans . AngularJS directives are extended HTML attributes with the prefix ng-.

The ng-app directive initializes an AngularJS application.

The ng-init directive initializes application data.

The ng-model directive binds the value of HTML controls (input, select, textarea) to application data.

ng-app code example :

```
<div ng-app="myApp" ng-controller="myCtrl">
  {{ firstName + " " + lastName }}
</div>

<script>
var app = angular.module("myApp", []);
app.controller("myCtrl", function($scope) {
  $scope.firstName = "John";
  $scope.lastName = "Doe";
});

</script>
```

ng-init code example :

```
<element ng-init="expression" ></element>
```

The ng-init directive evaluates the given expression(s)

ng-model code example :

```
<div ng-app="myApp" ng-controller="myCtrl">
  Name: <input ng-model="name">
</div>


<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
  $scope.name = "John Doe";
});
</script>
```

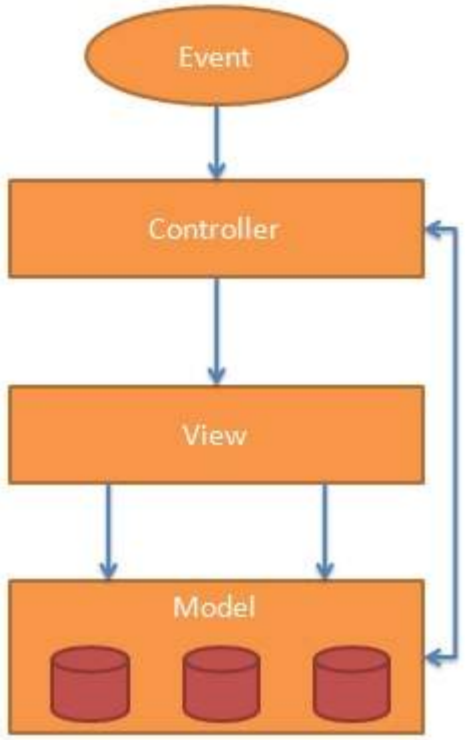
With the ng-model directive you can bind the value of an input field to a variable created in AngularJS.

5. Explain MVC architecture in context of AngularJS.

Ans. Model **V**iew **C**ontroller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts −

- **Model** − It is the lowest level of the pattern responsible for maintaining data.
- **View** − It is responsible for displaying all or a portion of the data to the user.
- **Controller** − It is a software Code that controls the interactions between the Model and View.

MVC is popular because it isolates the application logic from the user interface layer and supports separation of concerns. The controller receives all requests for the application and then works with the model to prepare any data needed by the view. The view then uses the data prepared by the controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.



The Model

The model is responsible for managing application data. It responds to the request from view and to the instructions from controller to update itself.

The View

A presentation of data in a particular format, triggered by the controller's decision to present the data. They are script-based template systems such as JSP, ASP, PHP and very easy to integrate with AJAX technology.

The Controller

The controller responds to user input and performs interactions on the data model objects. The controller receives input, validates it, and then performs business operations that modify the state of the data model.

AngularJS is a MVC based framework. In the coming chapters, we will see how AngularJS uses MVC methodology.

Outcomes:

CO4: Implement web application using React JS, Angular JS, Json and CBOR

Conclusion:
(Conclusion to be based on objectives and outcomes achieved)

In this experiment, we learnt all about angularjs, its forms and clientside validations, tables, lists, animations etc. All the tasks were implemented in the webpages.

Grade: AA/AB/BB/BC/CC/CD/DD/FF

Signature of faculty in-charge with date

References: Books/ Journals/ Websites:

- “Web Technologies: Black Book”, Dreamtech publication
- http://www.w3schools.com/