

# WebTech Lab DA-3

Binayak Bishnu

20BIT0155

**Kindly note: In copy paste from editor to word, sometimes certain characters always copied incorrectly  
Specially “-“ in ng-controller, angular-route etc.**

## Qs1

1. Develop a JSON program for creating 20 students personal information using the following fields as inputs and print the outputs in table format (Use any Javascript/JQuery/AngularJS).

student name, date\_of\_birth(date,month,year),parents(fathers\_name,mothers\_name), blood\_group,email,phone(landline,mobile),address(door\_no,street\_name,place\_name, pincode, country),degree(ug,pg and others).

```
<!DOCTYPE html>
<html>

<head>
    <title>Exercise 1</title>
    <script
src="http://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <style>
        td {
            text-align: left;
            color: black;
        }

        th {
            text-align: center;
            color: white;
        }
    </style>
</head>
```

```
<body>
    <center>
        <table style="border: 1px solid black;" id="detail_table">
            <form id="form1" class="form1class" action="">
                <thead>
                    <tr>
                        <td colspan="6" style="font-weight:bold; text-align:center; font-size: 2em;">
                            Personal Details
                        </td>
                    </tr>
                </thead>
                <tr>
                    <td><label for="student_name">Enter Name: </label></td>
                    <td><input type="text" id="student_name" placeholder="Enter your name..." required></td>
                </tr>
                <tr>
                    <td><label for="dob">Enter Date of Birth: </label></td>
                    <td><input type="number" id="date" placeholder="Enter date..." required></td>
                    <td><input type="number" id="month" placeholder="Enter month..." required></td>
                    <td><input type="number" id="year" placeholder="Enter year..." required></td>
                </tr>
                <tr>
                    <td><label for="parents">Enter Name of parents:</label></td>
                    <td><input type="text" id="father" placeholder="Enter father's name..." required></td>
                    <td><input type="text" id="mother" placeholder="Enter mother's name..." required></td>
                </tr>
                <tr>
                    <td><label for="blood_group">Enter Blood Group:</label></td>
                    <td><input type="text" id="blood_group" placeholder="Enter Blood group..." required></td>
                </tr>
                <tr>
                    <td><label for="email">Enter email: </label></td>
                    <td><input type="text" id="email" placeholder="Enter email..." required></td>
                </tr>
                <tr>
                    <td><label for="phone">Enter Contact numbers: </label></td>
```

```

        <td><input type="number" id="mobile_number"
placeholder="Enter mobile number..." required></td>
            <td><input type="number" id="landline" placeholder="Enter
landline number..." required></td>
        </tr>
        <tr>
            <td><label for="address">Enter Address: </label></td>
            <td><input type="number" id="door_no" placeholder="Enter
door_no..." required></td>
            <td><input type="text" id="street_name" placeholder="Enter
street_name..." required></td>
            <td><input type="text" id="place_name" placeholder="Enter
place_name..." required></td>
            <td><input type="number" id="pincode" placeholder="Enter
pincode..." required></td>
            <td><input type="text" id="country" placeholder="Enter
country..." required></td>
        </tr>
        <tr>
            <td><label for="degree">Enter Degree: </label></td>
            <td><input type="text" id="UG" placeholder="Enter UG...">
required></td>
            <td><input type="text" id="PG" placeholder="Enter PG...">
required></td>
            <td><input type="text" id="Others" placeholder="Enter
other..." required></td>
        </tr>
    <tfoot>
        <tr style="align-items: center;">
            <td colspan="6" style="text-align: center; padding-top:
40px">
                <input type="button" value="Submit" id="submit"
name="submitbutton" />
            </td>
        </tr>
    </tfoot>
</form>
</table>
<p id="display"></p>
<div id="div1"></div>
</center>
<script>
$(document).ready(function () {
    $("#submit").click(function () {
        details = [
            {
                "student_name": $("#student_name").val(),
                "dob": "
```

```

{
    "date": $("#date").val(),
    "month": $("#month").val(),
    "year": $("#year").val()
},
"parents":
{
    "father": $("#father").val(),
    "mother": $("#mother").val()
},
"blood_group": $("#blood_group").val(),
"email": $("#email").val(),
"phone":
{
    "mobile_number": $("#mobile_number").val(),
    "landline": $("#landline").val(),
},
"address":
{
    "door_no": $("#door_no").val(),
    "street_name": $("#street_name").val(),
    "place_name": $("#place_name").val(),
    "pincode": $("#pincode").val(),
    "country": $("#country").val()
},
"degree":
{
    "UG": $("#UG").val(),
    "PG": $("#PG").val(),
    "Others": $("#Others").val(),
}
}
];
var tbl = $("<table/>").attr("id", "mytable");
$("#div1").append(tbl);
for (var i = 0; i < details.length; i++) {
    var tr = "<tr>";
    var td1 = "<td>" + details[i]["student_name"] + "</td>";
    var td2 = "<td>" + details[i]["dob"]["date"] + "-" +
details[i]["dob"]["month"]
        + "-" + details[i]["dob"]["year"] + "</td>";
    var td3 = "<td>" + details[i]["parents"]["father"] + "," +
details[i]["parents"]["mother"] + "</td>";
    var td4 = "<td>" + details[i]["blood_group"] + "</td>";
    var td5 = "<td>" + details[i]["email"] + "</td>";
    var td6 = "<td>" + details[i]["phone"]["mobile_number"] +
"," +
details[i]["phone"]["landline"] + "</td>";

```

```

        var td7 = "<td>" + details[i]["address"]["door_no"] + "," +
            details[i]["address"]["street_name"] + "," +
            details[i]["address"]["place_name"] + "," +
            details[i]["address"]["pincode"] + "," +
            details[i]["address"]["country"] + "</td>";
        var td8 = "<td>" + details[i]["degree"]["UG"] + "," +
            details[i]["degree"]["PG"] + "," +
            details[i]["degree"]["Others"] + "</td>";
        $("#mytable").append(tr + td1 + td2 + td3 + td4 + td5 + td6
+ td7 + td8);
    }
});
</script>
</body>

</html>

```



Exercise 1

127.0.0.1:5500/Web\_Tech\_A2:L17,18/WebTech\_L17\_18/Assessment-3/Qs-1.html

**Personal Details**

Enter Name:	<input type="text" value="Enter your name..."/>
Enter Date of Birth:	<input type="text" value="Enter date..."/> <input type="text" value="Enter month..."/> <input type="text" value="Enter year..."/>
Enter Name of parents:	<input type="text" value="Enter father's name..."/> <input type="text" value="Enter mother's name..."/>
Enter Blood Group:	<input type="text" value="Enter Blood group..."/>
Enter email:	<input type="text" value="Enter email..."/>
Enter Contact numbers:	<input type="text" value="Enter mobile number..."/> <input type="text" value="Enter landline number..."/>
Enter Address:	<input type="text" value="Enter door_no..."/> <input type="text" value="Enter street_name..."/> <input type="text" value="Enter place_name..."/> <input type="text" value="Enter pincode..."/> <input type="text" value="Enter country..."/>
Enter Degree:	<input type="text" value="Enter UG..."/> <input type="text" value="Enter PG..."/> <input type="text" value="Enter other..."/>
<input type="button" value="Submit"/>	

BII

Exercise 1

127.0.0.1:5500/Web\_Tech\_A2:L17\_18/WebTech\_L17\_18/Assessment-3/Qs-1.html

Personal Details				
Enter Name:	BB			
Enter Date of Birth:	1	1	2002	
Enter Name of parents:	qwerty	asdfgh		
Enter Blood Group:	A-ve			
Enter email:	abcd@gmail.com			
Enter Contact numbers:	1235678930	9876540321		
Enter Address:	1	fdq	gfrew	700001 India
Enter Degree:	BTech	Nan	Nan	
<input type="button" value="Submit"/>				

Binayak Bishnu 11-09-2002 ABD ,DEF O+ve example.com 1234567890,0987654321 2,abc,def,700000,India BTech,NaN,NaN  
 BB 1-1-2002 qwerty,asdfgh A-ve abcd@gmail.com 1235678930,9876540321 1,fdq,gfrew,700001,India BTech,NaN,NaN

## Qs2

2. Develop an angular shopping application to buy books and remove if not required.

If the user enters a book in the text box and clicks addbook button, the book is added and if the user clicks on cross symbol next to each book given in the picture below, the book is removed from the list.

If the book is selected twice, display an appropriate error message. Use angular animations to add and remove items.

### Books

Web Technologies	X
Operating System	X
Database systems	X
Computer Architecture	X

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <title>Exercise 2</title>
```

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

<body>
    <div ng-app="myShoppingList" ng-cloak ng-controller="myCtrl" style="max-
width:400px;">
        <header>
            <h3>Books</h3>
        </header>
        <ul>
            <li ng-repeat="x in products">
                {{x}}
                <span ng-click="removeItem($index)" style="cursor:pointer;">&ampnbspx</span>
            </li>
        </ul>
        <div>
            <div>
                <input placeholder="Add books here" ng-model="addMe">
            </div>
            <div>
                <button ng-click="addItem()">Add</button>
            </div>
        </div>
        <p style="color:red">{{errortext}}</p>
    </div>
</div>
<script>
    var app = angular.module("myShoppingList", []);
    app.controller("myCtrl", function ($scope) {
        $scope.products = ["Web Technologies Simplified", "Ron Larsen",
"Stats for College", "TOC Easy"];
        $scope.addItem = function () {
            $scope.errortext = "";
            if (!$scope.addMe) {
                $scope.errortext = "Add something";
                return;
            }
            if ($scope.products.indexOf($scope.addMe) == -1) {
                $scope.products.push($scope.addMe);
            }
            else {
                $scope.errortext = "The item is already in your shopping
list.";
            }
        }
    })

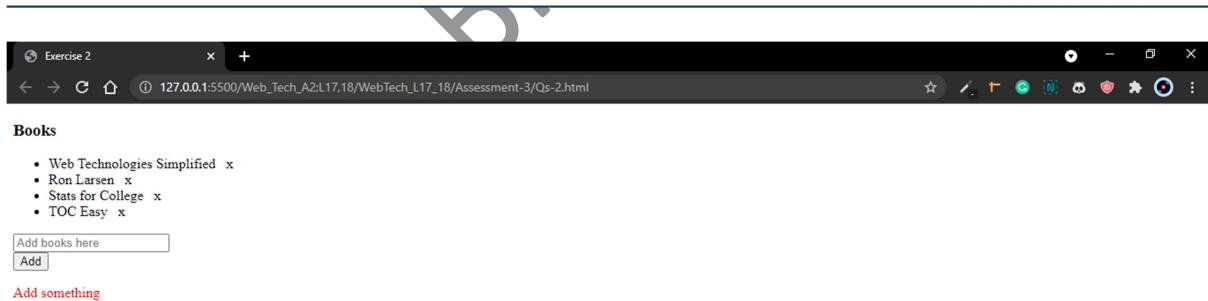
```

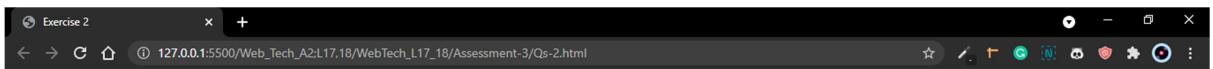
```

        }
        $scope.removeItem = function (x) {
            $scope.errortext = "";
            $scope.products.splice(x, 1);
        }
    });
</script>
</body>

</html>

```





### Books

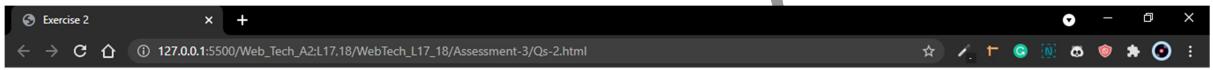
- Web Technologies Simplified x
- Ron Larsen x
- Stats for College x
- TOC Easy x

Ron Larsen

Add

The item is already in your shopping list.

20b.



### Books

- Web Technologies Simplified x
- Ron Larsen x
- Stats for College x
- TOC Easy x
- STS x

STS

Add



### **Qs3**

3. Product details are maintained in JSON file and gets it using AJAX. Develop the page contains a dropdown list which lists the value of product name. While clicking Availability button display the details of corresponding selected product as Product ID, name, price, stock and type.

### **JSON**

```
{  
    "p1": {  
        "id": "1",  
        "name": "Product 1",  
        "price": "Rs. 200",  
        "stock": "3",  
        "type": "type 1"  
    },  
    "p2": {  
        "id": "2",  
        "name": "Product 2",  
        "price": "Rs. 340",  
        "stock": "1",  
        "type": "type 2"  
    },  
    "p3": {  
        "id": "3",  
        "name": "Product 3",  
        "price": "Rs. 1000",  
        "stock": "31",  
        "type": "type 3"  
    },  
    "p4": {  
        "id": "4",  
        "name": "Product 4",  
        "price": "Rs. 714",  
        "stock": "0",  
        "type": "type 4"  
    },  
    "p5": {  
        "id": "5",  
        "name": "Product 5",  
        "price": "Rs. 20",  
        "stock": "10",  
        "type": "type 5"  
    },  
}
```

```
"p6": {  
    "id": "6",  
    "name": "Product 6",  
    "price": "Rs. 1200",  
    "stock": "100",  
    "type": "type 6"  
}  
}  
}
```

## HTML

```
<html>  
  
<head>  
    <title>Exercise 3</title>  
  
<style>  
    body {  
        text-align: center;  
    }  
  
    h1 {  
        margin-top: 5%;  
        margin-bottom: 5%;  
    }  
  
    select {  
        outline: none;  
        background: black;  
        border: none;  
        padding: 7px;  
        color: white;  
        border-radius: 5px;  
    }  
  
    option {  
        background-color: black;  
        color: white;  
    }  
  
.button {  
    width: 20%;  
    padding: 7px;  
    color: black;  
    background-color: rgba(0, 0, 0, 0);  
    transition: 0.5s;  
}
```

```
.button:hover {
    color: white;
    background-color: black;
}

.tb {
    display: none;
}

table {
    border: 1px solid black;
    margin: auto;
    margin-top: 10%;
}

th {
    padding: 7px;
    width: 150px;
    border-left: 0.5px solid black;
    border-right: 0.5px solid black;
    border-bottom: 0.5px solid black;
}

td {
    text-align: center;
    padding: 7px;
    width: 150px;
    border-left: 0.5px solid black;
    border-right: 0.5px solid black;
}

```

</style>

```
<script type="text/javascript" src="Qs-3data.json"></script>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

```

</head>

```
<body>
    <h1>Product Availability</h1>
    <select name="option list" id="list">
        <option value="none" selected disabled hidden> Choose Products
    </option>
        <option value="Product 1">Product 1</option>
        <option value="Product 2">Product 2</option>
        <option value="Product 3">Product 3</option>
        <option value="Product 4">Product 4</option>
        <option value="Product 5">Product 5</option>
        <option value="Product 6">Product 6</option>
    </select>

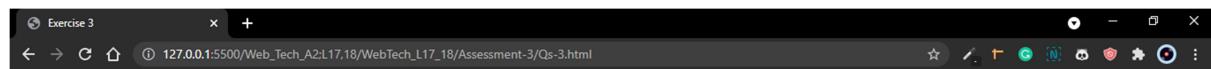
```

```
<input type="button" class="button" value="Check availability" />
<h3>Click on Check Availability To See Details</h3>
<div class="tb">
  <table>
    <tr>
      <th>Product ID</th>
      <th>Name</th>
      <th>Price</th>
      <th>Stock</th>
      <th>Type</th>
    </tr>
    <tr>
      <td class="id"></td>
      <td class="name"></td>
      <td class="price"></td>
      <td class="stock"></td>
      <td class="type"></td>
    </tr>
  </table>
</div>

<script>
  document.querySelector(".button").addEventListener('click', function ()
{
  var select1 = document.querySelector("#list");
  var name = select1.options[select1.selectedIndex].value;
  if (name === 'none') {
    alert("Select from the dropdown list");
  }
  else {
    $("h3").hide();
    $(".tb").show();
    $.getJSON("Qs-3data.json", function (obj) {
      $.each(obj, function (key, value) {
        if (value.name === name) {
          document.querySelector(".name").textContent =
value.name;
          document.querySelector(".id").textContent =
value.id;
          document.querySelector(".price").textContent =
value.price;
          document.querySelector(".stock").textContent =
value.stock;
          document.querySelector(".type").textContent =
value.type;
        }
      });
    });
  });
}</script>
```

```
        }
    });
</script>
</body>

</html>
```

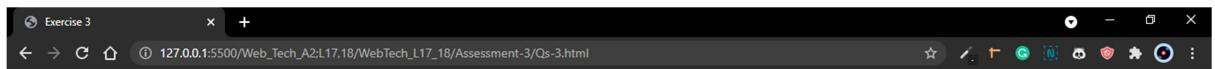


## Product Availability

Click on Check Availability To See Details



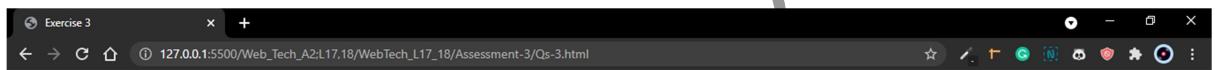
Click on Check Availability To See Details



## Product Availability

Product 2

Product ID	Name	Price	Stock	Type
2	Product 2	Rs. 340	1	type 2



## Product Availability

Product 6

Product ID	Name	Price	Stock	Type
6	Product 6	Rs. 1200	100	type 6

## Qs4

4. Develop an angular application to validate the given form. Clearly mention the status of validation like required, invalid or valid message to the respective field.

Your Name (required)  
 This field is required.

Your Email (required)  
 Please enter a valid email address.

Your Phone  
 Please enter a valid number.

Website  
 Please enter a valid URL.

Credit Card  
 Please enter a valid credit card number.

Date Of Birth  
 Please enter a valid date.

```
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/angular.min.js"></script>
<style>
    input.ng-invalid.ng-dirty {
        border-bottom: 2px solid red;
    }
</style>

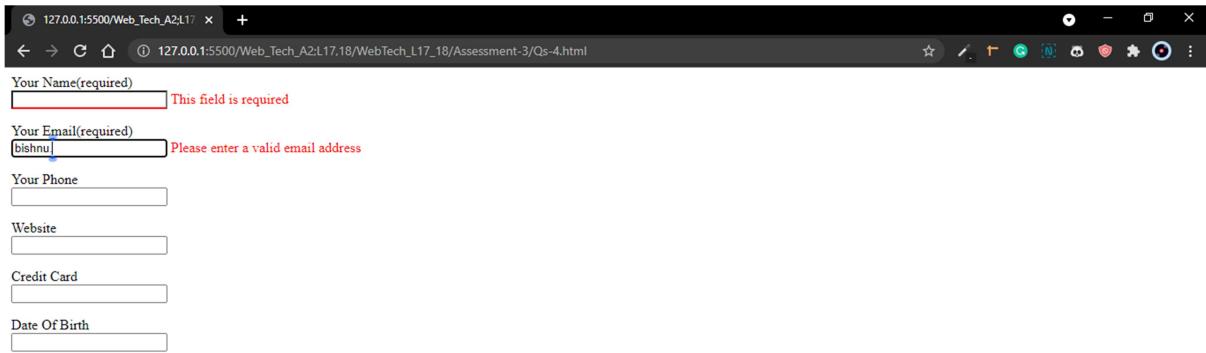
<body>
    <form name="form1" ng-app="myApp" ng-controller="validateCtrl" novalidate>
        <p>Your Name(required)<br><input type="text" name="name" ng-model="name" required>
            <span style="color:red" ng-show="form1.name.$dirty && form1.name.$invalid">
                <span ng-show="form1.name.$error.required">This field is required</span>
            </span>
        </p>
        <p>Your Email(required)<br><input type="email" name="email" ng-model="email" required>
```

```

        <span style="color:red" ng-show="form1.email.$dirty &&
form1.email.$invalid">
            <span ng-show="form1.email.$error.required">This field is
required</span>
            <span ng-show="form1.email.$error.email">Please enter a valid
email address</span>
        </span>
    </p>
    <p>Your Phone<br><input type="text" name="phone" ng-model="phone" ng-
pattern="/[0-9]{10}/" required>
        <span style="color:red" ng-show="form1.phone.$dirty &&
form1.phone.$invalid">
            <span ng-show="form1.phone.$error.required">This field is
required</span>
            <span ng-show="form1.phone.$error.pattern">Invalid phone</span>
        </span>
    </p>
    <p>Website<br><input type="text" name="webs" ng-model="webs"
ng-pattern="/https?:\/\/(www\.)?[-a-zA-Z0-
9@:%._\+~#=]{1,256}\.([a-zA-Z0-9()]{1,6}\b([-a-zA-Z0-9()@:%_\+.~#?&//=]*)/"
required>
        <span style="color:red" ng-show="form1.webs.$dirty &&
form1.webs.$invalid">
            <span ng-show="form1.webs.$error.required">This field is
required</span>
            <span ng-show="form1.webs.$error.pattern">Invalid
website</span>
        </span>
    </p>
    <p>Credit Card<br><input type="text" name="creditcard" ng-
model="creditcard" ng-pattern="/^([0-9]{16})$/">
        <span style="color:red" ng-show="form1.creditcard.$dirty &&
form1.creditcard.$invalid">
            <span ng-show="form1.creditcard.$error.pattern">Invalid
creditcard</span>
        </span>
    </p>
    <p>Date Of Birth<br><input type="text" name="dob" ng-model="dob"
ng-pattern="/^(0[1-9]|1[2-9]|3[01])[- /.](0[1-9]|1[012])[-
.](19|20)\d\d$/">
        <span ng-show="form1.dob.$dirty && form1.dob.$invalid">
            <span ng-show="form1.dob.$error.pattern">Invalid dob</span>
        </span>
    </p>
</form>
<script>
    var app = angular.module('myApp', []);
    app.controller('validateCtrl', function ($scope) {

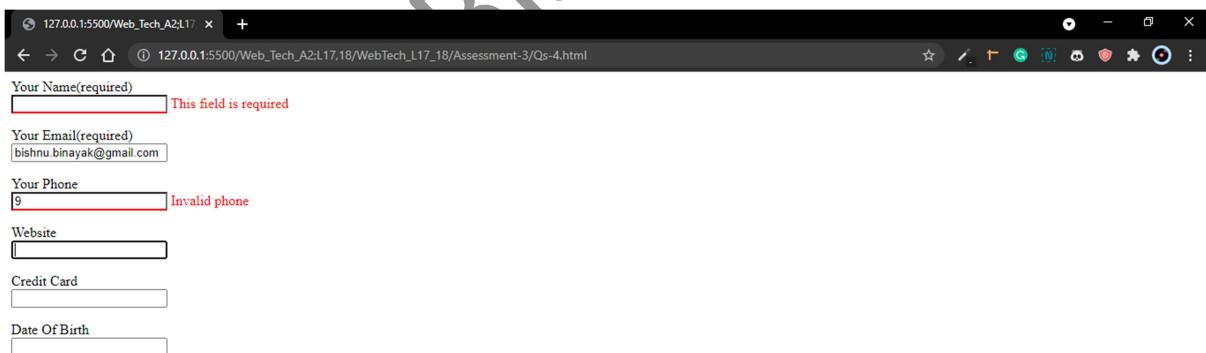
```

```
});  
</script>  
</body>  
  
</html>
```



A screenshot of a web browser window displaying a form. The URL in the address bar is 127.0.0.1:5500/Web\_Tech\_A2:L17. The form fields and their validation messages are:

- Your Name(required)  This field is required
- Your Email(required)  Please enter a valid email address
- Your Phone
- Website
- Credit Card
- Date Of Birth



A screenshot of a web browser window displaying a form. The URL in the address bar is 127.0.0.1:5500/Web\_Tech\_A2:L17. The form fields and their validation messages are:

- Your Name(required)  This field is required
- Your Email(required)
- Your Phone  Invalid phone
- Website
- Credit Card
- Date Of Birth

## Qs5

5. Develop an Angular Single Page Application to show the details of weather, movie and currency rates of today. (Use templateUrl)

Now, if you run the application and click the **Weather**, the app displays like below. Create a separate file weather.html and displayed it in a same page. Maintain weather details in JSON format. While clicking the submit button, your script display weather details for the corresponding city value which has entered in text field. Similar movie and currency details should be displayed in same page (Use template).

## MAIN HTML

```
<html>

<head>
    <title>Exercise 5</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/1.8.0/angular-route.js"></script>
    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
    <style>
        ul {
            list-style-type: none;
            margin-top: 0;
            margin: 0;
            padding: 0;
            overflow: hidden;
        }

        li {
            float: left;
            margin: 2%;
            border-radius: 5px;
            padding: 0.1%;
        }

        li:hover {
            background-color: rgb(218, 217, 214);
        }

        li a {
            display: block;
            color: black;
        }
    </style>
</head>
<body>
    <ul>
        <li>
            <a href="#">Weather</a>
        </li>
        <li>
            <a href="#">Movie</a>
        </li>
        <li>
            <a href="#">Currency</a>
        </li>
    </ul>
</body>
</html>
```

```
        font-weight: bolder;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
    }

    .active {
        background-color: black;
        border-radius: 5px;
        color: white;
    }

    h1 {
        margin: 2%;
    }

    input {
        width: 50%;
        padding: 9px;
        outline: none;
        border: none;
        border-radius: 5px;
        background-color: rgb(236, 244, 247);
        margin-left: 2%;
    }

    button {
        background-color: black;
        border-radius: 5px;
        color: white;
        border: none;
        width: 7%;
        padding: 7px;
        outline: none;
        transition: 0.5s;
        margin-left: 2%;
    }

    .show {
        margin-top: 10%;
        margin-left: 2%;
        margin-bottom: 0;
        background-color: rgb(243, 236, 236);
        width: 96%;
        height: 28.5%;
    }
}

h2,
```

```

        h3 {
            margin-left: 2%;
            margin-top: 2%;
        }
    </style>
</head>

<body ng-app="app">
    <ul>
        <li> <a href="#!/weather" class="active">Weather</a> </li>
        <li> <a href="#!/movie">Movie details</a></li>
        <li><a href="#!/currency">Currency Rates</a></li>
    </ul>
    <div ng-view></div>

    <script>
        var app = angular.module("app", ["ngRoute"]);
        app.config(function ($routeProvider) {
            $routeProvider
                .when("/weather", {
                    templateUrl: "Qs-5weather.html"
                })
                .when("/movie", {
                    templateUrl: "moviedetail.html"
                })
                .when("/currency", {
                    templateUrl: "currencydetail.html"
                });
        });
    </script>
</body>

</html>

```

## WEATHER HTML

```

<html>

    <head>
        <title>19BIT0244 Question 5</title>
        <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></sc
ript>
        <script
src="http://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
        <script src="Qs-5weather.js"></script>
        <script src="Qs-5weatherdata.json"></script>

```

```
<style>
    ul {
        list-style-type: none;
        margin: 0;
        padding: 0;
        overflow: hidden;
    }

    li {
        float: left;
        margin: 2%;
        padding: 0.1%;
    }

    li a {
        display: block;
        color: black;
        font-weight: bolder;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
    }

    .active {
        background-color: black;
        border-radius: 5px;
        color: white;
    }

    h1 {
        margin-left: 39%;
    }

    input {
        width: 50%;
        padding: 9px;
        border-radius: 5px;
        background-color: chartreuse;
        margin-left: 24%;
    }

    button {
        background-color: black;
        border-radius: 5px;
        color: white;
        width: 12%;
        padding: 7px;
        outline: none;
    }

```

```
        transition: 0.5s;
        margin-left: 42%;
        margin-top: 10px;
    }

    button:hover {
        background-color: #bbac58;
        transition: 0.5s;
    }

.show {
    margin-top: 10%;
    margin-left: 24%;
    margin-bottom: 0;
    background-color: rgb(243, 236, 236);
    width: 50%;
    border-radius: 20px;
    height: 27.3%;
}

h2,
h3 {
    margin-left: 2%;
    margin-top: 2%;
}

```

</style>

</head>

```
<body ng-app="weather-app">

    <h1> Yahoo Weather!</h1>
    <input type="text" id="name" placeholder="Enter City Name" class="city"
required><br><br>
    <input type="text" placeholder="Enter State Name" class="state"
required><br><br>
    <button>Submit</button>
    <show>
        <div class="show">
            <h2></h2>
            <h3 class="one"></h3>
            <h3 class="two"></h3>
        </div>
    </show>

    <script>
        var app1 = angular.module('weather-app', []);
        app1.directive('show', function () {
            var lin;
```

```

        lin = function (scope, element, attrs) {
            document.querySelector("button").addEventListener('click', () => {
                alert("clicked");
            });
        });
    </script>

</body>
</html>

```

## WEATHER JS

```

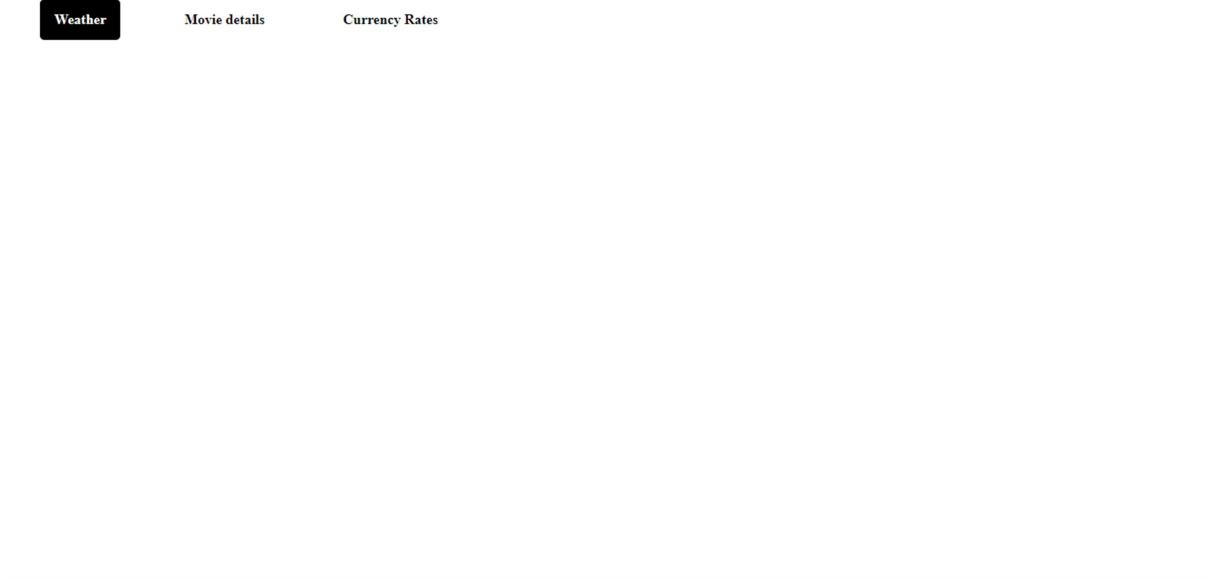
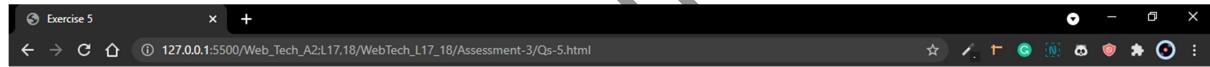
var app1 = angular.module('weather-app', []);
app1.directive('show', function () {
    var lin;
    lin = function (scope, element, attrs) {
        document.querySelector("button").addEventListener('click', () => {
            alert("clicked");
            var city = document.querySelector(".city").value;
            $.getJSON("Qs-5weatherdata.json", function (obj) {
                $.each(obj, function (key, value) {
                    if (value.name === city) {
                        document.querySelector("h2").textContent = "City , State : " + value.name + " , " + value.state;
                        document.querySelector(".one").textContent =
                            "Temperature : " + value.temp;
                        document.querySelector(".two").textContent =
                            "Conditions : " + value.condition;
                    }
                });
            });
        });
    return {
        restrict: 'E',
        link: lin
    }
});

```

## WEATHER JSON

```
{  
  "weather": {  
    "name": "kolkata",  
    "state": "West Bengal",  
    "temp": "35",  
    "condition": "Very Humid"  
  },  
  "weather1": {  
    "name": "Delhi",  
    "state": "Delhi",  
    "temp": "45",  

```



## Qs6

6. Develop an web application using AngularJS code for a small Meal Planner web page called Plan-It!. For simplicity, consider a "full day meal plan" as a breakfast, lunch, and dinner meal types. Below are two screenshots of the Plan-It! page:

**Plan-It!**

Click the button to generate a random 3-meal menu plan!

[Fetch Random Menu](#)

**Initial View**

**Plan-It!**

Click the button to generate a random 3-meal menu plan!

[Fetch Random Menu](#)

**Original Cheerios with Plain Soymilk (Breakfast)**

1 cup of Original Cheerios with 1/2 cup of soymilk - a good source of fiber!

Food Groups:

- Dairy
- Grains

**Froot Loops Cereal (Lunch)**

A baggy of rainbow Froot Loops - a classic college student lunch.

Food Groups:

- Grains

**Pinkie Pies (Dinner)**

Two pies made with love and filled full of flavors. Blueberry, strawberry, you name it, you have it!

Food Groups:

- Grains
- Fruit
- Other

**After Fetching Random Full-Day Menu**

```
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Exercise 6</title>
```

```
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/angular.min.js"></sc
ript>
</head>

<body>

<div ng-app="myApp" ng-controller="jsCtrl">
  <h1>Plan-It!</h1>
  <p>
    Click the button to generate a random 3-meal menu plan!
    <button id="day-btn" ng-click="fetchmenu()">Fetch Random
  Menu</button>
  </p>
  <section id="day-results" class="hidden">
    <p>Rand index generated breakfast:
    {{rand_index_generated_breakfast}}</p>
    <p>Rand index generated lunch: {{rand_index_generated_lunch}}</p>
    <p>Rand index generated dinner: {{rand_index_generated_dinner}}</p>
    <p>See console.log for other details</p>
    <article id="breakfast">
      <h2><span class="name">{{breakfastname}}</span>
    (Breakfast)</h2>
      <hr>
      <p class="description">{{breakfastdesc}}</p>
      <p>Food Groups:</p>
      <ul class="food-groups">
        <!-- <li ng-hide="myvar" ng-repeat="x in
  foodmenu.breakfast.foodgroups">{{x}}</li> -->
        <li ng-hide="myvar" ng-repeat="x in
  breakfastgrps">{{x}}</li>
      </ul>
    </article>
    <article id="lunch">
      <h2><span class="name">{{lunchname}}</span> (Lunch)</h2>
      <hr>
      <p class="description">{{lunchdesc}}</p>
      <p>Food Groups:</p>
      <ul class="food-groups">
        <li ng-hide="myvar" ng-repeat="y in lunchgrps">{{y}}</li>
      </ul>
    </article>
    <article id="dinner">
      <h2><span class="name">{{dinnername}}</span> (Dinner)</h2>
      <hr>
      <p class="description">{{dinnerdesc}}</p>
      <p>Food Groups:</p>
      <ul class="food-groups">
```

```
        <li ng-hide="myvar" ng-repeat="z in dinnergrps">{{z}}</li>
    </ul>
</article>
</section>
</div>

<script>
var app = angular.module('myApp', []);
app.controller('jsCtrl', function ($scope) {
    $scope.foodmenu = {
        "breakfast": [
            {
                "name": "Blueberry Oatmeal",
                "description": "One cup of hot oatmeal with 1/4 cup of
fresh blueberries.",
                "foodgroups": ["Grains", "Fruit"]
            },
            {
                "name": "Bread",
                "description": "Bread with butter, cheese or jam",
                "foodgroups": ["Bread", "Butter", "Cheese", "Jam"]
            },
            {
                "name": "Roti for breakfast",
                "description": "Indian torilla with vegetables",
                "foodgroups": ["Grains", "Vegetables"],
            },
        ],
        "lunch": [
            {
                "name": "Froot Loops",
                "description": "A baggy of rainbow Froot Loops - a
classic college student lunch.",
                "foodgroups": ["Grains"]
            },
            {
                "name": "Rice",
                "description": "Rice with other complements",
                "foodgroups": ["Grains"]
            },
            {
                "name": "Spaghetti for lunch",
                "description": "Two pies made with love and filled full
of flavors. Blueberry, strawberry, you name it, you have it!",
                "foodgroups": ["Grains", "Fruit"]
            },
        ],
        "dinner": [

```

```

        {
            "name": "Spaghetti with Tomato Sauce",
            "description": "Two pies made with love and filled full
of flavors. Blueberry, strawberry, you name it, you have it!",
            "foodgroups": ["Grains", "Fruit"],
        },
        {
            "name": "Roti",
            "description": "Indian torilla with vegetables",
            "foodgroups": ["Grains", "Vegetables"],
        },
        {
            "name": "Rice for dinner",
            "description": "Rice with other complements",
            "foodgroups": ["Grains"],
        },
    ],
};

var i = 0;
var j = 0;
$scope.myvar = true;
$scope.fetchmenu = function () {
    $scope.myvar = false;
    $scope.fgps = "";
    // console.log($scope.foodmenu)
    // for (i in $scope.foodmenu) {
    // console.log(i);
    console.log($scope.foodmenu);

    var rand_index_breakfast = Math.floor((Math.random() * 3) + 0);
    console.log(rand_index_breakfast);
    $scope.rand_index_generated_breakfast = rand_index_breakfast;

    console.log($scope.foodmenu['breakfast']);
    console.log($scope.foodmenu['breakfast'][rand_index_breakfast])
;

    console.log($scope.foodmenu['breakfast'][rand_index_breakfast].name);
    $scope.breakfastname =
    $scope.foodmenu['breakfast'][rand_index_breakfast].name;
    $scope.breakfastdesc =
    $scope.foodmenu['breakfast'][rand_index_breakfast].description;
    $scope.breakfastgrps =
    $scope.foodmenu['breakfast'][rand_index_breakfast].foodgroups;

    var rand_index_lunch = Math.floor((Math.random() * 3) + 0);
    console.log(rand_index_lunch);
    $scope.rand_index_generated_lunch = rand_index_lunch;
}

```

```
        console.log($scope.foodmenu['lunch']);
        console.log($scope.foodmenu['lunch'][rand_index_breakfast]);
        console.log($scope.foodmenu['lunch'][rand_index_breakfast].name
);
        $scope.lunchname =
$scope.foodmenu['lunch'][rand_index_lunch].name;
        $scope.lunchdesc =
$scope.foodmenu['lunch'][rand_index_lunch].description;
        $scope.lunchgrps =
$scope.foodmenu['lunch'][rand_index_lunch].foodgroups;

        var rand_index_dinner = Math.floor((Math.random() * 3) + 0);
        console.log(rand_index_dinner);
        $scope.rand_index_generated_dinner = rand_index_dinner;

        console.log($scope.foodmenu['dinner']);
        console.log($scope.foodmenu['dinner'][rand_index_dinner]);
        console.log($scope.foodmenu['dinner'][rand_index_dinner].name);
        $scope.dinnername =
$scope.foodmenu['dinner'][rand_index_dinner].name;
        $scope.dinnerdesc =
$scope.foodmenu['dinner'][rand_index_dinner].description;
        $scope.dinnergrps =
$scope.foodmenu['dinner'][rand_index_dinner].foodgroups;
        // }
    }
});
</script>
</body>

</html>
```

Exercise 6

Dimensions: Responsive ▾ 1280 x 768 75% No throttling

HTML body

Styles Computed Layout Event Listeners

Filter :hover :cls +

Console Issues Network request blocking

Console top Filter

Default levels 1 Issue 1

Plan-It!

Click the button to generate a random 3-meal menu plan! [Fetch Random Menu](#)

Rand index generated breakfast:

Rand index generated lunch:

Rand index generated dinner:

See console.log for other details

**(Breakfast)**

Food Groups:

**(Lunch)**

Food Groups:

**(Dinner)**

Food Groups:

Exercise 6

Dimensions: Responsive ▾ 1280 x 768 75% ▾ No throttling ▾

100 200 300 400 500 600 700 800 900 1000 1100 1200

## Plan-It!

Click the button to generate a random 3-meal menu plan! [Fetch Random Menu](#)

Rand index generated breakfast: 1

Rand index generated lunch: 1

Rand index generated dinner: 2

See console.log for other details

### Bread (Breakfast)

Bread with butter, cheese or jam

Food Groups:

- Bread
- Butter
- Cheese
- Jam

### Rice (Lunch)

Rice with other complements

Food Groups:

- Grains

### Rice for dinner (Dinner)

Rice with other complements

Food Groups:

HTML body

Styles Computed Layout Event Listeners ▾

Filter :hover .cls + ▾

Console Issues Network request blocking

Default levels ▾ 1 Issue: 1

```
Os-6.html:118
↳ {breakfast: Array(3), lunch: Array(3), dinner: Array(3)}
1 Os-6.html:121
↳ (3) [{}]
2 Os-6.html:124
3 Os-6.html:125
↳ {name: 'Bread', description: 'Bread with butter, cheese or jam', foodgroups: Array(4)}
Bread Os-6.html:126
1 Os-6.html:132
2 Os-6.html:135
3 Os-6.html:136
↳ {name: 'Rice', description: 'Rice with other complements', foodgroups: Array(1)}
Rice Os-6.html:137
1 Os-6.html:143
2 Os-6.html:146
3 Os-6.html:147
↳ {name: 'Rice for dinner', description: 'Rice with other complements', foodgroups: Array(1)}
Rice for dinner Os-6.html:148

```

Exercise 6

127.0.0.1:5500/Web\_Tech\_A2:L17.18/WebTech\_L17\_18/Assessment-3/Qs-6.html

Dimensions: Responsive 1280 x 768 75% No throttling

Plan-It!

Click the button to generate a random 3-meal menu plan! Fetch Random Menu

Rand index generated breakfast: 0

Rand index generated lunch: 1

Rand index generated dinner: 2

See console.log for other details

**Blueberry Oatmeal (Breakfast)**

One cup of hot oatmeal with 1/4 cup of fresh blueberries.

Food Groups:

- Grains
- Fruity

**Rice (Lunch)**

Rice with other complements

Food Groups:

- Grains

**Rice for dinner (Dinner)**

Rice with other complements

Food Groups:

- Grains

Elements Console Network request blocking

html body

Styles Computed Layout Event Listeners

Filter : hover .cls +, □

Console Issues Network request blocking

Default levels 1 Issue: 1

```
1
Rice for dinner
Os-6.html:148
Os-6.html:118
↳ {breakfast: Array(3), lunch: Array(3), dinner: Array(3)}
0
↳ (3) [{}]
1
↳ (3) [{}]
2
↳ (3) [{}]
3
↳ (3) [{}]
4
↳ (3) [{}]
5
↳ (3) [{}]
6
↳ (3) [{}]
7
↳ (3) [{}]
8
↳ (3) [{}]
9
↳ (3) [{}]
10
↳ (3) [{}]
11
↳ (3) [{}]
12
↳ (3) [{}]
13
↳ (3) [{}]
14
↳ (3) [{}]
15
↳ (3) [{}]
16
↳ (3) [{}]
17
↳ (3) [{}]
18
↳ (3) [{}]
19
↳ (3) [{}]
20
↳ (3) [{}]
21
↳ (3) [{}]
22
↳ (3) [{}]
23
↳ (3) [{}]
24
↳ (3) [{}]
25
↳ (3) [{}]
26
↳ (3) [{}]
27
↳ (3) [{}]
28
↳ (3) [{}]
29
↳ (3) [{}]
30
↳ (3) [{}]
31
↳ (3) [{}]
32
↳ (3) [{}]
33
↳ (3) [{}]
34
↳ (3) [{}]
35
↳ (3) [{}]
36
↳ (3) [{}]
37
↳ (3) [{}]
38
↳ (3) [{}]
39
↳ (3) [{}]
40
↳ (3) [{}]
41
↳ (3) [{}]
42
↳ (3) [{}]
43
↳ (3) [{}]
44
↳ (3) [{}]
45
↳ (3) [{}]
46
↳ (3) [{}]
47
↳ (3) [{}]
48
↳ (3) [{}]
49
↳ (3) [{}]
50
↳ (3) [{}]
51
↳ (3) [{}]
52
↳ (3) [{}]
53
↳ (3) [{}]
54
↳ (3) [{}]
55
↳ (3) [{}]
56
↳ (3) [{}]
57
↳ (3) [{}]
58
↳ (3) [{}]
59
↳ (3) [{}]
60
↳ (3) [{}]
61
↳ (3) [{}]
62
↳ (3) [{}]
63
↳ (3) [{}]
64
↳ (3) [{}]
65
↳ (3) [{}]
66
↳ (3) [{}]
67
↳ (3) [{}]
68
↳ (3) [{}]
69
↳ (3) [{}]
70
↳ (3) [{}]
71
↳ (3) [{}]
72
↳ (3) [{}]
73
↳ (3) [{}]
74
↳ (3) [{}]
75
↳ (3) [{}]
76
↳ (3) [{}]
77
↳ (3) [{}]
78
↳ (3) [{}]
79
↳ (3) [{}]
80
↳ (3) [{}]
81
↳ (3) [{}]
82
↳ (3) [{}]
83
↳ (3) [{}]
84
↳ (3) [{}]
85
↳ (3) [{}]
86
↳ (3) [{}]
87
↳ (3) [{}]
88
↳ (3) [{}]
89
↳ (3) [{}]
90
↳ (3) [{}]
91
↳ (3) [{}]
92
↳ (3) [{}]
93
↳ (3) [{}]
94
↳ (3) [{}]
95
↳ (3) [{}]
96
↳ (3) [{}]
97
↳ (3) [{}]
98
↳ (3) [{}]
99
↳ (3) [{}]
100
↳ (3) [{}]
101
↳ (3) [{}]
102
↳ (3) [{}]
103
↳ (3) [{}]
104
↳ (3) [{}]
105
↳ (3) [{}]
106
↳ (3) [{}]
107
↳ (3) [{}]
108
↳ (3) [{}]
109
↳ (3) [{}]
110
↳ (3) [{}]
111
↳ (3) [{}]
112
↳ (3) [{}]
113
↳ (3) [{}]
114
↳ (3) [{}]
115
↳ (3) [{}]
116
↳ (3) [{}]
117
↳ (3) [{}]
118
↳ (3) [{}]
119
↳ (3) [{}]
120
↳ (3) [{}]
121
↳ (3) [{}]
122
↳ (3) [{}]
123
↳ (3) [{}]
124
↳ (3) [{}]
125
↳ (3) [{}]
126
↳ (3) [{}]
127
↳ (3) [{}]
128
↳ (3) [{}]
129
↳ (3) [{}]
130
↳ (3) [{}]
131
↳ (3) [{}]
132
↳ (3) [{}]
133
↳ (3) [{}]
134
↳ (3) [{}]
135
↳ (3) [{}]
136
↳ (3) [{}]
137
↳ (3) [{}]
138
↳ (3) [{}]
139
↳ (3) [{}]
140
↳ (3) [{}]
141
↳ (3) [{}]
142
↳ (3) [{}]
143
↳ (3) [{}]
144
↳ (3) [{}]
145
↳ (3) [{}]
146
↳ (3) [{}]
147
↳ (3) [{}]
148
↳ (3) [{}]
```

Exercise 6

127.0.0.1:5500/Web\_Tech\_A2:L17.18/WebTech\_L17\_18/Assessment-3/Qs-6.html

Dimensions: Responsive 1280 x 768 75% No throttling

Plan-It!

Click the button to generate a random 3-meal menu plan! Fetch Random Menu

Rand index generated breakfast: 0

Rand index generated lunch: 1

Rand index generated dinner: 0

See console.log for other details

**Blueberry Oatmeal (Breakfast)**

One cup of hot oatmeal with 1/4 cup of fresh blueberries.

Food Groups:

- Grains
- Fruity

**Rice (Lunch)**

Rice with other complements

Food Groups:

- Grains

**Spaghetti with Tomato Sauce (Dinner)**

Two pies made with love and filled full of flavors. Blueberry, strawberry, you name it, you have it!

Food Groups:

- Grains

Elements Console Network request blocking

html body

Styles Computed Layout Event Listeners

Filter : hover .cls +, □

Console Issues Network request blocking

Default levels 1 Issue: 1

```
1
Rice for dinner
Os-6.html:148
Os-6.html:118
↳ {breakfast: Array(3), lunch: Array(3), dinner: Array(3)}
0
↳ (3) [{}]
1
↳ (3) [{}]
2
↳ (3) [{}]
3
↳ (3) [{}]
4
↳ (3) [{}]
5
↳ (3) [{}]
6
↳ (3) [{}]
7
↳ (3) [{}]
8
↳ (3) [{}]
9
↳ (3) [{}]
10
↳ (3) [{}]
11
↳ (3) [{}]
12
↳ (3) [{}]
13
↳ (3) [{}]
14
↳ (3) [{}]
15
↳ (3) [{}]
16
↳ (3) [{}]
17
↳ (3) [{}]
18
↳ (3) [{}]
19
↳ (3) [{}]
20
↳ (3) [{}]
21
↳ (3) [{}]
22
↳ (3) [{}]
23
↳ (3) [{}]
24
↳ (3) [{}]
25
↳ (3) [{}]
26
↳ (3) [{}]
27
↳ (3) [{}]
28
↳ (3) [{}]
29
↳ (3) [{}]
30
↳ (3) [{}]
31
↳ (3) [{}]
32
↳ (3) [{}]
33
↳ (3) [{}]
34
↳ (3) [{}]
35
↳ (3) [{}]
36
↳ (3) [{}]
37
↳ (3) [{}]
38
↳ (3) [{}]
39
↳ (3) [{}]
40
↳ (3) [{}]
41
↳ (3) [{}]
42
↳ (3) [{}]
43
↳ (3) [{}]
44
↳ (3) [{}]
45
↳ (3) [{}]
46
↳ (3) [{}]
47
↳ (3) [{}]
48
↳ (3) [{}]
49
↳ (3) [{}]
50
↳ (3) [{}]
51
↳ (3) [{}]
52
↳ (3) [{}]
53
↳ (3) [{}]
54
↳ (3) [{}]
55
↳ (3) [{}]
56
↳ (3) [{}]
57
↳ (3) [{}]
58
↳ (3) [{}]
59
↳ (3) [{}]
60
↳ (3) [{}]
61
↳ (3) [{}]
62
↳ (3) [{}]
63
↳ (3) [{}]
64
↳ (3) [{}]
65
↳ (3) [{}]
66
↳ (3) [{}]
67
↳ (3) [{}]
68
↳ (3) [{}]
69
↳ (3) [{}]
70
↳ (3) [{}]
71
↳ (3) [{}]
72
↳ (3) [{}]
73
↳ (3) [{}]
74
↳ (3) [{}]
75
↳ (3) [{}]
76
↳ (3) [{}]
77
↳ (3) [{}]
78
↳ (3) [{}]
79
↳ (3) [{}]
80
↳ (3) [{}]
81
↳ (3) [{}]
82
↳ (3) [{}]
83
↳ (3) [{}]
84
↳ (3) [{}]
85
↳ (3) [{}]
86
↳ (3) [{}]
87
↳ (3) [{}]
88
↳ (3) [{}]
89
↳ (3) [{}]
90
↳ (3) [{}]
91
↳ (3) [{}]
92
↳ (3) [{}]
93
↳ (3) [{}]
94
↳ (3) [{}]
95
↳ (3) [{}]
96
↳ (3) [{}]
97
↳ (3) [{}]
98
↳ (3) [{}]
99
↳ (3) [{}]
100
↳ (3) [{}]
101
↳ (3) [{}]
102
↳ (3) [{}]
103
↳ (3) [{}]
104
↳ (3) [{}]
105
↳ (3) [{}]
106
↳ (3) [{}]
107
↳ (3) [{}]
108
↳ (3) [{}]
109
↳ (3) [{}]
110
↳ (3) [{}]
111
↳ (3) [{}]
112
↳ (3) [{}]
113
↳ (3) [{}]
114
↳ (3) [{}]
115
↳ (3) [{}]
116
↳ (3) [{}]
117
↳ (3) [{}]
118
↳ (3) [{}]
119
↳ (3) [{}]
120
↳ (3) [{}]
121
↳ (3) [{}]
122
↳ (3) [{}]
123
↳ (3) [{}]
124
↳ (3) [{}]
125
↳ (3) [{}]
126
↳ (3) [{}]
127
↳ (3) [{}]
128
↳ (3) [{}]
129
↳ (3) [{}]
130
↳ (3) [{}]
131
↳ (3) [{}]
132
↳ (3) [{}]
133
↳ (3) [{}]
134
↳ (3) [{}]
135
↳ (3) [{}]
136
↳ (3) [{}]
137
↳ (3) [{}]
138
↳ (3) [{}]
139
↳ (3) [{}]
140
↳ (3) [{}]
141
↳ (3) [{}]
142
↳ (3) [{}]
143
↳ (3) [{}]
144
↳ (3) [{}]
145
↳ (3) [{}]
146
↳ (3) [{}]
147
↳ (3) [{}]
148
↳ (3) [{}]
```

## Qs7

7. Develop a web application using AngularJS add behavior to the following HTML code. The page contains a text input box with an id of foodname, and a drop-down list with an id of foodgroup.
- The user types a name of a food item into the foodname text box, such as apple or Cookie, selects a food group from the drop-down list, such as dairy or fruit, and then clicks the button with an id of eat.
  - When the eat button is clicked, any element on the page that matches all of the following criteria will be removed from the page:
    - The element is an img element that has a class of the food.
    - The element's food group matches the group chosen in the foodgroup drop-down list. Food groups are represented as class attributes. This is in addition to the food class; recall that class attributes can specify multiple classes separated by spaces. For example, a jug of milk would have the following element:

```

```
    - The food item's name is the same as the text in the foodname box. The food's name is stored as the image's alt attribute. For example, if the user types cookie, img elements with an alt of cookie will be removed. Your code should be case-insensitive; for example, coOKIe should match images with an alt of cookie.

```
<!DOCTYPE html>
<html>

<head>
  <title>Exercise 7</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/angular.min.js"></script>

  <style>
    img {
      width: 250px;
    }
  </style>
</head>
```

```

<body ng-app="myApp" ng-controller="jsCtrl1">
  Name of Food: <input id="foodname" type="text" ng-model="foodname" />
  Food Group:
  <select id="foodgroup" ng-model="foodgroupdrop">
    <option value="dairy">dairy</option>
    <option value="fats">fats</option>
    <option value="fruit">fruit</option>
    <option value="meat">meat</option>
    <option value="veggies">veggies</option>
  </select>

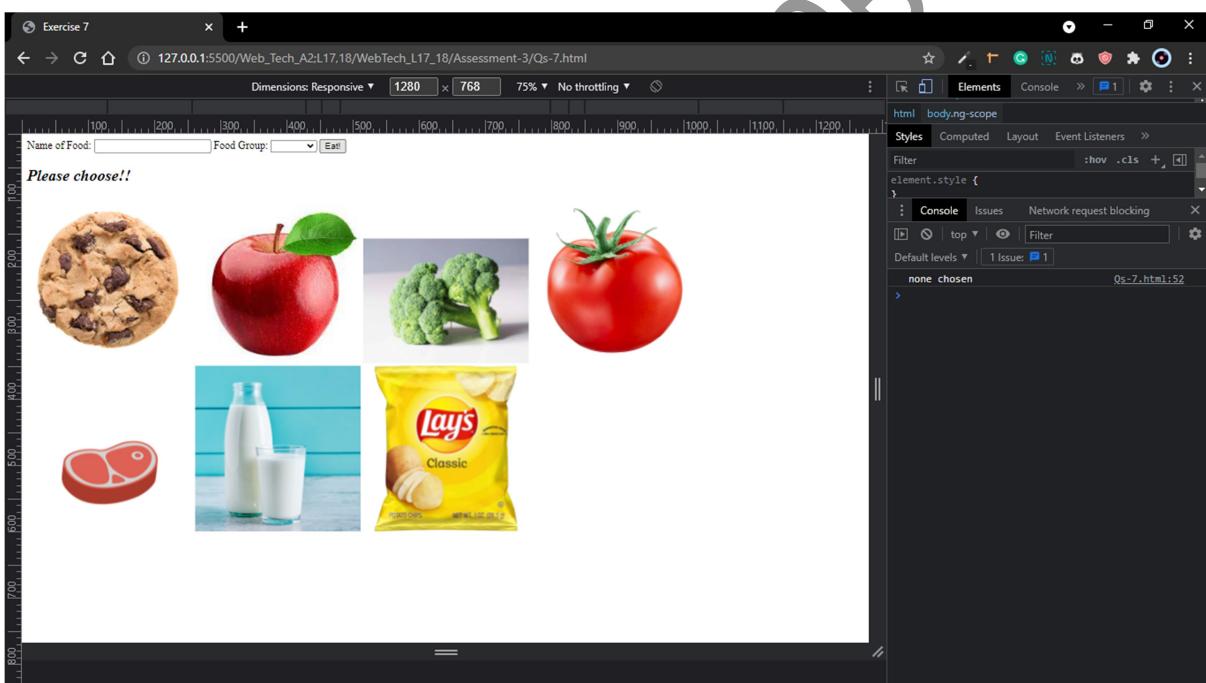
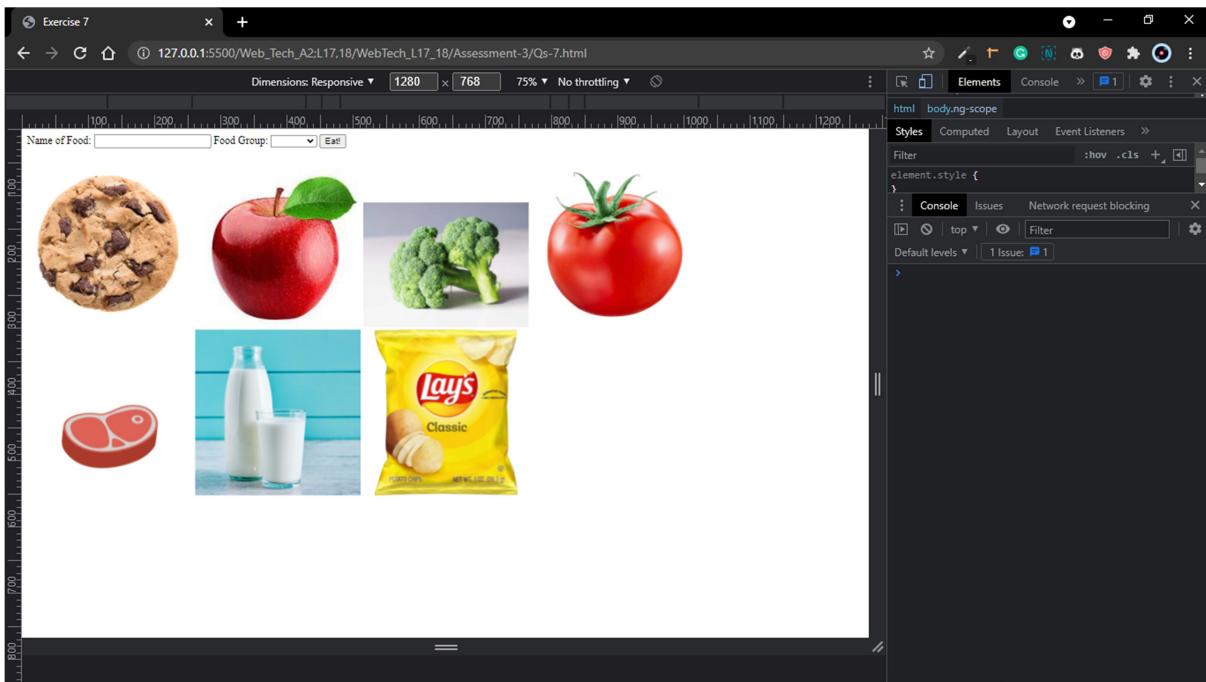
  <button id="eat" ng-click="remove()">Eat!</button>

  <h2><i>{{errorcombo}}</i></h2>
  <p>
    
    
    
    
    
    
    
  </p>
  <p>{{arrvalue}}</p>
  <script>
    var app = angular.module('myApp', []);
    app.controller('jsCtrl1', function ($scope) {
      $scope.foodname = "";
      $scope.foodgroupdrop = "";
      $scope.remove = function () {
        $scope.combinedchoice = $scope.foodname + $scope.foodgroupdrop;
        if ($scope.combinedchoice == "") {
          $scope.combinedchoice = "none chosen";
        }
        console.log($scope.combinedchoice);

        switch ($scope.combinedchoice) {
          case 'cookiefats':
            $scope.cookie = "true";

```

```
$scope.errorcombo = "";
break;
case 'applefruit':
$scope.apple = "true";
$scope.errorcombo = "";
break;
case 'tomatofruit':
$scope.tomato = "true";
$scope.errorcombo = "";
break;
case 'milkdairy':
$scope.milk = "true";
$scope.errorcombo = "";
break;
case 'chipsfats':
$scope.chips = "true";
$scope.errorcombo = "";
break;
case 'steakmeat':
$scope.steak = "true";
$scope.errorcombo = "";
break;
case 'none chosen':
$scope.errorcombo = "Please choose!!";
break;
default:
$scope.errorcombo = "Wrong combo!!";
}
});
});
</script>
</body>
</html>
```



Exercise 7

127.0.0.1:5500/Web\_Tech\_A2:L17.18/WebTech\_L17\_18/Assessment-3/Qs-7.html

Dimensions: Responsive 1280 x 768 75% No throttling

Name of Food: cookie Food Group: meat

Wrong combo!!

The screenshot shows a web browser window titled "Exercise 7" displaying a form with a "Name of Food" input field containing "cookie" and a "Food Group" dropdown set to "meat". A message "Wrong combo!!" is displayed above a grid of food images. The grid contains a cookie, an apple, broccoli, a tomato, a piece of meat (steak), a bottle of milk, and a bag of Lay's potato chips. The browser's developer tools are open, showing the element inspector with the class ".cookie" selected.

Exercise 7

127.0.0.1:5500/Web\_Tech\_A2:L17.18/WebTech\_L17\_18/Assessment-3/Qs-7.html

Dimensions: Responsive 1280 x 768 75% No throttling

Name of Food: cookie Food Group: fats

The screenshot shows a web browser window titled "Exercise 7" displaying a form with a "Name of Food" input field containing "cookie" and a "Food Group" dropdown set to "fats". The same message "Wrong combo!!" is displayed above a grid of food images. The grid contains an apple, broccoli, a tomato, a piece of meat (steak), a bottle of milk, and a bag of Lay's potato chips. The browser's developer tools are open, showing the element inspector with the class ".cookie" selected.