Master of Science (Computer Application) (M.Sc. CA) Programme

M.Sc. CA Sem - I
AY 2024-25

103-01 : Web Development Frameworks

by

Seat No.	Roll No.	Name of Student
	01	Jenil Nareshkumar Ambawala

Submitted to :
Prof. Kalpesh Patel

1) Calculate the Compound interest. All inputs should be textbox, calculate amount on all textbox change event. No need any button to click and calculate.

```
<!DOCTYPE html>
<html ng-app="compoundInterestApp">
<head>
  <title>Compound Interest Calculator</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 20px;
      background-color: #f4f4f4;
    }
    .container {
      max-width: 600px;
      margin: 0 auto;
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
    h1{
      text-align: center;
      color: #333;
    }
    label {
      display: block;
      margin: 10px 0 5px;
      font-weight: bold;
    }
    input[type="number"] {
      width: calc(100% - 22px);
      padding: 10px;
      margin-bottom: 20px;
      border: 1px solid #ddd;
      border-radius: 4px;
    }
    h2 {
      text-align: center;
      color: #333;
    }
    .result {
```

```
            background-color: #f9f9f9;
            border: 1px solid #ddd;
            padding: 10px;
            border-radius: 4px;
            margin: 10px 0;
        }
    </style>
</head>
<body ng-controller="InterestController">

    <div class="container">
        <h1>Compound Interest Calculator</h1>

        <label for="principal">Principal (P): </label>
        <input id="principal" type="number" ng-model="principal" />

        <label for="rate">Annual Interest Rate (r %): </label>
        <input id="rate" type="number" ng-model="rate" />

        <label for="timesCompounded">Number of Times Compounded per Year (n): </label>
        <input id="timesCompounded" type="number" ng-model="timesCompounded" />

        <label for="years">Number of Years (t): </label>
        <input id="years" type="number" ng-model="years" />

        <div class="result">
            <h2>Compound Interest: {{ compoundInterest | number:2 }}</h2>
            <h2>Total Amount: {{ totalAmount | number:2 }}</h2>
        </div>
    </div>

    <script>
        var app = angular.module('compoundInterestApp', []);

        app.controller('InterestController', function($scope) {
            $scope.$watchGroup(['principal', 'rate', 'timesCompounded', 'years'], function() {
                var P = parseFloat($scope.principal) || 0;
                var r = parseFloat($scope.rate) / 100 || 0;
                var n = parseFloat($scope.timesCompounded) || 1;
                var t = parseFloat($scope.years) || 0;

                var A = P * Math.pow((1 + r / n), (n * t));
                var CI = A - P;

                $scope.totalAmount = A;
                $scope.compoundInterest = CI;
```

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

</body>

</html>
```

Output:-

Compound Interest Calculator

Principal (P):

5000

Annual Interest Rate (r %):

12

Number of Times Compounded per Year (n):

2

Number of Years (t):

4

Compound Interest: 2,969.24

Total Amount: 7,969.24

2) Create order form with a total price updated in real time. Display four services with its name and price, it should be look like toggle button. When user click on one service price will be added/subtract in the Total.

```
<!DOCTYPE html>
<html ng-app="orderApp">
<head>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .container {
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      width: 300px;
      transition: all 0.3s ease;
    }
    .container:hover {
      box-shadow: 0 0 15px rgba(0, 0, 0, 0.2);
    }
    .service {
      display: flex;
      justify-content: space-between;
      align-items: center;
      padding: 10px;
      margin-bottom: 10px;
      border: 1px solid #ccc;
      border-radius: 4px;
      cursor: pointer;
      transition: background-color 0.3s ease, border-color 0.3s ease;
    }
```

```css
.service.active {
    background-color: #d4edda;
    border-color: #c3e6cb;
}
.service:hover {
    background-color: #e9f7ef;
}
.total {
    text-align: center;
    font-size: 1.2em;
    font-weight: bold;
    margin-top: 20px;
    color: #333;
}
</style>
</head>
<body ng-controller="OrderController">


    <div class="container">
                <h1>Order Form</h1>
                <h3>Please select service</h3>
        <div class="service" ng-repeat="service in services" ng-click="toggleService(service)" ng-class="{'active': service.active}">
            <span>{{service.name}}</span>
            <span>{{service.price | currency:"₹"}}</span>
        </div>
        <div class="total">
            Total: {{total | currency:"₹"}}
        </div>
    </div>

    <script>
        angular.module('orderApp', [])
        .controller('OrderController', ['$scope', function($scope) {
            $scope.services = [
                {name: 'Service 1', price: 500, active: false},
                {name: 'Service 2', price: 700, active: false},
                {name: 'Service 3', price: 400, active: false},
                {name: 'Service 4', price: 900, active: false}
            ];

            $scope.toggleService = function(service) {
                service.active = !service.active;
                $scope.calculateTotal();
            };
```

```
        $scope.calculateTotal = function() {
            $scope.total = $scope.services.reduce(function(sum, service) {
                return sum + (service.active ? service.price : 0);
            }, 0);
        };

        $scope.calculateTotal();
    }]);
  </script>
</body>

</html>
```

Output:-

3) Create student collection with ID, Student name and Blood group. List out all students in table format. Filter a list of students by typing into a search textbox.

```
<!DOCTYPE html>
<html ng-app="studentApp">
<head>
  <title>Student Collection</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .container {
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      width: 500px;
    }
    table {
      width: 100%;
      border-collapse: collapse;
      margin-top: 20px;
    }
    th, td {
      padding: 10px;
      text-align: left;
      border-bottom: 1px solid #ddd;
    }
    th {
      background-color: #f2f2f2;
    }
    input[type="text"] {
      width: 100%;
      padding: 8px;
      box-sizing: border-box;
```

```
        border: 1px solid #ccc;
        border-radius: 4px;
      }
      .no-results {
        text-align: center;
        color: #888;
        margin-top: 20px;
      }
    </style>
</head>
<body ng-controller="StudentController">
    <div class="container">
      <h2>Student Collection</h2>
      <input type="text" ng-model="searchText" placeholder="Search students...">
      <table>
        <thead>
          <tr>
            <th>ID</th>
            <th>Student Name</th>
            <th>Blood Group</th>
          </tr>
        </thead>
        <tbody>
          <tr ng-repeat="student in filteredStudents = (students | filter:searchText)">
            <td>{{student.id}}</td>
            <td>{{student.name}}</td>
            <td>{{student.bloodGroup}}</td>
          </tr>
        </tbody>
      </table>
      <div class="no-results" ng-if="filteredStudents.length === 0">
        Record is not available
      </div>
    </div>

    <script>
      angular.module('studentApp', [])
      .controller('StudentController', ['$scope', function($scope) {
        $scope.students = [
          {id: 1, name: 'John Doe', bloodGroup: 'A+'},
          {id: 2, name: 'Jane Smith', bloodGroup: 'B+'},
          {id: 3, name: 'Michael Johnson', bloodGroup: 'O-'},
          {id: 4, name: 'Emily Davis', bloodGroup: 'AB+'}
        ];
      }]);
    </script>
```

```
</body>
</html>
```

Output:-

Student Collection

Search students...

ID	Student Name	Blood Group
1	John Doe	A+
2	Jane Smith	B+
3	Michael Johnson	O-
4	Emily Davis	AB+

Student Collection

John Doe

ID	Student Name	Blood Group
1	John Doe	A+

4) Event management: Create the first page to listing all events with the delete link. Create second page to add new event. Fields: eventTitle, eventDate, fromTime, toTime, and description.

```html
<!DOCTYPE html>
<html ng-app="eventApp">
<head>
 <title>Event Management</title>
 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
 <script src="app.js"></script>
 <style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f9f9f9;
    margin: 20px;
  }
  h1, h2 {
    color: #333;
  }
  ul {
    list-style-type: none;
    padding: 0;
  }
  li {
    background-color: #fff;
    margin: 10px 0;
    padding: 10px;
    border: 1px solid #ddd;
    border-radius: 5px;
    display: flex;
    justify-content: space-between;
    align-items: center;
  }
  button {
    background-color: #ff6666;
    color: #fff;
    border: none;
    padding: 5px 10px;
    border-radius: 3px;
    cursor: pointer;
  }
  button:hover {
    background-color: #ff3333;
  }
  form {
```

```
      background-color: #fff;
      padding: 20px;
      border: 1px solid #ddd;
      border-radius: 5px;
    }
    label {
      display: block;
      margin: 10px 0 5px;
    }
    input, textarea {
      width: 100%;
      padding: 8px;
      margin-bottom: 10px;
      border: 1px solid #ddd;
      border-radius: 3px;
    }
    input[type="submit"] {
      background-color: #4CAF50;
      color: white;
      border: none;
      cursor: pointer;
    }
    input[type="submit"]:hover {
      background-color: #45a049;
    }
  </style>
</head>
<body ng-controller="EventController">
  <h1>Event Management</h1>

  <!-- Event List -->
  <ul>
    <li ng-repeat="event in events">
      {{event.eventTitle}} - {{event.eventDate}} ({{event.fromTime}} to {{event.toTime}})
      <button ng-click="deleteEvent($index)">Delete</button>
    </li>
  </ul>

  <!-- Add Event Form -->
  <h2>Add New Event</h2>
  <form ng-submit="addEvent()">
    <label>Event Title:</label>
    <input type="text" ng-model="newEvent.eventTitle" required>
    <br>
    <label>Event Date:</label>
    <input type="date" ng-model="newEvent.eventDate" required>
```

```
<br>
<label>From Time:</label>
<input type="time" ng-model="newEvent.fromTime" required>
<br>
<label>To Time:</label>
<input type="time" ng-model="newEvent.toTime" required>
<br>
<label>Description:</label>
<textarea ng-model="newEvent.description" required></textarea>
<br>
<input type="submit" value="Add Event">
</form>

<script>
  var app = angular.module('eventApp', []);

  app.controller('EventController', function($scope) {
    $scope.events = [
      { eventTitle: 'Event 1', eventDate: '2024-10-12', fromTime: '10:00', toTime: '12:00',
description: 'Description 1' },
      { eventTitle: 'Event 2', eventDate: '2024-10-13', fromTime: '14:00', toTime: '16:00',
description: 'Description 2' }
    ];

    $scope.addEvent = function() {
      $scope.events.push($scope.newEvent);
      $scope.newEvent = {};
    };

    $scope.deleteEvent = function(index) {
      $scope.events.splice(index, 1);
    };
  });
</script>
</body>
</html>
```

Output:-

Event Management

Event 1 - 2024-10-12 (10:00 to 12:00)   Delete

Event 2 - 2024-10-13 (14:00 to 16:00)   Delete

Add New Event

Event Title:

```
|
```

Event Date:

```
dd - mm - yyyy
```

From Time:

```
-- : -- --
```

To Time:

```
-- : -- --
```

Description:

```

```

Add Event

5) Create registration page. FirstName <Textbox>, LastName<Textbox>, Course<Dropdown> bind from array of course, AcceptTerms <Checkbox>. Apply appropriate validation with proper error message.

```html
<!DOCTYPE html>
<html ng-app="registrationApp">
<head>
   <title>Registration Form</title>
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>
<body ng-controller="RegistrationController">
   <div class="container">
      <h2>Registration Form</h2>
      <form name="registrationForm" ng-submit="submitForm()" novalidate>
         <div class="form-group">
            <label for="firstName">First Name:</label>
            <input type="text" class="form-control" id="firstName" name="firstName" ng-
model="user.firstName" required>
            <div ng-show="registrationForm.firstName.$touched &&
registrationForm.firstName.$invalid">
               <span class="text-danger">First Name is required.</span>
            </div>
         </div>
         <div class="form-group">
            <label for="lastName">Last Name:</label>
            <input type="text" class="form-control" id="lastName" name="lastName" ng-
model="user.lastName" required>
            <div ng-show="registrationForm.lastName.$touched &&
registrationForm.lastName.$invalid">
               <span class="text-danger">Last Name is required.</span>
            </div>
         </div>
         <div class="form-group">
            <label for="course">Course:</label>
            <select class="form-control" id="course" name="course" ng-model="user.course"
ng-options="course for course in courses" required>
               <option value="">Select a course</option>
            </select>
            <div ng-show="registrationForm.course.$touched &&
registrationForm.course.$invalid">
               <span class="text-danger">Course selection is required.</span>
            </div>
         </div>
```

```
      <div class="form-group">
        <label>
          <input type="checkbox" name="acceptTerms" ng-model="user.acceptTerms"
required> Accept Terms
        </label>
        <div ng-show="registrationForm.acceptTerms.$touched &&
registrationForm.acceptTerms.$invalid">
          <span class="text-danger">You must accept the terms.</span>
        </div>
      </div>
      <button type="submit" class="btn btn-primary" ng-
disabled="registrationForm.$invalid">Register</button>
    </form>
  </div>
  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script>


            angular.module('registrationApp', [])
            .controller('RegistrationController', function($scope) {
            $scope.courses = ['Mathematics', 'Physics', 'Chemistry', 'Biology', 'Computer
Science'];
            $scope.user = {};

            $scope.submitForm = function() {
                    if ($scope.registrationForm.$valid) {
                            alert('Form submitted successfully!');
    }
  };
});

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

Output:-

Registration Form

First Name:

Jenil

127.0.0.1:5500 says

Form submitted successfully!

OK

Last Name:

Ambawala

Course:

Computer Science

☑ Accept Terms

Register

Registration Form

First Name:

First Name is required.

Last Name:

Last Name is required.

Course:

Select a course

Course selection is required.

☐ Accept Terms

You must accept the terms.

Register

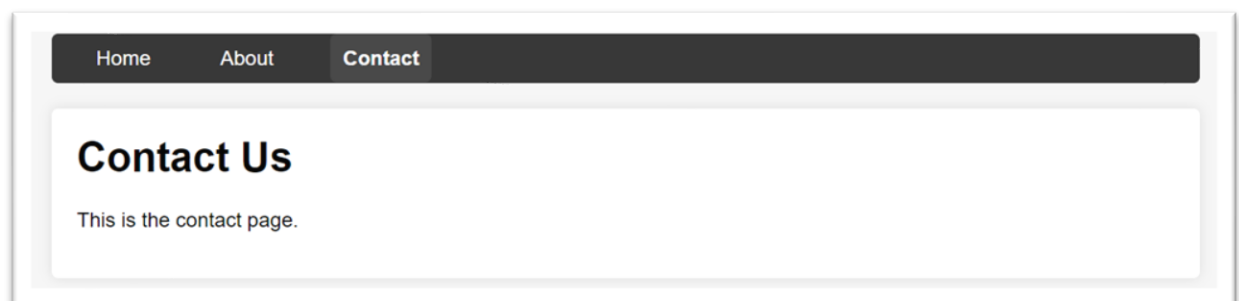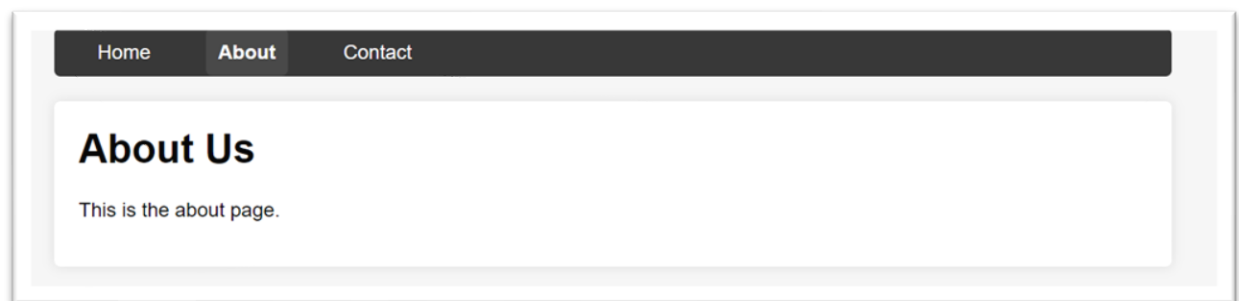**6) Create navigation menu (Home, About, Contact). Implement routing concept.**

```html
<!DOCTYPE html>
<html ng-app="myApp">
<head>
  <title>AngularJS Navigation Example</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular-route.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f4f4f4;
    }
    .container {
      max-width: 900px;
      margin: 0 auto;
      padding: 20px;
    }
    .nav-menu {
      background-color: #333;
      padding: 10px;
      border-radius: 5px;
      margin-bottom: 20px;
    }
    .nav-menu a {
      color: white;
      text-decoration: none;
      margin: 0 15px;
      padding: 10px;
      border-radius: 5px;
      transition: background-color 0.3s, font-weight 0.3s;
    }
    .nav-menu a:hover {
      background-color: #575757;
    }
    .nav-menu a.active {
      font-weight: bold;
      background-color: #444;
    }
    .content {
```

```
        padding: 20px;
        background-color: white;
        border-radius: 5px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      }
      h1 {
        margin-top: 0;
      }
    </style>
  </head>
  <body>
    <div ng-controller="MainCtrl" class="container">
      <div class="nav-menu">
        <a href="#!/home" ng-class="{active: isActive('/home')}">Home</a>
        <a href="#!/about" ng-class="{active: isActive('/about')}">About</a>
        <a href="#!/contact" ng-class="{active: isActive('/contact')}">Contact</a>
      </div>
      <div class="content" ng-view></div>
    </div>

    <script>
      // Define the AngularJS module and configure routes
      angular.module('myApp', ['ngRoute'])
        .config(function($routeProvider) {
          $routeProvider
            .when('/home', {
              template: '<h1>Welcome to the Home Page!</h1>'
            })
            .when('/about', {
              template: '<h1>About Us</h1><p>This is the about page.</p>'
            })
            .when('/contact', {
              template: '<h1>Contact Us</h1><p>This is the contact page.</p>'
            })
            .otherwise({
              redirectTo: '/home'
            });
        })
        .controller('MainCtrl', function($scope, $location) {
          $scope.isActive = function(route) {
            return $location.path() === route;
          };
        });
    </script>
  </body>
</html>
```

Output:-

7) Design your own student profile page with style and allow changing only profile picture. Uploaded picture must be stored in local and that picture will show in next execution.

```html
<!DOCTYPE html>
<html ng-app="profileApp">
<head>
  <title>Student Profile</title>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <style>
    body {
      background-color: #f8f9fa;
      font-family: Arial, sans-serif;
    }
    .profile-container {
      max-width: 400px;
      margin: 50px auto;
      padding: 20px;
      background-color: #ffffff;
      border-radius: 10px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      text-align: center;
    }
    .profile-pic {
      width: 150px;
      height: 150px;
      border-radius: 50%;
      object-fit: cover;
      border: 5px solid #007bff;
      margin-bottom: 20px;
    }
    .btn-primary {
      background-color: #007bff;
      border-color: #007bff;
    }
    .btn-primary:hover {
      background-color: #0056b3;
      border-color: #0056b3;
    }
    h2 {
      color: #333333;
      margin-bottom: 20px;
    }
    h3 {
```

```
            color: #007bff;
            margin-bottom: 10px;
        }
        p {
            color: #666666;
        }
    </style>
</head>
<body ng-controller="ProfileController">
    <div class="profile-container">
        <h2>Student Profile</h2>
        <img ng-src="{{profilePic}}" class="profile-pic" alt="Profile Picture">
        <input type="file" id="fileInput" style="display: none;"
onchange="angular.element(this).scope().uploadFile(this)">
        <button class="btn btn-primary" ng-click="changePicture()">Change Picture</button>
        <h3>{{student.name}}</h3>
        <p>{{student.course}}</p>
    </div>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
    <script>
        var app = angular.module('profileApp', []);

        app.controller('ProfileController', function($scope) {
            // Load profile picture from local storage
            $scope.profilePic = localStorage.getItem('profilePic') || 'default-abc.jpg';

            // Sample student data
            $scope.student = {
                name: 'Patel Tejash',
                course: 'Computer Science'
            };

            // Function to trigger file input
            $scope.changePicture = function() {
                document.getElementById('fileInput').click();
            };

            // Function to handle file upload
            $scope.uploadFile = function(element) {
                var reader = new FileReader();
                reader.onload = function(event) {
                    $scope.$apply(function() {
                        $scope.profilePic = event.target.result;
                        localStorage.setItem('profilePic', event.target.result);
                    });
```
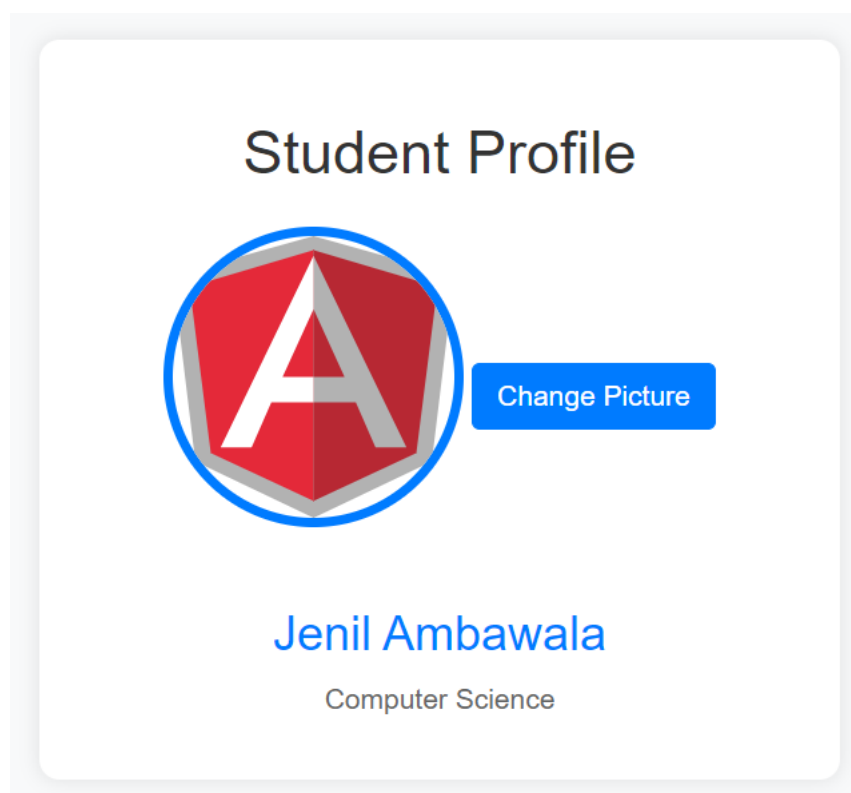
```
            };
            reader.readAsDataURL(element.files[0]);
        };
    });
  </script>
</body>
</html>
```

Output:-

8) Perform CRUD operation for user management with validation. Fields: id, empName, salary, isManager, two extra field for employee detail Operation: get(), post() and delete() API URL: https://beeceptor.com/crud-api/

```html
<!DOCTYPE html>

<html lang="en" ng-app="userApp">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>User Management</title>

    <link rel="stylesheet" href="styles.css">

    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

    <script>

        angular.module('userApp', [])

            .controller('UserController', function($scope) {

                $scope.users = [

                    { id: 1, empName: 'John Doe', salary: 50000 },

                    { id: 2, empName: 'Jane Smith', salary: 60000 }

                ];

                $scope.user = {};

                $scope.errorMessage = '';

                let userIdCounter = 3; // Counter for unique IDs


                $scope.createUser = function() {

                    if ($scope.user.empName && $scope.user.salary) {

                        if ($scope.user.id) {

                            // Update existing user

                            const index = $scope.users.findIndex(u => u.id === $scope.user.id);

                            $scope.users[index] = { ...$scope.user };

                        } else {

                            // Add new user
```

```
                    $scope.user.id = userIdCounter++;

                    $scope.users.push({ ...$scope.user });

                }

                $scope.user = {}; // Clear the form

                $scope.errorMessage = ''; // Clear error message

            } else {

                $scope.errorMessage = 'Please fill in all fields.';

            }

        };


        $scope.editUser = function(user) {

            $scope.user = angular.copy(user); // Copy user details to the form for editing

        };


        $scope.deleteUser = function(index) {

            $scope.users.splice(index, 1); // Remove user

        };

    });

</script>

<style>

    /* Same styles as before */

</style>

</head>

<body ng-controller="UserController">


    <h1>User Management</h1>


    <form ng-submit="createUser()">

      <div>

        <label for="empName">Employee Name:</label>

        <input type="text" id="empName" ng-model="user.empName" required>
```

```html
    </div>
    <div>
      <label for="salary">Salary:</label>
      <input type="number" id="salary" ng-model="user.salary" required>
    </div>
    <div>
      <button type="submit">{{ user.id ? 'Update User' : 'Add User' }}</button>
    </div>
    <div class="error">{{ errorMessage }}</div>
  </form>

  <h2>User List</h2>
  <table>
    <thead>
      <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Salary</th>
        <th>Actions</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="user in users">
        <td>{{ user.id }}</td>
        <td>{{ user.empName }}</td>
        <td>{{ user.salary }}</td>
        <td>
          <button ng-click="editUser(user)">Edit</button>
          <button ng-click="deleteUser($index)">Delete</button>
        </td>
      </tr>
```

                        </tbody>

                    </table>

                </body>

</html>

Output:-

User Management

Employee Name: [ ]
Salary: [ ]
[Add User]

User List

ID	Name	Salary	Actions	
1	Anaya Sharma	5000	[Edit]	[Delete]
2	Vraj	5000	[Edit]	[Delete]

9) Create ExpressJS API to manage books in library. "isbn" field is require and unique. "available" boolean filed to manage book availability. Take other 7 fields for book attribute. get(): /book get(): /book/3-540-96331-6 post() : /book put(): /book/3-540-96331-6 patch(): /book/3-540-96331-6/true delete: /book/3-540-96331-6

10) Create AngularJS application based on above library API.

(BOTH COMBINED BELOW).

```
Server.js

const express = require('express');

const mongoose = require('mongoose');

const bodyParser = require('body-parser');

const cors = require('cors');

const path = require('path');


const app = express();

const PORT = process.env.PORT || 3000;


app.use(cors());

app.use(bodyParser.json());

app.use(express.static(path.join(__dirname, 'public')));


mongoose.connect('mongodb+srv://vjariwala745:password123*@cluster0.oiefq.mongodb.net/',

  {

  useNewUrlParser: true,

  useUnifiedTopology: true,

});


const bookSchema = new mongoose.Schema({

  isbn: { type: String, required: true, unique: true },

  title: String,

  author: String,
```

```
    genre: String,

    publishedYear: Number,

    pages: Number,

    available: { type: Boolean, default: true },

    description: String,

});


const Book = mongoose.model('Book', bookSchema);


app.get('/book', async (req, res) => {

    const books = await Book.find();

    res.json(books);

});


app.get('/book/:isbn', async (req, res) => {

    const book = await Book.findOne({ isbn: req.params.isbn });

    if (book) {

        res.json(book);

    } else {

        res.status(404).send('Book not found');

    }

});


app.post('/book', async (req, res) => {

    const book = new Book(req.body);

    await book.save();

    res.status(201).json(book);

});


app.put('/book/:isbn', async (req, res) => {

    const book = await Book.findOneAndUpdate({ isbn: req.params.isbn }, req.body, { new:
true });
```

```
      if (book) {

        res.json(book);

      } else {

        res.status(404).send('Book not found');

      }

    });


    app.patch('/book/:isbn/:available', async (req, res) => {

      const book = await Book.findOneAndUpdate({ isbn: req.params.isbn }, { available:
    req.params.available === 'true' }, { new: true });

      if (book) {

        res.json(book);

      } else {

        res.status(404).send('Book not found');

      }

    });


    app.delete('/book/:isbn', async (req, res) => {

      const result = await Book.deleteOne({ isbn: req.params.isbn });

      if (result.deletedCount > 0) {

        res.status(204).send();

      } else {

        res.status(404).send('Book not found');

      }

    });


    app.listen(PORT, () => {

      console.log(`Server is running on http://localhost:${PORT}`);

    });


    Public/index.html

    <!DOCTYPE html>
```

```html
<html lang="en" ng-app="libraryApp">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Library Management</title>

  <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

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

</head>

<body ng-controller="BookController">

  <h1>Library Management</h1>


  <h2>Add Book</h2>

  <form ng-submit="addBook()">

    <input type="text" ng-model="newBook.isbn" placeholder="ISBN" required />

    <input type="text" ng-model="newBook.title" placeholder="Title" required />

    <input type="text" ng-model="newBook.author" placeholder="Author" />

    <input type="text" ng-model="newBook.genre" placeholder="Genre" />

    <input type="number" ng-model="newBook.publishedYear" placeholder="Published
Year" />

    <input type="number" ng-model="newBook.pages" placeholder="Pages" />

    <textarea ng-model="newBook.description" placeholder="Description"></textarea>

    <button type="submit">Add Book</button>

  </form>


  <h2>All Books</h2>

  <ul>

    <li ng-repeat="book in books">

      <strong>{{book.title}}</strong> by {{book.author}}

      <button ng-click="deleteBook(book.isbn)">Delete</button>

      <button ng-click="toggleAvailability(book)">{{book.available ? 'Mark Unavailable' :
'Mark Available'}}</button>
```

```
            </li>

        </ul>

</body>

</html>


Public/app.js

const app = angular.module('libraryApp', []);

app.controller('BookController', function ($scope, $http) {

    $scope.books = [];

    $scope.newBook = {};

    $scope.fetchBooks = function () {

        $http.get('/book').then(function (response) {

            $scope.books = response.data;

        });

    };

    $scope.addBook = function () {

        $http.post('/book', $scope.newBook).then(function (response) {

            $scope.books.push(response.data);

            $scope.newBook = {};

            $scope.fetchBooks();

        });

    };

    $scope.deleteBook = function (isbn) {

        $http.delete(`/book/${isbn}`).then(function () {

            $scope.fetchBooks();

        });

    };

    $scope.toggleAvailability = function (book) {

        book.available = !book.available;

        $http.patch(`/book/${book.isbn}/${book.available}`).then(function () {

            $scope.fetchBooks();
```

```
        });
    };

    $scope.fetchBooks();

});
```

Output:-