| **Experiment No. – 4** | | | | |
| --- | --- | --- | --- | --- |
| **Date of Performance:** | **29/1/25** | | | |
| **Date of Submission:** | **5/2/25** | | | |
| Program Execution/  formation/  correction/  ethical practices  (06) | Timely  Submission  (01) | Viva  (03) | Experiment  Total (10) | Sign with Date |
|  |  |  |  |  |

**Experiment No. 4**

**Title:** Design Web services using AngularJS Framework.

1. **Aim:** Design a single page Web application using AngularJS Framework.
2. **Course Outcome:** Understand how TypeScript and AngularJS framework can build dynamic, responsive single-page web applications
3. **Learning Objectives:** To understand the concept and implementation of AngularJS services for modular code design and to demonstrate how services help in sharing data and logic across controllers efficiently.
4. **Requirement:**

**Software Requirements:**

* **Operating System**: Windows/Linux/macOS
* **Text Editor / IDE**: Visual Studio Code, Sublime Text, or Atom
* **Browser**: Google Chrome, Mozilla Firefox (with developer tools)
* **Web Server**: Node.js (optional, for running a local server)
* **AngularJS Library**: AngularJS 1.x (included via CDN or local)

**Hardware Requirements:**

* **Processor**: Minimum 1 GHz processor (Intel or AMD)
* **RAM**: Minimum 2 GB (4 GB recommended)
* **Hard Disk**: At least 500 MB of free space
* **Display**: 1024x768 resolution or higher

1. Related Theory:

### **What is AngularJS?**

AngularJS is a structural JavaScript framework for dynamic web applications developed by Google. It allows you to use HTML as your template language and extend HTML's syntax to express your application's components clearly and succinctly.

### **Key Features**

* Two-way data binding
* Dependency injection
* Directives
* MVC architecture
* Single Page Application (SPA) support

### **Single Page Application (SPA)**

A Single Page Application is a web app that loads a single HTML page and dynamically updates the content as the user interacts with the app, without refreshing the page.

**Advantages of SPA:**

1. Faster Loading Time
2. Seamless User Experience
3. Reduced Server Load
4. Reusability of Code
5. Easier Development and Testing
6. **Procedure:**
7. Set up your project folder

* Create a folder for your project.
* Add an index.html file.

1. Include AngularJS

* Add AngularJS via CDN in the <head> of your HTML:  
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

1. Create AngularJS Application

* Initialize AngularJS app using ng-app and controller with ng-controller.

1. Design the HTML layout

* Use AngularJS expressions ({{ }}) and directives (ng-model, ng-click) to bind data and handle events.

1. Add functionality

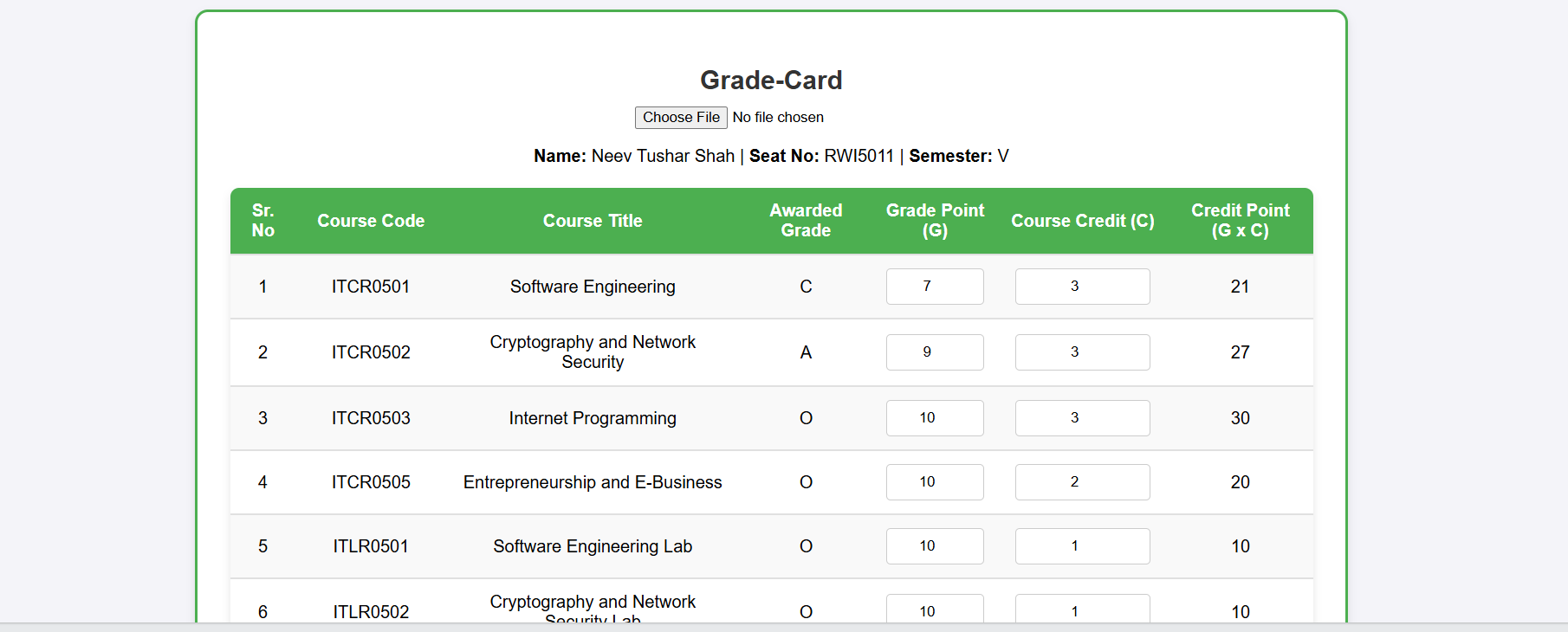
* Define your module and controller in a <script> tag or separate app.js file.

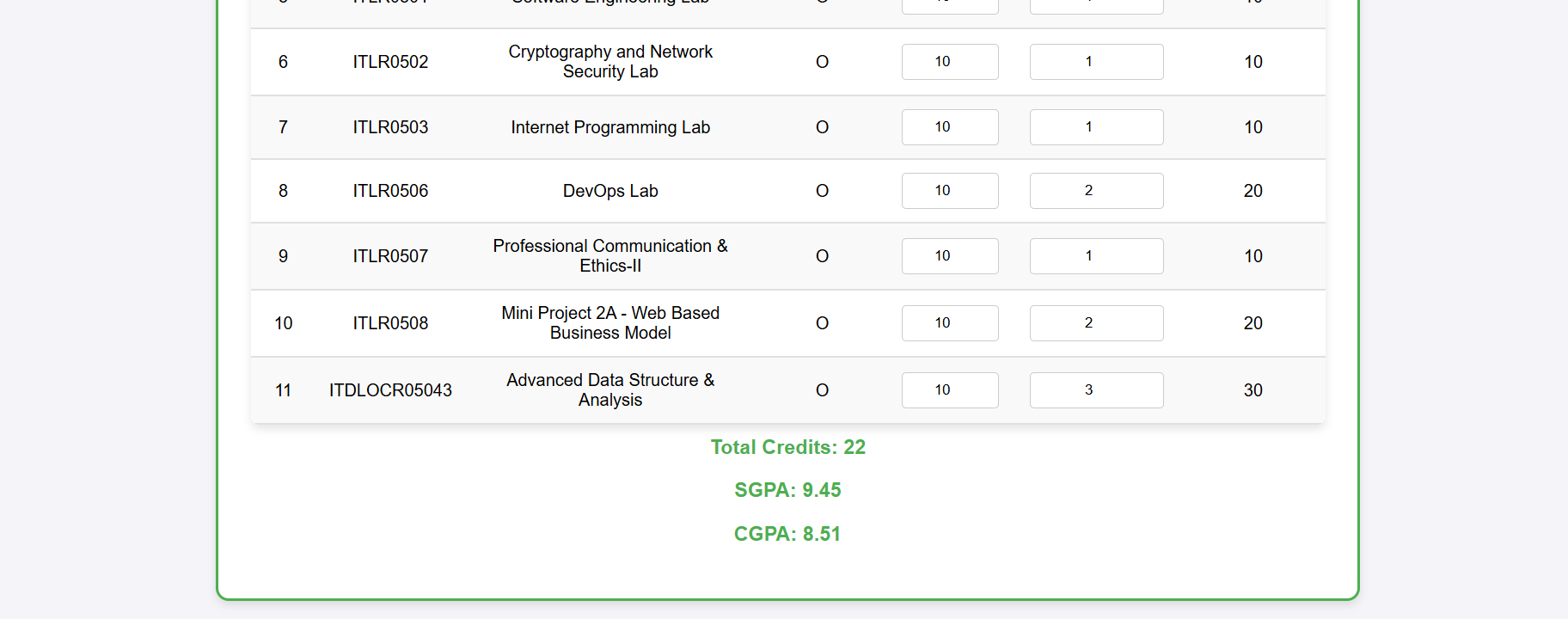
1. Run the application

* Open index.html in a browser to see your SPA in action.

1. **Program and Output:**

| <!DOCTYPE html>  <html ng-app="resultApp">  <head>  <title>Result App</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: #f4f4f9;  margin: 0;  padding: 20px;  display: flex;  justify-content: center;  align-items: center;  }  .result-container {  background-color: #ffffff;  width: 90%;  max-width: 1000px;  border-radius: 12px;  box-shadow: 0 4px 8px rgba(0,0,0,0.1);  padding: 30px;  text-align: center;  border: 3px solid #4CAF50;  }  table {  width: 100%;  border-collapse: collapse;  margin-top: 20px;  border-radius: 8px;  overflow: hidden;  box-shadow: 0 4px 8px rgba(0,0,0,0.1);  }  th, td {  padding: 12px;  text-align: center;  border-bottom: 2px solid #e0e0e0;  }  th {  background-color: #4CAF50;  color: white;  font-weight: bold;  }  tr:nth-child(even) {  background-color: #f9f9f9;  }  input[type="number"] {  width: 80%;  padding: 8px;  border: 1px solid #ccc;  border-radius: 4px;  text-align: center;  outline: none;  transition: border-color 0.3s;  }  input[type="number"]:focus {  border-color: #4CAF50;  }  img {  width: 120px;  height: 120px;  object-fit: cover;  border: 2px solid #4CAF50;  border-radius: 50%;  margin-bottom: 10px;  }  h2 {  color: #333;  margin-bottom: 10px;  }  h3 {  color: #4CAF50;  margin-top: 10px;  }  </style>  </head>  <body ng-controller="ResultController">  <div class="result-container">  <h2>Grade-Card</h2>    <input type="file" accept="image/\*" onchange="angular.element(this).scope().uploadPhoto(event)" />  <div ng-if="student.photo">  <img ng-src="{{ student.photo }}" alt="Student Photo">  </div>    <p><strong>Name:</strong> {{ student.name }} | <strong>Seat No:</strong> {{ student.seatNo }} | <strong>Semester:</strong> {{ student.semester }}</p>    <table>  <tr>  <th>Sr. No</th>  <th>Course Code</th>  <th>Course Title</th>  <th>Awarded Grade</th>  <th>Grade Point (G)</th>  <th>Course Credit (C)</th>  <th>Credit Point (G x C)</th>  </tr>  <tr ng-repeat="subject in subjects">  <td>{{ $index + 1 }}</td>  <td>{{ subject.code }}</td>  <td>{{ subject.title }}</td>  <td>{{ subject.grade }}</td>  <td><input type="number" ng-model="subject.gradePoint" min="0" max="10"></td>  <td><input type="number" ng-model="subject.credit" min="0"></td>  <td>{{ subject.gradePoint \* subject.credit }}</td>  </tr>  </table>    <h3>Total Credits: {{ calculateTotalCredits() }}</h3>  <h3>SGPA: {{ calculateSGPA() | number:2 }}</h3>  <h3>CGPA: {{ calculateCGPA() | number:2 }}</h3>  </div>    <script>  angular.module('resultApp', []).controller('ResultController', function($scope) {  $scope.student = {  name: 'Neev Tushar Shah',  seatNo: 'RWI5011',  semester: 'V',  photo: ''  };    $scope.subjects = [  { code: 'ITCR0501', title: 'Software Engineering', grade: 'C', gradePoint: 7, credit: 3 },  { code: 'ITCR0502', title: 'Cryptography and Network Security', grade: 'A', gradePoint: 9, credit: 3 },  { code: 'ITCR0503', title: 'Internet Programming', grade: 'O', gradePoint: 10, credit: 3 },  { code: 'ITCR0505', title: 'Entrepreneurship and E-Business', grade: 'O', gradePoint: 10, credit: 2 },  { code: 'ITLR0501', title: 'Software Engineering Lab', grade: 'O', gradePoint: 10, credit: 1 },  { code: 'ITLR0502', title: 'Cryptography and Network Security Lab', grade: 'O', gradePoint: 10, credit: 1 },  { code: 'ITLR0503', title: 'Internet Programming Lab', grade: 'O', gradePoint: 10, credit: 1 },  { code: 'ITLR0506', title: 'DevOps Lab', grade: 'O', gradePoint: 10, credit: 2 },  { code: 'ITLR0507', title: 'Professional Communication & Ethics-II', grade: 'O', gradePoint: 10, credit: 1 },  { code: 'ITLR0508', title: 'Mini Project 2A - Web Based Business Model', grade: 'O', gradePoint: 10, credit: 2 },  { code: 'ITDLOCR05043', title: 'Advanced Data Structure & Analysis', grade: 'O', gradePoint: 10, credit: 3 }  ];    $scope.calculateTotalCredits = function() {  return $scope.subjects.reduce((sum, subject) => sum + subject.credit, 0);  };    $scope.calculateSGPA = function() {  let totalCreditPoints = 0;  let totalCredits = 0;  $scope.subjects.forEach(subject => {  totalCreditPoints += subject.gradePoint \* subject.credit;  totalCredits += subject.credit;  });  return totalCredits ? totalCreditPoints / totalCredits : 0;  };    $scope.calculateCGPA = function() {  let previousCGPA = 8.04;  let previousSemesters = 2;  let currentSGPA = $scope.calculateSGPA();  let totalSemesters = previousSemesters + 1;  return ((previousCGPA \* previousSemesters) + currentSGPA) / totalSemesters;  };    $scope.uploadPhoto = function(event) {  let reader = new FileReader();  reader.onload = function(e) {  $scope.$apply(function() {  $scope.student.photo = e.target.result;  });  };  reader.readAsDataURL(event.target.files[0]);  };  });  </script>  </body>  </html> |
| --- |

****

****

1. **Conclusion:**

In this experiment by following the above steps, a single-page web application can be effectively developed using AngularJS. The framework simplifies the process of building dynamic and interactive web interfaces by using data binding and MVC concepts, reducing the need for manually manipulating the DOM or writing complex JavaScript code.

1. **Questions:**
2. **What is a Single Page Application (SPA), and how does AngularJS help in building it?**

**Answer:** A Single Page Application (SPA) is a web application that loads a single HTML page and dynamically updates content without refreshing the page. AngularJS supports SPA development through features like two-way data binding, routing, and directives, which allow dynamic content updates and component-based architecture. It enables faster, smoother user interactions within a single browser window.

1. **What are the key components of an AngularJS application used to create a SPA?  
   Answer:** Key components of an AngularJS application used in SPA development include:

* ng-app: Initializes the AngularJS application.
* ng-controller: Defines a controller for a section of the UI.
* ng-model: Binds input fields to variables in the controller.
* $scope: Links the controller and the view (HTML).

These components together enable dynamic, interactive, and modular web page creation.

### **How does AngularJS manage navigation in a single-page application without reloading the page?** **Answer:** AngularJS uses the ngRoute module (or third-party modules like ui-router) to handle client-side routing. It defines routes using $routeProvider, mapping URLs to templates and controllers. When the user navigates, AngularJS dynamically loads the corresponding view and controller into a single HTML page using ng-view, without reloading the page from the server.

### 