## Name :Tejas Patel

## Class :M.s.c (Computer Application )

## Roll No: 07

**Example-1 : Write a code in angularjs to display "hello world".**

Create an HTML file (e.g., index.html)

<!DOCTYPE html>

<html ng-app="myApp">

<head>

<title>Hello World Example</title>

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

</head>

<body>

<div ng-controller="HelloWorldController">

<h1>{{ greeting }}</h1>

</div>

<script>

var app = angular.module("myApp", []);

app.controller("HelloWorldController", function ($scope) {

$scope.greeting = "Hello World";

});

</script>

</body>

</html>

**Example-2 : Display the factorial of a given number using AngularJS.**

<!DOCTYPE html>

<html ng-app="FactorialApp">

<head>

<title>Factorial Calculator</title>

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

</head>

<body>

<div ng-controller="FactorialController">

<h1>Factorial Calculator</h1>

<div>

<label for="numberInput">Enter a number:</label>

<input type="number" id="numberInput" ng-model="number" />

<button ng-click="calculateFactorial()">Calculate Factorial</button>

</div>

<div ng-show="result !== null">

<p ng-if="number < 0">Factorial is not defined for negative numbers.</p>

<p ng-if="number === 0 || number === 1">Factorial of {{ number }} is 1.</p>

<p ng-if="number > 1">Factorial of {{ number }} is {{ result }}.</p>

</div>

</div>

<script>

var app = angular.module("FactorialApp", []);

app.controller("FactorialController", function($scope) {

$scope.number = null;

$scope.result = null;

$scope.calculateFactorial = function() {

var num = parseInt($scope.number);

if (num < 0) {

$scope.result = null;

} else if (num === 0 || num === 1) {

$scope.result = 1;

} else {

var factorial = 1;

for (var i = 2; i <= num; i++) {

factorial \*= i;

}

$scope.result = factorial;

}

};

});

</script>

</body>

</html>

**Example-3 : Accept name, age and city and display them using AngularJS.**

Create an HTML file user-info.html as follows:

<!DOCTYPE html>

<html ng-app="UserInfoApp">

<head>

<title>User Information</title>

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

</head>

<body>

<div ng-controller="UserInfoController">

<h1>User Information</h1>

<div>

<label for="nameInput">Name:</label>

<input type="text" id="nameInput" ng-model="user.name" />

</div>

<div>

<label for="ageInput">Age:</label>

<input type="number" id="ageInput" ng-model="user.age" />

</div>

<div>

<label for="cityInput">City:</label>

<input type="text" id="cityInput" ng-model="user.city" />

</div>

<button ng-click="displayUserInfo()">Display User Info</button>

<div ng-show="showInfo">

<p>Name: {{ user.name }}</p>

<p>Age: {{ user.age }}</p>

<p>City: {{ user.city }}</p>

</div>

</div>

<script>

var app = angular.module("UserInfoApp", []);

app.controller("UserInfoController", function($scope) {

$scope.user = {

name: "",

age: null,

city: ""

};

$scope.showInfo = false;

$scope.displayUserInfo = function() {

$scope.showInfo = true;

};

});

</script>

</body>

</html>

**Example-4 : Display serial number, name and age of employees from**

Step-1 : Create an HTML file employee-list.html as shown below.

Step-2 : Create a CSV file named employees.csv with sample data as shown below:

Name, Age

Evaan, 18

Rutvik, 22

Pankti, 20

Pranali, 28

Step-3: Place both the HTML file (employee-list.html) and the CSV file (employees.csv) in

the same directory.

<!DOCTYPE html>

<html ng-app="EmployeeApp">

<head>

<title>Employee List</title>

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

</head>

<body>

<div ng-controller="EmployeeController">

<h1>Employee List</h1>

<table>

<thead>

<tr>

<th>Serial Number</th>

<th>Name</th>

<th>Age</th>

</tr>

</thead>

<tbody>

<tr ng-repeat="employee in employees">

<td>{{ $index + 1 }}</td>

<td>{{ employee.name }}</td>

<td>{{ employee.age }}</td>

</tr>

</tbody>

</table>

</div>

<script>

var app = angular.module("EmployeeApp", []);

app.controller("EmployeeController", function($scope, $http) {

$http.get('employees.csv').then(function(response) {

var csvData = response.data;

var employees = [];

var rows = csvData.split('\n');

for (var i = 1; i < rows.length; i++) {

var row = rows[i].split(',');

if (row.length >= 3) {

employees.push({

name: row[0],

age: row[1]

});

}

}

$scope.employees = employees;

});

});

</script>

</body>

</html>

**Example-5 : Accept Roll\_Number, Name and Age of students through textbox**

<!DOCTYPE html>

<html ng-app="studentApp">

<head>

<title>Student Information Form</title>

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

</head>

<body ng-controller="StudentController">

<h2>Student Information Form</h2>

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

<label for="rollNumber">Roll Number:</label>

<input type="text" id="rollNumber" ng-model="student.rollNumber" required><br><br>

<label for="name">Name:</label>

<input type="text" id="name" ng-model="student.name" required><br><br>

<label for="age">Age:</label>

<input type="number" id="age" ng-model="student.age" required><br><br>

<button type="submit">Submit</button>

<button type="button" ng-click="clearForm()">Clear</button>

</form>

<div ng-show="message" ng-bind="message"></div>

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

</body>

</html>

from flask import Flask, request, jsonify

import csv

app = Flask(\_\_name)

@app.route('/save-student', methods=['POST'])

def save\_student():

try:

student = request.get\_json()

with open('student\_data.csv', 'a', newline='') as file:

writer = csv.writer(file)

writer.writerow([student['rollNumber'], student['name'], student['age']])

return jsonify({'message': 'Student data saved successfully.'})

except Exception as e:

return jsonify({'message': 'An error occurred while saving student data.'})

if \_\_name\_\_ == '\_\_main\_\_':

app.run()

---------------------------------------------------------------------

<!DOCTYPE html>

<html ng-app="studentApp">

<head>

<title>Student Information Form</title>

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

</head>

<body ng-controller="StudentController">

<h2>Student Information Form</h2>

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

<label for="rollNumber">Roll Number:</label>

<input type="text" id="rollNumber" ng-model="student.rollNumber" required><br><br>

<label for="name">Name:</label>

<input type="text" id="name" ng-model="student.name" required><br><br>

<label for="age">Age:</label>

<input type="number" id="age" ng-model="student.age" required><br><br>

<button type="submit">Submit</button>

<button type="button" ng-click="clearForm()">Clear</button>

</form>

<div ng-show="message" ng-bind="message"></div>

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

</body>

</html>

---------------------------------------------------------------------

Step-2:

Code to create app.js :

---------------------------------------------------------------------

var app = angular.module('studentApp', []);

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

$scope.student = {};

$scope.submitForm = function () {

// Data validation

if (!$scope.student.rollNumber || !$scope.student.name || !$scope.student.age) {

$scope.message = "Please fill in all fields.";

return;

}

// Send the student data to the server

$http.post('/save-student', $scope.student)

.then(function (response) {

$scope.message = response.data.message;

$scope.clearForm();

})

.catch(function (error) {

$scope.message = "An error occurred while saving student data.";

});

};

$scope.clearForm = function () {

$scope.student = {};

$scope.message = '';

};

});

---------------------------------------------------------------------

Step-3:

------------------------------------------------------------------

const express = require('express');

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

const fs = require('fs');

const app = express();

const port = 5000;

app.use(bodyParser.json());

app.post('/save-student', (req, res) => {

const student = req.body;

// Data validation

if (!student.rollNumber || !student.name || !student.age) {

return res.status(400).json({ message: 'Please fill in all fields.' });

}

// Save the student data to a CSV file

const csvRow = `${student.rollNumber},${student.name},${student.age}\n`;

fs.appendFile('student\_data.csv', csvRow, (err) => {

if (err) {

return res.status(500).json({ message: 'An error occurred while saving student data.' });

}

return res.json({ message: 'Student data saved successfully.' });

});

});

app.listen(port, () => {

console.log(`Server is running on port ${port}`);

});

---------------------------------------------------------------------

Step-4:

Step-1: Set up HTML file name “index.html” :

---------------------------------------------------------------------

<!DOCTYPE html>

<html ng-app="loginApp">

<head>

<title>Login Page</title>

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

</head>

<body ng-controller="LoginController">

<h2>Login Page</h2>

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

<label for="userId">User ID:</label>

<input type="text" id="userId" ng-model="credentials.userId" required><br><br>

<label for="password">Password:</label>

<input type="password" id="password" ng-model="credentials.password" required><br><br>

<button type="submit">Login</button>

</form>

<div ng-show="message" ng-bind="message"></div>

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

</body>

</html>

---------------------------------------------------------------------

Step-2: Create an AngularJS application and controller in “app.js” file:

---------------------------------------------------------------------

var app = angular.module('loginApp', []);

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

$scope.credentials = {};

$scope.login = function () {

// Simulate user authentication by sending a request to the server

ANGULARJS PRACTICAL NOTES : BY DR. SNEHAL K JOSHI

By: Dr. Snehal K Joshi |

$http.post('/authenticate', $scope.credentials)

.then(function (response) {

if (response.data.authenticated) {

$scope.message = 'Login successful.';

} else {

$scope.message = 'Invalid user ID or password.';

}

})

.catch(function (error) {

$scope.message = 'An error occurred while logging in.';

});

};

});

---------------------------------------------------------------------

Step-3: Create server.js file as shown below that create a simple server to handle the

authentication in Node.js using Express.

Make sure that Node.js is installed on your server. If it is not installed then you can install the

required packages using npm (npm install express body-parser).

Also create a csv filed named 'user\_credentials.csv' that contains userid and password.

---------------------------------------------------------------------

const express = require('express');

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

const fs = require('fs');

const app = express();

const port = 5000;

app.use(bodyParser.json());

// Read the CSV file where user credentials are stored

const userCredentials = fs.readFileSync('user\_credentials.csv', 'utf-8').split('\n');

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

const credentials = req.body;

// Data validation

if (!credentials.userId || !credentials.password) {

return res.status(400).json({ authenticated: false });

}

// Check if the provided credentials match any row in the CSV file

for (const line of userCredentials) {

const [userId, password] = line.split(',');

if (userId === credentials.userId && password === credentials.password) {

return res.json({ authenticated: true });

}

}

return res.json({ authenticated: false });

});

app.listen(port, () => {

console.log(`Server is running on port ${port}`);

});

---------------------------------------------------------------------

Step-4: Now run the Node.js server using the following command:

---------------------------------------------------------------------

node server.js

---------------------------------------------------------------------

**Example-8 : Write an AngularJS program that demonstrates data binding.**

<!DOCTYPE html>

<html ng-app="dataBindingApp">

<head>

<title>Data Binding Example</title>

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

</head>

<body ng-controller="DataBindingController">

<h2>Data Binding Example</h2>

<form>

<label for="name">Name:</label>

<input type="text" id="name" ng-model="name" placeholder="Enter your name" />

</form>

<br>

<label>Entered Name: {{ name }}</label>

<script>

var app = angular.module('dataBindingApp', []);

app.controller('DataBindingController', function ($scope) {

$scope.name = ""; // Initialize the name variable

// You can add more logic here if needed

});

</script>

</body>

</html>

**Example-9 : Write angularJS code that ask the user to create a custom**

<!DOCTYPE html>

<html ng-app="customDirectiveApp">

<head>

<title>Custom Directive Example</title>

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

</head>

<body>

<h2>Custom Directive Example</h2>

<div ng-controller="MainController">

<div custom-bg-color>Hover over me to change my background color</div>

</div>

<script>

var app = angular.module('customDirectiveApp', []);

app.directive('customBgColor', function () {

return {

restrict: 'A', // Attribute directive

link: function (scope, element) {

element.on('mouseover', function () {

element.css('background-color', 'lightblue');

});

element.on('mouseout', function () {

element.css('background-color', '');

});

}

};

});

app.controller('MainController', function ($scope) {

// Controller logic can be added here if needed

});

</script>

</body>

</html>

**Example-10 : What is the principles of dependency injection and**

**how to structure a service?**

angular.module('myApp', [])

.service('myService', function() {

// Service logic and properties

this.data = [];

// Service method to add an item

this.addItem = function(item) {

this.data.push(item);

};

});

------------------------------------------------------------

**Example-11 : Write a code to build a service that fetches data from**

<!DOCTYPE html>

<html ng-app="bookApp">

ANGULARJS PRACTICAL NOTES : BY DR. SNEHAL K JOSHI

By: Dr. Snehal K Joshi |

<head>

<title>Book List</title>

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

</head>

<body ng-controller="BookController">

<h2>Book List</h2>

<ul>

<li ng-repeat="book in books">{{ book.title }}</li>

</ul>

<script>

var app = angular.module('bookApp', []);

app.service('BookService', function($http) {

this.getBooks = function() {

return $http.get('https://jsonplaceholder.typicode.com/posts'); // Replace with your API

endpoint

};

});

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

$scope.books = [];

BookService.getBooks()

.then(function(response) {

$scope.books = response.data;

})

.catch(function(error) {

console.error('Error fetching data:', error);

});

});

</script>

</body>

</html>

**Example-12 : Create a multi-page application using AngularJS routing.**

i) HTML Structure:

<!DOCTYPE html>

<html ng-app="multiPageApp">

<head>

<title>Multi-Page App</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>

</head>

<body>

<h2>Multi-Page Application</h2>

<ul>

<li><a href="#/home">Home</a></li>

<li><a href="#/about">About</a></li>

<li><a href="#/contact">Contact</a></li>

</ul>

<div ng-view></div>

<script>

var app = angular.module('multiPageApp', ['ngRoute']);

</script>

</body>

</html>

-------------------------------------------------------------------------------

ii) Route Configuration:

app.config(function ($routeProvider) {

$routeProvider

ANGULARJS PRACTICAL NOTES : BY DR. SNEHAL K JOSHI

By: Dr. Snehal K Joshi |

.when('/home', {

templateUrl: 'home.html',

})

.when('/about', {

templateUrl: 'about.html',

})

.when('/contact', {

templateUrl: 'contact.html',

})

.otherwise({

redirectTo: '/home',

});

});

-------------------------------------------------------------------------------

iii) Create Templates:

.when('/home', {

templateUrl: 'home.html',

controller: 'HomeController',

})

-------------------------------------------------------------------------------

v)Controller Logic (Optional):

app.controller('HomeController', function ($scope) {

// Controller logic for the Home page

});

vi)Testing:

**Example-13 : Create a registration form that includes various form fields**

**such as name, email, and password. The form should demonstrate how to**

**validate the form inputs and display error messages for invalid data. Ensure**

**that they use AngularJS's built-in validation features..**

Answer:

Step-1: HTML Form:

Set up the HTML form with various form fields and include AngularJS in your page.

---------------------------------------------------------------------

<!DOCTYPE html>

<html ng-app="registrationApp">

<head>

<title>Registration Form</title>

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

</head>

<body ng-controller="RegistrationController">

<h2>Registration Form</h2>

<form name="registrationForm" ng-submit="submitForm()" novalidate>

<label for="name">Name:</label>

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

<span class="error" ng-show="registrationForm.name.$invalid &&

registrationForm.name.$touched">Name is required</span><br><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" ng-model="user.email" required>

<span class="error" ng-show="registrationForm.email.$invalid &&

registrationForm.email.$touched">Invalid email</span><br><br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" ng-model="user.password" required>

<span class="error" ng-show="registrationForm.password.$invalid &&

registrationForm.password.$touched">Password is required</span><br><br>

<button type="submit" ng-disabled="registrationForm.$invalid">Register</button>

</form>

<script>

var app = angular.module('registrationApp', []);

app.controller('RegistrationController', function ($scope) {

$scope.user = {};

$scope.submitForm = function () {

if ($scope.registrationForm.$valid) {

// Form is valid, submit the registration data

console.log('Registration data:', $scope.user);

}

};

</script>

</body>

</html>

---------------------------------------------------------------------

**Example-14 : Write an AngularJS code that display two options : read file**

**and write into file. When write file option is selected, it accept rollno, name**

**of student and age using form and store in csv file when submit button is**

**clicked. When read file option is selected, it read all records available in t he**

**csv file.**

Answer:

Step-1: Create following html file. This file

index.html.

--------------------------------------------------------------------------

<!DOCTYPE html>

<html ng-app="fileApp">

<head>

<title>File Operations</title>

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

</head>

<body>

<h2>File Operations</h2>

<a href="#/write">Write File</a> | <a href="#/read">Read File</a>

<div ng-view></div>

<script>

var app = angular.module('fileApp', ['ngRoute']);

app.config(function ($routeProvider) {

$routeProvider

.when('/write', {

templateUrl: 'write.html',

controller: 'WriteController'

})

.when('/read', {

templateUrl: 'read.html',

controller: 'ReadController'

})

.otherwise({ redirectTo: '/write' });

});

</script>

</body>

</html>

------------------------------------------------------------------

Step-2:Write File Form:

Create a form for writing data to the CSV file (write.html):

------------------------------------------------------------------

<h3>Write File</h3>

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

<label for="rollNumber">Roll Number:</label>

<input type="text" id="rollNumber" ng-model="student.rollNumber" required><br><br>

<label for="name">Name:</label>

<input type="text" id="name" ng-model="student.name" required><br><br>

<label for="age">Age:</label>

<input type="number" id="age" ng-model="student.age" required><br><br>

<button type="submit">Submit</button>

</form>

------------------------------------------------------------------

Step-3:Write Controller:

Create a controller to handle writing data to the CSV file:

------------------------------------------------------------------

app.controller('WriteController', function ($scope) {

$scope.student = {};

$scope.writeToFile = function () {

// Data validation and CSV writing logic should go here

// In a real application, this would involve server-side logic.

};

});

------------------------------------------------------------------

Step-4:Read File Page:

Create a view for reading data from the CSV file (read.html):

------------------------------------------------------------------

<h3>Read File</h3>

<!-- Display the CSV file data here -->

------------------------------------------------------------------

Step-5:Read Controller (Optional):

If you want to read data from the CSV file, create a controller and implement the logic to read

the file. In a real-world application, this might involve a server-side script.