**Week-9**

**Develop an angular JS form to apply CSS and Events**

**Aim:** The aim of this experiment is to develop a **form using AngularJS** that applies **CSS styles**

**dynamically** and handles **events** such as button clicks, input changes, and form

submission. This experiment helps in understanding how to bind data, manipulate CSS

classes, and handle user interactions using AngularJS.

**Recommended Hardware / Software Requirements:**

* Intel based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100 MB free disk space.
* Nodejs, VS Code, MongoDB

Source Code:

<!DOCTYPE html>

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

<head>

<meta charset="UTF-8">

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

<title>AngularJS Form with CSS and Events</title>

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

<style>

body { font-family: Arial, sans-serif; text-align: center; }

.container { width: 50%; margin: auto; padding: 20px; border: 1px solid #ddd; border-radius: 10px; box-shadow: 2px 2px 10px #aaa; }

.highlight { border: 2px solid green; background-color: #eaffea; }

.error { color: red; font-size: 12px; }

input, select { width: 100%; padding: 8px; margin: 5px 0; border: 1px solid #ddd; }

button { padding: 10px; background: green; color: white; border: none; cursor: pointer; }

.hover-effect { background-color: yellow; cursor: pointer; }

</style>

</head>

<body ng-controller="FormController">

<h2>AngularJS Form with CSS & Events</h2>

<div class="container">

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

<label>Full Name:</label>

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

ng-class="{'highlight': isFocused}"

ng-focus="isFocused=true" ng-blur="isFocused=false">

<div class="error" ng-show="userForm.fullName.$touched && userForm.fullName.$invalid">

Name is required.

</div>

<label>Email:</label>

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

<div class="error" ng-show="userForm.email.$touched && userForm.email.$invalid">

Enter a valid email.

</div>

<label>Favorite Color:</label>

<select ng-model="user.color" ng-change="colorChanged()" required>

<option value="">Select a color</option>

<option value="red">Red</option>

<option value="blue">Blue</option>

<option value="green">Green</option>

</select>

<p>Your selected color: <span ng-style="{'color': user.color}">{{ user.color }}</span></p>

<br>

<button type="submit" ng-disabled="userForm.$invalid" ng-mouseover="hover=true" ng-mouseleave="hover=false"

ng-class="{'hover-effect': hover}">

Submit

</button>

</form>

<h3 style="color: green;" ng-show="submitted">Form Submitted Successfully!</h3>

</div>

<script>

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

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

$scope.user = {};

$scope.isFocused = false;

$scope.hover = false;

$scope.submitted = false;

$scope.colorChanged = function() {

console.log("Color changed to:", $scope.user.color);

};

$scope.submitForm = function() {

if ($scope.userForm.$valid) {

$scope.submitted = true;

console.log("Form Data:", $scope.user);

}

};

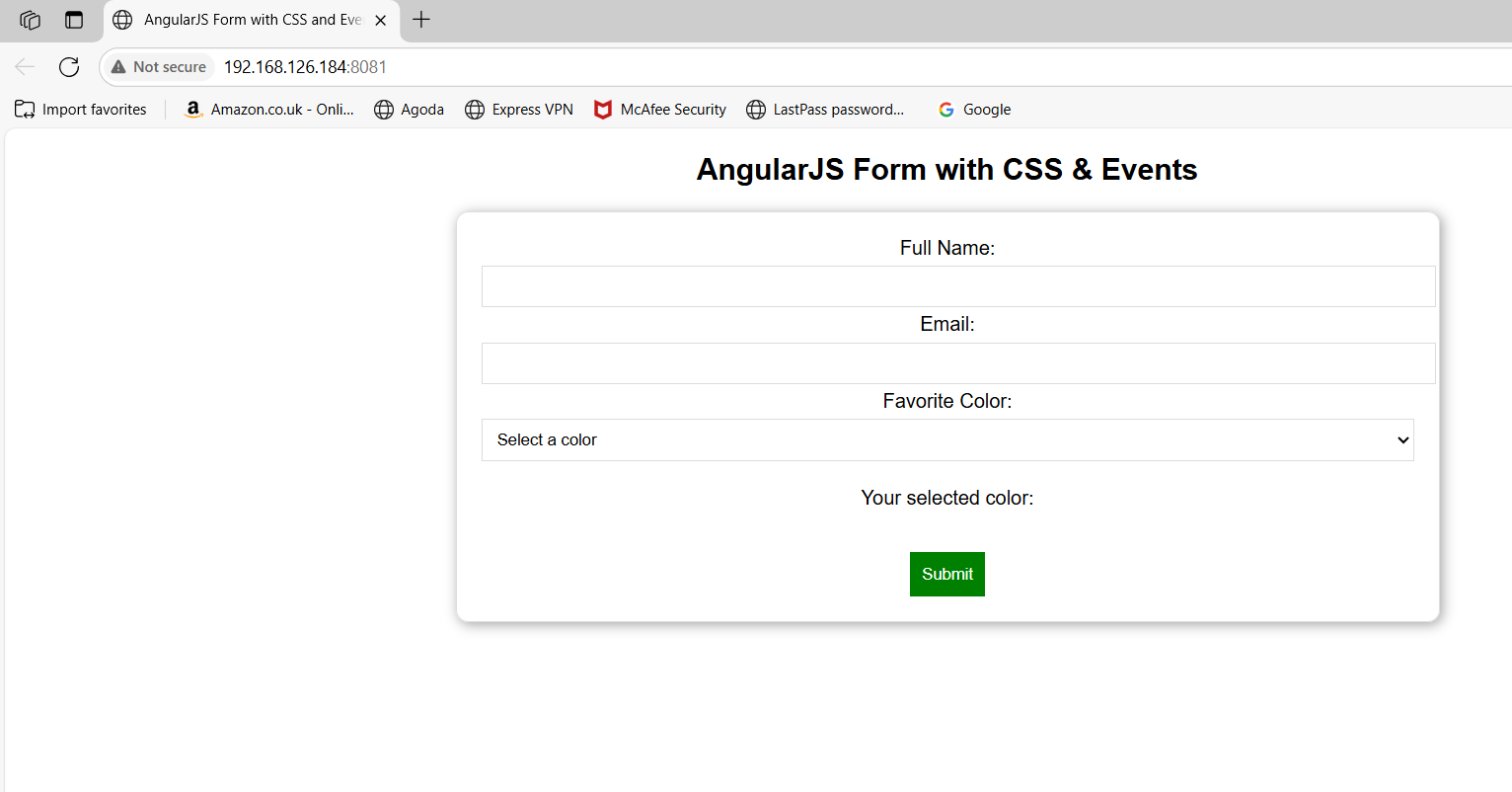
});

</script>

</body>

</html>

**Output:**

****

**Viva Questions:**

1.What is AngularJS?

**AngularJS** is an open-source JavaScript framework developed by **Google** for building **dynamic web applications**. It allows developers to use HTML as the template language and extend HTML's syntax to express application components clearly and succinctly.

2.What are the advantages of using AngularJS?

**Two-Way Data Binding**

* Automatically synchronizes data between the **model** and the **view**.
* Reduces the need to write a lot of boilerplate code to keep the UI in sync with the data.

**MVC Architecture**

* Helps in separating concerns:
  + **Model**: Handles data.
  + **View**: Displays data.
  + **Controller**: Connects the model and view.
* Makes applications easier to manage, scale, and debug.

**Dependency Injection (DI)**

* Manages components and services automatically.
* Makes it easier to **test**, **reuse**, and **maintain** code.

3.What is the difference between AngularJS and Angular?

| **Feature/Aspect** | **AngularJS (1.x)** | **Angular (2+)** |
| --- | --- | --- |

|  |  |  |
| --- | --- | --- |
| **Architecture** | MVC (Model-View-Controller) | Component-based |

|  |  |  |
| --- | --- | --- |
| **Language** | JavaScript | TypeScript (superset of JavaScript) |

|  |  |  |
| --- | --- | --- |
| **Mobile Support** | Limited / Not optimized | Built with mobile-first approach |

|  |  |  |
| --- | --- | --- |
| **Performance** | Slower for large apps | Much faster due to better change detection |

|  |  |  |
| --- | --- | --- |
| **Data Binding** | Two-way binding | Two-way & one-way (more control) |

1. What is ng-model in AngularJS?

ng-model is a **directive** in AngularJS that binds the **value of HTML controls** (like input, textarea, select) to **application data (model)**.

1. What is $scope in AngularJS?

In AngularJS, **$scope** is an **object** that refers to the **application model**. It acts as a **bridge between the controller and the view (HTML)**.

**Week-10**

**Develop a Job Registration form and validate it using angular JS.**

**Aim**: The aim of this experiment is to develop a **Job Registration Form** using **AngularJS** and implement **form validation**. This experiment helps in understanding how to:

* Create a **dynamic form** using **AngularJS directives**.
* Implement **real-time validation** for required fields, email format, and password strength.
* Display error messages when validation fails.
* Enable or disable the submit button based on validation.

**Recommended Hardware / Software Requirements:**

* Intel based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100 MB free disk space.
* Nodejs, VS Code, MongoDB

**Source Code:**

<!DOCTYPE html>

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

<head>

<meta charset="UTF-8">

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

<title>Job Registration Form</title>

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

<style>

body { font-family: Arial, sans-serif; text-align: center; }

.container { width: 40%; margin: auto; }

.error { color: red; font-size: 12px; }

input, select { width: 100%; padding: 8px; margin: 5px 0; }

button { padding: 10px; background: green; color: white; border: none; cursor: pointer; }

</style>

</head>

<body ng-controller="JobController">

<h2>Job Registration Form</h2>

<div class="container">

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

<label>Full Name:</label>

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

<div class="error" ng-show="jobForm.fullName.$touched && jobForm.fullName.$invalid">

Name is required.

</div>

<label>Email:</label>

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

<div class="error" ng-show="jobForm.email.$touched && jobForm.email.$invalid">

Enter a valid email.

</div>

<label>Phone Number:</label>

<input type="text" name="phone" ng-model="user.phone" ng-pattern="/^\d{10}$/" required>

<div class="error" ng-show="jobForm.phone.$touched && jobForm.phone.$invalid">

Enter a valid 10-digit phone number.

</div>

<label>Password:</label>

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

<div class="error" ng-show="jobForm.password.$touched && jobForm.password.$invalid">

Password is required.

</div>

<label>Confirm Password:</label>

<input type="password" name="confirmPassword" ng-model="user.confirmPassword" required ng-change="checkPassword()">

<div class="error" ng-show="user.password !== user.confirmPassword">

Passwords do not match.

</div>

<label>Job Role:</label>

<select name="jobRole" ng-model="user.jobRole" required>

<option value="">Select a role</option>

<option value="Developer">Developer</option>

<option value="Designer">Designer</option>

<option value="Manager">Manager</option>

</select>

<div class="error" ng-show="jobForm.jobRole.$touched && jobForm.jobRole.$invalid">

Please select a job role.

</div>

<br>

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

</form>

<h3 style="color: green;" ng-show="submitted">Registration Successful!</h3>

</div>

<script>

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

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

$scope.user = {};

$scope.submitted = false;

$scope.checkPassword = function() {

if ($scope.user.password !== $scope.user.confirmPassword) {

$scope.jobForm.confirmPassword.$setValidity("passwordMatch", false);

} else {

$scope.jobForm.confirmPassword.$setValidity("passwordMatch", true);

}

};

$scope.submitForm = function() {

if ($scope.jobForm.$valid) {

$scope.submitted = true;

console.log("Form Submitted!", $scope.user);

}

};

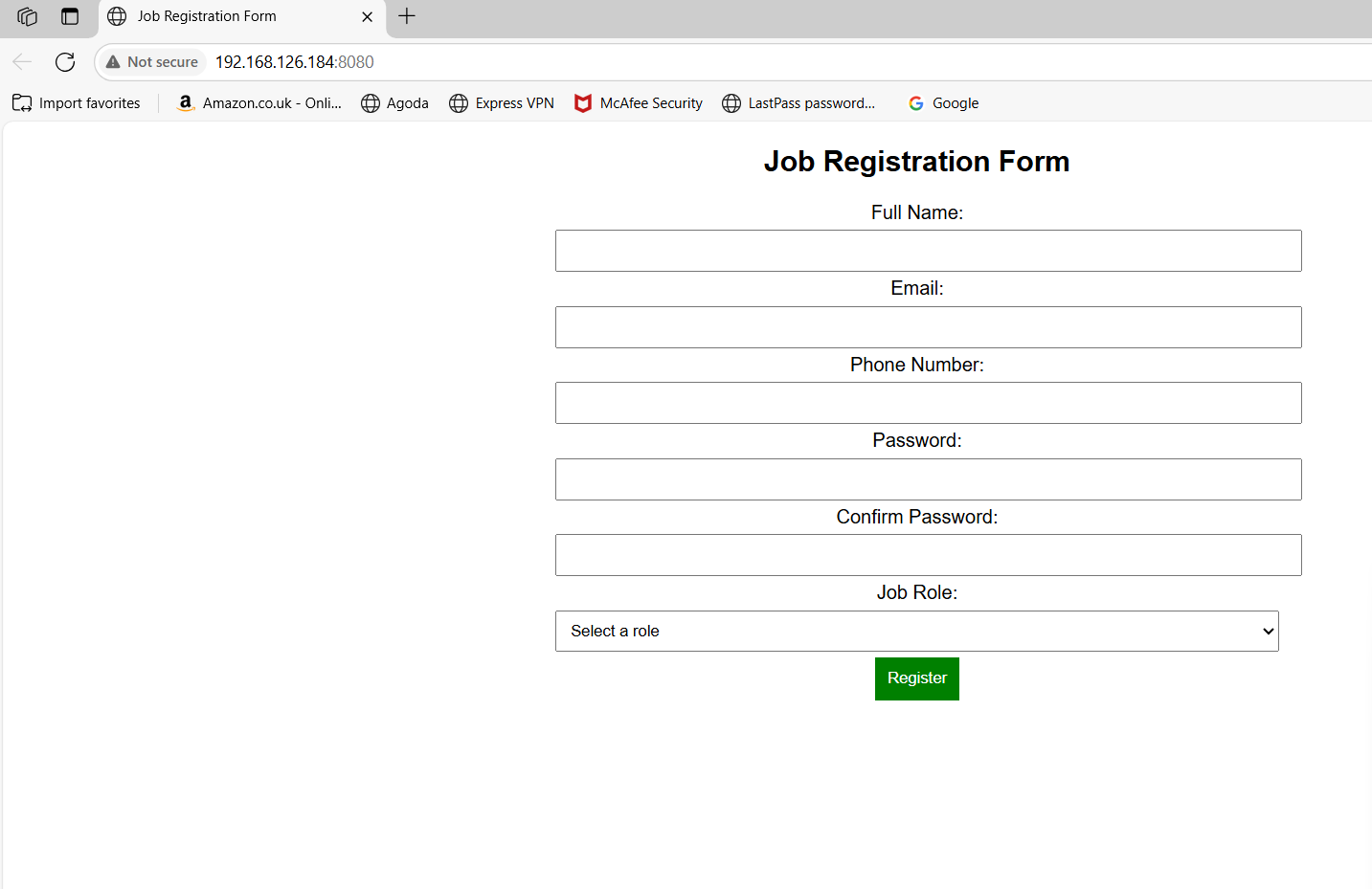
});

</script>

</body>

</html>

**Output:**



**Viva Questions:**

* 1. What are the key features of AngularJS

**Two-Way Data Binding**

Automatically syncs data between the **model (JavaScript objects)**

and the **view (HTML)**.

Any change in the UI updates the model and vice versa — in real time!

**2. MVC Architecture**

AngularJS follows the **Model-View-Controller** pattern:

* + **Model**: The data
  + **View**: The presentation (UI)
  + **Controller**: Business logic
  1. What is the difference between ng-show and ng-if?

| **Feature** | **ng-show** | **ng-if** |
| --- | --- | --- |

|  |  |  |
| --- | --- | --- |
| **Behavior** | Shows or hides the element (via CSS) | Adds or removes the element from the DOM |

|  |  |  |
| --- | --- | --- |
| **DOM Presence** | Element is **always** present in the DOM | Element is **created/destroyed** dynamically |

|  |  |  |
| --- | --- | --- |
| **Performance** | Better for **frequent toggling** | Better for **initial load or rarely toggled** |

|  |  |  |
| --- | --- | --- |
| **CSS Used** | Applies display: none when false | No CSS involved — removes element entirely |

3.What is novalidate in an HTML form, and why is it used?

* novalidate is a **Boolean attribute** that you can add to an HTML <form> element to **disable the browser’s built-in form validation**.

4.What is $dirty, $pristine, $valid, and $invalid in AngularJS forms?

| **Property** | **Description** |
| --- | --- |
| **$pristine** | True if the field has **not been touched/modified** yet. |
| **$dirty** | True if the field **has been modified** by the user. |
| **$valid** | True if the field passes **all validation rules** (like required, type, etc.). |
| **$invalid** | True if the field **fails** any validation rule. |