| | |
|---|---|
| **Name of Student** | **MANAV PUNJABI** |
| **Class Roll No** | **D15A_ 44** |
| **D.O.P.** | |
| **D.O.S.** | |
| **Sign and Grade** | |

## Experiment –8: AngularJS

1) **Aim:** To study AngularJS 2)

**Problem Statement:**

a) Demonstrate with an AngularJS code one way data binding and two way data binding in AngularJS

b) Implement a basic authentication system for a web application using AngularJS. Create a simple login page that takes a username and password, and upon submission, checks for a hardcoded set of credentials. If the credentials are valid, display a success message; otherwise, show an error message. Demonstrate AngularJS controller, module and form directives.

c) Users want to search for books by title, author, or genre. To accomplish this, develop an AngularJS custom filter named bookFilter and include it into the application.

d) Create a reusable and modular custom AngularJS service to handle user authentication. Include this service into an application.

1. What are directives?

**Directives** are special tokens in the HTML that tell AngularJS to do something (like attach behavior or manipulate DOM elements).

**Common directives:**

- ng-model
- ng-bind
- ng-repeat
- ng-if
- ng-show / ng-hide
- ng-click

---

2. What is data binding in AngularJS?

**Data binding** in AngularJS is the automatic synchronization of data between the **model (JavaScript objects)** and the **view (HTML UI)**. This means when the data in the model changes, the view reflects those changes instantly, and when the user updates data in the view (like input fields), the model gets updated as well. AngularJS mainly uses **two-way data binding**, making it easier to build dynamic and interactive web applications without writing a lot of manual DOM manipulation code.

---

3. How is form validation done in AngularJS?

AngularJS provides **built-in validation directives** like:

- required
- ng-minlength
- ng-maxlength

- ng-pattern

It also uses $valid, $dirty, $touched, and $error to manage form validation states.

---

4. What is the use of AngularJS Controllers in the application?

AngularJS **Controllers** are used to define the application's **business logic** and to manage the **data** that is used in the view. They act as a **bridge between the view (HTML)** and the **model (JavaScript data)**. Inside a controller, we can define functions and variables using $scope, which can then be accessed and used in the HTML. Controllers help keep the code organized and make it easier to manage and update the application logic.

---

5. What is the use of AngularJS Filters in the application?

Filters are used to **format and transform data** in the view (e.g., filtering arrays, formatting dates, changing case).

**Common filters:**

- uppercase
- lowercase
- currency
- date
- filter
- orderBy

Code:

**Auth-app.html:**

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AngularJS Auth Service</title>
                                                                                <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script> </head>

<body ng-app="authApp" ng-controller="AuthController">

    <h2>Login Page</h2>
    <form ng-submit="login()">
        <label>Username:</label>
        <input type="text" ng-model="user.username" required><br><br>

        <label>Password:</label>
        <input type="password" ng-model="user.password" required><br><br>

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

    <p style="color: green;" ng-if="message">{{ message }}</p>
```

```
    <p style="color: red;" ng-if="error">{{ error }}</p>

    <script>         var app =
angular.module("authApp",
[]);

        // Authentication Service         app.service("AuthService", function
() {             var validCredentials = { username: "admin", password:
"1234" };

            this.authenticate = function (user) {                 return
user.username === validCredentials.username &&
user.password === validCredentials.password;
            };
        });

        // Controller         app.controller("AuthController", function
($scope, AuthService) {
            $scope.user = {};

            $scope.login = function () {                 if
(AuthService.authenticate($scope.user)) {
                    $scope.message = "Login successful!";
                    $scope.error = "";
```

```
            } else {
                $scope.error = "Invalid credentials!";
                $scope.message = "";
            }
        };
    });
  </script>

</body>

</html>
```

**Book-app.html:**

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AngularJS Book Filter</title>
                                                                    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script> </head>

<body ng-app="bookApp" ng-controller="BookController">
```

```
    <h2>Search for Books</h2>
    <input type="text" ng-model="searchText" placeholder="Search books...">

    <ul>
        <li ng-repeat="book in books | bookFilter:searchText">
            <strong>{{ book.title }}</strong> by {{ book.author }} ({{ book.genre }})
        </li>
    </ul>

    <script>         var app =
angular.module("bookApp", []);

        // Custom filter         app.filter("bookFilter", function () {
return function (books, searchText) {                 if (!searchText)
return books;                 searchText = searchText.toLowerCase();
return books.filter(function (book) {                     return
book.title.toLowerCase().includes(searchText) ||
book.author.toLowerCase().includes(searchText) ||
book.genre.toLowerCase().includes(searchText);

                });
            };
        });
```

```
        // Controller         app.controller("BookController",
function ($scope) {
            $scope.books = [
                { title: "Harry Potter", author: "J.K. Rowling", genre: "Fantasy" },
                { title: "The Hobbit", author: "J.R.R. Tolkien", genre: "Fantasy" },
                { title: "1984", author: "George Orwell", genre: "Dystopian" }
            ];
        });
    </script>

</body>

</html>
```

**Index.html:**

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>AngularJS Data Binding</title>
                                                                    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script> </head>
```

```
<body ng-app="myApp" ng-controller="myController">
    <h2>One-Way Data Binding</h2>
    <p>{{ message }}</p> <!-- One-way binding (only displays the value) -->

    <h2>Two-Way Data Binding</h2>
    <input type="text" ng-model="message"> <!-- Two-way binding -->
    <p>Updated Message: {{ message }}</p>

    <script>         var app = angular.module("myApp",
[]);         app.controller("myController", function
($scope) {
            $scope.message = "Hello, AngularJS!";
        });
    </script>
</body>

</html>
```

3) **Output:**

# One-Way Data Binding

Hello, AngularJS!

# Two-Way Data Binding

Hello, AngularJS!

Updated Message: Hello, AngularJS!