Name : Prajjwal Pandey

Class : D15A

Roll no. : 32

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.

3) **Theory:**
    1. What are directives? Name some of the most commonly used directives in AngularJS application.

        Directives are special tokens in AngularJS (starting with `ng-`) that tell the HTML how to behave. They extend HTML with custom behavior.

Common directives:

- `ng-model` – binds input fields to model data
- `ng-bind` – binds data to HTML
- `ng-controller` – attaches a controller to the view
- `ng-repeat` – repeats elements based on a collection
- `ng-if` / `ng-show` / `ng-hide` – conditionally display elements

2. What is data binding in AngularJS?

   Data binding is the automatic synchronization of data between the model (JavaScript) and the view (HTML).

- **One-way binding**: Model → View
- **Two-way binding**: Model ⇄ View (via `ng-model`)

3. How is form validation done in angularJS

   AngularJS uses built-in validation directives like `required`, `ng-minlength`, `ng-pattern`, etc., and provides `$valid`, `$invalid`, `$touched`, and `$dirty` states to show validation feedback.

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

   Controllers in AngularJS define the application logic and data. They act as a bridge between the view and the model, using `$scope` to pass data and functions.

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

Filters are used to format data before displaying it in the view. They can be used to sort, filter, or transform data in expressions (e.g., `{{ name | uppercase }}`).

4) **Output:**

CODE:

Index.html

```
<!DOCTYPE html>

<html ng-app="myApp">

<head>

    <meta charset="utf-8">

    <title>AngularJS Demo</title>

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

    <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; margin: 20px; }

        input { margin: 5px; padding: 5px; }

        .success { color: green; }

        .error { color: red; }

    </style>

</head>

<body ng-controller="MainController">



    <h2>🔁 One-Way Data Binding</h2>
```

```
    <p>{{ oneWay }}</p>

    <h2>🔁 Two-Way Data Binding</h2>
    <input type="text" ng-model="twoWay">
    <p>You typed: {{ twoWay }}</p>

    <hr>

    <h2>🔐 Login Form</h2>
    <form ng-submit="login()" name="loginForm" novalidate>
        <input type="text" ng-model="loginData.username" placeholder="Username" required />
        <input type="password" ng-model="loginData.password" placeholder="Password" required />
        <button type="submit">Login</button>
    </form>
    <p class="success" ng-if="successMsg">{{ successMsg }}</p>
    <p class="error" ng-if="errorMsg">{{ errorMsg }}</p>

    <hr>

    <h2>📚 Book Search</h2>
    <input type="text" ng-model="searchQuery" placeholder="Search by title, author, or genre">
```

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


</body>
</html>
```

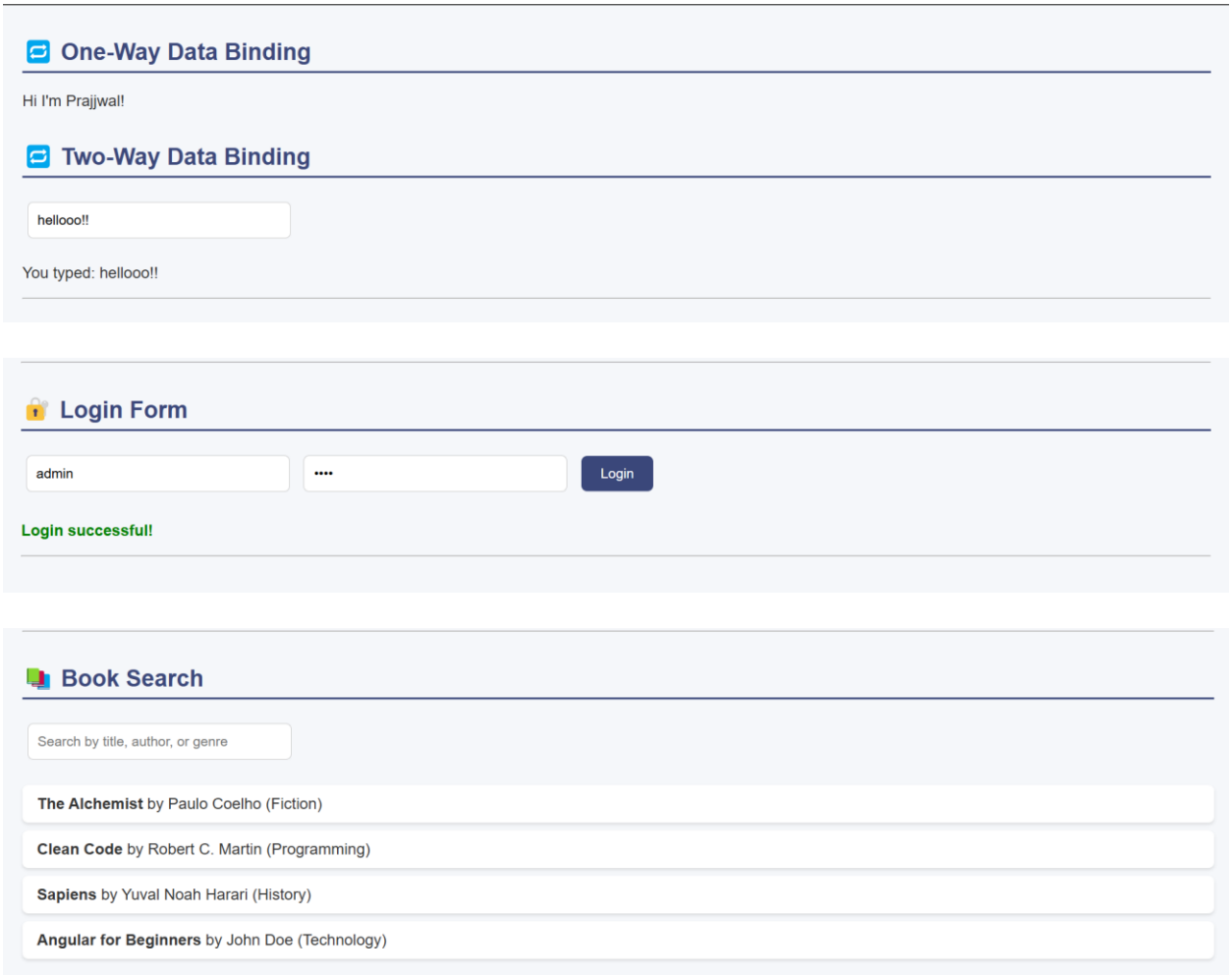
app.js

```
// Module definition
var app = angular.module('myApp', []);


// Controller
app.controller('MainController', function($scope, AuthService) {
    // One-way binding example
    $scope.oneWay = "Hi I'm Prajjwal!";


    // Two-way binding example
    $scope.twoWay = "";


    // Login data
    $scope.loginData = {};
```

```
    // Login handler
    $scope.login = function() {
        if (AuthService.authenticate($scope.loginData.username, $scope.loginData.password)) {
            $scope.successMsg = "Login successful!";
            $scope.errorMsg = "";
        } else {
            $scope.errorMsg = "Invalid username or password.";
            $scope.successMsg = "";
        }
    };

    // Book list
    $scope.books = [
        { title: 'The Alchemist', author: 'Paulo Coelho', genre: 'Fiction' },
        { title: 'Clean Code', author: 'Robert C. Martin', genre: 'Programming' },
        { title: 'Sapiens', author: 'Yuval Noah Harari', genre: 'History' },
        { title: 'Angular for Beginners', author: 'John Doe', genre: 'Technology' }
    ];
});
```

```
// Custom filter for book search
app.filter('bookFilter', function() {
    return function(books, query) {
        if (!query) return books;
        query = query.toLowerCase();
        return books.filter(function(book) {
            return book.title.toLowerCase().includes(query) ||
                   book.author.toLowerCase().includes(query) ||
                   book.genre.toLowerCase().includes(query);
        });
    };
});

// Custom AuthService
app.service('AuthService', function() {
    var validUsername = "admin";
    var validPassword = "1234";

    this.authenticate = function(username, password) {
        return username === validUsername && password === validPassword;
    };
});
```

OUTPUT:


One-Way Data Binding
Hi I'm Prajjwal!
Two-Way Data Binding
hellooo!!
You typed: hellooo!!
Login Form
admin
Login
Login successful!
Book Search
Search by title, author, or genre
The Alchemist by Paulo Coelho (Fiction)
Clean Code by Robert C. Martin (Programming)
Sapiens by Yuval Noah Harari (History)
Angular for Beginners by John Doe (Technology)