

Experiment –8: AngularJS

| | |
|-----------------|-------------------|
| Name of Student | <u>PRANAV POL</u> |
| Class Roll No | <u>D15A 41</u> |
| D.O.P. | |
| D.O.S. | |
| Sign and Grade | |

Problem Statement

1. Demonstrate One-Way and Two-Way Data Binding in AngularJS
2. Implement a Basic Authentication System
 - o Create a simple login page that takes a username and password.
 - o Validate credentials against a hardcoded set.
 - o Display success or error message based on validation.
 - o Demonstrate AngularJS controller, module, and form directives.
3. Implement a Book Search Feature
 - o Develop a custom AngularJS filter named `bookFilter` for searching books by title, author, or genre.
4. Create a Modular Custom Authentication Service
 - o Develop a reusable authentication service and integrate it into an application.

Theory

What are Directives in AngularJS?

Directives in AngularJS are special attributes or elements that extend HTML functionality by manipulating the DOM, handling events, and defining reusable components.

Commonly Used Directives in AngularJS

| Directive | Description |
|--------------------------------|---|
| <code>ng-app</code> | Defines the root element of an AngularJS application. |
| <code>ng-controller</code> | Attaches a controller to a specific section. |
| <code>ng-model</code> | Binds form inputs to the scope. |
| <code>ng-bind</code> | Dynamically updates text content. |
| <code>ng-show / ng-hide</code> | Shows or hides elements conditionally. |
| <code>ng-repeat</code> | Iterates over collections. |
| <code>ng-click</code> | Handles button clicks. |
| <code>ng-if</code> | Removes or adds elements based on conditions. |
| <code>ng-class</code> | Applies CSS classes dynamically. |

What is Data Binding in AngularJS?

Data Binding synchronizes data between the model (JavaScript variables) and the view (HTML template).

Types of Data Binding

| Type | Description |
|----------------------|---|
| One-Way Data Binding | UI updates when the model changes, but not vice versa. Example: <code>{} expression {}</code> |
| Two-Way Data Binding | Model and view are always in sync. Example: <code>ng-model="data"</code> |

Form Validation in AngularJS

AngularJS provides built-in validation using directives like `required`, `minlength`, `maxlength`, and `pattern`.

Example of Form Validation:

```
<form name="myForm">
  <input type="text" name="username" ng-model="user.name" required minlength="3">
    <span ng-show="myForm.username.$error.required">This field is required</span>
    <span ng-show="myForm.username.$error.minlength">Minimum 3 characters
required</span>
</form>
```

 If the user enters less than 3 characters, an error message is displayed.

Use of AngularJS Controllers

AngularJS Controllers manage data and business logic using `$scope`.

Example of an AngularJS Controller:

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

app.controller('MainController', function($scope) {
  $scope.greeting = "Hello, AngularJS!";
  $scope.updateGreeting = function() {
    $scope.greeting = "Welcome to AngularJS!";
  };
});
```

Using the Controller in HTML:

```
<div ng-controller="MainController">
  <p>{{ greeting }}</p>
  <button ng-click="updateGreeting()">Change Greeting</button>
</div>
```

 Clicking the button updates the greeting value.

Use of AngularJS Filters

Filters format and transform data in views/templates.

Commonly Used Filters

| Filter | Description |
|--------|-------------|
|--------|-------------|

| | |
|------------------------|--|
| <code>uppercase</code> | Converts text to uppercase. |
| <code>lowercase</code> | Converts text to lowercase. |
| <code>currency</code> | Formats a number as currency. |
| <code>date</code> | Formats a date. |
| <code>filter</code> | Filters an array based on a condition. |
| <code>orderBy</code> | Sorts an array by a specific property. |

Example of Using Filters in AngularJS

```
<p>{{ "hello world" | uppercase }}</p> <!-- Output: HELLO WORLD -->
<p>{{ 1000 | currency }}</p> <!-- Output: ₹1,000.00 (or $1,000.00) -->
<p>{{ myDate | date:'fullDate' }}</p> <!-- Output: Wednesday, March 27, 2025 -->
```

Implementation

1. One-Way and Two-Way Data Binding

```
<div ng-app="myApp" ng-controller="DataBindingController">
  <h3>One-Way Binding:</h3>
  <p>{{ message }}</p>

  <h3>Two-Way Binding:</h3>
  <input type="text" ng-model="message">
</div>

var app = angular.module('myApp', []);
app.controller('DataBindingController', function($scope) {
  $scope.message = "Hello, AngularJS!";
});
```

2. Basic Authentication System

```
<div ng-app="authApp" ng-controller="AuthController">
  <form ng-submit="login()">
    <input type="text" ng-model="username" placeholder="Username" required>
```

```

<input type="password" ng-model="password" placeholder="Password" required>
<button type="submit">Login</button>
</form>
<p>{{ message }}</p>
</div>
var authApp = angular.module('authApp', []);
authApp.controller('AuthController', function($scope) {
  var validUser = { username: "admin", password: "password123" };

  $scope.login = function() {
    if ($scope.username === validUser.username && $scope.password ===
validUser.password) {
      $scope.message = "Login successful!";
    } else {
      $scope.message = "Invalid credentials!";
    }
  };
});

```

3. Custom Book Search Filter

```

app.filter('bookFilter', function() {
  return function(books, searchText) {
    if (!searchText) return books;
    return books.filter(book =>
      book.title.toLowerCase().includes(searchText.toLowerCase()) ||
      book.author.toLowerCase().includes(searchText.toLowerCase()) ||
      book.genre.toLowerCase().includes(searchText.toLowerCase())
    );
  };
});

```

4. Custom Authentication Service

```

app.service('AuthService', function() {
  this.authenticate = function(username, password) {
    return username === "admin" && password === "password123";
  };
});

```

Output

Binding
<!DOCTYPE html>

```
<html lang="en" ng-app="bindingApp">

<head>

    <meta charset="UTF-8">

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

    <title>AngularJS Data Binding Demo</title>

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

    <script src="app.js"></script> <!-- External AngularJS Script -->

    <style>

        body { font-family: Arial, sans-serif; text-align: center; margin: 50px; }

        input { padding: 8px; margin: 10px; }

        .container { max-width: 500px; margin: auto; border: 1px solid #ccc; padding: 20px; border-radius: 10px; }

    </style>

</head>

<body ng-controller="BindingController">

    <div class="container">

        <h2>One-Way Data Binding (Updated from UI)</h2>

        <p>Enter your name: <input type="text" ng-model="username"></p>

        <p>Interpolation: {{ username }}</p>

        <p>Using ng-bind: <span ng-bind="username"></span></p>

        <h2>Two-Way Data Binding (Model & UI Sync)</h2>

        <p>Type something: <input type="text" ng-model="message"></p>

        <p>You typed: {{ message }}</p>

    </div>

```

```
<h2>Modify Model Programmatically</h2>

<button ng-click="changeValues()">Change Values</button>

</div>

</body>

</html>
```

App.js

```
// Define AngularJS module

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

// Define controller

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

    $scope.username = "AngularJS User"; // Initial value for One-Way Binding

    $scope.message = "pranav"; // Two-Way Binding variable

    // Function to modify values dynamically

    $scope.changeValues = function() {

        $scope.username = "Updated User";

        $scope.message = "Updated Message!";

    };

});
```

One-Way Data Binding (Updated from UI)

Enter your name:

Interpolation: AngularJS User

Using ng-bind: AngularJS User

Two-Way Data Binding (Model & UI Sync)

Type something:

You typed: pranav

Modify Model Programmatically

One-Way Data Binding (Updated from UI)

Enter your name:

Interpolation: roll 41

Using ng-bind: roll 41

Two-Way Data Binding (Model & UI Sync)

Type something:

You typed: pranav

Modify Model Programmatically

Change Values

Book search

```
<!DOCTYPE html>
<html lang="en" ng-app="bookApp">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AngularJS Book Search</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
  <script src="app.js"></script> <!-- External AngularJS Script -->
<style>
  body { font-family: Arial, sans-serif; text-align: center; margin: 50px; }
  .container { max-width: 600px; margin: auto; padding: 20px; border: 1px solid #ccc; border-radius: 10px; }
  input { width: 100%; padding: 10px; margin: 10px 0; }
  table { width: 100%; border-collapse: collapse; margin-top: 20px; }
  th, td { padding: 10px; border: 1px solid #ddd; }
  th { background-color: #f4f4f4; }
```

```

</style>
</head>
<body ng-controller="BookController">

<div class="container">
  <h2>Book Search by pranav pol</h2>

  <!-- Search input -->
  <input type="text" ng-model="searchText" placeholder="Search by title, author, or genre">

  <!-- Display book list -->
  <table>
    <thead>
      <tr>
        <th>Title</th>
        <th>Author</th>
        <th>Genre</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="book in books | bookFilter:searchText">
        <td>{{ book.title }}</td>
        <td>{{ book.author }}</td>
        <td>{{ book.genre }}</td>
      </tr>
    </tbody>
  </table>

</div>

</body>
</html>

```

```

App.js
// Define AngularJS module
var app = angular.module('bookApp', []);

// Define controller
app.controller('BookController', function($scope) {
  $scope.searchText = ""; // Stores the search input

  // Sample book data
  $scope.books = [
    { title: "The Great Gatsby", author: "F. Scott Fitzgerald", genre: "Classic" },

```

```

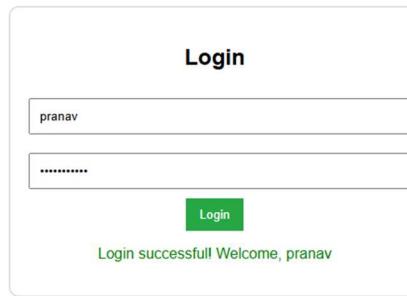
        { title: "To Kill a Mockingbird", author: "Harper Lee", genre: "Fiction" },
        { title: "1984", author: "George Orwell", genre: "Dystopian" },
        { title: "Pride and Prejudice", author: "Jane Austen", genre: "Romance" },
        { title: "Moby-Dick", author: "Herman Melville", genre: "Adventure" }
    ];
});

// Custom filter for searching books
app.filter('bookFilter', function() {
    return function(books, searchText) {
        if (!searchText) return books; // Return all books if no search input

        searchText = searchText.toLowerCase();

        return books.filter(function(book) {
            return book.title.toLowerCase().includes(searchText) ||
                book.author.toLowerCase().includes(searchText) ||
                book.genre.toLowerCase().includes(searchText);
        });
    };
});

```



Book Search by pranav pol

Search by title, author, or genre

| Title | Author | Genre |
|-----------------------|---------------------|-----------|
| The Great Gatsby | F. Scott Fitzgerald | Classic |
| To Kill a Mockingbird | Harper Lee | Fiction |
| 1984 | George Orwell | Dystopian |
| Pride and Prejudice | Jane Austen | Romance |
| Moby-Dick | Herman Melville | Adventure |

Book Search by pranav pol

the

| Title | Author | Genre |
|------------------|---------------------|---------|
| The Great Gatsby | F. Scott Fitzgerald | Classic |

Login

```
<!DOCTYPE html>
<html lang="en" ng-app="authApp">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<title>AngularJS Authentication</title>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
<script src="authService.js"></script> <!-- Authentication Service -->
<script src="app.js"></script> <!-- Main AngularJS App -->
<style>
  body { font-family: Arial, sans-serif; text-align: center; margin: 50px; }
  .container { max-width: 400px; margin: auto; padding: 20px; border: 1px solid #ccc; border-radius: 10px; }
  input { width: 100%; padding: 10px; margin: 10px 0; }
  button { width: 100%; padding: 10px; margin: 10px 0; }
  .error { color: red; }
</style>
</head>
<body ng-controller="AuthController">

  <div class="container" ng-show="!isAuthenticated">
    <h2>Login</h2>
    <input type="text" ng-model="user.username" placeholder="Username">
    <input type="password" ng-model="user.password" placeholder="Password">
    <button ng-click="login()">Login</button>
    <p class="error" ng-show="errorMessage">{{ errorMessage }}</p>
  </div>

  <div class="container" ng-show="isAuthenticated">
    <h2>Welcome, {{ currentUser.username }}!</h2>
    <p>You are now logged in.</p>
    <button ng-click="logout()">Logout</button>
  </div>

</body>
</html>

```

```

Authservice.js
// Define AngularJS module
var app = angular.module('authApp', []);

// Authentication Service
app.factory('AuthService', function() {
  var currentUser = null; // Stores authenticated user

  var users = [
    { username: "pranav", password: "12345" },
    { username: "user", password: "123456" }
  ];

```

```

return {
  login: function(username, password) {
    var user = users.find(u => u.username === username && u.password === password);
    if (user) {
      currentUser = user;
      return { success: true, user: user };
    } else {
      return { success: false, message: "Invalid credentials" };
    }
  },
  logout: function() {
    currentUser = null;
  },
  getUser: function() {
    return currentUser;
  },
  isAuthenticated: function() {
    return currentUser !== null;
  }
};
});

```

App.js

```

// Define Controller
app.controller('AuthController', function($scope, AuthService) {
  $scope.user = {} // Stores login input
  $scope.isAuthenticated = AuthService.isAuthenticated();
  $scope.currentUser = AuthService.getUser();
  $scope.errorMessage = "";

  // Login Function
  $scope.login = function() {
    var result = AuthService.login($scope.user.username, $scope.user.password);
    if (result.success) {
      $scope.isAuthenticated = true;
      $scope.currentUser = result.user;
      $scope.errorMessage = "";
    } else {
      $scope.errorMessage = result.message;
    }
  };
});

```

```
// Logout Function  
$scope.logout = function() {  
    AuthService.logout();  
    $scope.isAuthenticated = false;  
    $scope.currentUser = null;  
};  
});
```

Login

Welcome, pranav!

You are now logged in.

Conclusion: -

This practical explores key AngularJS concepts, including data binding, authentication, custom filters, and services. It demonstrates one-way and two-way data binding, showcasing how data flows between the model and view. A basic authentication system is implemented using AngularJS controllers, modules, and form directives. A custom filter (bookFilter) is created for searching books by title, author, or genre, while a modular authentication service ensures reusability and maintainability. These implementations highlight AngularJS's capabilities in building dynamic and interactive web applications.