Experiment – 8: Angular JS

Name of Student	Shravani Anil Patil
Class Roll No	D15A_37
D.O.P.	25.03.25
D.O.S.	
Sign and Grade	

Aim: To study AngularJS

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.

Theory:

1. What are directives? Name some of the most commonly used directives in AngularJS applications.

Directives in AngularJS are special attributes that extend the functionality of HTML by adding behavior to elements. They help in creating reusable components and dynamic web applications.

Commonly used directives in AngularJS:

- **ng-app**: Defines the root element of an AngularJS application.

- **ng-model**: Binds an input field to a variable in the scope.

- **ng-bind**: Displays the value of an expression in the HTML.

- **ng-repeat**: Iterates over a collection and generates HTML elements dynamically.

- **ng-if**: Conditionally includes or removes elements from the DOM.

- **ng-show/ng-hide**: Shows or hides elements based on a condition.

2. What is data binding in AngularJS?

Data binding in AngularJS is the synchronization of data between the model (JavaScript variables) and the view (HTML). It eliminates the need for manual DOM manipulation.

Types of data binding in AngularJS:

One-way data binding: The view reflects changes in the model, but changes in the view do not update the model.
Two-way data binding: Any change in the model updates the view and vice versa. This is achieved using the ng-model directive.

3. How is form validation done in AngularJS?

Form validation in AngularJS is handled using built-in validation directives and properties of the ng-model object.

Steps for form validation:
1. Use AngularJS validation directives like:

   ○   required (ensures a field is not empty)

   ○   ng-minlength and ng-maxlength (restrict character length)

   ○   ng-pattern (enforces a specific format using regex)

   ○   type="email" (validates email format)

2. Use the $valid, $invalid, $touched, and $dirty properties of ng-model to check form status.

3. Show error messages dynamically using ng-show or ng-if.

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

AngularJS Controllers manage the application's data and logic. They act as a bridge between the view and the model.

Functions of Controllers:

Define properties and methods for the scope ($scope).
Control the application's business logic.
Interact with services like $http for data fetching.
Handle user interactions.

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

Filters in AngularJS are used to format and transform data displayed in the view without modifying the original data.

Common AngularJS Filters:

uppercase and lowercase: Convert text case.
currency: Formats numbers as currency.
date: Formats dates.
filter: Filters an array based on a condition.
orderBy: Sorts an array.

Code:

index.html

```html
<!DOCTYPE html>
<html ng-app="myApp">
<head>
  <title>AngularJS Authentication Demo</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
                                                  <link           rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f7fa;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }
    .container {
      background-color: white;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
      max-width: 600px;
      width: 100%;
      text-align: center;
    }
    h1 {
      color: #333;
      margin-bottom: 20px;
```

```css
}

form {
    display: flex;
    flex-direction: column;
}

label {
    margin-bottom: 5px;
    text-align: left;
    font-size: 14px;
    color: #333;
}

input[type="text"], input[type="password"] {
    padding: 10px;
    margin: 10px 0 20px;
    border: 1px solid #ccc;
    border-radius: 4px;
    font-size: 14px;
    width: 100%;
}

button {
    background-color: #eec5a9;
    color: white;
    padding: 12px;
    border: none;
    border-radius: 4px;
    font-size: 16px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

button:hover {
    background-color: #f1d2b5;
}
.success {
    color: #e5c7eb;
    font-size: 16px;
    margin-top: 10px;
}

.error {
```

```css
        color: red;
        font-size: 16px;
        margin-top: 10px;
}
.book-search {
        margin-top: 20px;
}

input[type="text"] {
        width: calc(100% - 20px);
        padding: 10px;
        border-radius: 4px;
        border: 1px solid #ccc;
        margin-bottom: 15px;
        font-size: 14px;
}

ul {
        list-style: none;
        padding: 0;
}

li {
        padding: 10px;
        margin: 5px 0;
        border: 1px solid #ddd;
        border-radius: 4px;
        background-color: #f9f9f9;
        font-size: 14px;
        display: flex;
        justify-content: space-between;
        align-items: center;
}

li:hover {
        background-color: #f1f1f1;
}

.book-title {
        font-weight: bold;
}

.book-icon {
        margin-right: 10px;
```

```html
          color: #a1e4e9;
      }

    .no-results {
        font-style: italic;
        color: #888;
    }
    .login-icon {
        font-size: 30px;
        color: #cfad85;
    }
  </style>
</head>
<body ng-controller="MainController">
  <div class="container">
    <h1>Login Page</h1>
    <form ng-submit="login()">
      <div>
        <i class="fas fa-user login-icon"></i>
      </div>
      <label for="username">Username:</label>
      <input type="text" ng-model="user.username" placeholder="Enter Username" required />

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

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

    <p ng-if="message" ng-class="{'success': isSuccess, 'error': !isSuccess}">{{ message }}</p>

    <div ng-if="isLoggedIn" class="book-search">
      <h2>Book Search</h2>
       <input type="text" ng-model="searchQuery" placeholder="Search for books by title, author,
or genre" />

      <ul>
        <li ng-repeat="book in books | bookFilter:searchQuery">
          <i class="fas fa-book book-icon"></i>
          <span class="book-title">{{ book.title }}</span> by {{ book.author }}
          <span class="book-genre">({{ book.genre }})</span>
        </li>
      </ul>
```

```html
        <p ng-if="books | bookFilter:searchQuery | length == 0" class="no-results">
            No books found.
        </p>
    </div>
  </div>
  <script src="app.js"></script>
</body>
</html>
```

App.js

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

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

    $scope.books = [
        { title: 'The Hobbit', author: 'J.R.R. Tolkien', genre: 'Fantasy' },
        { title: '1984', author: 'George Orwell', genre: 'Dystopian' },
        { title: 'Pride and Prejudice', author: 'Jane Austen', genre: 'Romance' },
        { title: 'The Great Gatsby', author: 'F. Scott Fitzgerald', genre: 'Classic' },
        {title: 'Wings of Fire', author: 'A.P.J Abdul Kalam', genre: 'Biography'}
    ];

    $scope.user = { username: '', password: '' };

    $scope.message = '';
    $scope.isSuccess = false;
    $scope.isLoggedIn = false;

    $scope.login = function() {
        if (AuthService.login($scope.user.username, $scope.user.password)) {
            $scope.isSuccess = true;
            $scope.message = 'Login successful!';
            $scope.isLoggedIn = true;
        } else {
            $scope.isSuccess = false;
            $scope.message = 'Invalid username or password.';
            $scope.isLoggedIn = false;
        }
    };
});

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

```
        });
    };
});

app.service('AuthService', function() {
    var hardcodedCredentials = [
        { username: 'Madhura', password: 'madhura' },
        { username: 'Rujuta', password: 'rujuta' },
        { username: 'Shravani', password: 'shravani' }
    ];

    this.login = function(username, password) {
        return hardcodedCredentials.some(function(user) {
            return user.username === username && user.password === password;
        });
    };
});
```
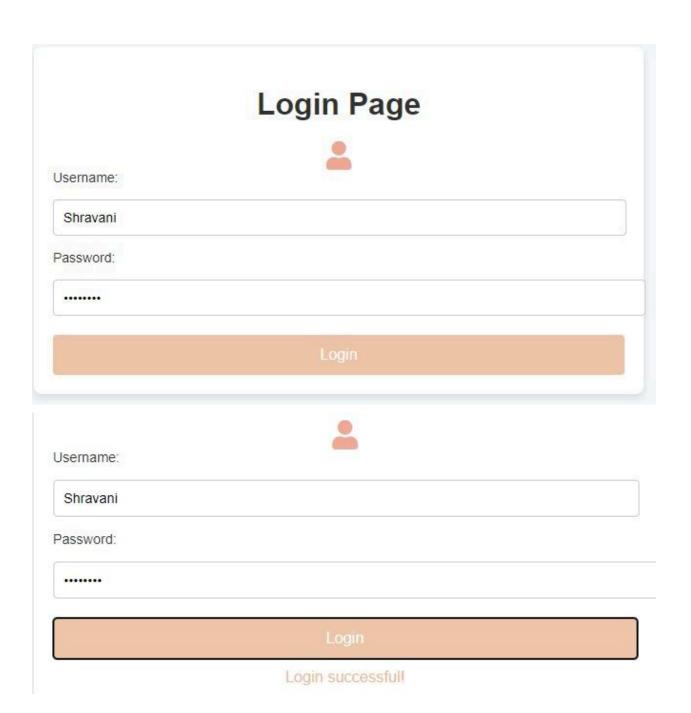
Output:

Login Page

Username:

Shravani

Password:

••••••••

Login

Username:

Shravani

Password:

••••••••

Login

Login successful!

Book Search

Search for books by title, author, or genre

📙	**The Hobbit**	by J.R.R. Tolkien	(Fantasy)
📙	**1984**	by George Orwell	(Dystopian)
📙	**Pride and Prejudice**	by Jane Austen	(Romance)
📙	**The Great Gatsby**	by F. Scott Fitzgerald	(Classic)
📙	**Wings of Fire**	by A.P.J Abdul Kalam	(Biography)

Book Search

win

📙	**Wings of Fire**	by A.P.J Abdul Kalam	(Biography)