

Experiment 8 : To study Angular JS

| | |
|------------------------|-----------------------|
| Name of Student | Ronak Katariya |
| Class Roll No | D15A / 23 |
| D.O.P. | |
| 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 application

Directives are one of the core features of AngularJS that allow developers to extend HTML functionality. They are special markers on DOM elements (such as attributes, elements, or CSS classes) that tell AngularJS to attach specific behaviors to those elements or transform them.

Commonly Used Directives in AngularJS:

1. ng-app: Defines the root element of an AngularJS application.
2. ng-model: Binds the value of an input, select, or textarea to a variable.
3. ng-bind: Replaces the content of an HTML element with the value of an expression.
4. ng-repeat: Iterates over an array or collection to generate repeated elements.
5. ng-if: Conditionally includes or removes elements from the DOM.
6. ng-show / ng-hide: Shows or hides an element based on a Boolean expression.
7. ng-click: Binds a click event to a function in the controller.

2. What is data binding in AngularJS?

Data binding is the process of synchronizing data between the model and the view. AngularJS supports two types of data binding:

One-way Data Binding: The model updates the view, but changes in the view do not affect the model. Example:

```
<span ng-bind="message"></span>
```

Two-way Data Binding: The model and view are linked such that changes in one reflect in the other. Example:

```
<input type="text" ng-model="username">  
<p>Hello, {{username}}!</p>
```

Two-way data binding is one of AngularJS's most powerful features, reducing the need for manual DOM manipulation.³

3. How is form validation done in AngularJS?

Form validation in AngularJS ensures that user input is correct before submission. AngularJS provides built-in directives for form validation:

1. ng-required: Ensures that an input field is mandatory.
2. ng-minlength / ng-maxlength: Sets minimum and maximum character limits for input fields.

3. ng-pattern: Validates input based on a regular expression pattern.
4. ng-disabled: Disables a form element based on an expression.

Example:

```
<form name="userForm">
  <input type="email" name="email" ng-model="userEmail" ng-
required="true">
  <span ng-show="userForm.email.$error.required">Email is required.</span>
</form>
```

AngularJS tracks form states such as \$pristine, \$dirty, \$valid, and \$invalid to provide real-time validation feedback.

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

AngularJS controllers are JavaScript functions used to define application logic and manage the flow of data between the view and the model. Controllers are attached to the DOM using the ng-controller directive.

Key Functions of Controllers:

1. Define scope variables: Controllers bind data to the view using the \$scope object.
2. Handle business logic: Controllers process user input and manipulate the model accordingly.
3. Communicate with services: They fetch data from APIs or services.

Example:

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

In the view:

```
<div ng-controller="MainController">
  <p>{{ message }}</p>
</div>
```

Controllers improve the maintainability of AngularJS applications by separating concerns.

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

Filters in AngularJS modify data before displaying it in the view. They can be used within expressions and directives like `ng-repeat`.

Commonly Used Filters:

1. uppercase / lowercase: Converts text to upper or lower case.
2. currency: Formats numbers as currency.
3. date: Formats date values.
4. filter: Filters an array based on a specified condition.
5. orderBy: Sorts an array by a specified property.

Example:

```
<p>{{ "hello world" | uppercase }}</p>
```

```
<p>{{ 1000 | currency }}</p>
```

```
<p>{{ myDate | date:'short' }}</p>
```

Filters enhance the readability of data and improve user experience.

Output:

Index.html

```
// AngularJS Examples for Data Binding, Authentication, Filtering, and Service
```

```
// a) One-way and Two-way Data Binding
```

```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <title>Data Binding</title>
```

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

```
</head>
```

```
<body ng-controller="BindingController">
```

```
  <h2>One-Way Binding:</h2>
```

```
  <p>{{ staticText }}</p>
```

```
  <h2>Two-Way Binding:</h2>
```

```
  <input type="text" ng-model="dynamicText">
```

```
  <p>You typed: {{ dynamicText }}</p>
```

```
<script>
```

```
  angular.module('bindingApp', []).controller('BindingController', function($scope) {  
    $scope.staticText = "This is one-way binding";
```

```
$scope.dynamicText = "Editable text";
});
</script>
</body>
</html>
```

// b) Basic Authentication Example

```
<!DOCTYPE html>
<html ng-app="authApp">
<head>
  <title>Login</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="LoginController">
  <form name="loginForm" ng-submit="login()">
    <label>Username:</label>
    <input type="text" ng-model="user.username" required><br>
    <label>Password:</label>
    <input type="password" ng-model="user.password" required><br>
    <input type="submit" value="Login">
  </form>
  <p>{{ message }}</p>

  <script>
    angular.module('authApp', []).controller('LoginController', function($scope) {
      const validUser = { username: 'admin', password: '1234' };
      $scope.user = {};
      $scope.message = "";

      $scope.login = function() {
        if ($scope.user.username === validUser.username && $scope.user.password ===
validUser.password) {
          $scope.message = 'Login Successful!';
        } else {
          $scope.message = 'Invalid Credentials!';
        }
      };
    });
  </script>
</body>
</html>
```

// c) Custom Filter for Book Search

```
<!DOCTYPE html>
<html ng-app="bookApp">
<head>
  <title>Book Filter</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
```

```

<body ng-controller="BookController">
  <input type="text" ng-model="searchText" placeholder="Search books...">
  <ul>
    <li ng-repeat="book in books | bookFilter:searchText">
      {{ book.title }} by {{ book.author }} [{{ book.genre }}]
    </li>
  </ul>

  <script>
    angular.module('bookApp', []).controller('BookController', function($scope) {
      $scope.books = [
        { title: '1984', author: 'George Orwell', genre: 'Dystopian' },
        { title: 'Moby Dick', author: 'Herman Melville', genre: 'Adventure' },
        { title: 'The Great Gatsby', author: 'F. Scott Fitzgerald', genre: 'Classic' }
      ];
    }).filter('bookFilter', function() {
      return function(books, searchText) {
        if (!searchText) return books;
        searchText = searchText.toLowerCase();
        return books.filter(book =>
          book.title.toLowerCase().includes(searchText) ||
          book.author.toLowerCase().includes(searchText) ||
          book.genre.toLowerCase().includes(searchText)
        );
      };
    });
  </script>
</body>
</html>

```

// d) Reusable Authentication Service

```

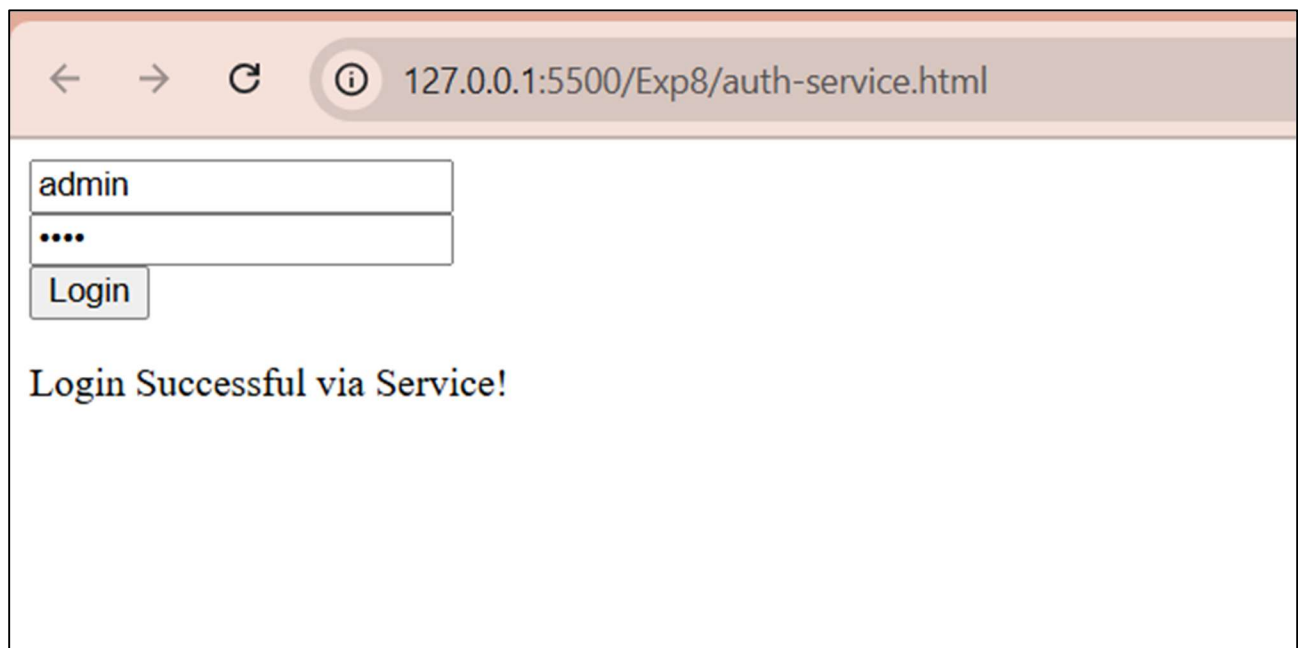
<!DOCTYPE html>
<html ng-app="authServiceApp">
<head>
  <title>Service Auth</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="ServiceLoginController">
  <form ng-submit="login()">
    <input type="text" ng-model="user.username" placeholder="Username" required><br>
    <input type="password" ng-model="user.password" placeholder="Password" required><br>
    <input type="submit" value="Login">
  </form>
  <p>{{ message }}</p>

  <script>
    angular.module('authServiceApp', [])
      .service('AuthService', function() {
        const validUser = { username: 'admin', password: '1234' };

```





```
this.authenticate = function(username, password) {  
    return username === validUser.username && password === validUser.password;  
};  
})  
.controller('ServiceLoginController', function($scope, AuthService) {  
    $scope.user = {};  
    $scope.message = "";  
  
    $scope.login = function() {  
        if (AuthService.authenticate($scope.user.username, $scope.user.password)) {  
            $scope.message = 'Login Successful via Service!';  
        } else {  
            $scope.message = 'Invalid Login via Service!';  
        }  
    };  
});  
</script>  
</body>  
</html>
```

Authentication-Service:



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5500/Exp8/auth-service.html". The page contains a login form with two input fields: the first contains the text "admin" and the second contains four dots "....". Below the input fields is a button labeled "Login". Below the button, the message "Login Successful via Service!" is displayed in a bold, black, serif font.

One-Way & Two-Way Data Binding Demo

    127.0.0.1:5500/Exp8/data-binding.html





One-Way Binding:

This is one-way binding

Two-Way Binding:

You typed: Ronak Katariya

Login Page





    127.0.0.1:5500/Exp8/login.html

Username:





Password:

Login Successful!

Book Finder Page

    127.0.0.1:5500/Exp8/book-filter.html

- 1984 by George Orwell [Dystopian]
- Moby Dick by Herman Melville [Adventure]
- The Great Gatsby by F. Scott Fitzgerald [Classic]

    127.0.0.1:5500/Exp8/book-filter.html

- 1984 by George Orwell [Dystopian]