Experiment 8 : To study Angular JS

Name of Student	**Prathamesh Palve**
Class Roll No	**D15A_31**
D.O.P.	**20/03/2025**
D.O.S.	**27/03/2025**
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:-

Directives in AngularJS

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.

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:

html
CopyEdit

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

1.

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

html
CopyEdit

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

2.

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

Form Validation 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:

html
CopyEdit

```
<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.

Use of AngularJS Controllers in Applications

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:

javascript
CopyEdit

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

In the view:

html
CopyEdit

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

Controllers improve the maintainability of AngularJS applications by separating concerns.

Use of AngularJS Filters in Applications

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:

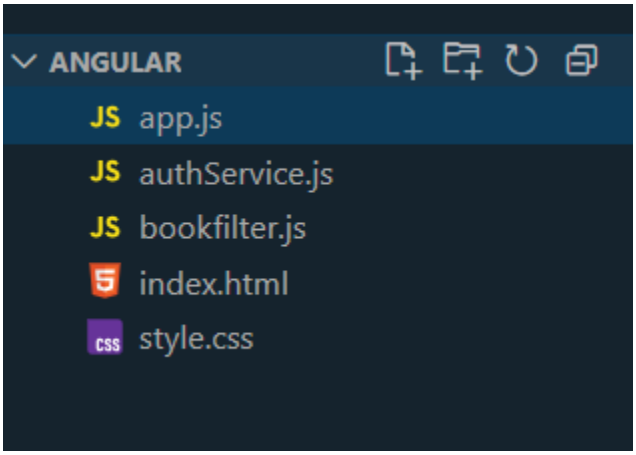
1. **uppercase / lowercase**: Converts text to upper or lower case.
2. **currency**: Formats numbers as currency.
3. **date**: Formats date values.
4. **filter**: Filters an array based on a specified condition.
5. **orderBy**: Sorts an array by a specified property.

Example:

html
CopyEdit

```
<p>{{ "hello world" | uppercase }}</p>
<p>{{ 1000 | currency }}</p>
<p>{{ myDate | date:'short' }}</p>
```

Filters enhance the readability of data and improve user experience. CODE:-



Index.html

```
<!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 Demo</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"
></script>
    <script src="app.js"></script>
    <script src="authService.js"></script>
    <script src="bookfilter.js"></script>
    <link rel="stylesheet" href="style.css">
</head>
<body ng-controller="MainController">

    <h2>🔗 One-Way and Two-Way Data Binding</h2>
    <p>One-way binding: <span>{{ message }}</span></p>
    <p>Two-way binding: <input type="text" ng-model="message"></p>

    <hr>

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

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

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

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

    <hr>

    <h2>📚 Book Search</h2>
    <input type="text" ng-model="searchText" placeholder="Search
books...">

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

</body>
</html>
```

app.js

```
var app = angular.module("BookApp", []);

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

    // One-way and Two-way Binding Example
    $scope.message = "Hello, AngularJS!";

    // Authentication System
    $scope.user = {};
    $scope.authMessage = "";

    $scope.login = function () {
```

```
        if (AuthService.authenticate($scope.user.username,
$scope.user.password)) {
            $scope.authMessage = "✅ Login successful!";
        } else {
            $scope.authMessage = "❌ Invalid credentials. Try again.";
        }
    };

    // Book Search Feature
    $scope.books = [
        { title: "The Alchemist", author: "Paulo Coelho", genre: "Fiction"
},
        { title: "Sapiens", author: "Yuval Noah Harari", genre: "History"
},
        { title: "The Pragmatic Programmer", author: "Andrew Hunt", genre:
"Technology" },
        { title: "1984", author: "George Orwell", genre: "Dystopian" }
    ];
});
```

authService.js

```
app.service("AuthService", function () {
    var validUser = {
        username: "sanket",
        password: "more"
    };

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

bookfilter.js

```
app.filter("bookFilter", function () {
    return function (books, searchText) {
        if (!searchText) return books;

        searchText = searchText.toLowerCase();
        return books.filter(function (book) {
```

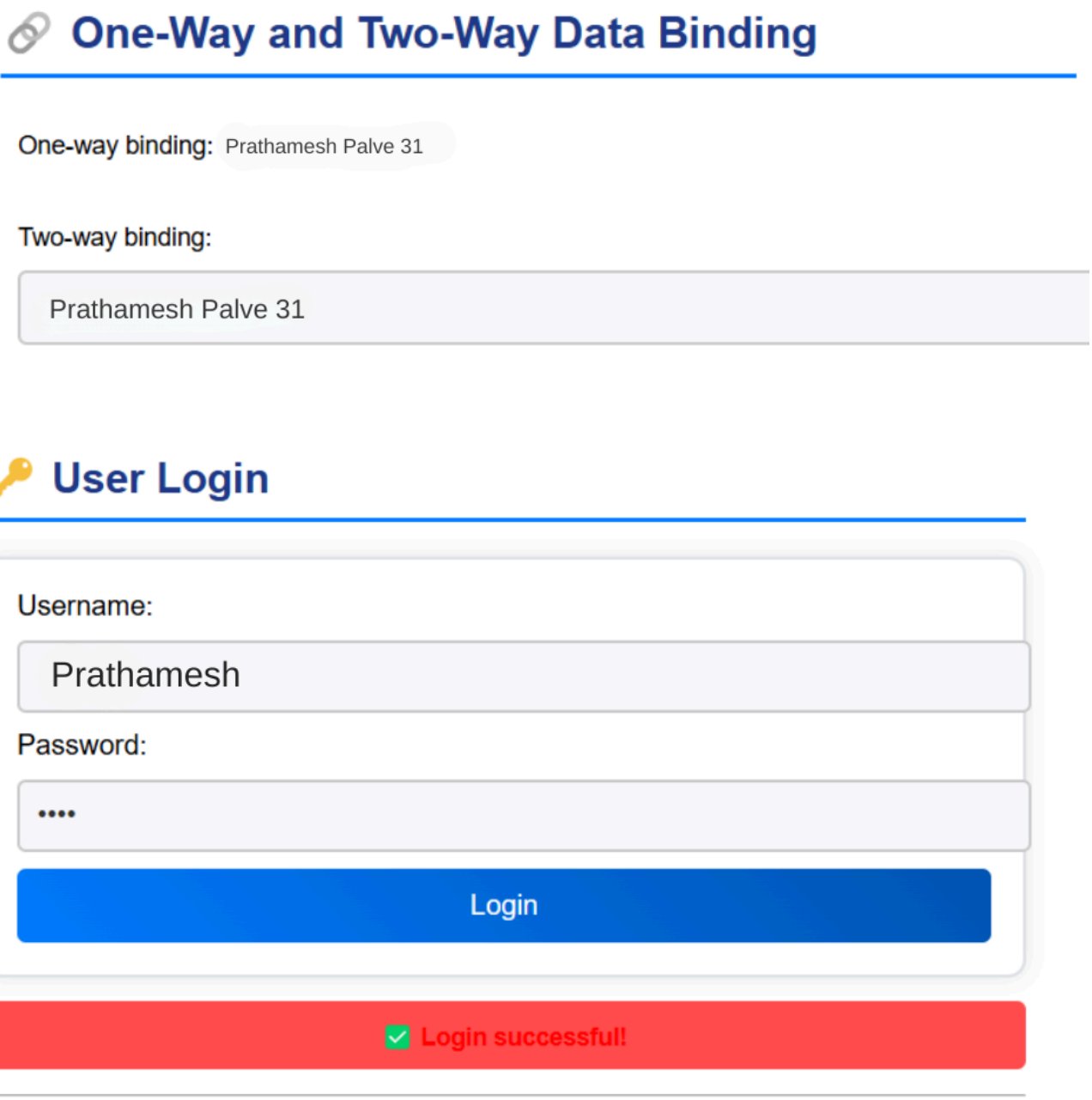
```
            return book.title.toLowerCase().includes(searchText) ||
                   book.author.toLowerCase().includes(searchText) ||
                   book.genre.toLowerCase().includes(searchText);
        });
    };
});
```

Style.css

```
/* Global Styles */
body {
    font-family: 'Poppins', sans-serif;
    background-color: #f0f5ff;  /* Light Blue Background */
    padding: 20px;
    max-width: 600px;
    margin: auto;
    box-shadow: 0px 0px 15px rgba(0, 0, 0, 0.1);
    border-radius: 10px;
    background: white;
}

/* Heading Styles */
h2 {
    color: #1e3a8a; /* Deep Blue */
    border-bottom: 3px solid #007BFF;
    padding-bottom: 8px;
}
```

OUTPUT:-


One-Way and Two-Way Data Binding
One-way binding: Prathamesh Palve 31
Two-way binding:
Prathamesh Palve 31
User Login
Username:
Prathamesh
Password:
••••
Login
Login successful!

Book Search

Search books...

The Alchemist by Paulo Coelho (Fiction)

Sapiens by Yuval Noah Harari (History)

The Pragmatic Programmer by Andrew Hunt (Technology)

1984 by George Orwell (Dystopian)

Book Search

Sapiens

Sapiens by Yuval Noah Harari (History)