Experiment –8: AngularJS

Name of Student	**Snehal Anilkumar Patil**
Class Roll No	**D15A / 38**
D.O.P.	
D.O.S.	
Sign and Grade	

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 commonly used directives in AngularJS applications.

What are Directives:

Directives in AngularJS are special attributes or elements that help extend HTML functionality. They tell AngularJS how to modify or control the DOM (Document Object Model).

Commonly Used Directives in AngularJS:

1. ng-model – Binds form elements (input, textarea) to scope variables.
2. ng-bind – Displays data from the scope in HTML elements.
3. ng-repeat – Loops through an array and generates repeated elements dynamically.
4. ng-if – Conditionally includes or removes an element based on a condition.
5. ng-show / ng-hide – Shows or hides an element based on a condition.
6. ng-click – Binds a click event to a function in the controller.
7. ng-init – Initializes variables in the scope.
8. ng-controller – Attaches a controller to a section of the page.

2. What is data binding in AngularJS?

Explanation:

Data binding in AngularJS is a powerful feature that allows automatic synchronization of data between the view (HTML) and the model (JavaScript controller).

Types of Data Binding in AngularJS:

1. One-Way Data Binding ({{ variable }})
   - The data flows only from the model to the view.
   - Changes in the model update the view, but the view cannot modify the model.
2. Two-Way Data Binding (ng-model)
   - The data flows both ways—from the model to the view and from the view back to the model.
   - Example:

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

   - When the user types in the input, name updates in the controller dynamically.

3. How is form validation done in AngularJS?

Explanation:

AngularJS provides built-in form validation to check user input and display error messages dynamically.

Steps for Form Validation in AngularJS:

1. Use ng-model to bind input fields to scope variables.
2. Use AngularJS built-in validation directives:
   - required → Field cannot be empty.
   - ng-minlength / ng-maxlength → Minimum & maximum character limits.
   - ng-pattern → Regular expressions for validation (e.g., emails, numbers).

Example:

```
<form name="myForm">
  <input type="email" name="userEmail" ng-model="email" required>
```

```
   <span ng-show="myForm.userEmail.$error.required">Email is required!</span>
   <span ng-show="myForm.userEmail.$error.email">Invalid email!</span>
</form>
```

- If the user enters an **invalid email**, an error message appears dynamically.

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

Explanation:

AngularJS controllers are used to control the data and behavior of the application.

Purpose of Controllers:

1. Manage application logic – Controllers contain the JavaScript functions to handle user interactions.
2. Handle data binding – Controllers store data in $scope, which AngularJS binds to the view.
3. Interact with services – Controllers fetch data from APIs using services (e.g., $http).

Example of a Controller:

```
var app = angular.module("myApp", []);
app.controller("myController", function ($scope) {
   $scope.message = "Hello, AngularJS!";
});
<div ng-controller="myController">
   <p>{{ message }}</p>
</div>
```

- The controller manages the data (message), and AngularJS updates the UI dynamically.

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

Explanation:

Filters in AngularJS are used to format and transform data before displaying it in the view.

Common Filters in AngularJS:

1. uppercase / lowercase → Converts text to upper/lower case.
2. currency → Formats numbers as currency.
3. date → Formats dates in different styles.
4. limitTo → Limits the number of displayed items in an array.
5. filter → Filters an array based on user input (e.g., searching in a list).

Example:

```
<p>{{ "hello world" | uppercase }}</p>  <!-- Output: HELLO WORLD -->

<p>{{ price | currency }}</p>  <!-- If price = 1000, Output: $1,000.00 -->

<input type="text" ng-model="searchText">
<ul>
   <li ng-repeat="book in books | filter:searchText">{{ book.title }}</li>
</ul>
```

- **Search Functionality:** filter:searchText dynamically updates the list based on user input.

4)Code and Screenshot of Output:

a) code one way data binding and two way data binding in AngularJS:

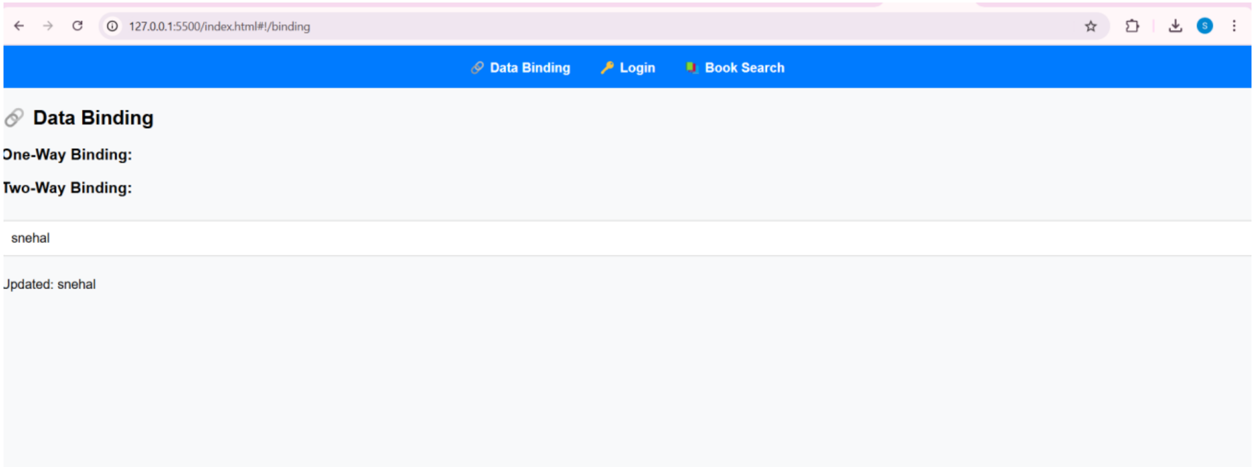
Code:

```
<h2>🔗 Data Binding</h2>

<h3>One-Way Binding:</h3>
<p>{{ oneWayText }}</p>

<h3>Two-Way Binding:</h3>
<input type="text" ng-model="twoWayText">
<p>Updated: {{ twoWayText }}</p>

app.controller("BindingController", function ($scope) {
    $scope.oneWayText = "This is one-way binding!";
    $scope.twoWayText = "Edit me!";
});
```



🔗 **Data Binding**

One-Way Binding:

Two-Way Binding:

snehal

Updated: snehal

LOGIN:

Code:

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

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

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

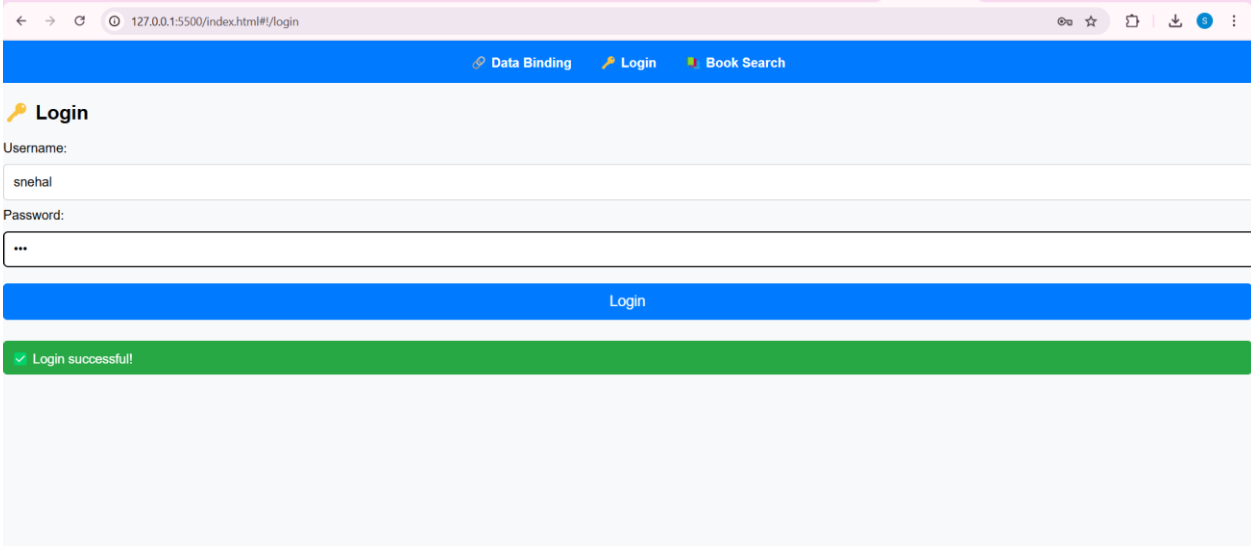
```
<p ng-class="messageClass">{{ message }}</p>
```

```
app.controller("AuthController", function ($scope, AuthService) {
    $scope.user = { username: "", password: "" };
    $scope.message = "";
    $scope.messageClass = "";

    $scope.login = function () {
        if (AuthService.authenticate($scope.user.username, $scope.user.password)) {
            $scope.message = "✅ Login successful!";
            $scope.messageClass = "success";
        } else {
            $scope.message = "❌ Invalid username or password!";
            $scope.messageClass = "error";
        }
    };
});

app.service("AuthService", function () {
    var validUser = { username: "snehal", password: "123" };

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



🔑 Login

Username:

snehal

Password:

•••

✓ Login successful!

BOOK SEARCH:

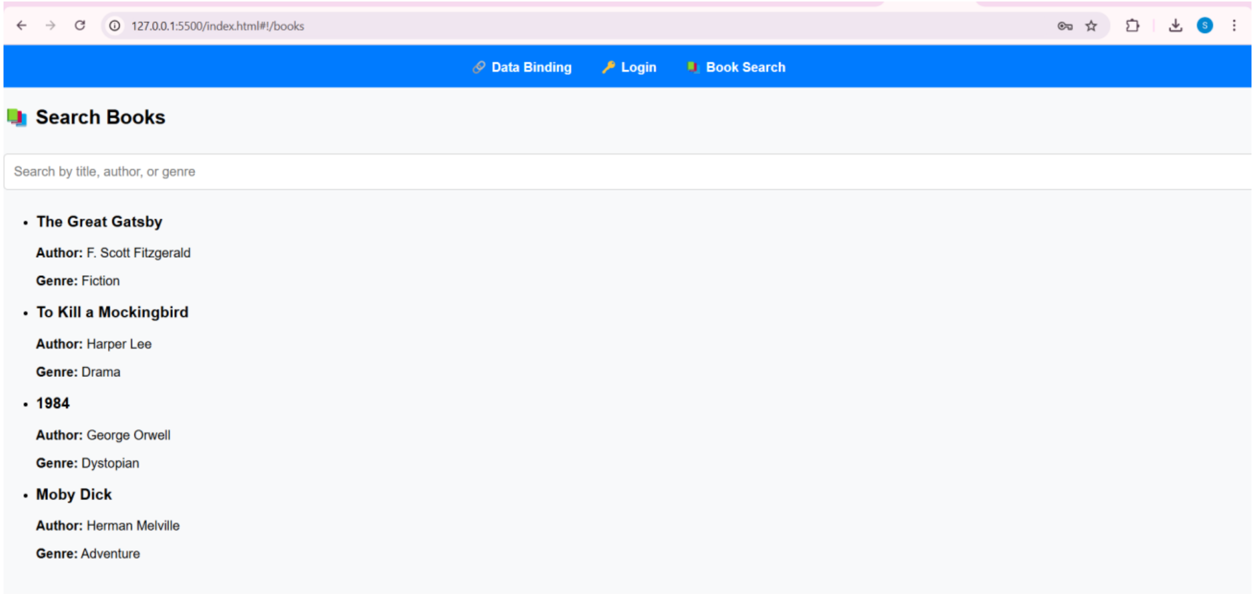
Code:

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

<ul>
    <li ng-repeat="book in books | bookFilter:searchText">
        <h3>{{ book.title }}</h3>
        <p><strong>Author:</strong> {{ book.author }}</p>
        <p><strong>Genre:</strong> {{ book.genre }}</p>
    </li>
</ul>

app.controller("BookController", function ($scope) {
    $scope.books = [
        { title: "The Great Gatsby", author: "F. Scott Fitzgerald", genre: "Fiction" },
        { title: "To Kill a Mockingbird", author: "Harper Lee", genre: "Drama" },
        { title: "1984", author: "George Orwell", genre: "Dystopian" },
        { title: "Moby Dick", author: "Herman Melville", genre: "Adventure" }
    ];
});
```

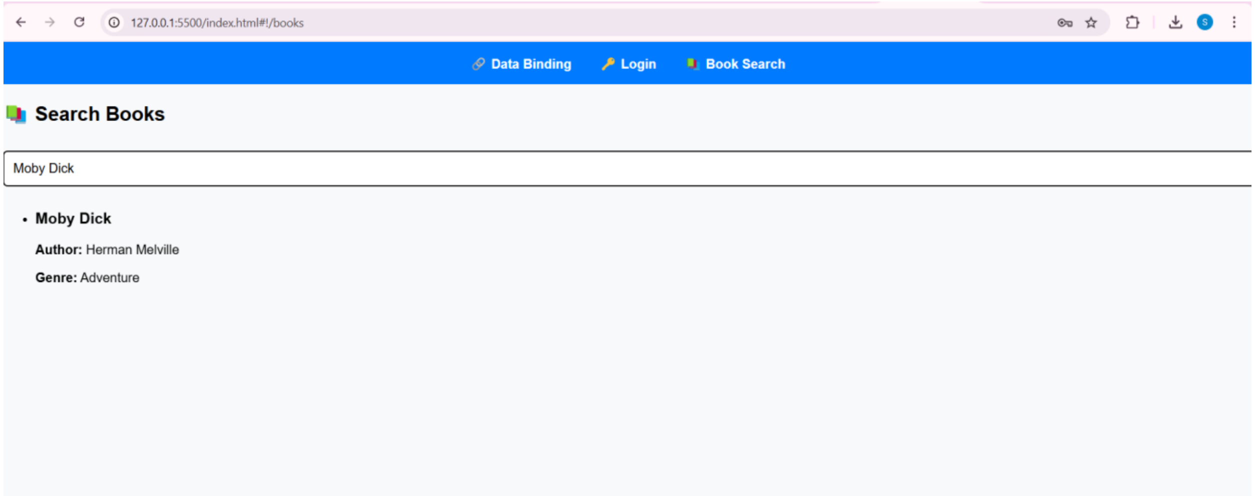
```
app.filter("bookFilter", function () {
    return function (books, searchText) {
        if (!searchText) return books;
        searchText = searchText.toLowerCase();
        return books.filter(function (book) {
            return (
                book.title.toLowerCase().includes(searchText) ||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



Search Books

Search by title, author, or genre

- **The Great Gatsby**
  Author: F. Scott Fitzgerald
  Genre: Fiction
- **To Kill a Mockingbird**
  Author: Harper Lee
  Genre: Drama
- **1984**
  Author: George Orwell
  Genre: Dystopian
- **Moby Dick**
  Author: Herman Melville
  Genre: Adventure



Search Books

Moby Dick

- **Moby Dick**
  Author: Herman Melville
  Genre: Adventure