1

Manav Jawrani 19.

DATE:

Exp 3: Angular JS.

Aim — To study Angular JS

Theory —

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

→. In angular JS, directives are a way to create ~~ua~~ reusable components that can be added to HTML. They allow you to extend HTML with new attributes and elements, which can be used to add functionality or change the behavior of existing HTML elements.
Some most commonly used directives are—

1. ng-model— It is used for two-way data binding between a model and a form input element.
2. ng-repeat — This directive is used to repeat a set of HTML elements based on an array or object.
3. ng-show/ng-hide — These directives are used to conditionally show or hide HTML elements.
4. ng-click — This directive is used to add click event handlers to HTML elements, allowing to trigger actions.

2. What is data binding in Angular JS?

→ 1. Data binding is a feature in Angular JS that lets you sync data between your application model and user interface.

FOR EDUCATIONAL USE

2. It allows you to bind the content of HTML elements to properties or expressions in your Javascript code.

3. There are different types of data binding, such as one-way, two-way and event binding.

4. Data binding simplifies the process of creating dynamic and interactive web applications in Angular JS.

3. How form validation is done in Angular JS?

→ Form validation in Angular JS is a built-feature that allows you to validate user input in forms using a declarative syntax. Here's how it works:

1. Add the 'ng-app' directive to the root element of your application to initialize Angular JS.
2. Add the 'ng-form' directive to the form element you want to validate and give it a name.
3. Add the 'ng-model' directive to each input field in your form to bind it to avariable in your AngularJS controller.
4. Add the 'required' attribute to each input field you want to make required.
5. Add the 'ng-submit" directive to the form element and specify a function to execute when the form is submitted.
6. In your controller define the function you specified in previous step and use the '$invali property of form object to check if form valid ornot.

4. What is the use of Angular JS controllers in the application?

→. The main uses of controllers are-

a. Separating concerns and creating a clean application architecture.

b. Binding data to the view through the '$Scope' object.

c. Handling user input and events by defining event handlers and functions.

d. Interacting with services and APIs to retrieve and manipulate data.

c. Unit testing to ensure reliable and maintainable code.

5. What is the use of Angular JS filters in the application?

→1. Formatting data for display ~~of~~ in a variety of ways.

2. Sorting data in ascending or descending order, and by multiple criteria.

3. Filtering data based on specific criteria, such as value or property.

4. Paginating large datasets for better user experience.

5. Creating custom filters for maximum flexibility and customization.

Implementation :

Q1.

Code :

app.js

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

app.controller('todoController', function($scope) {
    // Initialize the list of todos
    $scope.todos = [];

    // Add a new todo item to the list
    $scope.addTodo = function() {
        if ($scope.newTodoText) {
            $scope.todos.push({
                text: $scope.newTodoText,
                done: false
            });
            $scope.newTodoText = '';
        }
    };

    // Remove completed todo items from the list
    $scope.removeCompleted = function() {
        for (var i = $scope.todos.length - 1; i >= 0; i--) {
            if ($scope.todos[i].done) {
                $scope.todos.splice(i, 1);
            }
        }
    };

    // Update an existing todo item in the list
    $scope.updateTodo = function(todo) {
        todo.editing = false;
        todo.text = todo.updatedText;
    };
```

```
  // Enable editing mode for a todo item
  $scope.editTodo = function(todo) {
    todo.editing = true;
    todo.updatedText = todo.text;
  };
});

app.directive('confirmClick', ['$window', function($window) {
  return {
    restrict: 'A',
    link: function(scope, element, attr) {
      var msg = attr.confirmClick || 'Are you sure?';
      var clickAction = attr.ngClick;
      element.bind('click', function(event) {
        if ($window.confirm(msg)) {
          scope.$eval(clickAction);
        }
      });
    }
  };
}]);
```

index.html

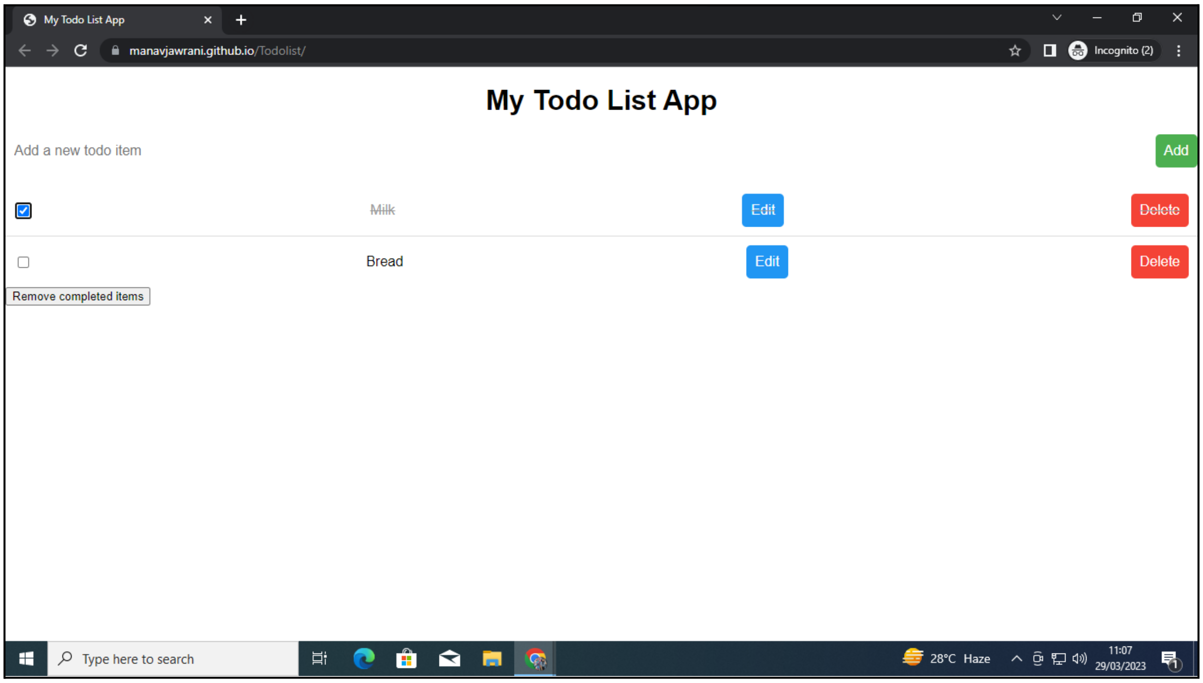
```
<!DOCTYPE html>
<html>

<head>
  <title>My Todo List App</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }
```

```
h1 {
   text-align: center;
   margin: 20px 0;
}

form {
   display: flex;
   align-items: center;
   margin-bottom: 20px;
}

form input[type="text"] {
   flex: 1;
   padding: 10px;
   font-size: 16px;
   border: none;
   border-radius: 5px;
}

form button {
   margin-left: 10px;
   padding: 10px;
   font-size: 16px;
   border: none;
   border-radius: 5px;
   background-color: #4CAF50;
   color: #fff;
   cursor: pointer;
}

ul {
   list-style: none;
   margin: 0;
   padding: 0;
```

```
}

li {
   display: flex;
   align-items: center;
   justify-content: space-between;
   padding: 10px;
   font-size: 16px;
   border-bottom: 1px solid #ddd;
}

li:last-child {
   border-bottom: none;
}

li input[type="text"] {
   flex: 1;
   padding: 10px;
   font-size: 16px;
   border: none;
   border-radius: 5px;
}

li button {
   margin-left: 10px;
   padding: 10px;
   font-size: 16px;
   border: none;
   border-radius: 5px;
   background-color: #f44336;
   color: #fff;
   cursor: pointer;
}

li button.edit {
      background-color: #2196F3;
    }

    li button.save {
      background-color: #4CAF50;
    }

    .completed {
      text-decoration: line-through;
      color: #999;
    }
  </style>
</head>

<body ng-app="myTodoListApp">
  <div ng-controller="todoController">
    <h1>My Todo List App</h1>

    <form ng-submit="addTodo()">
                  <input type="text" placeholder="Add a new todo item"
ng-model="newTodoText">
      <button type="submit">Add</button>
    </form>

    <ul>
          <li ng-repeat="todo in todos" ng-class="{completed: todo.done, editing:
todo.editing}">
        <input type="checkbox" ng-model="todo.done">
             <span ng-hide="todo.editing" ng-class="{completed: todo.done}">{{
todo.text }}</span>
        <input type="text" ng-show="todo.editing" ng-model="todo.updatedText">
                              <button class="edit" ng-hide="todo.editing"
ng-click="editTodo(todo)">Edit</button>
                             <button class="save" ng-show="todo.editing"
ng-click="updateTodo(todo)">Save</button>
```

```
                    <button confirm-click="Are you sure you want to delete this item?"
ng-click="todos.splice($index, 1)">Delete</button>
            </li>
        </ul>

          <button confirm-click="Are you sure you want to remove all completed items?"
ng-click="removeCompleted()">Remove completed items</button>

        <div ng-show="showNotification" class="notification">
            {{notificationMessage}}
        </div>
    </div>

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

</html>
```

Output :



Q2.

Code :

app.js

```
angular.module('studentApp', [])

  .controller('StudentController', function () {
    var self = this;

    // initialize empty student object and student list array
    self.newStudent = {};
    self.studentList = [];

    // function to add a new student to the list
    self.addStudent = function (studentForm) {
      // validate form inputs
      if (studentForm.$valid) { // validation occurs here
        self.studentList.push(self.newStudent);
        self.newStudent = {};
        studentForm.$setPristine();
      }
    };

    // function to delete a student from the list
    self.deleteStudent = function (index) {
      self.studentList.splice(index, 1);
    };
  });
```

index.html

```
<!DOCTYPE html>
<html lang="en" ng-app="studentApp">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
   <title>Student Management System</title>
                                                                          <script
src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.8.2/angular.min.js"></script>
   <script src="app.js"></script>
   <link rel="stylesheet" href="style.css" />
</head>

<body ng-controller="StudentController as studentCtrl">
   <h1>Student Management System</h1>
   <form name="studentForm" ng-submit="studentCtrl.addStudent(studentForm)">
      <label for="firstName">First Name:</label>
                              <input   type="text"   id="firstName"   name="firstName"
ng-model="studentCtrl.newStudent.firstName" ng-required="true">

      <label for="lastName">Last Name:</label>
                              <input   type="text"   id="lastName"   name="lastName"
ng-model="studentCtrl.newStudent.lastName" ng-required="true">

      <label for="address">Address:</label>
                                <input   type="text"   id="address"   name="address"
ng-model="studentCtrl.newStudent.address" ng-required="true">

      <label for="mobileNo">Mobile No:</label>
                              <input   type="tel"   id="mobileNo"   name="mobileNo"
ng-model="studentCtrl.newStudent.mobileNo"              ng-pattern="/^\d{10}$/"
ng-required="true">

      <label for="class">Class:</label>
                                     <input   type="text"   id="class"   name="class"
ng-model="studentCtrl.newStudent.class" ng-required="true">

      <button type="submit">Add Student</button>
   </form>
    <input type="text" ng-model="searchQuery" placeholder="Search by first name or
last name">
```

```
    <table>
        <thead>
            <tr>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Address</th>
                <th>Mobile No</th>
                <th>Class</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <tr ng-repeat="student in studentCtrl.studentList | filter:searchQuery">
                <td>{{student.firstName | uppercase}}</td>
                <td>{{student.lastName | uppercase}}</td>
                <td>{{student.address}}</td>
                <td>{{student.mobileNo}}</td>
                <td>{{student.class}}</td>
                <td><button ng-click="studentCtrl.deleteStudent($index)">Delete</button></td>
            </tr>
        </tbody>
    </table>
</body>

</html>
```

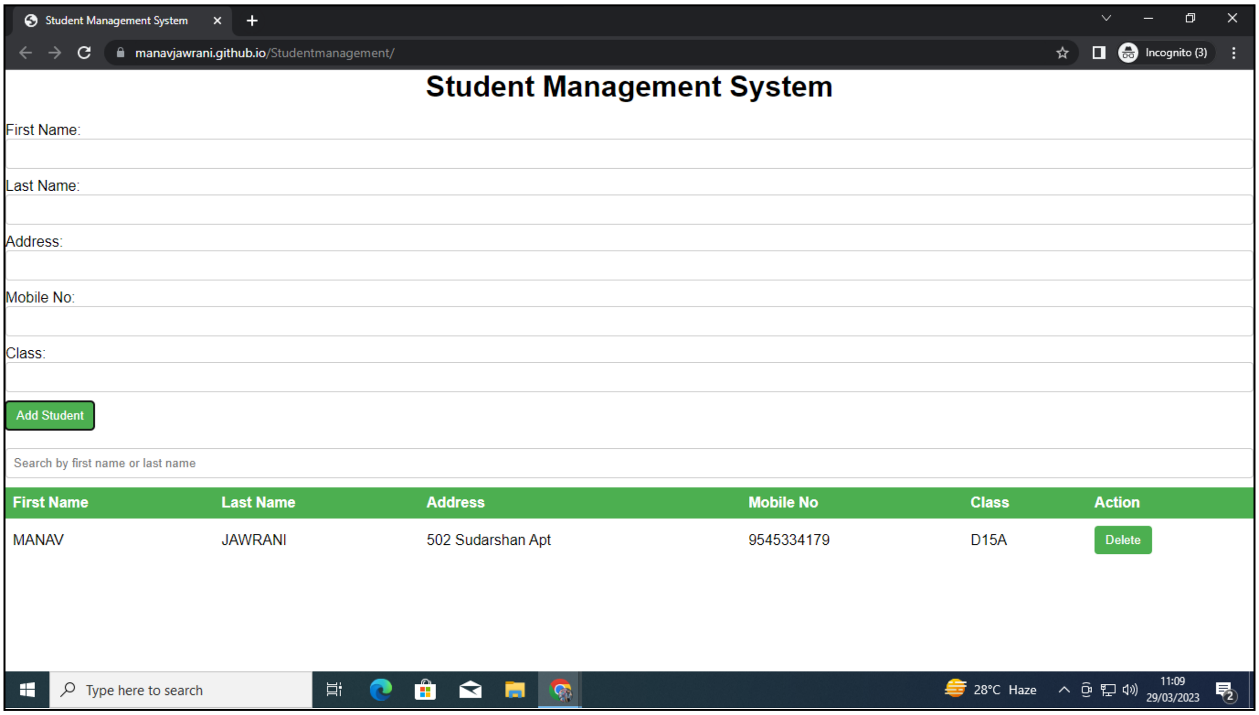
style.css

```
/* General Styles */
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}
```

```
h1 {
  margin-top: 0;
  margin-bottom: 20px;
  text-align: center;
}

form {
  margin-bottom: 20px;
}

input[type="text"],
input[type="tel"] {
  padding: 8px;
  margin-bottom: 10px;
  border-radius: 4px;
  border: 1px solid #ccc;
  width: 100%;
  box-sizing: border-box;
}

input[type="submit"],
button {
  padding: 8px 12px;
  background-color: #4CAF50;
  color: #fff;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

input[type="submit"]:hover,
button:hover {
  background-color: #3e8e41;
}
```

```
table {
    border-collapse: collapse;
    width: 100%;
}

th,
td {
    text-align: left;
    padding: 8px;
}

th {
    background-color: #4CAF50;
    color: #fff;
}

tr:nth-child(even) {
    background-color: #f2f2f2;
}

/* Responsive Styles */
@media (max-width: 768px) {

    table,
    thead,
    tbody,
    th,
    td,
    tr {
        display: block;
    }

    th {
        position: absolute;
        top: -9999px;
        left: -9999px;
    }

    td {
        border: none;
        position: relative;
        padding-left: 50%;
        text-align: right;
    }

    td:before {
        position: absolute;
        top: 6px;
        left: 6px;
        width: 45%;
        padding-right: 10px;
        white-space: nowrap;
        content: attr(data-label);
        font-weight: bold;
        text-align: left;
    }
}
```

Output :


Student Management System
manavjawrani.github.io/Studentmanagement/
Incognito (3)
Type here to search
28°C Haze
11:09
29/03/2023

Student Management System

First Name:

Last Name:

Address:

Mobile No:

Class:

Add Student

Search by first name or last name

First Name	Last Name	Address	Mobile No	Class	Action
MANAV	JAWRANI	502 Sudarshan Apt	9545334179	D15A	Delete