1. Demonstrate a simple hello world program using angular.

```
<!DOCTYPE html>

<html>

<head>

  <title>Hello World AngularJS</title>

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

</head>

<div>

  <h1>{{ 'Hello World from AngularJS!' }}</h1>

</div>

</html>
```
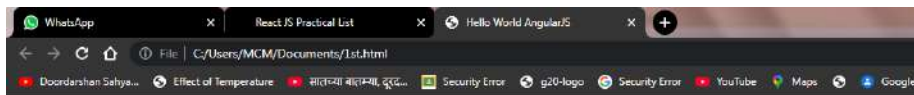
OUTPUT:

2. Demonstrate the angular js script to implement built in directives in html

```
<!DOCTYPE html>

<html>

<head>

  <title>AngularJS Addition</title>

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

</head>

<body ng-app="" style="padding: 20px;">

  <h1>{{ 'Hello World from AngularJS!' }}</h1>

 <input type="number" ng-model="num1" placeholder="First Number" />

  <input type="number" ng-model="num2" placeholder="Second Number" />

 <h3>Sum: {{ num1 + num2 }}</h3>

</body>

</html>
```
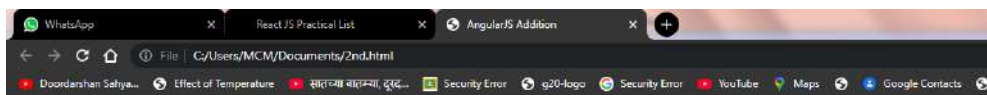
OUTPUT:

3.Demonstrate a angular js script to add modules and controller.

```html
<!DOCTYPE html>
<html>
<head>
  <title>Full Name using AngularJS</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-app="myApp" ng-controller="myCtrl" style="padding: 20px; font-family: Arial;">
  <h2>AngularJS First & Last Name Example</h2>
  <p>
    First Name:
    <input type="text" ng-model="firstName" />
  </p>
  <p>
    Last Name:
    <input type="text" ng-model="lastName" />
  </p>
  <p>
    <strong>Full Name:</strong> {{ firstName + " " + lastName }}
  </p>
  <script>
    // Define module and controller
    var app = angular.module("myApp", []);
    app.controller("myCtrl", function($scope) {
      $scope.firstName = "Aachal";
      $scope.lastName = "Borle";
    });
  </script>
</body>
</html>
```

OUTPUT:

AngularJS First & Last Name Example

First Name: Aachal

Last Name: Borle

Full Name: Aachal Borle

4)Write a angular js app for creating custom directive which display a current date and current time in elements,attributes,class and comments.

```html
<!DOCTYPE html>

<html lang="en" ng-app="myApp">

<head>

 <meta charset="UTF-8">

 <title>Custom Directive - Date & Time</title>

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

 <style>

  .custom-time {

   font-weight: bold;

   color: darkgreen;

  }

 </style>

</head>

<body ng-controller="myCtrl">

 <h2>Custom Directive to Show Date & Time</h2>

<!-- Element Directive -->

 <current-date-time></current-date-time>

 <!-- Attribute Directive -->

 <p current-date-time></p>

 <!-- Class Directive -->

 <div class="current-date-time"></div>

 <!-- Comment Directive -->

 <!-- directive: current-date-time -->

 <script>

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

  app.controller('myCtrl', function($scope) {

   // Controller not needed here for logic, included for structure

  });

  app.directive('currentDateTime', function() {

   return {
```

```
    restrict: 'EACM',  // Element, Attribute, Class, Comment

    replace: true,

    template: '<div>{{ currentTime }}</div>',

    link: function(scope, element) {

     function updateTime() {

      const now = new Date();

      scope.currentTime = now.toLocaleString();

      scope.$applyAsync(); // ensures binding updates

     }

     updateTime();

     setInterval(updateTime, 1000); // update every second

    }

   };

  });

 </script>

</body>

</html>
```

OUTPUT:

Custom Directive to Show Date & Time

7/23/2025, 9:03:49 AM
7/23/2025, 9:03:49 AM
7/23/2025, 9:03:49 AM
7/23/2025, 9:03:49 AM

5)Demonstrate number,currency,Uppercase and Lowercase filters.

```html
<!DOCTYPE html>

<html>

<head>

 <title>AngularJS: Student Info Formatting</title>

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

</head>

<body ng-app="myApp" ng-controller="myCtrl" style="padding: 20px; font-family: Arial;">

 <h2>Student Information</h2>

 <p>

  First Name:

  <input type="text" ng-model="firstName" />

 </p>

 <p>

  Last Name:

  <input type="text" ng-model="lastName" />

 </p>

 <p>

  College Name:

  <input type="text" ng-model="college" />

 </p>

 <p>

  Price:

  <input type="number" ng-model="price" />

 </p>

 <hr>

 <h3>Entered Information (Formatted):</h3>

 <p>First Name: {{ firstName | uppercase }}</p>

 <p>Last Name: {{ lastName }}</p>

 <p>College: {{ college | lowercase }}</p>

 <p>Price:{{ price | currency:'₹' }}</p>

 <script>
```

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

    app.controller("myCtrl", function($scope) {

      $scope.firstName = "";

      $scope.lastName = "";

      $scope.college = "";

      $scope.price = null;

    });

  </script>

</body>

</html>
```
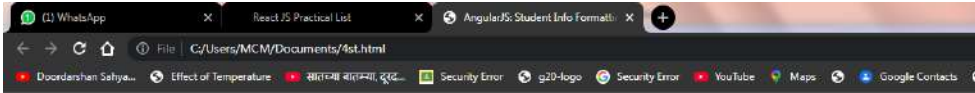
OUTPUT:

6.Write angular js app for displaying current date in 10 different format using date filter.

<!DOCTYPE html>

<html>

<head>

 <title>AngularJS Date Formats</title>

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

</head>

<body ng-app="myApp" ng-controller="myCtrl" ">

 <h2>Current Date in Different Formats</h2>

 <p>Default Format: {{ today }}</p>

 <p>Full Date: {{ today | date:'fullDate' }}</p>

 <p>Short Date: {{ today | date:'shortDate' }}</p>

 <p>Medium Date: {{ today | date:'mediumDate' }}</p>

 <p>Custom Format (dd-MM-yyyy): {{ today | date:'dd-MM-yyyy' }}</p>

 <p>Time (hh:mm:ss a): {{ today | date:'hh:mm:ss a' }}</p>

 <script>

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

  app.controller("myCtrl", function($scope) {
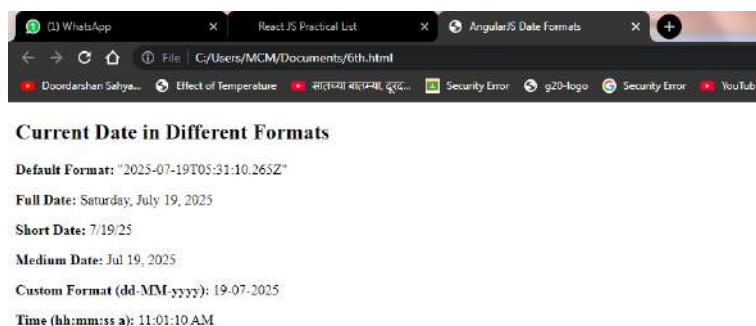
   $scope.today = new Date();

  });

 </script>

</body>

</html>

OUTPUT:

7)Write a angular js app which sort array object data in ascending and descending by using orderby filters.

```html
<!DOCTYPE html>

<html>

<head>

 <title>orderBy x.name x.country</title>

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

</head>

<body ng-app="myApp" ng-controller="myCtrl" style="padding: 20px; font-family: Arial;">

 <h2>Country List (Sorted by Name)</h2>

 <ul>

  <li ng-repeat="x in names | orderBy:'country'">

   {{ x.name+','+ x.country }}

  </li>

 </ul>

 <script>

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

  app.controller("myCtrl", function($scope) {

   $scope.names = [

    { name: 'John', country: 'USA' },

    { name: 'Amit', country: 'India' },

    { name: 'Li', country: 'China' },

    { name: 'Ken', country: 'Japan' },

    { name: 'Paul', country: 'UK' },

    { name: 'Neha', country: 'India' }

   ];

  });

 </script>

</body>

</html>
```
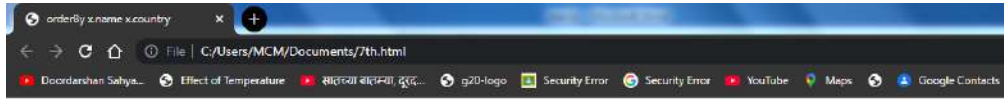
OUTPUT:

Country List (Sorted by Name)

- Li,China
- Amit,India
- Neha,India
- Ken,Japan
- Paul,UK
- John,USA

8.Demonstrate simple form using angularjs script.

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="myCtrl">
<form>
   Name: <input type="text" ng-model="name">
   <br><br>
   <button type="button" ng-click="show()">Display</button>
   <button type="button" ng-click="reset()">Reset</button>
 </form>
 <p><b>Output:</b> {{output}}</p>
<script>
   angular.module("myApp", [])
   .controller("myCtrl", function($scope){
     $scope.show = function(){ $scope.output = $scope.name; };
     $scope.reset = function(){ $scope.name = ""; $scope.output = ""; };
   });
 </script>
</body>
</html>
```

OUTPUT:

9)Demonstrate angular program that allows user to input their first name,last name and display their full name.

```html
<!DOCTYPE html>

<html lang="en" ng-app="myApp">

<head>

  <meta charset="UTF-8">

  <title>Full Name Display - AngularJS</title>

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

</head>

<body ng-controller="nameCtrl">

  <h2>Enter Your Name</h2>

  <label>First Name:</label>

  <input type="text" ng-model="firstName"><br>

<label>Last Name:</label>

  <input type="text" ng-model="lastName"><br>

  <h3>Your Full Name is:</h3>

  <p>{{ getFullName() }}</p>

  <script>

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

    app.controller('nameCtrl', function($scope) {

      $scope.firstName = '';

      $scope.lastName = '';

 $scope.getFullName = function() {

      return $scope.firstName + ' ' + $scope.lastName;

    };

   });

  </script>

</body>

</html>
```

OUTPUT:

Enter Your Name

First Name: AACHAL
Last Name: BORLE

Your Full Name is:

AACHAL BORLE

10)Implement a simple angular calculator application that can perform basic mathematical operation (addition,substraction,multiplication,division)based on user input.

```html
<!DOCTYPE html>

<html lang="en" ng-app="calcApp">

<head>

  <meta charset="UTF-8">

  <title>AngularJS Calculator</title>

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

</head>

<body ng-controller="calcCtrl">

  <h2>Simple AngularJS Calculator</h2>

  <label>Number 1:</label>

  <input type="number" ng-model="num1"><br>

  <label>Number 2:</label>

  <input type="number" ng-model="num2"><br>

  <button ng-click="add()">Add</button>

  <button ng-click="subtract()">Subtract</button>

  <button ng-click="multiply()">Multiply</button>

  <button ng-click="divide()">Divide</button>

  <h3>Result: {{ result }}</h3>

  <script>

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

    app.controller('calcCtrl', function($scope) {

      $scope.num1 = 0;

      $scope.num2 = 0;

      $scope.result = 0;

      $scope.add = function() {

        $scope.result = $scope.num1 + $scope.num2;

      };

      $scope.subtract = function() {

        $scope.result = $scope.num1 - $scope.num2;
```

```
        };

    $scope.multiply = function() {

        $scope.result = $scope.num1 * $scope.num2;

        };

    $scope.divide = function() {

        if ($scope.num2 === 0) {

            $scope.result = 'Cannot divide by zero';

        } else {

            $scope.result = $scope.num1 / $scope.num2;

        }

        };

    });

 </script>

</body>

</html>
```
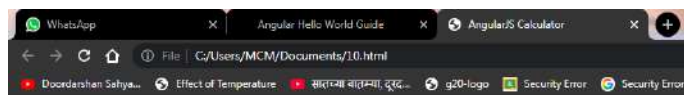
OUTPUT:

Simple AngularJS Calculator

Number 1: 4
Number 2: 4
Add | Subtract | Multiply | Divide

Result: 1

11)Demonstrate an angular application that can calculate factorial and compute square based on given user input.

```html
<!DOCTYPE html>

<html lang="en" ng-app="mathApp">

<head>

  <meta charset="UTF-8">

  <title>Factorial and Square Calculator</title>

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

</head>

<body ng-controller="mathCtrl">

  <h2>AngularJS Factorial and Square Calculator</h2>

  <label>Enter a Number:</label>

  <input type="number" ng-model="number"><br>

  <button ng-click="calculateFactorial()">Calculate Factorial</button>

  <button ng-click="calculateSquare()">Calculate Square</button>

  <h3>Factorial: {{ factorialResult }}</h3>

  <h3>Square: {{ squareResult }}</h3>

  <script>

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

    app.controller('mathCtrl', function($scope) {

      $scope.number = 0;

      $scope.factorialResult = ";

      $scope.squareResult = ";

  $scope.calculateFactorial = function() {

      let n = $scope.number;

      if (n < 0) {

        $scope.factorialResult = 'Invalid (negative number)';

      } else {

        let fact = 1;

        for (let i = 1; i <= n; i++) {

          fact *= i;

        }
```

```
        $scope.factorialResult = fact;

      }

    };

    $scope.calculateSquare = function() {

      $scope.squareResult = $scope.number * $scope.number;

    };

  });

 </script>

</body>

</html>
```
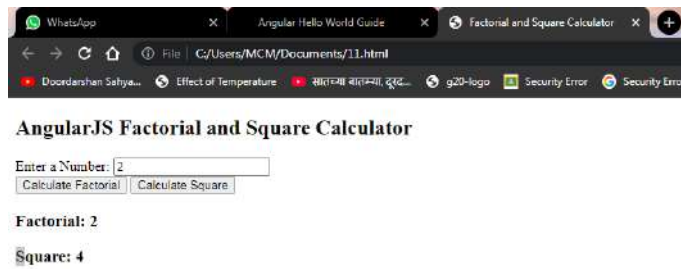
Output:

AngularJS Factorial and Square Calculator

Enter a Number: 2
Calculate Factorial Calculate Square

Factorial: 2

Square: 4

12)Implement an angular application that displays details of students and their CGPA allow users to read the number of students and display the count.

```
<!DOCTYPE html>

<html lang="en" ng-app="studentApp">

<head>

 <meta charset="UTF-8">

 <title>Student CGPA List</title>

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

</head>

<body ng-controller="studentCtrl">

<h2>Student Details with CGPA</h2>

 <table>

  <tr>

   <th>Sr. No.</th>

   <th>Name</th>

   <th>CGPA</th>

  </tr>

  <tr ng-repeat="student in students">

   <td>{{ $index + 1 }}</td>

   <td>{{ student.name }}</td>

   <td>{{ student.cgpa }}</td>

  </tr>

 </table>

 <h3>Total Number of Students: {{ students.length }}</h3>

 <script>

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

  app.controller('studentCtrl', function($scope) {

   $scope.students = [

    { name: 'Rahul', cgpa: 8.4 },

    { name: 'Sneha', cgpa: 9.1 },

    { name: 'Amit', cgpa: 7.8 },

    { name: 'Priya', cgpa: 8.9 },
```

```
      { name: 'Karan', cgpa: 7.5 }

    ];

  });

 </script>

</body>

</html>
```

OUTPUT:

Student Details with CGPA

Sr. No.	Name	CGPA
1	Rahul	8.4
2	Sneha	9.1
3	Amit	7.8
4	Priya	8.9
5	Karan	7.5

Total Number of Students: 5

13)Implement an angular program to create a login form with validation for the username and passwords fields.

```html
<!DOCTYPE html>

<html ng-app="myApp">

<head>

 <title>Simple Login Form</title>

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

</head>

<body ng-controller="loginCtrl">

 <h2>Login Form</h2>

 <form name="loginForm" ng-submit="login()" novalidate>

   Username: <input type="text" ng-model="username" name="username" required><br>

   <span ng-show="loginForm.username.$touched && loginForm.username.$invalid" style="color:red;">

    Username is required

   </span><br><br>

   Password: <input type="password" ng-model="password" name="password" required ng-minlength="6"><br>

   <span ng-show="loginForm.password.$touched && loginForm.password.$invalid" style="color:red;">

    Password is required (min 6 characters)

   </span><br><br>

   <button type="submit" ng-disabled="loginForm.$invalid">Login</button>

 </form>

 <p style="color:green;" ng-show="message">{{ message }}</p>

 <script>

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

  app.controller('loginCtrl', function($scope) {

   $scope.login = function() {

    if ($scope.username && $scope.password.length >= 6) {

     $scope.message = "Login successful!";

    }

   };

  });

 </script>
```

</body>

</html>

OUTPUT:

Login Form

Username: maheshborlemiki@gmail.com

Password: ••••••••••

Login

Login successful!

14)Implement an angular application that displays a list of shopping items.allow users to add and remove items from the list using directives and controllers.

```
<!DOCTYPE html>

<html lang="en" ng-app="shoppingApp">

<head>

 <meta charset="UTF-8">

 <title>Shopping List</title>

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

</head>

<body ng-controller="shoppingCtrl">

 <h2>Shopping List</h2>

 <!-- Input for adding new item -->

 <input type="text" ng-model="newItem" placeholder="Enter item name">

 <button ng-click="addItem()">Add Item</button>

 <!-- List of items -->

 <ul>

  <li ng-repeat="item in items">

   {{ item }}

   <button ng-click="removeItem($index)">Remove</button>

  </li>

 </ul>

 <!-- Total count -->

 <p>Total Items: {{ items.length }}</p>

 <script>

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

  app.controller('shoppingCtrl', function($scope) {

   // Initial list

   $scope.items = ['Milk', 'Bread', 'Eggs'];

   // Add new item

   $scope.addItem = function() {

    if ($scope.newItem) {

     $scope.items.push($scope.newItem);
```

```
        $scope.newItem = "; // Clear input

      }

    };

    // Remove item by index

    $scope.removeItem = function(index) {

      $scope.items.splice(index, 1);

    };

  });

</script>

</body>

</html>
```
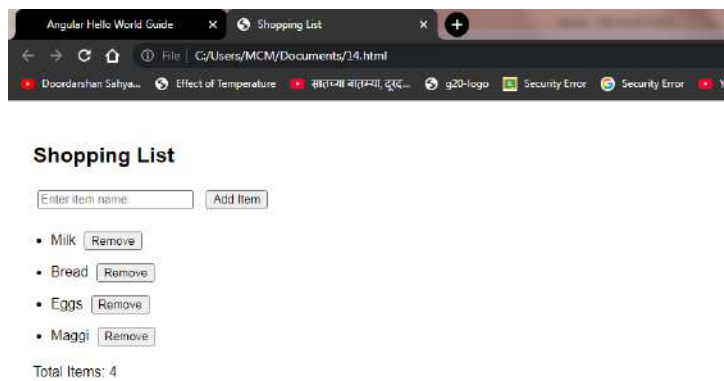
OUTPUT:

15)Implement the angular application that displays a list of employees and their salries allow users to search for employees by name and salary.

```html
<!DOCTYPE html>

<html lang="en" ng-app="employeeApp">

<head>

 <meta charset="UTF-8">

 <title>Employee Salary Search</title>

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

</head>

<body ng-controller="employeeCtrl">

 <h2>Employee List with Salary</h2>

 <!-- Search Filters -->

 <input type="text" ng-model="searchName" placeholder="Search by name">

 <input type="number" ng-model="searchSalary" placeholder="Search by salary">

 <!-- Employee Table -->

 <table>

  <tr>

   <th>Employee Name</th>

   <th>Salary</th>

  </tr>

  <tr ng-repeat="emp in employees | filter:{ name: searchName, salary: searchSalary }">

   <td>{{ emp.name }}</td>

   <td>{{ emp.salary }}</td>

  </tr>

 </table>

 <p>Total Employees: {{ (employees | filter:{ name: searchName, salary: searchSalary }).length }}</p>

 <script>

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

  app.controller('employeeCtrl', function($scope) {

   $scope.employees = [

    { name: 'Rahul', salary: 45000 },

    { name: 'Sneha', salary: 55000 },
```

```
      { name: 'Amit', salary: 35000 },

      { name: 'Priya', salary: 50000 },

      { name: 'Karan', salary: 60000 }

    ];

  });

 </script>

</body>

</html>
```

OUTPUT:

16)Demonstrate the angular program to create a simple to-do list application .allow users to add,edit and delete task.

```
<!DOCTYPE html>
<html ng-app="todoApp">
<head>
<meta charset="UTF-8">
<title>AngularJS To-Do List</title>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></s cript>
</head>
<body ng-controller="TodoController">
<h2>Simple AngularJS To-Do List</h2>
<input type="text" ng-model="newTask" placeholder="Enter task">
<button ng-click="addTask()">
    {{ editIndex !== null ? 'Update Task' : 'Add Task' }}
</button>
<ul>
<li ng-repeat="task in tasks track by $index">
    {{ task }}
<span>
<button ng-click="editTask($index)">Edit</button>
<button ng-click="deleteTask($index)">Delete</button>
</span>
</li>
</ul><script>
angular.module('todoApp', [])
    .controller('TodoController', ['$scope', function($scope) {

    $scope.tasks = [];
    $scope.newTask = '';
    $scope.editIndex = null;
    $scope.addTask = function() {        if
($scope.newTask.trim()) {          if
($scope.editIndex !== null) {
        $scope.tasks[$scope.editIndex] = $scope.newTask.trim();
        $scope.editIndex = null;
      } else {
        $scope.tasks.push($scope.newTask.trim());
      }
      $scope.newTask = '';
     }
    };
    $scope.editTask = function(index) {
     $scope.newTask = $scope.tasks[index];
     $scope.editIndex = index;
    };

    $scope.deleteTask = function(index) {
$scope.tasks.splice(index, 1);        if ($scope.editIndex
=== index) {
      $scope.newTask = '';
      $scope.editIndex = null;
     }
    };
    }]);
</script>
</body>
```
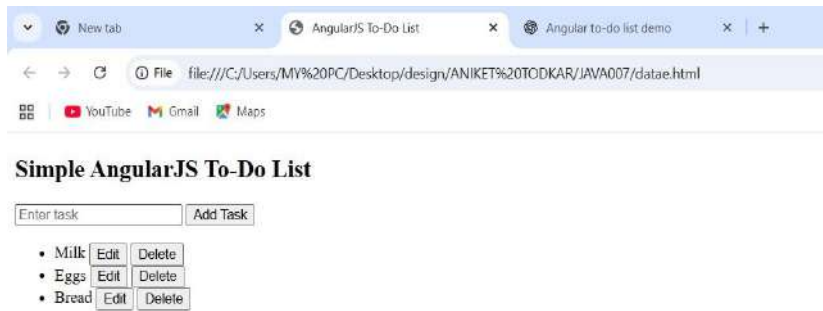
</html>

OUTPUT:

Simple AngularJS To-Do List

Enter task   Add Task

- Milk Edit   Delete
- Eggs Edit   Delete
- Bread Edit   Delete

17)Implement angular application that allows users to maintain a collection of items.the application should display current total number of items and this count should automatically update as added or removed users should be able to add items to the collection and remove them as needed.

Step1:Create Angular application

ng new item-collection-app

cd item-collection-app

ng serve

Step2:Generate component

ng generate component item-collection

Step3:item-collection-component.ts

```
import { Component } from '@angular/core';

@Component({

  selector: 'app-item-collection',

  templateUrl: './item-collection.component.html',

  styleUrls: ['./item-collection.component.css']

})

export class ItemCollectionComponent {

  items: string[] = [];  // Array to store items

  newItem: string = '';  // Input model

// Add item to collection

  addItem() {

   if(this.newItem.trim() !== '') {

    this.items.push(this.newItem.trim());

    this.newItem = '';

   }

  }

  // Remove item by index

  removeItem(index: number) {

   this.items.splice(index, 1);

  }

  // Get total items count

  get totalItems(): number {
```

```
    return this.items.length;
  }
}
```

item-collection-component.html

```html
<div class="container">
  <h2>Item Collection</h2>
    <p><strong>Total Items:</strong> {{ totalItems }}</p>
<input [(ngModel)]="newItem" placeholder="Enter item" />
  <button (click)="addItem()">Add Item</button>
 <ul>
    <li *ngFor="let item of items; let i = index">
     {{ item }}
     <button (click)="removeItem(i)">Remove</button>
    </li>
  </ul>
</div>
```

Step4:app.module.ts

```typescript
import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import { FormsModule } from '@angular/forms';

import { AppComponent } from './app.component';

import { ItemCollectionComponent } from './item-collection/item-collection.component';

@NgModule({
 declarations: [
   AppComponent,
   ItemCollectionComponent
 ],
 imports: [
   BrowserModule,
   FormsModule   // Required for ngModel
 ],
 providers: [],
```

```
  bootstrap: [AppComponent]

})

export class AppModule { }
```

Step5:display component in app template(app.component.html)

```
<app-item-collection></app-item-collection>
```

OUTPUT:

Item Collection

Total Items: 3

Enter item   Add Item

• Item 1
• Item 2   Remove
• Item 3   Remove
• Item 4   Remove

18)Demonstrate installation steps in react js.

19)Demonstrate a Simple Hello world program using React js.

function App() {

 return (

<div>

<h1>Hello World from React!</h1>

</div>

 );

}

export default App;

Output:
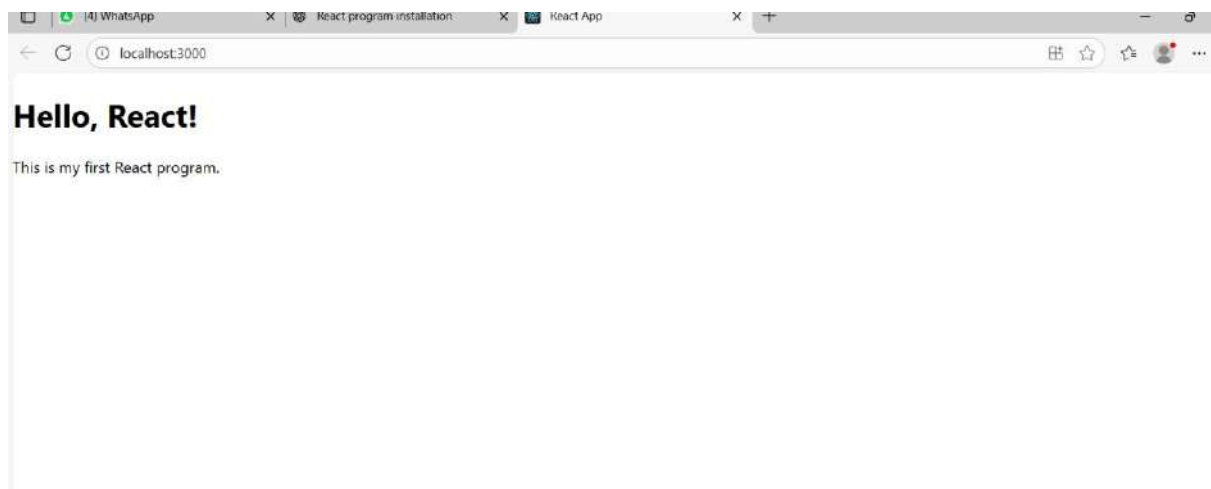
20)Demonstarte a simple reactjs application to navigate pages using routing

```jsx
import React from 'react';
import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';
// Home Page Component
function Home() {
  return (
<div>
<h2>Home Page</h2>
<p>Welcome to the Home page!</p>
</div>
  );
}
// About Page Component
function About() {
  return (
<div>
<h2>About Page</h2>
<p>This is the About page.</p>
</div>
  );
}
// App Component with Routing
function App() {
  return (
<Router>
<div>
<nav>
<Link to="/">Home</Link> | <Link to="/about">About</Link>
</nav>
<hr />
<Routes>
<Route path="/" element={<Home />} />
```

```
<Route path="/about" element={<About />} />

</Routes>

</div>

</Router>

);

}

export default App;
```

OUTPUT:
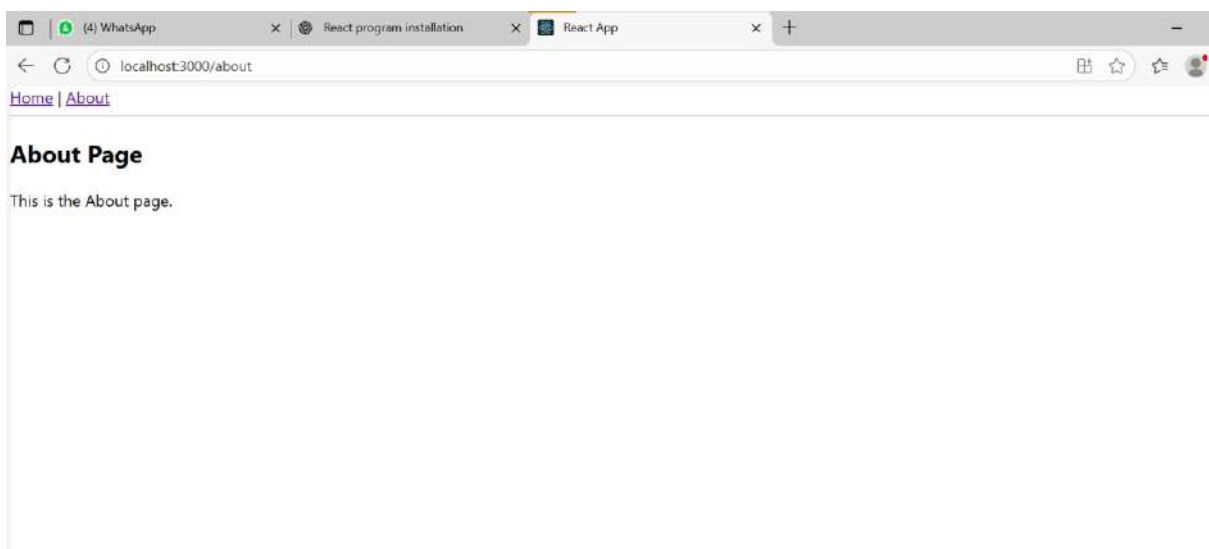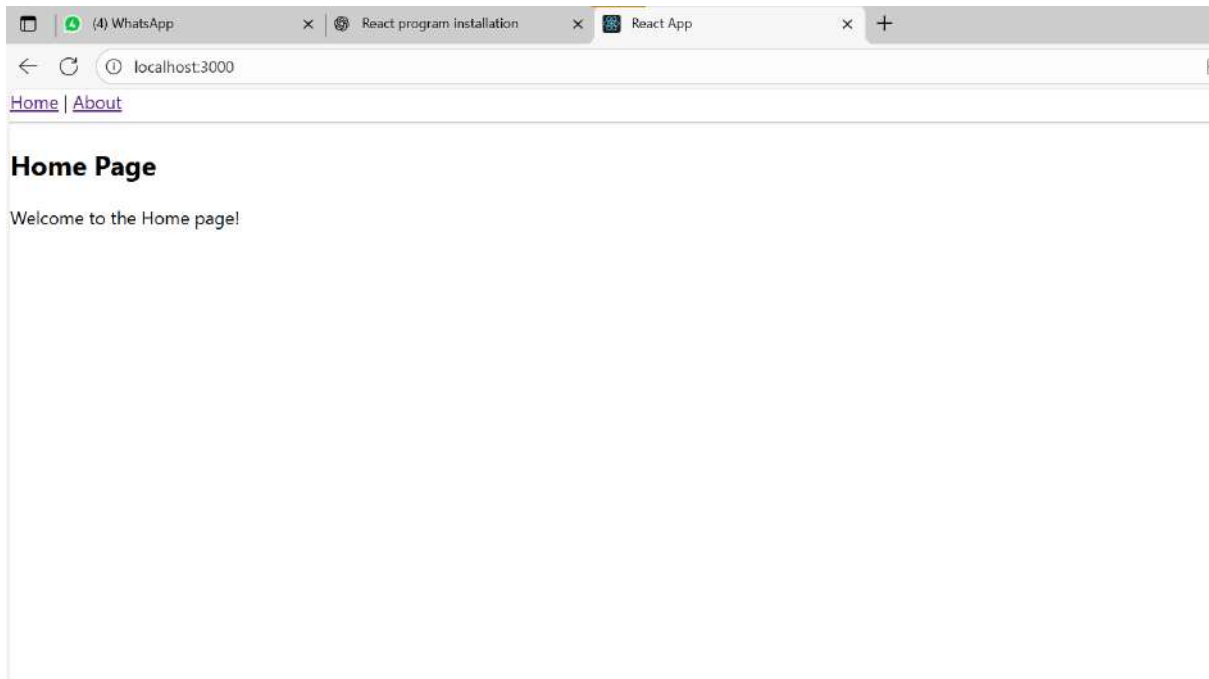
21)Implement a html document to demonstrating the slideup ,slide down ,and slide toggle using jquery

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>jQuery Slide Effects Demo</title>

<!-- jQuery CDN -->

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<style>

     #box {

        width: 300px;

        height: 150px;

        background-color: lightblue;

        line-height: 150px;

        text-align: center;

        margin: 20px auto;

        font-size: 20px;

     }

     button {

        margin: 5px;

        padding: 10px 15px;

        font-size: 16px;

     }

</style>

</head>

<body>

<h2 style="text-align:center;">jQuery Slide Effects</h2>

<div id="box">Slide Me!</div>

<div style="text-align:center;">

<button id="slideUpBtn">Slide Up</button>

<button id="slideDownBtn">Slide Down</button>
```

```
<button id="slideToggleBtn">Slide Toggle</button>

</div>

<script>

$(document).ready(function(){

    // Slide Up

    $("#slideUpBtn").click(function(){

        $("#box").slideUp("slow");

    });

    // Slide Down

    $("#slideDownBtn").click(function(){

        $("#box").slideDown("slow");

    });

    // Slide Toggle

    $("#slideToggleBtn").click(function(){

        $("#box").slideToggle("slow");

    });

});

</script>

</body>

</html>
```

OUTPUT:

Slide up

jQuery Slide Effects
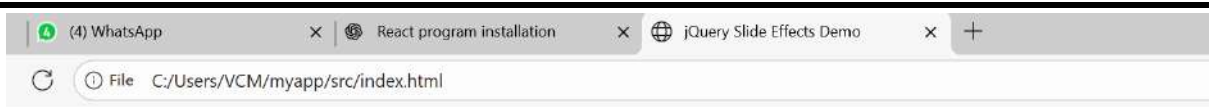
Slide Up   Slide Down   Slide Toggle

Slide down

jQuery Slide Effects

Slide Me!

Slide Up   Slide Down   Slide Toggle

Slide toggle

jQuery Slide Effects

Slide Up   Slide Down   Slide Toggle

22)Create a React service to fetch weather information from OpenWeatherMap API and display current & historical weather using Chart.js.

Step1:set up a your react project

npx create-react-app weather-app

cd weather-app

```
Microsoft Windows [Version 10.0.19045.5608]
(c) Microsoft Corporation. All rights reserved.

C:\Users\VCM>npx create-react-app weather-app

Creating a new React app in C:\Users\VCM\weather-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...


added 1323 packages in 4m

270 packages are looking for funding
  run `npm fund` for details
Git repo not initialized Error: Command failed: git --version
    at genericNodeError (node:internal/errors:983:15)
    at wrappedFn (node:internal/errors:537:14)
    at checkExecSyncError (node:child_process:916:11)
    at execSync (node:child_process:988:15)
    at tryGitInit (C:\Users\VCM\weather-app\node_modules\react-scripts\scripts\init.js:46:5)
    at module.exports (C:\Users\VCM\weather-app\node_modules\react-scripts\scripts\init.js:276:7)
    at [eval]:3:14
    at runScriptInThisContext (node:internal/vm:209:10)
    at node:internal/process/execution:446:12
    at [eval]-wrapper:6:24 {
  status: 1,
  signal: null,
  output: [ null, null, null ],
  pid: 11112,
  stdout: null,
  stderr: null
}

Installing template dependencies using npm...

added 17 packages, and changed 1 package in 19s

270 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...
```

Install dependancies:

```
removed 1 package, and audited 1340 packages in 7s

270 packages are looking for funding
  run `npm fund` for details

9 vulnerabilities (3 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

Success! Created weather-app at C:\Users\VCM\weather-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd weather-app
  npm start

Happy hacking!

C:\Users\VCM>cd weather
The system cannot find the path specified.

C:\Users\VCM>cd weather-app
```

Step2: Create a weather service

Create a file src/services/weatherService.js:

```
import axios from "axios";

const API_KEY = "YOUR_API_KEY_HERE"; // Replace with your OpenWeatherMap API key

const BASE_URL = "https://api.openweathermap.org/data/2.5";

// Fetch current weather by city

export const getCurrentWeather = async (city) => {

  const response = await axios.get(`${BASE_URL}/weather`, {

    params: {

      q: city,

      units: "metric",

      appid: API_KEY,

    },

  });

  return response.data;

};

// Fetch historical weather by coordinates and timestamp

export const getHistoricalWeather = async (lat, lon, dt) => {

  const response = await axios.get(`${BASE_URL}/onecall/timemachine`, {

    params: {
```

```
      lat: lat,

      lon: lon,

      dt: dt, // UNIX timestamp (seconds)

      units: "metric",

      appid: API_KEY,

    },

  });

  return response.data;

};
```

Create a weather chart component

Create src/components/WeatherChart.js:

```
import React, { useState, useEffect } from "react";

import { getCurrentWeather, getHistoricalWeather } from "../services/weatherService";

import { Line } from "react-chartjs-2";

import { Chart, registerables } from "chart.js";

Chart.register(...registerables);

const WeatherChart = () => {

  const [city, setCity] = useState("London"); // Default city

  const [currentWeather, setCurrentWeather] = useState(null);

  const [historicalData, setHistoricalData] = useState([]);

  const [loading, setLoading] = useState(false);

  const [error, setError] = useState("");

  // Fetch weather when city changes

  const fetchWeather = async () => {

    if (!city) return;

    setLoading(true);

    setError("");

    setCurrentWeather(null);

    setHistoricalData([]);
```

```
    try {

      // Current weather

      const current = await getCurrentWeather(city);

      setCurrentWeather(current);

      // Coordinates

      const lat = current.coord.lat;

      const lon = current.coord.lon;

      const historical = [];

      for (let i = 1; i <= 5; i++) {

        const dt = Math.floor((Date.now() - i * 24 * 60 * 60 * 1000) / 1000); // UNIX timestamp

        try {

          const data = await getHistoricalWeather(lat, lon, dt);

          historical.push({ dt: `-${i} day`, temp: data.current.temp });

        } catch (err) {

          console.error(`Failed day ${i}:`, err);

        }

      }

      setHistoricalData(historical.reverse());

    } catch (err) {

      console.error(err);

      setError("Failed to fetch weather. Check city name or API key.");

    } finally {

      setLoading(false);

    }

  };

  return (

<div style={{ width: "600px", margin: "20px auto", textAlign: "center" }}>

<h2>Weather Info</h2>
```

```jsx
      {/* City Input */}

<div style={{ marginBottom: "20px" }}>

<input

    type="text"

    value={city}

    onChange={(e) => setCity(e.target.value)}

    placeholder="Enter city"

    style={{ padding: "5px 10px", fontSize: "16px" }}

  />

<button onClick={fetchWeather} style={{ marginLeft: "10px", padding: "5px 10px" }}>

    Get Weather

</button>

</div>

      {/* Loading */}

      {loading &&<p>Loading...</p>}

      {/* Error */}

      {error &&<p style={{ color: "red" }}>{error}</p>}

      {/* Current Weather */}

      {currentWeather && (

<div style={{ marginBottom: "20px" }}>

<p>

<strong>City:</strong> {currentWeather.name}

</p>

<p>

<strong>Temperature:</strong> {currentWeather.main.temp} °C

</p>

<p>

<strong>Weather:</strong> {currentWeather.weather[0].description}

</p>
```

```jsx
      <p>
        <strong>Humidity:</strong> {currentWeather.main.humidity} %
      </p>
    </div>
  )}

  {/* Historical Chart */}
  {historicalData.length > 0 && (
    <div>
      <h3>Past 5 Days Temperature</h3>
      <Line
        data={{
          labels: historicalData.map((d) => d.dt),
          datasets: [
            {
              label: "Temperature (°C)",
              data: historicalData.map((d) => d.temp),
              fill: false,
              borderColor: "blue",
              tension: 0.1,
            },
          ],
        }}
      />
    </div>
  )}
</div>
  );
};
```

```
export default WeatherChart;
```

Use Component in App.js

Replace src/App.js with:

```
import React from "react";

import WeatherChart from "./components/WeatherChart";

function App() {

  return (

<div className="App">

<WeatherChart />

</div>

  );

}

export default App;
```
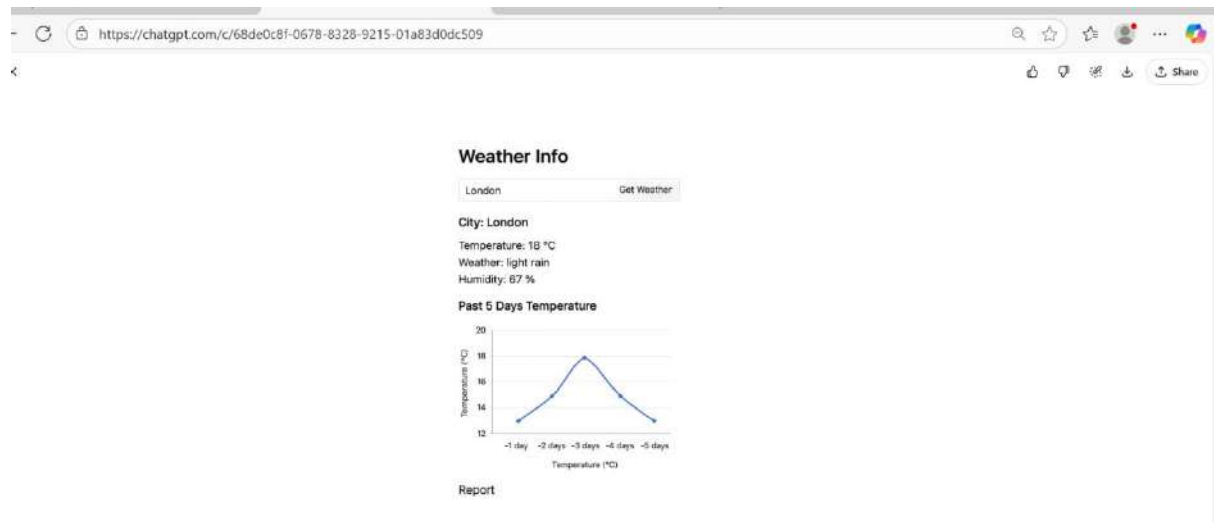
Step4:run the react app in cmd

23) Implement a simple login from using react js

Step1:Create react app

npx create-react-app login-form

cd login-form

npm start
Step2:components/Login.js

```
import React, { useState } from "react";

function Login()

  const [email, setEmail] = useState("");

  const [password, setPassword] = useState("");

  const [message, setMessage] = useState("");

  const handleSubmit = (e) => {

e.preventDefault();

  // Simple validation

  if (email === "user@example.com" && password === "123456") {

    setMessage("Login successful!");

  } else {

    setMessage("Invalid email or password.");

  }

  };

return (

<div style={{ maxWidth: "400px", margin: "50px auto", padding: "20px", border: "1px solid #ccc",
borderRadius: "8px" }}>

<h2>Login Form</h2>

<form onSubmit={handleSubmit}>

<div style={{ marginBottom: "10px" }}>

<label>Email:</label>

<input  type="email"

 value={email}

  onChange={(e) => setEmail(e.target.value)}

 style={{ width: "100%", padding: "8px", marginTop: "5px" }}  required />

</div>
```

```jsx
<div style={{ marginBottom: "10px" }}>

<label>Password:</label>

<input

    type="password"

    value={password}

    onChange={(e) => setPassword(e.target.value)}

    style={{ width: "100%", padding: "8px", marginTop: "5px" }}

    required

  />

</div>

<button type="submit" style={{ padding: "10px 20px" }}>

    Login

</button>

</form>

  {message &&<p style={{ marginTop: "20px", fontWeight: "bold" }}>{message}</p>}

</div>

  );

}
export default Login;
```

Step3:Use Login.js in App.js

Replace App.js content with:

Step4:run thenapp npm start

```jsx
import React from "react";

import Login from "./components/Login"; // import from components folder

function App() {

  return (

<div>

<Login />

</div>

  );

}
```
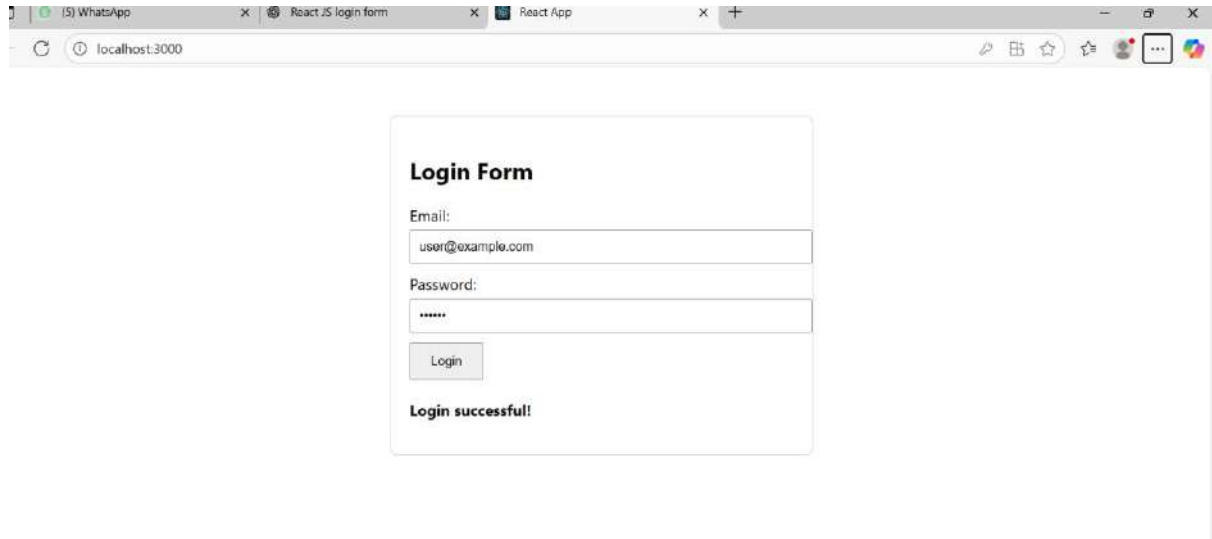
export default App;

Output:

Login Form

Email:

user@example.com

Password:

••••••

Login

Login successful!

24)Demonstrate a Events in React js

1)components/eventdemo.js
```
importReact, { useState } from"react";

functionEventDemo() {
const [count, setCount] = useState(0);
const [inputValue, setInputValue] = useState("");

// Button click event
consthandleClick = () => {
setCount(count + 1);
  };

// Input change event
consthandleChange = (e) => {
setInputValue(e.target.value);
  };

// Form submit event
consthandleSubmit = (e) => {
   e.preventDefault();
alert(`Form submitted with value: ${inputValue}`);
  };

return (
<divstyle={{maxWidth: "400px", margin: "50pxauto", padding: "20px", border: "1pxsolid #ccc", borderRadius:
"8px" }}>
<h2>React Events Demo</h2>

    {/* Click Event */}
<divstyle={{marginBottom: "20px" }}>
<buttononClick={handleClick}>Click Me</button>
<p>Button clicked {count} times</p>
</div>

    {/* Input Change Event */}
<divstyle={{marginBottom: "20px" }}>
<input
type="text"
placeholder="Type something..."
value={inputValue}
onChange={handleChange}
style={{width: "100%", padding: "8px" }}
     />
<p>Input value: {inputValue}</p>
</div>

    {/* Form Submit Event */}
<formonSubmit={handleSubmit}>
<input
type="text"
placeholder="Enter text for submit"
value={inputValue}
onChange={handleChange}
style={{width: "100%", padding: "8px", marginBottom: "10px" }}
     />
<buttontype="submit">Submit</button>
</form>
</div>
  );
}
```

```
exportdefaultEventDemo;

2)App.js (use the component)
import React from "react";
import EventDemo from "./components/EventDemo";
function App() {
  return (
<div>
<EventDemo />
</div>
  );
}
export default App;
```

OUTPUT:

25)Demonstrate search filter in react js.

components/SearchFilter.js

```
import React, { useState } from "react";

function SearchFilter() {
  const [searchTerm, setSearchTerm] = useState("");

  // Sample data

  const items = [
    "Apple",
    "Banana",
    "Orange",
    "Mango",
    "Pineapple",
    "Grapes",
    "Strawberry"
  ];
// Filtered list based on search term
  const filteredItems = items.filter((item) =>
    item.toLowerCase().includes(searchTerm.toLowerCase())
  );

  return (
<div style={{ maxWidth: "400px", margin: "50px auto", padding: "20px", border: "1px solid #ccc",
borderRadius: "8px" }}>
<h2>Search Filter Demo</h2>

    {/* Search Input */}
<input
    type="text"
    placeholder="Search items..."
    value={searchTerm}
    onChange={(e) => setSearchTerm(e.target.value)}
    style={{ width: "100%", padding: "8px", marginBottom: "20px" }}
  />
```

```jsx
      {/* Filtered List */}

<ul>

    {filteredItems.length > 0 ? (

     filteredItems.map((item, index) =><li key={index}>{item}</li>)

    ) : (

<li>No items found</li>

    )}

</ul>

</div>

  );

}

export default SearchFilter;
```

App.js (use the component)

```jsx
import React from "react";

import SearchFilter from "./components/SearchFilter";

function App() {

  return (

<div>

<SearchFilter />

</div>

  );

}

export default App;
```

OUTPUT:
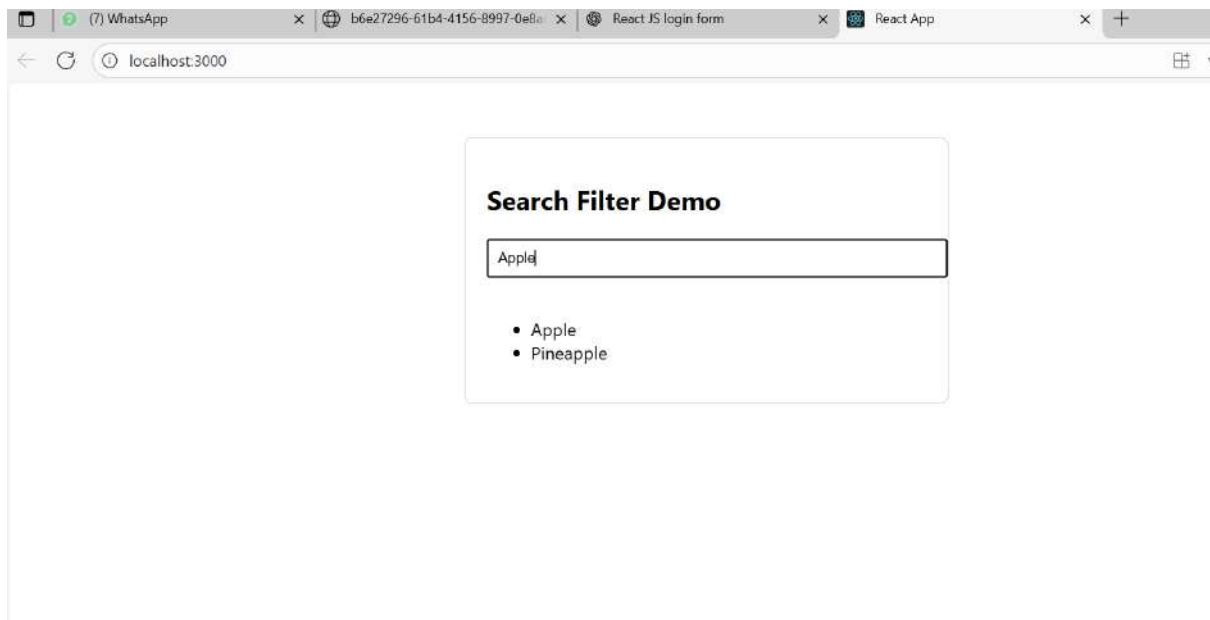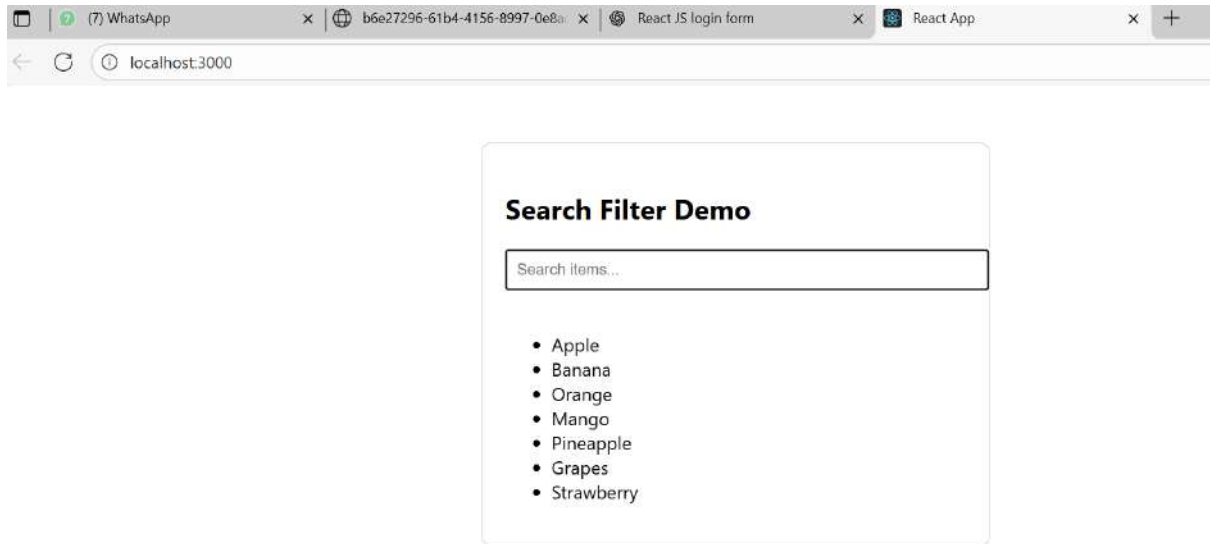
26)Implement a program to create a simple calculator application using reactjs.

components/Calculator.js

```
import React, { useState } from "react";

function Calculator() {

 const [num1, setNum1] = useState("");

 const [num2, setNum2] = useState("");

 const [result, setResult] = useState("");

 // Handle operation

 const calculate = (operation) => {

  const a = parseFloat(num1);

  const b = parseFloat(num2);

  if (isNaN(a) || isNaN(b)) {

   setResult("Please enter valid numbers");

   return;

  }

  switch (operation) {

   case "add":

    setResult(a + b);

    break;

   case "subtract":

    setResult(a - b);

    break;

   case "multiply":

    setResult(a * b);

    break;

   case "divide":

    if (b === 0) {

     setResult("Cannot divide by zero");

    } else {

     setResult(a / b);

    }

    break;
```

```jsx
      default:
        setResult("Invalid operation");
    }
  };
  return (
    <div style={{ maxWidth: "400px", margin: "50px auto", padding: "20px", border: "1px solid #ccc",
    borderRadius: "8px" }}>
      <h2>Simple Calculator</h2>
      <input
        type="number"
        placeholder="Enter first number"
        value={num1}
        onChange={(e) => setNum1(e.target.value)}
        style={{ width: "100%", padding: "8px", marginBottom: "10px" }}
      />
      <input
        type="number"
        placeholder="Enter second number"
        value={num2}
        onChange={(e) => setNum2(e.target.value)}
        style={{ width: "100%", padding: "8px", marginBottom: "10px" }}
      />
      <div style={{ display: "flex", justifyContent: "space-between", marginBottom: "10px" }}>
        <button onClick={() => calculate("add")}>+</button>
        <button onClick={() => calculate("subtract")}>-</button>
        <button onClick={() => calculate("multiply")}>*</button>
        <button onClick={() => calculate("divide")}>/</button>
      </div>
      <h3>Result: {result}</h3>
    </div>
  );
}
```

```
export default Calculator;

App.js

import React from "react";

import Calculator from "./components/Calculator";

function App() {

 return (

<div>

<Calculator />

</div>

 );

}

export default App;
```

OUTPUT:

Simple Calculator

12

34

| + | - | * | / |

Result: 408

Simple Calculator

12

34

| + | - | * | / |

Result: 0.35294117647058826

27) Implement a product page that displays detailed product info ,images,reviews and purchases options inreact js.

```
// ProductPage.js

import React, { useState } from "react";

// Sample product data

const product = {

  name: "Wireless Headphones",

  price: 149.99,

  description:

    "High-quality wireless headphones with noise cancellation and 20 hours battery life.",

  images: [

    "https://via.placeholder.com/400x300?text=Headphone+1",

    "https://via.placeholder.com/400x300?text=Headphone+2",

    "https://via.placeholder.com/400x300?text=Headphone+3",

  ],

  reviews: [

    { id: 1, name: "John", rating: 5, comment: "Excellent sound quality!" },

    { id: 2, name: "Alice", rating: 4, comment: "Very comfortable to wear." },

  ],

};

function ProductPage() {

  const [selectedImage, setSelectedImage] = useState(product.images[0]);

  const [quantity, setQuantity] = useState(1);

  const handleAddToCart = () => {

    alert(`Added ${quantity} item(s) of ${product.name} to cart.`);

  };

  return (

<div style={{ maxWidth: "800px", margin: "20px auto", fontFamily: "Arial" }}>

<h1>{product.name}</h1>

<h2>${product.price}</h2>

<p>{product.description}</p>
```

```jsx
      {/* Images */}
<div style={{ display: "flex", gap: "10px", marginBottom: "20px" }}>
<img
      src={selectedImage}
      alt="Selected"
      style={{ width: "400px", height: "300px", objectFit: "cover" }}
    />
<div style={{ display: "flex", flexDirection: "column", gap: "10px" }}>
      {product.images.map((img, index) => (
<img
          key={index}
          src={img}
          alt={`Thumbnail ${index}`}
          style={{
           width: "100px",
           height: "75px",
           objectFit: "cover",
           cursor: "pointer",
           border: selectedImage === img ? "2px solid blue" : "1px solid gray",
          }}
          onClick={() => setSelectedImage(img)}
        />
      ))}
</div>
</div>
    {/* Purchase Options */}
<div style={{ marginBottom: "20px" }}>
<label>
      Quantity:{" "}
<input
        type="number"
        min="1"
```

```jsx
        value={quantity}

        onChange={(e) => setQuantity(Number(e.target.value))}

      />

    </label>

    <button

        style={{ marginLeft: "10px", padding: "5px 15px" }}

        onClick={handleAddToCart}>

        Add to Cart

    </button>

    </div>

    {/* Reviews */}

    <div>

    <h3>Reviews</h3>

        {product.reviews.map((review) => (

    <div

            key={review.id}

            style={{ borderBottom: "1px solid #ccc", padding: "10px 0" }}>

    <strong>{review.name}</strong> - Rating: {review.rating}/5

    <p>{review.comment}</p>

    </div>

        ))}

    </div>

    </div>

  );

}

export default ProductPage;

App.js

// App.js

import React from "react";

import ProductPage from "./ProductPage";

function App() {

  return (
```

```
<div className="App">

<ProductPage />

</div>

  );

}
```

export default App;

OUTPUT:

Wireless Headphones
$149.99

High-quality wireless headphones with noise
cancellation and 20 hours battery life.

Quantity: Add to Cart

Reviews

John ★★★★★
"Excellent sound quality!"

Alice ★★★★☆
"Very comfortable to wear."

28) Implement a dynamic website using demonstratting web technologies (html,css,javascript).

Index.html

```html
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Dynamic Web Demo</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="container">

        <h1>Dynamic Item List</h1>

        <p>Total Items: <span id="item-count">0</span></p>

<input type="text" id="item-input" placeholder="Enter item name">

        <button id="add-item-btn">Add Item</button>

 <ul id="item-list">

        <!-- Dynamic items will appear here -->

    </ul>

  </div>

    <script src="script.js"></script>

</body>

</html>
```

Style.css

```css
body {

    font-family: Arial, sans-serif;

    background-color: #f9f9f9;

    display: flex;

    justify-content: center;

    padding-top: 50px;

}
```

```css
.container {

    background-color: white;

    padding: 30px;

    border-radius: 10px;

    box-shadow: 0 0 15px rgba(0,0,0,0.2);

    width: 400px;

    text-align: center;

}

input[type="text"] {

    padding: 10px;

    width: 70%;

    margin-bottom: 10px;

    border-radius: 5px;

    border: 1px solid #ccc;

}

button {

    padding: 10px 15px;

    border: none;

    border-radius: 5px;

    background-color: #28a745;

    color: white;

    cursor: pointer;

}

button:hover {

    background-color: #218838;

}

ul {

    list-style-type: none;

    padding: 0;

}
```

```css
li {
    background-color: #e9ecef;
    margin: 5px 0;
    padding: 10px;
    border-radius: 5px;
    display: flex;
    justify-content: space-between;
}
li button {
    background-color: #dc3545;
}
li button:hover {
    background-color: #c82333;
}
```

Script.js

```javascript
const addItemBtn = document.getElementById('add-item-btn');
const itemInput = document.getElementById('item-input');
const itemList = document.getElementById('item-list');
const itemCount = document.getElementById('item-count');
let count = 0;
// Function to add item
addItemBtn.addEventListener('click', () => {
    const itemName = itemInput.value.trim();
    if(itemName === "") {
        alert("Please enter an item name!");
        return;
    }
    // Create list item
    const li = document.createElement('li');
    li.textContent = itemName;
    // Create remove button
    const removeBtn = document.createElement('button');
```

```
    removeBtn.textContent = "Remove";

    removeBtn.addEventListener('click', () => {

        itemList.removeChild(li);

        count--;

        itemCount.textContent = count;

    });

    li.appendChild(removeBtn);

    itemList.appendChild(li);

    count++;

    itemCount.textContent = count;

    itemInput.value = "";

});
```
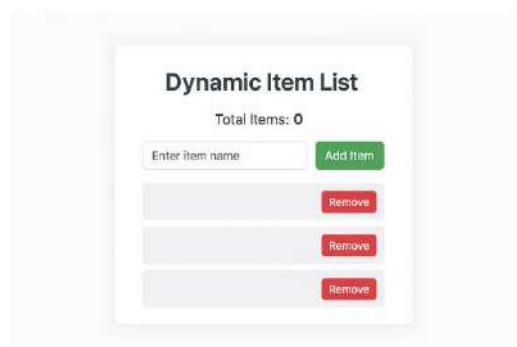
OUTPUT:

29)Implement advanced dynamic website using react js.

Step1:project setup

npx create-react-app advanced-website

cd advanced-website

npm install react-router-dom axios

npm start

Step2:project structure

src/

  components/

    Header.js

    Footer.js

    ProductCard.js

  pages/

    Home.js

    Products.js

    About.js

  App.js

  index.js

Step3:Routing in App.js

```
import React from 'react';

import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';

import Header from './components/Header';

import Footer from './components/Footer';

import Home from './pages/Home';

import Products from './pages/Products';

import About from './pages/About';

function App() {

 return (

  <Router>

   <Header />

   <Routes>

    <Route path="/" element={<Home />} />
```

```jsx
      <Route path="/products" element={<Products />} />

      <Route path="/about" element={<About />} />

    </Routes>

    <Footer />

  </Router>

 );

}

export default App;
```

Header.js

```jsx
// Header.js

import React from 'react';

import { Link } from 'react-router-dom';

function Header() {

 return (

  <header style={{ padding: '10px', background: '#333', color: 'white' }}>

   <h1>Advanced Website</h1>

   <nav>

    <Link to="/" style={{ color: 'white', margin: '10px' }}>Home</Link>

    <Link to="/products" style={{ color: 'white', margin: '10px' }}>Products</Link>

    <Link to="/about" style={{ color: 'white', margin: '10px' }}>About</Link>

   </nav>

  </header>

 );

}

export default Header;
```

```jsx
// Footer.js

import React from 'react';

function Footer() {

 return (

  <footer style={{ padding: '10px', background: '#333', color: 'white', marginTop: '20px' }}>

   <p>&copy; 2025 Advanced Website</p>

  </footer>
```

```
  );
}
export default Footer;
```

home page(dynamic content)

```
// Home.js
import React, { useState } from 'react';
function Home() {
  const [count, setCount] = useState(0);
  return (
    <div style={{ padding: '20px' }}>
      <h2>Welcome to Advanced Website</h2>
      <p>This is a dynamic website built using ReactJS.</p>
        <h3>Dynamic Counter:</h3>
      <p>Current Count: {count}</p>
      <button onClick={() => setCount(count + 1)}>Increase</button>
      <button onClick={() => setCount(count - 1)}>Decrease</button>
    </div>
  );
}
export default Home;
```

products page(API fetching)

```
// Products.js
import React, { useEffect, useState } from 'react';
import axios from 'axios';
import ProductCard from '../components/ProductCard';
function Products() {
  const [products, setProducts] = useState([]);

  useEffect(() => {
    // Fetching dummy API
    axios.get('https://fakestoreapi.com/products')
      .then(res => setProducts(res.data))
```

```jsx
      .catch(err => console.error(err));
  }, []);

  return (
    <div style={{ padding: '20px' }}>
      <h2>Products</h2>
      <div style={{ display: 'flex', flexWrap: 'wrap', gap: '20px' }}>
        {products.map(product => (
          <ProductCard key={product.id} product={product} />
        ))}
      </div>
    </div>
  );
}

export default Products;
```

productcard.js

```jsx
// ProductCard.js
import React from 'react';

function ProductCard({ product }) {
  return (
    <div style={{ border: '1px solid #ccc', padding: '10px', width: '200px' }}>
      <img src={product.image} alt={product.title} style={{ width: '100px', height: '100px' }} />
      <h4>{product.title}</h4>
      <p>Price: ${product.price}</p>
      <button>Add to Cart</button>
    </div>
  );
}

export default ProductCard;

// About.js
import React from 'react';

function About() {
  return (
```

```jsx
    <div style={{ padding: '20px' }}>

      <h2>About Us</h2>

      <p>This website demonstrates advanced ReactJS features like routing, API calls, dynamic state updates, and reusable components.</p>

    </div>

  );

}

export default About;
```
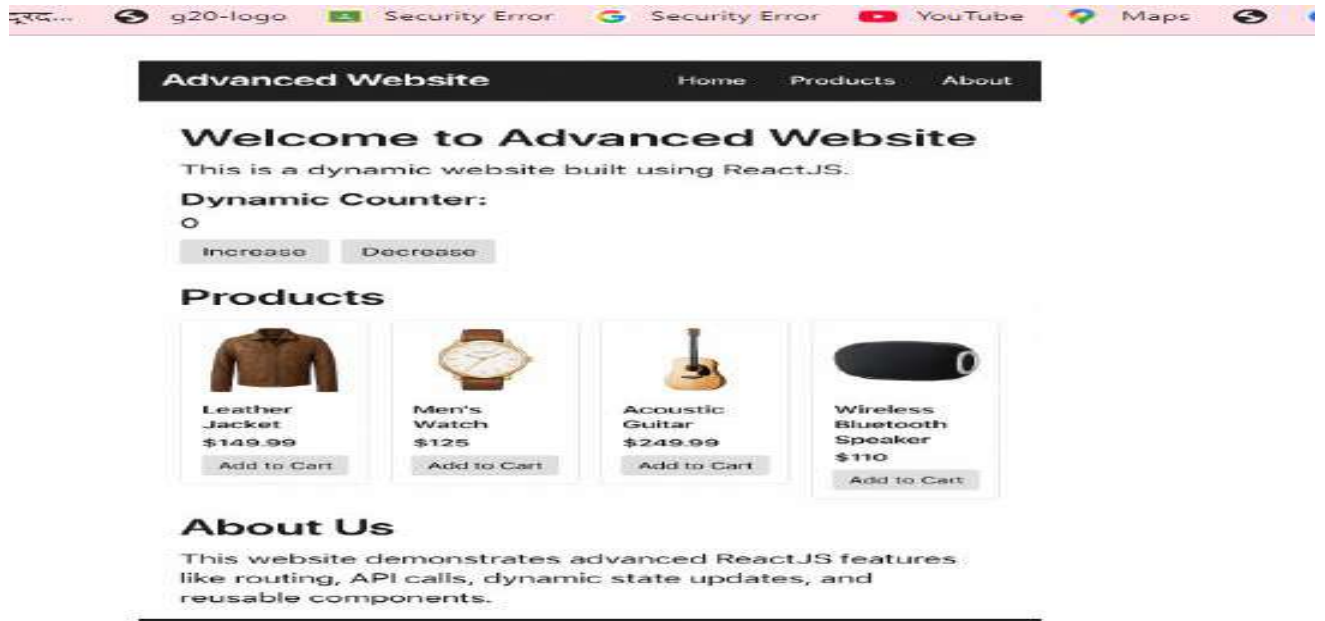
OUTPUT:

30) Create a react application for the student management system having registration.login,contact,about pages and implementing routing through navigate through those pages.

Step1:install react app

npx create-react-app student-management

cd student-management

npm install react-router-dom

npm start

Step2:project structure

src/

 ├─ components/

 │   ├─ Navbar.js

 ├─ pages/

 │   ├─ Home.js

 │   ├─ Registration.js

 │   ├─ Login.js

 │   ├─ Contact.js

 │   ├─ About.js

 ├─ App.js

 ├─ index.js

//Navbar.js

import React from "react";

import { Link } from "react-router-dom";

const Navbar = () => {

 return (

  <nav style={{ padding: "10px", backgroundColor: "#f2f2f2" }}>

   <Link to="/" style={{ margin: "10px" }}>Home</Link>

   <Link to="/registration" style={{ margin: "10px" }}>Registration</Link>

   <Link to="/login" style={{ margin: "10px" }}>Login</Link>

   <Link to="/contact" style={{ margin: "10px" }}>Contact</Link>

   <Link to="/about" style={{ margin: "10px" }}>About</Link>

  </nav>

```js
  );
};
export default Navbar;
```

Home.js

```js
import React from "react";

const Home = () => {
  return <h2>Welcome to Student Management System</h2>;
};

export default Home;
```

Registration.js

```js
import React, { useState } from "react";

const Registration = () => {
  const [formData, setFormData] = useState({ name: "", email: "", password: "" });

  const handleChange = (e) => {
    setFormData({...formData, [e.target.name]: e.target.value});
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    alert(`Registered: ${JSON.stringify(formData)}`);
  };

  return (
    <div>
      <h2>Registration</h2>
      <form onSubmit={handleSubmit}>
        <input type="text" name="name" placeholder="Name" onChange={handleChange} required /><br />
        <input type="email" name="email" placeholder="Email" onChange={handleChange} required /><br />
        <input type="password" name="password" placeholder="Password" onChange={handleChange} required /><br />
        <button type="submit">Register</button>
      </form>
    </div>
  );
```

```
};

export default Registration;

login.js

import React, { useState } from "react";

const Login = () => {

  const [email, setEmail] = useState("");

  const [password, setPassword] = useState("");

  const handleSubmit = (e) => {

   e.preventDefault();

   alert(`Login attempted with email: ${email}`);

  };

  return (

   <div>

     <h2>Login</h2>

     <form onSubmit={handleSubmit}>

      <input type="email" placeholder="Email" value={email} onChange={(e) => setEmail(e.target.value)}
required /><br />

      <input type="password" placeholder="Password" value={password} onChange={(e) =>
setPassword(e.target.value)} required /><br />

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

     </form>

   </div>

  );

};


export default Login;

contact.js

import React from "react";

const Contact = () => {

  return (

   <div>

     <h2>Contact Us</h2>

     <p>Email: info@studentms.com</p>
```

```jsx
      <p>Phone: +91 1234567890</p>

    </div>

  );

};

export default Contact;

About.js

import React from "react";

const About = () => {

  return (

    <div>

      <h2>About Student Management System</h2>

      <p>This system manages student registration, login and information efficiently.</p>

    </div>

  );

};

export default About;

Setup Routing in App.js

import React from "react";

import { BrowserRouter as Router, Routes, Route } from "react-router-dom";

import Navbar from "./components/Navbar";

import Home from "./pages/Home";

import Registration from "./pages/Registration";

import Login from "./pages/Login";

import Contact from "./pages/Contact";

import About from "./pages/About";

function App() {

  return (

    <Router>

      <Navbar />

      <div style={{ padding: "20px" }}>

        <Routes>

          <Route path="/" element={<Home />} />
```

```
          <Route path="/registration" element={<Registration />} />

          <Route path="/login" element={<Login />} />

          <Route path="/contact" element={<Contact />} />

          <Route path="/about" element={<About />} />

        </Routes>

      </div>

    </Router>

  );

}

export default App;
```

Output: