1. Demonstrate Simple Hello World program using Angular.

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

  <p>Hello {{'World'}}</p>

</body>
</html>
```

Output

2. Demonstrate Angular JS script to implement Built-in Directives.

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

<input ng-model="name" placeholder="Enter name">
<p>Hello {{name}}</p>

<ul>
  <li ng-repeat="x in ['Apple','Banana','Mango']">{{x}}</li>
</ul>

</body>
</html>
```

Output

3. Demonstrate Angular JS script to add modules and controller.

```
<!DOCTYPE html>
<html ng-app="app">
<head>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-controller="mainCtrl">

<p>Hello {{name}}</p>

<script>
angular.module("app", [])
.controller("mainCtrl", function($scope){
  $scope.name = "Vani";
});
</script>

</body>
</html>
```

Output

Hello Vani

4. Write Angular JS app for creating custom directive which displays current date and current time in elements, attributes, class, and comment.

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

<show-date></show-date>

<script>
angular.module("app", [])
.directive("showDate", () => ({ template: new Date().toLocaleString() }));
</script>

</body>
</html>
```

Output

15/10/2025, 9:41:03 am

5. Demonstrate number, currency, uppercase, and lowercase Filters.

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

<p>Number Filter: {{1234.567 | number:2}}</p>
<p>Currency Filter: {{1500 | currency}}</p>
<p>Uppercase Filter: {{"angularjs filter" | uppercase}}</p>
<p>Lowercase Filter: {{"HELLO WORLD" | lowercase}}</p>

</body>
</html>
```

Output

6. Write Angular JS app for displaying current date in 10 different formats using date filter.

```
<!DOCTYPE html>
<html ng-app="dateApp">
<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-controller="DateController">

  <h3>Current Date in Different Formats</h3>

  <p>1. Default: {{today | date}}</p>
  <p>2. Short: {{today | date:'short'}}</p>
  <p>3. Medium: {{today | date:'medium'}}</p>
  <p>4. Full Date: {{today | date:'fullDate'}}</p>
  <p>5. Long Date: {{today | date:'longDate'}}</p>
  <p>6. Short Date: {{today | date:'shortDate'}}</p>
  <p>7. Medium Date: {{today | date:'mediumDate'}}</p>
  <p>8. Short Time: {{today | date:'shortTime'}}</p>
  <p>9. Medium Time: {{today | date:'mediumTime'}}</p>
  <p>10. Custom (dd/MM/yyyy HH:mm:ss): {{today | date:'dd/MM/yyyy HH:mm:ss'}}</p>

  <script>
    angular.module('dateApp', [])
      .controller('DateController', function($scope) {
        $scope.today = new Date();
      });
  </script>

</body>
</html>
```

Output

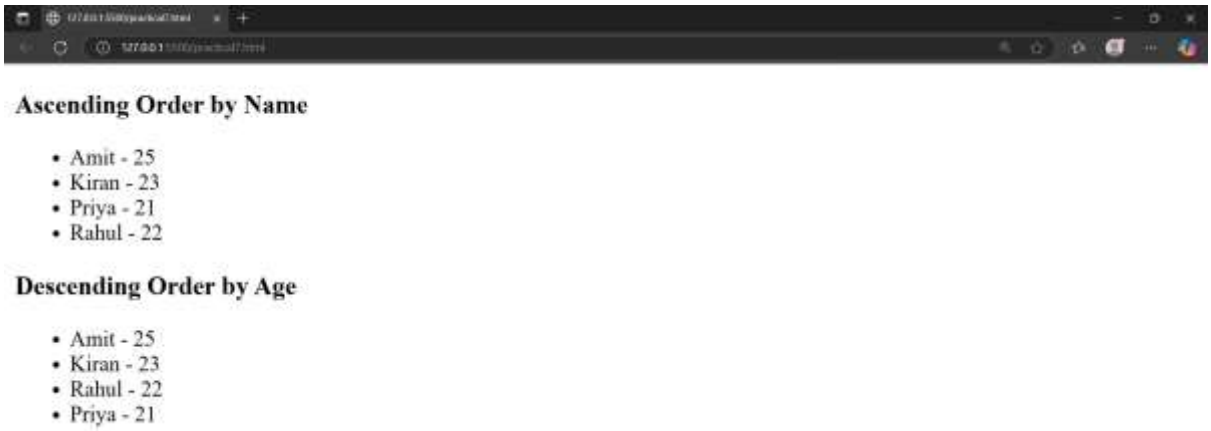
Current Date in Different Formats

1. Default: Oct 15, 2025
2. Short: 10/15/25 9:50 AM
3. Medium: Oct 15, 2025 9:50:33 AM
4. Full Date: Wednesday, October 15, 2025
5. Long Date: October 15, 2025
6. Short Date: 10/15/25
7. Medium Date: Oct 15, 2025
8. Short Time: 9:50 AM
9. Medium Time: 9:50:33 AM
10. Custom (dd/MM/yyyy HH:mm:ss): 15/10/2025 09:50:33

7. Write Angular JS app which sorts array object data in ascending and descending order using orderBy filter.

```
<!DOCTYPE html>
<html ng-app>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-init="students=[
  {name:'Rahul', age:22},
  {name:'Amit', age:25},
  {name:'Priya', age:21},
  {name:'Kiran', age:23}
]">

<h3>Ascending Order by Name</h3>
<ul>
  <li ng-repeat="s in students | orderBy:'name'">{{s.name}} - {{s.age}}</li>
</ul>

<h3>Descending Order by Age</h3>
<ul>
  <li ng-repeat="s in students | orderBy:'age':true">{{s.name}} - {{s.age}}</li>
</ul>

</body>
</html>
```

Output



Ascending Order by Name

- Amit - 25
- Kiran - 23
- Priya - 21
- Rahul - 22

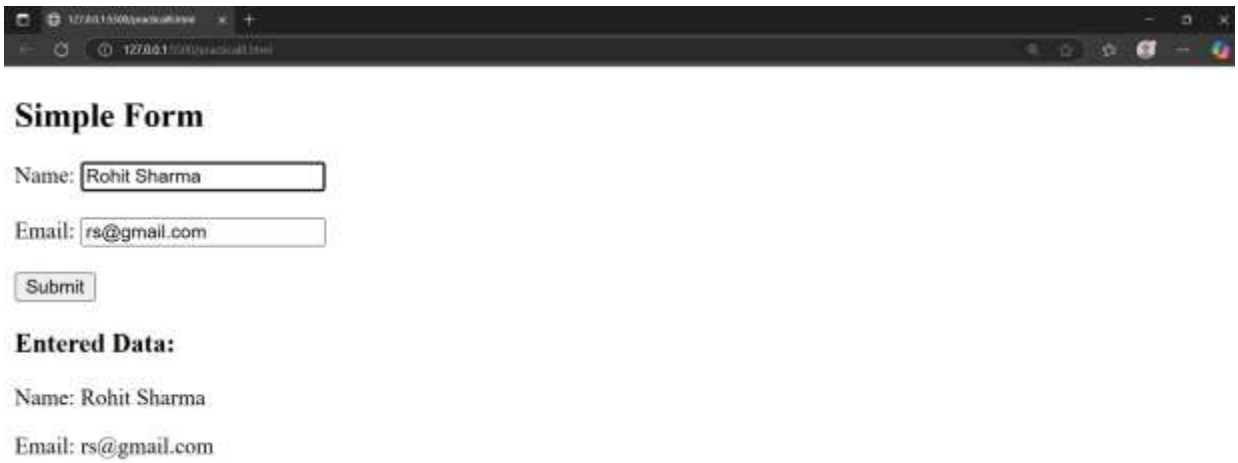
Descending Order by Age

- Amit - 25
- Kiran - 23
- Rahul - 22
- Priya - 21

8. Demonstrate simple form using Angular JS script.

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

<h2>Simple Form</h2>

<form ng-init="user={}">
  Name: <input type="text" ng-model="user.name"><br><br>
  Email: <input type="email" ng-model="user.email"><br><br>
  <button type="submit">Submit</button>
</form>

<h3>Entered Data:</h3>
<p>Name: {{user.name}}</p>
<p>Email: {{user.email}}</p>
</body>
</html>
```

Output



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

```
<!DOCTYPE html>

<html ng-app>

<head>

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

</head>

<body>


<input ng-model="f"> <input ng-model="l">

<p>Full Name: {{f + " " + l}}</p>


</body>

</html>
```

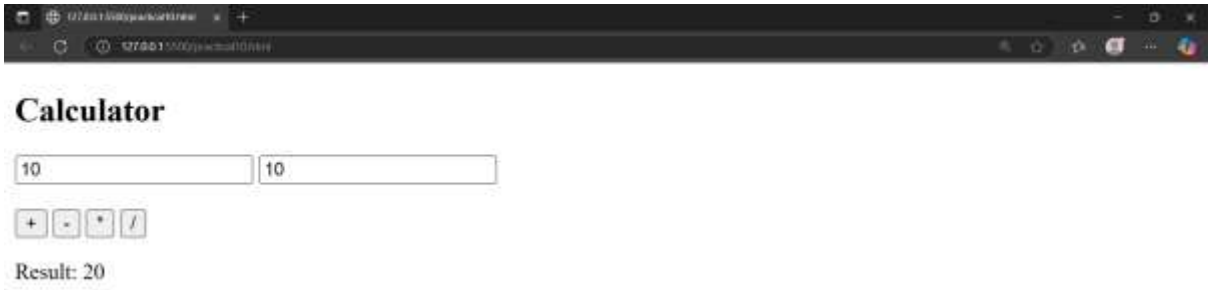
Output

10. Implement a simple Angular calculator application (addition, subtraction, multiplication, division).

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

<h2>Calculator</h2>
<div ng-init="a=0; b=0; result=0">
  <input type="number" ng-model="a" placeholder="a">
  <input type="number" ng-model="b" placeholder="b"><br><br>
  <button ng-click="result=a+b">+</button>
  <button ng-click="result=a-b">-</button>
  <button ng-click="result=a*b">*</button>
  <button ng-click="result=b!=0?a/b:'Division by zero'">/</button>
  <p>Result: {{result}}</p>
</div>

</body>
</html>
```

Output

+



-

Calculator

10 3

+ - * /

Result: 7

*

Calculator

10 3

+ - * /

Result: 30

/

Calculator

10 3

+ - * /

Result: 3.3333333333333335

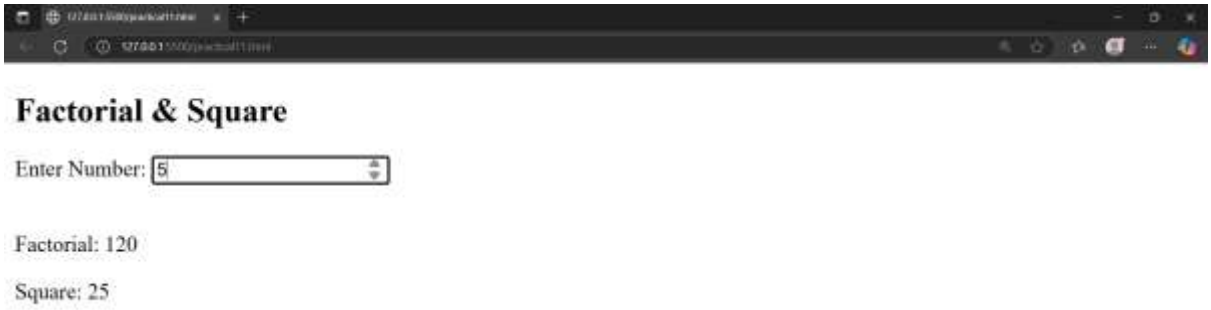
11. Demonstrate an Angular application that calculates factorial and square of a given number.

```
<!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="MainCtrl">

<h2>Factorial & Square</h2>
<div>
  Enter Number: <input type="number" ng-model="n" min="0"><br><br>
  <p>Factorial: {{ fact }}</p>
  <p>Square: {{ n*n }}</p>
</div>

<script>
  angular.module('myApp', [])
    .controller('MainCtrl', ['$scope', function($scope) {
      $scope.n = 0;
      $scope.fact = 1;

      $scope.$watch('n', function(value) {
        value = parseInt(value) || 0;
        let f = 1;
        for (let i = 1; i <= value; i++) f *= i;
        $scope.fact = f;
      });
    }]);
</script>
```

</body>

</html>

Output



12. Implement an Angular application that displays details of students and their CGPA; show student count.

```
<!DOCTYPE html>
<html ng-app="studentApp">
<body ng-controller="C as ctrl">

<h2>Student Details</h2>
Number: <input type="number" ng-model="ctrl.n" ng-change="ctrl.gen()">
<p>Total Students: {{ ctrl.students.length }}</p>

<table border="1" ng-if="ctrl.students.length">
<tr><th>Roll</th><th>Name</th><th>CGPA</th></tr>
<tr ng-repeat="s in ctrl.students">
  <td>{{s.roll}}</td><td>{{s.name}}</td><td>{{s.cgpa}}</td>
</tr>
</table>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
<script>
angular.module('studentApp', []).controller('C',function(){
  this.students=[];
  this.gen=()=>{
    this.students=[];
    for(let i=1;i<=this.n;i++) this.students.push({roll:i,name:'Student
'+i,cgpa:(Math.random()*4).toFixed(2)});
  }
});
</script>
</body>
</html>
```

Output

Student Details

Number: 5

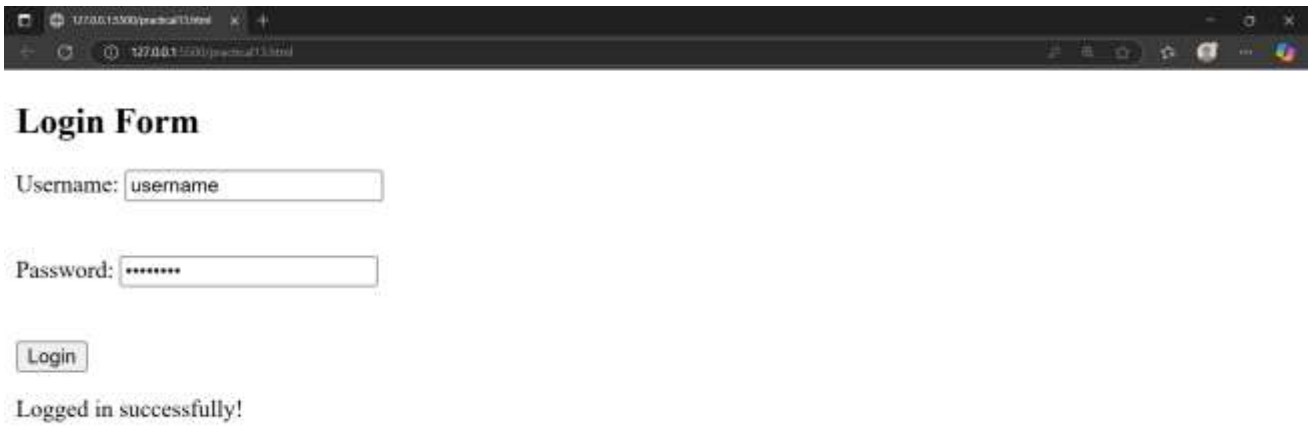
Total Students: 5

Roll	Name	CGPA
1	Student 1	2.08
2	Student 2	0.09
3	Student 3	1.96
4	Student 4	3.26
5	Student 5	2.74

13. Implement an Angular program to create a login form with validation for username and password fields.

```
<!DOCTYPE html>
<html ng-app>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-init="logged=false">
<h2>Login Form</h2>
<form name="loginForm" novalidate ng-submit="logged=true">
  Username:
  <input type="text" ng-model="username" name="username" required minlength="3"><br>
  <span ng-show="loginForm.username.$touched && loginForm.username.$invalid">Invalid Username</span><br><br>
  Password:
  <input type="password" ng-model="password" name="password" required minlength="6"><br>
  <span ng-show="loginForm.password.$touched && loginForm.password.$invalid">Invalid Password</span><br><br>
  <button type="submit" ng-disabled="loginForm.$invalid">Login</button>
</form>
<p ng-if="logged">Logged in successfully!</p>
</body>
</html>
```

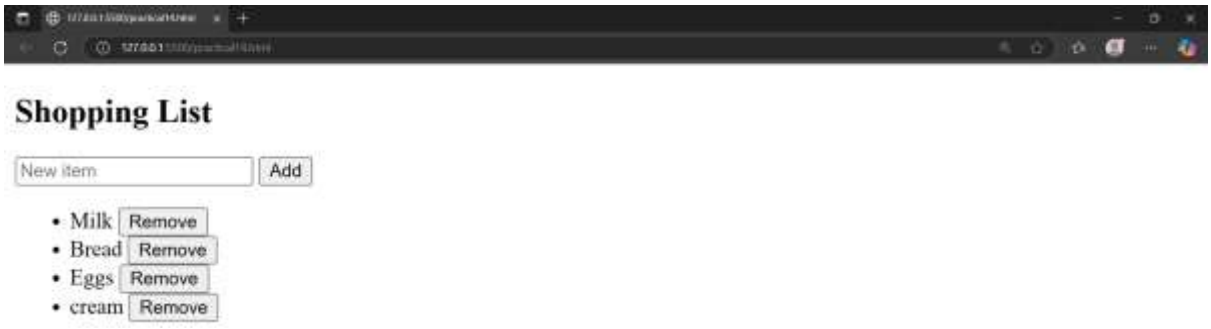
Output



14. Implement an Angular application that displays a list of shopping items; allow add/remove items using directives and controllers.

```
<!DOCTYPE html>
<html ng-app>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-init="items=['Milk','Bread','Eggs']; newItem=''">

<h2>Shopping List</h2>
<input ng-model="newItem" placeholder="New item">
<button ng-click="items.push(newItem); newItem=''">Add</button>
<ul>
  <li ng-repeat="it in items">
    {{it}} <button ng-click="items.splice($index,1)">Remove</button>
  </li>
</ul>
</body>
</html>
```

Output

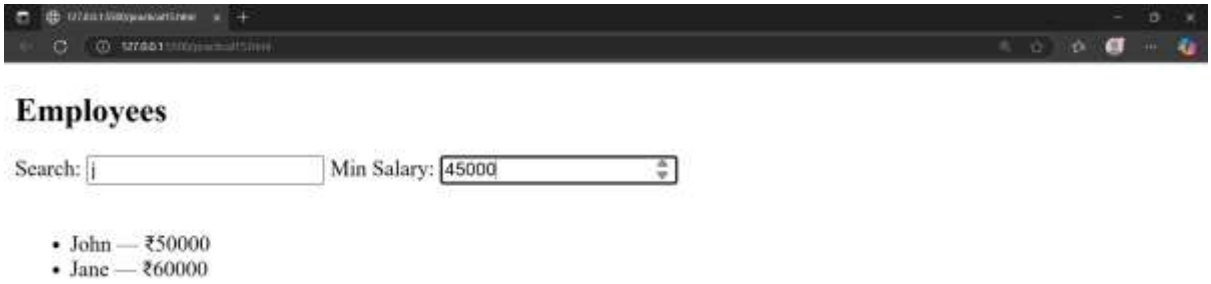


15. Implement an Angular application that displays a list of employees and their salaries; allow search by name and salary.

```
<!DOCTYPE html>
<html ng-app>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-init="
employees=[
  {name:'John',salary:50000},
  {name:'Jane',salary:60000},
  {name:'Sam',salary:45000}
]; search=''; minSalary=0">

<h2>Employees</h2>
Search: <input ng-model="search" placeholder="Name">
Min Salary: <input type="number" ng-model="minSalary"><br><br>

<ul>
  <li ng-repeat="e in employees | filter:{name:search} | filter:salaryFilter(minSalary)">
    {{e.name}} — ₹{{e.salary}}
  </li>
</ul>

<script>
  angular.module('app', []).filter('salaryFilter', function() {
    return function(items, minSalary) {
      if (!items) return [];
      return items.filter(e => e.salary >= (minSalary || 0));
    };
  });
```

```
</script>


</body>

</html>
```

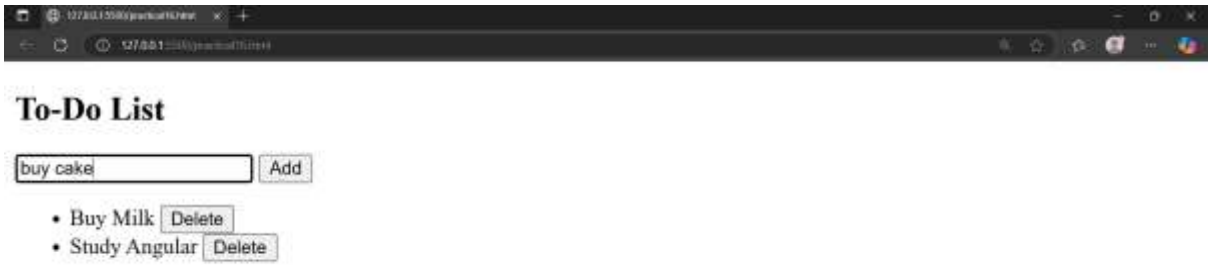
Output

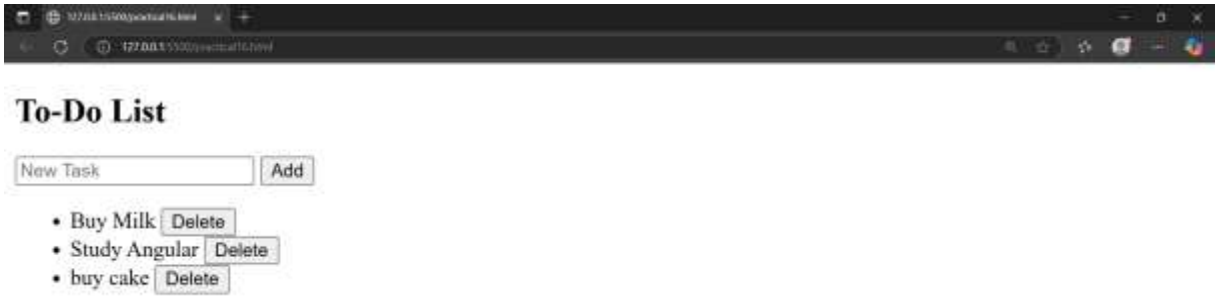


16. Demonstrate an Angular program to create a simple to-do list application; allow add, edit, delete tasks.

```
<!DOCTYPE html>
<html ng-app>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
</head>
<body ng-init="tasks=['Buy Milk','Study Angular']; newTask=''">
<h2>To-Do List</h2>
<input ng-model="newTask" placeholder="New Task">
<button ng-click="tasks.push(newTask); newTask=''">Add</button>
<ul>
  <li ng-repeat="t in tasks">
    {{t}} <button ng-click="tasks.splice($index,1)">Delete</button>
  </li>
</ul>
</body>
</html>
```

Output

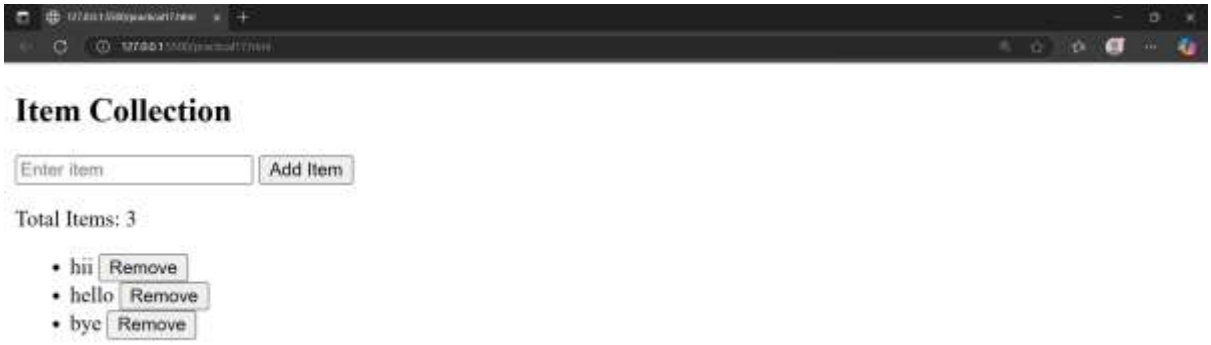




17. Implement Angular application that maintains a collection of items and automatically updates the total count as items are added/removed.

```
<!DOCTYPE html>
<html ng-app>
<body ng-init="items=[]; newItem=''">

<h2>Item Collection</h2>
<input type="text" ng-model="newItem" placeholder="Enter item">
<button ng-click="items.push(newItem); newItem=''">Add Item</button>

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

<ul>
  <li ng-repeat="item in items">
    {{ item }} <button ng-click="items.splice($index,1)">Remove</button>
  </li>
</ul>

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

Output



18. Demonstrate installation steps of React JS.

Install Node.js & npm:

To check node version

```
PS M:\Demo> node -v
v20.18.0
```

To check npm version

```
PS M:\Demo> npm -v
11.6.1
```

Create React app (using Create React App):

```
PS M:\Demo> npx create-react-app my-app
Need to install the following packages:
create-react-app@5.1.0
Ok to proceed? (y) y
```

Go to the folder

```
PS M:\Demo> cd my-app
PS M:\Demo\my-app>
```

To run the project

```
PS M:\Demo\my-app> npm start
```

```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.20.224.1:3000
```

19. Create Simple Hello World Program in React JS.

change code in src folder ----> app.js

```
import React from 'react';

function App() {
  return <h1>Hello World!</h1>;
}

export default App;
```

Output

```
PS M:\Demo\my-app> npm start
```

```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.20.224.1:3000
```

Hello World!

20. Demonstrate a simple React JS Application to navigate pages using Routing.

change code in src folder ----> app.js

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

function Home() { return <h2>Home Page</h2>; }
function About() { return <h2>About Page</h2>; }

function App() {
  return (
    <Router>
      <nav><Link to="/">Home</Link> | <Link to="/about">About</Link></nav>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </Router>
  );
}
export default App;
```

Output

```
PS M:\Demo\my-app> npm install react-router-dom

added 5 packages, and audited 1345 packages in 4s
```

```
PS M:\Demo\my-app> npm start
```

```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.20.224.1:3000
```

Home | About

Home Page

Home | About

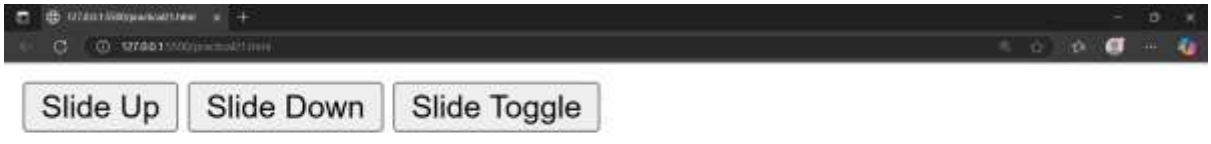
About Page

21. Implement a HTML document demonstrating Slide Up, Slide Down & Slide Toggle using jQuery.

```
<!DOCTYPE html>
<html>
<head>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>

<p id="text">Hello! Slide me!</p>
<button id="up">Slide Up</button>
<button id="down">Slide Down</button>
<button id="toggle">Slide Toggle</button>

<script>
$("#up").click(()=>$("#text").slideUp());
$("#down").click(()=>$("#text").slideDown());
$("#toggle").click(()=>$("#text").slideToggle());
</script>
</body>
</html>
```

Output

Slide up



Slide down

Slide toggle (We can do up and down)

22. Create a service in React that fetches weather information from openweathermap.org and displays current & historical data using chart.js.

change code in src folder ----> app.js

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


export default function Weather() {
  const [data, setData] = useState(null);


  useEffect(() => {
    fetch(
      'https://api.open-
meteo.com/v1/forecast?latitude=28.6139&longitude=77.209&daily=temperature_2m_max
&timezone=auto'
    )
      .then(res => res.json())
      .then(json => setData(json.daily))
      .catch(() => setData(null));
  }, []);


  if (!data) return <p>Loading weather...</p>;


  return (
    <div>
      <h2>7-Day Max Temperature Forecast for New Delhi</h2>
      <ul>
        {data.time.map((date, i) => (
          <li key={date}>
            {date}: {data.temperature_2m_max[i]}°C
          </li>
```

```
      ))}
    </ul>
  </div>
 );
}
```

Output

```
PS M:\Demo\my-app> npm install axios chart.js react-chartjs-2

added 6 packages, and audited 1351 packages in 4s
```

```
PS M:\Demo\my-app> npm start
```

```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.20.224.1:3000
```

7-Day Max Temperature Forecast for New Delhi

- 2025-10-15: 30.7°C
- 2025-10-16: 30.5°C
- 2025-10-17: 30°C
- 2025-10-18: 29.7°C
- 2025-10-19: 30°C
- 2025-10-20: 30.1°C
- 2025-10-21: 30.2°C

23. Implement a Simple Login form using React JS.

change code in src folder ----> app.js

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

function Login() {
  const [user,setUser] = useState('');
  const [pass,setPass] = useState('');
  const [msg,setMsg] = useState('');

  const handleLogin = () => setMsg(`Logged in as ${user}`);

  return (
    <div>
      <input placeholder="Username" value={user} onChange={e=>setUser(e.target.value)} />
      <input placeholder="Password" type="password" value={pass}
onChange={e=>setPass(e.target.value)} />
      <button onClick={handleLogin}>Login</button>
      <p>{msg}</p>
    </div>
  );
}
export default Login;
```

Output

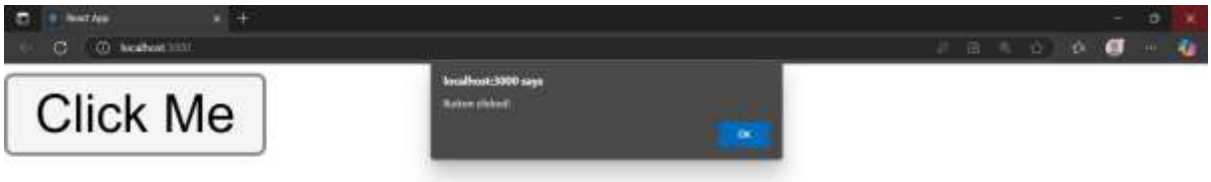
Username – user , password - pass

24. Demonstrate Events in React JS.

change code in src folder ----> app.js

```
import React from 'react';

function Events() {
  const handleClick = () => alert('Button clicked!');
  return <button onClick={handleClick}>Click Me</button>;
}
export default Events;
```

Output

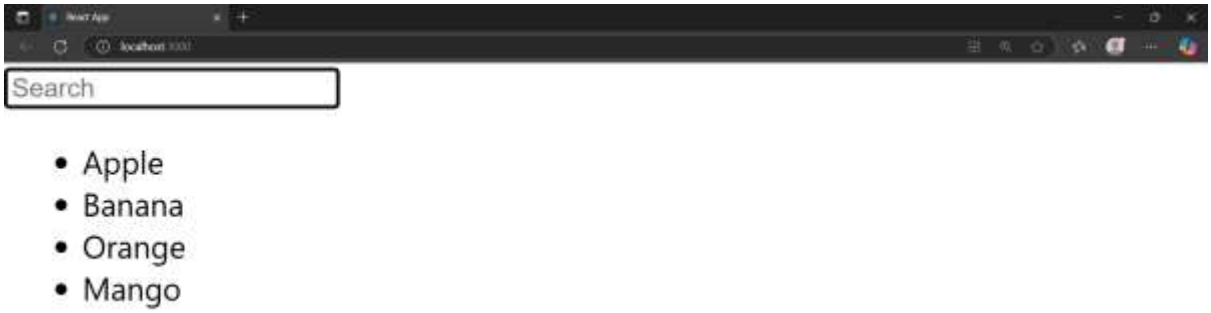


25. Demonstrate Search filter in React JS.

change code in src folder ----> app.js

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

function SearchList() {
  const [search,setSearch] = useState('');
  const items = ['Apple','Banana','Orange','Mango'];
  return (
    <div>
      <input placeholder="Search" onChange={e=>setSearch(e.target.value)} />
      <ul>{items.filter(i=>i.toLowerCase().includes(search.toLowerCase())).map(i=><li key={i}>{i}</li>)}</ul>
    </div>
  );
}

export default SearchList;
```

Output



apple

- Apple

26. Implement a program to create a simple calculator Application using React JS.

change code in src folder ----> app.js

```
import React, { useState } from "react";
const Calculator = () => {
  const [input, setInput] = useState("");
  return (
    <div>
      <input value={input} readOnly />
      {"1234567890+-*/".split("").map(c => <button
onClick={()=>setInput(input+c)}>{c}</button>)}
      <button onClick={()=>setInput(eval(input))}>=</button>
      <button onClick={()=>setInput("")}>C</button>
    </div>
  );
};
export default Calculator;
```

Output


4+5
1
2
3
4
5
6
7
8
9
0
+
-
*
/
=
C


9
1
2
3
4
5
6
7
8
9
0
+
-
*
/
=
C

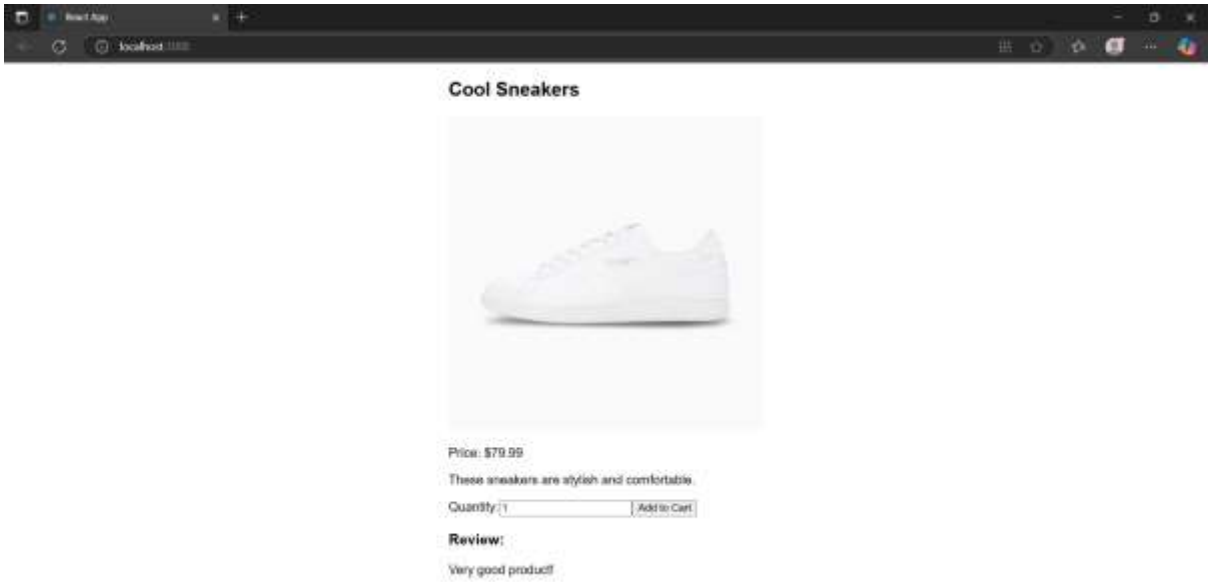
27. Implement a product page that displays detailed product information, images, reviews, and purchase options in React JS.

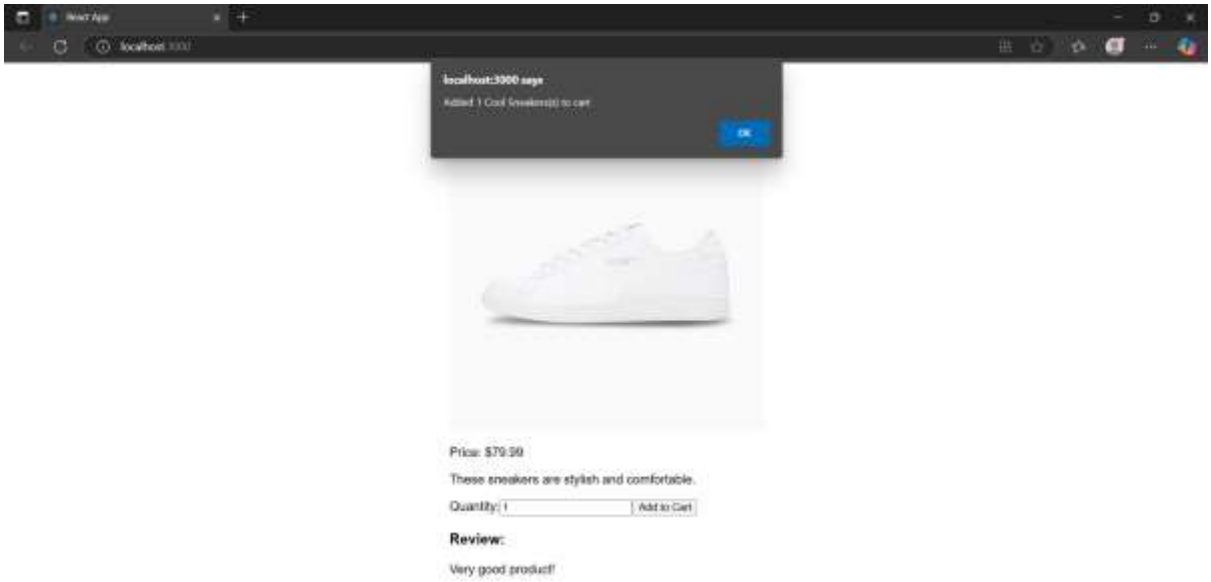
change code in src folder ----> app.js

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

function App() {
  const product = {
    name: 'Cool Sneakers',
    price: 79.99,
    image:
'https://images.puma.com/image/upload/f_auto,q_auto,b_rgb:fafafa,w_750,h_750/global/394371/02/sv01/fnd/IND/fmt/png/Smashic-Comfort-Casual-Sneakers',
    description: 'These sneakers are stylish and comfortable.',
    review: 'Very good product!',
  };

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

  const handleAddToCart = () => {
    alert(`Added ${quantity} ${product.name}(s) to cart`);
  };

  return (
    <div style={{ maxWidth: 400, margin: '20px auto', fontFamily: 'Arial' }}>
      <h2>{product.name}</h2>
      <img src={product.image} alt="Product" style={{ width: '100%' }} />
      <p>Price: ${product.price}</p>
      <p>{product.description}</p>

      <label>
        Quantity:
        <input
```

```
        type="number"
        value={quantity}
        min={1}
        onChange={e => setQuantity(Number(e.target.value))}
       />
     </label>

     <button onClick={handleAddToCart}>Add to Cart</button>

     <h3>Review:</h3>
     <p>{product.review}</p>
    </div>
  );
}

export default App;
```

Output


Cool Sneakers
Price: $79.99
These sneakers are stylish and comfortable.
Quantity: 1
Add to Cart
Review:
Very good product!


localhost:3000 says
Added 1 Cool Sneakers(s) to cart
OK
Price: $79.99
These sneakers are stylish and comfortable.
Quantity: 1
Add to Cart
Review:
Very good product!

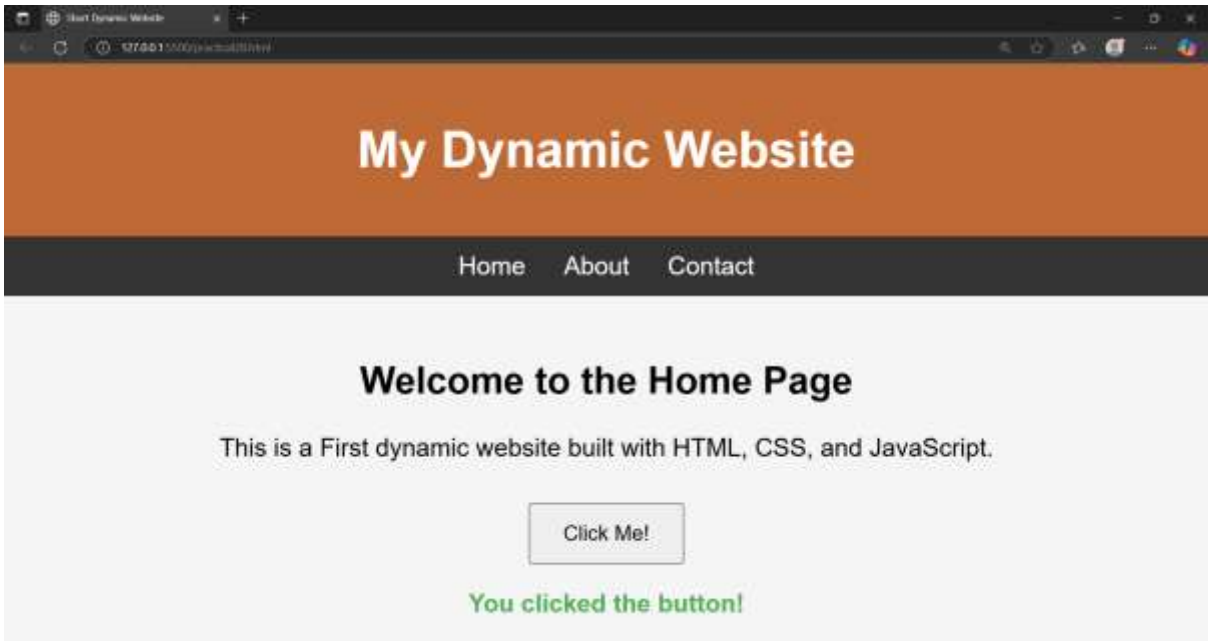
28. Implement a dynamic website demonstrating web technologies (HTML, CSS, JavaScript).

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Short Dynamic Website</title>
<style>
  body { font-family: Arial, sans-serif; margin: 0; padding: 0; background: #f5f5f5; }
  header { background: #bf6934; color: white; padding: 15px; text-align: center; }
  nav { background: #333; padding: 10px; text-align: center; }
  nav a { color: white; margin: 0 10px; text-decoration: none; }
  nav a:hover { text-decoration: underline; }
  main { padding: 20px; text-align: center; }
  button { padding: 10px 20px; margin-top: 10px; cursor: pointer; }
  #dynamicText { color: #4CAF50; font-weight: bold; }
</style>
</head>
<body>

<header>
  <h1>My Short Dynamic Website</h1>
</header>

<nav>
  <a href="#home" onclick="showSection('home')">Home</a>
  <a href="#about" onclick="showSection('about')">About</a>
  <a href="#contact" onclick="showSection('contact')">Contact</a>
</nav>
```

```
<main>
  <section id="home">
    <h2>Welcome to the Home Page</h2>
    <p>This is a short dynamic website built with HTML, CSS, and JavaScript.</p>
    <button onclick="changeText()">Click Me!</button>
    <p id="dynamicText">Hello World!</p>
  </section>

  <section id="about" style="display:none">
    <h2>About Us</h2>
    <p>We create simple, dynamic websites using modern web technologies.</p>
  </section>

  <section id="contact" style="display:none">
    <h2>Contact Us</h2>
    <p>Email: contact@example.com</p>
    <p>Phone: +1234567890</p>
  </section>
</main>

<script>
  function changeText() {
    const text = document.getElementById("dynamicText");
    text.innerText = text.innerText === "Hello World!" ? "You clicked the button!" : "Hello World!";
  }

  function showSection(sectionId) {
    const sections = document.querySelectorAll("main section");
    sections.forEach(sec => sec.style.display = "none");
    document.getElementById(sectionId).style.display = "block";
  }
```

```
</script>


</body>
</html>
```

Output



29. Implement advanced dynamic website using React JS.

change code in src folder ----> app.js

```
import React, { useState } from 'react';
import { BrowserRouter as Router, Routes, Route, Link, useParams } from 'react-router-dom';

// Dummy Products
const products = [
  { id: 1, name: 'Sneakers', price: 79.99, description: 'Comfy and cool sneakers.' },
  { id: 2, name: 'Jacket', price: 99.99, description: 'Warm and stylish jacket.' },
  { id: 3, name: 'Hat', price: 19.99, description: 'Trendy summer hat.' },
];

// Home Page
function Home({ addToCart }) {
  return (
    <div style={{ padding: 20 }}>
      <h2 style={{ color: '#333' }}>🛍️ All Products</h2>
      {products.map(p => (
        <div
          key={p.id}
          style={{
            border: '1px solid #ddd',
            padding: 15,
            marginBottom: 15,
            borderRadius: 8,
            backgroundColor: '#f9f9f9',
          }}
        >
          <h3 style={{ color: '#007bff' }}>{p.name}</h3>
          <p>Price: <strong style={{ color: 'green' }}>${p.price}</strong></p>
```

```
        <Link to={`/product/${p.id}`} style={{ color: '#0066cc' }}>View Details</Link>
      </div>
    ))}
  </div>
 );
}

// Product Detail Page
function ProductPage({ addToCart }) {
 const { id } = useParams();
 const product = products.find(p => p.id === parseInt(id));

 return (
  <div style={{ padding: 20 }}>
   <h2 style={{ color: '#007bff' }}>{product.name}</h2>
   <p>{product.description}</p>
   <p>Price: <strong style={{ color: 'green' }}>${product.price}</strong></p>
   <button
    onClick={() => addToCart(product)}
    style={{
     backgroundColor: '#28a745',
     color: 'white',
     padding: '10px 15px',
     border: 'none',
     borderRadius: 4,
     cursor: 'pointer',
     marginTop: 10,
    }}
   >
    Add to Cart
   </button>
```

```
    </div>
  );
}

// Cart Page
function Cart({ cart }) {
  return (
    <div style={{ padding: 20 }}>
      <h2 style={{ color: '#333' }}>🛒 Shopping Cart</h2>
      {cart.length === 0 ? (
        <p style={{ color: '#888' }}>Your cart is empty.</p>
      ) : (
        <ul style={{ listStyleType: 'none', paddingLeft: 0 }}>
          {cart.map((item, i) => (
            <li
              key={i}
              style={{
                marginBottom: 10,
                backgroundColor: '#f1f1f1',
                padding: 10,
                borderRadius: 4,
              }}
            >
              {item.name} — <strong style={{ color: 'green' }}>${item.price}</strong>
            </li>
          ))}
        </ul>
      )}
    </div>
  );
}
```

```
// Main App Component
function App() {
  const [cart, setCart] = useState([]);

  const addToCart = product => {
    setCart([...cart, product]);
  };

  return (
    <Router>
      <nav style={{
        padding: 10,
        backgroundColor: '#343a40',
        color: 'white',
        display: 'flex',
        justifyContent: 'space-between',
      }}>
        <div>
          <Link to="/" style={{ color: 'white', marginRight: 15, textDecoration: 'none' }}>🏠 Home</Link>
          <Link to="/cart" style={{ color: 'white', textDecoration: 'none' }}>🛒 Cart ({cart.length})</Link>
        </div>
      </nav>

      <Routes>
        <Route path="/" element={<Home addToCart={addToCart} />} />
        <Route path="/product/:id" element={<ProductPage addToCart={addToCart} />} />
        <Route path="/cart" element={<Cart cart={cart} />} />
      </Routes>
    </Router>
  );
```

}

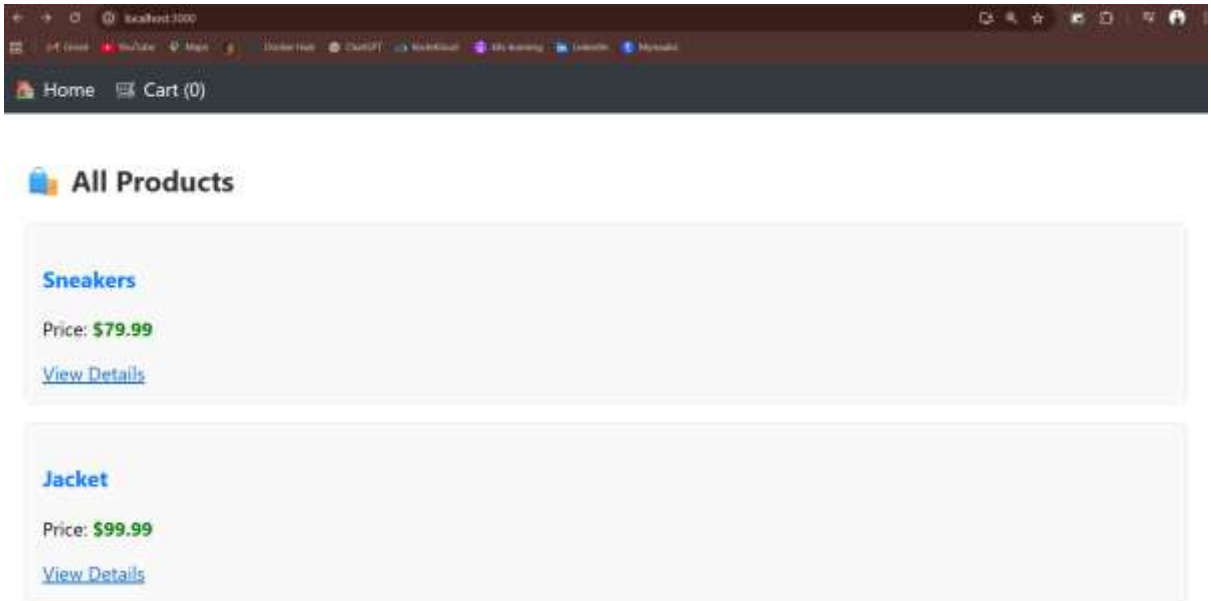
export default App;

Output

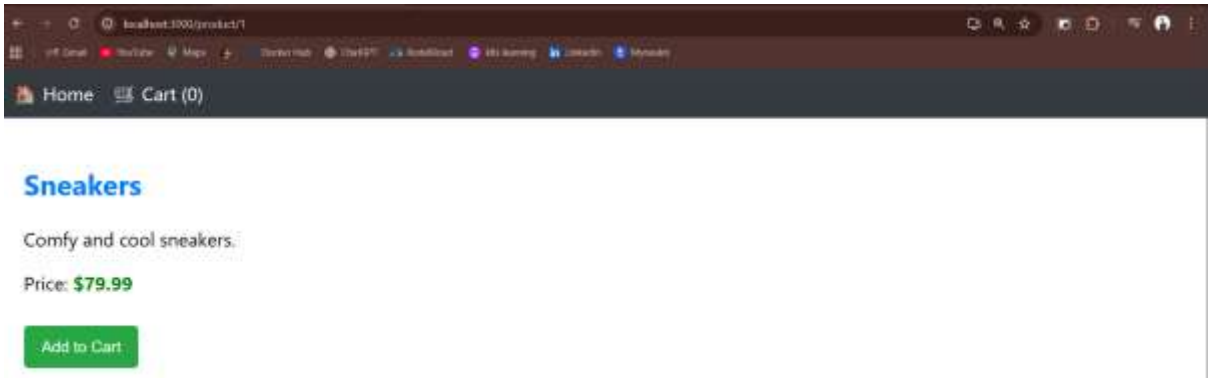
```
PS M:\Demo\my-app> npm start
```

```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.20.224.1:3000
```

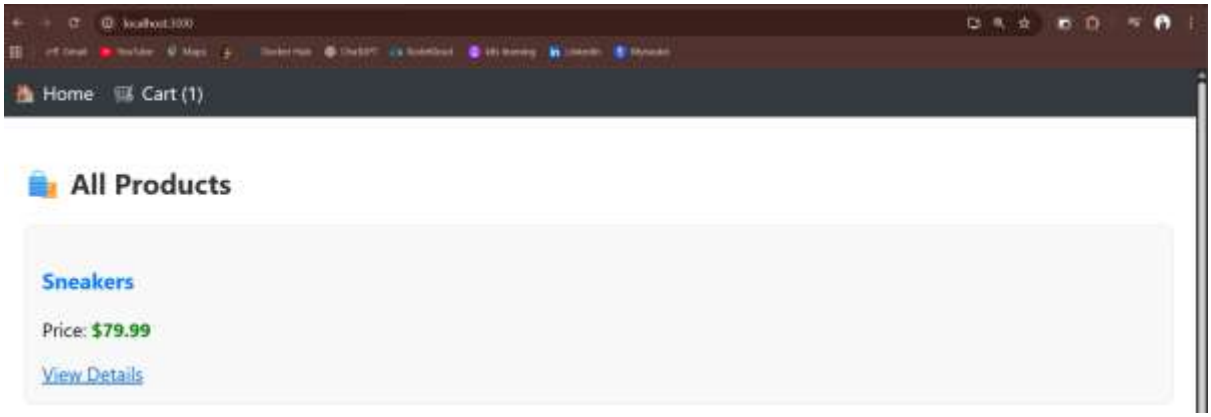
Home page



View details



Carts



30. Create a React application for the Student Management System having registration, login, contact, about pages and implement routing to navigate through these pages.

change code in src folder ----> app.js

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

// Pages
const Home = () => <h2>🏠 Welcome to Student Management System</h2>;

const Registration = () => (
  <div>
    <h2>📝 Registration</h2>
    <input type="text" placeholder="Name" /><br /><br />
    <input type="email" placeholder="Email" /><br /><br />
    <button>Register</button>
  </div>
);

const Login = () => (
  <div>
    <h2>🔐 Login</h2>
    <input type="email" placeholder="Email" /><br /><br />
    <input type="password" placeholder="Password" /><br /><br />
    <button>Login</button>
  </div>
);

const Contact = () => (
  <div>
    <h2>Contact Us</h2>
```

```
    <p>Email: contact@example.com</p>
    <p>Phone: +1234567890</p>
  </div>
);

const About = () => (
  <div>
    <h2>About Us</h2>
    <p>We manage student data efficiently.</p>
  </div>
);

function App() {
  return (
    <Router>
      <div style={{ padding: "10px", fontFamily: "Arial" }}>
        {/* Navigation */}
        <nav style={{ marginBottom: "20px", backgroundColor: "#f0f0f0", padding: "10px" }}>
          <Link to="/" style={{ margin: "0 10px" }}>Home</Link>
          <Link to="/register" style={{ margin: "0 10px" }}>Register</Link>
          <Link to="/login" style={{ margin: "0 10px" }}>Login</Link>
          <Link to="/contact" style={{ margin: "0 10px" }}>Contact</Link>
          <Link to="/about" style={{ margin: "0 10px" }}>About</Link>
        </nav>

        {/* Routing */}
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/register" element={<Registration />} />
          <Route path="/login" element={<Login />} />
          <Route path="/contact" element={<Contact />} />
```

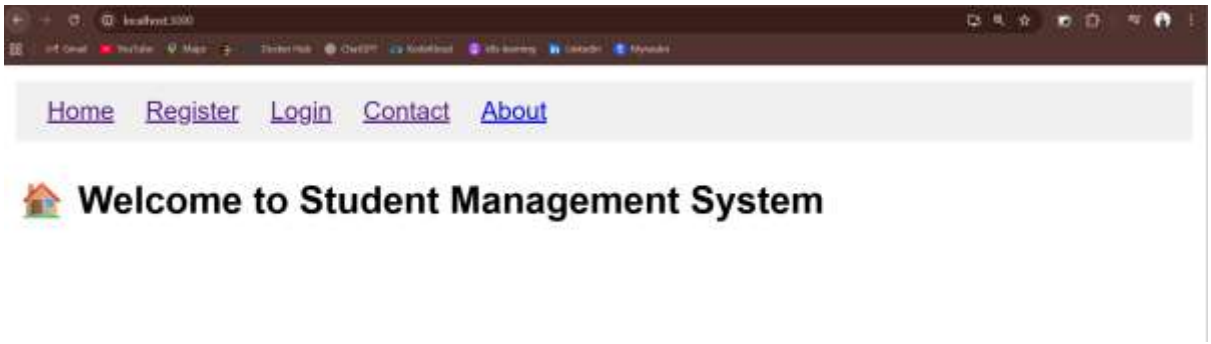
```
        <Route path="/about" element={<About />} />
      </Routes>
    </div>
  </Router>
 );
}

export default App;
```

Output

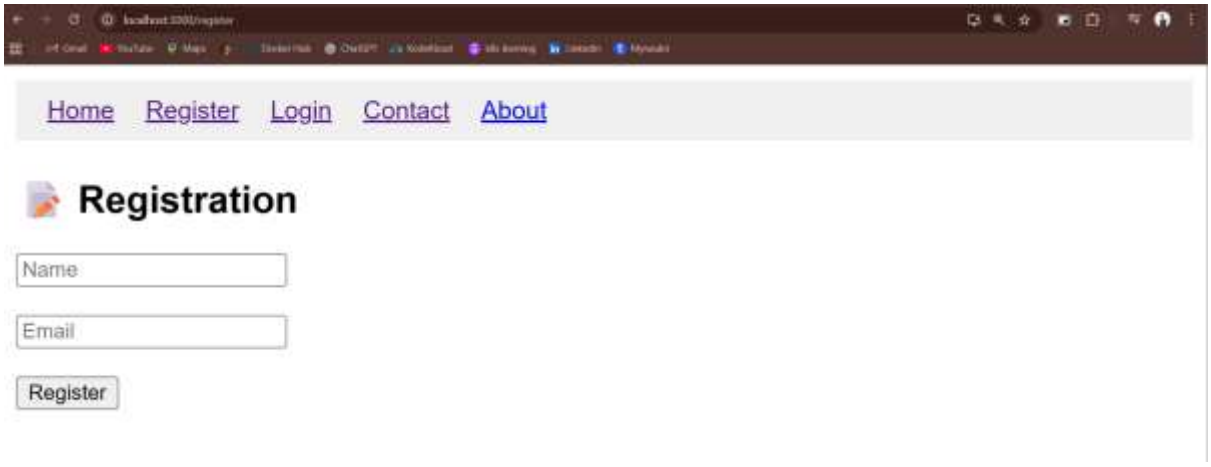
```
PS M:\Demo\my-app> npm start
```

```
Compiled successfully!

You can now view my-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://172.20.224.1:3000
```

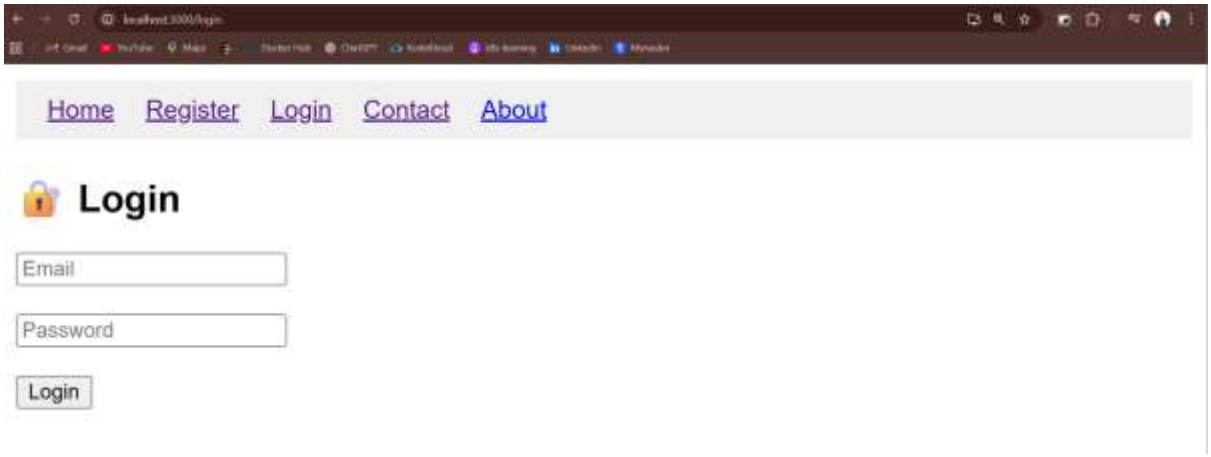
Home page



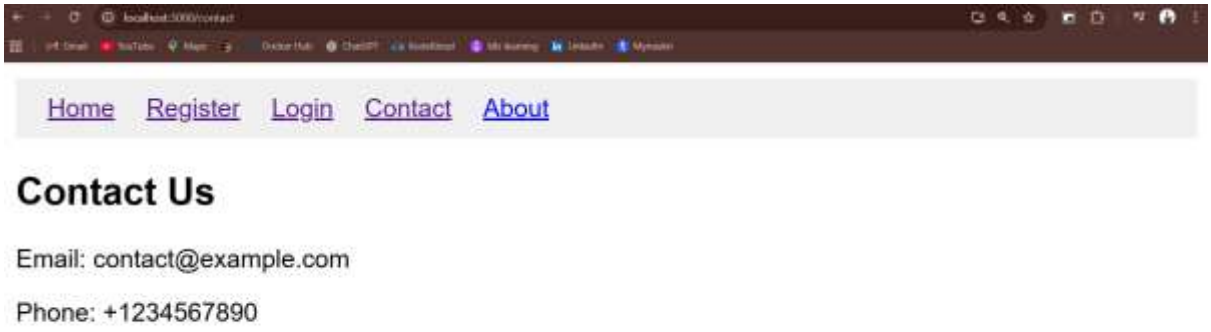
Register page



Login page



Contact page



About page