

AWT LAB PROGRAMS

1. Create a page with Tailwind CSS product cards styled using max-w-sm, shadow-md, rounded-lg, and hover effects like hover:shadow-xl. Style the image with object-cover, rounded-t-lg, h-48 and text with font-bold, text-gray-700, and text-sm.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cards</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100 py-8 px-4">
  <div class="max-w-7xl mx-auto grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3
lg:grid-cols-4 gap-6">

    <div class="max-w-sm rounded-lg shadow-md hover:shadow-xl overflow-hidden
bg-white">
      
      <div class="p-4">
        <h3 class="font-bold text-gray-700 text-sm">Iphone 15</h3>
        <p class="font-bold text-gray-700 mt-2">₹69999</p>
      </div>
    </div>

    <div class="max-w-sm rounded-lg shadow-md hover:shadow-xl overflow-hidden
bg-white">
      
      <div class="p-4">
        <h3 class="font-bold text-gray-700 text-sm">Iphone 16</h3>
        <p class="font-bold text-gray-700 mt-2">₹75000</p>
      </div>
    </div>

    <div class="max-w-sm rounded-lg shadow-md hover:shadow-xl overflow-hidden
bg-white">
      
      <div class="p-4">
        <h3 class="font-bold text-gray-700 text-sm">Iphone 16pro max</h3>
        <p class="font-bold text-gray-700 mt-2">₹120000</p>
      </div>
    </div>
```

AWT LAB PROGRAMS

```
<div class="max-w-sm rounded-lg shadow-md hover:shadow-xl overflow-hidden
bg-white">
  
  <div class="p-4">
    <h3 class="font-bold text-gray-700 text-sm">Galaxy S24 Ultra</h3>
    <p class="font-bold text-gray-700 mt-2">₹105000</p>
  </div>
</div>
</div>
</body>
</html>
```

HOVERING CODE:-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Hover Cards</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100 py-8 px-4">
  <div class="max-w-7xl mx-auto grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3
lg:grid-cols-4 gap-6">

    <div class="relative max-w-sm rounded-lg shadow-md hover:shadow-red-500/50
overflow-hidden bg-white group">
      
      <div class="p-4">
        <h3 class="font-bold text-gray-700 text-sm">Iphone 15</h3>
        <p class="font-bold text-gray-700 mt-2">₹69999</p>
      </div>
      <div class="absolute top-0 left-0 p-2 opacity-0 group-hover:opacity-100
transition-opacity duration-300 bg-black bg-opacity-50 text-white text-xs">
        <p><strong>Top-left Hover:</strong> Rounded edges,Shadow effect</p>
      </div>

      <div class="absolute bottom-0 left-0 p-2 opacity-0 group-hover:opacity-100
transition-opacity duration-300 bg-black bg-opacity-50 text-white text-xs">
        <p><strong>Bottom-left Hover:</strong> Product description</p>
      </div>
```

AWT LAB PROGRAMS

</div>

```
<div class="relative max-w-sm rounded-lg shadow-md hover:shadow-yellow-500/50
overflow-hidden bg-white group">
  
  <div class="p-4">
    <h3 class="font-bold text-gray-700 text-sm">Iphone 16</h3>
    <p class="font-bold text-gray-700 mt-2">₹75000</p>
  </div>
  <div class="absolute top-0 left-0 p-2 opacity-0 group-hover:opacity-100
transition-opacity duration-300 bg-black bg-opacity-50 text-white text-xs">
    <p><strong>Top-left Hover:</strong> Rounded edges,Shadow effect</p>
  </div>

  <div class="absolute bottom-0 left-0 p-2 opacity-0 group-hover:opacity-100
transition-opacity duration-300 bg-black bg-opacity-50 text-white text-xs">
    <p><strong>Bottom-left Hover:</strong> Product description</p>
  </div>
</div>
```

```
<div class="relative max-w-sm rounded-lg shadow-md hover:shadow-green-500/50
overflow-hidden bg-white group">
  
  <div class="p-4">
    <h3 class="font-bold text-gray-700 text-sm">Iphone 16pro max</h3>
    <p class="font-bold text-gray-700 mt-2">₹120000</p>
  </div>
  <div class="absolute top-0 left-0 p-2 opacity-0 group-hover:opacity-100
transition-opacity duration-300 bg-black bg-opacity-50 text-white text-xs">
    <p><strong>Top-left Hover:</strong> Rounded edges</p>
  </div>
  <div class="absolute top-0 right-0 p-2 opacity-0 group-hover:opacity-100
transition-opacity duration-300 bg-black bg-opacity-50 text-white text-xs">
    <p><strong>Top-right Hover:</strong> Shadow effect</p>
  </div>
  <div class="absolute bottom-0 left-0 p-2 opacity-0 group-hover:opacity-100
transition-opacity duration-300 bg-black bg-opacity-50 text-white text-xs">
    <p><strong>Bottom-left Hover:</strong> Product description</p>
  </div>
  <div class="absolute bottom-0 right-0 p-2 opacity-0 group-hover:opacity-100
transition-opacity duration-300 bg-black bg-opacity-50 text-white text-xs">
    <p><strong>Bottom-right Hover:</strong> More details</p>
  </div>
</div>
```

AWT LAB PROGRAMS

```
<div class="relative max-w-sm rounded-lg shadow-md hover:shadow-purple-500/50
overflow-hidden bg-white group">
  
  <div class="p-4">
    <h3 class="font-bold text-gray-700 text-sm">Galaxy S24 Ultra</h3>
    <p class="font-bold text-gray-700 mt-2">₹105000</p>
  </div>
  <div class="absolute top-0 left-0 p-2 opacity-0 group-hover:opacity-100
transition-opacity duration-300 bg-black bg-opacity-50 text-white text-xs">
    <p><strong>Top-left Hover:</strong> Rounded edges,Shadow effect</p>
  </div>

  <div class="absolute bottom-0 left-0 p-2 opacity-0 group-hover:opacity-100
transition-opacity duration-300 bg-black bg-opacity-50 text-white text-xs">
    <p><strong>Bottom-left Hover:</strong> Product description</p>
  </div>

</div>

</div>
</body>
</html>
```

AWT LAB PROGRAMS

2. Create a developer portfolio website using Bootstrap with a responsive navbar, a hero section featuring display-4 text and a btn-primary button, and a projects section using a grid (row and col-md-4) to display three card components. Include a contact form with form-group fields for name, email, and message, and finish with a footer featuring social media icons.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Developer Portfolio</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
  <script src="https://kit.fontawesome.com/a076d05399.js"></script>

  <style>
    /* Custom styles for color changes */
    body {
      background-color: #000; /* Set background color to black */
      color: white; /* Set text color to white */
    }

    .navbar {
      background-color: #000; /* Navbar background color black */
    }

    .navbar-brand {
      color: white; /* Navbar brand text color */
    }

    .navbar-nav .nav-link {
      color: white; /* Navbar links color */
    }

    .navbar-nav .nav-link:hover {
      color: #ccc; /* On hover, navbar links become grey */
    }

    #hero {
      background-color: #eee7e7; /* Hero section background to black */
      color: rgb(12, 12, 12); /* Hero text color to white */
    }
```

AWT LAB PROGRAMS

```
#contact {
  background-color: #eee7e7; /* Hero section background to black */
  color: rgb(12, 12, 12); /* Hero text color to white */
}

.btn-primary {
  background-color: #ccc; /* Primary button color to grey */
  border-color: #ccc; /* Set border to grey */
}

.btn-primary:hover {
  background-color: #fff; /* On hover, button background turns white */
  color: #000; /* Text turns black */
}

.card {
  background-color: #333; /* Dark background for cards */
  border: 1px solid #444; /* Light grey border for cards */
}

.card-title, .card-text {
  color: white; /* Text color inside cards */
}

footer {
  background-color: #000; /* Footer background to black */
  color: white; /* Footer text color */
}
</style>
</head>
<body>

<nav class="navbar navbar-expand-lg navbar-dark">
  <a class="navbar-brand" href="#">Canvil Joyal Lobo</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav ml-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#hero">Home</a>
      </li>
    </ul>
  </div>
</nav>
```

AWT LAB PROGRAMS

```
<li class="nav-item">
  <a class="nav-link" href="#projects">Projects</a>
</li>
<li class="nav-item">
  <a class="nav-link" href="#contact">Contact</a>
</li>
</ul>
</div>
</nav>
```

```
<header id="hero" class="jumbotron text-center">
  <h1 class="display-4">Welcome to My Portfolio!</h1>
  <p class="lead">I am a passionate developer who loves to build amazing web
applications.</p>
  <a class="btn btn-primary btn-lg" href="#projects" role="button">See My Projects</a>
</header>
```

```
<section id="projects" class="container my-5">
  <h2 class="text-center mb-5">My Projects</h2>
  <div class="row">
    <div class="col-md-4">
      <div class="card mb-4 shadow-sm">
        
        <div class="card-body">
          <h5 class="card-title">Smart S-Mart</h5>
          <p class="card-text">Shopping Website</p>
          <a href="#" class="btn btn-primary">View Project</a>
        </div>
      </div>
    </div>
    <div class="col-md-4">
      <div class="card mb-4 shadow-sm">
        
        <div class="card-body">
          <h5 class="card-title">Database Mini Project</h5>
          <p class="card-text">Backend databse project</p>
          <a href="#" class="btn btn-primary">View Project</a>
        </div>
      </div>
    </div>
    <div class="col-md-4">
      <div class="card mb-4 shadow-sm">
        
        <div class="card-body">
```

AWT LAB PROGRAMS

```
<h5 class="card-title">Career Compass</h5>
<p class="card-text">Android Application on Career Compass.</p>
<a href="#" class="btn btn-primary">View Project</a>
</div>
</div>
</div>
</div>
</section>

<section id="contact" class="bg-light py-5">
  <div class="container">
    <h2 class="text-center mb-5">Contact Me</h2>
    <form>
      <div class="form-group">
        <label for="name">Your Name</label>
        <input type="text" class="form-control" id="name" placeholder="Enter your name"
required>
      </div>
      <div class="form-group">
        <label for="email">Email address</label>
        <input type="email" class="form-control" id="email" placeholder="Enter your
email" required>
      </div>
      <div class="form-group">
        <label for="message">Message</label>
        <textarea class="form-control" id="message" rows="4" placeholder="Your
message" required></textarea>
      </div>
      <button type="submit" class="btn btn-primary">Send Message</button>
    </form>
  </div>
</section>

<footer class="text-center py-4">
  <p>&copy; 2025 Canvil Joyal Lobo | All Rights Reserved</p>
  <div>
    <a href="#" class="text-white mx-2"><i class="fab fa-facebook"></i></a>
    <a href="#" class="text-white mx-2"><i class="fab fa-twitter"></i></a>
    <a href="#" class="text-white mx-2"><i class="fab fa-linkedin"></i></a>
    <a href="#" class="text-white mx-2"><i class="fab fa-github"></i></a>
  </div>
</footer>
```


AWT LAB PROGRAMS

```
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"
rel="stylesheet">
```

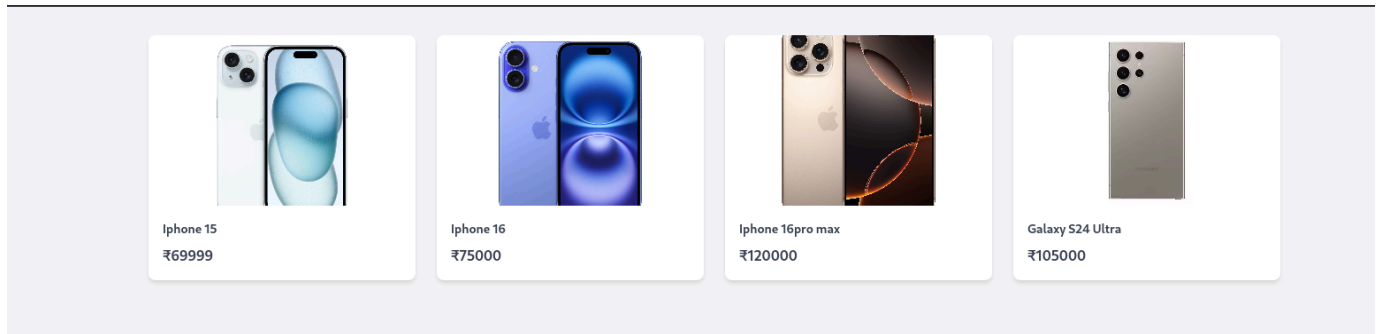
```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"></script>
>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

```
</body>
```

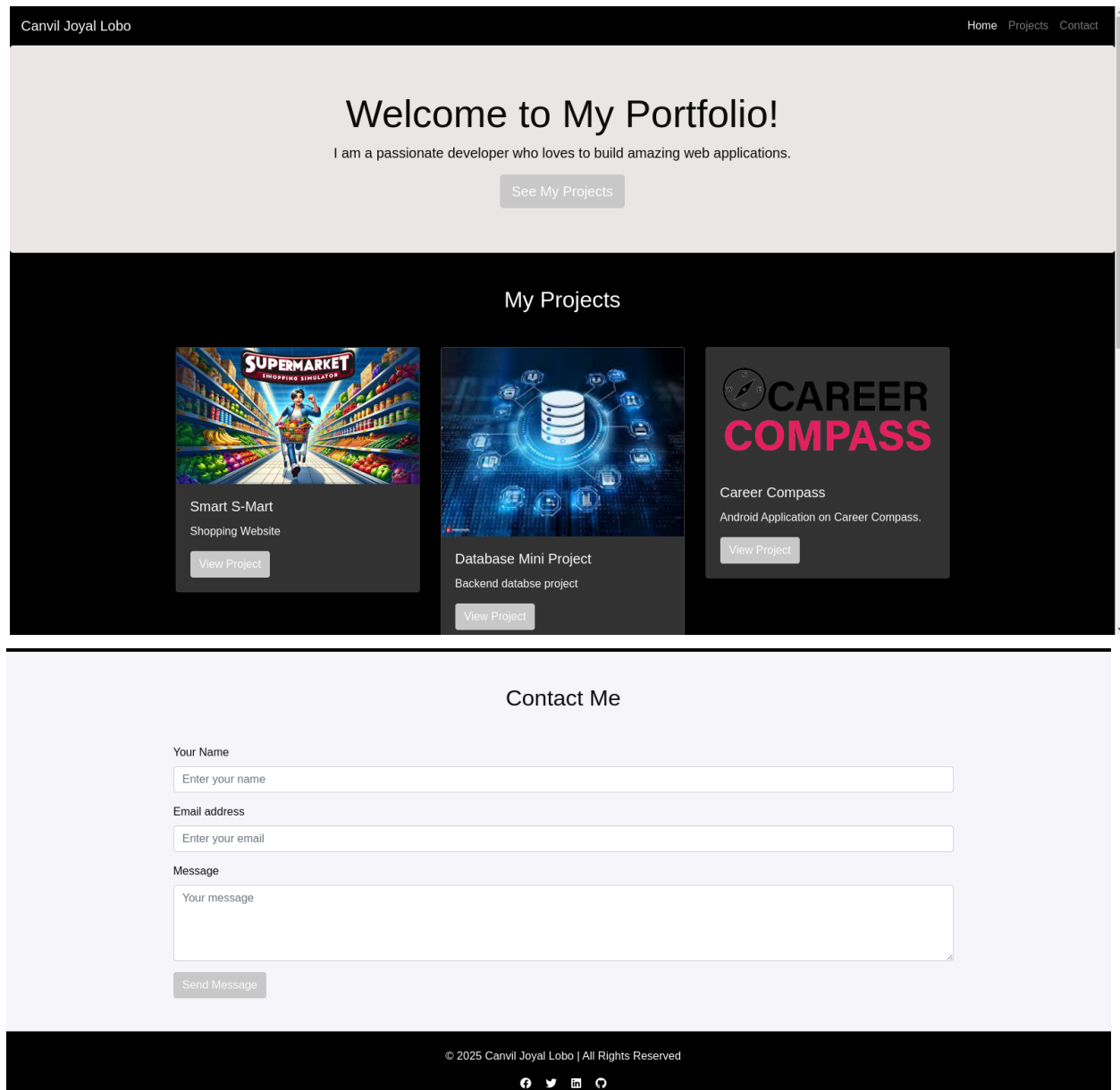
```
</html>
```

AWT LAB PROGRAMS

OUTPUT PRG1:-



OUTPUT PRG2:-



AWT LAB PROGRAMS

3. Create a React application that includes a form with a textbox component, a dropdown component, and a submit button component. Each form element should be implemented as a separate, reusable component.

[APP.js]

```
import React, { useState } from 'react';
import { Container, Typography, Box } from '@mui/material';
import Textbox from './Textbox';
import Dropdown from './Dropdown';
import SubmitButton from './SubmitButton';

const App = () => {
  // Form state
  const [name, setName] = useState("");
  const [selectedOption, setSelectedOption] = useState("");
  const [formData, setFormData] = useState(null);

  const options = ['MCA', 'MBA', 'MCOM'];

  // Handle textbox change
  const handleNameChange = (event) => {
    setName(event.target.value);
  };

  // Handle dropdown change
  const handleDropdownChange = (event) => {
    setSelectedOption(event.target.value);
  };

  // Handle form submit
  const handleSubmit = () => {
    setFormData({ name, selectedOption });
    alert(`Form Submitted! Name: ${name}, Selected Option: ${selectedOption}`);
  };

  return (
    <Container maxWidth="sm" sx={{ textAlign: 'center', marginTop: '50px' }}>
      <Typography variant="h4" gutterBottom>
        React Form Example
      </Typography>

      <Textbox label="Enter Name" value={name} onChange={handleNameChange} />
```

AWT LAB PROGRAMS

```
<Dropdown label="Select Option" value={selectedOption} options={options}
onChange={handleDropdownChange} />
<SubmitButton onClick={handleSubmit} />

{formData && (
  <Box sx={{ marginTop: '20px' }}>
    <Typography variant="h6">Submitted Data:</Typography>
    <Typography>Name: {formData.name}</Typography>
    <Typography>Selected Option: {formData.selectedOption}</Typography>
  </Box>
)}
</Container>
);
};

export default App;
```

[SubmitButton.js]

```
import React from 'react';
import { Button } from '@mui/material';

const SubmitButton = ({ onClick }) => {
  return (
    <Button
      variant="contained"
      color="primary"
      fullWidth
      onClick={onClick}
    >
      Submit
    </Button>
  );
};

export default SubmitButton;
```

[Dropdown.js]

```
import React from 'react';
import { MenuItem, Select, FormControl, InputLabel } from '@mui/material';
```

AWT LAB PROGRAMS

```
const Dropdown = ({ label, value, options, onChange }) => {
  return (
    <FormControl fullWidth sx={{ marginBottom: '20px' }}>
      <InputLabel>{label}</InputLabel>
      <Select
        value={value}
        onChange={onChange}
        label={label}
      >
        {options.map((option, index) => (
          <MenuItem key={index} value={option}>
            {option}
          </MenuItem>
        ))}
      </Select>
    </FormControl>
  );
};

export default Dropdown;
```

[Textbox.js]

```
import React from 'react';
import { TextField } from '@mui/material';

const Textbox = ({ label, value, onChange }) => {
  return (
    <TextField
      label={label}
      variant="outlined"
      fullWidth
      value={value}
      onChange={onChange}
      sx={{ marginBottom: '20px' }}
    />
  );
};

export default Textbox;
```

AWT LAB PROGRAMS

4. Build a React application that includes a form to accept a first name and a last name using Material UI components. Instead of a standard “Submit” button, implement a “Greet Me” button that, when clicked, will display an alert with the message “Hello [first name] [last name]!”.

```
import React, { useState } from 'react';
import { TextField, Button, Container, Typography } from '@mui/material';
```

```
function App() {
  const [firstName, setFirstName] = useState("");
  const [lastName, setLastName] = useState("");
```

```
  const handleFirstNameChange = (event) => {
    setFirstName(event.target.value);
  };

```

```
  const handleLastNameChange = (event) => {
    setLastName(event.target.value);
  };

```

```
  const handleGreetMeClick = () => {
    alert(' Hello ${firstName} ${lastName}! ');
  };

```

```
  return (
    <>
      {}
      <div
        style={{
          backgroundColor: 'black',
          height: '100vh',
          margin: 0,
          display: 'flex',
          justifyContent: 'center',
          alignItems: 'center',
        }}
      >
        {}
        <Container
          maxWidth="sm"
          sx={{
            backgroundColor: 'orange',
            padding: '20px',

```

AWT LAB PROGRAMS

```
        borderRadius: '10px',
      }}
    >
    <Typography variant="h4" gutterBottom>
      Greet Me Application
    </Typography>

    {}
    <TextField
      label="First Name"
      variant="outlined"
      fullWidth
      value={firstName}
      onChange={handleFirstNameChange}
      sx={{
        marginBottom: '20px',
      }}
    />

    {}
    <TextField
      label="Last Name"
      variant="outlined"
      fullWidth
      value={lastName}
      onChange={handleLastNameChange}
      sx={{
        marginBottom: '20px',
      }}
    />

    {}
    <Button
      variant="contained"
      onClick={handleGreetMeClick}
      fullWidth
      sx={{
        backgroundColor: 'black',
        color: 'white',
      }}
    >
      Greet Me
    </Button>
  </Container>
```

AWT LAB PROGRAMS

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

```
export default App;
```

//with error message with empty fields

```
import React, { useState } from 'react';
import { TextField, Button, Container, Typography, Alert } from '@mui/material';
```

```
function App() {
  const [firstName, setFirstName] = useState("");
  const [lastName, setLastName] = useState("");
  const [error, setError] = useState(""); // error message
```

```
  const handleFirstNameChange = (event) => {
    setFirstName(event.target.value);
  };

```

```
  const handleLastNameChange = (event) => {
    setLastName(event.target.value);
  };

```

```
  const handleGreetMeClick = () => {
    if (!firstName || !lastName) {
      setError('Please enter both first and last names');
    } else {
      setError(""); // Clear the error if both fields are filled
      alert(`Hello ${firstName} ${lastName}!`);
    }
  };

```

```
  return (
    <>
      <div
        style={{
          backgroundColor: 'black',
          height: '100vh',
          margin: 0,
          display: 'flex',
          justifyContent: 'center',
          alignItems: 'center',
```


AWT LAB PROGRAMS

```
    }}
  >
  <Container
    maxWidth="sm"
    sx={{
      backgroundColor: 'orange',
      padding: '20px',
      borderRadius: '10px',
    }}
  >
  <Typography variant="h4" gutterBottom>
    Greet Me Application
  </Typography>

  {/* Display error message if there's an error */}
  {error && (
    <Alert severity="error" sx={{ marginBottom: '20px' }}>
      {error}
    </Alert>
  )}

  <TextField
    label="First Name"
    variant="outlined"
    fullWidth
    value={firstName}
    onChange={handleFirstNameChange}
    sx={{
      marginBottom: '20px',
    }}
  />

  <TextField
    label="Last Name"
    variant="outlined"
    fullWidth
    value={lastName}
    onChange={handleLastNameChange}
    sx={{
      marginBottom: '20px',
    }}
  />

  <Button
```

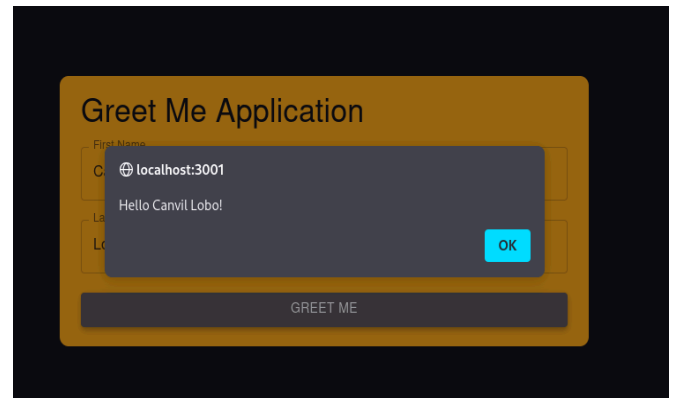
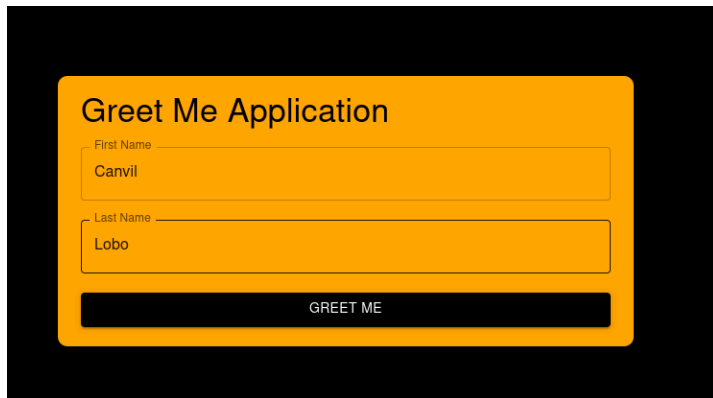
AWT LAB PROGRAMS

```
        variant="contained"
        onClick={handleGreetMeClick}
        fullWidth
        sx={{
            backgroundColor: 'black',
            color: 'white',
        }}
    >
        Greet Me
    </Button>
</Container>
</div>
</>
);
}

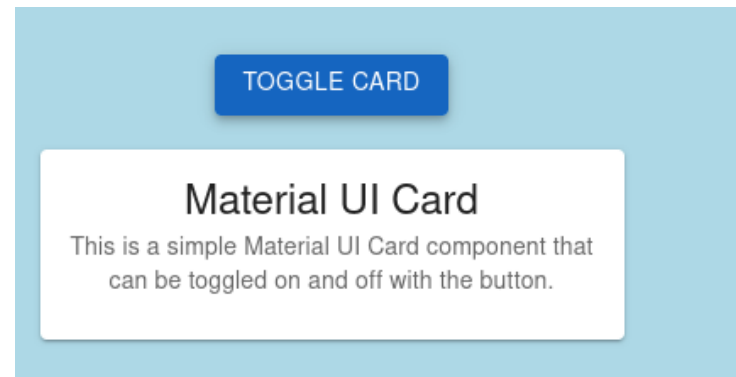
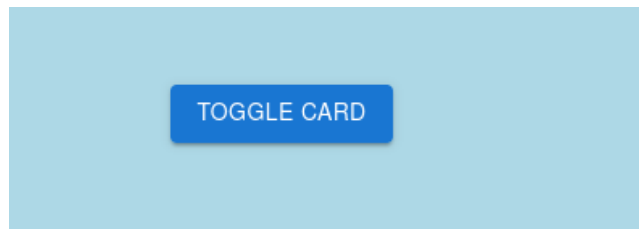
export default App;
```

AWT LAB PROGRAMS

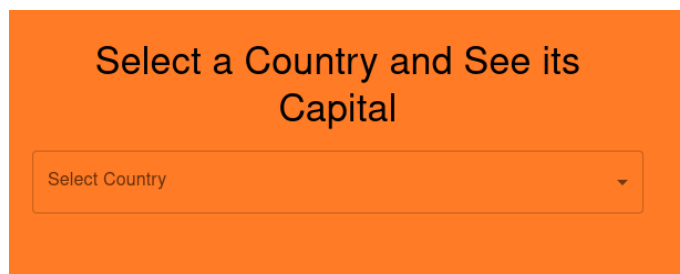
OUTPUT PRG 4:-



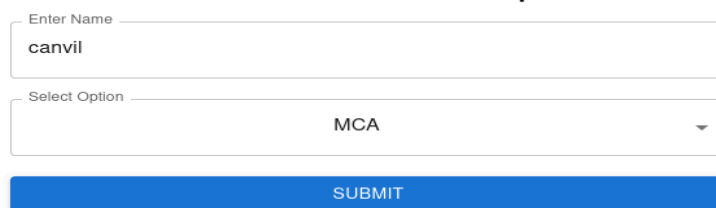
OUTPUT PRG5:-



OUTPUT PART-B 1:-



React Form Example



Submitted Data:
Name: canvil
Selected Option: MCA

AWT LAB PROGRAMS

5. Build a simple React application that includes a Material UI button. When the button is clicked, it should toggle the visibility of a Material UI Card component

[App.js]

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

```
import { Button, Card, CardContent, Typography } from '@mui/material';
```

```
function ToggleCard() {
```

```
  const [showCard, setShowCard] = useState(false); // State to toggle card visibility
```

```
  const handleToggleCard = () => {
```

```
    setShowCard(!showCard); // Toggle the value of showCard on button click
```

```
  };
```

```
  return (
```

```
    <div
```

```
      style={{
```

```
        display: 'flex',
```

```
        justifyContent: 'center',
```

```
        alignItems: 'center',
```

```
        height: '100vh',
```

```
        backgroundColor: 'lightblue',
```

```
      }}  
    >
```

```
    <div style={{ textAlign: 'center' }}>
```

```
      <Button
```

```
        variant="contained"
```

```
        color="primary"
```

```
        onClick={handleToggleCard}
```

```
        sx={{ marginBottom: '20px' }}  
      >
```

```
        Toggle Card
```

```
      </Button>
```

```
    { /* Show the card if showCard is true */}
```

```
    {showCard && (
```

```
      <Card sx={{ maxWidth: 345, margin: '0 auto' }}>
```

```
        <CardContent>
```

```
          <Typography variant="h5" component="div">
```

```
            Material UI Card
```

```
          </Typography>
```

```
          <Typography variant="body2" color="text.secondary">
```

This is a simple Material UI Card component that can be toggled on and off with the button.

AWT LAB PROGRAMS

```
        </Typography>
      </CardContent>
    </Card>
  )}
</div>
</div>
);
}

export default ToggleCard;
```

[PART-B]

1. Develop a React application that dynamically displays the capital city of a selected country. Implement the country selection dropdown as a separate, reusable component. The main application component should integrate this dropdown component and, upon selection of a country, display the corresponding capital city.

[App.js]

```
import React, { useState } from 'react';
import { Container, Typography, Box } from '@mui/material';
import CountryDropdown from './CountryDropdown';
```

```
const App = () => {
```

```
  const countries = [
    { name: 'USA', capital: 'Washington, D.C.' },
    { name: 'Germany', capital: 'Berlin' },
    { name: 'France', capital: 'Paris' },
    { name: 'Japan', capital: 'Tokyo' },
    { name: 'India', capital: 'New Delhi' },
  ];
```

```
  const [selectedCountry, setSelectedCountry] = useState("");
  const [capital, setCapital] = useState("");
```

```
  const handleCountryChange = (event) => {
    const selectedCountry = event.target.value;
    setSelectedCountry(selectedCountry);
```

```

    const country = countries.find((country) => country.name === selectedCountry);
    if (country) {
      setCapital(country.capital);
    } else {
      setCapital("");
    }
  };
};
```

```
return (
  <div
    style={{
      backgroundColor: '#ff7b25',
      minHeight: '100vh',
      padding: '20px',
    }}
  >
```

AWT LAB PROGRAMS

```
>
<Container maxWidth="sm" sx={{ textAlign: 'center', marginTop: '50px' }}>
  <Typography variant="h4" gutterBottom>
    Select a Country and See its Capital
  </Typography>

  <CountryDropdown
    countries={countries}
    selectedCountry={selectedCountry}
    onCountryChange={handleCountryChange}
  />

  {capital && (
    <Box sx={{ marginTop: '20px' }}>
      <Typography variant="h6">Capital: {capital}</Typography>
    </Box>
  )}
</Container>
</div>
);
};
```

export default App;

[Countrydropdown.js]

```
import React from 'react';
import { MenuItem, Select, FormControl, InputLabel } from '@mui/material';

const CountryDropdown = ({ countries, selectedCountry, onCountryChange }) => {
  return (
    <FormControl fullWidth sx={{ marginBottom: '20px' }}>
      <InputLabel>Select Country</InputLabel>
      <Select
        value={selectedCountry}
        onChange={onCountryChange}
        label="Select Country"
      >
        {countries.map((country, index) => (
          <MenuItem key={index} value={country.name}>
            {country.name}
          </MenuItem>
        ))}
      </Select>
    </FormControl>
  )
}
```

AWT LAB PROGRAMS

```
        </FormControl>
    );
};

export default CountryDropdown;
```


[PART-B]

3. Create a custom component for rendering each joke present in an array. Using the map function, map through each object in the array. Use the custom component to render each object.

[// src/App.js]

```
import React, { useState } from 'react';
import Joke from './components/joke';
const App = () => {
  // Data for jokes by each day
  const jokesByDay = {
    Monday: { setup: 'Why don't skeletons fight each other?', punchline: 'They don't have the guts.' },
    Tuesday: { setup: 'What do you call fake spaghetti?', punchline: 'An impasta!' },
    Wednesday: { setup: 'Why did the scarecrow win an award?', punchline: 'Because he was outstanding in his field!' },
    Thursday: { setup: 'Why don't eggs tell jokes?', punchline: 'They'd crack each other up!' },
    Friday: { setup: 'How do you organize a space party?', punchline: 'You planet!' },
    Saturday: { setup: 'Why can't your nose be 12 inches long?', punchline: 'Because then it would be a foot!' },
    Sunday: { setup: 'What did one ocean say to the other ocean?', punchline: 'Nothing, they just waved.' }
  };
```

```
const [selectedDay, setSelectedDay] = useState('Monday');
```

```
const handleDayChange = (event) => {
  setSelectedDay(event.target.value);
};
```

```
return (
  <div>
    <h1>Jokes of the Week</h1>

    {}
    <select onChange={handleDayChange} value={selectedDay}>
      {Object.keys(jokesByDay).map((day) => (
        <option key={day} value={day}>
          {day}
        </option>
      ))}
    </select>
  </div>
);
```

AWT LAB PROGRAMS

```
    </select>

    {}
    <div className="joke-container">
      <h2>{selectedDay}</h2>
      <Joke
        setup={jokesByDay[selectedDay].setup}
        punchline={jokesByDay[selectedDay].punchline}
      />
    </div>
  </div>
);
};

export default App;
```

[//react/b3/src/components/joke.js]

```
import React from 'react';

// Custom component to render each joke
const Joke = ({ setup, punchline }) => {
  return (
    <div className="joke">
      <p><strong>{setup}</strong></p>
      <p>{punchline}</p>
    </div>
  );
};

export default Joke;
```

[react/b3/src/index.js]

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
reportWebVitals();
```

AWT LAB PROGRAMS

```
</React.StrictMode>
);
```

```
// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

```
[/* src/index.css */]
```

```
body {
  font-family: Arial, sans-serif;
  background-color: burlywood;
}
```

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

```
.joke-container {
  background-color: #f9f9f9;
  padding: 20px;
  margin: 15px auto;
  border: 1px solid #ddd;
  border-radius: 8px;
  width: 80%;
  max-width: 600px;
}
```

```
select {
  display: block;
  margin: 10px auto;
  padding: 10px;
  font-size: 16px;
  border-radius: 5px;
  border: 1px solid #ddd;
}
```

```
h2 {
  text-align: center;
  margin-bottom: 10px;
```

AWT LAB PROGRAMS

```
}

.joke {
  background-color: #fff;
  padding: 10px;
  border-radius: 5px;
}

.joke p {
  margin: 5px 0;
}
```

[// automatic time rendering per day//]

[//App.js]

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

```
// Array of jokes with specific dates
```

```
const jokesArray = [
  { date: '2025-01-24', joke: 'Why don't skeletons fight each other? They don't have the guts.' },
  { date: '2025-01-25', joke: 'What do you call fake spaghetti? An impasta!' },
  { date: '2025-01-26', joke: 'Why did the scarecrow win an award? Because he was outstanding in his field.' },
  { date: '2025-01-27', joke: 'I told my wife she was drawing her eyebrows too high. She looked surprised.' },
  { date: '2025-01-28', joke: 'Parallel lines have so much in common. It's a shame they'll never meet.' },
  // Add more jokes for different dates...
];
```

```
function App() {
  const [jokeOfTheDay, setJokeOfTheDay] = useState(null);
  const [currentDateTime, setCurrentDateTime] = useState("");

  useEffect(() => {
    // Function to update the current date, time, and joke
    const updateDateTimeAndJoke = () => {
      const today = new Date().toISOString().split('T')[0]; // Get today's date in YYYY-MM-DD format
      const currentDateTime = new Date().toLocaleString(); // Format: "MM/DD/YYYY, HH:MM:SS AM/PM"
      setCurrentDateTime(currentDateTime);
    }
  });
}
```

AWT LAB PROGRAMS

```
// Find the joke for today's date
const foundJoke = jokesArray.find(joke => joke.date === today);
if (foundJoke) {
  setJokeOfTheDay(foundJoke.joke);
} else {
  setJokeOfTheDay("No joke available for today!");
}
};

// Update immediately on mount
updateDateTimeAndJoke();

// Set an interval to update every 1 second (1000 milliseconds)
const intervalId = setInterval(updateDateTimeAndJoke, 1000);
// Cleanup the interval when the component is unmounted
return () => clearInterval(intervalId);
}, []); // Empty array means this effect runs once when the component is mounted
return (
  <div className="App">
    <h1>Joke of the Day</h1>
    {jokeOfTheDay ? (
      <div>
        <Joke joke={jokeOfTheDay} />
        <p><strong>Current Date and Time:</strong> {currentDateTime}</p>
      </div>
    ) : (
      <p>Loading...</p>
    )}
  </div>
);
}
export default App;

[//Joke.js]
import React from 'react';
// Joke component to render the joke
function Joke({ joke }) {
  return (
    <div className="joke">
      <p>{joke}</p>
    </div>
  );
}
export default Joke;
```

AWT LAB PROGRAMS

OUTPUT PARTB-3:-

Jokes of the Week

Tuesday ▾

Tuesday

What do you call fake spaghetti?
An impasta!

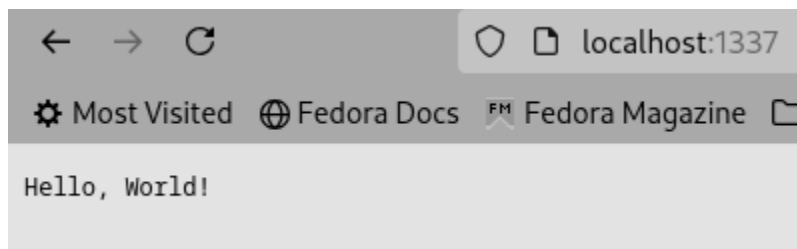
Jokes of the Week

Monday ▾

Monday

Why don't skeletons fight each other?
They don't have the guts.

OUTPUT PARTB-5:-



Joke of the Day

Why don't skeletons fight each other? They don't have the guts.

Current Date and Time: 24/1/2025, 12:26:22 pm

AWT LAB PROGRAMS

[PART-B]

5. Create an HTTP server listening on port 1337, which sends Hello, World! to the browser and using Express.

[// App.js (React code)]

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

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

  useEffect(() => {
    // Fetch data from the backend server
    fetch('http://localhost:1337')
      .then(response => response.text())
      .then(data => setMessage(data))
      .catch(error => console.error('Error fetching data:', error));
  }, []);

  return (
    <div className="App">
      <h1>{message}</h1>
    </div>
  );
}

export default App;
```

[server.js]

[//using express]

```
const express = require('express'); // Import the Express module
const app = express();              // Create an Express application

const PORT = 1337;                  // Define the port to listen on

// Set up the route for the root path
app.get('/', (req, res) => {
  res.send('Hello, World!');        // Send 'Hello, World!' as the response
});

// Start the server and listen on port 1337
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

AWT LAB PROGRAMS

[//without using express]

[// server.js]

```
const http = require('http');
// Create the server
const server = http.createServer((req, res) => {
  // Add CORS headers to allow requests from localhost:3000
  res.setHeader('Access-Control-Allow-Origin', 'http://localhost:3000'); // Allow requests
from React app
  res.setHeader('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE'); // Allow
common HTTP methods
  res.setHeader('Access-Control-Allow-Headers', 'Content-Type'); // Allow custom
headers
  // Set the response content type
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  // Send the response
  res.end('Hello, World!\n');
});
// Start the server on port 1337
server.listen(1337, () => {
  console.log('Server is running on http://localhost:1337');
});
```


AWT LAB PROGRAMS

2. Develop a react application and:

- i) Build your own Button component and render it three times. On user click, it should alert which button was clicked. (React JS).
- ii) Use the useState React hook to track how many times a button is clicked, and display the number

//Button.js

```
import React, { useState } from 'react';
import './App.css'; // Import the CSS

const Button = ({ buttonId }) => {
  // Use useState to keep track of the number of times the button is clicked
  const [clickCount, setClickCount] = useState(0);

  // Function to handle the click
  const handleClick = () => {
    setClickCount(clickCount + 1);
    alert(`Button ${buttonId} was clicked!`);
  };

  return (
    <div>
      <button onClick={handleClick}>
        Button {buttonId} (Clicked {clickCount} times)
      </button>
    </div>
  );
};

export default Button;
```

//App.js

```
import React from 'react';
import Button from './Button';
import './App.css'; // Importing the CSS file

function App() {
  return (
    <div className="App">
      <h1>Button Click Tracker</h1>
      <Button buttonId={1} />
      <Button buttonId={2} />
      <Button buttonId={3} />
    </div>
  );
}
```

AWT LAB PROGRAMS

```
);  
}
```

```
export default App;
```

```
/* App.css */
```

```
/* Set the background color and center everything */
```

```
body {  
  font-family: Arial, sans-serif;  
  background-color: #f0f0f0;  
  margin: 0;  
  padding: 0;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
}
```

```
/* Center the content and give some margin */
```

```
.App {  
  text-align: center;  
  padding: 20px;  
  background-color: #fff;  
  border-radius: 10px;  
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);  
  width: 300px;  
}
```

```
/* Heading styling */
```

```
h1 {  
  font-size: 24px;  
  margin-bottom: 20px;  
  color: #333;  
}
```

```
/* Styling the button container */
```

```
button {  
  background-color: #4CAF50;  
  color: white;  
  border: none;  
  padding: 10px 20px;  
  margin: 10px;  
  font-size: 16px;  
  cursor: pointer;
```

AWT LAB PROGRAMS

```
border-radius: 5px;  
transition: background-color 0.3s, transform 0.2s ease;  
width: 80%;  
}
```

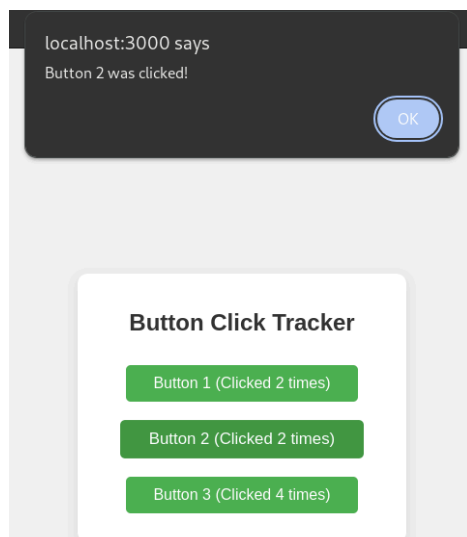
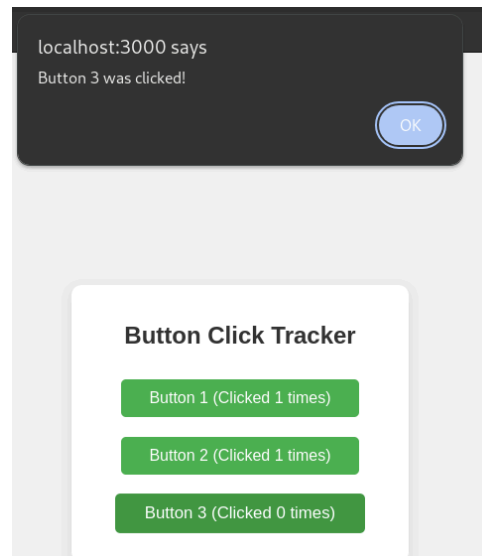
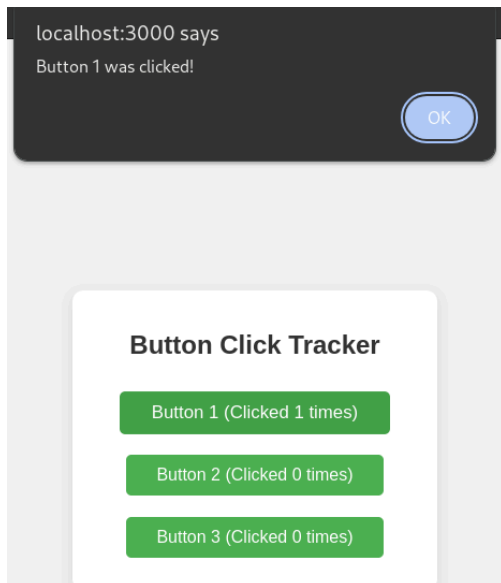
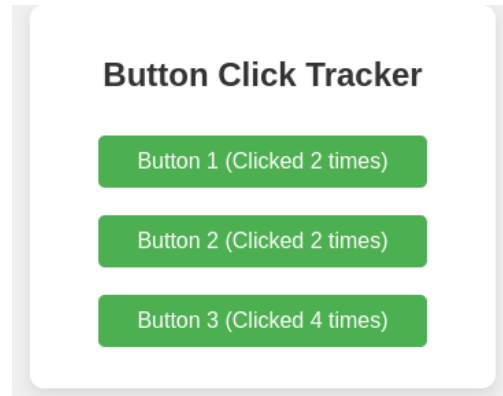
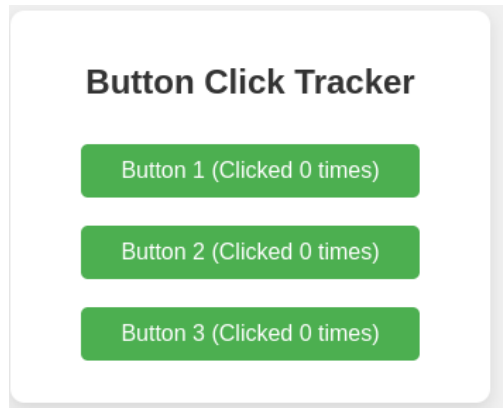
```
/* Hover effect on buttons */  
button:hover {  
    background-color: #45a049;  
    transform: scale(1.05);  
}
```

```
/* Styling the button when clicked */  
button:active {  
    background-color: #388e3c;  
}
```

```
/* Optional: Add some space around the buttons */  
button + button {  
    margin-top: 15px;  
}
```

AWT LAB PROGRAMS

OUTPUT PARTB2:-



AWT LAB PROGRAMS

4. create a multi page React application with a navigation bar component and routes using react-router-dom.

// src/App.js

```
import React from 'react';
import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';
import Navbar from './components/Navbar';
import Home from './pages/Home';
import About from './pages/About';
import Contact from './pages/Contact';
import './App.css'; // Import CSS
```

```
function App() {
  return (
    <Router>
      <div className="App">
        <Navbar />
        <div className="content">
          <Routes>
            <Route path="/" element={<Home />} />
            <Route path="/about" element={<About />} />
            <Route path="/contact" element={<Contact />} />
          </Routes>
        </div>
      </div>
    </Router>
  );
}
```

```
export default App;
```

// src/components/Navbar.js

```
import React from 'react';
import { Link } from 'react-router-dom';
import './Navbar.css'; // Import specific CSS for Navbar
```

```
const Navbar = () => {
  return (
    <nav className="navbar">
      <div className="navbar-container">
        <h1 className="navbar-title">My React App</h1>
        <ul className="navbar-links">
          <li><Link to="/">Home</Link></li>
          <li><Link to="/about">About</Link></li>
        </ul>
      </div>
    </nav>
  );
}
```

AWT LAB PROGRAMS

```
        <li><Link to="/contact">Contact</Link></li>
    </ul>
</div>
</nav>
);
};
```

export default Navbar;

// src/pages/Home.js

import React from 'react';

```
const Home = () => {
  return (
    <div className="page">
      <h2>Welcome to the Home Page</h2>
      <p>This is the homepage of the React app.</p>
    </div>
  );
};
```

export default Home;

// src/pages/About.js

import React from 'react';

```
const About = () => {
  return (
    <div className="page">
      <h2>About Us</h2>
      <p>This page tells you more about us.</p>
    </div>
  );
};
```

export default About;

// src/pages/Contact.js

import React from 'react';

```
const Contact = () => {
  return (
    <div className="page">
      <h2>Contact Us</h2>
```

AWT LAB PROGRAMS

```
        <p>Get in touch with us via this page.</p>
    </div>
);
};
```

```
export default Contact;
```

```
/* App.css */
```

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f4;
}
```

```
.App {
  text-align: center;
}
```

```
.content {
  margin-top: 20px;
  padding: 20px;
}
```

```
/* Basic styling for the page content */
.page {
  background-color: white;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
```

```
h2 {
  color: #333;
}
```

```
/* Optional: Add spacing and style for the paragraph */
p {
  color: #555;
  font-size: 16px;
}
```

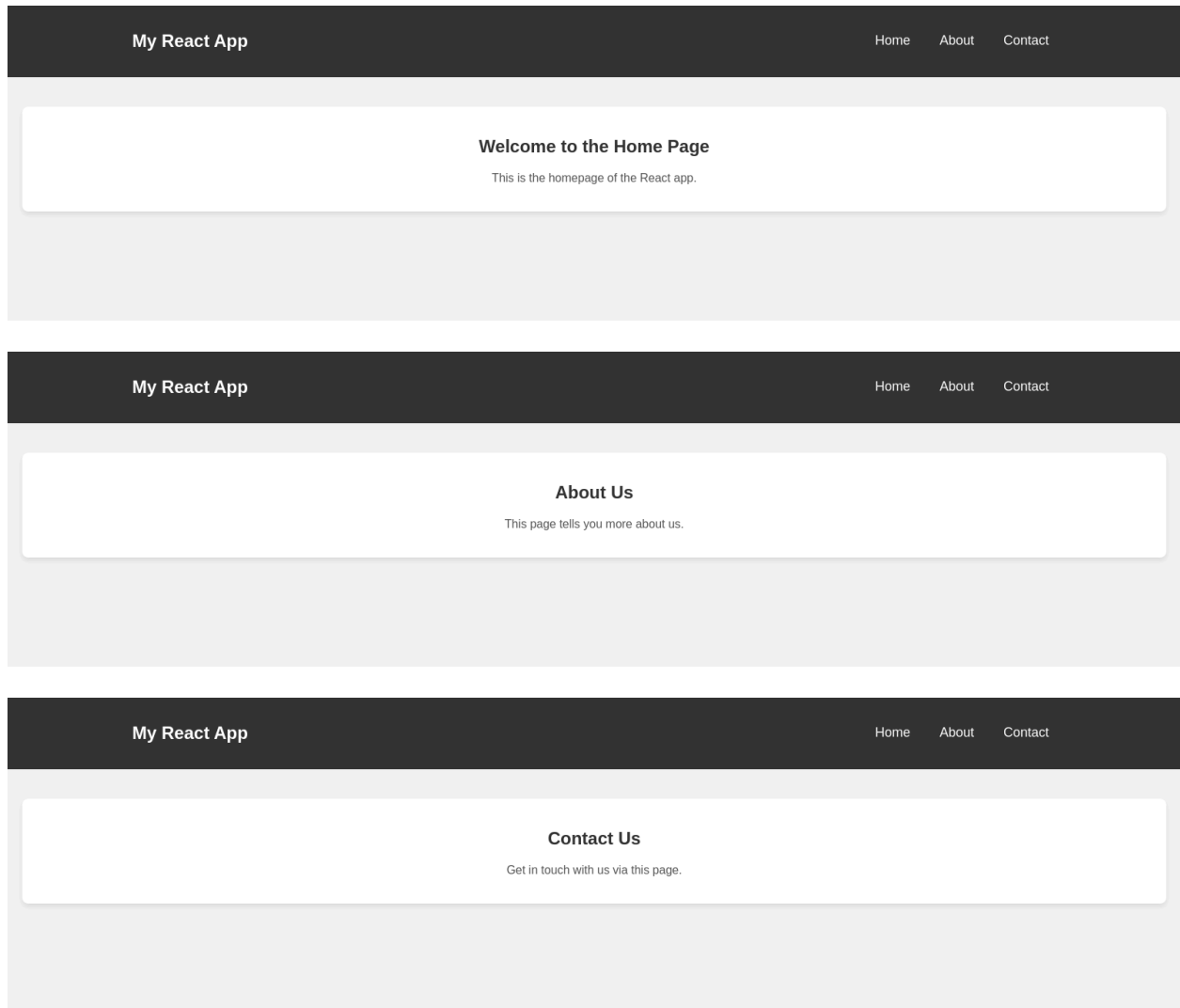
AWT LAB PROGRAMS

/ Navbar.css */*

```
.navbar {  
    background-color: #333;  
    color: white;  
    padding: 15px 0;  
    font-family: Arial, sans-serif;  
}  
  
.navbar-container {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    width: 80%;  
    margin: 0 auto;  
}  
  
.navbar-title {  
    font-size: 24px;  
    margin-left: 10px;  
}  
  
.navbar-links {  
    list-style: none;  
    display: flex;  
    gap: 20px;  
}  
  
.navbar-links li {  
    margin-right: 20px;  
}  
  
.navbar-links a {  
    color: white;  
    text-decoration: none;  
    font-size: 18px;  
}  
  
.navbar-links a:hover {  
    color: #4CAF50;  
}
```


AWT LAB PROGRAMS

OUTPUT PARTB4:-



AWT LAB PROGRAMS

6. Build a React app that fetches, adds, and deletes items from a mock backend server using the fetch API. Items should be displayed, and changes should update both the frontend and backend.

//src/Compomets/ItemList.js

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

```
const ItemList = () => {  
  const [items, setItems] = useState([]); // Initialize with an empty list  
  const [newItem, setNewItem] = useState("");
```

```
  // Fetch items from the mock backend (optional if backend has data)  
  const fetchItems = async () => {  
    const response = await fetch('https://jsonplaceholder.typicode.com/todos');  
    const data = await response.json();  
    setItems(data.slice(0, 0)); // Ensure empty array on initial fetch  
  };
```

```
  // Handle adding a new item  
  const addItem = async () => {  
    if (newItem.trim()) {  
      const newTodo = {  
        title: newItem,  
        completed: false,  
      };
```

```
      // POST the new item to the backend (mocking the backend response)  
      const response = await fetch('https://jsonplaceholder.typicode.com/todos', {  
        method: 'POST',  
        headers: {  
          'Content-Type': 'application/json',  
        },  
        body: JSON.stringify(newTodo),  
      });
```

```
      const addedItem = await response.json();  
      setItems([addedItem, ...items]); // Add the new item to the list  
      setNewItem("");  
    }  
  };
```

```
  // Handle deleting an item  
  const deleteItem = async (id) => {  
    // DELETE the item from the backend
```

AWT LAB PROGRAMS

```
    await fetch(`https://jsonplaceholder.typicode.com/todos/${id}`, {
      method: 'DELETE',
    });

    // Update the state after deletion
    setItems(items.filter(item => item.id !== id)); // Remove item locally
  };

  useEffect(() => {
    fetchItems(); // Fetch the items when the component mounts (optional for empty list)
  }, []);

  return (
    <div>
      <h2>Item Management</h2>

      <div>
        <input
          type="text"
          value={newItem}
          onChange={e => setNewItem(e.target.value)}
          placeholder="Enter new item"
        />
        <button onClick={addItem}>Add Item</button>
      </div>

      <ul>
        {items.map((item) => (
          <li key={item.id}>
            <span>{item.title}</span>
            <button className="delete" onClick={() => deleteItem(item.id)}>Delete</button>
          </li>
        ))}
      </ul>
    </div>
  );
};

export default ItemList;

//App.js
import React, { useState, useEffect } from 'react';
import './App.css';
```

AWT LAB PROGRAMS

```
function App() {
  const [items, setItems] = useState([]);
  const [newItem, setNewItem] = useState("");

  // Fetch items from the mock backend
  useEffect(() => {
    fetchItems();
  }, []);

  const fetchItems = async () => {
    const response = await fetch('http://localhost:5000/items');
    const data = await response.json();
    setItems(data);
  };

  const addItem = async () => {
    if (newItem.trim()) {
      const newItemData = { name: newItem };

      // POST the new item to the mock backend
      const response = await fetch('http://localhost:5000/items', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(newItemData),
      });

      const addedItem = await response.json();
      setItems([...items, addedItem]); // Add the new item to the frontend
      setNewItem(""); // Clear the input
    }
  };

  const deleteItem = async (id) => {
    // DELETE the item from the mock backend
    await fetch(`http://localhost:5000/items/${id}`, {
      method: 'DELETE',
    });

    // Remove the item from the frontend
    setItems(items.filter(item => item.id !== id));
  };

  return (
```

AWT LAB PROGRAMS

```
<div className="App">
  <h1>Item Management</h1>
  <div>
    <input
      type="text"
      value={newItem}
      onChange={(e) => setNewItem(e.target.value)}
      placeholder="Enter new item"
    />
    <button onClick={addItem}>Add Item</button>
  </div>

  <ul>
    {items.map(item => (
      <li key={item.id}>
        <span>{item.name}</span>
        <button className="delete" onClick={() => deleteItem(item.id)}>
          Delete
        </button>
      </li>
    ))}
  </ul>
</div>
);
}
```

export default App;

//App.css

```
.App {
  font-family: Arial, sans-serif;
  text-align: center;
  padding: 20px;
}
```

```
h1 {
  color: #333;
}
```

```
input {
  padding: 10px;
  font-size: 16px;
  margin-right: 10px;
}
```

AWT LAB PROGRAMS

```
button {  
  padding: 10px;  
  font-size: 16px;  
  background-color: #4CAF50;  
  color: white;  
  border: none;  
  cursor: pointer;  
}
```

```
button:hover {  
  background-color: #45a049;  
}
```

```
ul {  
  list-style-type: none;  
  padding: 0;  
}
```

```
li {  
  margin: 10px;  
  padding: 10px;  
  background-color: #f9f9f9;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}
```

```
button.delete {  
  background-color: #f44336;  
}
```

```
button.delete:hover {  
  background-color: #d32f2f;  
}
```

```
//db.json  
{  
  "items": [  
    {  
      "id": "1",  
      "name": "Item 1"  
    },  
    {
```

AWT LAB PROGRAMS

```
    "id": "2",  
    "name": "Item 2"  
  },  
  {  
    "id": "3",  
    "name": "Item 3"  
  }  
]  
}
```

//package.json

```
{  
  "name": "react-items-app",  
  "version": "1.0.0",  
  "dependencies": {  
    "react": "^17.0.2",  
    "react-dom": "^17.0.2",  
    "react-scripts": "4.0.3",  
    "json-server": "^0.16.3"  
  },  
  "scripts": {  
    "start": "react-scripts start",  
    "server": "json-server --watch db.json --port 5000"  
  }  
}
```

AWT LAB PROGRAMS

OUTPUT PARTB6:-

Item Management

Enter new item

Add Item

Item 1	Delete
Item 2	Delete
Item 3	Delete
101	Delete
102	Delete

Item Management

Enter new item

Add Item

Item 1	Delete
Item 2	Delete
Item 3	Delete

```
[
  {
    "id": "1",
    "name": "Item 1"
  },
  {
    "id": "2",
    "name": "Item 2"
  },
  {
    "id": "3",
    "name": "Item 3"
  },
  {
    "id": "d380",
    "name": "101"
  },
  {
    "id": "8c29",
    "name": "102"
  }
]
```

```
[
  {
    "id": "1",
    "name": "Item 1"
  },
  {
    "id": "2",
    "name": "Item 2"
  },
  {
    "id": "3",
    "name": "Item 3"
  }
]
```