Synoptic
Batch 1

Q1(a) - React Functional Component to Render Table

```
import React from 'react';

const EmployeeTable = () => {
  const data = [{"id": "E101", "department": "HR"}, {"id": "E102",
"department": "Finance"}];
  return (
    <table>
      <thead>
        <tr>
          <th>Employee ID</th>
          <th>Department</th>
        </tr>
      </thead>
      <tbody>
        {data.map((item, index) => (
          <tr key={index}>
            <td>{item.id}</td>
            <td>{item.department}</td>
          </tr>
        ))}
      </tbody>
    </table>
  );
}

export default EmployeeTable;
```

Output:

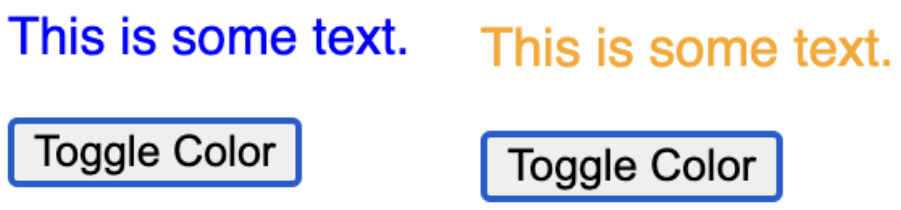
Employee ID	Department
E101	HR
E102	Finance

Q1(b) - React Class Component to Toggle Text Color

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

```
class ToggleTextColor extends Component {
  constructor(props) {
    super(props);
    this.state = {
      color: "Blue"
    };
  }

  toggleColor = () => {
    this.setState(prevState => ({
      color: prevState.color === "Blue" ? "Orange" : "Blue"
    }));
  }

  render() {
    return (
      <div>
        <p style={{ color: this.state.color }}>This is some
text.</p>
        <button onClick={this.toggleColor}>Toggle Color</button>
      </div>
    );
  }
}

export default ToggleTextColor;
```

Output:

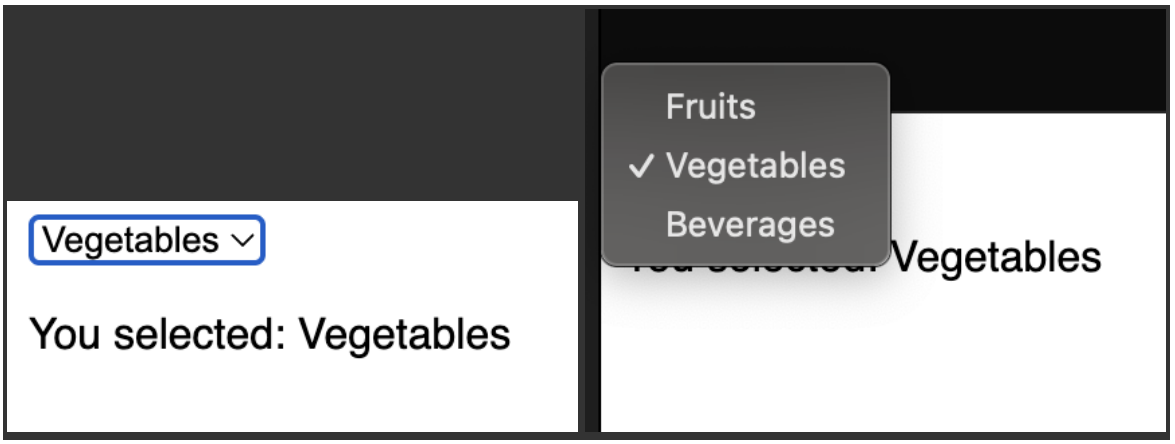


Q2 - React Dropdown with Categories

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

const CategoryDropdown = () => {
```

```
  const categories = ["Fruits", "Vegetables", "Beverages"];
  const [selectedCategory, setSelectedCategory] = useState('');

  return (
    <div>
      <select onChange={e => setSelectedCategory(e.target.value)}>
        {categories.map((category, index) => (
          <option key={index} value={category}>{category}</option>
        ))}
      </select>
      <p>You selected: {selectedCategory}</p>
    </div>
  );
}

export default CategoryDropdown;
```

Output:

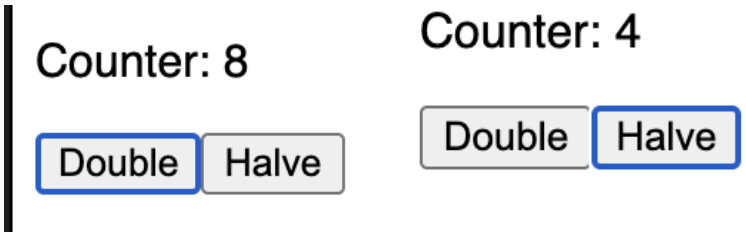


Q3 - React Counter with "Double" and "Halve"

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

const Counter = () => {
  const [count, setCount] = useState(1);

  const doubleCount = () => setCount(count * 2);
  const halveCount = () => setCount(count / 2);

  return (
    <div>
      <p>Counter: {count}</p>
```

```
      <button onClick={doubleCount}>Double</button>
      <button onClick={halveCount}>Halve</button>
    </div>
  );
}

export default Counter;
```

Output:

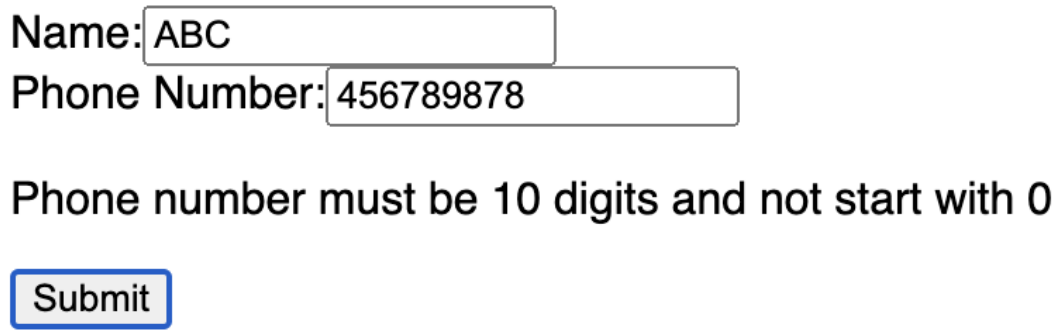


Q4 - React Form with Validation for Name and Phone Number

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

const Form = () => {
  const [name, setName] = useState('');
  const [phone, setPhone] = useState('');
  const [errors, setErrors] = useState({});

  const validateForm = () => {
    let errors = {};
    if (!/^[a-zA-Z]{3,}$/.test(name)) errors.name = "Name must be
at least 3 characters and contain only alphabets";
    if (!/^\d{10}$/.test(phone) || phone.startsWith('0'))
errors.phone = "Phone number must be 10 digits and not start with
0";
    setErrors(errors);
    return Object.keys(errors).length === 0;
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    if (validateForm()) {
      alert(`Name: ${name}, Phone: ${phone}`);
    }
  };
```

```
  return (
    <form onSubmit={handleSubmit}>
      <div>
        <label>Name:</label>
        <input type="text" value={name} onChange={e =>
setName(e.target.value)} />
        {errors.name && <p>{errors.name}</p>}
      </div>
      <div>
        <label>Phone Number:</label>
        <input type="text" value={phone} onChange={e =>
setPhone(e.target.value)} />
        {errors.phone && <p>{errors.phone}</p>}
      </div>
      <button type="submit">Submit</button>
    </form>
  );
}

export default Form;
```

Output:

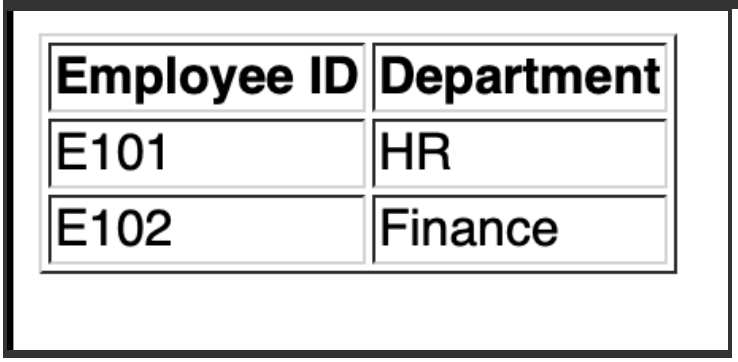


Batch 1:

1. React Functional Component to Render a List of Employees

```
import React from 'react';

const EmployeeTable = () => {
  const data = [{"id": "E101", "department": "HR"}, {"id": "E102",
"department": "Finance"}];
```

```
  return (
    <table border="1">
      <thead>
        <tr>
          <th>Employee ID</th>
          <th>Department</th>
        </tr>
      </thead>
      <tbody>
        {data.map((item, index) => (
          <tr key={index}>
            <td>{item.id}</td>
            <td>{item.department}</td>
          </tr>
        ))}
      </tbody>
    </table>
  );
}

export default EmployeeTable;
```



Employee ID	Department
E101	HR
E102	Finance

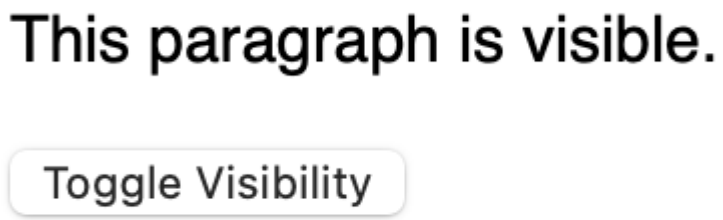
2. React Class Component to Toggle Visibility of a Paragraph

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

class ToggleVisibility extends Component {
  constructor(props) {
    super(props);
    this.state = {
      visible: true,
    };
  }

  toggleVisibility = () => {
    this.setState(prevState => ({
```

```
      visible: !prevState.visible,
    }));
  }

  render() {
    return (
      <div>
        {this.state.visible && <p>This paragraph is visible.</p>}
        <button onClick={this.toggleVisibility}>Toggle
Visibility</button>
      </div>
    );
  }
}

export default ToggleVisibility;
```

Before:

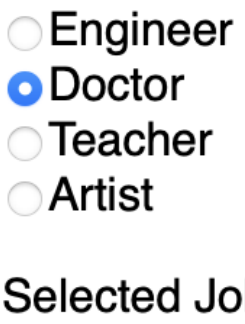
After:



3. React Functional Component to Render Radio Button Group

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

const JobSelection = () => {
  const jobs = ["Engineer", "Doctor", "Teacher", "Artist"];
  const [selectedJob, setSelectedJob] = useState("");
```

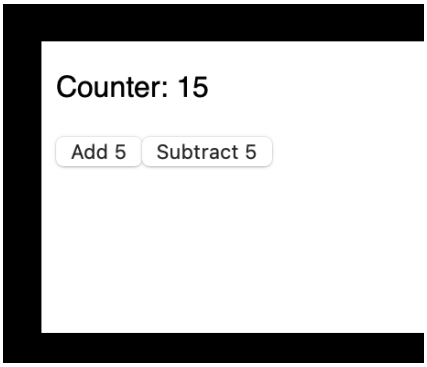
```
  const handleChange = (event) => {
    setSelectedJob(event.target.value);
  };

  return (
    <div>
      {jobs.map((job, index) => (
        <div key={index}>
          <input
            type="radio"
            id={job}
            name="job"
            value={job}
            onChange={handleChange}
          />
          <label htmlFor={job}>{job}</label>
        </div>
      ))}
      <p>Selected Job: {selectedJob}</p>
    </div>
  );
}

export default JobSelection;
```



4. React Functional Component with Counter (Add 5 and Subtract 5)

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

const Counter = () => {
  const [count, setCount] = useState(0);

  const addFive = () => setCount(count + 5);
  const subtractFive = () => setCount(count - 5);
```

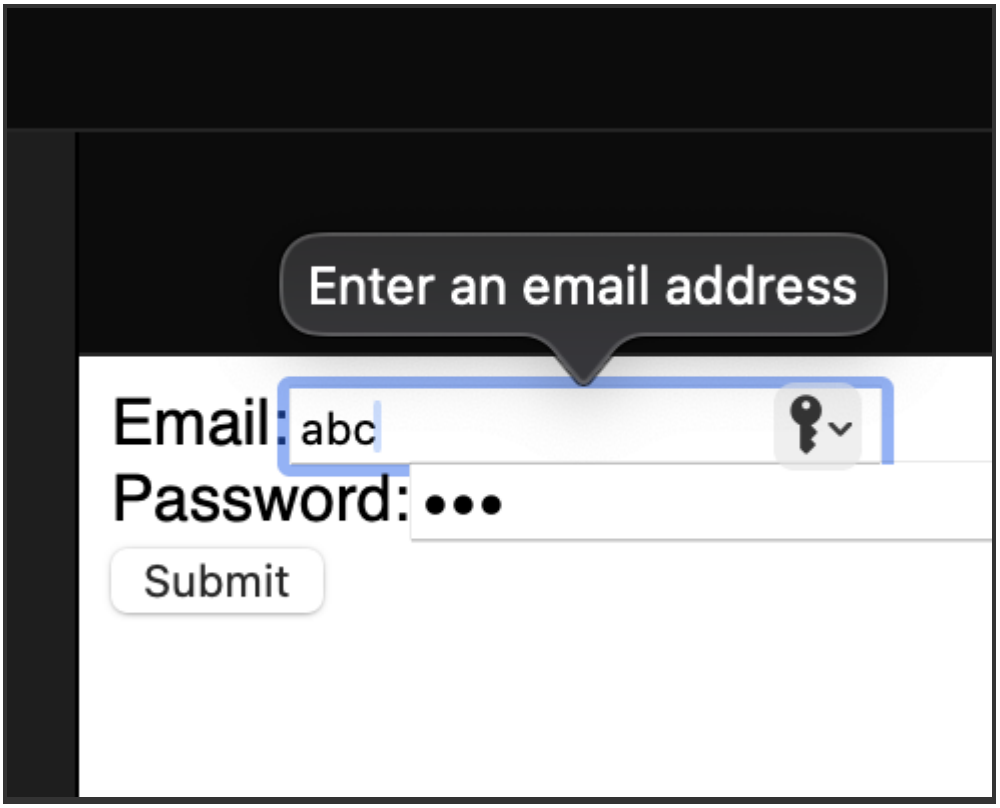
```
  return (
    <div>
      <p>Counter: {count}</p>
      <button onClick={addFive}>Add 5</button>
      <button onClick={subtractFive}>Subtract 5</button>
    </div>
  );
}

export default Counter;
```



5. React Form with Validation for Email and Password

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

const Form = () => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [errors, setErrors] = useState({});

  const validateForm = () => {
    const errorMessages = {};
    const emailRegex =
/^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

    if (!emailRegex.test(email)) errorMessages.email = "Please enter a
valid email.";
    if (password.length < 8) errorMessages.password = "Password must be
at least 8 characters.";

    setErrors(errorMessages);
    return Object.keys(errorMessages).length === 0;
  };
```

```
  const handleSubmit = (e) => {
    e.preventDefault();
    if (validateForm()) {
      alert(`Email: ${email}, Password: ${password}`);
    }
  };

  return (
    <form onSubmit={handleSubmit}>
      <div>
        <label>Email:</label>
        <input
          type="email"
          value={email}
          onChange={e => setEmail(e.target.value)}
        />
        {errors.email && <p>{errors.email}</p>}
      </div>
      <div>
        <label>Password:</label>
        <input
          type="password"
          value={password}
          onChange={e => setPassword(e.target.value)}
        />
        {errors.password && <p>{errors.password}</p>}
      </div>
      <button type="submit">Submit</button>
    </form>
  );
}

export default Form;
```

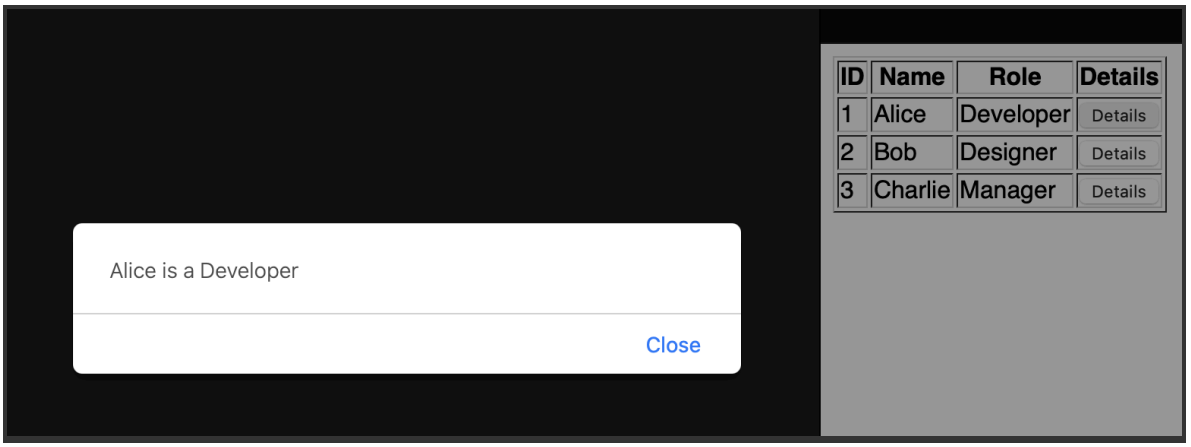


Batch 2:

1. React Functional Component for Dynamic HTML Table

```jsx
import React from 'react';

const EmployeeTable = () => {
  const data = [
    { id: 1, name: "Alice", role: "Developer" },
    { id: 2, name: "Bob", role: "Designer" },
    { id: 3, name: "Charlie", role: "Manager" }
  ];

  const handleDetails = (name, role) => {
    alert(`${name} is a ${role}`);
  };

  return (
    <table border="1">
      <thead>
        <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Role</th>
          <th>Details</th>
```

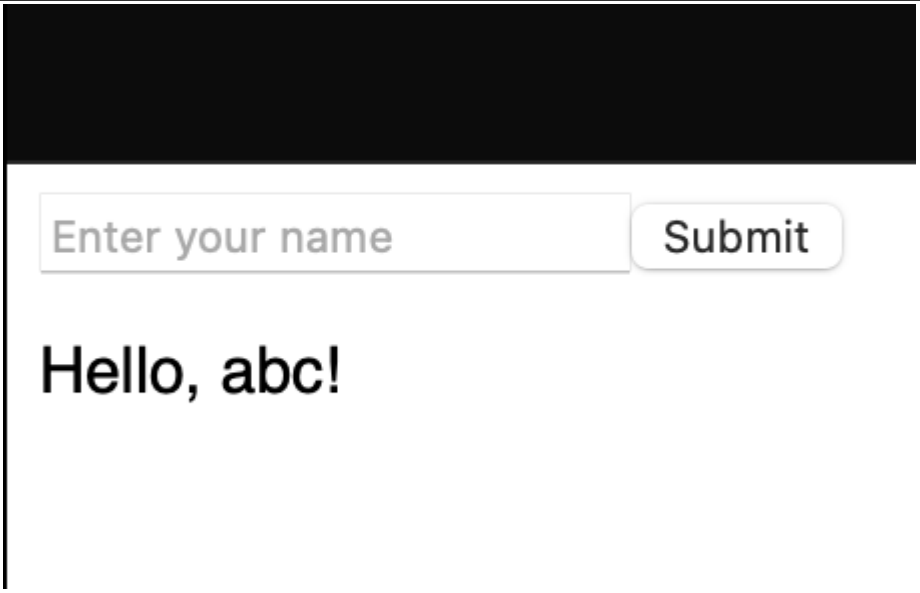
```
        </tr>
      </thead>
      <tbody>
        {data.map((item) => (
          <tr key={item.id}>
            <td>{item.id}</td>
            <td>{item.name}</td>
            <td>{item.role}</td>
            <td>
              <button onClick={() => handleDetails(item.name,
item.role)}>Details</button>
            </td>
          </tr>
        ))}
      </tbody>
    </table>
  );
};

export default EmployeeTable;
```



2. React Functional Component with Input and Submit Button

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

const GreetingForm = () => {
  const [name, setName] = useState('');
  const [greeting, setGreeting] = useState('');

  const handleSubmit = () => {
    setGreeting(`Hello, ${name}!`);
    setName('');
  };
```

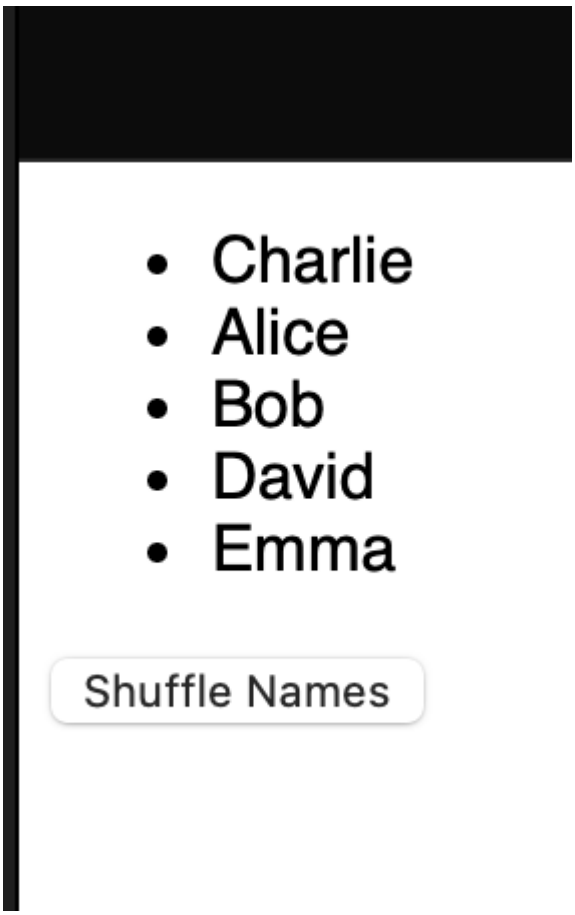
```
  return (
    <div>
      <input
        type="text"
        value={name}
        onChange={(e) => setName(e.target.value)}
        placeholder="Enter your name"
      />
      <button onClick={handleSubmit}>Submit</button>
      {greeting && <p>{greeting}</p>}
    </div>
  );
};

export default GreetingForm;
```



3. React Functional Component to Shuffle Student Names

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

const ShuffleNames = () => {
  const students = ["Alice", "Bob", "Charlie", "David", "Emma"];
  const [shuffledNames, setShuffledNames] = useState(students);

  const shuffleArray = (array) => {
    const shuffled = [...array];
    for (let i = shuffled.length - 1; i > 0; i--) {
      const j = Math.floor(Math.random() * (i + 1));
      [shuffled[i], shuffled[j]] = [shuffled[j], shuffled[i]];
    }
```

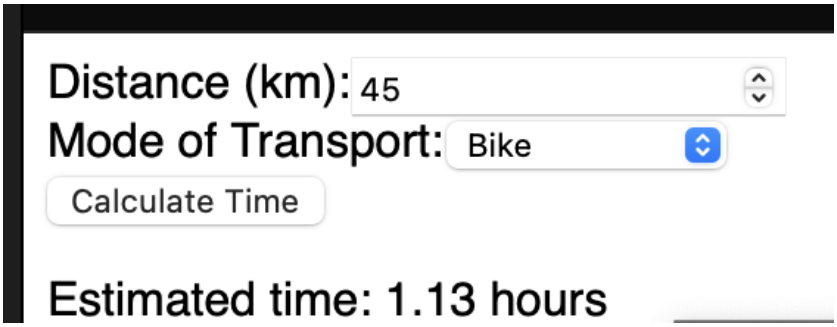
```
    return shuffled;
  };

  const handleShuffle = () => {
    setShuffledNames(shuffleArray(students));
  };

  return (
    <div>
      <ul>
        {shuffledNames.map((name, index) => (
          <li key={index}>{name}</li>
        ))}
      </ul>
      <button onClick={handleShuffle}>Shuffle Names</button>
    </div>
  );
};

export default ShuffleNames;
```

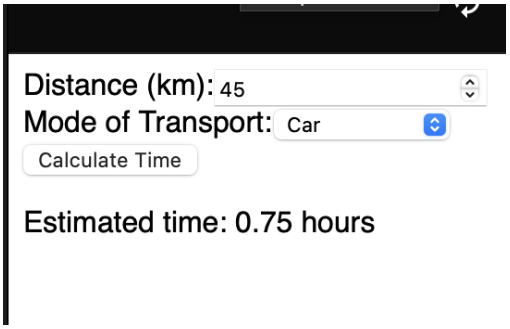


4. React Functional Component with Distance and Mode of Transport Form

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

const TransportForm = () => {
  const [distance, setDistance] = useState('');
  const [mode, setMode] = useState('');
  const [time, setTime] = useState('');

  const handleSubmit = () => {
    const distanceValue = parseFloat(distance);
    if (distanceValue <= 0 || !mode) {
      alert("Please enter a valid distance and mode of transport.");
      return;
    }

    let speed = 0;
    if (mode === "Car") speed = 60;
    else if (mode === "Bike") speed = 40;
    else if (mode === "Walk") speed = 5;

    const estimatedTime = (distanceValue / speed).toFixed(2);
    setTime(estimatedTime);
  };

  return (
    <div>
      <form onSubmit={(e) => e.preventDefault()}>
        <div>
          <label>Distance (km):</label>
          <input
            type="number"
            value={distance}
            onChange={(e) => setDistance(e.target.value)}
            placeholder="Enter distance"
          />
        </div>
        <div>
          <label>Mode of Transport:</label>
          <select value={mode} onChange={(e) =>
setMode(e.target.value)}>
            <option value="">Select Mode</option>
            <option value="Car">Car</option>
            <option value="Bike">Bike</option>
            <option value="Walk">Walk</option>
```

```
          </select>
        </div>
        <button onClick={handleSubmit}>Calculate Time</button>
      </form>
      {time && <p>Estimated time: {time} hours</p>}
    </div>
  );
};

export default TransportForm;
```

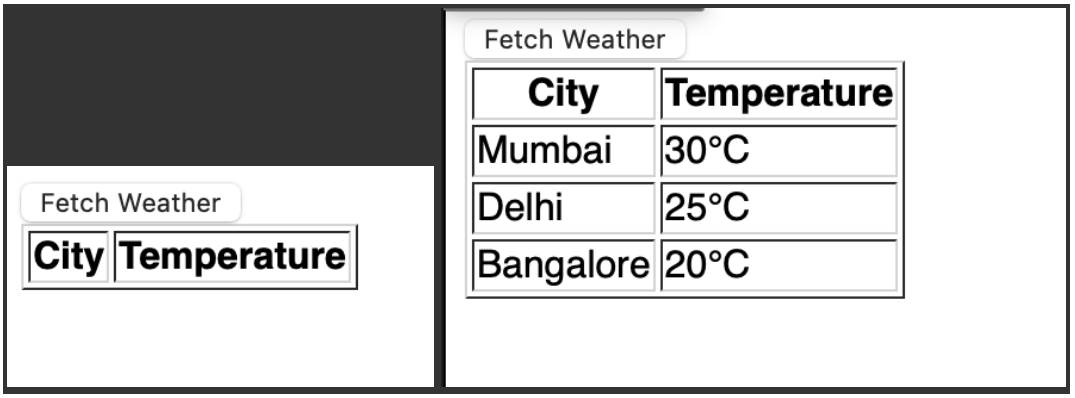




5. WeatherDashboard with WeatherDetails Componentx

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

class WeatherDashboard extends Component {
  constructor(props) {
    super(props);
    this.state = {
      weatherData: []
    };
  }

  fetchWeather = () => {
    const weatherData = [
      { city: "Mumbai", temperature: "30°C" },
```

```
      { city: "Delhi", temperature: "25°C" },
      { city: "Bangalore", temperature: "20°C" }
    ];
    this.setState({ weatherData });
  };

  render() {
    return (
      <div>
        <button onClick={this.fetchWeather}>Fetch Weather</button>
        <WeatherDetails weatherData={this.state.weatherData} />
      </div>
    );
  }
}

const WeatherDetails = ({ weatherData }) => {
  return (
    <table border="1">
      <thead>
        <tr>
          <th>City</th>
          <th>Temperature</th>
        </tr>
      </thead>
      <tbody>
        {weatherData.map((weather, index) => (
          <tr key={index}>
            <td>{weather.city}</td>
            <td>{weather.temperature}</td>
          </tr>
        ))}
      </tbody>
    </table>
  );
};

export default WeatherDashboard;
```



Batch 2:

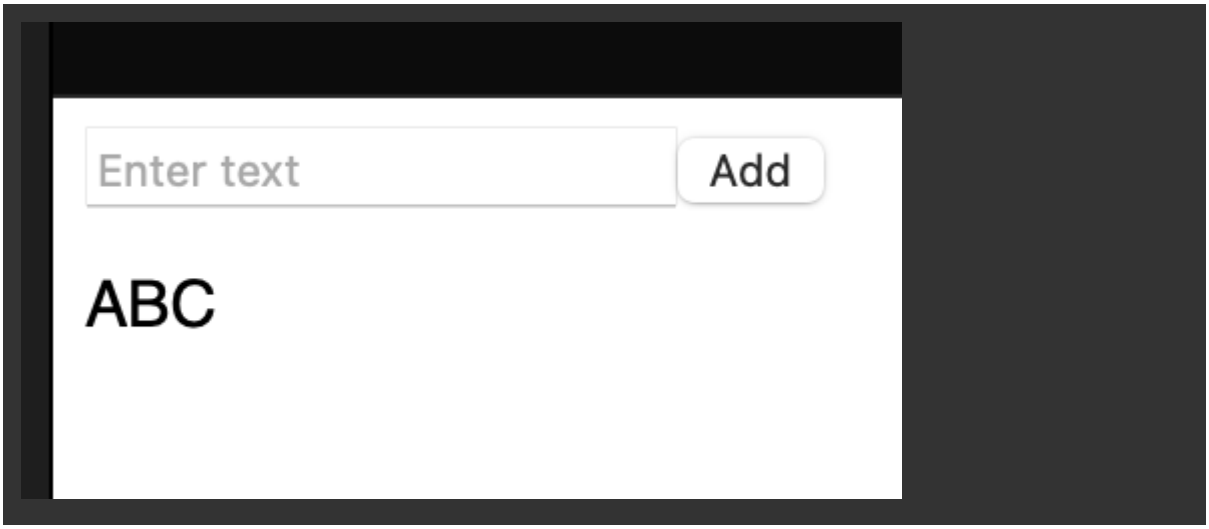
1. React Functional Component to Render an Ordered List with Remove Button

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

const ItemList = () => {
  const initialItems = [
    { id: 1, item: "React" },
    { id: 2, item: "JavaScript" },
    { id: 3, item: "CSS" }
  ];

  const [items, setItems] = useState(initialItems);

  const handleRemove = (id) => {
    setItems(items.filter(item => item.id !== id));
  };

  return (
    <div>
      <ol>
        {items.map((item) => (
          <li key={item.id}>
            {item.item}
            <button onClick={() =>
handleRemove(item.id)}>Remove</button>
          </li>
        ))}
      </ol>
    </div>
  );
```

```
};

export default ItemList;
```

2. React Functional Component with Input and Add Button

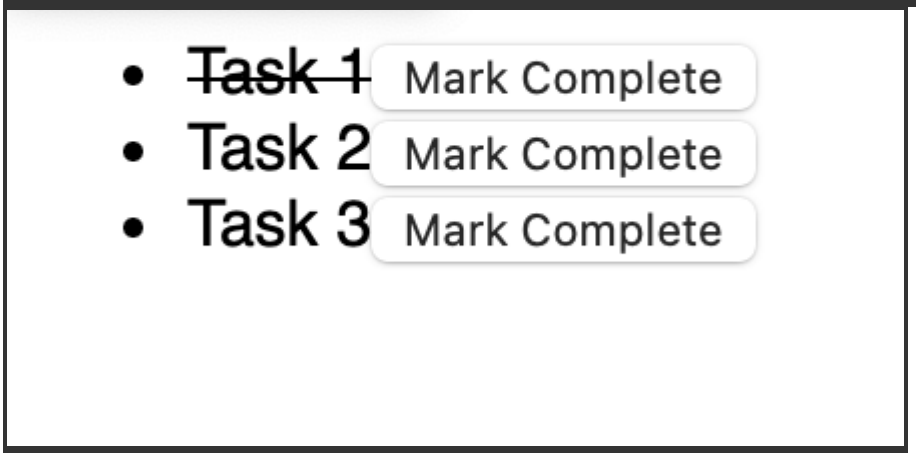
```
import React, { useState } from 'react';

const AddItem = () => {
  const [inputText, setInputText] = useState('');
  const [displayText, setDisplayText] = useState('');

  const handleAdd = () => {
    setDisplayText(inputText);
    setInputText('');
  };

  return (
    <div>
      <input
        type="text"
        value={inputText}
        onChange={(e) => setInputText(e.target.value)}
        placeholder="Enter text"
      />
      <button onClick={handleAdd}>Add</button>
      {displayText && <p>{displayText}</p>}
    </div>
  );
};

export default AddItem;
```



3. React Functional Component with Tasks and Mark Complete Button

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

const TaskList = () => {
  const initialTasks = [
    { id: 1, name: "Task 1", completed: false },
    { id: 2, name: "Task 2", completed: false },
    { id: 3, name: "Task 3", completed: false }
  ];

  const [tasks, setTasks] = useState(initialTasks);

  const markComplete = (id) => {
    setTasks(tasks.map(task =>
      task.id === id ? { ...task, completed: !task.completed } : task
    ));
  };

  return (
    <div>
      <ul>
        {tasks.map((task) => (
          <li
            key={task.id}
            style={{ textDecoration: task.completed ? 'line-through' :
'none' }}
          >
            {task.name}
            <button onClick={() => markComplete(task.id)}>Mark
Complete</button>
```

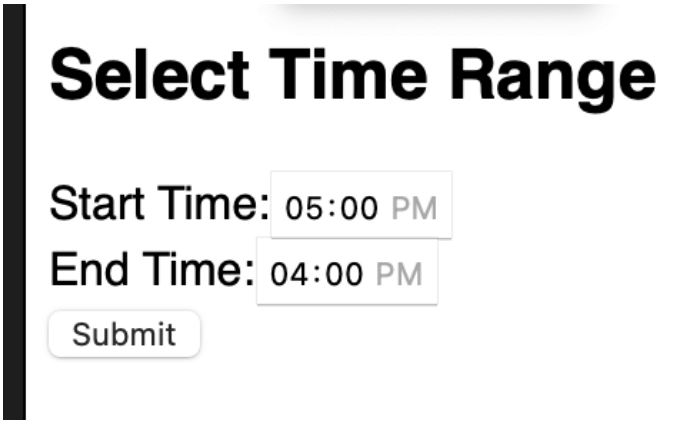
```
          </li>
        ))}
      </ul>
    </div>
  );
};

export default TaskList;
```



4. React Functional Component with Time Validation Form

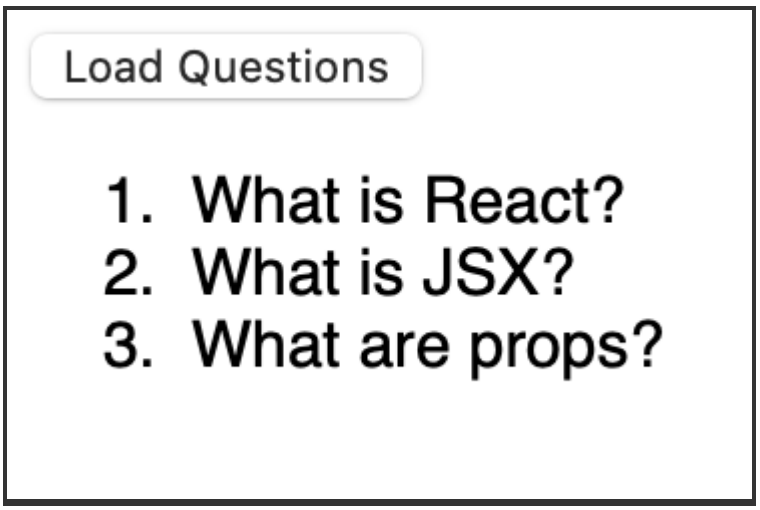
```
import { useState } from "react";

const TimeSelectionForm = () => {
  const [startTime, setStartTime] = useState("");
  const [endTime, setEndTime] = useState("");
  const [message, setMessage] = useState("");

  const handleSubmit = (e) => {
    e.preventDefault();

    if (startTime && endTime) {
      if (endTime > startTime) {
        setMessage(`Start Time: ${startTime}, End Time: ${endTime}`);
      } else {
        setMessage("Error: End Time must be later than Start Time.");
      }
    } else {
      setMessage("Error: Please select both Start Time and End Time.");
    }
  };
```

```
  return (
    <div>
      <h2>Select Time Range</h2>
      <form onSubmit={handleSubmit}>
        <div>
          <label>Start Time:</label>
          <input
            type="time"
            value={startTime}
            onChange={(e) => setStartTime(e.target.value)}
            required
          />
        </div>
        <div>
          <label>End Time:</label>
          <input
            type="time"
            value={endTime}
            onChange={(e) => setEndTime(e.target.value)}
            required
          />
        </div>
        <button type="submit">Submit</button>
      </form>
      {message && <p>{message}</p>}
    </div>
  );
};

export default TimeSelectionForm;
```



5. Class Component QuizManager with Question List

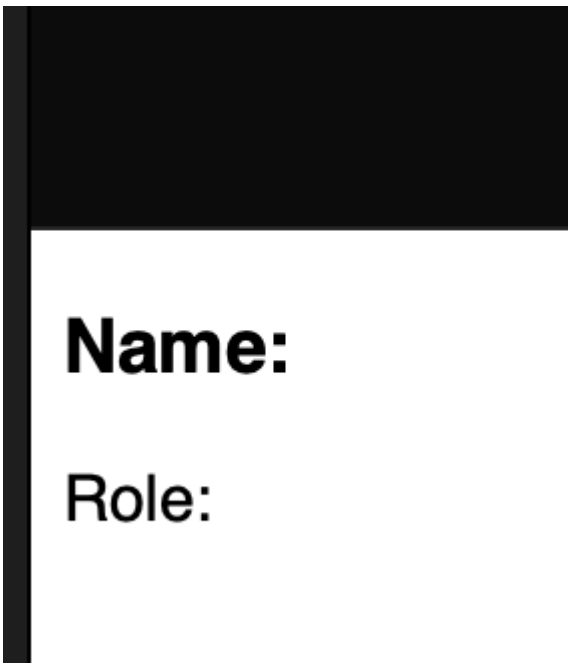
```
import React, { Component } from 'react';

class QuizManager extends Component {
  constructor(props) {
    super(props);
    this.state = {
      questions: []
    };
  }

  loadQuestions = () => {
    const questions = [
      { id: 1, question: "What is React?" },
      { id: 2, question: "What is JSX?" },
      { id: 3, question: "What are props?" }
    ];
    this.setState({ questions });
  };

  render() {
    return (
      <div>
        <button onClick={this.loadQuestions}>Load Questions</button>
        <QuestionList questions={this.state.questions} />
      </div>
    );
  }
}

const QuestionList = ({ questions }) => {
  return (
    <ol>
      {questions.map((q) => (
        <li key={q.id}>{q.question}</li>
      ))}
    </ol>
  );
};

export default QuizManager;
```



Batch 3:

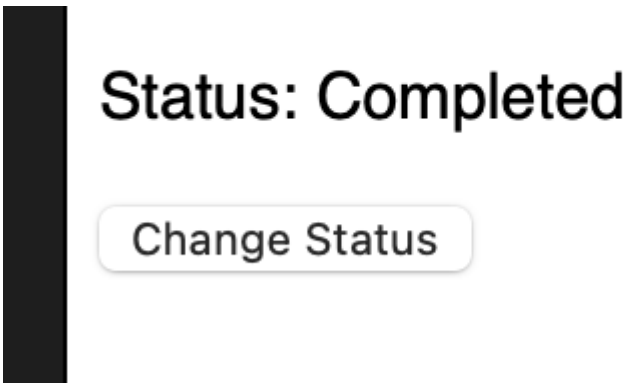
1. UserCard Component (with Name and Role as Props)

```jsx
import React from 'react';

const UserCard = ({ name, role }) => {
  return (
    <div className="card">
      <h3>Name: {name}</h3>
      <p>Role: {role}</p>
    </div>
  );
};

export default UserCard;
```



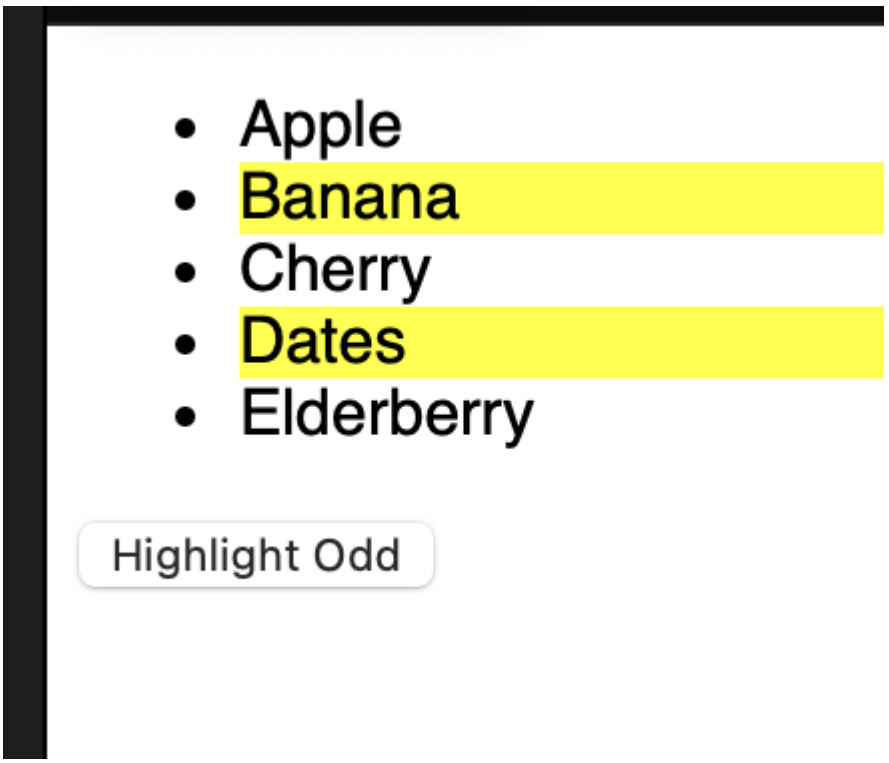
2. Task Status Toggle Component (using useState Hook)

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

const TaskStatus = () => {
  const [status, setStatus] = useState('Pending');

  const toggleStatus = () => {
    setStatus(status === 'Pending' ? 'Completed' : 'Pending');
  };

  return (
    <div>
      <p>Status: {status}</p>
      <button onClick={toggleStatus}>Change Status</button>
    </div>
  );
};

export default TaskStatus;
```



3. Highlight Odd Items in List Component

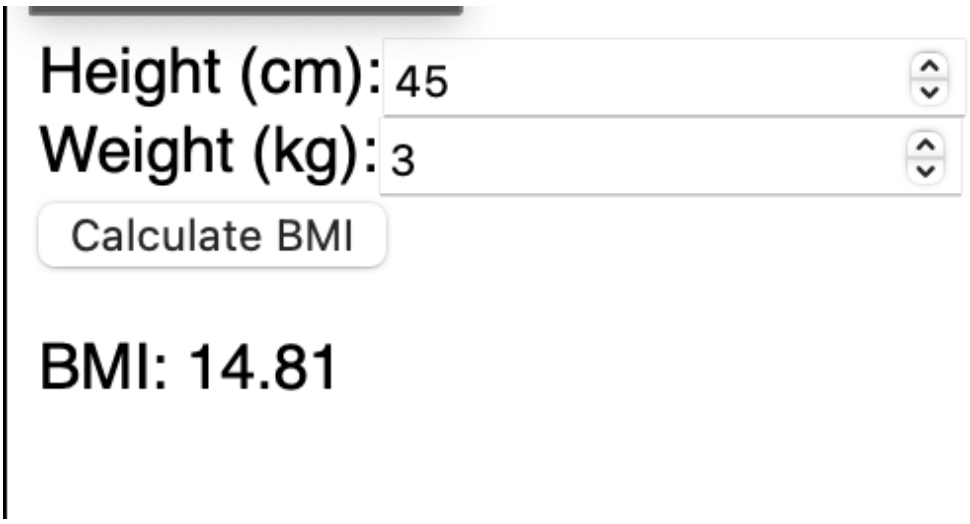
```
import React, { useState } from 'react';

const HighlightOddItems = () => {
  const items = ["Apple", "Banana", "Cherry", "Dates", "Elderberry"];
  const [highlightedItems, setHighlightedItems] = useState(items);

  const highlightOdd = () => {
    const updatedItems = items.map((item, index) => ({
      text: item,
      isHighlighted: index % 2 !== 0,
    }));
    setHighlightedItems(updatedItems);
  };

  return (
    <div>
      <ul>
        {highlightedItems.map((item, index) => (
          <li
            key={index}
            style={{ backgroundColor: item.isHighlighted ? 'yellow' :
'transparent' }}
          >
            {item.text}
          </li>
        ))}
      </ul>
      <button onClick={highlightOdd}>Highlight Odd</button>
    </div>
  );
};

export default HighlightOddItems;
```



4. BMI Calculation Form (with Height and Weight Validation)

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

const BMIForm = () => {
  const [height, setHeight] = useState('');
  const [weight, setWeight] = useState('');
  const [bmi, setBmi] = useState(null);
  const [error, setError] = useState('');

  const calculateBMI = () => {
    if (height <= 0 || weight <= 0) {
      setError("Height and weight must be positive numbers.");
      setBmi(null);
    } else {
      const calculatedBMI = (weight / ((height / 100) ** 2)).toFixed(2);
      setBmi(calculatedBMI);
      setError('');
    }
  };

  return (
    <div>
      <form onSubmit={(e) => e.preventDefault()}>
        <div>
```

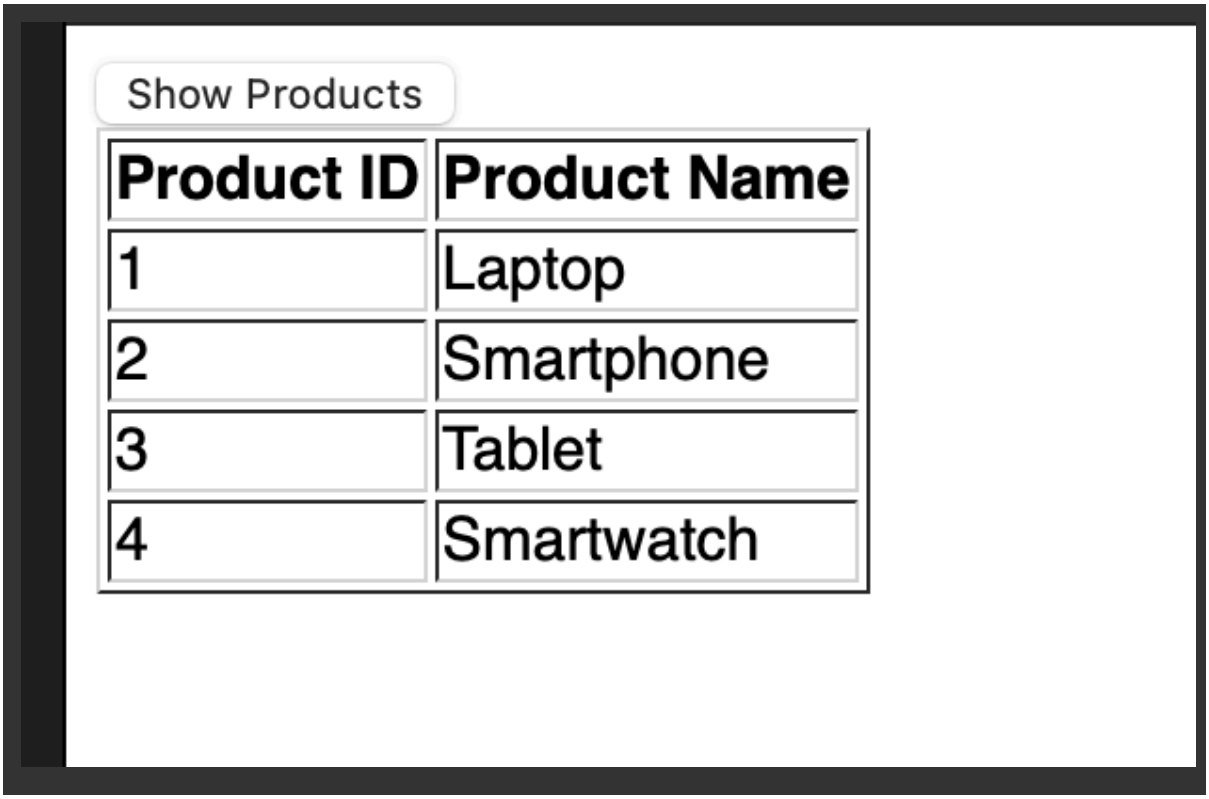
```
          <label>Height (cm):</label>
          <input
            type="number"
            value={height}
            onChange={(e) => setHeight(e.target.value)}
          />
        </div>
        <div>
          <label>Weight (kg):</label>
          <input
            type="number"
            value={weight}
            onChange={(e) => setWeight(e.target.value)}
          />
        </div>
        <button onClick={calculateBMI}>Calculate BMI</button>
      </form>
      {error && <p style={{ color: 'red' }}>{error}</p>}
      {bmi && <p>BMI: {bmi}</p>}
    </div>
  );
};

export default BMIForm;
```

- 

5. ProductManager Class Component and ProductList Functional Component

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

```
class ProductManager extends Component {
  constructor(props) {
    super(props);
    this.state = {
      products: []
    };
  }

  loadProducts = () => {
    const products = ["Laptop", "Smartphone", "Tablet", "Smartwatch"];
    this.setState({ products });
  };

  render() {
    return (
      <div>
        <button onClick={this.loadProducts}>Show Products</button>
        <ProductList products={this.state.products} />
      </div>
    );
  }
}

const ProductList = ({ products }) => {
  return (
    <table border="1">
      <thead>
        <tr>
          <th>Product ID</th>
          <th>Product Name</th>
        </tr>
      </thead>
      <tbody>
        {products.map((product, index) => (
          <tr key={index}>
            <td>{index + 1}</td>
            <td>{product}</td>
          </tr>
        ))}
      </tbody>
    </table>
  );
};

export default ProductManager;
```



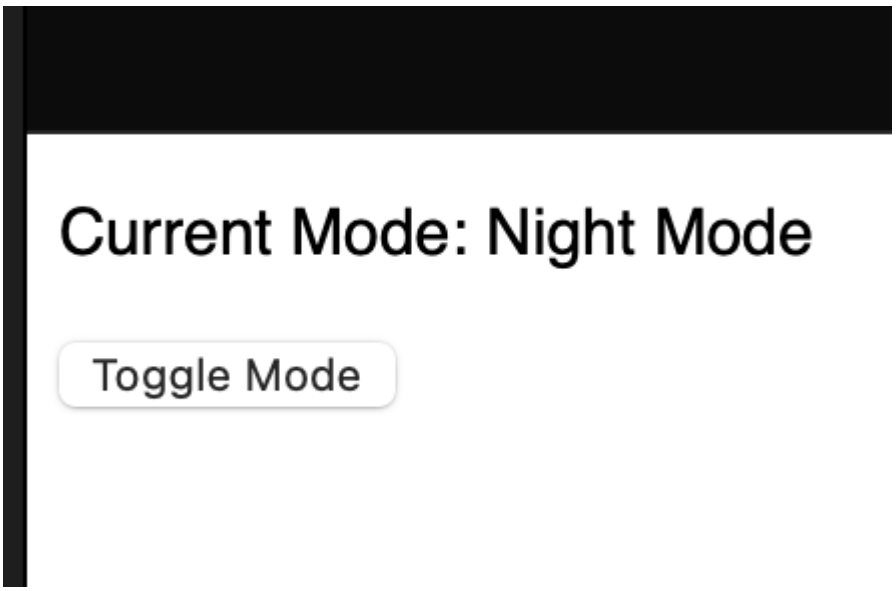
Batch 3:

1. CityCard Component (with cityName and population as Props)

```jsx
import React from 'react';

const CityCard = ({ cityName, population }) => {
  return (
    <div className="card">
      <h3>City: {cityName}</h3>
      <p>Population: {population}</p>
    </div>
  );
};

export default CityCard;
```

2. Day/Night Mode Toggle Component (using useState)

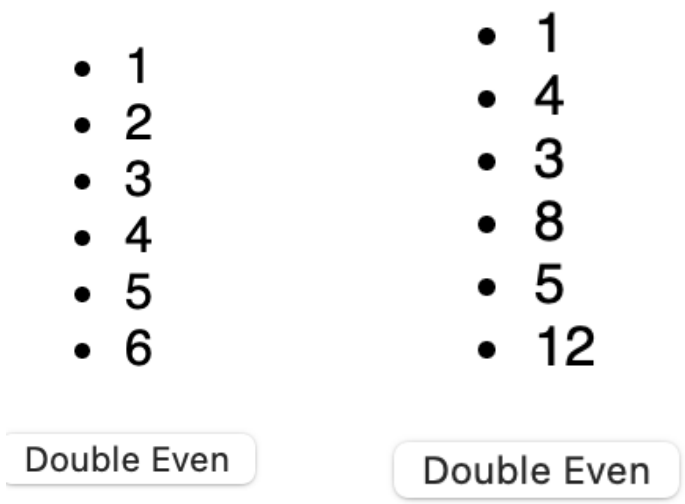
```jsx
import React, { useState } from 'react';

const ModeToggle = () => {
  const [mode, setMode] = useState('Day Mode');

  const toggleMode = () => {
    setMode(mode === 'Day Mode' ? 'Night Mode' : 'Day Mode');
  };

  return (
    <div>
      <p>Current Mode: {mode}</p>
      <button onClick={toggleMode}>Toggle Mode</button>
    </div>
  );
};

export default ModeToggle;
```



3. Double Even Numbers in List Component

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

const DoubleEvenNumbers = () => {
  const initialNumbers = [1, 2, 3, 4, 5, 6];
  const [numbers, setNumbers] = useState(initialNumbers);

  const doubleEvenNumbers = () => {
    const updatedNumbers = numbers.map((num, index) =>
      index % 2 === 1 ? num * 2 : num
    );
    setNumbers(updatedNumbers);
  };

  return (
    <div>
      <ul>
        {numbers.map((num, index) => (
          <li key={index}>{num}</li>
        ))}
      </ul>
      <button onClick={doubleEvenNumbers}>Double Even</button>
    </div>
  );
};
```

```
export default DoubleEvenNumbers;
```



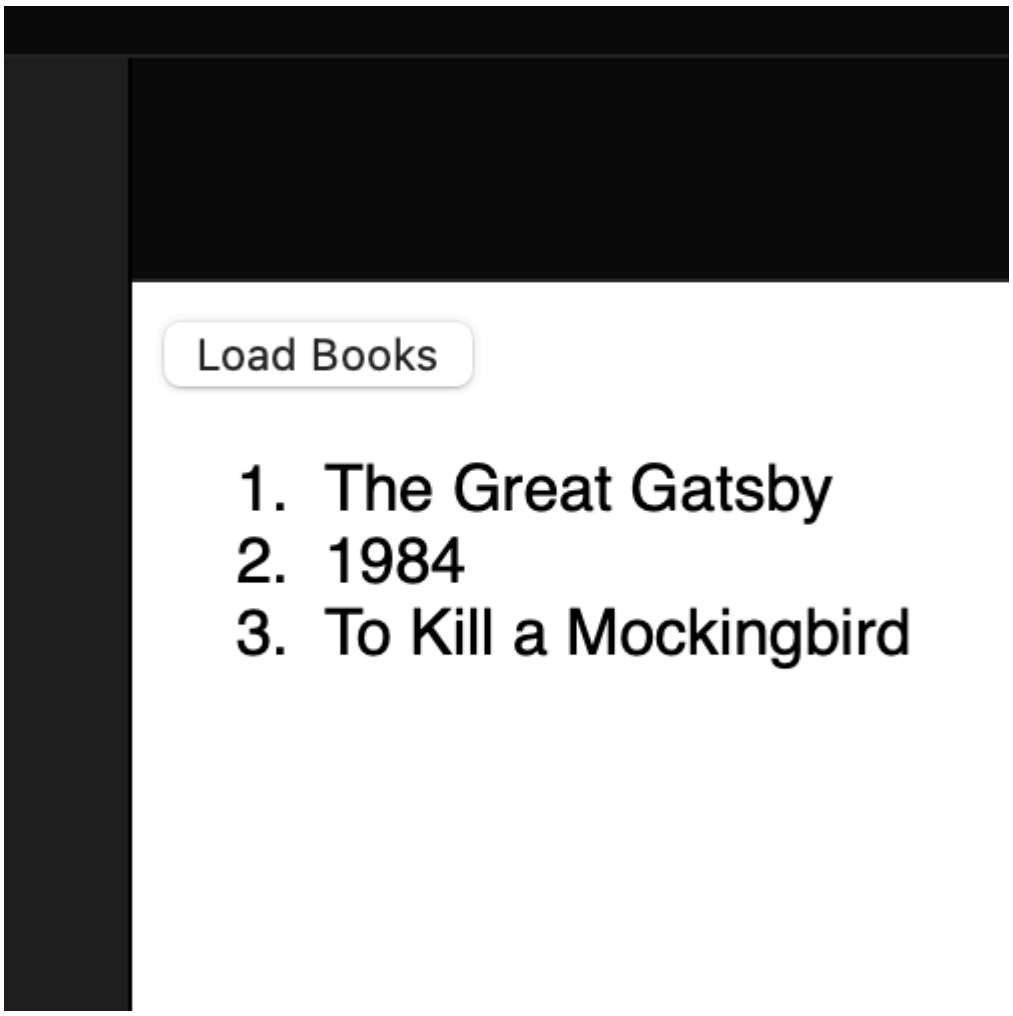
4. Date of Birth and Favorite Color Form Component

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

const FormComponent = () => {
  const [dob, setDob] = useState('');
  const [color, setColor] = useState('');
  const [error, setError] = useState('');
  const [confirmation, setConfirmation] = useState('');

  const handleSubmit = () => {
    if (!dob || !color) {
      setError("Both Date of Birth and Favorite Color are required.");
      setConfirmation('');
    } else {
      setError('');
      setConfirmation(`Date of Birth: ${dob}, Favorite Color:
${color}`);
    }
  };

  return (
    <div>
```

```
      <form onSubmit={(e) => e.preventDefault()}>
        <div>
          <label>Date of Birth:</label>
          <input
            type="date"
            value={dob}
            onChange={(e) => setDob(e.target.value)}
          />
        </div>
        <div>
          <label>Favorite Color:</label>
          <select
            value={color}
            onChange={(e) => setColor(e.target.value)}
          >
            <option value="">Select Color</option>
            <option value="Red">Red</option>
            <option value="Green">Green</option>
            <option value="Blue">Blue</option>
          </select>
        </div>
        <button onClick={handleSubmit}>Submit</button>
      </form>
      {error && <p style={{ color: 'red' }}>{error}</p>}
      {confirmation && <p>{confirmation}</p>}
    </div>
  );
};

export default FormComponent;
```

5. LibraryManager Class Component and BookList Functional Component

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

class LibraryManager extends Component {
  constructor(props) {
    super(props);
    this.state = {
      books: []
    };
  }

  loadBooks = () => {
    const books = [
      "The Great Gatsby",
      "1984",
      "To Kill a Mockingbird"
    ];
    this.setState({ books });
  };

  render() {
    return (
      <div>
        <button onClick={this.loadBooks}>Load Books</button>
        <BookList books={this.state.books} />
      </div>
    );
  }
}

const BookList = ({ books }) => {
  return (
    <ol>
      {books.map((book, index) => (
        <li key={index}>{book}</li>
      ))}
    </ol>
  );
};

export default LibraryManager;
```


Load Books
1. The Great Gatsby
2. 1984
3. To Kill a Mockingbird