Full Stack Development

React JS – Hands on Assignment -2

Mansi shelar
Div -B
2401184

1. Add a button that disables itself after 3 clicks (use state to track clicks).

Code:

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

export default function ClickLimitButton() {
  // State to track the number of clicks
  const [clickCount, setClickCount] = useState(0);

  // Handle button clicks
  const handleClick = () => {
    setClickCount((prevCount) => prevCount + 1);
  };

  return (
    <div className="flex flex-col items-center justify-center min-h-screen bg-
gray-100">
      <button
        onClick={handleClick}
        disabled={clickCount >= 3} // Disable button after 3 clicks
        className={`px-4 py-2 text-white font-bold rounded-lg shadow-lg ${
          clickCount >= 3 ? "bg-gray-400 cursor-not-allowed" : "bg-blue-500"
        }`}
      >
        {clickCount >= 3 ? "Disabled" : `Click Me (${3 - clickCount} left)`}
      </button>
    </div>
  );
}
```

Output :

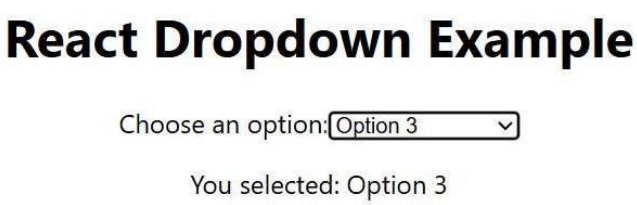
2. Implement a <select> dropdown that updates state with the selected value.

Code:

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

function Dropdown() {
  // Initialize state to store the selected value
  const [selectedValue, setSelectedValue] = useState("");

  // Handler for the dropdown change event
  const handleChange = (event) => {
    setSelectedValue(event.target.value); // Update state with the selected
value
  };

  return (
    <div className="flex flex-col items-center mt-4">
      <label htmlFor="options" className="text-lg font-medium mb-2">
        Choose an option:
      </label>
      <select
        id="options"
        value={selectedValue}
        onChange={handleChange}
        className="p-2 border rounded-lg shadow-md"
      >
        <option value="" disabled>
          Select an option
        </option>
        <option value="Option 1">Option 1</option>
        <option value="Option 2">Option 2</option>
        <option value="Option 3">Option 3</option>
      </select>
      {selectedValue && (
        <p className="mt-4 text-lg font-medium">
          You selected: <span className="text-blue-600">{selectedValue}</span>
        </p>
      )}
    </div>
  );
}
```

```
export default Dropdown;
```

Output:



3. Build a AADHAR Registration form with an input field and a submit button.

Code:

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

function AadharForm() {
  // State for AADHAR input and error message
  const [aadhar, setAadhar] = useState("");
  const [error, setError] = useState("");

  // Handler for input change
  const handleInputChange = (event) => {
    const value = event.target.value;
    // Allow only numbers and ensure max 12 characters
    if (/^\d{0,12}$/.test(value)) {
      setAadhar(value);
      setError(""); // Clear error if input is valid
    }
  };

  // Handler for form submission
  const handleSubmit = (event) => {
    event.preventDefault(); // Prevent page refresh
    if (aadhar.length !== 12) {
      setError("AADHAR number must be exactly 12 digits.");
    } else {
      alert(`AADHAR number ${aadhar} submitted successfully!`);
      setAadhar(""); // Reset input field
    }
  };
```

```
  return (
    <div className="flex flex-col items-center mt-8">
      <h1 className="text-2xl font-bold mb-4">AADHAR Registration Form</h1>
      <form
        onSubmit={handleSubmit}
        className="flex flex-col items-center p-4 border rounded-lg shadow-md
w-80"
      >
        <label htmlFor="aadhar" className="text-lg font-medium mb-2">
          Enter AADHAR Number:
        </label>
        <input
          type="text"
          id="aadhar"
          value={aadhar}
          onChange={handleInputChange}
          className="p-2 border rounded-lg w-full mb-2"
          placeholder="Enter 12-digit AADHAR"
          required
        />
        {error && <p className="text-red-600 text-sm mb-2">{error}</p>}
        <button
          type="submit"
          className="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-
blue-700"
        >
          Submit
        </button>
      </form>
    </div>
  );
}

export default AadharForm;
```

Output:

AADHAR Registration Form

Enter AADHAR Number: 123456789101 Submit

4. Add Form Validation in above program.

Code:

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

function AadharForm() {
  // State for AADHAR input, error message, and submission status
  const [aadhar, setAadhar] = useState("");
  const [error, setError] = useState("");
  const [submitted, setSubmitted] = useState(false);

  // Validate AADHAR number format (12 digits)
  const validateAadhar = (number) => {
    return /^\d{12}$/.test(number);
  };

  // Handler for input change
  const handleInputChange = (event) => {
    const value = event.target.value;
    // Allow only numbers and limit to 12 characters
    if (/^\d{0,12}$/.test(value)) {
      setAadhar(value);
      setError(""); // Clear error when input is valid
      setSubmitted(false); // Reset submission status on change
    }
  };

  // Handler for form submission
  const handleSubmit = (event) => {
    event.preventDefault(); // Prevent page refresh
    if (!validateAadhar(aadhar)) {
      setError("Invalid AADHAR number. It must be exactly 12 digits.");
      setSubmitted(false);
    } else {
      setError("");
      setSubmitted(true);
      alert(`AADHAR number ${aadhar} submitted successfully!`);
      setAadhar(""); // Reset input field
    }
  };

  return (
    <div className="flex flex-col items-center mt-8">
```

```
      <h1 className="text-2xl font-bold mb-4">AADHAR Registration Form</h1>
      <form
        onSubmit={handleSubmit}
        className="flex flex-col items-center p-4 border rounded-lg shadow-md
w-80"
      >
        <label htmlFor="aadhar" className="text-lg font-medium mb-2">
          Enter AADHAR Number:
        </label>
        <input
          type="text"
          id="aadhar"
          value={aadhar}
          onChange={handleInputChange}
          className={`p-2 border rounded-lg w-full mb-2 ${
            error ? "border-red-500" : "border-gray-300"
          }`}
          placeholder="Enter 12-digit AADHAR"
          required
        />
        {error && <p className="text-red-600 text-sm mb-2">{error}</p>}
        <button
          type="submit"
          className="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-
blue-700"
          disabled={submitted} // Disable button after successful submission
        >
          {submitted ? "Submitted" : "Submit"}
        </button>
        {submitted && (
          <p className="text-green-600 text-sm mt-2">Form submitted
successfully!</p>
        )}
      </form>
    </div>
  );
}

export default AadharForm;
```

Output:


AADHAR Registration Form
Enter AADHAR Number:
Enter 12-digit AADHAR
Submitted
Form submitted successfully!

5. Write a program to demonstrate functional component life cycle phases.

Code:

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

function LifecycleDemo() {
  const [count, setCount] = useState(0);
  const [text, setText] = useState("");

  //  Mounting: This runs once when the component is first rendered
  useEffect(() => {
    console.log("Component Mounted");
    alert("Welcome! Component Mounted.");

    // Cleanup during unmounting
    return () => {
      console.log("Component Unmounted");
      alert("Goodbye! Component Unmounted.");
    };
  }, []); // Empty dependency array ensures this runs only once

  //  Updating: Runs whenever `count` or `text` state changes
  useEffect(() => {
    console.log(`Count updated to ${count}`);
  }, [count]); // Dependency on `count`

  useEffect(() => {
    console.log(`Text updated to: ${text}`);
  }, [text]); // Dependency on `text`

  return (
    <div className="flex flex-col items-center mt-8">
      <h1 className="text-2xl font-bold mb-4">React Functional Component
Lifecycle</h1>
      <div className="mb-4">
        <p className="text-lg">Count: {count}</p>
        <button
          onClick={() => setCount(count + 1)}
```

```
          className="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-
blue-700 mr-2"
        >
          Increment Count
        </button>
        <button
          onClick={() => setCount(0)}
          className="bg-gray-600 text-white px-4 py-2 rounded-lg hover:bg-
gray-700"
        >
          Reset Count
        </button>
      </div>
      <div>
        <input
          type="text"
          value={text}
          onChange={(e) => setText(e.target.value)}
          placeholder="Type something..."
          className="p-2 border rounded-lg w-64 mb-2"
        />
      </div>
    </div>
  );
}

export default LifecycleDemo;
```

Output:

6. Create a timer that updates the seconds elapsed every seconds.

Code:

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

const Timer = () => {
  const [seconds, setSeconds] = useState(0); // State to track seconds elapsed

  useEffect(() => {
    // Start a timer when the component mounts
    const interval = setInterval(() => {
      setSeconds((prevSeconds) => prevSeconds + 1); // Increment seconds
    }, 1000);

    // Cleanup: Stop the timer when the component unmounts
    return () => clearInterval(interval);
  }, []);

  return (
    <div className="timer-container">
      <h1>Timer</h1>
      <p>Seconds Elapsed: {seconds}</p>
    </div>
  );
};

export default Timer;
```

Output:

7. Use useState to manage the visibility of a paragraph (toggle text with a button).

Code:

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

const ToggleText = () => {
  const [isVisible, setIsVisible] = useState(false); // State to manage
visibility

  const toggleParagraph = () => {
    setIsVisible((prevVisible) => !prevVisible); // Toggle visibility
  };

  return (
    <div className="toggle-container">
      <h1>Toggle Paragraph Visibility</h1>
      <button onClick={toggleParagraph} className="toggle-button">
        {isVisible ? "Hide Text" : "Show Text"}
      </button>
      {isVisible && (
        <p className="toggle-paragraph">
          This is a paragraph of text that can be toggled on and off.
        </p>
      )}
    </div>
  );
};

export default ToggleText;
```

Output:


Toggle Paragraph Visibility
Hide Text
This is a paragraph of text that can be toggled on and off.