

pls generate the relavant codes for these lab programs pls make sure it is in a form that a complete begennir can install and run these code in any system make as in detail as possible explain each step of making it

List of Experiments:

- 1.Create a simple react application (Hello world)
- 2.Create basic calculator application using react js
- 3.Build a music store application using react components.
- 4.Create simple web server application using Node js
- 5.Create a basic Express website using Node js
- 6.Design a simple user login system using Node js
- 7.Design Node blog system
- 8.Design book store system
- 9.Design a Portfolio App
- 10.Design a Simple Online Quiz System
- 11.Create a Job Listing Site
- 12.Build a Product Inventory System in Node.js + MySQL
- 13.Build a Chat Application using React and Socket.io
- 14.Create a Student Result Management App
- 15.Design Node E learning system

Certainly! Below you'll find step-by-step guides—with code, installation, and explanations suitable for complete beginners—for several of the listed lab programs (React and Node.js). Each program will include the full setup and coding steps, written for clarity. Due to space, I'll explain the first few in depth, and outline the approach for remaining ones. Just ask for any specific code samples if you want expanded detail for any additional experiment.

1. Create a Simple React Application (Hello World)

Step 1: Install Node.js

- Download Node.js from <https://nodejs.org> and install it. This includes the "npm" package manager.

Step 2: Create a React App

Open your terminal and run:

```
npx create-react-app hello-world-app
```

- This creates a new folder called `hello-world-app` with all React files.

Step 3: Start Your App

```
cd hello-world-app  
npm start
```

- `npm start` runs the React development server. Your browser should open to <http://localhost:3000>.

Step 4: Edit to Show "Hello World"

Open `src/App.js` in your code editor (Notepad, VS Code, etc). Replace the content with:

```
function App() {  
  return (  
    <div>  
      <h1>Hello World</h1>  
    </div>  
  );  
}  
  
export default App;
```

Save the file. The browser will update to show "Hello World".

2. Create a Basic Calculator Application Using React JS

Step 1: Setup React App

Follow the steps above to create a React app called `react-calculator`.

Step 2: Edit the App

Replace the code in `src/App.js` with a simple calculator:

```
import React, { useState } from "react";  
  
function App() {  
  const [num1, setNum1] = useState(0);  
  const [num2, setNum2] = useState(0);
```

```

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

const handleSum = () => setResult(Number(num1) + Number(num2));
const handleSub = () => setResult(Number(num1) - Number(num2));
const handleMul = () => setResult(Number(num1) * Number(num2));
const handleDiv = () => setResult(Number(num1) / Number(num2));

return (
  <div>
    <h1>Simple Calculator</h1>
    <input type="number" value={num1} onChange={e => setNum1(e.target.value)} placeholder="Num 1" />
    <input type="number" value={num2} onChange={e => setNum2(e.target.value)} placeholder="Num 2" />
    <div>
      <button onClick={handleSum}>Add</button>
      <button onClick={handleSub}>Subtract</button>
      <button onClick={handleMul}>Multiply</button>
      <button onClick={handleDiv}>Divide</button>
    </div>
    <h2>Result: {result}</h2>
  </div>
);
}

export default App;

```

Save, the app now works as a calculator.

3. Build a Music Store Application Using React Components

Step 1: Setup Project

Create your app:

```

npx create-react-app music-store
cd music-store

```

Step 2: Add Music Store Components

Replace `src/App.js` with this code:

```

import React, { useState } from "react";

const albums = [
  { id: 1, title: "Album One", artist: "Artist A", price: 10 },
  { id: 2, title: "Album Two", artist: "Artist B", price: 12 },
  { id: 3, title: "Album Three", artist: "Artist C", price: 9 }
];

function App() {
  const [cart, setCart] = useState([]);

```

```

const addToCart = album => setCart([...cart, album]);

return (
  <div>
    <h1>Music Store</h1>
    <h2>Albums</h2>
    {albums.map(album => (
      <div key={album.id}>
        <span>{album.title} by {album.artist} - ${album.price}</span>
        <button onClick={() => addToCart(album)}>Add to Cart</button>
      </div>
    ))}
    <h2>Cart</h2>
    <ul>
      {cart.map((album, idx) => (
        <li key={idx}>{album.title} - ${album.price}</li>
      ))}
    </ul>
  </div>
);
}

export default App;

```

Run `npm start` to see the Music Store in action.

4. Create Simple Web Server Application Using Node.js

Step 1: Install Node.js

- Node.js must be installed as above.

Step 2: Create the App

Create a folder, e.g. `node-server`, and inside it create a file called `server.js`.

```

// server.js
const http = require("http");

const server = http.createServer((req, res) => {
  res.writeHead(200, {"Content-Type": "text/plain"});
  res.end("Hello from Node.js server!");
});

server.listen(3000, () => {
  console.log("Server is running at http://localhost:3000");
});

```

Step 3: Run the Server

In terminal, navigate to the folder, then run:

```
node server.js
```

Open your browser and go to <http://localhost:3000>.

5. Create a Basic Express Website Using Node.js

Step 1: Install Express

1. Create a folder, e.g. `express-site`.
2. Run in terminal:

```
npm init -y  
npm install express
```

Step 2: `app.js` Code

Create a file called `app.js`:

```
const express = require('express');  
const app = express();  
  
app.get('/', (req, res) => {  
  res.send('Welcome to Express!');  
});  
  
app.listen(3000, () => {  
  console.log('Express server running on http://localhost:3000');  
});
```

Step 3: Run the Site

```
node app.js
```

Open your browser at <http://localhost:3000> to see your message.

Next Steps for Other Projects

For each project (login system, blogs, book store, etc.), the steps are similar:

1. **Set up Node.js/React and any necessary dependencies.**
2. **Create your main server file (Node/Express) or main app file (React).**
3. **Add the basic logic for the app.**

4. Run and test accordingly.

Let me know any specific program you'd like a complete beginner's walkthrough for (from 6 to 15, for example: Login System, Blog System, Book Store, Portfolio App, etc.) and I'll provide a detailed code and explanation for that experiment!