

pls continue

Certainly! Continuing from where the previous answer left off, here are detailed, step-by-step guides for the next set of beginner-friendly Node.js projects (experiments 6–10). Each includes install instructions, code samples, and explanations suitable for a complete beginner.

6. Design a Simple User Login System Using Node.js

Step 1: Set Up Your Project

```
mkdir node-login-system
cd node-login-system
npm init -y
npm install express body-parser
```

Step 2: Create app.js

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();

app.use(bodyParser.urlencoded({ extended: true }));

// Dummy user for demonstration
const user = { username: 'admin', password: 'admin123' };

// Login form
app.get('/', (req, res) => {
  res.send(`<form method="post" action="/login">
    <input name="username" type="text" placeholder="Username"/>
    <input name="password" type="password" placeholder="Password"/>
    <button type="submit">Login</button>
  </form>`);
});

// Handle login
app.post('/login', (req, res) => {
  const { username, password } = req.body;
  if (username === user.username && password === user.password) {
    res.send('Login successful! Welcome.');
```

```
console.log('Login app running on http://localhost:3000');
});
```

Step 3: Run It

```
node app.js
```

Visit <http://localhost:3000> and try logging in with username: admin, password: admin123.

7. Design Node Blog System (Basic, No Database)

Step 1: Set Up Project

```
mkdir node-blog
cd node-blog
npm init -y
npm install express body-parser
```

Step 2: app.js Code

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();

let posts = [];

app.use(bodyParser.urlencoded({ extended: true }));

// Display posts and a form
app.get('/', (req, res) => {
  let postList = posts.map(p => `

## ${p.title}</h2><p>${p.content}</p>`).join('<hr>'); res.send(`${postList}<form method="post" action="/add"> <input name="title" placeholder="Title"><br/> <textarea name="content" placeholder="Content"></textarea><br/> <button type="submit">Add Post</button> </form>`); }); // Add post app.post('/add', (req, res) => { posts.push({ title: req.body.title, content: req.body.content }); res.redirect('/'); }); app.listen(3000, () => { console.log('Blog app running on http://localhost:3000'); });


```

8. Design Book Store System (Basic Demo)

Step 1: Setup

```
mkdir book-store
cd book-store
npm init -y
npm install express body-parser
```

Step 2: app.js Code

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();

let books = [
  { title: "The Hobbit", author: "J.R.R. Tolkien", price: 15 },
  { title: "1984", author: "George Orwell", price: 12 }
];

app.use(bodyParser.urlencoded({ extended: true }));

app.get('/', (req, res) => {
  let bookList = books.map(b => `<li>${b.title} by ${b.author} - ${b.price}</li>`).join('');
  res.send(`<h1>Book Store</h1><ul>${bookList}</ul>
    <form method="post" action="/add">
      <input name="title" placeholder="Title"/>
      <input name="author" placeholder="Author"/>
      <input name="price" placeholder="Price" type="number"/>
      <button type="submit">Add Book</button>
    </form>`);
});

app.post('/add', (req, res) => {
  books.push({ title: req.body.title, author: req.body.author, price: req.body.price });
  res.redirect('/');
});

app.listen(3000, () => {
  console.log('Book store running on http://localhost:3000');
});
```

9. Design a Portfolio App (React)

Step 1: Create React App

```
npx create-react-app my-portfolio
cd my-portfolio
```

Step 2: Edit `src/App.js`

```
function App() {
  return (
    <div>
      <h1>Your Name</h1>
      <h2>Portfolio</h2>
      <ul>
        <li>Project 1: Portfolio Website</li>
        <li>Project 2: Calculator App</li>
        <li>Project 3: Blog Platform</li>
      </ul>
      <p>Contact: youremail@example.com</p>
    </div>
  );
}

export default App;
```

Run with:

```
npm start
```

10. Design a Simple Online Quiz System (Basic, Node.js)

Step 1: Setup

```
mkdir quiz-app
cd quiz-app
npm init -y
npm install express body-parser
```

Step 2: `app.js` Code

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();

const questions = [
  { q: "2 + 2?", a: "4" },
  { q: "Capital of France?", a: "Paris" }
];

app.use(bodyParser.urlencoded({ extended: true }));

app.get('/', (req, res) => {
  let quiz = questions.map((item, idx) =>
    `<label>${item.q}<input name="q${idx}" /></label><br/>`).join('');
  res.send(`<form method="post" action="/submit">${quiz}<button type="submit">Submit</button>`);
});
```

```
app.post('/submit', (req, res) => {
  let score = questions.reduce((acc, q, idx) => acc + (req.body[`q${idx}`] == q.a ? 1 : 0), 0);
  res.send(`Score: ${score}/${questions.length}`);
});

app.listen(3000, () => {
  console.log('Quiz app running on http://localhost:3000');
});
```

You can run any of these by saving the code as `app.js` and following the install instructions. If you'd like the remaining experiments (11–15) explained step-by-step, just reply with "continue" or specify a number for deeper detail!