

Task 1 : HTML Structure and Basic Server Interaction

Objective : Introduce the concept of server-side rendering and basic form submissions.

Steps 1: Create an HTML structure with forms for user input.

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Form Submission</title>

</head>

<body>

    <h2>User Form</h2>

    <form action="/submit" method="POST">

        <label for="name">Name:</label>

        <input type="text" id="name" name="name"
required>

        <button type="submit">Submit</button>

    </form>

</body>

</html>
```

Step 2 : Set up a simple Node.js server using Express.

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

const app = express();
app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));

app.use(bodyParser.urlencoded({ extended: true }));
// Serve the form
app.get('/', (req, res) => {
  res.render('index');
});
// Handle form submission
app.post('/submit', (req, res) => {
  const { name } = req.body;
  res.send(<h2>Form submitted! Hello, ${name}</h2>);
});
// Start the server
app.listen(3000, () => {
  console.log('Server is running on http://localhost:3000');
});
```

Step 3 : Create server-side endpoints to handle form submissions.

```
npm init -y
```

```
npm install express body-parser ejs
```

Step 4 : Use server-side rendering (e.g., EJS) to dynamically generate HTML.

```
node server.js
```