

Ex: No: 12

Bind server and client side and deploy as a deliverable application

Project setup:

Step 1. Building the frontend React application

```
npx create-react-app client
```

Step 2: Building the Backend Node.js server

Open a new terminal and create a new directory called server and initialize a Node.js project in it. Let's install the required dependencies for our application.

```
npm install express cors
```

For development purposes, we will install nodemon as a development dependency.

```
npm install nodemon --save-dev
```

Lastly, we will add a start script in the package.json file to start the server.

```
"scripts": {  
  "start": "node server.js",  
  "dev": "nodemon server.js"  
}
```

2,a)Creating the server: Start with creating a new file named server.js in the server folder.

i)Import express and cors

```
const express = require('express');  
const cors = require('cors');
```

ii)Create an Express application

```
const app = express();  
  
app.use(cors());
```

iii) Create a GET route

```
app.get('/message', (req, res) => {  
    res.json({ message: "Hello from server!" });  
});
```

iv) Start the server

```
app.listen(8000, () => {  
    console.log(`Server is running on port 8000.`);  
});
```

The final server.js file will look like this:

```
const express = require("express");  
const cors = require("cors");  
const app = express();  
  
app.use(cors());  
app.use(express.json());  
  
app.get("/message", (req, res) => {  
    res.json({ message: "Hello from server!" });  
});  
  
app.listen(8000, () => {  
    console.log(`Server is running on port 8000.`);  
});
```

v) To start the backend server open your terminal and run the following command:

npm run dev

This command will start the server on port 8000.

Step 3: Connect React with Node.js

Open the App.js file in the src folder and then replace the code with the following code.

```
import React, { useState, useEffect } from "react";
import "./App.css";

function App() {
  const [message, setMessage] = useState("");

  useEffect(() => {
    fetch("http://localhost:8000/message")
      .then((res) => res.json())
      .then((data) => setMessage(data.message));
  }, []);

  return (
    <div className="App">
      <h1>{message}</h1>
    </div>
  );
}

export default App
```

Open your browser and visit <http://localhost:3000/> where you will see the message “**Hello from server!**” on the screen.