

Lab 6 - Web Server Creation using NodeJS

- Rohit Arrunachalam R IOT B (21110433)

App.jsx (Frontend)

```
import React, { useEffect, useState } from 'react';
import reactLogo from './assets/react.svg'
import viteLogo from '/vite.svg'
import './App.css'
import axios from 'axios'; // Import Axios
function App() {
  const [message, setMessage] = useState("")
  useEffect(() => {
    axios.get('http://localhost:5000/api/hello') // Use
    Axios to make a GET request
      .then((response) => setMessage(response.data.message))
      .catch((error) => console.error(error));
  }, []);

  return (
    <>
    <h1>{message}</h1>
    </>
  )
}

export default App
```

server.js (Backend)

```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors'); // Import the cors middleware

const app = express();
const port = process.env.PORT || 5000;
```

```
// Define the CORS options
const corsOptions = {
  origin: 'http://127.0.0.1:5173', // Allow requests from
this origin
  methods: 'GET,HEAD,PUT,PATCH,POST,DELETE',
  credentials: true, // Allow cookies to be sent with the
request
  optionsSuccessStatus: 204, // Respond with a 204 status
for preflight requests
};

// Use cors middleware with the specified options
app.use(cors(corsOptions));

// Parse JSON and form data
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

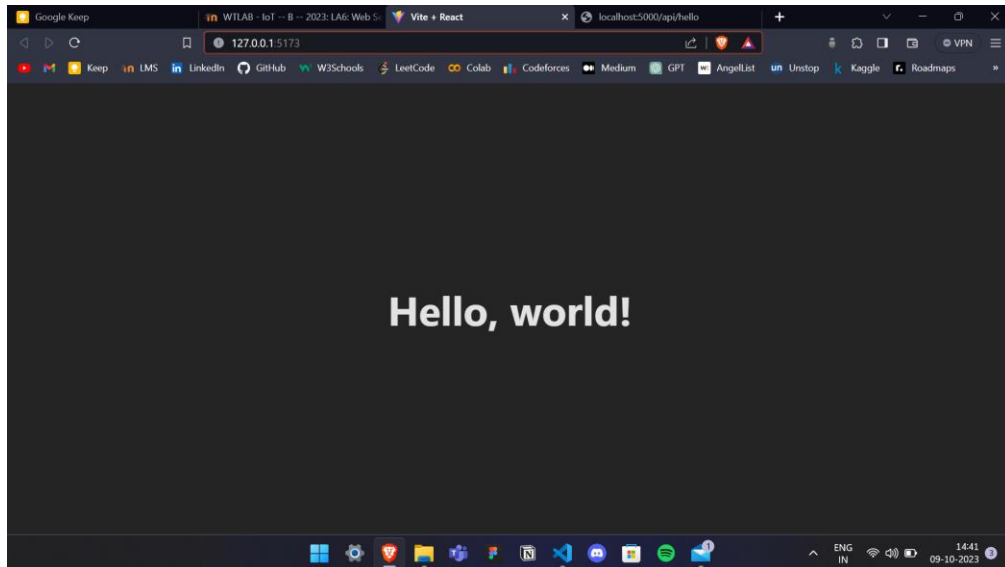
// Serve static files (React frontend)
app.use(express.static('public'));

// Define a basic API route
app.get('/api/hello', (req, res) => {
  res.json({ message: 'Hello, world!' });
});

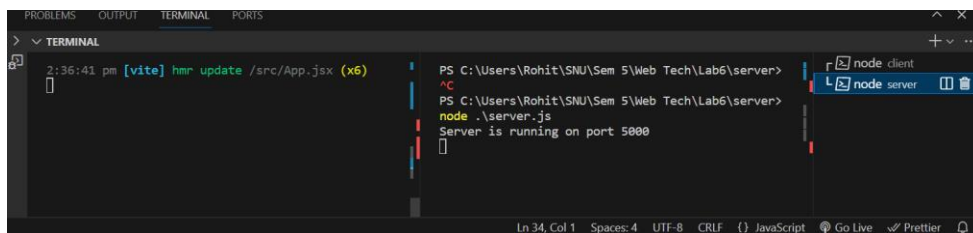
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

Output

Frontend Output



Terminal Output



Backend Output

