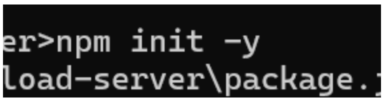
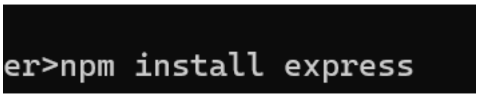
5. Create a Node.js program using Express that allows users to download a file. Your program should have the following endpoint: • /download/:filename: Accepts a GET request with a parameter filename. • The program should read the file with the given filename from the server's file system and send it as a response. • If the file does not exist, respond with a 404 status code and a message "File not found".

Server.js

```
const express = require('express');
const path = require('path');
const fs = require('fs');
const app = express();
const port = 3000;

// Middleware to serve static files if needed (optional)
app.use(express.static('public'));

// Endpoint to download a file
app.get('/download/:filename', (req, res) => {
  const filename = req.params.filename;
  const filePath = path.join(__dirname, 'files', filename); // Assuming files are in the "files" folder

  // Check if the file exists
  fs.exists(filePath, (exists) => {
    if (exists) {
      // If the file exists, send it as a response
      res.download(filePath, filename, (err) => {
        if (err) {
          console.error('Error sending the file:', err);
          res.status(500).send('Error downloading the file');
        }
      });
    } else {
      // If the file does not exist, return 404
      res.status(404).send('File not found');
    }
  });
});

// Start the server
app.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});
```

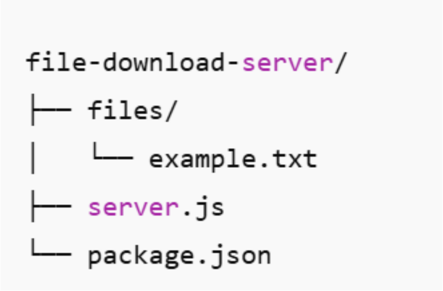


```
er>npm init -y
load-server\package.
```



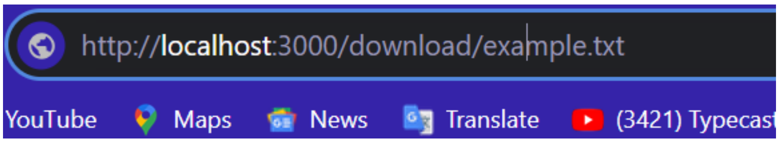
```
er>npm install express
```

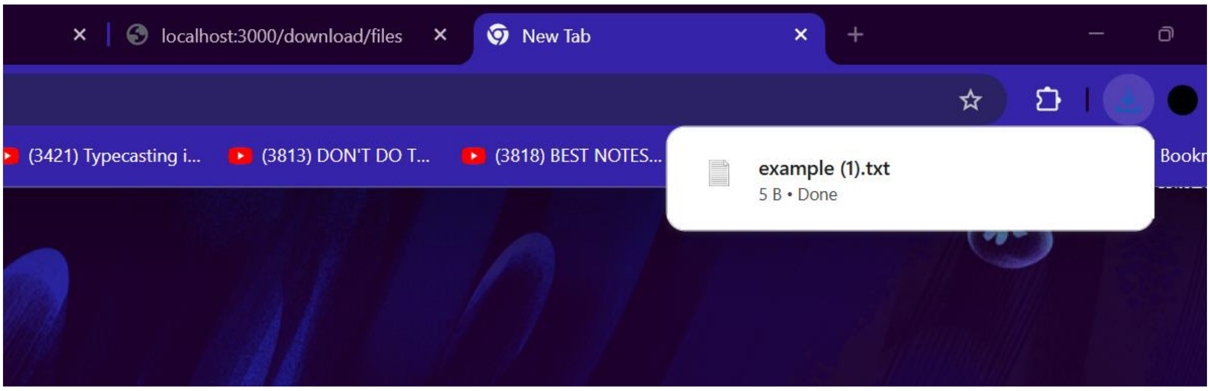
File structure:



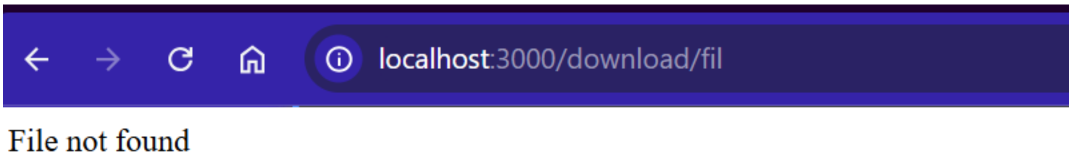
```
file-download-server/
├── files/
│   └── example.txt
├── server.js
└── package.json
```

OUTPUT:


http://localhost:3000/download/example.txt

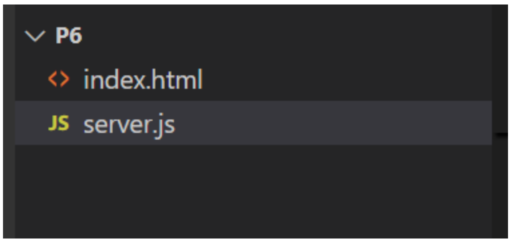


example (1).txt
5 B • Done


localhost:3000/download/fil

File not found

6. Create an HTTP server that handles JSON data sent in a POST request and serves an HTML file.



Server.js

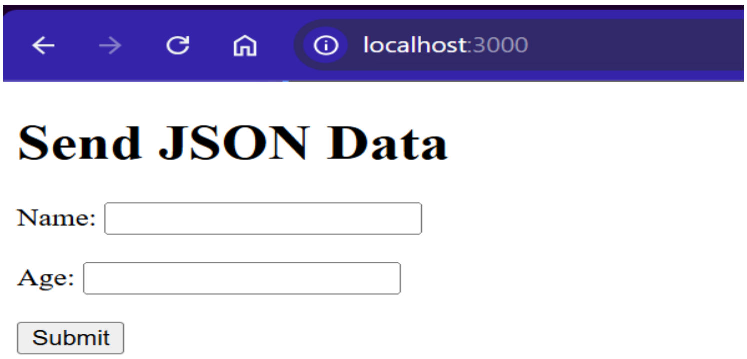
```
const http = require('http');

const fs = require('fs');

const path = require('path');

const url = require('url');


// Create the HTTP server

const server = http.createServer((req, res) => {

  const parsedUrl = url.parse(req.url, true);
```

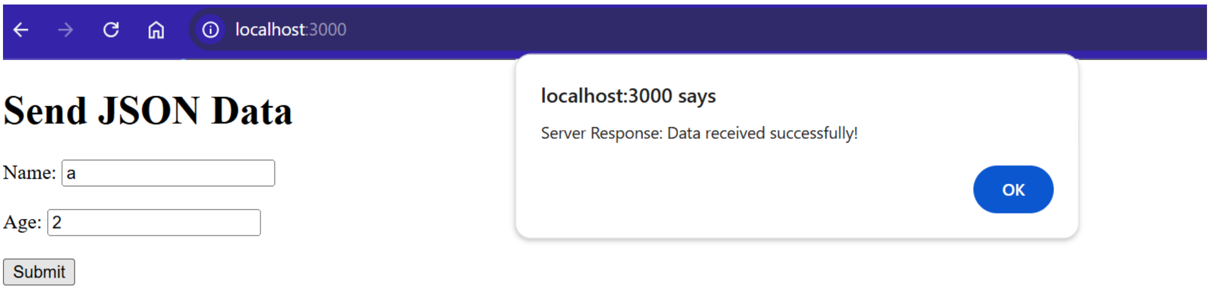
```
// Serve the HTML file for GET requests
if (req.method === 'GET' && parsedUrl.pathname === '/') {
  const filePath = path.join(__dirname, 'index.html'); // Path to the HTML file

  fs.readFile(filePath, (err, data) => {
    if (err) {
      res.statusCode = 500;
      res.end('Error reading the HTML file.');
    } else {
      res.statusCode = 200;
      res.setHeader('Content-Type', 'text/html');
      res.end(data); // Send the HTML file as a response
    }
  });
}

// Handle POST requests with JSON data
else if (req.method === 'POST' && parsedUrl.pathname === '/submit') {
  let body = '';

  // Collect the data sent in the POST request
  req.on('data', chunk => {
    body += chunk;
  });

  // Once the entire request body is received, handle it
  req.on('end', () => {
      try {
        // Parse the JSON data
        const jsonData = JSON.parse(body);

        // Log the received JSON data (you can perform further operations here)
        console.log('Received JSON data:', jsonData);

        // Respond with a success message
        res.statusCode = 200;
        res.setHeader('Content-Type', 'application/json');
        res.end(JSON.stringify({ message: 'Data received successfully!' }));
      } catch (err) {
        // Handle JSON parsing errors
        res.statusCode = 400;
        res.end('Invalid JSON');
      }
    });
  }

  // Handle unsupported routes or methods
  else {
    res.statusCode = 404;
    res.end('Not Found');
  }
});

// Set the server to listen on port 3000
server.listen(3000, () => {
  console.log('Server is listening on http://localhost:3000');
});
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Post JSON Example</title>
</head>
<body>
  <h1>Send JSON Data</h1>
  <form id="jsonForm">
    <label for="name">Name:</label>
    <input type="text" id="name" required><br><br>

    <label for="age">Age:</label>
    <input type="number" id="age" required><br><br>

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

  <script>
    // Handle form submission
    document.getElementById('jsonForm').addEventListener('submit', function (event) {
      event.preventDefault();
      const name = document.getElementById('name').value;
      const age = document.getElementById('age').value;

      // Prepare JSON data
      const jsonData = JSON.stringify({ name: name, age: age });

      // Send POST request with JSON data
      fetch('/submit', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: jsonData,
      })
      .then(response => response.json())
      .then(data => alert('Server Response: ' + data.message))
      .catch(error => alert('Error: ' + error));
    });
  </script>
</body>
</html>
```

OUTPUT:





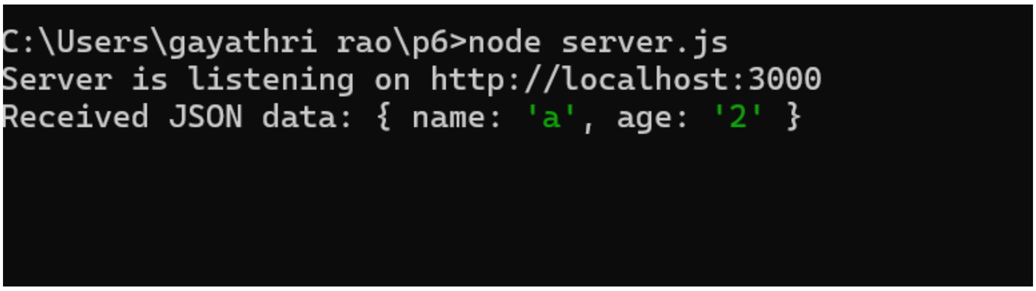
Send JSON Data

Name: a

Age: 2

localhost:3000 says

Server Response: Data received successfully!



```
C:\Users\gayathri rao\p6>node server.js
Server is listening on http://localhost:3000
Received JSON data: { name: 'a', age: '2' }
```