

Experiment-3

Student Name: Pranshu Muthreja
Branch: UIC
Semester: 3
Subject Name: Back End Technologies

UID: 23MCA20164
Section/Group: 3/B
Date of Performance: 24/08/2024
Subject Code: 23CAH-705

1. Aim/Overview of the practical:

Create a sign up page and store the inputted data by user in file

2. Task to be done :

Initialize a sign up page and store the inputted data by the user in file

3. Code for experiment/practical:

```
JS index.js  ×  {} data.json
Experiment-3 > JS index.js > ...
 1  const express = require('express');
 2  const bodyParser = require('body-parser');
 3  const fs = require('fs');
 4  const path = require('path');
 5
 6  const app = express();
 7  const PORT = 3000;
 8
 9  app.use(bodyParser.urlencoded({ extended: true }));
10  app.use(express.static('public'));
11
12  const dataFilePath = path.join(__dirname, 'data.json');
13
14  // Check if data.json exists, and if not, create it with an empty array
15  function initializeDataFile() {
16      if (!fs.existsSync(dataFilePath)) {
17          fs.writeFileSync(dataFilePath, '[]', 'utf8');
18      }
19  }
20
21  // Read employee data from data.json
22  function readData() {
23      const data = fs.readFileSync(dataFilePath, 'utf8');
24      return JSON.parse(data);
25  }
26
27  // Write employee data to data.json
28  function writeData(data) {
29      fs.writeFileSync(dataFilePath, JSON.stringify(data, null, 2), 'utf8');
30  }
```

Experiment-3 > JS index.js > ...

```

32 // Serve the HTML file
33 app.get('/', (req, res) => {
34   res.send(`
35     <!DOCTYPE html>
36     <html lang="en">
37     <head>
38       <meta charset="UTF-8">
39       <meta name="viewport" content="width=device-width, initial-scale=1.0">
40       <title>Employee Signup</title>
41       <style>
42         body {
43           font-family: Arial, sans-serif;
44           background-color: #f0f4f7;
45           margin: 0;
46           padding: 0;
47           display: flex;
48           justify-content: center;
49           align-items: center;
50           height: 100vh;
51         }
52
53         .signup-container {
54           background-color: #fff;
55           padding: 30px;
56           border-radius: 10px;
57           box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);
58           width: 100%;
59           max-width: 500px;
60           text-align: center;
61         }

```

Experiment-3 > JS index.js > app.get('/getEmployee') callback

```

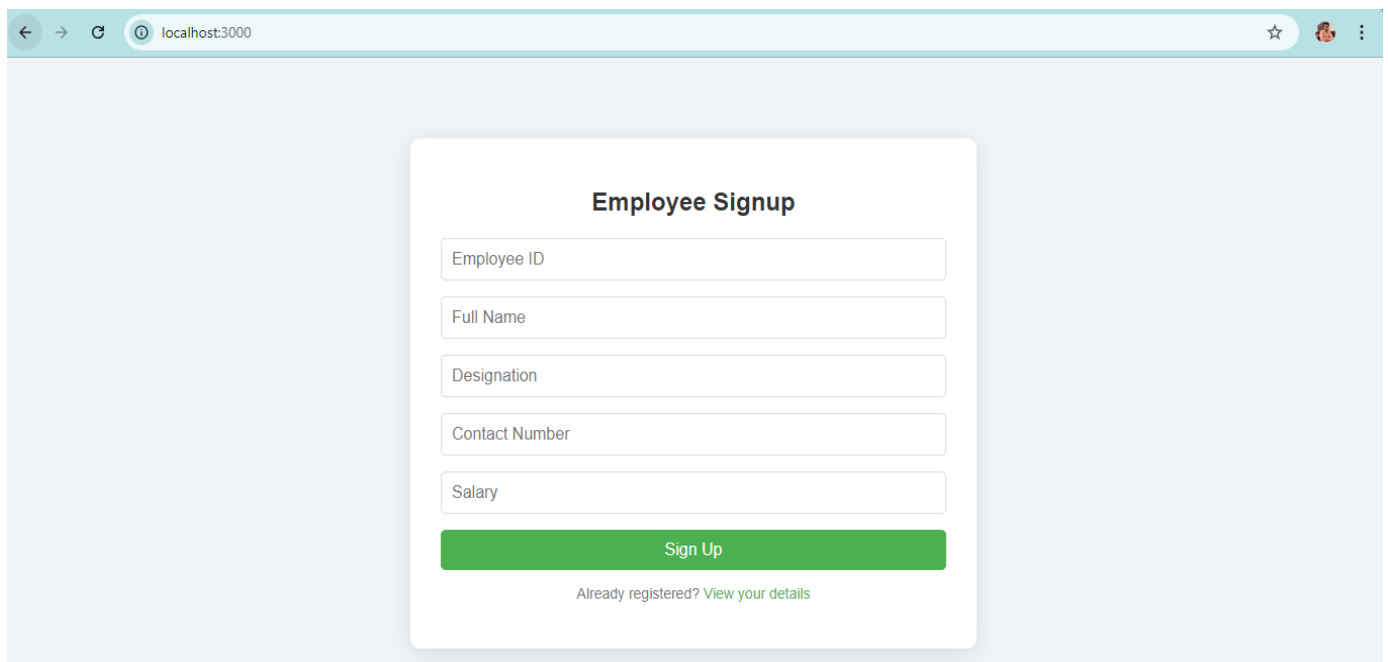
130 // Handle form submission to add an employee
131 app.post('/addEmployee', (req, res) => {
132   initializeDataFile(); // Ensure the data.json file is created
133   const { id, name, designation, contact, salary } = req.body;
134   let employees = readData();
135
136   const newEmployee = { id, name, designation, contact, salary };
137   employees.push(newEmployee);
138
139   writeData(employees);
140   res.send(`Employee ${name} added successfully!`);
141 });
142
143 // Handle fetching employee details by ID
144 app.get('/getEmployee', (req, res) => {
145   initializeDataFile(); // Ensure the data.json file is created
146   const { id } = req.query;
147   const employees = readData();
148
149   const employee = employees.find(emp => emp.id === id);
150
151   if (employee) {
152     res.send(`
153       <!DOCTYPE html>
154       <html lang="en">
155       <head>
156         <meta charset="UTF-8">
157         <meta name="viewport" content="width=device-width, initial-scale=1.0">
158         <title>Employee Details</title>
159         <style>
160           body {
161             font-family: Arial, sans-serif;

```

Experiment-3 > JS index.js > app.get('/getEmployee') callback

```
144 app.get('/getEmployee', (req, res) => {
216     <body>
217         <div class="container">
218             <h1>Employee Details</h1>
219             <div class="employee-details">
220                 <h2>${employee.name}</h2>
221                 <p><strong>ID:</strong> ${employee.id}</p>
222                 <p><strong>Designation:</strong> ${employee.designation}</p>
223                 <p><strong>Contact:</strong> ${employee.contact}</p>
224                 <p><strong>Salary:</strong> ${employee.salary}</p>
225             </div>
226             <a href="/">Go Back</a>
227         </div>
228     </body>
229 </html>
230 `);
231 } else {
232     res.send(`
233     <!DOCTYPE html>
234     <html lang="en">
235     <head>
236         <meta charset="UTF-8">
237         <meta name="viewport" content="width=device-width, initial-scale=1.0">
238         <title>Employee Not Found</title>
239         <style>
240             body {
241                 font-family: Arial, sans-serif;
242                 background-color: #f4f4f4;
243                 margin: 0;
244                 padding: 0;
245                 display: flex;
```

4. Result/Output/Writing Summary:



localhost:3000

Employee Signup

Employee ID

Full Name

Designation

Contact Number

Salary

Sign Up

Already registered? [View your details](#)



Learning outcomes (What I have learnt):

1. I have learnt to set up and configure routes for handling HTTP requests and serving files.
2. I have learnt to implement form validation to ensure required fields are filled out correctly.
3. I have learnt to interact with the file system using Node.js fs module.
4. I have learnt to organize code and project structure for maintainability and scalability.
5. I have gained experience in running and debugging a Node.js server application.