EXPERIMENT NO.2

Student Name: Dibya Das
UID: 23MCA20052
Branch: MCA
Section/Group: 23MCA6-A
Semester: 3rd
Date of Performance: 01/08/24
Subject: Back End Technologies
Subject Code:23CAH 705

- **Aim/Overview of the practical:**
  Understand the architecture and design principles of back-end systems, including APIs, web services, and serverless computing
- **Task to be done:**
  To store the data from our previous experiment (employee salary program) in a file and then use that file in another program, we can follow these steps:

1. **Store Data in a JSON File**: Ensure that our main program writes the data to a JSON file (employees.json).
2. **Read Data from JSON File in Another Program**: Create a new Node.js program that reads the data from the JSON file and performs some operations on it.

- **Concept used:**
  Node.js, HTML, CSS, Javascript, JSON file

- **Steps/Commands involved to perform practical:**

1. **employee.js**

```
var http = require('http');
var fs = require('fs');
var path = require('path');
var querystring = require('querystring');

// Load employees from file
const loadEmployees = () => {
  try {
    const data = fs.readFileSync('employees.json', 'utf8');
    return JSON.parse(data);
  } catch (error) {
```

```
    if (error.code === 'ENOENT') {
      console.error('Error: employees.json file not found');
    } else if (error instanceof SyntaxError) {
      console.error('Error: Invalid JSON in employees.json');
    } else {
      console.error('Error reading employees.json:', error);
    }
    return [];
  }
};

// Save employees to file
const saveEmployees = (employees) => {
  try {
    fs.writeFileSync('employees.json', JSON.stringify(employees, null, 2),
'utf8');
  } catch (error) {
    console.error('Error writing employees.json:', error);
  }
};

// Generate HTML content for employee table
const generateEmployeeTable = (employees) => {
  let tableContent = '<table
border="1"><tr><th>ID</th><th>Name</th><th>Email</th><th>Phone</th><th>Sal
ary</th><th>Action</th></tr>';
  employees.forEach(emp => {
    tableContent += `
      <tr>
        <td>${emp.id}</td>
        <td>${emp.name}</td>
        <td>${emp.email}</td>
        <td>${emp.phone}</td>
        <td>${emp.salary}</td>
        <td>
          <form method="POST" action="/update" style="display:inline;">
            <input type="hidden" name="id" value="${emp.id}">
            <input type="number" name="salary" value="${emp.salary}"
placeholder="New Salary" required>
            <button type="submit">Update</button>
          </form>
          <form method="POST" action="/delete" style="display:inline;">
            <input type="hidden" name="id" value="${emp.id}">
            <button type="submit">Delete</button>
          </form>
        </td>
      </tr>`;
  });
  tableContent += '</table>';
  return tableContent;
};
```

```
http.createServer(function(req, res) {
    if (req.method === 'GET') {
        if (req.url === '/') {
            res.writeHead(200, {'Content-Type': 'text/html'});
            res.end(`
                <html>
                    <head>
                        <title>Employee Portal</title>
                    </head>
                    <body>
                        <h1>Welcome to Employee Portal!</h1>
                        <button
onclick="redirectToEmployeeTable()">Employee Salary Table</button>
                        <button onclick="redirectToAddEmployee()">Add New
Employee</button>
                        <script>
                            function redirectToEmployeeTable() {
                                window.location.href = '/employee';
                            }
                            function redirectToAddEmployee() {
                                window.location.href = '/add_employee';
                            }
                        </script>
                    </body>
                </html>
            `);
        } else if (req.url === '/employee') {
            const employees = loadEmployees();
            const employeeTable = generateEmployeeTable(employees);

            res.writeHead(200, {'Content-Type': 'text/html'});
            res.end(`
                <html>
                    <head>
                        <title>Employee Salary Table</title>
                    </head>
                    <body>
                        <h1>Employee Salary Table</h1>
                        ${employeeTable}
                        <button onclick="redirectToAddEmployee()">Add New
Employee</button>
                        <script>
                            function redirectToAddEmployee() {
                                window.location.href = '/add_employee';
                            }
                        </script>
                    </body>
                </html>
            `);
        } else if (req.url === '/add_employee') {
            fs.readFile(path.join(__dirname, 'add_employee.html'),
function(err, data) {
```

```
                    if (err) {
                        res.writeHead(404, {'Content-Type': 'text/html'});
                        res.end('404 Not Found');
                    } else {
                        res.writeHead(200, {'Content-Type': 'text/html'});
                        res.end(data);
                    }
                });
            } else {
                res.writeHead(404, {'Content-Type': 'text/html'});
                res.end('<h1>404 Not Found</h1>');
            }
        } else if (req.method === 'POST') {
            let body = '';
            req.on('data', chunk => {
                body += chunk.toString();
            });
            req.on('end', () => {
                const parsedBody = querystring.parse(body);

                if (req.url === '/update') {
                    const id = parseInt(parsedBody.id);
                    const newSalary = parseFloat(parsedBody.salary);
                    const employees = loadEmployees();

                    const employee = employees.find(emp => emp.id === id);
                    if (employee) {
                        employee.salary = newSalary;
                        saveEmployees(employees);
                        res.writeHead(302, { 'Location': '/employee' });
                        res.end();
                    } else {
                        res.writeHead(404, {'Content-Type': 'text/html'});
                        res.end('<h1>Employee not found</h1>');
                    }
                } else if (req.url === '/delete') {
                    const id = parseInt(parsedBody.id);
                    let employees = loadEmployees();

                    employees = employees.filter(emp => emp.id !== id);
                    saveEmployees(employees);
                    res.writeHead(302, { 'Location': '/employee' });
                    res.end();
                } else if (req.url === '/add') {
                    const name = parsedBody.name;
                    const email = parsedBody.email;
                    const phone = parsedBody.phone;
                    const salary = parseFloat(parsedBody.salary);
                    const employees = loadEmployees();

                    const newId = employees.length ?
Math.max(...employees.map(emp => emp.id)) + 1 : 1;
```

```
                employees.push({ id: newId, name, email, phone, salary });
                saveEmployees(employees);
                res.writeHead(302, { 'Location': '/employee' });
                res.end();
            } else {
                res.writeHead(404, {'Content-Type': 'text/html'});
                res.end('<h1>404 Not Found</h1>');
            }
        });
    } else {
        res.writeHead(404, {'Content-Type': 'text/html'});
        res.end('<h1>404 Not Found</h1>');
    }
}).listen(8081);

console.log('Server running at http://127.0.0.1:8081/');
```

2. employees.json

```
[
  {
    "id": 1,
    "name": "Alice Johnson",
    "email": "qwe@gmail.com",
    "phone": 1234567890,
    "salary": 5000
  },
  {
    "id": 2,
    "name": "Bob Smith",
    "email": "qwe@gmail.com",
    "phone": 1234567890,
    "salary": 60004
  },
  {
    "id": 3,
    "name": "Charlie Brown",
    "email": "qwe@gmail.com",
    "phone": 1234567890,
    "salary": 55004
  },
  {
    "id": 4,
    "name": "Diana Prince",
    "email": "qwe@gmail.com",
    "phone": 1234567890,
    "salary": 70004
  },
```

```
  {
    "id": 5,
    "name": "Dibya Das",
    "email": "qwe@gmail.com",
    "phone": 1234567890,
    "salary": 50000
  },
  {
    "id": 6,
    "name": "henry white",
    "email": "1234@asd",
    "phone": "7896454123",
    "salary": 90000
  }
]
```


add-employee.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add New Employee</title>
</head>
<body>
    <h1>Add New Employee</h1>
    <form method="POST" action="/add">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required><br><br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required><br><br>
        <label for="phone">Phone:</label>
        <input type="text" id="phone" name="phone" required><br><br>
        <label for="salary">Salary:</label>
        <input type="number" id="salary" name="salary" required><br><br>
        <button type="submit">Add Employee</button>
    </form>
    <button onclick="goBack()">Go Back</button>
    <script>
        function goBack() {
            window.location.href = '/employee';
        }
```

```
    </script>
</body>
</html>
```

- **Output:**

Home page

Welcome to Employee Portal!

Employee Salary Table | Add New Employee

Adding new employee page

Add New Employee

Name: tamm Rose

Email: tftyf@cghch

Phone: 78894561230

Salary: 50000

Add Employee

Go Back

Employee Salary Table

ID	Name	Email	Phone	Salary	Action
1	Alice Johnson	qwe@gmail.com	1234567890	50006	50006 Update Delete
2	Bob Smith	qwe@gmail.com	1234567890	60004	60004 Update Delete
3	Charlie Brown	qwe@gmail.com	1234567890	55004	55004 Update Delete
4	Diana Prince	qwe@gmail.com	1234567890	70004	70004 Update Delete
5	Dibya Das	qwe@gmail.com	1234567890	50000	50000 Update Delete
6	henry white	1234@asd	7896454123	90000	90000 Update Delete
7	tamm Rose	tftyf@cghch	78894561230	50000	50000 Update Delete

Add New Employee

Update record

Employee Salary Table

ID	Name	Email	Phone	Salary	Action
1	Alice Johnson	qwe@gmail.com	1234567890	5000	5000 Update Delete
2	Bob Smith	qwe@gmail.com	1234567890	60004	60004 Update Delete
3	Charlie Brown	qwe@gmail.com	1234567890	55004	55004 Update Delete
4	Diana Prince	qwe@gmail.com	1234567890	70004	70004 Update Delete
5	Dibya Das	qwe@gmail.com	1234567890	50000	50000 Update Delete
6	henry white	1234@asd	7896454123	90000	90000 Update Delete
7	tamm Rose	tftyf@cghch	78894561230	50000	50000 Update Delete

Add New Employee

Delete record

Employee Salary Table

ID	Name	Email	Phone	Salary	Action
1	Alice Johnson	qwe@gmail.com	1234567890	5000	5000 Update Delete
2	Bob Smith	qwe@gmail.com	1234567890	60004	60004 Update Delete
3	Charlie Brown	qwe@gmail.com	1234567890	55004	55004 Update Delete
4	Diana Prince	qwe@gmail.com	1234567890	70004	70004 Update Delete
5	Dibya Das	qwe@gmail.com	1234567890	50000	50000 Update Delete
6	henry white	1234@asd	7896454123	90000	90000 Update Delete

Add New Employee

- **Learning Outcome :-**

1. Learnt about the concept of Node.js
2. Learnt to create a clickable button that directs us to another url.
3. CRUD operations on employee table
4. Manipulation on json file.