

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

Aim : You are building a Node.js application for a scheduling platform. As part of the functionality, you need to display the current date and time whenever a user interacts with the platform.

Practical 4 : Create REST API using NodeJS & Apply REST method to perform CRUD operations on resources at server regarding universities or Industries scenario.

1. Create a Rest API for universities or industry using NodeJS on any IDE
2. Perform CRUD operation on server using Postman.
3. Integrate it with HTML form, where it provides the option to POST the data of new employee, Get information of any employee based on ID, Get information of all employees, Update information of any employee based on their id and delete employee record based on their id

1. Code (without html) :

```
const express = require("express");
const bodyParser = require("body-parser");
const cors = require("cors");
const app = express();
app.use(cors());

app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

let port = 4200;
let emps = [
{
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
id: "1",
name: "Madhav",
email: "madhav@haribol.com",
phone: "1234567890",
designation: "CEO",
},
{
id: "2",
name: "Gopal",
email: "gopal@haribol.com",
phone: "0123456789",
designation: "HR",
},
{
id: "3",
name: "Damodar",
email: "damodar@haribol.com",
phone: "9876543210",
designation: "Finance",
},
{
id: "4",
name: "Keshav",
email: "keshav@haribol.com",
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
phone: "9630147852",  
designation: "HR",  
},  
];  
  
app.get("/", (req, res) => {  
res.send("Welcome to YSL Company REST API!");  
});  
  
app.get("/ysl/employees", (req, res) => {  
res.send(emps);  
});  
  
app.get("/ysl/employees/:id", (req, res) => {  
const emp = emps.find(({ id }) => id === req.params.id);  
  
if (!emp) res.status(404).send("Not found!");  
res.send(emp);  
});  
  
app.post("/ysl/employees", (req, res) => {  
let emp = {  
id: req.body.id,  
name: req.body.name,
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
email: req.body.email,  
phone: req.body.phone,  
dept: req.body.dept,  
};  
emps.push(emp);  
res.send(emp);  
});  
  
app.put("/ysl/employees/:id", (req, res) => {  
  let emp = emps.find(({ id }) => id === req.params.id);  
  if (!emp) res.status(404).send("Not found!");  
  emp.id = req.body.id;  
  emp.name = req.body.name;  
  emp.email = req.body.email;  
  emp.ln = req.body.ln;  
  emp.phone = req.body.phone;  
  emp.designation = req.body.designation;  
  res.send(emp);  
});  
  
app.delete("/ysl/employees/:id", (req, res) => {  
  const emp = emps.find(({ id }) => id === req.params.id);  
  if (!emp) res.status(404).send("Not found!");
```

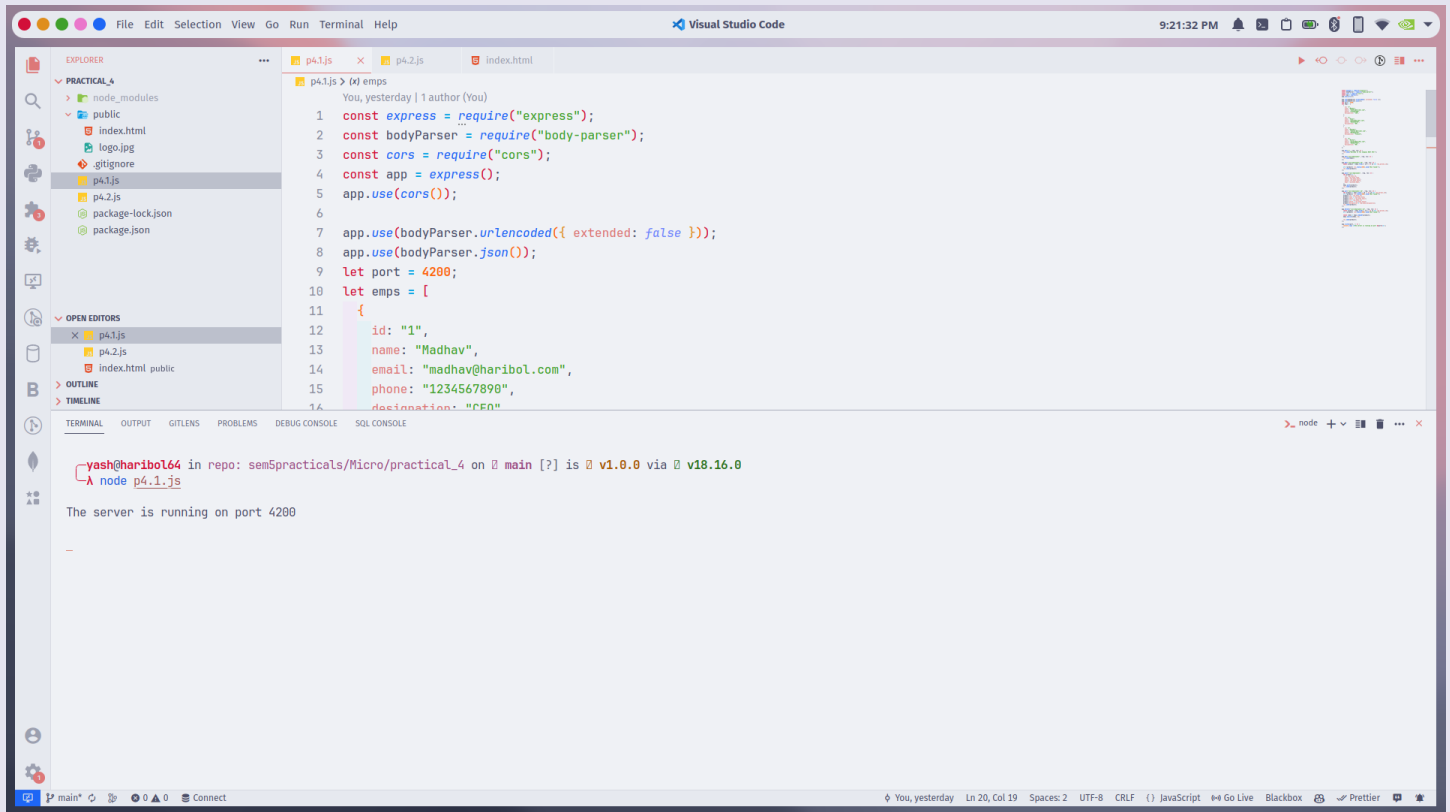
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
const index = emps.indexOf(emp);  
emps.splice(index, 1);  
  
res.send(emp);  
});  
  
app.listen(port, () => {  
  console.log(`\nThe server is running on port ${port}\n`);  
});
```

2. Screenshots (CRUD using Postman) :

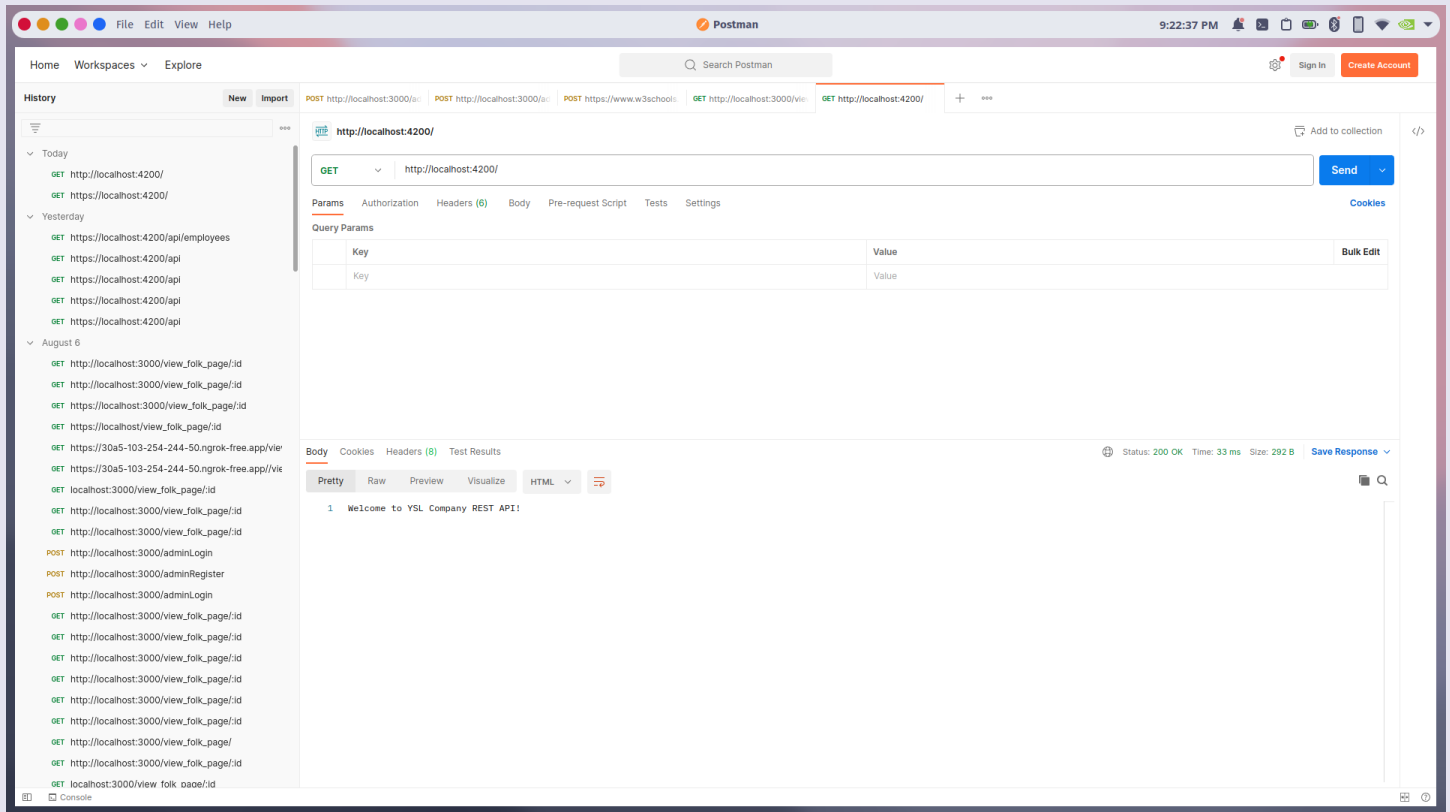
- Running local server

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4



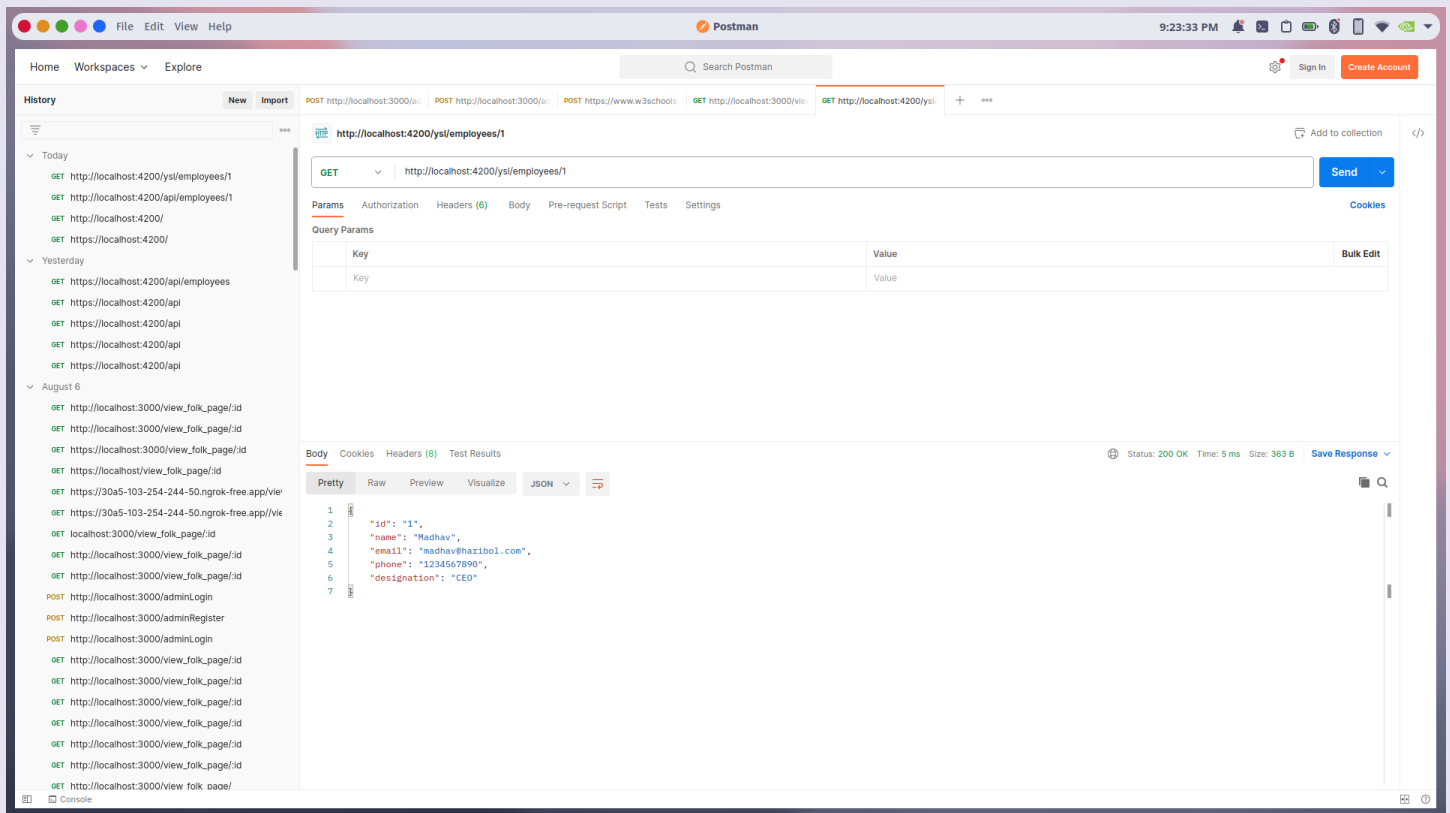
- Homepage of local server

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4



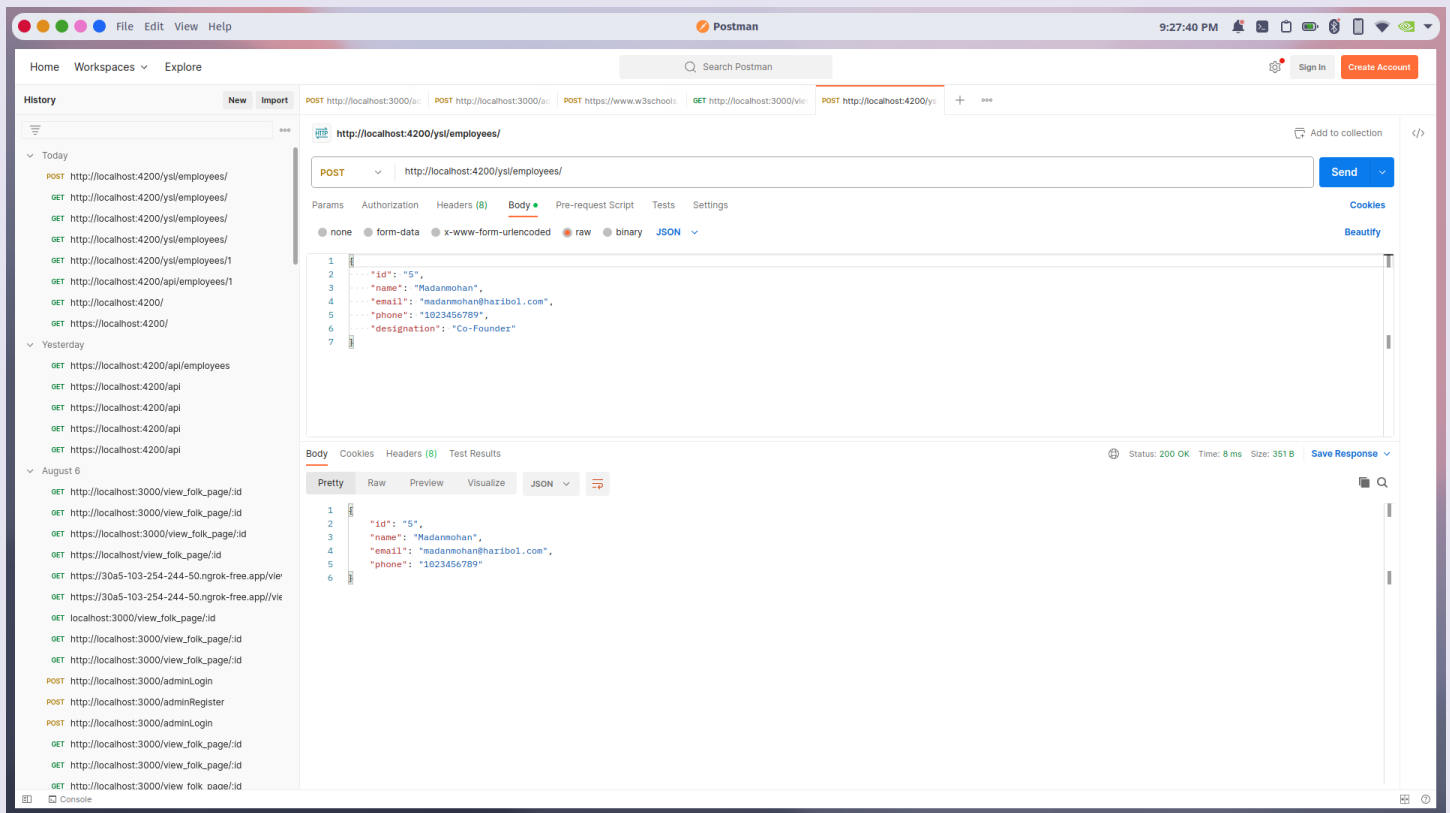
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

- GET details by ID



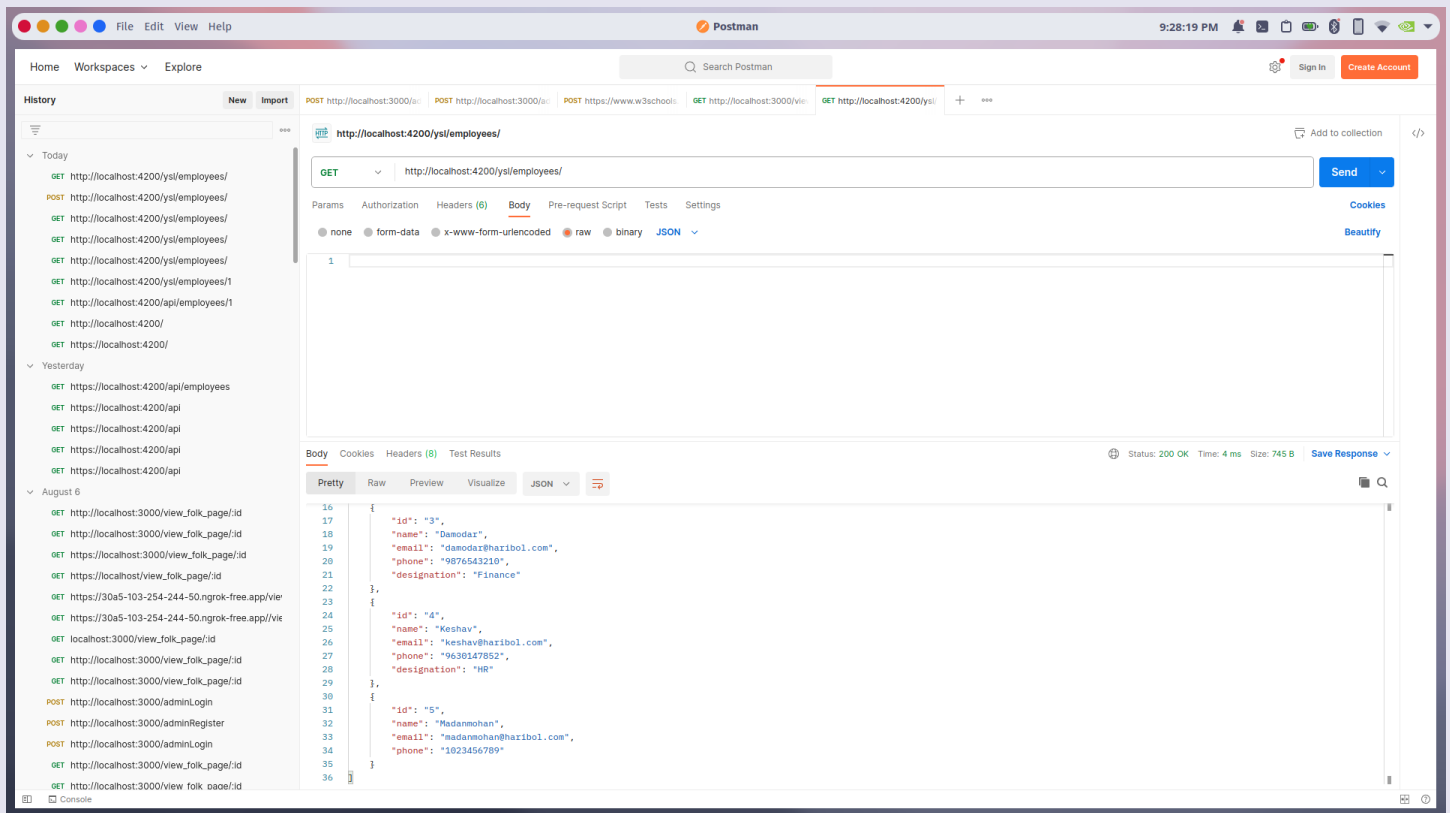
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

- Add details by POST



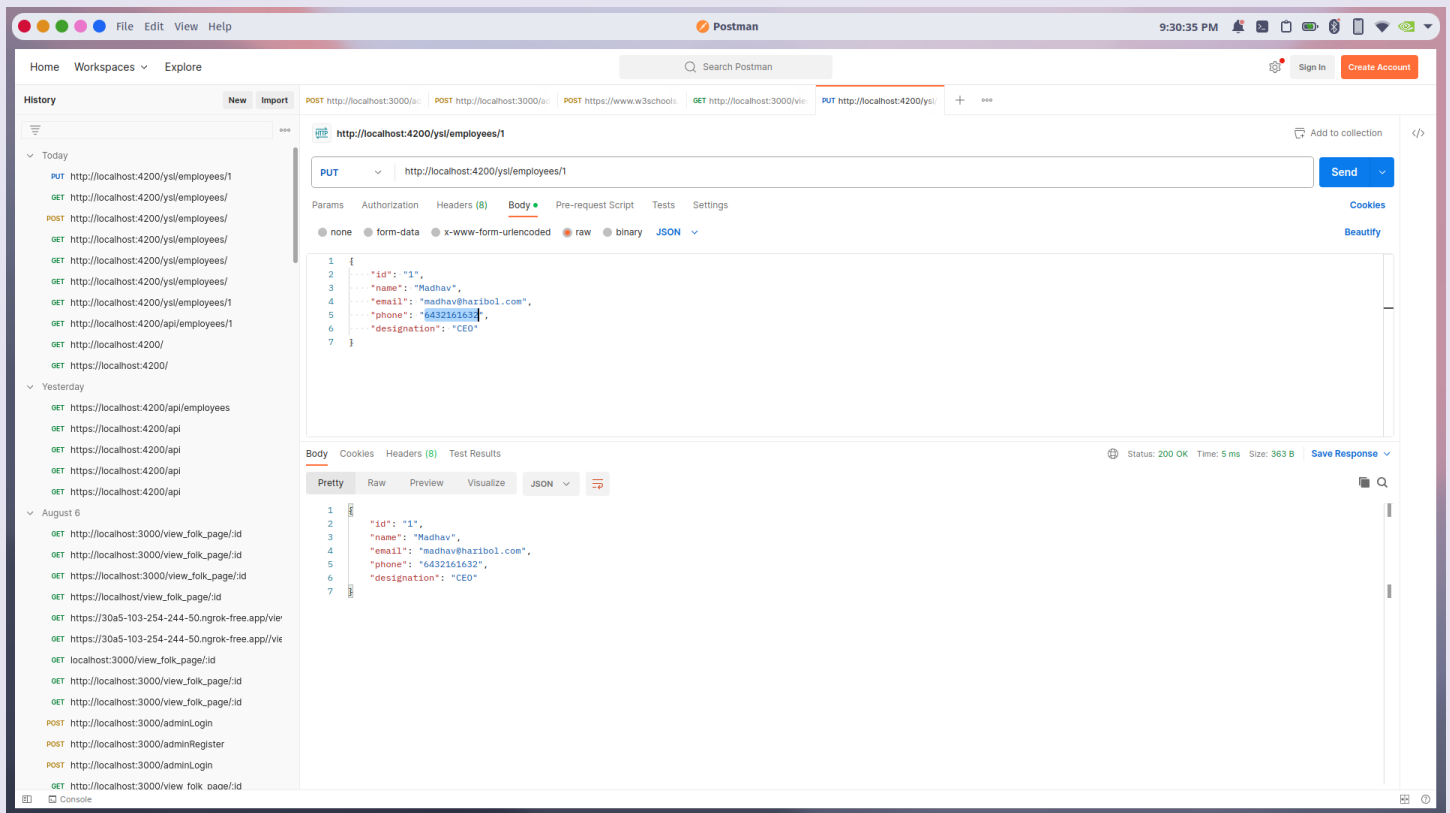
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

- Verify or GET all details



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

- Update details by PUT



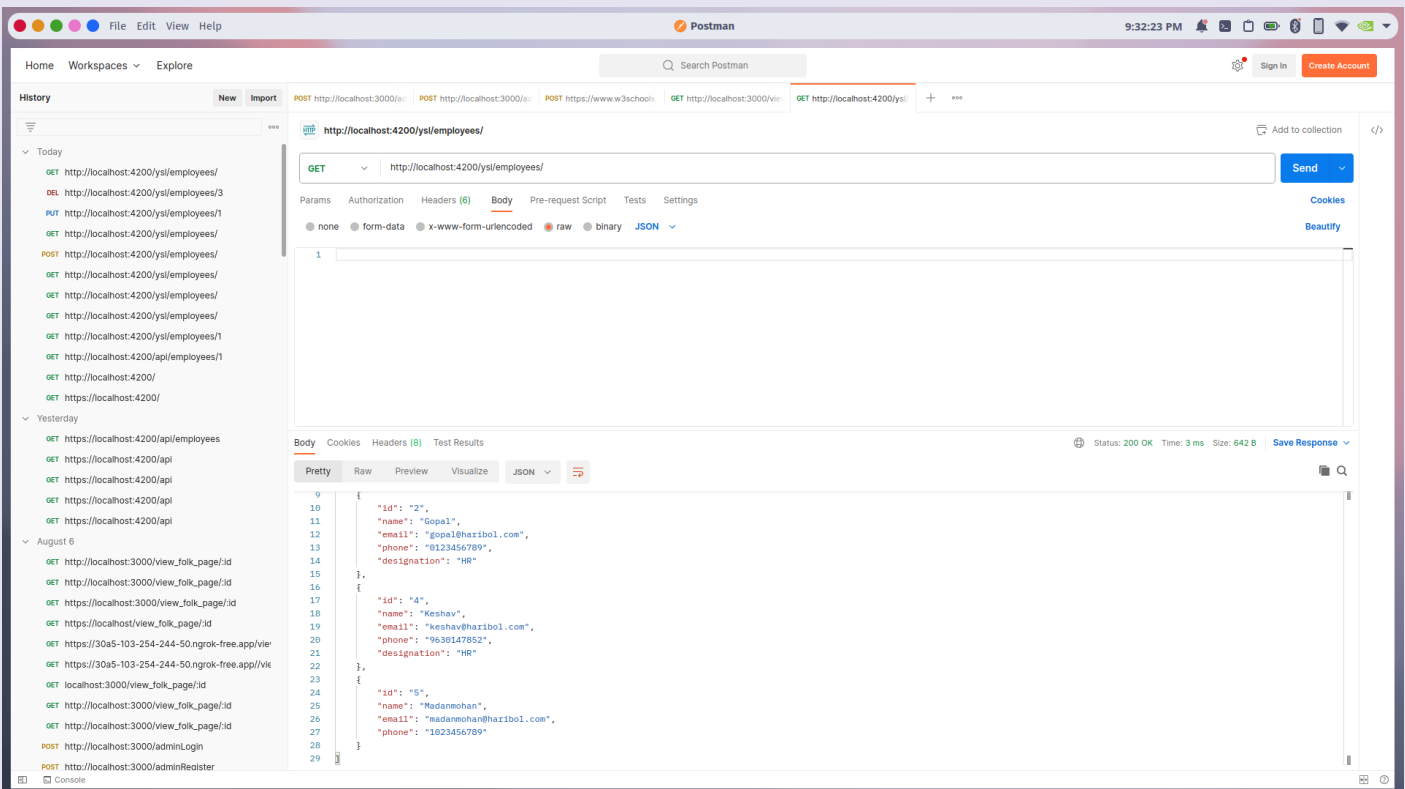
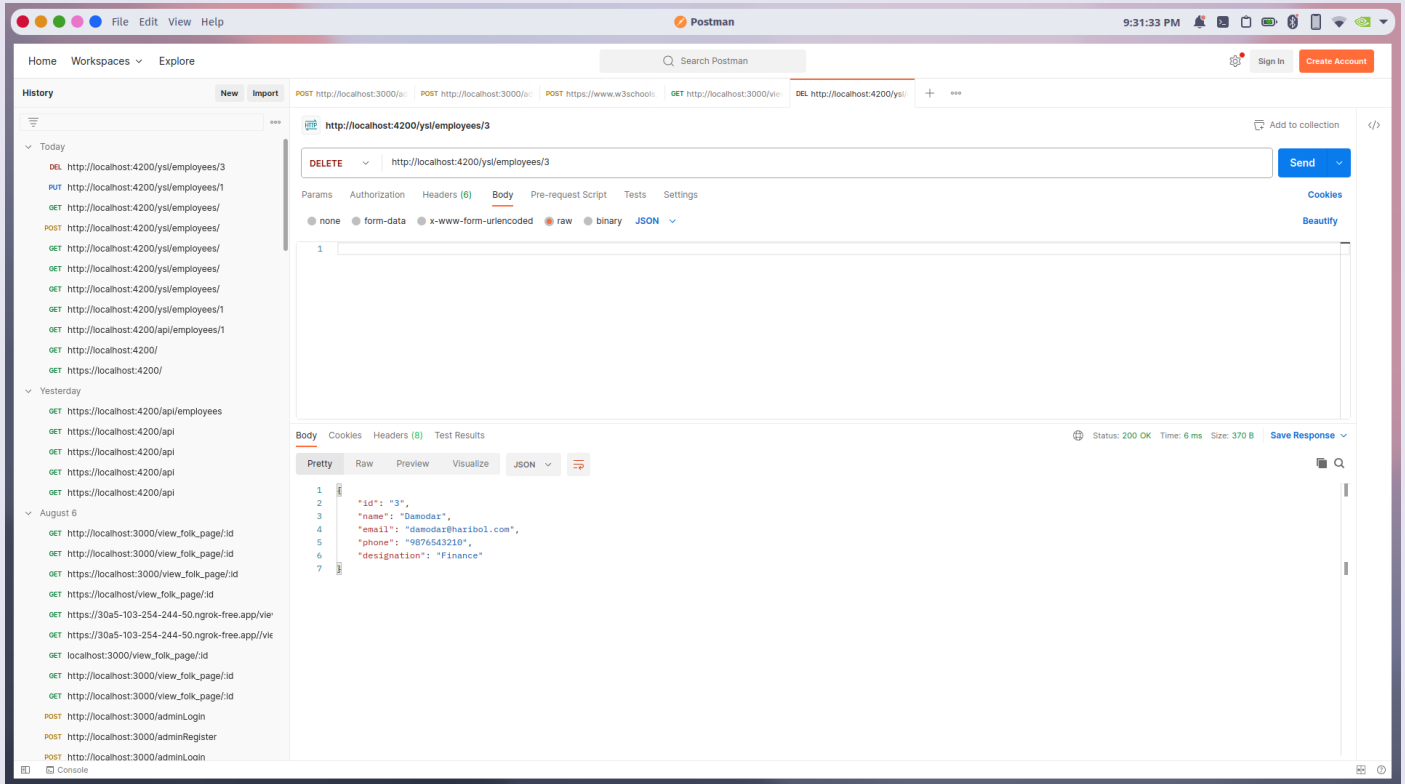
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 51

Microservices Practical 4

- DELETE a record by id



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

3. Code (JS with html integration) :

```
let express = require("express");  
let bodyParser = require("body-parser");  
let app = express();  
  
let port = 4200;  
  
let emps = [  
  {  
    id: "1",  
    name: "Madhav",  
    email: "madhav@haribol.com",  
    phone: "1234567890",  
    designation: "CEO",  
  },  
  {  
    id: "2",  
    name: "Gopal",  
    email: "gopal@haribol.com",  
    phone: "0123456789",  
    designation: "HR",  
  },  
  {
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
id: "3",
name: "Damodar",
email: "damodar@haribol.com",
phone: "9876543210",
designation: "Finance",
},
{
id: "4",
name: "Keshav",
email: "keshav@haribol.com",
phone: "9630147852",
designation: "HR",
},
];

app.use(bodyParser.urlencoded({ extended: false }));
app.use(express.static(__dirname + "/public"));

app.get("/", function (req, res) {
res.sendFile(__dirname + "/public/index.html");
});

app.get("/ysl/employees", (req, res) => {
res.send(emps);
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
});
```

```
app.get("/ysl/employees/get", (req, res) => {  
  const emp = emps.find(({ id }) => id === req.query.id);
```

```
  if (!emp) res.status(404).send("Not found!");  
  else res.send(emp);  
});
```

```
app.post("/ysl/employees/add", (req, res) => {  
  let emp = {  
    id: req.body.id,  
    name: req.body.name,  
    email: req.body.email,  
    phone: req.body.phone,  
    dept: req.body.dept,  
  };  
  emps.push(emp);  
  res.send(emp);  
});
```

```
app.post("/ysl/employees/change", (req, res) => {  
  let emp = emps.find(({ id }) => id === req.body.id);  
  if (!emp) res.status(404).send("Not found!");
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
else {  
  emp.name = req.body.name;  
  emp.email = req.body.email;  
  emp.phone = req.body.phone;  
  emp.designation = req.body.designation;  
  res.send(emp);  
}  
});  
  
app.post("/ysl/employees/delete", (req, res) => {  
  const emp = emps.find(({ id }) => id === req.body.id);  
  if (!emp) res.status(404).send("Not Found!");  
  else {  
    const index = emps.indexOf(emp);  
    emps.splice(index, 1);  
    res.send(emp);  
  }  
});  
  
app.listen(port, () => {  
  console.log(`\nThe server is running on port ${port}\n`);  
});
```


Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

Code (HTML) :

```
<html>

<head>

<title>Haribol Technologies Pvt. Lmt.</title>

<style>

html,

body {

background-color: #efe6e9;

}

* {

font-family: Ubuntu;

font-weight: bold;

}

h3 {

margin-top: 5px;

text-align: center;

font-size: 25px;

}

form {

color: #efe6e9;

padding-left: 20px;
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
padding-right: 20px;
padding-top: 20px;
padding-bottom: 20px;
background-color: #b75969;
width: 45%;
display: inline-block;
margin: 20px;
border-radius: 20px;
}
```

```
label {
margin: 10px;
}
```

```
input {
margin: 5px;
border-radius: 20px;
border: 2px solid #b75969;
padding: 10px;
width: 100%;
}
```

```
button {
display: inline-block;
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
padding: 10px 20px;
font-size: 20px;
cursor: pointer;
text-align: center;
text-decoration: none;
outline: none;
color: #fff;
background-color: #5e81cc;
border: none;
border-radius: 15px;
box-shadow: 0 7px #9999a9;
}

button:hover {
background-color: #a347ba;
}

button:active {
background-color: #a347ba;
box-shadow: 0 5px #797989;
transform: translateY(4px);
}
</style>
</head>
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
<body>

<header style="font-size: 32px; font-weight: bold; margin-left:
400px;margin-top: 30px; margin-bottom: 30px;">


&ensp;&ensp;Haribol Technologies Employee Database

</header>

<form action="/ysl/employees/add" method="POST">

<h3>Add New Employee</h3>

<label>Employee ID</label>

<br><br>

<input type="text" id='EmployeeID' name="id" required>

<br><br>

<label>Name</label>

<br><br>

<input type="text" id="name" name="name" required>

<br><br>

<label>Email ID</label>

<br><br>

<input type="text" id="email" name="email" required>

<br><br>

<label>Phone</label>

<br><br>
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
<input type="text" id="phone" name="phone" required>
<br><br>
<label>Department</label>
<br><br>
<input type="text" id="dept" name="dept" required>
<br><br>
<button type="reset">Reset</button>
<button type="submit">Submit</button>
<br><br>
</form>
<form action="/ysl/employees/change" method="POST">
<h3>Update an Employee</h3>
<label>Employee ID</label>
<br><br>
<input type="text" id='EmployeeID' name="id" required>
<br><br>
<label>Name</label>
<br><br>
<input type="text" id="name" name="name" required>
<br><br>
<label>Email ID</label>
<br><br>
<input type="text" id="email" name="email" required>
<br><br>
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

```
<label>Phone</label>

<br><br>

<input type="text" id="phone" name="phone" required>

<br><br>

<label>Department</label>

<br><br>

<input type="text" id="dept" name="dept" required>

<br><br>

<button type="reset">Reset</button>

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

<br><br>

</form>

<form action="/ysl/employees/get" method="GET">

<h3>View an Employee</h3>

<label>Employee ID</label>

<br><br>

<input type="text" id="EmployeeID" name="id" required>

<br><br>

<button type="reset">Reset</button>

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

<br><br>

</form>

<form action="/ysl/employees/delete" method="POST">

<h3>Delete an Employee</h3>
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

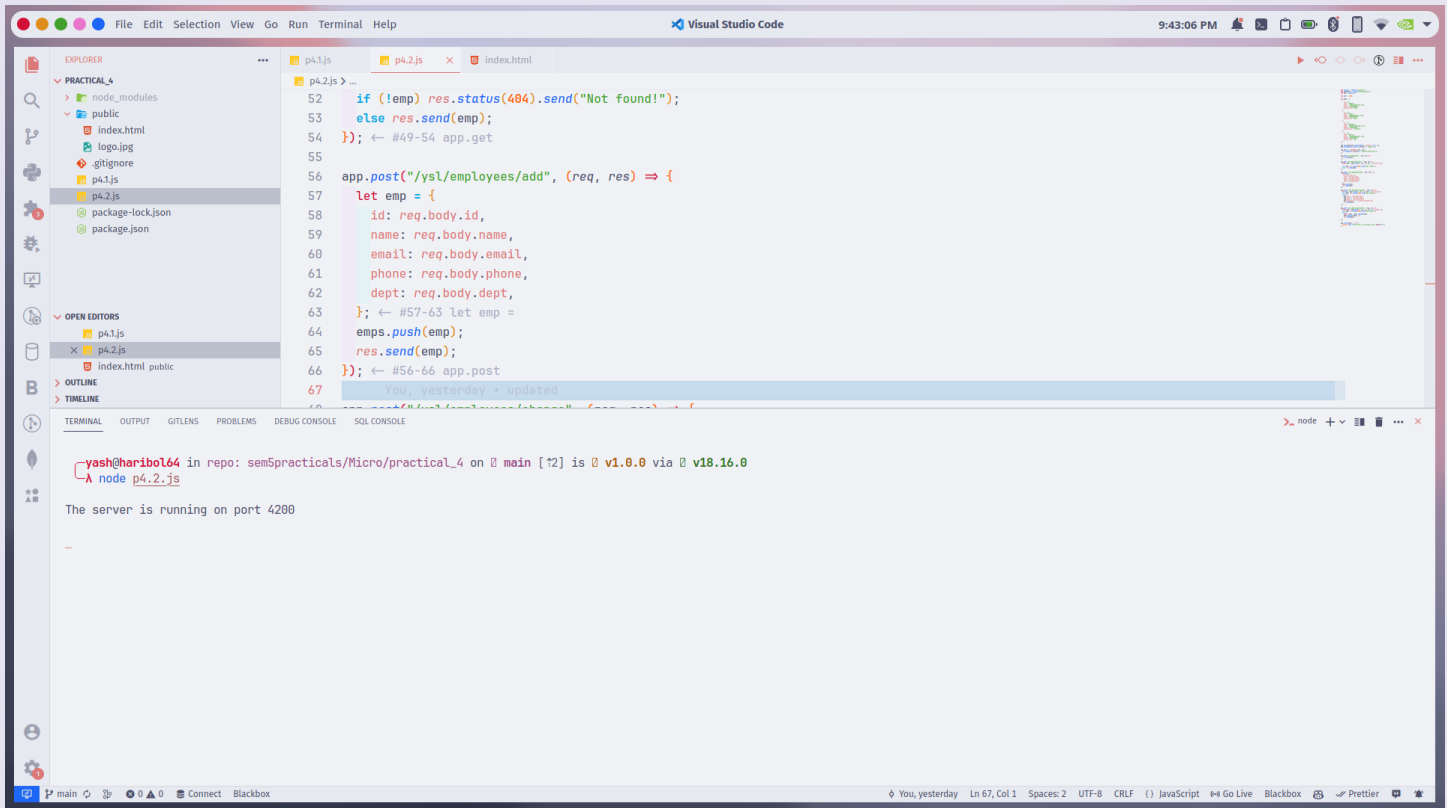
```
<br><br>
<label>Employee ID</label>
<input type="text" id='EmployeeID' name="id" required>
<br><br>
<button type="reset">Reset</button>
<button type="submit">Submit</button>
<br><br>
</form>
<form action="/ysl/employees" method="GET">
<h3>Get all Employee data</h3>
<button type="All Employees">Submit</button>
<br><br>
</form>
</body>

</html>
```

Screenshots :

- Running local server with html on it

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

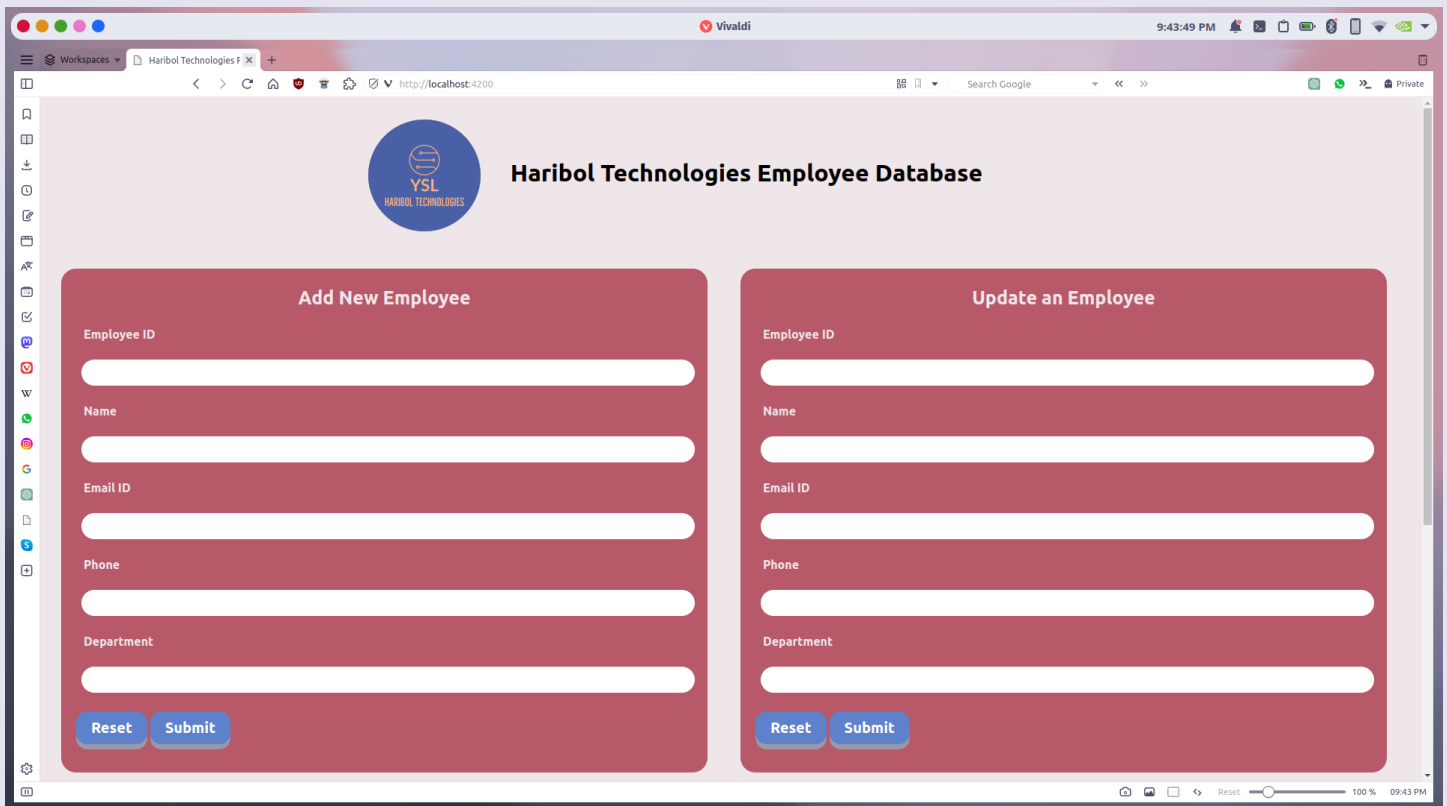


The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left displays the file structure of a project named 'PRACTICAL_4', including files like 'index.html', 'logo.jpg', 'package-lock.json', and 'package.json'. The main editor area shows the contents of 'p4.2.js', which contains a Node.js application using Express.js. The code defines a REST API for managing employees, including endpoints for GET, POST, and PUT. The terminal at the bottom shows the command 'node p4.2.js' being executed, and a message indicating that the server is running on port 4200.

```
52 if (!emp) res.status(404).send("Not found!");
53 else res.send(emp);
54 }); ← #49-54 app.get
55
56 app.post("/ysl/employees/add", (req, res) => {
57   let emp = {
58     id: req.body.id,
59     name: req.body.name,
60     email: req.body.email,
61     phone: req.body.phone,
62     dept: req.body.dept,
63   }; ← #57-63 let emp =
64   emps.push(emp);
65   res.send(emp);
66 }); ← #56-66 app.post
67
```

Terminal Output:

```
yash@haribol64 in repo: sem5practicals/Micro/practical_4 on  main [?] is v1.0.0 via v18.16.0
└─ node p4.2.js
The server is running on port 4200
```



The screenshot shows a web browser displaying the 'Haribol Technologies Employee Database' interface. The page features a header with the company logo and name. Below the header, there are two main sections: 'Add New Employee' and 'Update an Employee'. Each section contains a form with input fields for Employee ID, Name, Email ID, Phone, and Department. The 'Add New Employee' form has a 'Submit' button, while the 'Update an Employee' form has a 'Submit' button. Both forms also include a 'Reset' button. The browser's address bar shows the URL 'http://localhost:4200'.

Haribol Technologies Employee Database

Add New Employee

Employee ID:

Name:

Email ID:

Phone:

Department:

Update an Employee

Employee ID:

Name:

Email ID:

Phone:

Department:

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

- GET all details

The screenshot shows a web application interface with a sidebar on the left containing various icons. The main content area has a light pink background and contains several red-bordered boxes. At the top, there are two identical forms for adding or updating employee details, each with input fields for 'Phone' and 'Department', and 'Reset' and 'Submit' buttons. Below these, there are two more boxes: 'View an Employee' and 'Delete an Employee', each with an 'Employee ID' input field and 'Reset' and 'Submit' buttons. At the bottom, there is a box labeled 'Get all Employee data' with a 'Submit' button. The browser's address bar shows 'http://localhost:4200'.

The screenshot shows the same web application interface, but the 'Get all Employee data' button has been clicked. The main content area now displays a JSON array of employee data. The browser's address bar shows 'http://localhost:4200/ysl/employees'. The JSON data is as follows:

```
[{"id": "1", "name": "Madhav", "email": "madhav@haribol.com", "phone": "1234567890", "designation": "CEO"}, {"id": "2", "name": "Gopal", "email": "gopal@haribol.com", "phone": "0123456789", "designation": "HR"}, {"id": "3", "name": "Damodar", "email": "damodar@haribol.com", "phone": "9876543210", "designation": "Finance"}, {"id": "4", "name": "Keshav", "email": "keshav@haribol.com", "phone": "9630147852", "designation": "HR"}]
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

- Add details by POST

The screenshot shows a web browser window with the URL `http://localhost:4200`. The page title is "Haribol Technologies Employee Database". The main content area has a light pink background. On the left, there is a sidebar with various icons. The main content area contains two forms: "Add New Employee" and "Update an Employee". The "Add New Employee" form is on the left and has the following fields: "Employee ID" (value: 5), "Name" (value: Madanmohan), "Email ID" (value: madanmohan@haribol.com), "Phone" (value: 0987654321), and "Department" (value: Co-Founder). Below the fields are "Reset" and "Submit" buttons. The "Update an Employee" form is on the right and has the same fields but they are empty. It also has "Reset" and "Submit" buttons. The browser's address bar shows the URL `http://localhost:4200` and the page title "Haribol Technologies Employee Database". The browser's status bar shows the time 9:47:51 PM and the page is in Private mode.

Haribol Technologies Employee Database

Add New Employee

Employee ID: 5

Name: Madanmohan

Email ID: madanmohan@haribol.com

Phone: 0987654321

Department: Co-Founder

Reset Submit

Update an Employee

Employee ID:

Name:

Email ID:

Phone:

Department:

Reset Submit

The screenshot shows a web browser window with the URL `http://localhost:4200/ysl/em`. The page displays a JSON response from a POST request. The JSON object is: `{"id": "5", "name": "Madanmohan", "email": "madanmohan@haribol.com", "phone": "0987654321", "dept": "Co-Founder"}`. The browser's address bar shows the URL `http://localhost:4200/ysl/em` and the page title "Haribol Technologies Employee Database". The browser's status bar shows the time 9:48:12 PM and the page is in Private mode.

```
{
  "id": "5",
  "name": "Madanmohan",
  "email": "madanmohan@haribol.com",
  "phone": "0987654321",
  "dept": "Co-Founder"
}
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

- Update details by PUT

The screenshot shows a web browser window with the URL `http://localhost:4200`. The page title is "Haribol Technologies Employee Database". There are two main forms: "Add New Employee" and "Update an Employee". The "Update an Employee" form is active, showing the following fields:

- Employee ID: 1
- Name: Madhav
- Email ID: madhav@haribol.com
- Phone: 6432161632
- Department: CEO and Founder

At the bottom of the form are "Reset" and "Submit" buttons. The "Submit" button is highlighted with a mouse cursor.

The screenshot shows the browser's developer console with the following JSON response:

```
{ "id": "1", "name": "Madhav", "email": "madhav@haribol.com", "phone": "6432161632" }
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

- View details by id

The screenshot shows a web application interface with a sidebar on the left containing various icons. The main content area is divided into four panels. The top-left panel has input fields for 'Phone' and 'Department', with 'Reset' and 'Submit' buttons below. The top-right panel is identical. The bottom-left panel is titled 'View an Employee' and has an 'Employee ID' input field with '5' entered, and 'Reset' and 'Submit' buttons. The bottom-right panel is titled 'Delete an Employee' and has an 'Employee ID' input field and 'Reset' and 'Submit' buttons. At the bottom center, there is a panel titled 'Get all Employee data' with a 'Submit' button. The browser's address bar shows 'http://localhost:4200'.

The screenshot shows the web application after clicking the 'Submit' button in the 'View an Employee' panel. The browser's address bar now shows 'http://localhost:4200/employees/get?id=5'. The main content area displays a JSON object:

```
{ "id": "5", "name": "Madanmohan", "email": "madanmohan@haribol.com", "phone": "0987654321", "dept": "Co-Founder" }
```

. A 'Spectacle' screenshot tool overlay is visible on the right side of the screen, indicating a full-screen screenshot was saved.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

- DELETE details by id

The screenshot shows a web application interface with a sidebar on the left containing various icons. The main content area is divided into several sections:

- Top Left:** A form with two input fields labeled "Phone" and "Department", and two buttons labeled "Reset" and "Submit".
- Top Right:** A form with two input fields labeled "Phone" and "Department", and two buttons labeled "Reset" and "Submit".
- Middle Left:** A section titled "View an Employee" with an input field labeled "Employee ID" containing the value "5", and two buttons labeled "Reset" and "Submit".
- Middle Right:** A section titled "Delete an Employee" with an input field labeled "Employee ID" containing the value "2", and two buttons labeled "Reset" and "Submit". A mouse cursor is hovering over the "Submit" button.
- Bottom Left:** A section titled "Get all Employee data" with a single button labeled "Submit".

The browser's address bar shows the URL `http://localhost:4200`. The status bar at the bottom indicates the time is 9:51 PM and the page is 100% zoomed.

The screenshot shows the same web application interface, but the main content area is now a large white space. The browser's address bar shows the URL `http://localhost:4200/employees/delete`. The status bar at the bottom indicates the time is 9:51 PM and the page is 100% zoomed.

```
{ "id": "2", "name": "Gopal", "email": "gopal@haribol.com", "phone": "0123456789", "designation": "HR" }
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 51
Microservices Practical 4

- Verify by getting all details

