

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

Aim : Create REST API using NodeJS and apply REST method to perform CRUD operations on resources at server regarding universities or Industries scenario and deploy it over AWS Cloud

1. Create a Rest API for universities or industries using NodeJS on the cloud platform

- The REST API implementation of data management for employees in a company named Haribol Technologies, using HTTP methods via express module for CRUD operations is done here with code below.

Code :

```
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 = process.env.PORT || 3000;
let emps = [
{
  id: "1",
  name: "Madhav",
  email: "madhav@haribol.com",
  phone: "1234567890",
  designation: "CEO",
},
{
  id: "2",
  name: "Gopal",
}
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

```
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",
},
];
}

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);
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

```
) ;

app.post("/ysl/employees", (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.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);
});

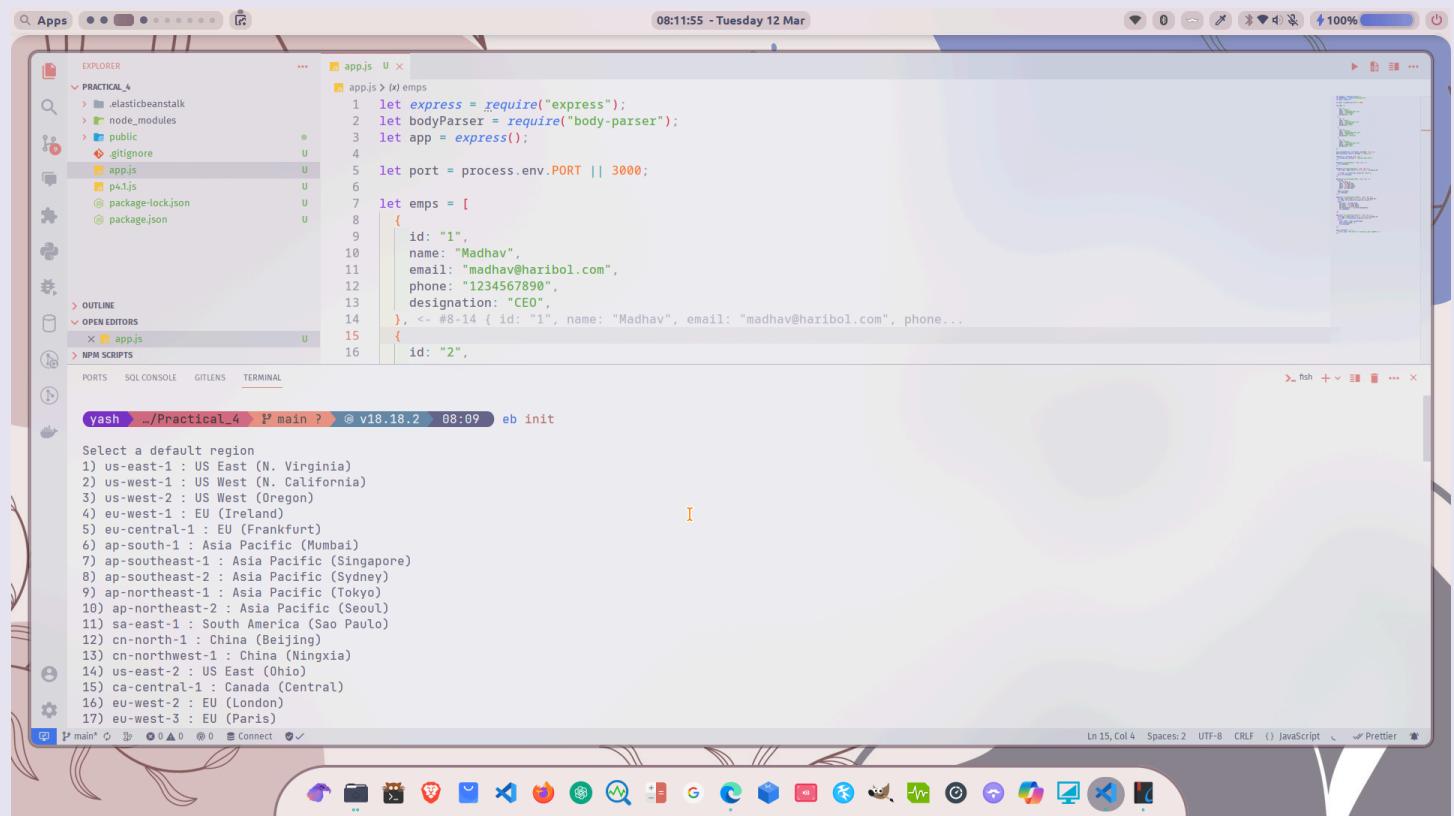
app.delete("/ysl/employees/:id", (req, res) => {
  const emp = emps.find(({ id }) => id === req.params.id);
  if (!emp) res.status(404).send("Not found!");

  const index = emps.indexOf(emp);
  emps.splice(index, 1);
});
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

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

b) Now, initialize the elastic bean service and create application using 'eb init' command in the node project directory



Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 4

- c) Select the preferred region and create new application with node as type, also disabling code commit if not required or not using the service

The screenshot shows a terminal window with the following text output:

```
Enter Application Name
(default is "Practical_4"):
Application Practical_4 has been created.

It appears you are using Node.js. Is this correct?
(Y/n): y
Select a platform branch.
1) Node.js 20 running on 64bit Amazon Linux 2023
2) Node.js 18 running on 64bit Amazon Linux 2023
3) Node.js 18 running on 64bit Amazon Linux 2
4) Node.js 16 running on 64bit Amazon Linux 2 (Deprecated)
5) Node.js 14 running on 64bit Amazon Linux 2 (Deprecated)
(default is 1):

Cannot setup CodeCommit because there is no Source Control setup, continuing with initialization
Do you want to set up SSH for your instances?
(Y/n): n
```

The terminal window is part of a larger interface with an Explorer sidebar on the left containing files like app.js, package.json, and package-lock.json. The status bar at the top shows the date and time as 08:12:23 - Tuesday 12 Mar.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 4

d) Now, create environment in the application using 'eb create env_name' command

The screenshot shows a Mac desktop with a terminal window and a code editor window.

Code Editor (VS Code):

- File: app.js
- Content:

```
let express = require("express");
let bodyParser = require("body-parser");
let app = express();
let port = process.env.PORT || 3000;
let emps = [
  {
    id: "1",
    name: "Madhav",
    email: "madhav@haribol.com",
    phone: "1234567890",
    designation: "CEO"
  },
  {
    id: "2",
    name: "John Doe",
    email: "john.doe@example.com",
    phone: "9876543210",
    designation: "Manager"
  }
];
```

Terminal:

```
yash ~/Practical_4 $ eb create EADEnv
Creating application version archive "app-240312_201240295631".
Uploading Practical_4/app-240312_201240295631.zip to S3. This may take a while.
Upload Complete.
Environment details for: EADEnv
  Application name: Practical_4
  Region: ap-south-1
  Deployed Version: app-240312_201240295631
  Environment ID: e-xtjre7xb2x
  Platform: arn:aws:elasticbeanstalk:ap-south-1::platform/Node.js 20 running on 64bit Amazon Linux 2023/6.1.1
  Tier: WebServer-Standard-1.0
  CNAME: UNKNOWN
  Updated: 2024-03-12 14:42:44.670000+00:00
  Printing Status:
2024-03-12 14:42:43     INFO     createEnvironment is starting.
```

System Tray:

- Icons for various applications like Finder, Mail, Safari, and others.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

- e) Wait for the operations to be completed in AWS cloud console and visit the URL afterwards

The screenshot shows the AWS Elastic Beanstalk environment dashboard for 'EADCenv'. The left sidebar shows the navigation path: Elastic Beanstalk > Environments > EADCenv. The main content area displays the 'EADCenv Info' page. It includes sections for 'Environment overview' (Health: Pending - View causes, Environment ID: e-xtrr7xb2x), 'Platform' (Node.js 20 running on 64bit Amazon Linux 2023/6.1.1), and 'Events' (2). The 'Events' section lists two entries: 'Using elasticbeanstalk-ap-south-1-471112912536 as Amazon S3 storage bucket for environment data.' and 'createEnvironment is starting.'.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

2. Perform CRUD operation on the server using Postman.

- a) In Postman, enter the URL specified in AWS application and first GET all employees details

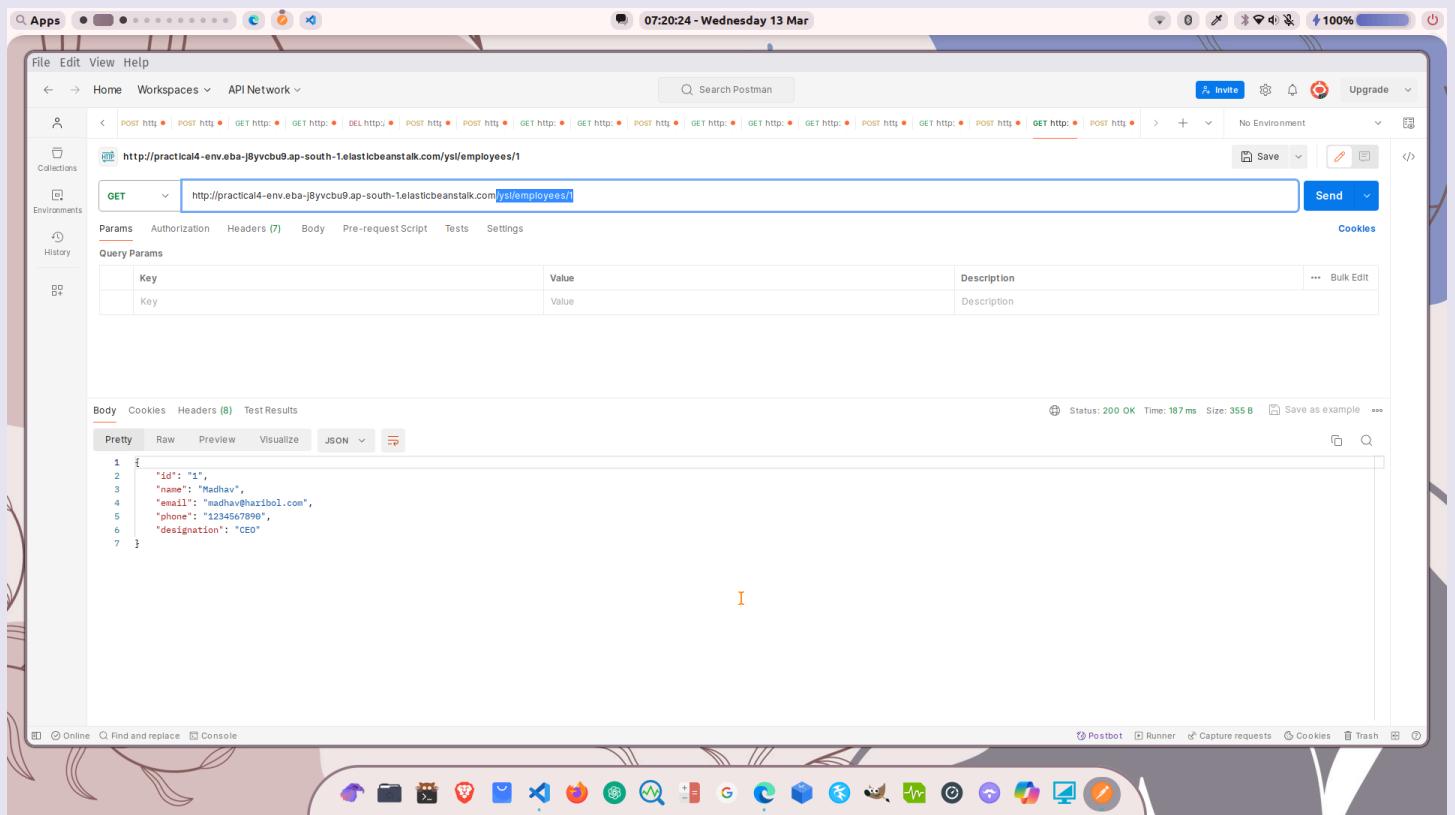
The screenshot shows the Postman application interface. The URL in the address bar is `http://practical4-env.eba-j8yvcbu9.ap-south-1.elasticbeanstalk.com/ysi/employees`. The method dropdown shows "GET". Below the URL, there are tabs for "Params", "Authorization", "Headers (7)", "Body", "Pre-request Script", "Tests", and "Settings". Under "Headers (7)", there is a "Key" column with "Content-Type" and "Value" column with "application/json". The "Body" tab is selected, showing a JSON response with three employee objects:

```
1 [  
2   {  
3     "id": "1",  
4     "name": "Madhav",  
5     "email": "madhav@haribol.com",  
6     "phone": "1234567890",  
7     "designation": "CEO"  
8   },  
9   {  
10    "id": "2",  
11    "name": "Gopal",  
12    "email": "gopal@haribol.com",  
13    "phone": "9876543210",  
14    "designation": "HR"  
15  },  
16  {  
17    "id": "3",  
18    "name": "Damodar",  
19    "email": "damodar@haribol.com",  
20    "phone": "9876543210"  
21  }]
```

The status bar at the bottom indicates "Status: 200 OK Time: 122 ms Size: 652 B".

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

b) To GET the specific employee's data using ID, enter ID after URL and endpoint



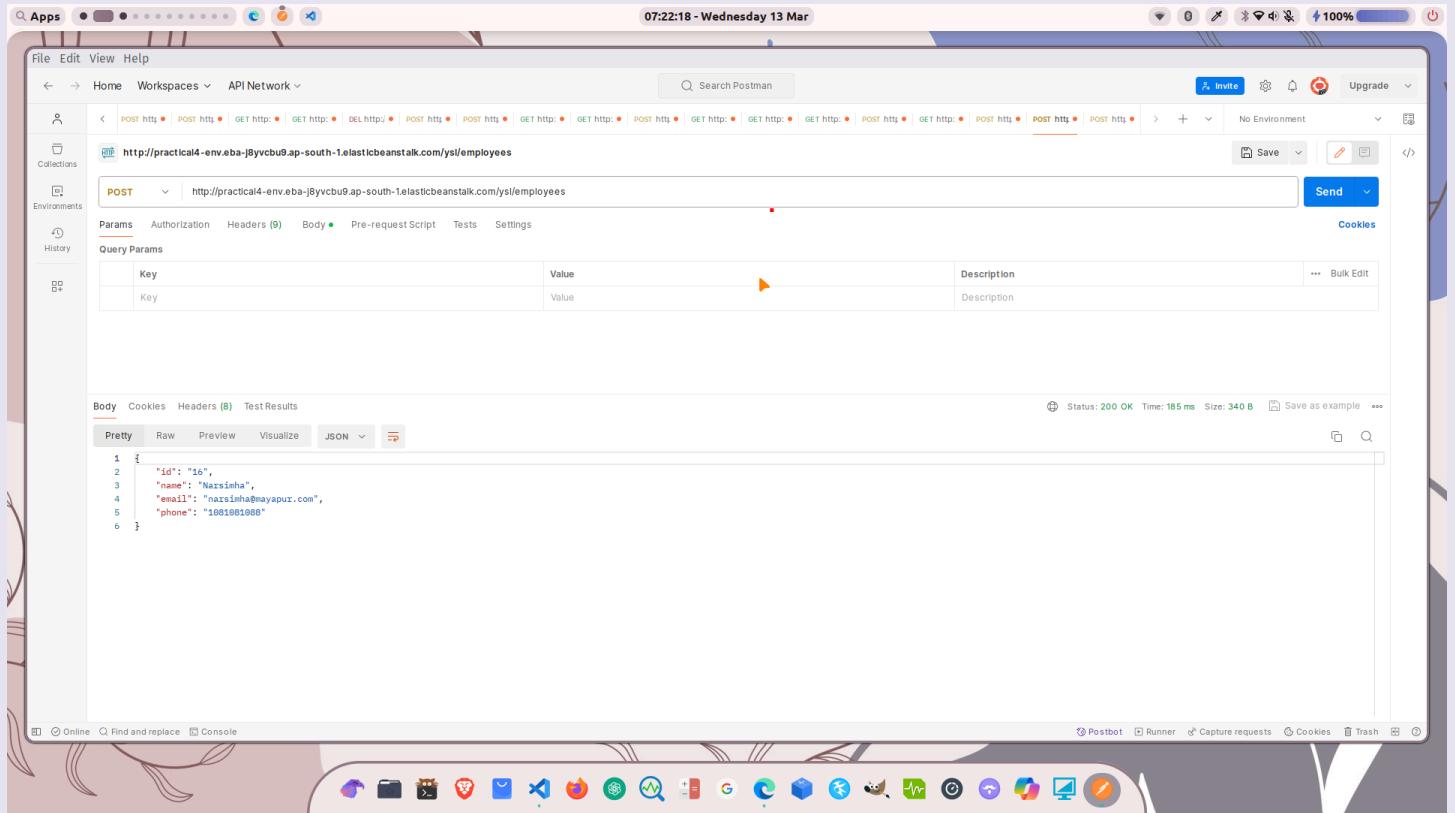
The screenshot shows the Postman application interface. In the top navigation bar, the URL `http://practical4-env.eba-j8yvcbu9.ap-south-1.elasticbeanstalk.com/ysl/employees/1` is entered. The method dropdown is set to `GET`. Below the URL, there are sections for `Params`, `Authorization`, `Headers (7)`, `Body`, `Pre-request Script`, `Tests`, and `Settings`. Under `Query Params`, there is a table with one row containing a key `Key` and a value `Value`. The `Body` tab is selected, showing a JSON response with the following content:

```
1 {
2   "id": "1",
3   "name": "Madhav",
4   "email": "madhav@haribol.com",
5   "phone": "1234567890",
6   "designation": "CEO"
7 }
```

The status bar at the bottom indicates `Status: 200 OK`, `Time: 187 ms`, and `Size: 355 B`.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

c) Now, add the details using POST



The screenshot shows the Postman application interface. The URL in the header is `http://practical4-env.eba-j8yvcbu9.ap-south-1.elasticbeanstalk.com/ysl/employees`. The method selected is `POST`. In the `Body` section, there is a `JSON` payload:

```
1 {
2     "id": "16",
3     "name": "Narsinha",
4     "email": "narsinha@mayaipur.com",
5     "phone": "1081081088"
6 }
```

The response status is `200 OK`, time is `185 ms`, and size is `340 B`. The JSON response body is identical to the posted data.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

d) Updating details can be done using PUT method

The screenshot shows the Postman application interface. The URL in the header is `http://practical4-env.eba-j8yvcbu9.ap-south-1.elasticbeanstalk.com/ysl/employees/16`. The method selected is `PUT`. The body contains the following JSON payload:

```
1 {
2   "id": "16",
3   "name": "Narsimhadev",
4   "email": "narsimha@mayapur.com",
5   "phone": "1081081088",
6   "designation": "Protector"
7 }
```

The response status is `200 OK`, time `146 ms`, and size `370 B`. The response body is identical to the request body.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

e) Also, delete operation can be similarly performed using DELETE method

The screenshot shows the Postman application interface. The URL in the address bar is `http://practical4-env.eba-j8yvcbu9.ap-south-1.elasticbeanstalk.com/ysl/employees/16`. The method selected is `DELETE`. The request body contains the following JSON data:

```
1 {
2   "id": "16",
3   "name": "Narsimhadev",
4   "email": "narsimha@mayapur.com",
5   "phone": "1081081088",
6   "designation": "Protector"
7 }
```

The response status is `200 OK`, time `21 ms`, and size `370 B`. The response body is identical to the request body.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

3. Integrate it with HTML form, where it provides the option to POST the data of new employees, 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 records based on their id

a) The code for HTML page integrating the node REST API is :

Code :

```
<html>

<head>
    <title>Haribol Technologies Pvt. Lmt.</title>
    <style>
        html,
        body {
            background-color: #efef6e9;
        }
        * {
            font-family: Ubuntu;
            font-weight: bold;
        }

        h3 {
            margin-top: 5px;
            text-align: center;
            font-size: 25px;
        }

        form {
            color: #efef6e9;
            padding-left: 20px;
            padding-right: 20px;
        }
    </style>
</head>
<body>
    <h3>Employee Information</h3>
    <form>
        <input type="text" placeholder="Employee ID" />
        <input type="text" placeholder="Name" />
        <input type="text" placeholder="Position" />
        <input type="text" placeholder="Salary" />
        <input type="button" value="Submit" />
    </form>
</body>
</html>
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

```
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;
    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;
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

```
        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>

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

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

```
<br><br>
<input type="text" id="email" name="email" required>
<br><br>
<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/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>
<label>Phone</label>
<br><br>
<input type="text" id="phone" name="phone" required>
<br><br>
```

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 4

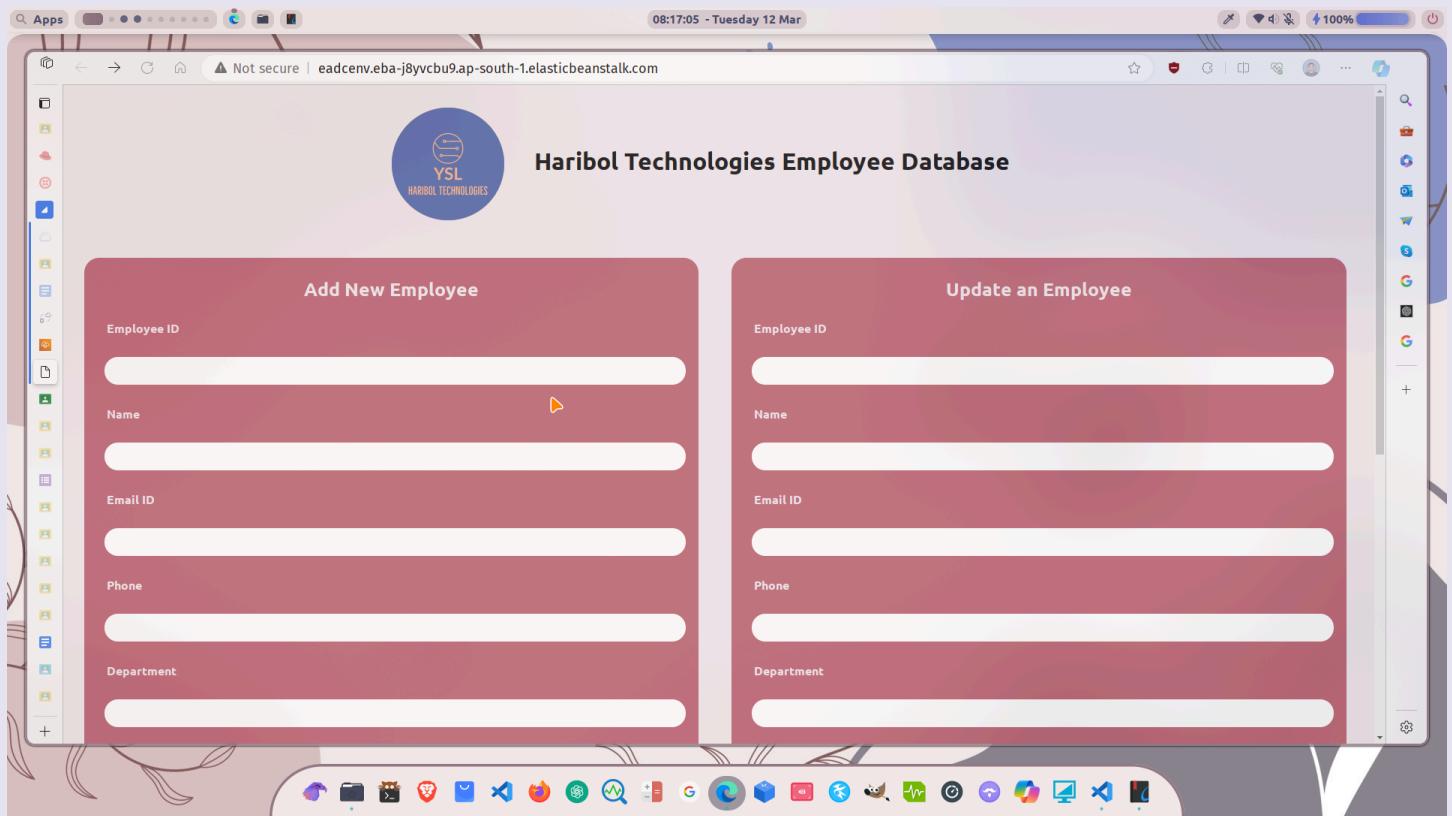
```
<label>Department</label>
<br><br>
<input type="text" id="dept" name="dept" required>
<br><br>
<button type="reset">Reset</button>
<button type="submit">Submit</button>
<br><br>
</form>
<form action="/ysl/employees/get" method="GET">
<h3>View an Employee</h3>
<label>Employee ID</label>
<br><br>
<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>
<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>
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

```
</body>
```

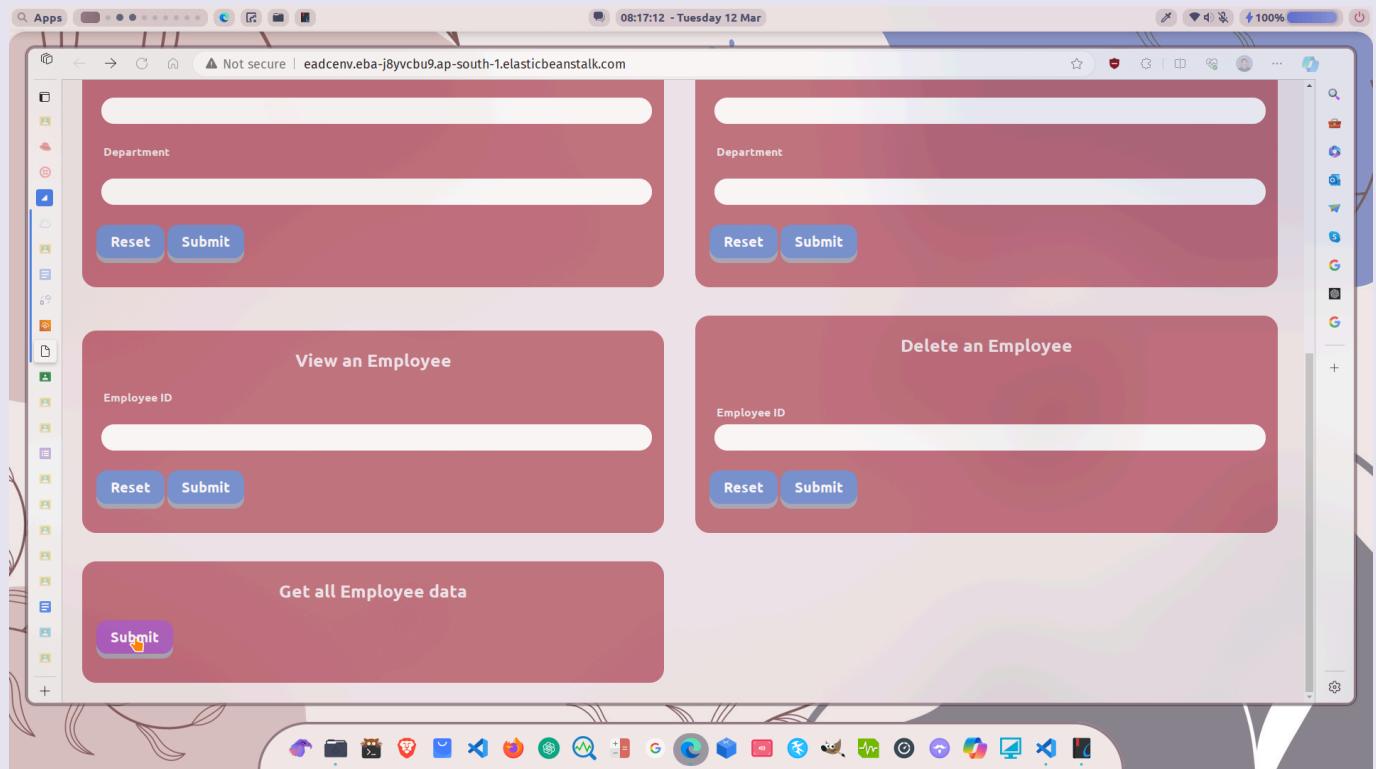
```
</html>
```

- b) Visit the URL specified by AWS application and the HTML page will be visible



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

c) Similar to POSTMAN, but here operations are performed using clicking buttons on page, like GET all data



```
[{"id": "1", "name": "Madhav", "email": "madhav@haribol.com", "phone": "1234567890", "designation": "CEO"}, {"id": "2", "name": "Gopal", "email": "gopal@haribol.com", "phone": "23456789", "designation": "HR"}, {"id": "3", "name": "Dandora", "email": "dandora@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 - 61
EADC Practical 4

d) Add details using POST

Add New Employee

Employee ID
16

Name
Narsimha

Email ID
narsimha@mapur.haribol

Phone
6351573711

Department
Protection

Reset Submit

Update an Employee

Employee ID

Name

Email ID

Phone

Department

Reset Submit

View an Employee

Delete an Employee

```
1: {  
2:   "id": "16",  
3:   "name": "Narsimha",  
4:   "email": "narsimha@mapur.haribol",  
5:   "phone": "6351573711",  
6:   "dept": "Protection"  
7: }
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

e) Update details using PUT

The screenshot shows a web browser window with two forms side-by-side. Both forms have identical fields: Employee ID, Name, Email ID, Phone, and Department. The left form is titled "Add New Employee" and has "16" in the Employee ID field. The right form is titled "Update an Employee" and also has "16" in the Employee ID field. Both forms have "Narsimha" in the Name field, "narsimha@mayaipurharibol" in the Email ID field, "6351573711" and "1081081088" respectively in the Phone field, and "Protection" in the Department field. Each form has "Reset" and "Submit" buttons at the bottom. Below the forms is a navigation bar with "View an Employee" and "Delete an Employee" buttons.

The screenshot shows a web browser window displaying a JSON response. The URL in the address bar is `eadcenv.eba-j8yvcbu9.ap-south-1.elasticbeanstalk.com/ysl/employees/change`. The JSON data is:

```
1 {  
2   "id": "16",  
3   "name": "Narsimha",  
4   "email": "narsimha@mayaipurharibol",  
5   "phone": "1081081088",  
6   "dept": "Protection"  
7 }
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

f) View specific data by ID

The screenshot shows a web browser window with four distinct forms:

- View an Employee:** Contains fields for Employee ID (with value 16), Department (Protection), and buttons for Reset and Submit.
- Delete an Employee:** Contains fields for Employee ID (with value 1081081088), Department (Protection), and buttons for Reset and Submit.
- Get all Employee data:** Contains a single Submit button.

The browser's address bar shows the URL: `Not secure | eadcenv.eba-j8yvcbu9.ap-south-1.elasticbeanstalk.com`. The top status bar indicates the time as 08:19:51 - Tuesday 12 Mar.

The screenshot shows a web browser window displaying a JSON response:

```
1 {  
2   "id": "16",  
3   "name": "Narsimha",  
4   "email": "narsimha@mayapur.haribol",  
5   "phone": "1081081088",  
6   "dept": "Protection"  
7 }
```

The browser's address bar shows the URL: `Not secure | eadcenv.eba-j8yvcbu9.ap-south-1.elasticbeanstalk.com/ysl/employees/get?id=16`. The top status bar indicates the time as 08:19:57 - Tuesday 12 Mar.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 4

g) Delete data by ID

The screenshot shows a web browser window with four distinct forms:

- Top Left Form:** Employee ID: 6351573711. Fields: Department, Protection. Buttons: Reset, Submit.
- Top Right Form:** Employee ID: 1081081088. Fields: Department, Protection. Buttons: Reset, Submit.
- Bottom Left Form:** Title: View an Employee. Employee ID: 16. Buttons: Reset, Submit.
- Bottom Right Form:** Title: Delete an Employee. Employee ID: 16. Buttons: Reset, Submit.

The screenshot shows a web browser window displaying a JSON response from the URL `eadcenv.eba-j8yvcbu9.ap-south-1.elasticbeanstalk.com/ysl/employees/delete`. The JSON data is as follows:

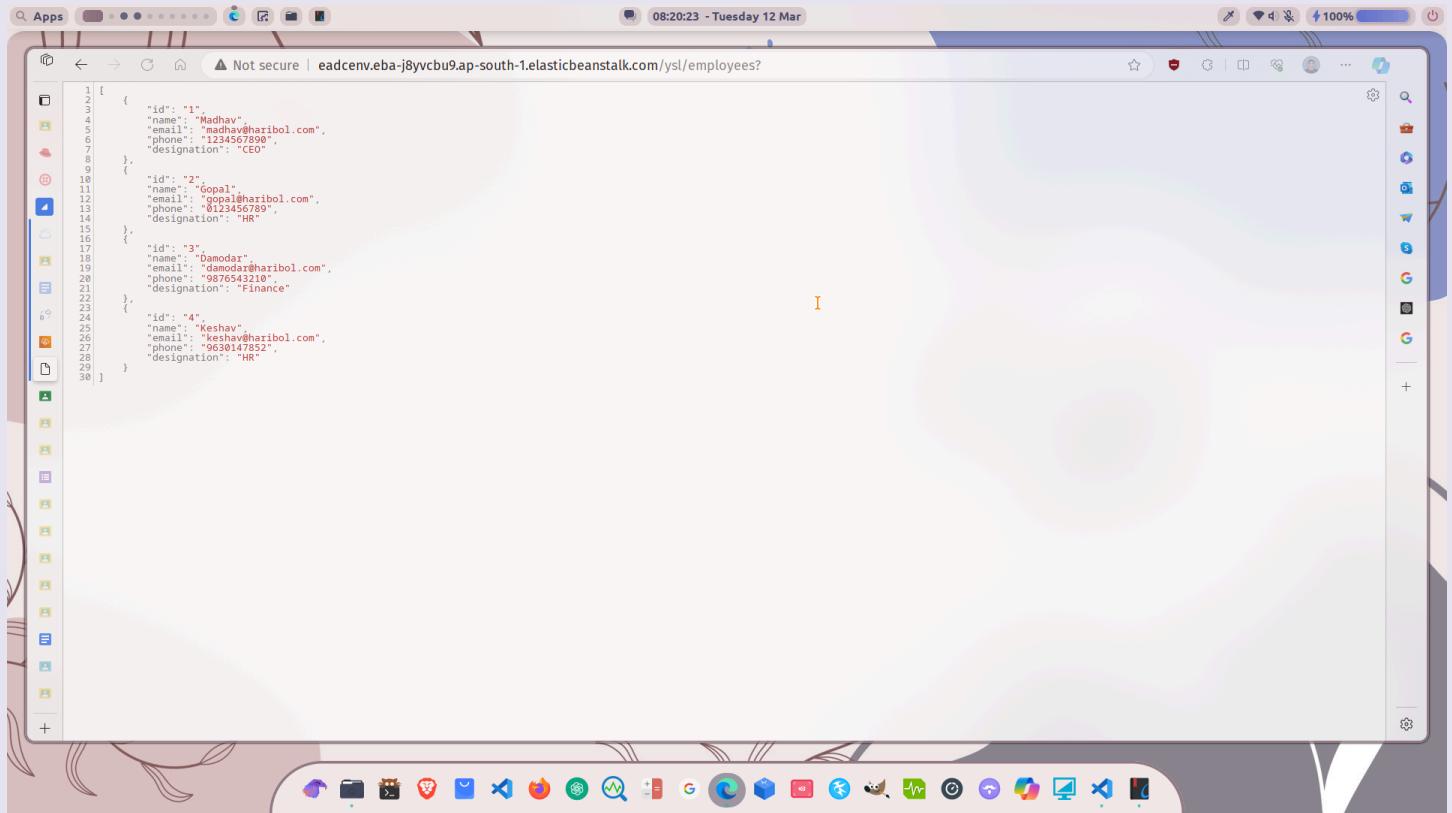
```
1 {  
2   "id": "16",  
3   "name": "Narsimha",  
4   "email": "narsimha@mayapur.haribol",  
5   "phone": "1081081088",  
6   "dept": "Protection"  
7 }
```

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 4



Not secure | eadcenv.eba-j8yvcbu9.ap-south-1.elasticbeanstalk.com/ysl/employees?

```
[{"id": "1", "name": "Madhav", "email": "madhav@haribol.com", "phone": "1234567890", "designation": "CEO"}, {"id": "2", "name": "Gopal", "email": "gopal@haribol.com", "phone": "9876543210", "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"}]
```