

NAME: KSHITIJ GUPTA
Enrolment Number: 21162101007
Sub: MS
Practical – 4

1. Create a Rest API for universities or industry using NodeJS on any IDE

.JS Source Code:

```
//21162101007_kshitijgupta
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());

var port = 8080;
var employees = [
  { id: 1, name: 'SHREE RAM', email: 'ram@shaketdham.com',
    phoneNumber: '1234567890', department: 'HR' },
  { id: 2, name: 'Jane Smith', email: 'jane@example.com',
    phoneNumber: '0987654321', department: 'IT' },
  // Add more employees here
];

app.get('/', (req, res) => {
  const Welcome_Page = `
    <html>
    <head>
    <style>
```

```

        body {
            font-family: 'Malgun Gothic', sans-serif;
        }
        h2 {
            color: Green;
        }
    </style>
</head>
<body>
    <center> <h2>Welcome to Employee Management REST
API!</h2></center>
</body>
</html>
`;

    res.send(Welcome_Page);
});

app.get('/api/employees', (req, res) => {

    const Emp_Table = `
<html>
<head>
    <style>
        table {
            border-collapse: collapse;
            width: 100%;
        }
        th, td {
            border: 1px solid black;
            padding: 8px;
            text-align: left;
        }
        th {
            background-color: lightgray;
        }
    </style>
</head>
<body>
    <h2>Employee List</h2>
    <table>
        <tr>
            <th>id</th>
            <th>name</th>

```

```

        <th>email</th>
        <th>phoneNumber</th>
        <th>department</th>
    </tr>
    ${employees.map(employee => `
        <tr>
            <td>${employee.id}</td>
            <td>${employee.name}</td>
            <td>${employee.email}</td>
            <td>${employee.phoneNumber}</td>
            <td>${employee.department}</td>
        </tr>
    `).join('')}
</table>
</body>
</html>
`;

    res.send(Emp_Table);
});

app.get('/api/employees/:id', (req, res) => {
    const employee = employees.find(({ id }) => id ===
parseInt(req.params.id));

    if (!employee) {
        res.status(404).send('<h2 style="font-family: Malgun
Gothic; color: darkred;">Oops... Can\'t find what you are
looking for!</h2>');
    } else {
        const Condition_output = `
            <html>
            <head>
                <style>
                    body {
                        font-family: 'Malgun Gothic', sans-
serif;
                    }
                    h2 {
                        color: darkred;
                    }
                </style>
            </head>
            <body>

```

```

        <h2>Employee Information</h2>
        <p><strong>ID:</strong> ${employee.id}</p>
        <p><strong>Name:</strong> ${employee.name}</p>
        <p><strong>Email:</strong>
${employee.email}</p>
        <p><strong>Phone Number:</strong>
${employee.phoneNumber}</p>
        <p><strong>Department:</strong>
${employee.department}</p>
    </body>
</html>
    `;

    res.send(Condition_output);
}
});

app.post('/api/employees', (req, res) => {
    var employee = {
        id: employees.length + 1,
        name: req.body.name,
        email: req.body.email,
        phoneNumber: req.body.phoneNumber,
        department: req.body.department
    };
    employees.push(employee);
    res.send(employee);
});

app.put('/api/employees/:id', (req, res) => {
    var employee = employees.find(({ id }) => id ===
parseInt(req.params.id));
    if (!employee)
        res.status(404).send('<h2 style="font-family: Malgun
Gothic; color: darkred;">Not Found!! </h2>');

    employee.name = req.body.name;
    employee.email = req.body.email;
    employee.phoneNumber = req.body.phoneNumber;
    employee.department = req.body.department;
    res.send(employee);
});

app.delete('/api/employees/:id', (req, res) => {

```

```

    const employee = employees.find(({ id }) => id ===
parseInt(req.params.id));
    if (!employee)
        res.status(404).send('<h2 style="font-family: Malgun
Gothic; color: darkred;"> Not Found!! </h2>');

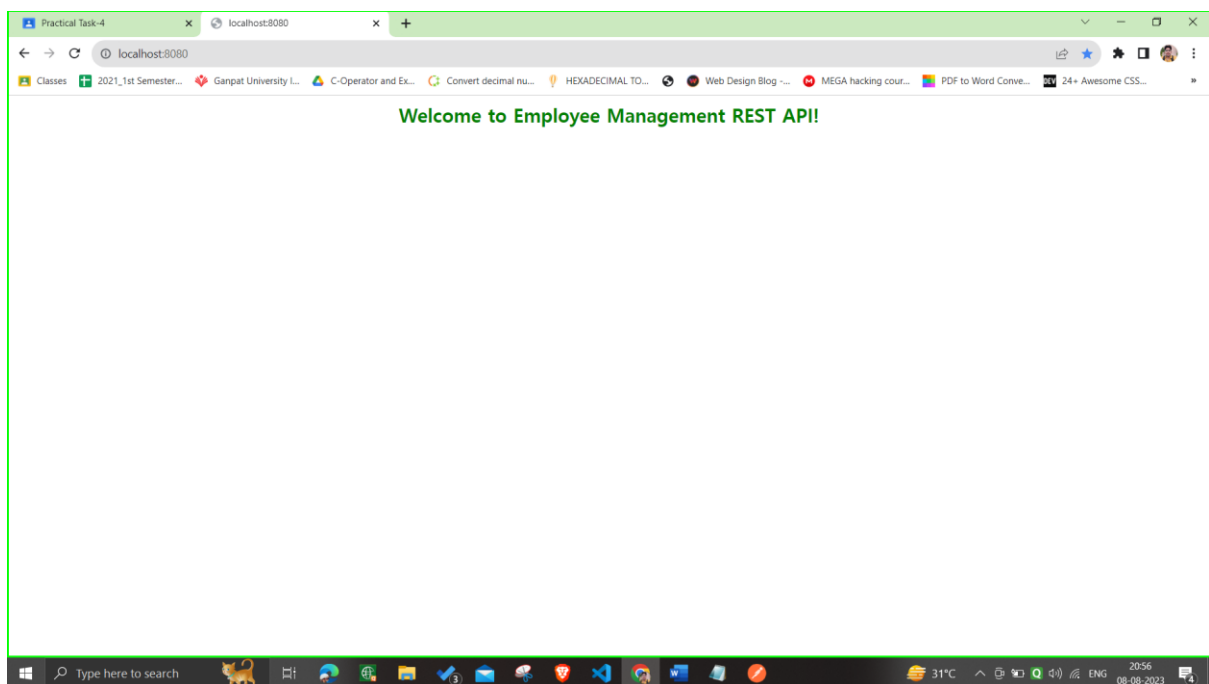
    const index = employees.indexOf(employee);
    employees.splice(index, 1);

    res.send(employee);
});

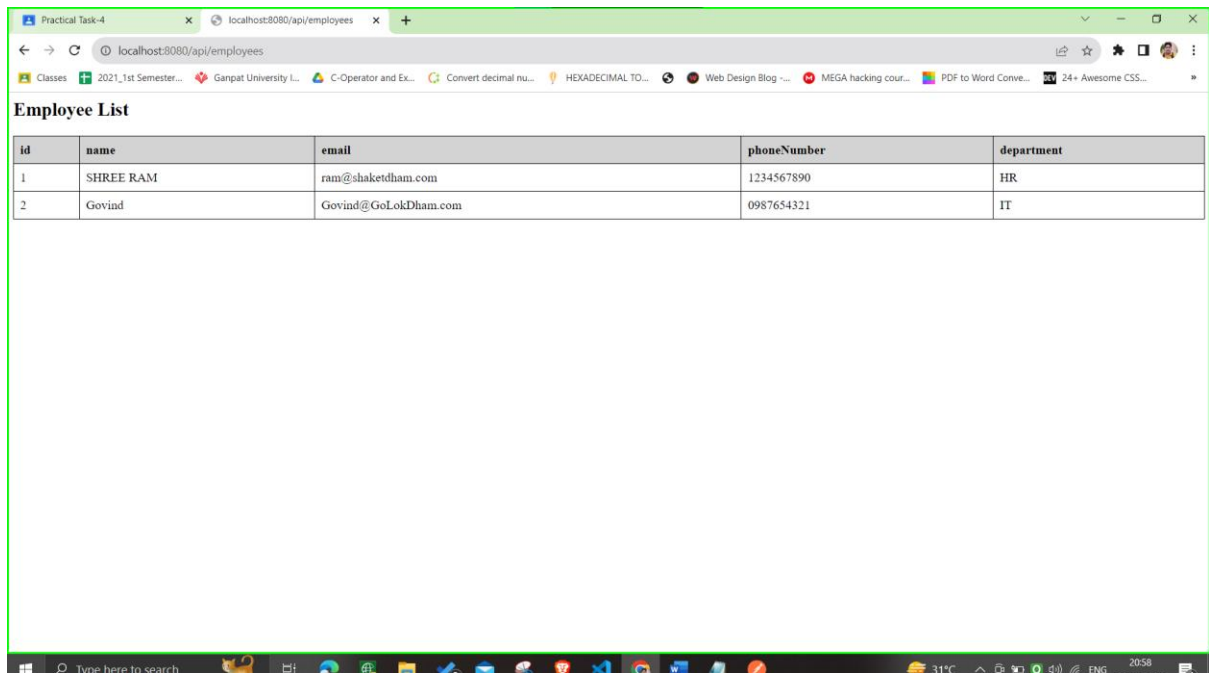
app.listen(port, () => {
    console.log('Server is running on port 8080');
});

```

OUTPUT:



Employee list:



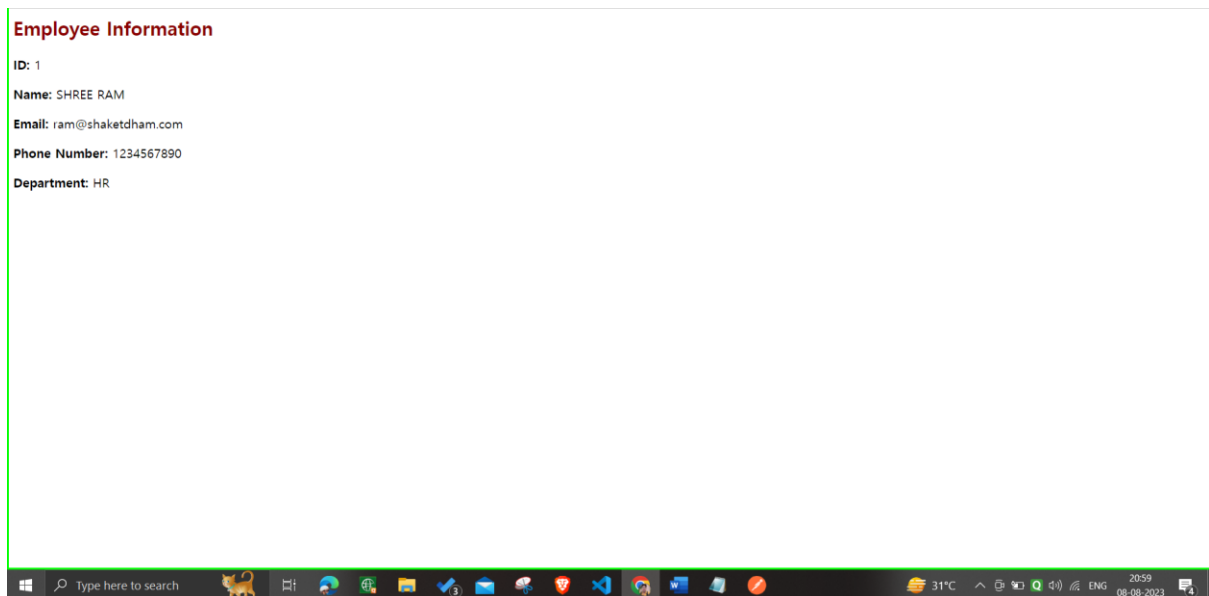
Practical Task-4

localhost:8080/ap/employees

Employee List

id	name	email	phoneNumber	department
1	SHREE RAM	ram@shaketdham.com	1234567890	HR
2	Govind	Govind@GoLokDham.com	0987654321	IT

Employee List Using ID:



Employee Information

ID: 1

Name: SHREE RAM

Email: ram@shaketdham.com

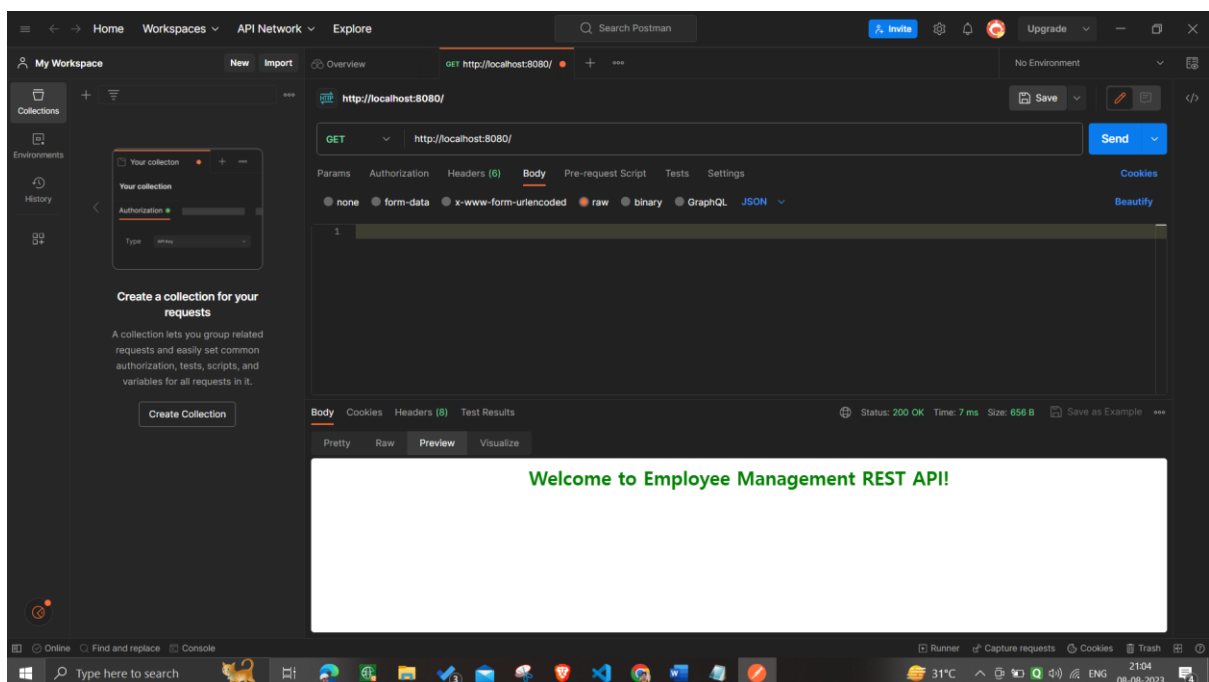
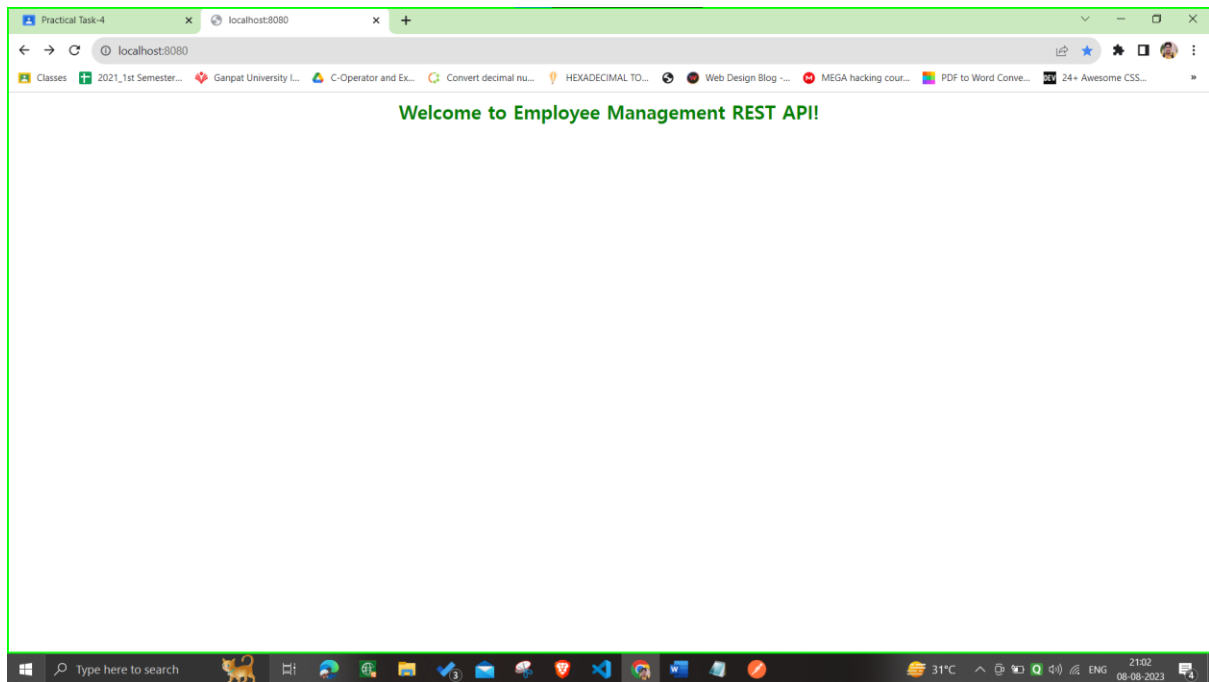
Phone Number: 1234567890

Department: HR

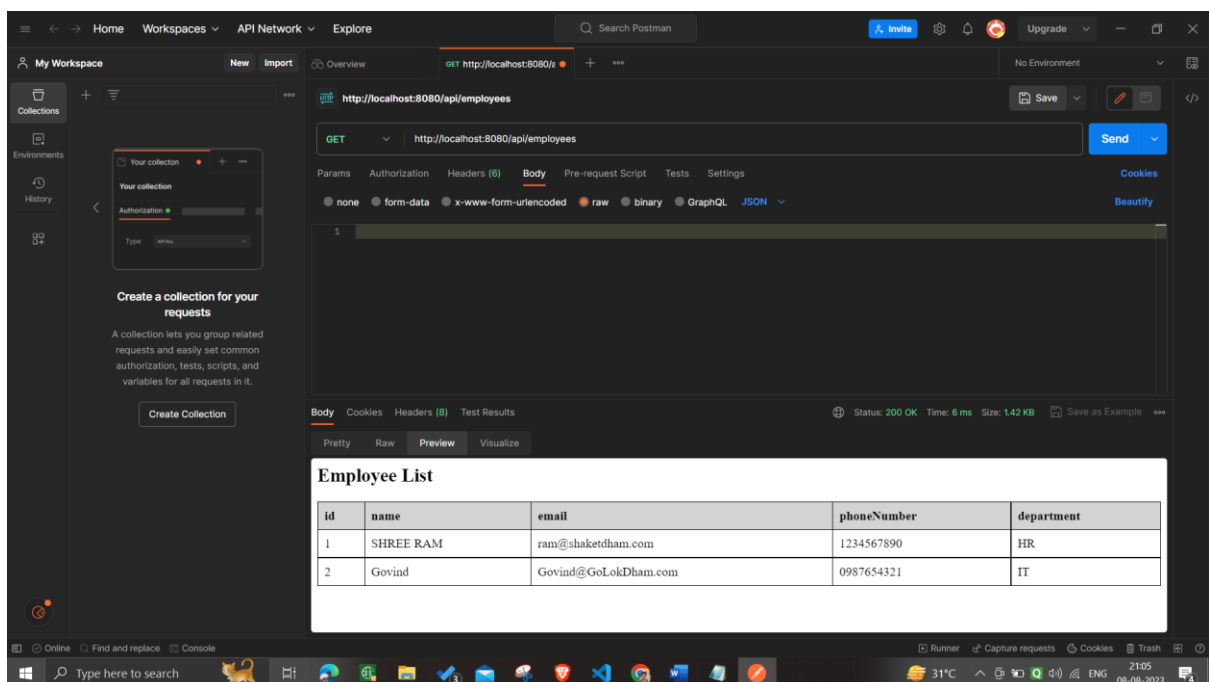
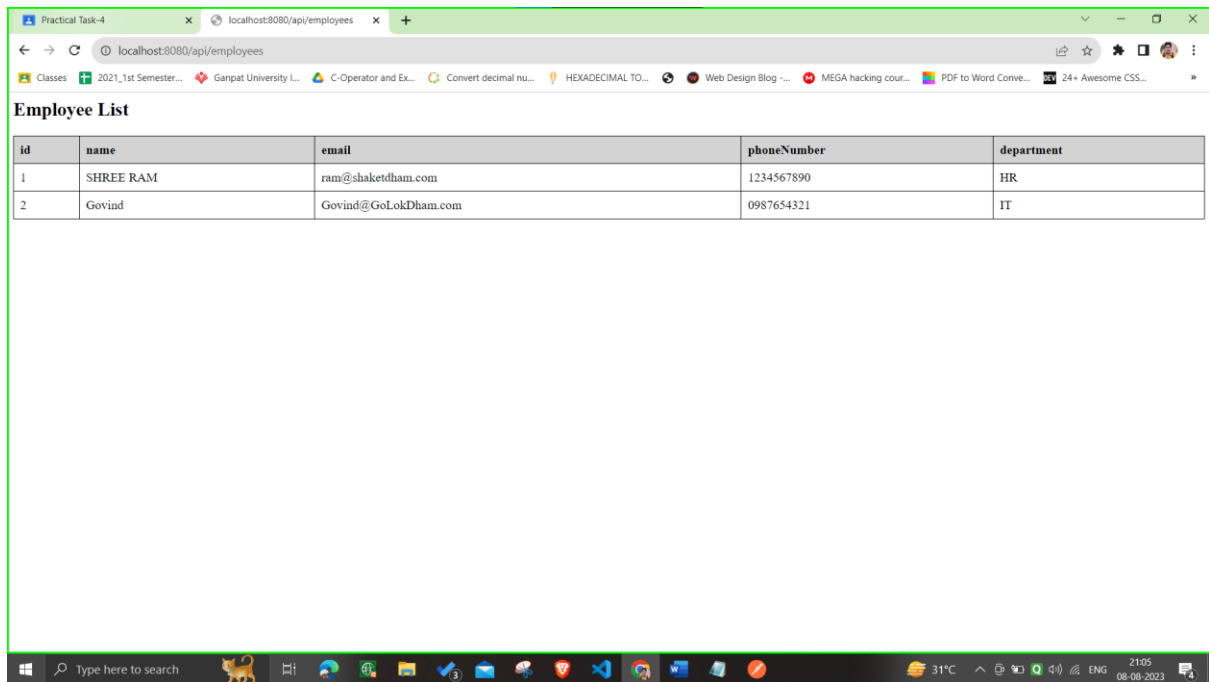
2. Perform CRUD operation on server using Postman.

1. GET Command's:

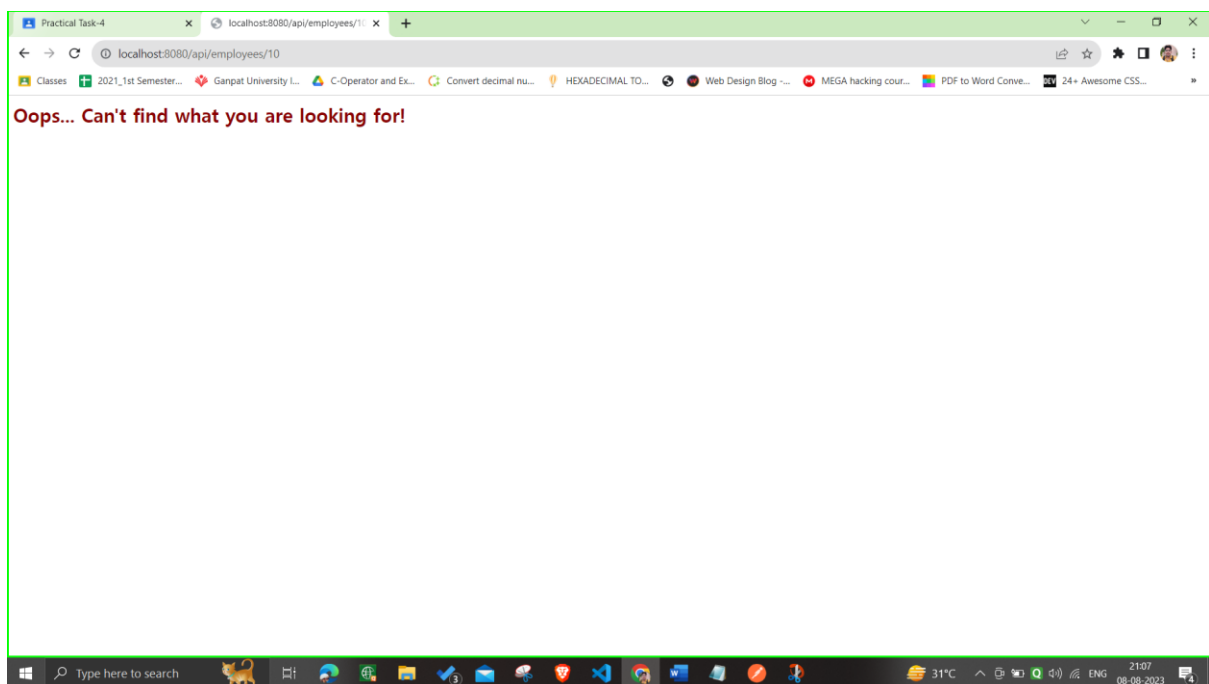
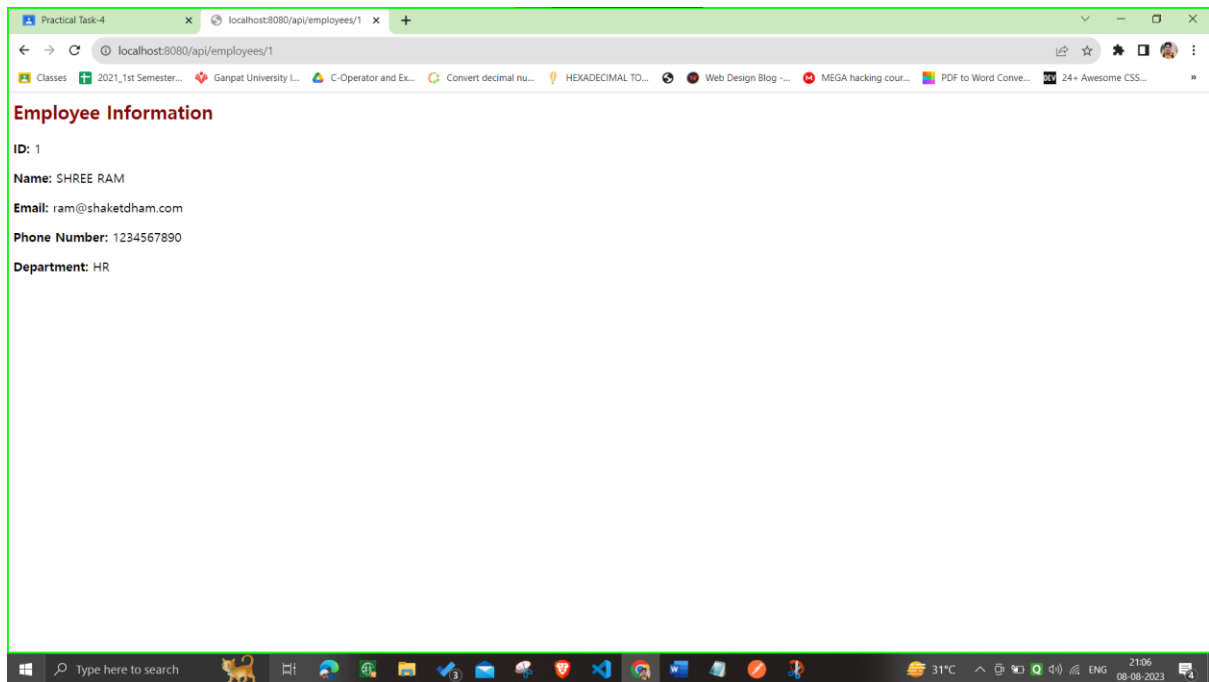
a. 1. <http://localhost:8080/>

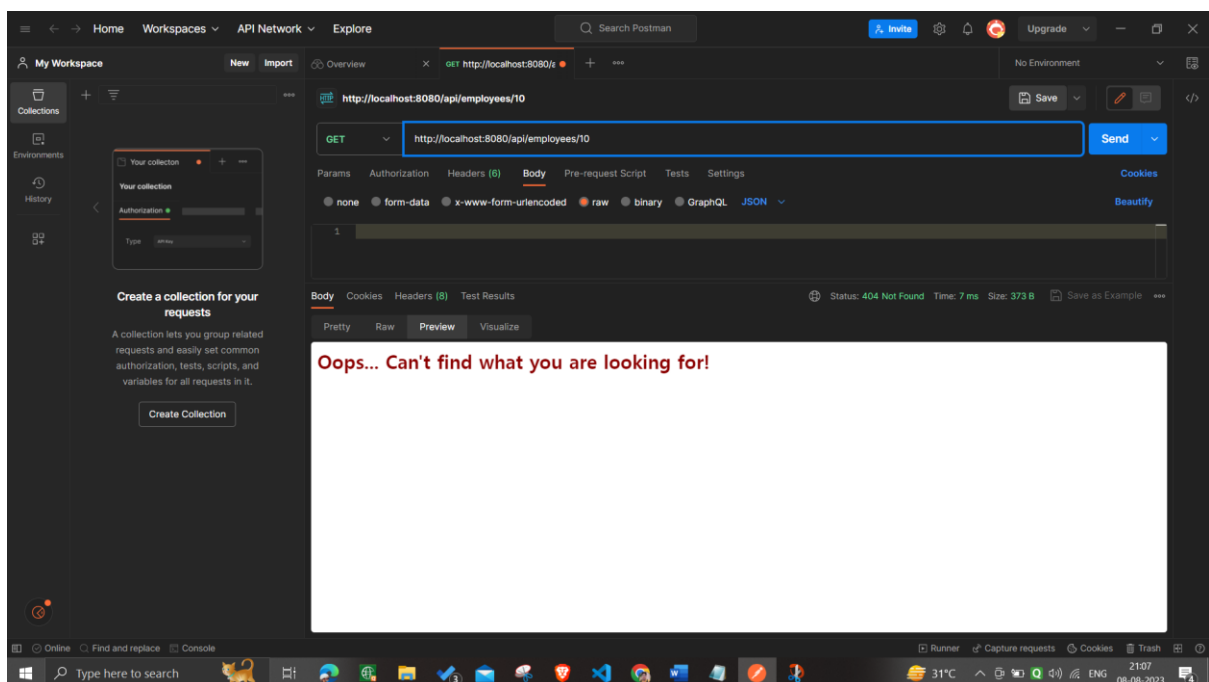
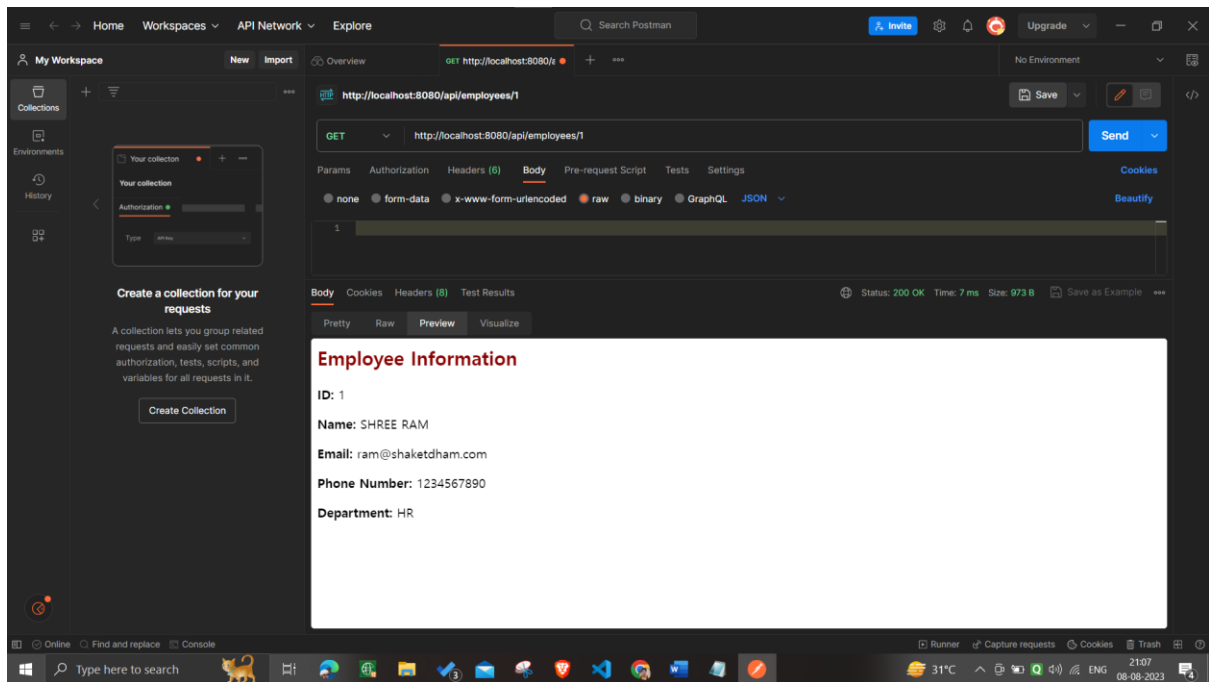


b. <http://localhost:8080/api/employees>



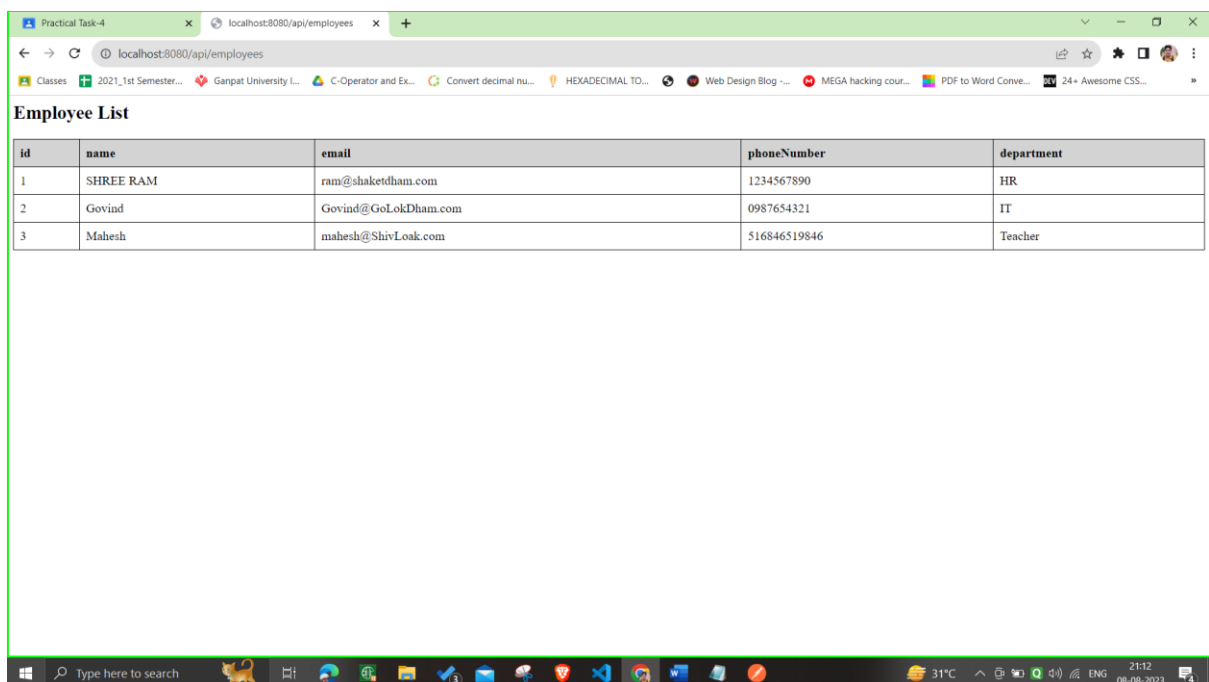
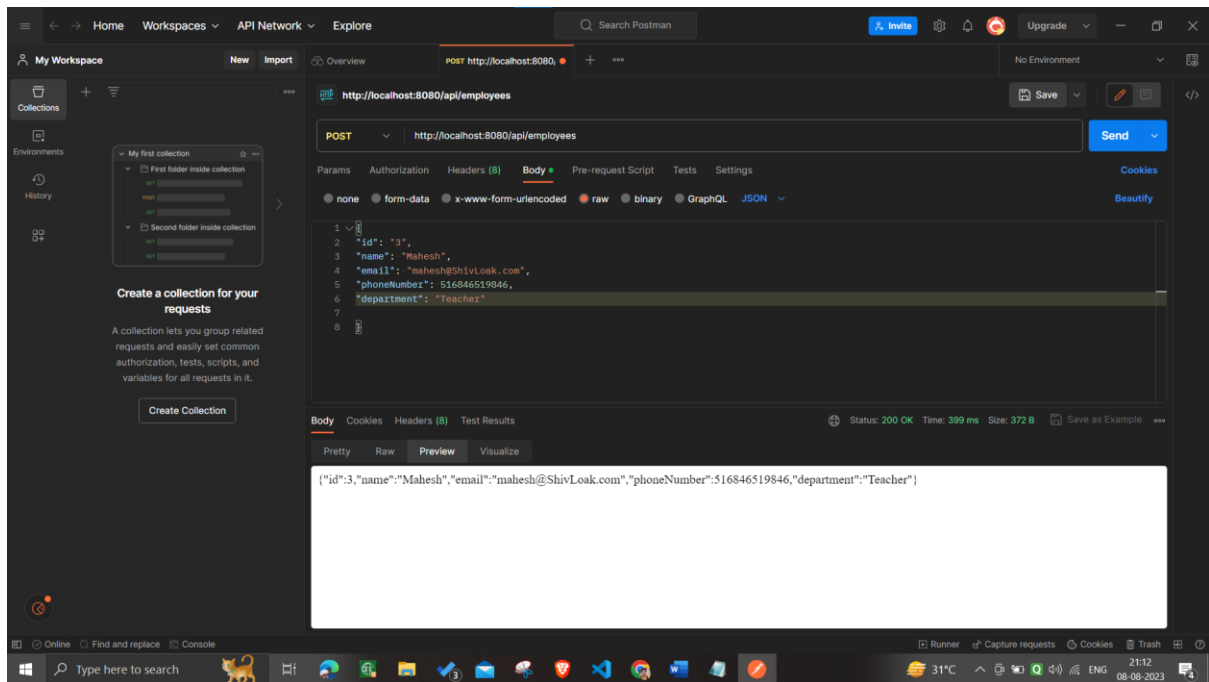
c. <http://localhost:8080/api/employees/1>



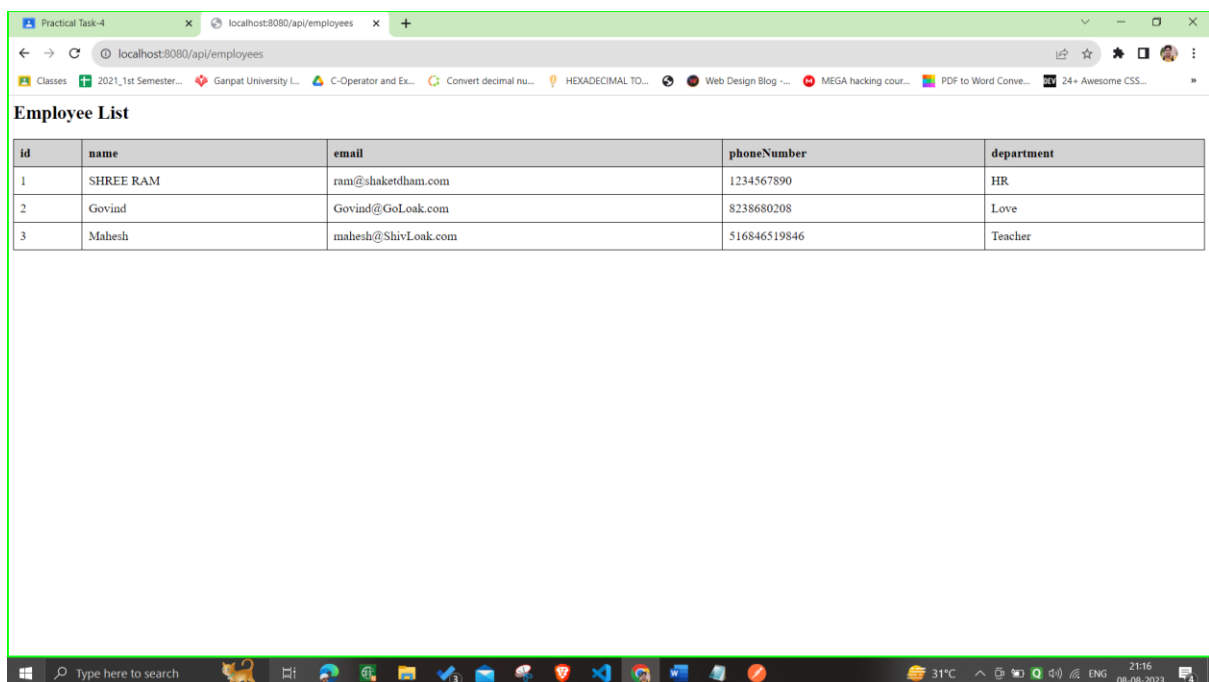
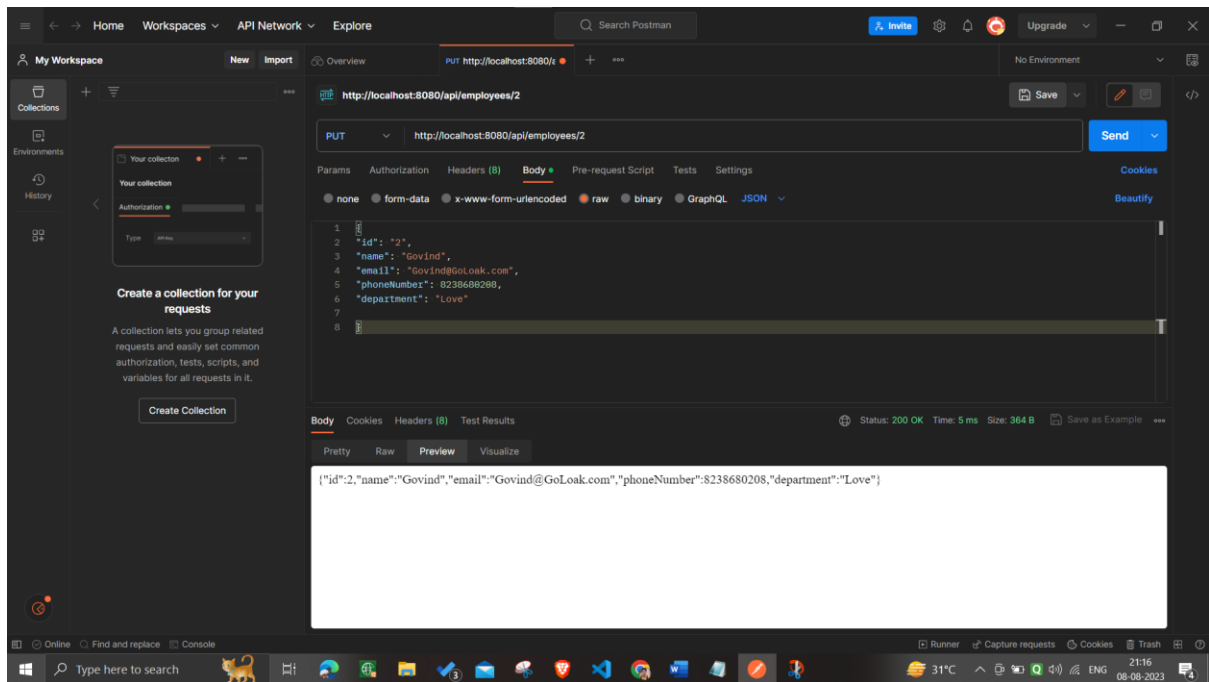


POST Commands:

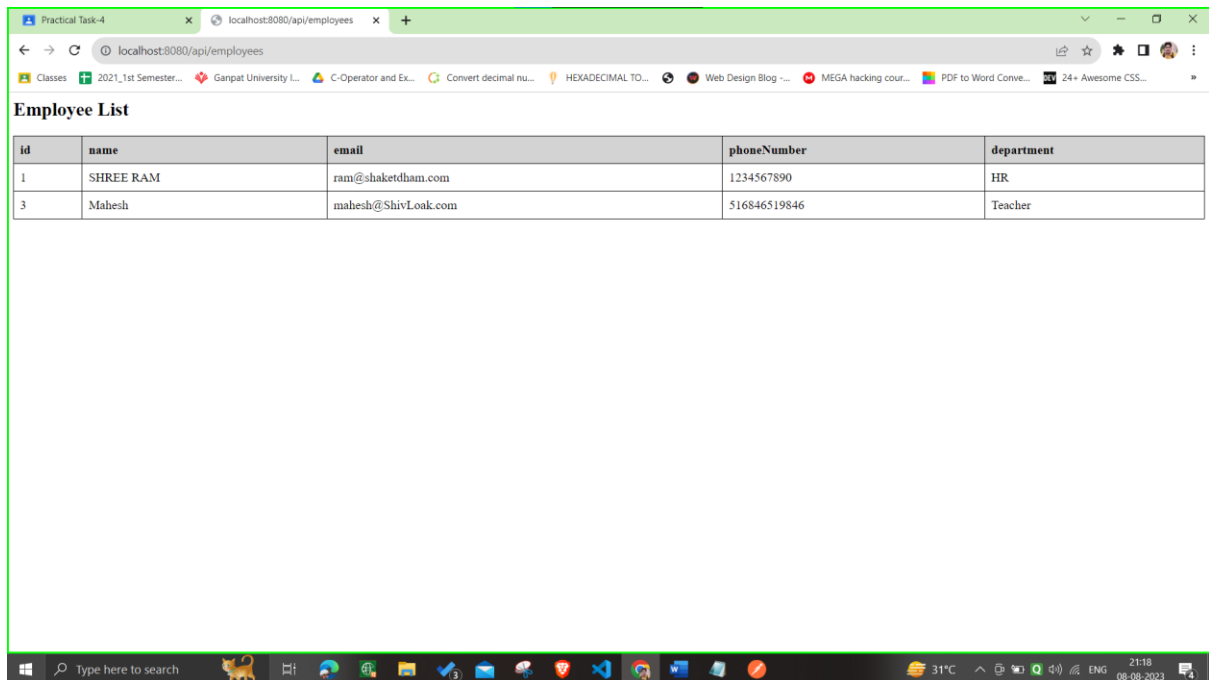
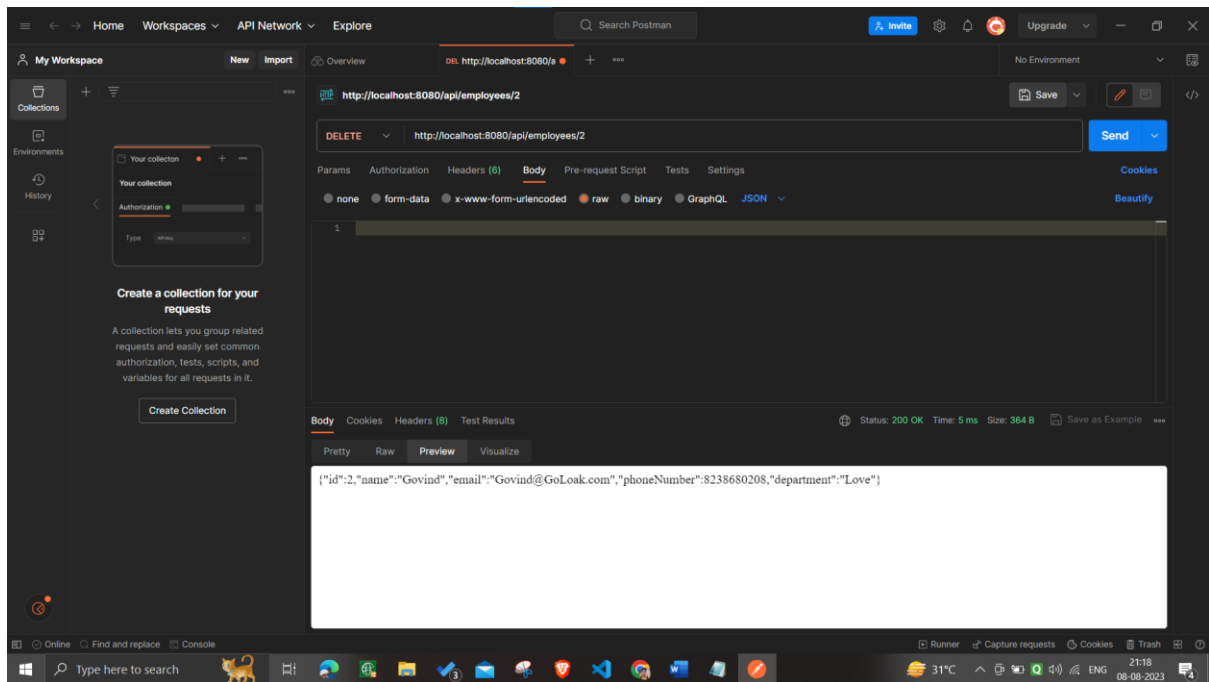
a) Add employees



b) Edit Employees



c) Delete Employees



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

App.js:

```
var express = require('express');
var path = require("path");
var bodyParser = require('body-parser');
var app = express();
var products = [
  { id: 1, name: 'SHREE RAM', email: 'ram@shaketdham.com',
    phoneNumber: '1234567890', department: 'HR' },
  { id: 2, name: 'Govind', email: 'Govind@GoLoak.com', phoneNumber:
    '0987654321', department: 'IT' }
];

app.use(bodyParser.urlencoded({extended: false}));
app.use(express.static(path.join(__dirname, './public')));
app.get('/', function(req, res){
  res.sendFile(path.join(__dirname, './public'));
});

app.get('/api/all', (req, res) => {

  res.send(products);
});

app.post('/api/get', (req, res) => {
  const product = products.find(({ id }) => id == req.body.id);

  if (!product)
    res.status(404).send('<h2 style="font-family: Malgun Gothic;
color: darkred;">Cant find what you are looking for!</h2>');
  res.send(product);
});

app.post('/api/employees', (req, res) => {
  var product = {
    id: req.body.id,
    name: req.body.name,
    email: req.body.email,
    phoneNumber: req.body.phoneNumber,
```

```

    department: req.body.department
  };
  products.push(product);
  res.send(product);
});

app.post('/api/put', (req, res) => {
  var product = products.find(({ id }) => id == req.body.id);
  if (!product) res.status(404).send('<h2 style="font-family: Malgun Gothic; color: darkred;">Not Found!! </h2>');
  product.name = req.body.name;
  product.email = req.body.fn;
  product.phoneNumber = req.body.ln;
  product.department = req.body.branch;
  res.send(product);
});

app.post('/api/delete', (req, res) => {

  const product = products.find(({ id }) => id == req.body.id);
  if (!product) res.status(404).send('<h2 style="font-family: Malgun Gothic; color: darkred;"> Not Found!! </h2>');

  const index = products.indexOf(product);
  products.splice(index, 1);

  res.send(product);
});
app.listen(8080);

```

Index.html code:

```

<!DOCTYPE html>
<html lang = "en">
<head>
  <meta charset = "UTF-8">
  <link rel = "stylesheet" href="style.css">

```

```

    <title> My Form </title>
</head>
<body>
    <header>
        <h1>Employee Database</h1>
    </header>
    <form action="/api/employees" method="POST">
        <fieldset>
            <h3>Add new Employee</h3>
<br><br>
            <label>Employee ID</label>
            <input type="number" id="id" name="id" required>
            <br><br>
            <label>Name</label>
            <input type="text" id="name" name="name" required>
            <br><br>
            <label>E-mail</label>
            <input type="email" id="email" name="email" required>
            <br><br>
            <label>Phone Number</label>
            <input type="number" id="phoneNumber" name="phoneNumber"
required>
            <br><br>
            <label>Department</label>
            <input type="text" id="department" name="department" required>
            <br><br>
            <button type="reset">Reset</button>
            <button type="submit">Submit</button>
        </fieldset>

    </form>
    <form action="/api/get" method="POST">
    <fieldset>
        <h3>View a Employee</h3>
        <br><br>
        <label>Employee ID</label>
        <input type="number" id="id" name="id" required>
        <br><br>
        <button type="reset">Reset</button>
        <button type="submit">Submit</button>
        <br><br><br>
    </fieldset>
    </form>
    <form action="/api/put" method="POST">
    <fieldset>

```



```

<h3>Update a Employee</h3>
<br><br>
  <label>Employee ID</label>
  <input type ="number" id = 'id' name="id" required>
  <br><br>
  <label>Name</label>
  <input type="text" id ="name" name="name" required>
  <br><br>
  <label>E-mail</label>
  <input type="email" id ="email" name="email" required>
  <br><br>
  <label>Phone Number</label>
  <input type="number" id ="phoneNumber" name="phoneNumber"
required>
  <br><br>
  <label>Department</label>
  <input type="text" id ="department" name="department" required>
  <br><br>

  <button type ="reset">Reset</button>
  <button type ="submit">Submit</button>
</fieldset>
</form>
<form action="/api/delete" method="POST">
  <fieldset>
    <h3>Delete a Employee</h3>
    <br><br>
    <label>Employee ID</label>
    <input type ="number" id = 'id' name="id" required>
    <br><br>
    <button type ="reset">Reset</button>
    <button type ="submit">Submit</button>
    <br><br><br>
  </fieldset>
</form>
<form action="/api/all" method="GET">
  <fieldset>
    <h3>Get all Employees data</h3>
    <br><br>
    <button type ="ALL Employees">Submit</button>
    <br><br><br>
  </fieldset>
</form>
</body>
</html>

```

Style.css:

```
/* Reset default margin and padding for all elements */
body, h1, h2, h3, p {
    margin: 0;
    padding: 0;
}

body {
    background-color: #f7f9fc;
    font-family: Arial, sans-serif;
}

header {
    text-align: center;
    height: 100px;
    padding-top: 20px;
    background-color: #5adeff;
    color: #ffffff;
    font-size: 24px;
}

h3 {
    margin-top: 20px;
    text-align: center;
    background-color: #34495e;
    padding: 10px 0;
    color: #ffffff;
    font-size: 18px;
}

form {
    width: 45%;
    margin: 20px auto;
    padding: 20px;
    background-color: #ffffff;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgb(95, 250, 188);
}

button {
```

```
display: block;
margin: 10px auto;
padding: 10px 20px;
background-color: #34c8db;
color: #ffffff;
border: none;
border-radius: 5px;
cursor: pointer;
transition: background-color 0.3s ease, transform 0.2s ease;
}

button:hover {
    background-color: #2980b9;
}

button:active {
    background-color: #1f6aa8;
    transform: translateY(2px);
}

#result {
    margin-top: 20px;
    padding: 20px;
    background-color: #f7f9fc;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    color: #8cff00;
    font-size: 14px;
}
```

OUTPUT:

Practical Task-4 x My Form x +

localhost:8080

Classes 2021_1st Semester... Ganpat University L... C-Operator and Ex... Convert decimal nu... HEXADECEMAL TO... Web Design Blog ... MEGA hacking cour... PDF to Word Conve... 24+ Awesome CSS...

Employee Database

Add new Employee

Employee ID

Name

E-mail

Phone Number

Department

View a Employee

Type here to search

31°C 00:00 09-08-2023

Practical Task-4 x My Form x +

localhost:8080

Classes 2021_1st Semester... Ganpat University L... C-Operator and Ex... Convert decimal nu... HEXADECEMAL TO... Web Design Blog ... MEGA hacking cour... PDF to Word Conve... 24+ Awesome CSS...

View a Employee

Employee ID

Update a Employee

Employee ID

Name

E-mail

Phone Number

Department

Type here to search

31°C 00:00 09-08-2023

Add new Employee:

Practical Task-4 x My Form x +

localhost:8080

Classes 2021_1st Semester... Ganpat University L... C-Operator and Ex... Convert decimal nu... HEXADECEMAL TO... Web Design Blog ~... MEGA hacking cour... PDF to Word Conve... 24+ Awesome CSS...

Employee Database

Add new Employee

Employee ID

Name

E-mail

Phone Number

Department

Practical Task-4 x localhost:8080/api/employees x +

localhost:8080/api/employees

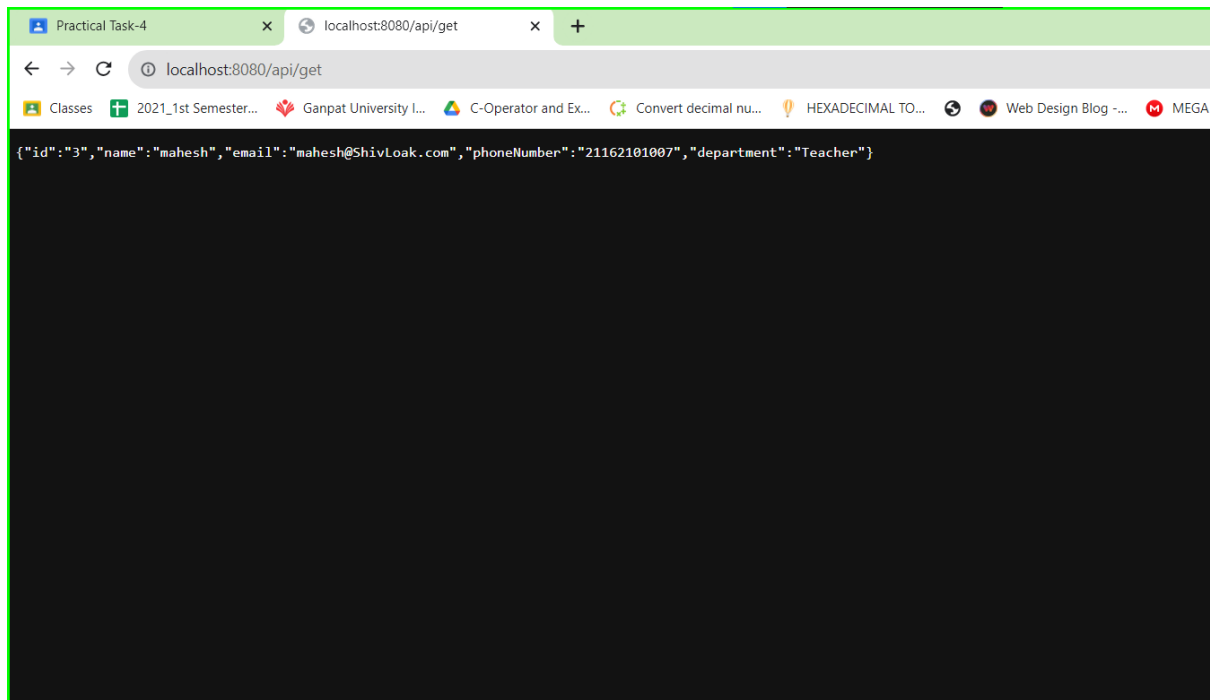
Classes 2021_1st Semester... Ganpat University L... C-Operator and Ex... Convert decimal nu... HEXADECEMAL TO... Web Design Blog ~... MEGA hacking cour... PDF to Word Conve... 24+ Awesc

```
{"id": "3", "name": "mahesh", "email": "mahesh@ShivLoak.com", "phoneNumber": "21162101007", "department": "Teacher"}
```

View an Employee:

View a Employee

Employee ID



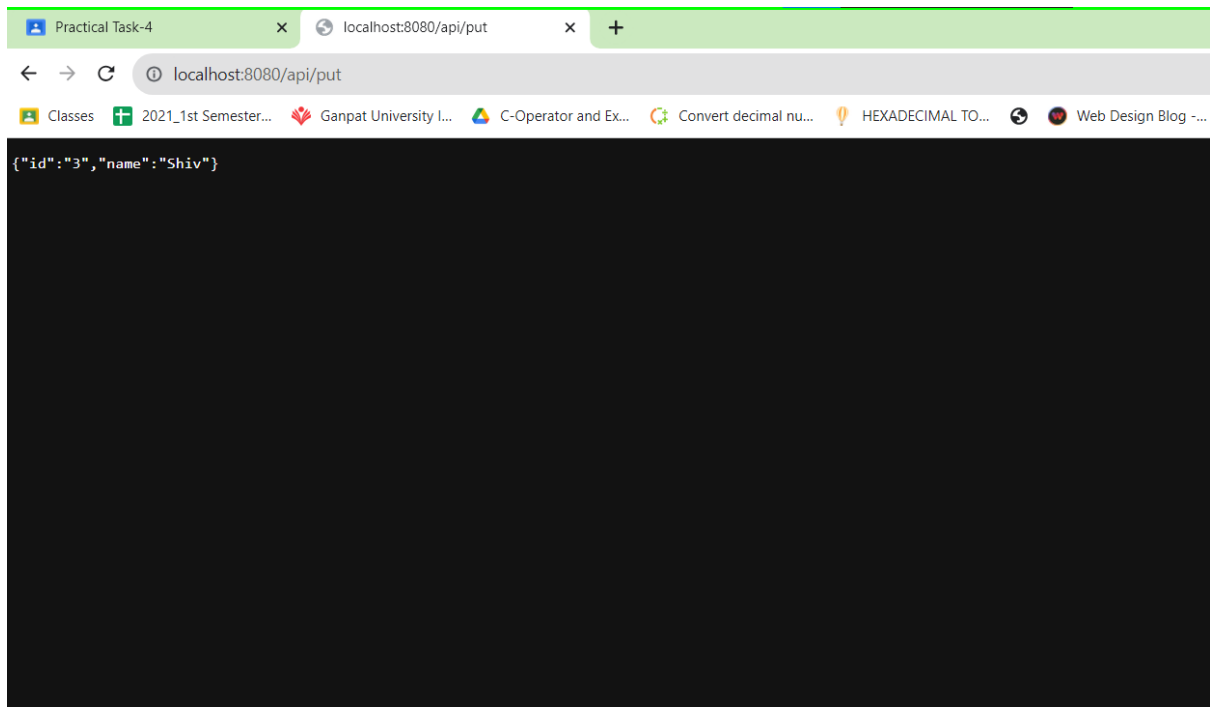
Update an Employee:

A screenshot of a web browser window showing a form titled 'Update a Employee'. The form contains the following fields and values:

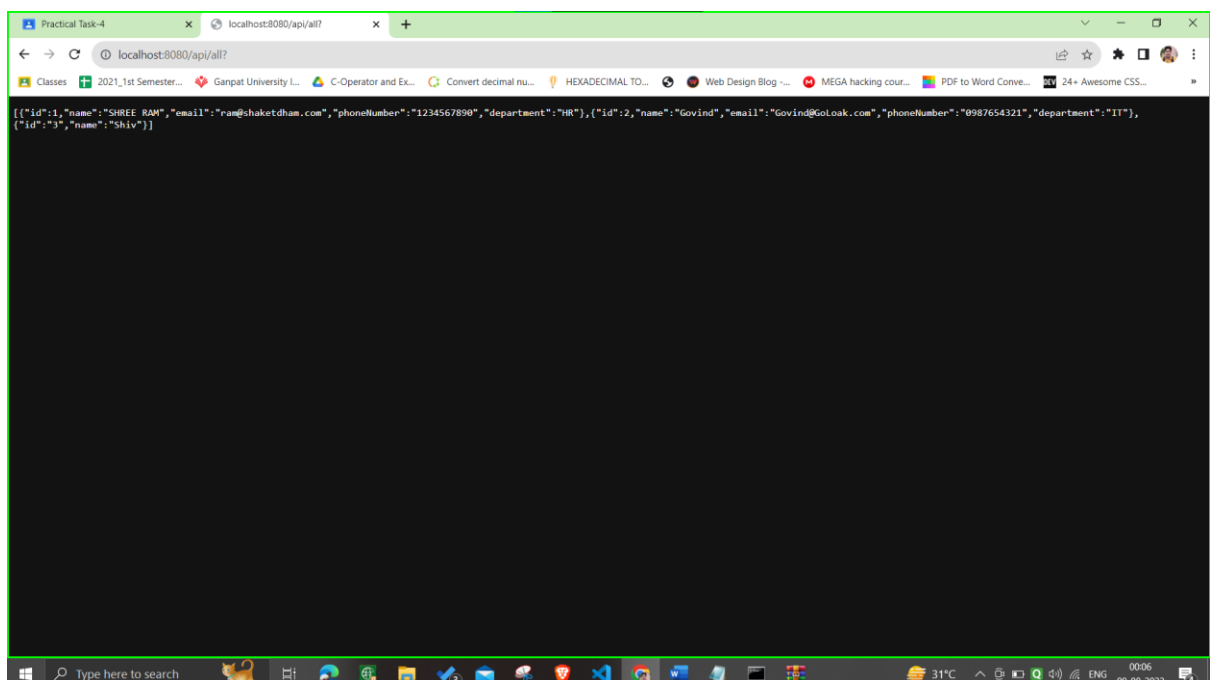
- Employee ID: 3
- Name: Shiv
- E-mail: mahesh@ShivLoak.com
- Phone Number: 21162101007
- Department: Teacher

Below the form are two buttons: 'Reset' and 'Submit'.

The browser's tab bar shows 'Practical Task-4' and 'My Form'. The address bar shows 'localhost:8080'. The bookmark bar includes 'Classes', '2021_1st Semester...', 'Ganpat University L...', 'C-Operator and Ex...', 'Convert decimal nu...', 'HEXADECIMAL TO...', 'Web Design Blog ~...', 'MEGA hacking cour...', 'PDF to Word Conve...', and '24+ Awesome'.



Get all Employees data:



Delete an Employee:

Delete a Employee

Employee ID

Reset

Submit

Practical Task-4

localhost:8080/api/all?

localhost:8080/api/all?

Classes 2021_1st Semester... Ganpat University L... C-Operator and Ex... Convert decimal nu... HEXADECIMAL TO... Web Design Blog ... MEGA hacking cour... PDF to Word Conve... 24+ Awesome CSS...

[[{"id":1,"name":"SHREE RAM","email":"ram@shaketdham.com","phoneNumber":"1234567890","department":"HR"}, {"id":2,"name":"Govind","email":"Govind@Goloak.com","phoneNumber":"0987654321","department":"IT"}]]