

PROGRAM:

Exercise 1.html

```
<html>

<head>

<script>

function VALIDATEDetail()

{

var name = document.forms["RegForm"]["Name"];
var email = document.forms["RegForm"]["EMail"];
var phone = document.forms["RegForm"]["Telephone"];
var what = document.forms["RegForm"]["Subject"];
var password = document.forms["RegForm"]["Password"];
var address = document.forms["RegForm"]["Address"];


    if (name.value == "")
    {
        window.alert("Please enter your name.");
        name.focus();
        return false;
    }


    if (address.value == "")
    {
        window.alert("Please enter your address.");
        name.focus();
        return false;
    }


    if (email.value == "")
    {
```

```
    window.alert("Please enter a valid e-mail address.");
    email.focus();
    return false;
}
```

```
if (email.value.indexOf("@", 0) < 0)
{
    window.alert("Please enter a valid e-mail address.");
    email.focus();
    return false;
}
```

```
if (email.value.indexOf(".", 0) < 0)
{
    window.alert("Please enter a valid e-mail address.");
    email.focus();
    return false;
}
```

```
if (phone.value == "")
{
    window.alert("Please enter your telephone number.");
    phone.focus();
    return false;
}
```

```
if (password.value == "")
{
    window.alert("Please enter your password");
    password.focus();
    return flase;
}
```

```
if (what.selectedIndex < 1)
{
    alert("Please enter your course.");
    what.focus();
    return false;
}
return true;
}</script>
```

```
<style>
VALIDATEDDETAIL {
    font-weight: bold ;
    float: left;
    width: 100px;
    text-align: left;
    margin-right: 10px;
    font-size:14px;
}
div {
box-sizing: border-box;
    width: 100%;
    border: 100px solid black;
    float: left;
    align-content: center;
    align-items: center;
}
```

```
form {
    margin: 0 auto;
    width: 600px;
```

```
}</style></head>
```

```
<body>
```

```
<h1 style="text-align: center;"> REGISTRATION FORM </h1>
```

```
<form name="RegForm" action="submit.php" onsubmit="return VALIDATEDetail()"
method="post">
```

```
<p>Name: <input type="text" size=65 name="Name"> </p><br>
```

```
<p> Address: <input type="text" size=65 name="Address"> </p><br>
```

```
<p>E-mail Address: <input type="text" size=65 name="EMail"> </p><br>
```

```
<p>Password: <input type="text" size=65 name="Password"> </p><br>
```

```
<p>Telephone: <input type="text" size=65 name="Telephone"> </p><br>
```

```
<p>SELECT YOUR COURSE
```

```
<select type="text" value="" name="Subject">
```

```
<option>BTECH</option>
```

```
<option>BBA</option>
```

```
<option>BCA</option>
```

```
<option>B.COM</option>
```

```
<option>VALIDATEDetail</option>
```

```
</select></p><br><br>
```

```
<p>Comments: <textarea cols="55" name="Comment"> </textarea></p>
```

```
<p><input type="submit" value="send" name="Submit">
```

```
<input type="reset" value="Reset" name="Reset">
```

```
</p>
```

```
</form>
```

```
</body>
```

```
</html>
```

Style.css

```
body {  
  font-family: sans-serif;  
  background-color: #f0f0f0; /* Light gray background */  
  margin: 0;  
  padding: 0;  
  display: flex;  
  flex-direction: column; /* Stack elements vertically */  
  align-items: center;  
  min-height: 100vh;  
}  
  
h1 {  
  text-align: center;  
  color: #333; /* Dark gray title */  
  margin-top: 20px;  
  margin-bottom: 20px;  
}  
  
form {  
  background-color: #fff; /* White form background */  
  padding: 20px;  
  border-radius: 8px;  
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);  
  width: 500px;  
  max-width: 90%;  
  margin-bottom: 20px; /* Space between form and buttons */  
}  
  
p {
```

```
margin-bottom: 10px;
color: #555;
}
```

```
input[type="text"],
select,
textarea {
  width: calc(100% - 22px);
  padding: 10px;
  margin-top: 5px;
  margin-bottom: 15px;
  border: 1px solid #ddd;
  border-radius: 4px;
  box-sizing: border-box;
}
```

```
select {
  height: 40px;
}
```

```
textarea {
  resize: vertical;
}
```

```
input[type="submit"],
input[type="reset"] {
  background-color: #007bff; /* Blue button */
  color: white;
  padding: 10px 15px;
  border: none;
```

```
border-radius: 4px;
cursor: pointer;
margin-right: 10px;
}
```

```
input[type="submit"]:hover,
input[type="reset"]:hover {
  background-color: #0056b3; /* Darker blue on hover */
}
```

```
/* Optional: Style the button container for better alignment */
.button-container {
  display: flex;
  justify-content: center;
  width: 100%;
}
```

Output:

The screenshot shows a web browser window with a single tab titled 'Exer1.html'. The address bar displays the file path 'D:/57_FSWD-LAB/EXERCISE1/Exer1.html'. The main content area features a 'REGISTRATION FORM' centered on a light gray background. The form is a white box with a light gray border, containing the following fields:

- Name:** A text input field containing 'sunil'.
- Address:** A text input field containing 'pothen'.
- E-mail Address:** A text input field containing 'sunil234@gmail.com'.
- Password:** A text input field containing 'sunil23'.
- Telephone:** A text input field containing '8418448455'.
- SELECT YOUR COURSE:** A dropdown menu with 'BCA' selected.
- Comments:** A text area with a small 'x' icon in the bottom right corner.

At the bottom of the form, there are two blue buttons: 'Send' and 'Reset'.

The Windows taskbar is visible at the bottom of the screen, showing the Start button, a search bar, and several application icons. The system tray on the right indicates a temperature of 32°C, 'Partly sunny' weather, the language 'ENG IN', and the date and time '13:58 03-04-2025'.

PROGRAM :

Exercise 2.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Fetch API Example</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      background-color: #08335c;

      display: flex;

      justify-content: center;

      align-items: center;

      height: 100vh;

      flex-direction: column;

    }

    .card {

      background: rgb(210, 212, 223);

      padding: 20px;

      border-radius: 10px;

      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

      margin: 10px;

      width: 300px;

      text-align: center;

    }

  </style>

</head>

<body>
```

```
<h2>Fetch API Data Display</h2>

<div id="card-container"></div>

<script>
  async function fetchData() {
    try {
      //const response = await fetch('https://jsonplaceholder.typicode.com/users');
      const response = await fetch('http://127.0.0.1:8081/users.json');
      const data = await response.json();

      const container = document.getElementById('card-container');
      container.innerHTML = ""; // Clear previous content
      data.slice(0, 5).forEach(user => {
        const card = document.createElement('div');
        card.classList.add('card');
        card.innerHTML = `
          <h3>${user.name}</h3>
          <p><strong>Email:</strong> ${user.email}</p>
          <p><strong>City:</strong> ${user.address.city}</p>
        `;
        container.appendChild(card);
      });
    } catch (error) {
      console.error('Error fetching data:', error);
    }
  }

  fetchData(); // Call function to fetch and display data
</script>

</body>

</html>
```

Users.json:

```
[
  {
    "id": 1,
    "name": "Sunil Aldo S A",
    "username": "sunil007",
    "email": "sunilado2004@srmvec.ac.in",
    "address": {
      "street": "Kulas Light",
      "suite": "Apt. 556",
      "city": "Potheri",
      "zipcode": "627005",
      "geo": {
        "lat": "-37.3159",
        "lng": "81.1496"
      }
    },
    "phone": "91-80729 22340",
    "website": "cits.org",
    "company": {
      "name": "Creative-Crona",
      "catchPhrase": "Multi-layered client-server neural-net",
      "bs": "harness real-time e-markets"
    }
  },
  {
    "id": 2,
    "name": "Sibi Karthick",
    "username": "Antonette",
    "email": "Shanna@melissa.tv",
```

```
"address": {
  "street": "Victor Plains",
  "suite": "Suite 879",
  "city": "Wisokyburgh",
  "zipcode": "90566-7771",
  "geo": {
    "lat": "-43.9509",
    "lng": "-34.4618"
  }
},
"phone": "010-692-6593 x09125",
"website": "anastasia.net",
"company": {
  "name": "Deckow-Crist",
  "catchPhrase": "Proactive didactic contingency",
  "bs": "synergize scalable supply-chains"
},
{
  "id": 3,
  "name": "Antro Jeffrie",
  "username": "Samantha",
  "email": "Nathan@yesenia.net",
  "address": {
    "street": "Douglas Extension",
    "suite": "Suite 847",
    "city": "McKenziehaven",
    "zipcode": "59590-4157",
    "geo": {
      "lat": "-68.6102",
```

```
    "lng": "-47.0653"
  }
},
"phone": "2-653-123-3337",
"website": "ramiro.info",
"company": {
  "name": "Romaguera-Jacobson",
  "catchPhrase": "Face to face bifurcated interface",
  "bs": "e-enable strategic applications"
}
},
{
  "id": 4,
  "name": "Fathima",
  "username": "Karianne",
  "email": "Julianne.OConner@kory.org",
  "address": {
    "street": "Hoeger Mall",
    "suite": "Apt. 692",
    "city": "South Elvis",
    "zipcode": "53919-4257",
    "geo": {
      "lat": "29.4572",
      "lng": "-164.2990"
    }
  },
  "phone": "493-170-9623 x156",
  "website": "kale.biz",
  "company": {
    "name": "Robel-Corkery",
```

```
    "catchPhrase": "Multi-tiered zero tolerance productivity",
    "bs": "transition cutting-edge web services"
  },
  {
    "id": 5,
    "name": "Ragul Gandhi",
    "username": "Kamren",
    "email": "Lucio_Hettinger@annie.ca",
    "address": {
      "street": "Skiles Walks",
      "suite": "Suite 351",
      "city": "Roscoeview",
      "zipcode": "33263",
      "geo": {
        "lat": "-31.8129",
        "lng": "62.5342"
      }
    },
    "phone": "(254)954-1289",
    "website": "demarco.info",
    "company": {
      "name": "Keebler LLC",
      "catchPhrase": "User-centric fault-tolerant solution",
      "bs": "revolutionize end-to-end systems"
    }
  },
  {
    "id": 6,
    "name": "Mrs. Dennis Schulist",
```

```
"username": "Leopoldo_Corkery",
"email": "Karley_Dach@jasper.info",
"address": {
  "street": "Norberto Crossing",
  "suite": "Apt. 950",
  "city": "South Christy",
  "zipcode": "23505-1337",
  "geo": {
    "lat": "-71.4197",
    "lng": "71.7478"
  }
},
"phone": "1-477-935-8478 x6430",
"website": "ola.org",
"company": {
  "name": "Considine-Lockman",
  "catchPhrase": "Synchronised bottom-line interface",
  "bs": "e-enable innovative applications"
},
{
  "id": 7,
  "name": "Kurtis Weissnat",
  "username": "Elwyn.Skiles",
  "email": "Telly.Hoeger@billy.biz",
  "address": {
    "street": "Rex Trail",
    "suite": "Suite 280",
    "city": "Howemouth",
    "zipcode": "58804-1099",
```

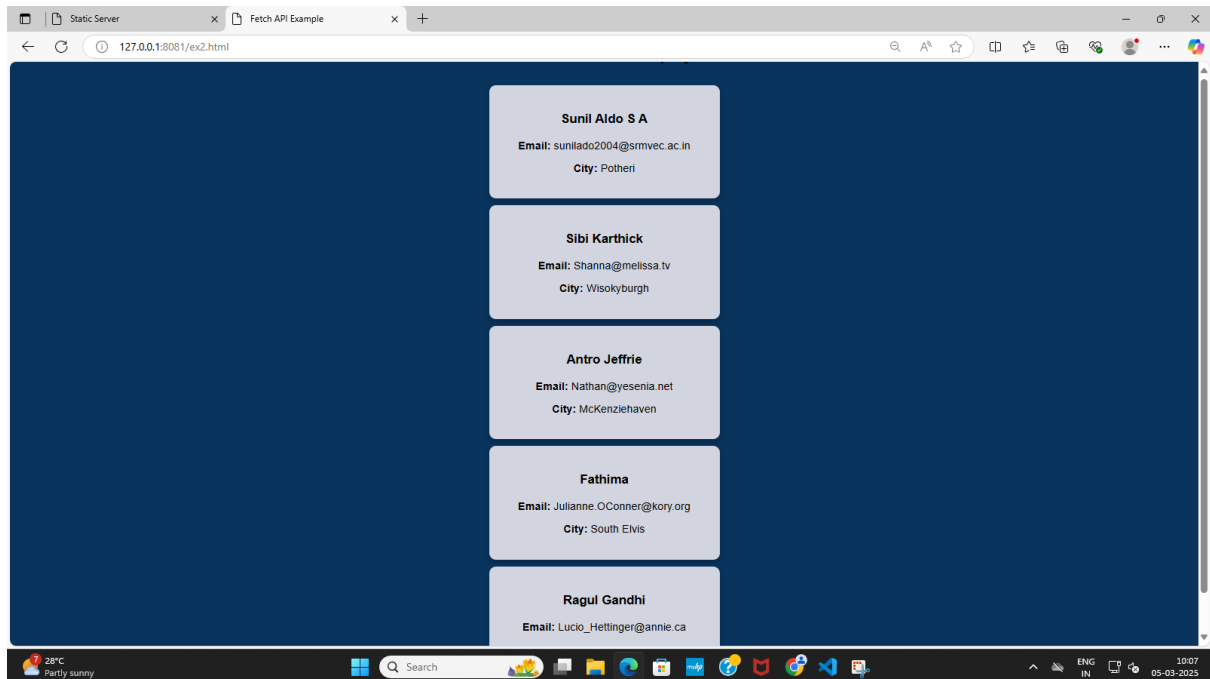
```
"geo": {
  "lat": "24.8918",
  "lng": "21.8984"
},
"phone": "210.067.6132",
"website": "elvis.io",
"company": {
  "name": "Johns Group",
  "catchPhrase": "Configurable multimedia task-force",
  "bs": "generate enterprise e-tailers"
},
{
  "id": 8,
  "name": "Nicholas Runolfsdottir V",
  "username": "Maxime_Nienow",
  "email": "Sherwood@rosamond.me",
  "address": {
    "street": "Ellsworth Summit",
    "suite": "Suite 729",
    "city": "Aliyaview",
    "zipcode": "45169",
    "geo": {
      "lat": "-14.3990",
      "lng": "-120.7677"
    }
  },
  "phone": "586.493.6943 x140",
  "website": "jacynthe.com",
```



```
"company": {
  "name": "Abernathy Group",
  "catchPhrase": "Implemented secondary concept",
  "bs": "e-enable extensible e-tailers"
},
{
  "id": 9,
  "name": "Glenna Reichert",
  "username": "Delphine",
  "email": "Chaim_McDermott@dana.io",
  "address": {
    "street": "Dayna Park",
    "suite": "Suite 449",
    "city": "Bartholomebury",
    "zipcode": "76495-3109",
    "geo": {
      "lat": "24.6463",
      "lng": "-168.8889"
    }
  },
  "phone": "(775)976-6794 x41206",
  "website": "conrad.com",
  "company": {
    "name": "Yost and Sons",
    "catchPhrase": "Switchable contextually-based project",
    "bs": "aggregate real-time technologies"
  },
  {
```

```
"id": 10,
"name": "Clementina DuBuque",
"username": "Moriah.Stanton",
"email": "Rey.Padberg@karina.biz",
"address": {
  "street": "Kattie Turnpike",
  "suite": "Suite 198",
  "city": "Lebsackbury",
  "zipcode": "31428-2261",
  "geo": {
    "lat": "-38.2386",
    "lng": "57.2232"
  }
},
"phone": "024-648-3804",
"website": "ambrose.net",
"company": {
  "name": "Hoeger LLC",
  "catchPhrase": "Centralized empowering task-force",
  "bs": "target end-to-end models"
}
}
```

Output:



PROGRAM:

Exercise 3.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Static Server</title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <h1>Welcome to My Node.js Server</h1>

  <p>This is a static HTML page served without Express.</p>

</body>

</html>
```

Ex_3_server.js

```
const http = require('http');
const fs = require('fs');
const path = require('path');

const PORT = 3000;
const PUBLIC_DIR = path.join(__dirname, 'public');

// MIME types for various file extensions
const mimeTypes = {
  '.html': 'text/html',
  '.css': 'text/css',
  '.js': 'text/javascript',
  '.json': 'application/json',
```

```
.jpg': 'image/jpeg',
.jpeg': 'image/jpeg',
.png': 'image/png',
.gif': 'image/gif',
.svg': 'image/svg+xml',
.ico': 'image/x-icon',
.woff': 'font/woff',
.woff2': 'font/woff2',
.ttf': 'font/ttf',
.eot': 'application/vnd.ms-fontobject',
};
```

```
const server = http.createServer((req, res) => {
  let filePath = path.join(PUBLIC_DIR, req.url === '/' ? 'ex_3.html' : req.url);

  // Get file extension
  let extname = path.extname(filePath).toLowerCase();

  // Default to text/html if file extension not found
  let contentType = mimeTypes[extname] || 'text/html';

  // Read and serve the file
  fs.readFile(filePath, (err, content) => {
    if (err) {
      if (err.code === 'ENOENT') {
        res.writeHead(404, { 'Content-Type': 'text/html' });
        res.end('<h1>404 - Not Found</h1>');
      } else {
        res.writeHead(500);
        res.end('<h1>500 - Server Error</h1><p>${err.code}</p>');
      }
    }
  });
});
```

```

    }
  } else {
    res.writeHead(200, { 'Content-Type': contentType });
    res.end(content, 'utf-8');
  }
});
});

```

```

server.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});

```

Server.js

```

const http = require('http');
const fs = require('fs');
const path = require('path');

const PORT = 3000;
const PUBLIC_DIR = path.join(__dirname, 'public');

const server = http.createServer((req, res) => {
  let filePath = path.join(PUBLIC_DIR, req.url === '/' ? 'ex_3.html' : req.url);

  // Get file extension
  let extname = path.extname(filePath);

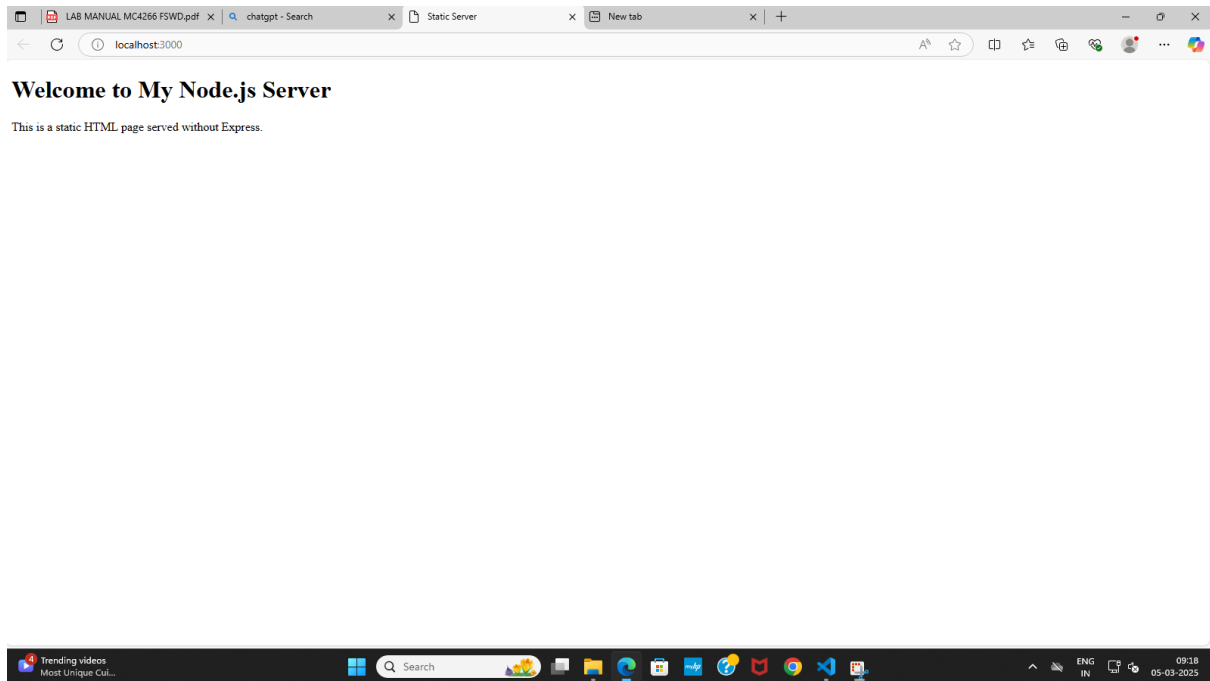
  // Set content type
  let contentType = 'text/html';
  if (extname === '.css') contentType = 'text/css';
  if (extname === '.js') contentType = 'text/javascript';

```

```
// Read and serve the file
fs.readFile(filePath, (err, content) => {
  if (err) {
    if (err.code === 'ENOENT') {
      res.writeHead(404, { 'Content-Type': 'text/html' });
      res.end('<h1>404 - Not Found</h1>');
    } else {
      res.writeHead(500);
      res.end(`Server Error: ${err.code}`);
    }
  } else {
    res.writeHead(200, { 'Content-Type': contentType });
    res.end(content, 'utf-8');
  }
});

server.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});
```

Output:



PROGRAM:

Exercise4.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Form Submission</title>

</head>

<body>

  <h2>Submit Your Details</h2>

  <form action="/submit" method="post">

    <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>

    <input type="submit" value="Submit">

  </form>

</body>

</html>
```

Ex_4_Server.js

```
const express = require('express');
const bodyParser = require('body-parser');
const fs = require('fs');

const app = express();
const PORT = 3000;
```

```
// Middleware to parse form data
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());

// Serve the HTML form (optional)
app.get('/', (req, res) => {
  res.sendFile(__dirname + '/index.html');
});

// Handle form submission
app.post('/submit', (req, res) => {
  const formData = req.body;

  // Read existing data
  let existingData = [];
  if (fs.existsSync('data.json')) {
    const fileContent = fs.readFileSync('data.json');
    existingData = JSON.parse(fileContent);
  }

  // Add new data
  existingData.push(formData);

  // Save to JSON file
  fs.writeFileSync('data.json', JSON.stringify(existingData, null, 2));

  res.send('Form data saved successfully!');
});
```

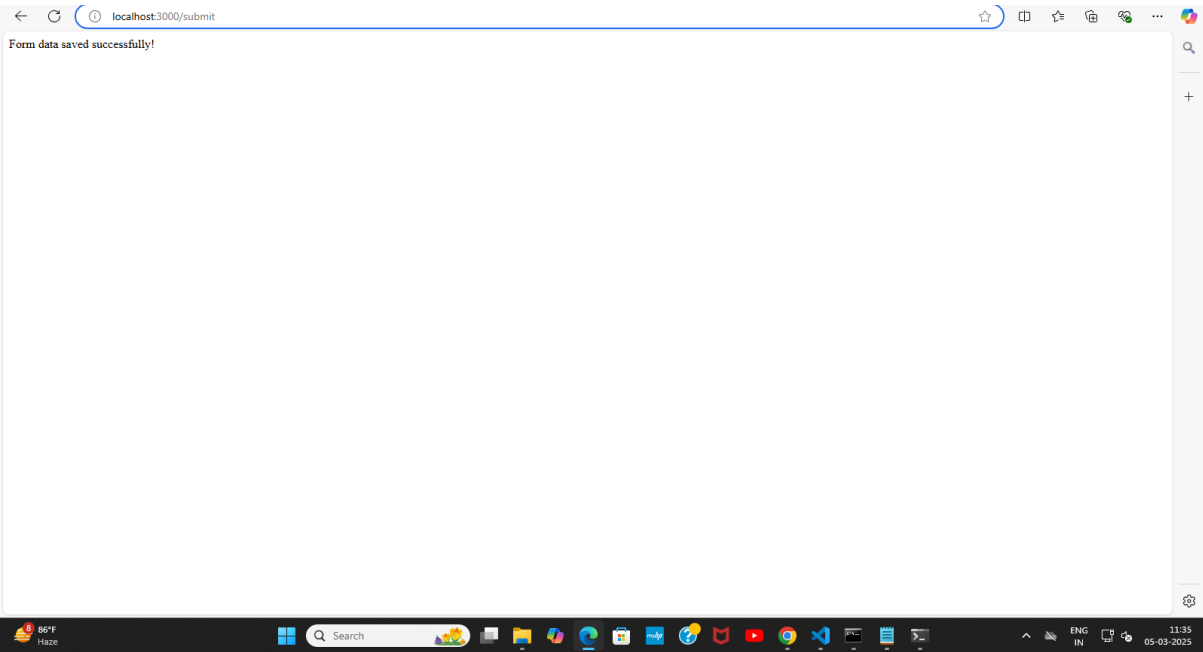
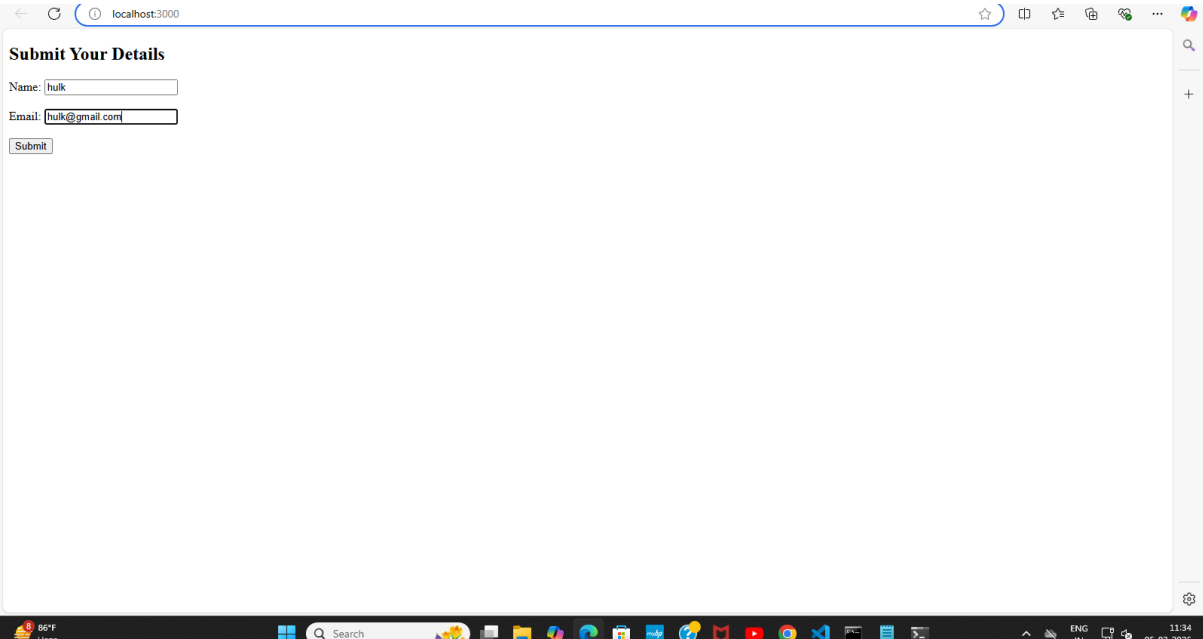
```
// Start the server

app.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});
```

Data.json

```
[
  {
    "name": "Hulk",
    "email": "Hulk@gmail.com"
  },
  {
    "name": "Hulk",
    "email": "Hulk@gmail.com"
  }
]
```

Output:



PROGRAM:

Exercise 5.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Customer CRUD App</title>


    <link rel="stylesheet" href="ex_5_style.css"> <!-- Link the CSS file -->

</head>

<body>


    <h2>Customer Management System</h2>


    <form id="customerForm">

        <input type="text" id="name" placeholder="Enter Name" required>

        <input type="text" id="city" placeholder="Enter City" required>

        <input type="text" id="mobile" placeholder="Enter Mobile No" required>

        <button type="submit">Add Customer</button>

    </form>


    <table>

        <thead>

            <tr>

                <th>Name</th>

                <th>City</th>

                <th>Mobile No</th>

                <th>Actions</th>

            </tr>
```

```

</thead>

<tbody id="customersList"></tbody>
</table>

<script>
  const API_URL = 'http://localhost:5000/customers';

  async function fetchCustomers() {
    const res = await fetch(API_URL);
    const customers = await res.json();

    document.getElementById('customersList').innerHTML = customers.map(customer
=> `
      <tr>
        <td><input type="text" value="${customer.name}" id="name-
${customer._id}"></td>
        <td><input type="text" value="${customer.city}" id="city-
${customer._id}"></td>
        <td><input type="text" value="${customer.mobile}" id="mobile-
${customer._id}"></td>
        <td>
          <button onclick="updateCustomer('${customer._id}')">Update</button>
          <button onclick="deleteCustomer('${customer._id}')">Delete</button>
        </td>
      </tr>
    `).join("");
  }

  async function addCustomer(event) {
    event.preventDefault();

    const name = document.getElementById('name').value;
    const city = document.getElementById('city').value;
    const mobile = document.getElementById('mobile').value;

```

```

    await fetch(API_URL, {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ name, city, mobile })
    });

    document.getElementById('customerForm').reset();
    fetchCustomers();
  }

  async function updateCustomer(id) {
    const name = document.getElementById(`name-${id}`).value;
    const city = document.getElementById(`city-${id}`).value;
    const mobile = document.getElementById(`mobile-${id}`).value;

    await fetch(`${API_URL}/${id}`, {
      method: 'PUT',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({ name, city, mobile })
    });

    fetchCustomers();
  }

  async function deleteCustomer(id) {
    await fetch(`${API_URL}/${id}`, { method: 'DELETE' });
    fetchCustomers();
  }

```

```
        document.getElementById('customerForm').addEventListener('submit', addCustomer);

        fetchCustomers();

    </script>

</body>

</html>
```

Exercise5 style.css

```
/* General Page Styling */

body {

    font-family: Arial, sans-serif;

    text-align: center;

    background-color: #f4f4f4;

    margin: 0;

    padding: 0;

}


/* Form Styling */

form {

    margin: 20px auto;

    width: 50%;

    padding: 15px;

    background: white;

    border-radius: 8px;

    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);

}


input {

    padding: 10px;

    margin: 5px;
```



```
width: 80%;  
border: 1px solid #ccc;  
border-radius: 5px;  
}  
  
/* Table Styling */  
table {  
    width: 60%;  
    margin: 20px auto;  
    border-collapse: collapse;  
    background: white;  
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);  
}  
  
th, td {  
    padding: 10px;  
    border: 1px solid #ddd;  
    text-align: center;  
}  
  
th {  
    background-color: #007bff;  
    color: white;  
}  
  
/* Button Styling */  
button {  
    padding: 8px 12px;  
    border: none;  
    cursor: pointer;
```

```
    color: white;
    border-radius: 5px;
}
```

```
button:hover {
    opacity: 0.8;
}
```

```
button:nth-child(1) { background-color: #28a745; } /* Update button - Green */
button:nth-child(2) { background-color: #dc3545; } /* Delete button - Red */
```

Ex_5_server.js

```
require('dotenv').config();
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');

const app = express();
app.use(express.json());
app.use(cors());

const MONGO_URI = process.env.MONGO_URI;
if (!MONGO_URI) {
    console.error("✗ MONGO_URI is not set in .env file");
    process.exit(1);
}

mongoose.connect(MONGO_URI, { useNewUrlParser: true, useUnifiedTopology: true })
    .then(() => console.log('✔ MongoDB Connected'))
    .catch(err => console.error(err));
```

```
// Customer Schema
```

```
const CustomerSchema = new mongoose.Schema({  
  name: { type: String, required: true },  
  city: { type: String, required: true },  
  mobile: { type: String, required: true }  
});  
  
const Customer = mongoose.model('Customer', CustomerSchema);
```

```
// Create
```

```
app.post('/customers', async (req, res) => {  
  try {  
    const newCustomer = new Customer(req.body);  
    await newCustomer.save();  
    res.json(newCustomer);  
  } catch (err) {  
    res.status(500).json({ error: err.message });  
  }  
});
```

```
// Read
```

```
app.get('/customers', async (req, res) => {  
  const customers = await Customer.find();  
  res.json(customers);  
});
```

```
// Update
```

```
app.put('/customers/:id', async (req, res) => {  
  try {  
    const updatedCustomer = await Customer.findByIdAndUpdate(req.params.id, req.body,  
    { new: true });
```

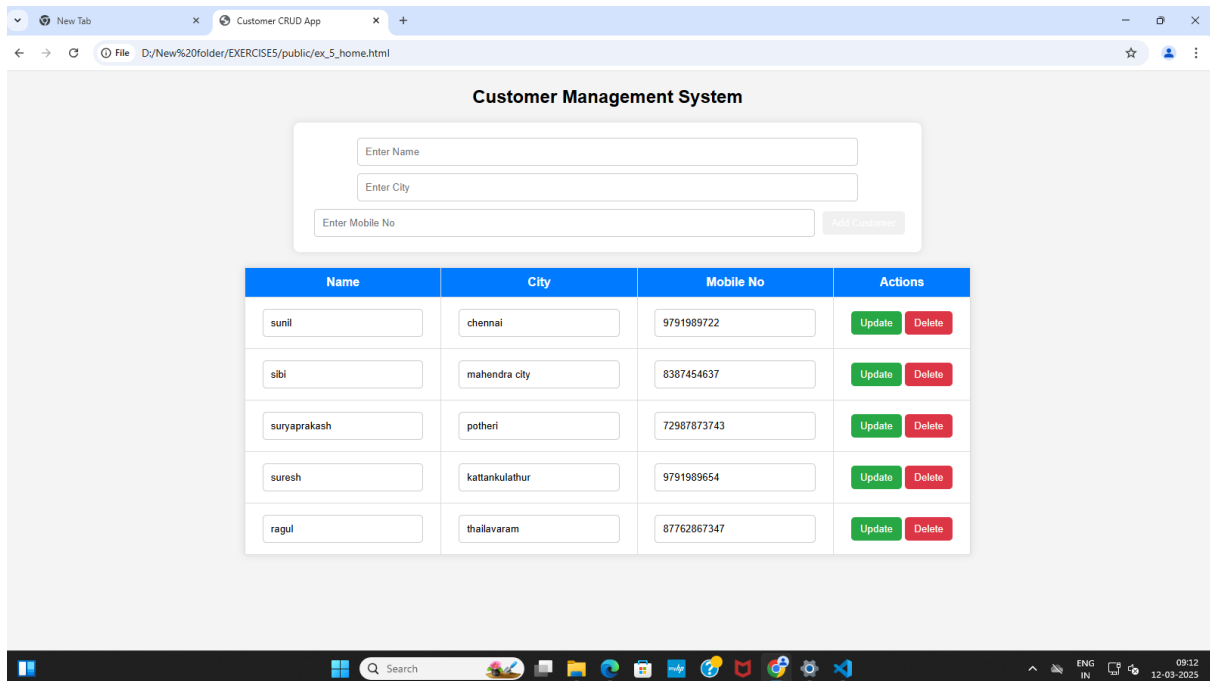
```
        res.json(updatedCustomer);
    } catch (err) {
        res.status(500).json({ error: err.message });
    }
});
```

// Delete

```
app.delete('/customers/:id', async (req, res) => {
    try {
        await Customer.findByIdAndDelete(req.params.id);
        res.json({ message: 'Customer deleted' });
    } catch (err) {
        res.status(500).json({ error: err.message });
    }
});
```

```
app.listen(5000, () => console.log('🚀 Server running on port 5000'));
```

Output:



The screenshot displays a web browser window with two tabs: 'New Tab' and 'Customer CRUD App'. The address bar shows the file path 'D:/New%20folder/EXERCISES/public/ex_5_home.html'. The main content area is titled 'Customer Management System' and features a form for adding a new customer. The form has three input fields: 'Enter Name', 'Enter City', and 'Enter Mobile No', followed by an 'Add Customer' button. Below the form is a table with five rows of customer data. Each row contains the customer's name, city, mobile number, and two buttons: 'Update' (green) and 'Delete' (red).

Name	City	Mobile No	Actions
<input type="text" value="sunil"/>	<input type="text" value="chennai"/>	<input type="text" value="9791989722"/>	<input type="button" value="Update"/> <input type="button" value="Delete"/>
<input type="text" value="sibi"/>	<input type="text" value="mahendra city"/>	<input type="text" value="8387454637"/>	<input type="button" value="Update"/> <input type="button" value="Delete"/>
<input type="text" value="suryaprakash"/>	<input type="text" value="potheri"/>	<input type="text" value="72987873743"/>	<input type="button" value="Update"/> <input type="button" value="Delete"/>
<input type="text" value="suresh"/>	<input type="text" value="kattankulathur"/>	<input type="text" value="9791989654"/>	<input type="button" value="Update"/> <input type="button" value="Delete"/>
<input type="text" value="ragul"/>	<input type="text" value="thailavaram"/>	<input type="text" value="87762867347"/>	<input type="button" value="Update"/> <input type="button" value="Delete"/>

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right indicates the language is 'ENG IN' and the date and time are '09:12 12-03-2025'.