

server.js

```
const express = require('express');
const cors = require('cors'); const app
= express();      app.use(cors());
app.use(express.json());
app.use(express.static('public'));
let contacts = [
  { id: 1, name: "Vidhya", email: "vidhya@gmail.com", phone: "9876543210" },
  { id: 2, name: "Karthik", email: "karthik@gmail.com", phone: "9876500000" }
];

// Get all contacts
app.get('/api/contacts', (req, res) => {
  res.json(contacts);
});
```

```
// Add new contact
app.post('/api/contacts', (req, res) => {
const newContact = {      id: Date.now(),
name: req.body.name,      email:
req.body.email,          phone: req.body.phone
};      contacts.push(newContact);      res.json({ message:
'Contact added successfully', contact: newContact });
});

// Update contact  app.put('/api/contacts/:id', (req, res) => {
const contactId = parseInt(req.params.id);      const index =
contacts.findIndex(c => c.id === contactId);      if (index !== -
1) {      contacts[index] = { ...contacts[index], ...req.body
};      res.json({ message: 'Contact updated', contact:
contacts[index] });
} else {      res.status(404).json({ message:
'Contact not found' });
}
});

// Delete contact  app.delete('/api/contacts/:id', (req,
res) => {  const contactId = parseInt(req.params.id);
contacts = contacts.filter(c => c.id !== contactId);
res.json({ message:
'Contact deleted successfully' });
});

const PORT = 5000;  app.listen(PORT, () => console.log(`Server running on
http://localhost:${PORT}`));
```

public/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>RESTful Contact Management API</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <h1>Contact Management System</h1>
  <div class="form-container">
    <input type="text" id="name" placeholder="Enter Name" />
    <input type="email" id="email" placeholder="Enter Email" />
    <input type="text" id="phone" placeholder="Enter Phone" />
    <button onclick="addContact()">Add Contact</button>
  </div>
  <h2>Contact List</h2>
  <ul id="contactList"></ul>
<script src="script.js"></script> </body>
```

</html>

public/script.js

```
const apiURL = '/api/contacts';    async function
fetchContacts() {    const res = await
fetch(apiURL);    const data = await res.json();
const list =
document.getElementById('contactList');
list.innerHTML = '';    data.forEach(contact => {
const li = document.createElement('li');
li.innerHTML = `
    <strong>${contact.name}</strong> - ${contact.email} - ${contact.phone}
    <button onclick="deleteContact(${contact.id})">Delete</button>
`;
list.appendChild(li);
});    }    async function addContact() {
const name =
document.getElementById('name').value;    const
email
= document.getElementById('email').value;    const
phone = document.getElementById('phone').value;
if (!name || !email ||
!phone) {        alert('Please fill all
fields');        return;
    }    await fetch(apiURL, {        method:
'POST',        headers: { 'Content-Type':
'application/json' },        body: JSON.stringify({
name, email, phone })
    });
document.getElementById('name').value = '';
document.getElementById('email').value = '';
document.getElementById('phone').value
= '';
fetchContacts();
}    async function deleteContact(id) {
await fetch(`${apiURL}/${id}`, { method:
'DELETE' });    fetchContacts();
}
fetchContacts();
```

public/style.css

```
body {    font-family: Arial, sans-  
serif;    text-align: center;  
background-color:  
#f4f4f4;
```

```

}    h1 {
color: #2d6cdf;
}

.form-container {
margin: 20px;
}    input {    padding:
8px;    margin: 5px;
border: 1px solid #ccc;
borderradius: 5px;
}
button
{    padding: 8px 12px;
background-color: #2d6cdf;    color:
white;    border: none;    border-
radius: 5px;    cursor:
pointer;    }
button:hover {
background-color: #1a4cad;
}    ul {    liststyle:
none;    padding: 0;
}    li {    background:
#fff;    margin: 10px
auto;    padding: 10px;
width: 300px;
borderradius:
5px;    box-shadow: 0 0 5px #ccc;
}

```

:

