CHAPTER – 11

Source Code

Source Code

Front - end Code

Index.html(signup page)

```
 <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Signup</title>
    <style>
        /* Paste your CSS code here */
        @import url("https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;
            500;600;700;800;900&display=swap");

        * {
          margin: 0;
          padding: 0;
          box-sizing: border-box;
          font-family: "Poppins", sans-serif;
        }

        body {
          display: flex;
```

```
justify-content: center;
align-items: center;
min-height: 100vh;
background: #223243;
}

.container {
padding: 40px;
border-radius: 60px;
border: 6px solid #fff; /* Changed border color to white */
box-shadow: -5px -5px 15px rgba(255, 255, 255, 0.1),
5px 5px 15px rgba(0, 0, 0, 0.35),
inset -5px -5px 15px rgba(255, 255, 255, 0.1),
inset 5px 5px 15px rgba(0, 0, 0, 0.35);
max-width: 900px; /* Adjust this value based on your preference */
}

.container h1 {
color: #13d3da;
font-weight: 500;
letter-spacing: 0.1em;
text-align: center;
margin-bottom: 20px;
}

.container form input {
padding: 15px;
border: none;
```

```
width: 100%;
background:  #223243; /* Changed background color to white */
border: 1px solid rgba(0, 0, 0, 0.1);

color:white; /* Changed text color to dark */
font-weight: 600;
          border-radius: 25px;
          font-size: 1.5em;
          margin-bottom: 20px;
          outline: none;
        }

        .container form button {
          padding: 15px;
          border: none;
          width: 100%;
          background: #00dfc4;
          color: #223243;
          font-weight: 500;
          border-radius: 25px;
          font-size: 1em;
          cursor: pointer;
          transition: background 0.3s ease-in-out;
        }

        .container form button:hover {
          background: #00c4ac;
        }
        .container a {
          color: #fff;
          text-decoration: none;
          text-align: center;
          display: block;
        }

        .container a:hover {
          text-decoration: underline;
        }

        .container form input:hover,
.container form button:hover {
  background: #00c4ac;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
}

.container form input {
  font-size: 1.2em;
}

.container form button {
  font-size: 1.2em;
}

.container form input:valid {
  border-color: #00dfc4;
}
```

```
.container form input:invalid {
  border-color:white;
}

.container form input[type="submit"] {
  transition: background 0.3s ease-in-out, box-shadow 0.3s ease-in-out;
}

.container a {
  font-size: 1em;
}

.container a:hover {
  text-decoration: underline;
}

    </style>
</head>
<body>
    <div class="container">
        <h1>Welcome to Contact Management System</h1> <!-- Added heading -->
        <form action="/signup" method="POST">
            <input type="text" name="name" placeholder="Name" required><br>
            <input type="tel" name="mobile" placeholder="Mobile Number" required><br>
            <input type="email" name="email" placeholder="Email" required><br>
            <input type="password" name="password" placeholder="Password" required><br>
            <button type="submit">Sign Up</button>
        </form>
```

```
 <!-- Add a link to navigate to the login page -->
        <a href="/login">Already have an account? Login here</a>
    </div>

</body>
</html>
```

Login.html(login page)

```
 <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <style>
            @import
url("https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;
600;700;800;900&display=swap");

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}

body {
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  background: #223243;
}

.container {
  padding: 50px;
  border-radius: 60px;
  border: 6px solid #fff; /* Changed border color to white */
  box-shadow: -5px -5px 15px rgba(255, 255, 255, 0.1),
  5px 5px 15px rgba(0, 0, 0, 0.35),
    inset -5px -5px 15px rgba(255, 255, 255, 0.1),
    inset 5px 5px 15px rgba(0, 0, 0, 0.35);
  max-width: 600px; /* Adjust this value based on your preference */
}

.container h1 {
  color: #13d3da;
  font-weight: 500;
  letter-spacing: 0.1em;
  text-align: center;
  margin-bottom: 20px;
}


.container form input {
  padding: 15px;
  border: none;
  width: 100%;
  background:  #223243; /* Changed background color to white */
  border: 1px solid rgba(0, 0, 0, 0.1);

  color:white; /* Changed text color to dark */
  font-weight: 600;
  border-radius: 25px;
  font-size: 1.5em;
  margin-bottom: 20px;
  outline: none;
}

.container form button {
  padding: 15px;
  border: none;
  width: 100%;
```

```
 background: #00dfc4;
  color: #223243;
  font-weight: 500;
  border-radius: 25px;
  font-size: 1em;
  cursor: pointer;
  transition: background 0.3s ease-in-out;
}

        .container form button:hover {
          background: #00c4ac;
        }

        .container a {
          color: #fff;
          text-decoration: none;
          text-align: center;
          display: block;
        }

        .container a:hover {
          text-decoration: underline;
        }

        .container form input:hover,
        .container form button:hover {
          background: #00c4ac;
          box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
        }

        .container form input {
          font-size: 1.2em;
        }
```

```
.container form button {
  font-size: 1.2em;
}

.container form input:valid {
  border-color: #00dfc4;
}

.container form input:invalid {
  border-color:white;
}

.container form input[type="submit"] {
  transition: background 0.3s ease-in-out, box-shadow 0.3s ease-in-out;
}

.container a {
  font-size: 1em;
}

.container a:hover {
  text-decoration: underline;
}



/* CSS for captcha */
.captcha {
    margin-bottom: 20px;
    text-align: center;
    font-size: 1.5em;
    font-weight: bold;
    cursor: pointer;
    background-color: #f0f0f0;
```

```
 border-radius: 5px;
    padding: 10px;
    width: 200px;
    display: inline-block;
    transition: background-color 0.3s ease;
}

.captcha:hover {
    background-color: #e0e0e0;
}

#captcha-input {
    margin-top: 10px;
    padding: 10px;
    font-size: 1em;
    border: 1px solid #ccc;
    border-radius: 5px;
    width: 200px;
}

    </style>
</head>
<body>
    <div class="container">
        <h1>Welcome to Contact Management System</h1>

        <form id="loginForm" action="/login" method="POST">
            <input type="text" name="name" placeholder="Name" required><br>
            <input type="password" name="password" placeholder="Password" required><br>

            <!-- Display captcha -->
            <div class="captcha" id="captcha-container"></div>
            <input type="text" id="captcha-input" placeholder="Enter Captcha" required><br>
```

```
  <button type="submit" id="loginBtn" onclick="login()">Login</button>

        </form>

        <a href="/">Don't have an account? Sign Up here</a>
    </div>

    <script>


        // Function to generate random captcha
        function generateCaptcha(length) {
            const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
            let captcha = '';
            for (let i = 0; i < length; i++) {
                captcha += characters.charAt(Math.floor(Math.random() * characters.length));
            }
            return captcha;
        }

        // Function to update captcha in the HTML
        function updateCaptcha() {
            const captchaContainer = document.getElementById('captcha-container');
            const captcha = generateCaptcha(6); // Change the length of captcha as per your preference
            captchaContainer.innerText = captcha;
        }

        // Execute this function after the page is loaded
        document.addEventListener("DOMContentLoaded", function() {
            updateCaptcha(); // Generate initial captcha

            // Add event listener to regenerate captcha on click
  document.getElementById('captcha-container').addEventListener('click', updateCaptcha);
        });

        // Add event listener for form submission
document.getElementById("loginForm").addEventListener("submit", function(event) {
    event.preventDefault(); // Prevent default form submission

    // Validate captcha
    const enteredCaptcha = document.getElementById('captcha-input').value;
    const generatedCaptcha = document.getElementById('captcha-container').innerText;
    if (enteredCaptcha === generatedCaptcha) {
        // Redirect to cms.html after successful login
        window.location.href = "cms.html";
    } else {
        alert("Captcha incorrect. Please try again.");
        // Regenerate captcha
        updateCaptcha();
        // Clear captcha input
        document.getElementById('captcha-input').value = '';
    }
});




    </script>
```

Cms.html(Contact management system)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact Management System</title>
    <style>
        /* Your existing CSS styles */
        body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

nav {
    background-color: #333;
    color: #fff;
    padding: 40px;
    font-size: 22px; /* Adjust the font size as needed */
}

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}

nav ul li {
    display: inline;
    margin-right: 10px;
}
```

```css
nav ul li a {
    color: #fff;
    text-decoration: none;
    padding: 10px 20px; /* Add padding to create button-like appearance */
    border-radius: 5px; /* Add rounded corners */
    background-color: #555; /* Add background color */
    transition: background-color 0.3s; /* Add transition effect for hover */
}

nav ul li a:hover {
    background-color: #777; /* Change background color on hover */
}


nav ul li a.active {
    font-weight: bold;
}


h1 {
    text-align: center;
    margin-top: 20px;
}

#contactForm {
    max-width: 600px;
    margin: 0 auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 10px;
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
    margin-top: 20px;
}
```

```
label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
}

input[type="text"],
input[type="email"],
input[type="tel"],
textarea {
    width: 100%;
    padding: 10px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    font-size: 16px;
    outline: none;
}

input[type="submit"],
button {
    width: 100%;
    padding: 10px;
    border: none;
    border-radius: 5px;
    background-color: #007bff;
    color: #fff;
    font-size: 18px;
    cursor: pointer;
    transition: background-color 0.3s;
}

input[type="submit"]:hover,
button:hover {
 background-color: #0056b3;
}

#searchContainer {
    text-align: center;
    margin-top: 20px;
}

#search {
    padding: 10px;
    width: 60%;
    border-radius: 5px;
    border: 1px solid #ccc;
    font-size: 16px;
    outline: none;
}

ul {
    list-style-type: none;
    padding: 0;
    margin: 20px auto;
    max-width: 800px;
}

li {
    background-color: #fff;
    padding: 20px;
    margin-bottom: 20px;
    border-radius: 10px;
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
    transition: transform 0.3s, box-shadow 0.3s;
}

li:hover {
 transform: translateY(-5px);
    box-shadow: 0 0 30px rgba(0, 0, 0, 0.2);
}

li button {
    padding: 8px 16px;
    background-color: #dc3545;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s;
}

li button.edit {
    background-color: #ffc107;
    margin-right: 10px;
}

li button.delete {
    background-color: #dc3545;
}

li button:hover {
    opacity: 0.8;
}
    </style>
    <script>
        function addContact(event) {
            event.preventDefault();

            var name = document.getElementById('name').value;
            var email = document.getElementById('email').value;
            var phoneNumber = document.getElementById('phoneNumber').value;
var address = document.getElementById('address').value;

            if (!email.endsWith('@gmail.com')) {
                alert("Email address must end with '@gmail.com'.");
                return;
            }

            var contact = {
                name: name,
                email: email,
                phoneNumber: phoneNumber,
                address: address
            };

            // Send POST request to server to save contact
            fetch('/add-contact', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                },
                body: JSON.stringify(contact),
            })
            .then(response => response.json())
            .then(data => {
                console.log('Success:', data);
                // Optionally, you can update the UI or perform any other actions upon successful saving
                // For example, you can reset the form and update the contact list
                document.getElementById('addContactForm').reset();
                updateContactList(); // Assuming you have a function to update the contact list
            })
            .catch((error) => {
                console.error('Error:', error);
                // Handle error, if any
```

```
 alert('Failed to save contact. Please try again.');
      });
    }
  </script>
</head>
<body>
  <nav>
    <ul>
      <li><a href="#" onclick="showAddContactForm()">Add Contact</a></li>
      <li><a href="#" onclick="displayAllContacts()">Display Contacts</a></li>
      <li><a href="#" onclick="showSearchBar()">Search Contact</a></li>
    </ul>
  </nav>

  <h1>Contact Management System</h1>

  <div id="contactForm" style="display: none;">
    <form id="addContactForm" onsubmit="addContact(event)">
      <label for="name">Name:</label><br>
      <input type="text" id="name" name="name" required><br>

      <label for="email">Email:</label><br>
      <input type="email" id="email" name="email" required><br>

      <label for="phoneNumber">Phone Number:</label><br>
      <input type="tel" id="phoneNumber" name="phoneNumber" required><br>

      <label for="address">Address:</label><br>
      <textarea id="address" name="address" required></textarea><br>

      <button type="submit" id="submitBtn">Add Contact</button>
            <button id="updateBtn" style="display: none;"
      onclick="updateContact()">Update Contact</button>
      <button id="cancelBtn" style="display: none;"
```

```
onclick="cancelEdit()">Cancel</button>
        </form>
    </div>

    <div id="searchContainer" style="display: none;">
        <label for="search">Search Contact:</label><br>
        <input type="text" id="search" oninput="searchContacts()" placeholder="Enter name to
search...">
    </div>

    <ul id="contactList"></ul>

    <script>
        var contacts = JSON.parse(localStorage.getItem('contacts')) || [];

        // Your existing JavaScript functions for displaying, editing, deleting contacts, etc.

        var contacts = JSON.parse(localStorage.getItem('contacts')) || [];

        function addContact(event) {
            event.preventDefault();

            var name = document.getElementById('name').value;
            var email = document.getElementById('email').value;
            var phoneNumber = document.getElementById('phoneNumber').value;
            var address = document.getElementById('address').value;

            if (!email.endsWith('@gmail.com')) {
                alert("Email address must end with '@gmail.com'.");
                return;
            }

            var contact = {
                name: name,
   email: email,
                   phoneNumber: phoneNumber,
                   address: address
               };

               contacts.push(contact);
               localStorage.setItem('contacts', JSON.stringify(contacts));
               document.getElementById('addContactForm').reset();

               // Check if "Display Contacts" is selected
               if (document.querySelector('nav ul li:nth-child(2) a').classList.contains('active')) {
                   updateContactList();
               }
           }

function deleteContact(index) {
    contacts.splice(index, 1);
    localStorage.setItem('contacts', JSON.stringify(contacts));
    updateContactList();
}

function editContact(index) {
    var contact = contacts[index];
    document.getElementById('name').value = contact.name;
    document.getElementById('email').value = contact.email;
    document.getElementById('phoneNumber').value = contact.phoneNumber;
    document.getElementById('address').value = contact.address;

    document.getElementById('submitBtn').style.display = 'none';
    document.getElementById('updateBtn').style.display = 'block';
    document.getElementById('cancelBtn').style.display = 'block';

    document.getElementById('updateBtn').setAttribute('data-index', index);
}
```

```
function updateContact() {
    var index = parseInt(document.getElementById('updateBtn').getAttribute('data-index'));
    var name = document.getElementById('name').value;
    var email = document.getElementById('email').value;
    var phoneNumber = document.getElementById('phoneNumber').value;
    var address = document.getElementById('address').value;

    if (!email.endsWith('@gmail.com')) {
        alert("Email address must end with '@gmail.com'.");
        return;
    }

    var contact = {
        name: name,
        email: email,
        phoneNumber: phoneNumber,
        address: address
    };

    contacts[index] = contact;
    localStorage.setItem('contacts', JSON.stringify(contacts));
    updateContactList();
    document.getElementById('addContactForm').reset();

    document.getElementById('submitBtn').style.display = 'block';
    document.getElementById('updateBtn').style.display = 'none';
    document.getElementById('cancelBtn').style.display = 'none';
}

function cancelEdit() {
    document.getElementById('addContactForm').reset();

    document.getElementById('submitBtn').style.display = 'block';
    document.getElementById('updateBtn').style.display = 'none';
document.getElementById('cancelBtn').style.display = 'none';
        }

        function searchContacts() {
            var searchText = document.getElementById('search').value.toLowerCase();
            var filteredContacts = contacts.filter(function(contact) {
                return contact.name.toLowerCase().includes(searchText);
            });

            updateContactList(filteredContacts);
        }

        function displayAllContacts() {
            updateContactList();
            document.getElementById('contactForm').style.display = 'none';
            document.getElementById('searchContainer').style.display = 'none';
            // Add active class to Display Contacts option
            document.querySelector('nav ul li:nth-child(2) a').classList.add('active');
        }

        function showAddContactForm() {
            document.getElementById('contactForm').style.display = 'block';
            document.getElementById('searchContainer').style.display = 'none';
            document.getElementById('contactList').innerHTML = '';
            // Remove active class from other options
            document.querySelector('nav ul li:nth-child(1) a').classList.remove('active');
            document.querySelector('nav ul li:nth-child(3) a').classList.remove('active');
        }

        function showSearchBar() {
            document.getElementById('searchContainer').style.display = 'block';
            document.getElementById('contactForm').style.display = 'none';
            document.getElementById('contactList').innerHTML = '';
            // Remove active class from other options
```

```
document.querySelector('nav ul li:nth-child(1) a').classList.remove('active');
            document.querySelector('nav ul li:nth-child(2) a').classList.remove('active');
        }

        function updateContactList(filteredContacts) {
            var contactList = document.getElementById('contactList');
            contactList.innerHTML = '';

            var contactsToDisplay = filteredContacts || contacts;

            contactsToDisplay.forEach(function(contact, index) {
                var contactItem = document.createElement('li');
                contactItem.innerHTML = `
                    <strong>Name:</strong> ${contact.name}<br>
                    <strong>Email:</strong> ${contact.email}<br>
                    <strong>Phone Number:</strong> ${contact.phoneNumber}<br>
                    <strong>Address:</strong> ${contact.address}<br>
                    <button class="edit" onclick="editContact(${index})">Edit</button>
                    <button class="delete" onclick="deleteContact(${index})">Delete</button><br><br>
                `;
                contactList.appendChild(contactItem);
            });
        }

        function editContact(index) {
            var contact = contacts[index];
            var newName = prompt("Enter new name", contact.name);
            var newEmail = prompt("Enter new email", contact.email);
            var newPhoneNumber = prompt("Enter new phone number", contact.phoneNumber);
            var newAddress = prompt("Enter new address", contact.address);

            if (newName !== null && newEmail !== null && newPhoneNumber !== null &&
newAddress !== null) {
                contacts[index] = {
```

```
name: newName,
                    email: newEmail,
                    phoneNumber: newPhoneNumber,
                    address: newAddress
                };
                localStorage.setItem('contacts', JSON.stringify(contacts)); // Update localStorage
                displayAllContacts(); // Update the displayed contacts
            }
        }

    </script>
</body>
</html>
```

Back - end Code

App.js(Connecting with mongo db)

```
const express = require('express');

const bodyParser = require('body-parser');

const mongoose = require('mongoose');

const path = require('path'); // Import path module



const app = express();

const PORT = 3000;



// Connect to MongoDB

mongoose.connect('mongodb://localhost:27017/signup-login-db', { useNewUrlParser: true,
useUnifiedTopology: true });

const db = mongoose.connection;



// Define schema for contact

const contactSchema = new mongoose.Schema({

    name: { type: String, required: true },

    email: { type: String, required: true },

    phoneNumber: String,

    address: String

});
```

```
// Create model for contact

const Contact = mongoose.model('Contact', contactSchema);


// Define schema for user

const userSchema = new mongoose.Schema({

    name: String,

    mobile: String,

    email: String,

    password: String

});


// Create model for user

const User = mongoose.model('User', userSchema);


// Middleware

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

app.use(express.json());


// Serve static files from the 'public' directory

app.use(express.static(path.join(__dirname, 'public'))); // Serve files from 'public' directory


// Routes
```

```
app.get('/', (req, res) => {
    res.sendFile(path.join(__dirname, 'index.html')); // Serve index.html from the root directory
});

// Signup route
app.post('/signup', (req, res) => {
    const { name, mobile, email, password } = req.body;

    // Check if email already exists
    User.findOne({ email: email }).exec()
    .then(user => {
        if (user) {
            res.status(400).send('Email already exists');
        } else {
            // Create new user
            const newUser = new User({
                name: name,
                mobile: mobile,
                email: email,
                password: password
            });
            newUser.save()
```

```
 .then(() => {
                         // Redirect to login page after successful signup
                    res.redirect('/login');
                })
                .catch(err => {
                    console.error(err);
                    res.status(500).send('Error occurred while signing up');
                });
        }
    })
    .catch(err => {
        console.error(err);
        res.status(500).send('Error occurred while checking email');
    });
});


// Login route
app.post('/login', (req, res) => {
    const { email, password } = req.body;:


    // Check if a user with the provided email and password exists in the database
    User.findOne({ email: email, password: password }, (err, user) => {
```

```
if (err) {
            // Handle any errors
            console.error(err);
            res.status(500).send('Internal Server Error');
        } else if (user) {
            // If user is found, login is successful, redirect to cms.html
            res.send('<script>window.open("/cms.html", "_blank");</script>'); // Open cms.html in a new tab
        } else {
            // If user is not found or password is incorrect, redirect back to login page
            res.redirect('/login');
        }
    });
});


// Add contact route
app.post('/add-contact', (req, res) => {
    const { name, email, phoneNumber, address } = req.body;

    // Create a new contact instance
    const newContact = new Contact({
        name,
 email,
        phoneNumber,
        address
    });

    // Save the contact to MongoDB
    newContact.save((err, savedContact) => {
        if (err) {
            console.error('Error saving contact:', err);
            return res.status(500).json({ error: "Failed to save contact", details: err });
        }
        console.log('Contact saved successfully:', savedContact);
        res.status(200).json({ message: "Contact saved successfully", contact: savedContact });
    });
});

// Serve login page
app.get('/login', (req, res) => {
    res.sendFile(path.join(__dirname, 'login.html')); // Serve login.html from the root directory
});

// Serve cms.html
```

```
app.get('/cms.html', (req, res) => {
    res.sendFile(path.join(__dirname, 'public', 'cms.html')); // Serve cms.html from the 'public' directory
});

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