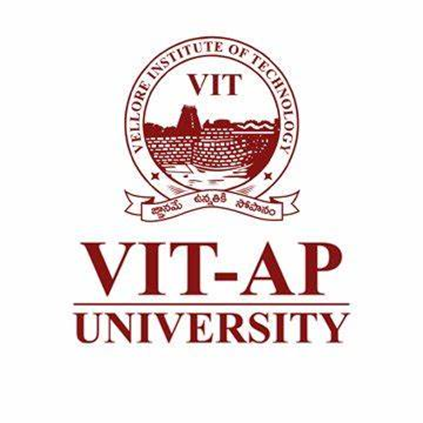
**FINAL REPORT**

Course Code: CSE1005

Course Name: SOFTWARE ENGINEERING

Semester: FALL SEM (2024-2025)

GROUP-3



Submitted By

JAITHRI.T-22BCE20513

B. JOHAN WESLEY - 22BCE20492

Sreeganesh M R - 22BCE7197

Kabir Sha - 22BCE7108

M SATYENDRA KUMAR - 22BCE7063

Submitted To

Prof. Shaik Kareemulla

**INDEX**

1.ABSTRACT

2.PROBLEM STATEMENT

3.SOFTWARE REQUIREMENT SPECIFICATION(SRS)

4.UML DIAGRAMS

5.WEBSITE CODES AND RESULTS

6.TEST CASES

7.CONCLUSION

8.RESULT

**ABSTRACT**

The Smart Metro Card Recharge System is a web-based application that significantly enhances the metro commuting experience by allowing users to recharge their smart cards online. This innovation eliminates the need for long waits at customer care centers. With a user-friendly interface, the app enables easy management of card balances and supports various payment methods, including net banking, credit/debit cards, and Paytm. Additional features like fare calculation, station navigation, and personalized settings further improve user convenience.

Designed to integrate seamlessly with the existing infrastructure of Hyderabad Metro Rail Limited (HMRL), the system employs secure RFID technology for quick balance updates at metro gates. By facilitating remote top-ups, it aims to reduce wait times and streamline the overall travel experience for passengers. The solution is also scalable, capable of handling high user traffic while maintaining smooth operation during peak hours, thereby contributing to a more reliable and efficient public transportation system.

**Problem Statement**

HMRL has this awesome system with Smart Cards for multiple journeys and Smart tokens for single trips. The Smart card can be topped up at any metro station's customer care centers and stays valid for a year from the last recharge. Despite the metro being a lifesaver for daily commuters, the long queues for card top-ups have been a hassle.

To tackle this issue, an online portal for smart card users has been launched. Now, commuters can recharge their metro cards online from home, skipping the long queues. This web app offers a user-friendly platform for daily metro riders to recharge their smart cards. Users need to register with HMRL and the app, and they can recharge using online net banking, credit/debit cards, or Paytm.

Additionally, the app provides services like maps, balance checks, and finding the nearest metro station. It's a game-changer for hassle-free metro travels. The convenience of online recharges coupled with special offers makes commuting smoother for everyone. This initiative aims to enhance the metro experience and make daily journeys more efficient and enjoyable for all passengers.

**Software Requirement Specification**

**Overall Description**

**Product Functions**

The application is designed to provide users with the convenience of recharging their metro cards from anywhere using any web-enabled device. Apart from the recharge feature, the platform offers users the ability to create a personalized login, granting access to various other functionalities such as navigation, fare calculation, and settings customization.

Through the navigation feature, users can input their location or utilize GPS to identify the nearest metro station. The fare calculator option facilitates users in determining distances, travel times, routes, and fares between any two metro stations. Moreover, the settings option allows users to configure their devices to receive updates and notifications from the application. Additionally, users can benefit from exclusive offers during online transactions, enhancing the overall user experience.

**User Characteristics**

The application is user-friendly, designed to assist individuals with varying levels of technological expertise in easily recharging their metro cards through simple interfaces and intuitive buttons. Users do not require any specialized knowledge or training to navigate and utilize the application efficiently, ensuring a seamless experience for all.

**General Constraints**

The application is typically restricted to Smart card users for accessing additional features like navigation and fare calculation. Individuals without a valid metro card linked to their name are unable to benefit from these supplementary services.

**Assumptions and Dependencies**

We assume that users signing in with a specific metro card number and phone number are already registered in the User info database by the metro card issuer using the admin login. Additionally, each metro station is equipped with an RFID punch facility at the four gates, enabling users to easily recharge their cards. This hardware dependency ensures a seamless process where users can punch their cards at the gates to transfer the balance conveniently.

The registration process involves the metro card issuer inputting user information into the database through the admin login, linking each metro card number with the corresponding phone number. This setup streamlines the recharge process for users, as they can simply punch their cards at the RFID-enabled gates at each station to add credit.

**External Interface Requirements**

**User Interfaces**

Login and Signup Interfaces

Online Payment and Enquiry and Success Interfaces

**Hardware Interfaces**

We need a basic display without high definition capabilities, and an internet connection is essential to facilitate online payments and data management in the HMRL database. Additionally, a secure RFID punch facility is necessary for users to conveniently recharge their cards by simply pressing the card against it.

**Software Interfaces**

We need a dedicated database named "USER INFO" accessible to both the application and the HMRL admin for metro card users to register on the app and access services. Additionally, a compact file called "Metro details" is required, containing information about metro stations, their respective line codes, and the fare table**.**

**Functional Requirements**

**FR 1:** Signup involves acquiring inputs such as Name, DOB, mobile number, email Id, and Metro Card Number to create an account. The validate() module validates the data by checking if the entered metro card has been sold and then feeds the validated user data into the "USER INFO" database for registration with the application.

**FR 2:** Login requires inputting the Metro Card Number and Mobile Phone Number. The function acquires this data and uses the validate() module to authenticate the user by matching the details in the database, ensuring the user's authorization to access the services.

**FR 3:** Online payment process involves acquiring valid data, Metro Card Number, Payment Option, Amount, and Rebate option for making payments to HMRL. The connect() module securely connects to the respective banking system for online money transfer. Once the bank processes the payment, the process() module updates the user's balance by adding the transferred amount and signals the RFID punch machine for the new payment.

**FR 4:** Enquiry module allows users to input options like navigate, fare, setting, or logout, and displays relevant data based on the selected option. This module contains sub-modules tailored to process different types of user requests accordingly.

**FR 5:** The (Enquiry) Fare Calculator function takes Metro station names as input and calculates details such as station distance, route, fare, and time. It utilizes the input() module to gather data, processes it using the process() function, and then displays the output through the display() module.

**FR 6:** The (Enquiry) Navigate feature accepts a Place Name as input and provides the corresponding Metro station as output. It employs the input() module to collect the data, processes it with the process() function, and displays the result using the display() module.

**FR 7:** The (Enquiry) Setting function allows users to input various options and confirms the selection. It uses the input() module to obtain the user's choices, processes them with the process() function, and sends a confirmation to the display() module for presentation.

**Performance Requirement**

Static:

The application accommodates an unlimited number of concurrent users for logging in and making online payments. It relies on two critical files: the User Information database, which stores data on issued metro cards, and the Metro details file, containing information on each metro station and fare details.

Dynamic:

Typically, the application can provide a response within 5 seconds post-processing and bank connection. For inquiries, the response time is as quick as 5 milliseconds. The system's throughput, around 3/ms, is influenced by the terminal's performance. This information indicates that the system operates efficiently with a standard throughput rate.

**Design Constraints**

Standard Compliance:

The system doesn't need to adhere to any standards since the data is entered online directly by the application.

Hardware Limitations:

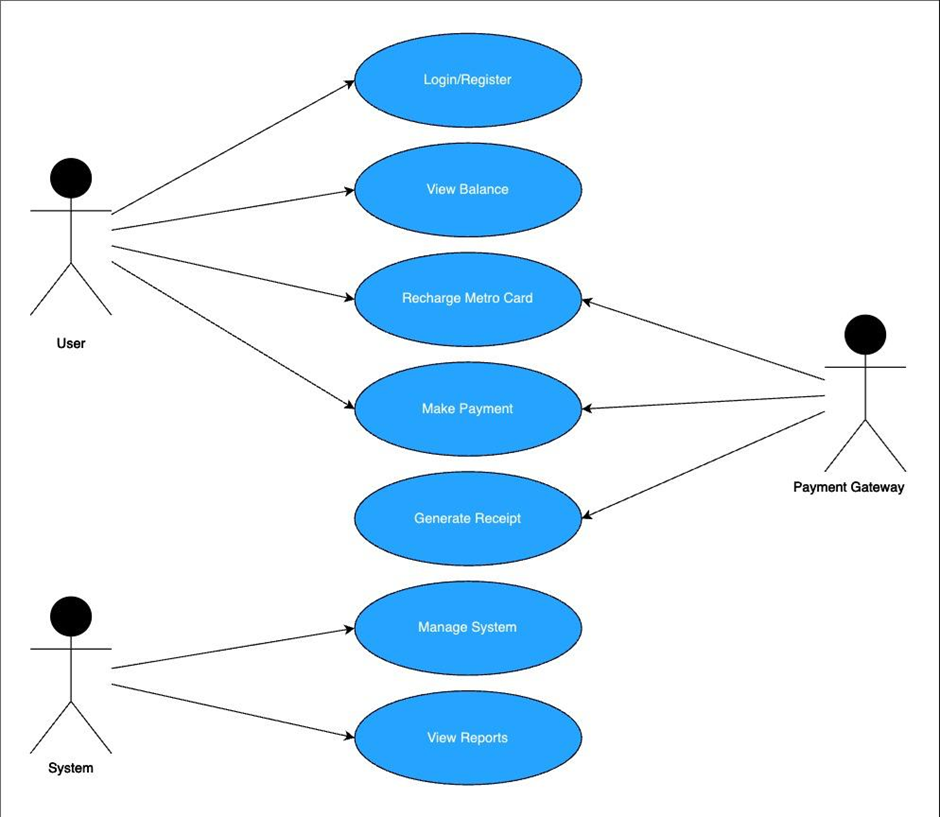
The software's functionality relies on the user tapping the card on the hardware at the metro station gate for efficiency.

Reliability and Fault Tolerance:

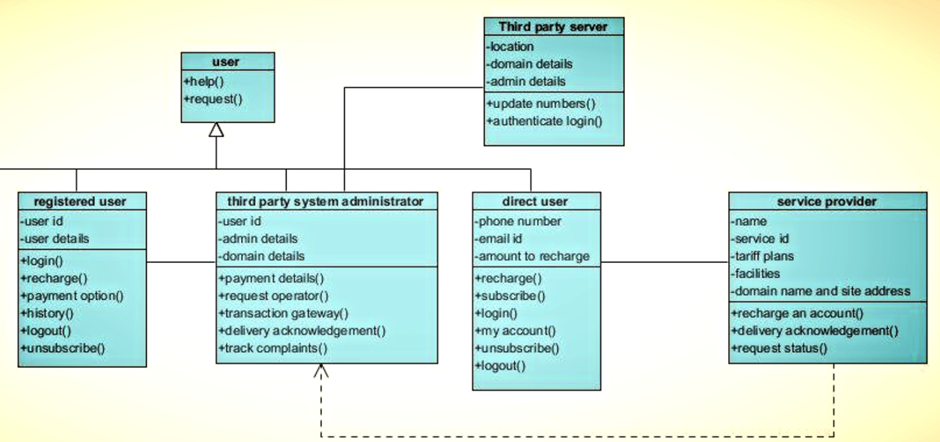
This software is reliable and designed to handle critical scenarios, ensuring 50% fault tolerance to maintain operational stability.

**UML DIAGRAMS**

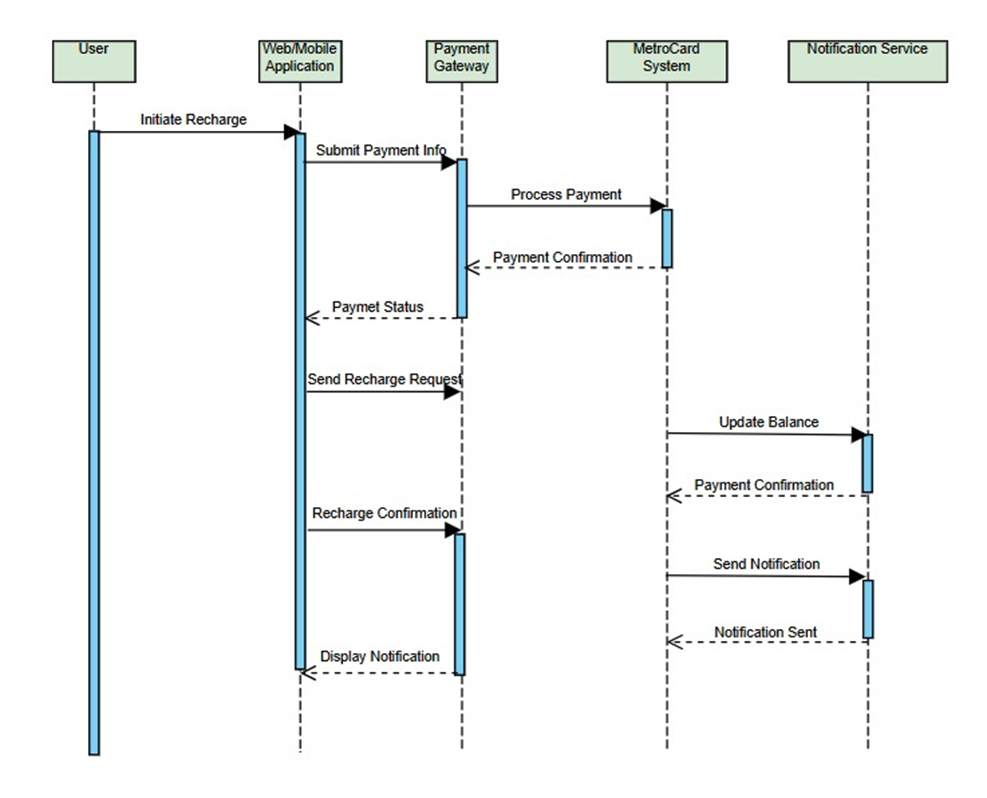
**Use Case Diagram**

****

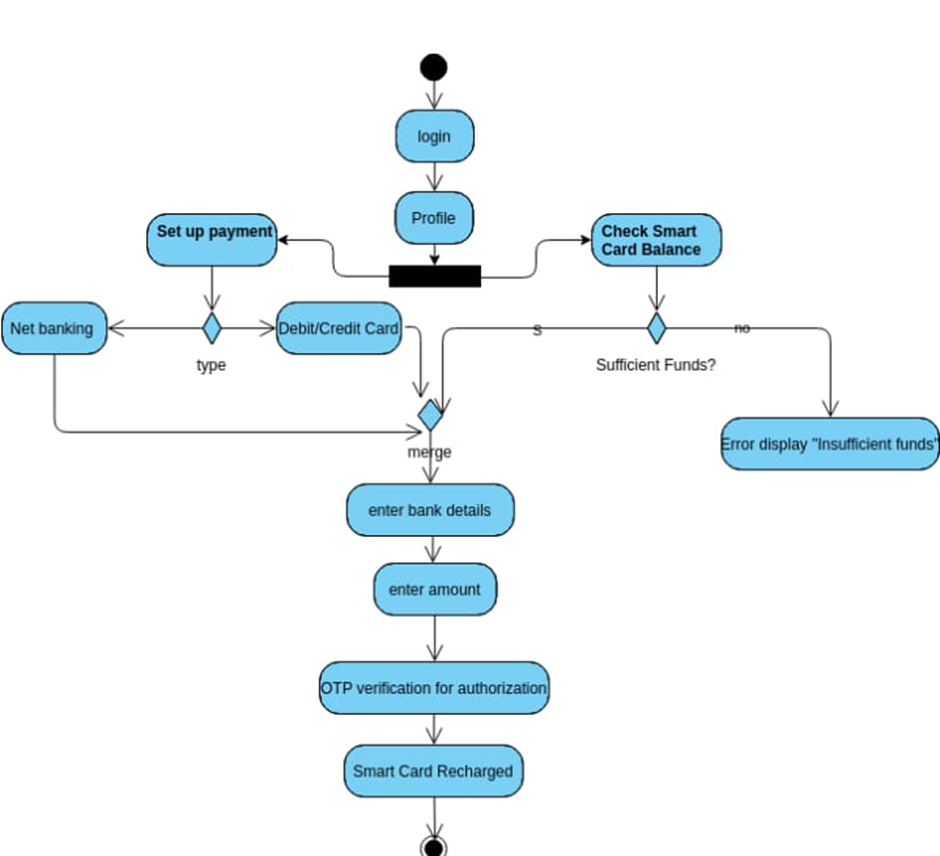
**Class Diagram**

****

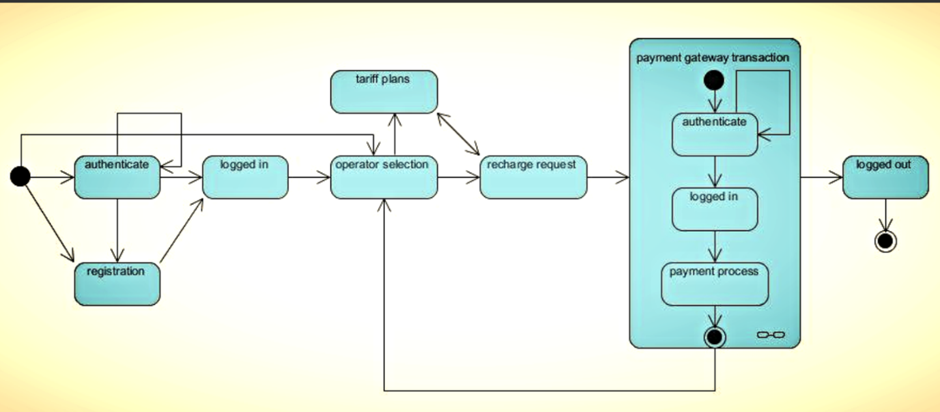
**Sequence Diagram**

****

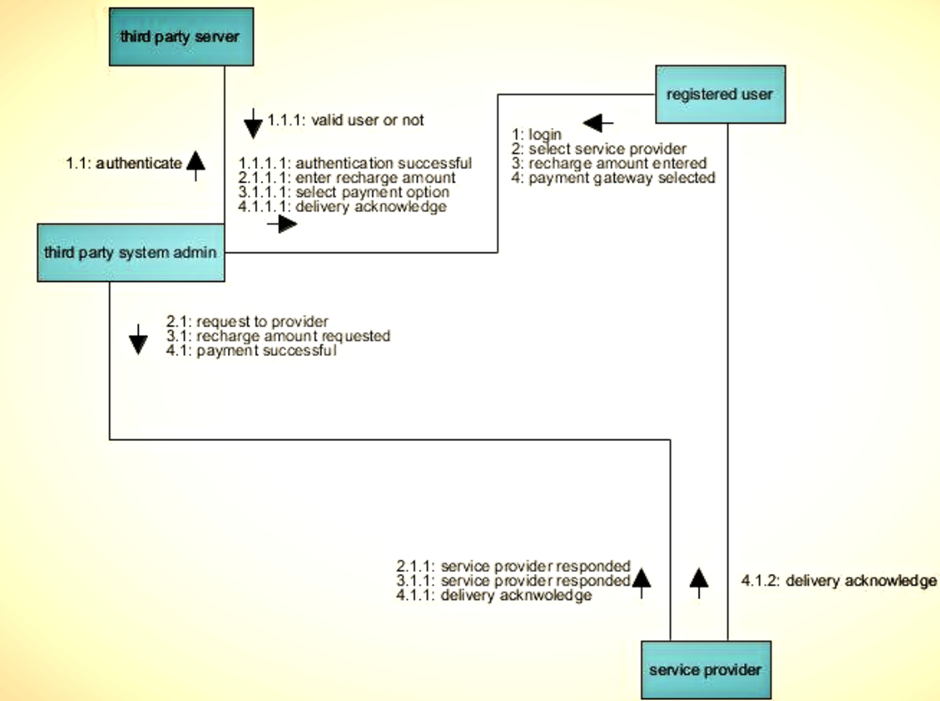
**Activity Diagram**

****

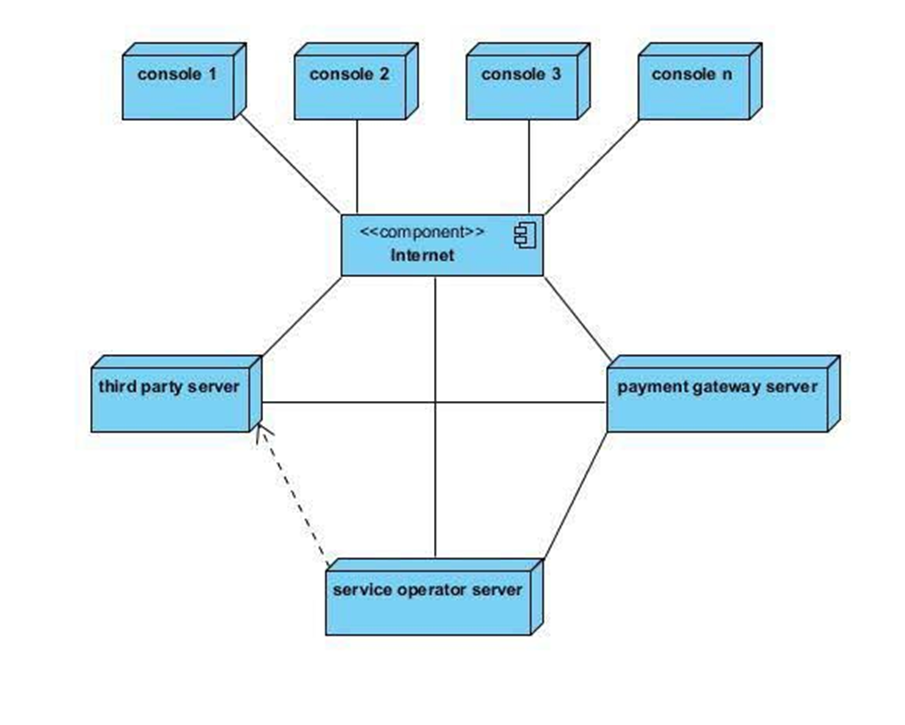
**State Chart Diagram**

****

**Collaboration Diagram**

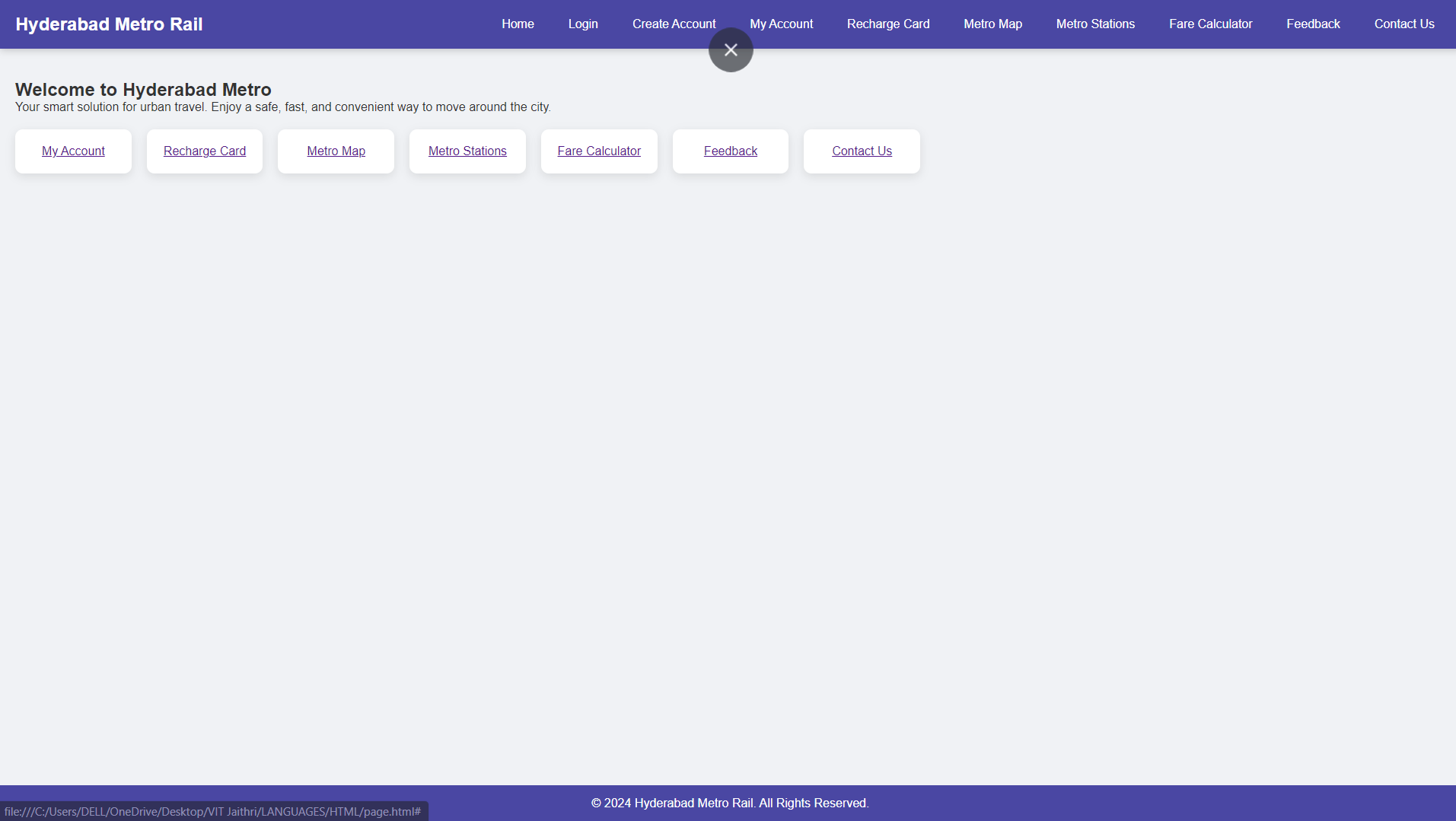
****

**Component and Deployment Diagram**

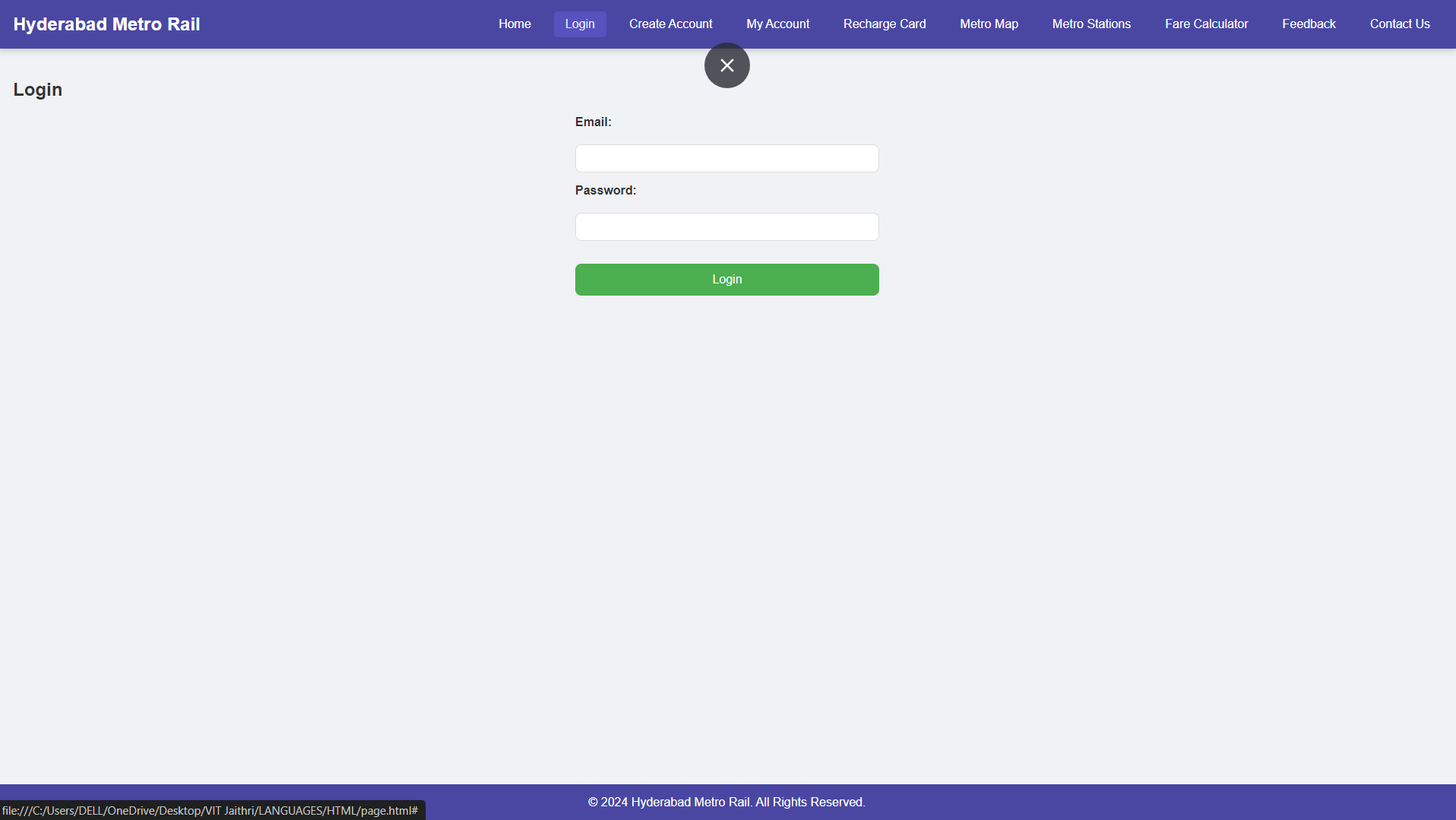
****

**WEBSITE CODES AND RESULTS**

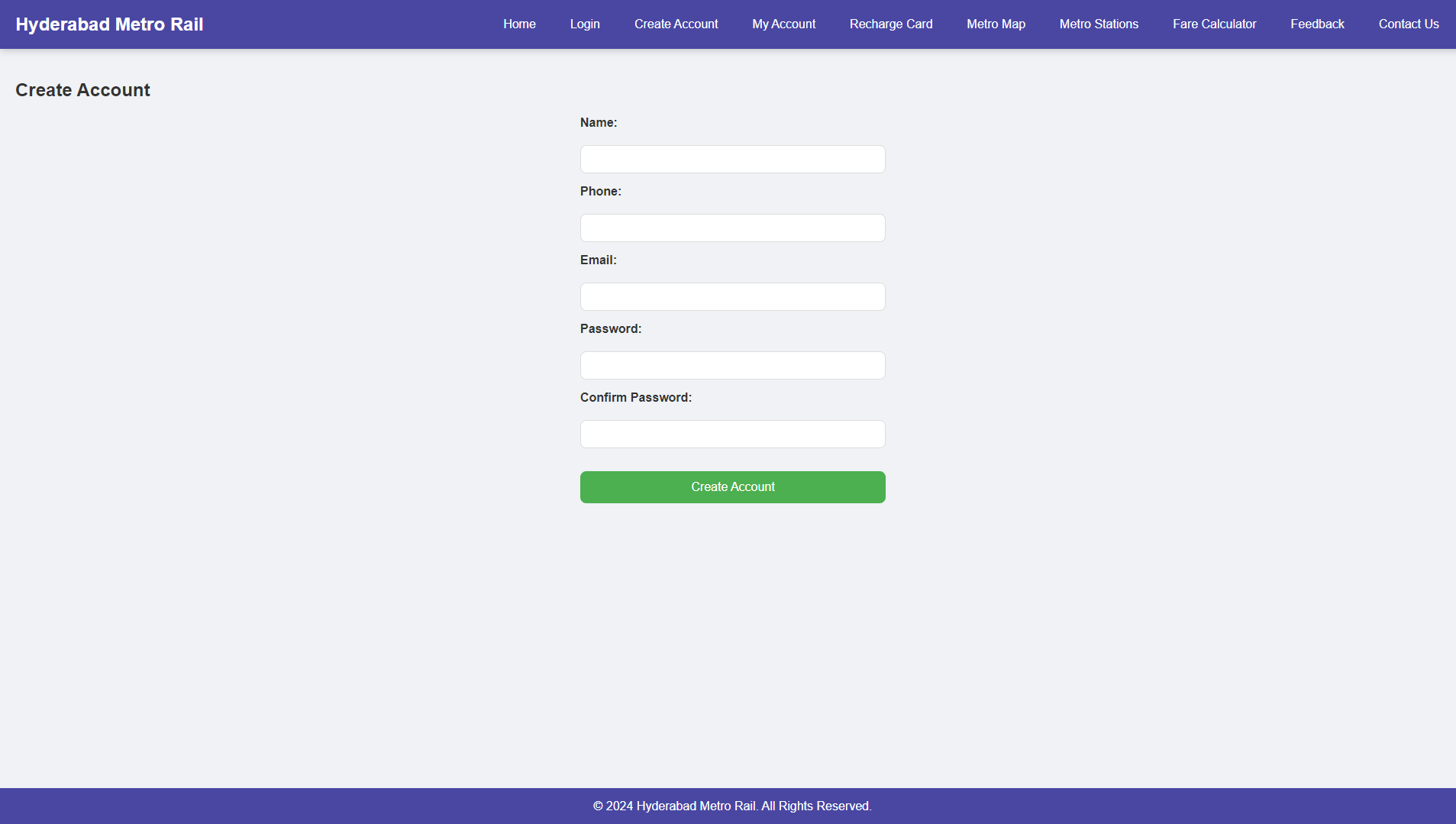
Home



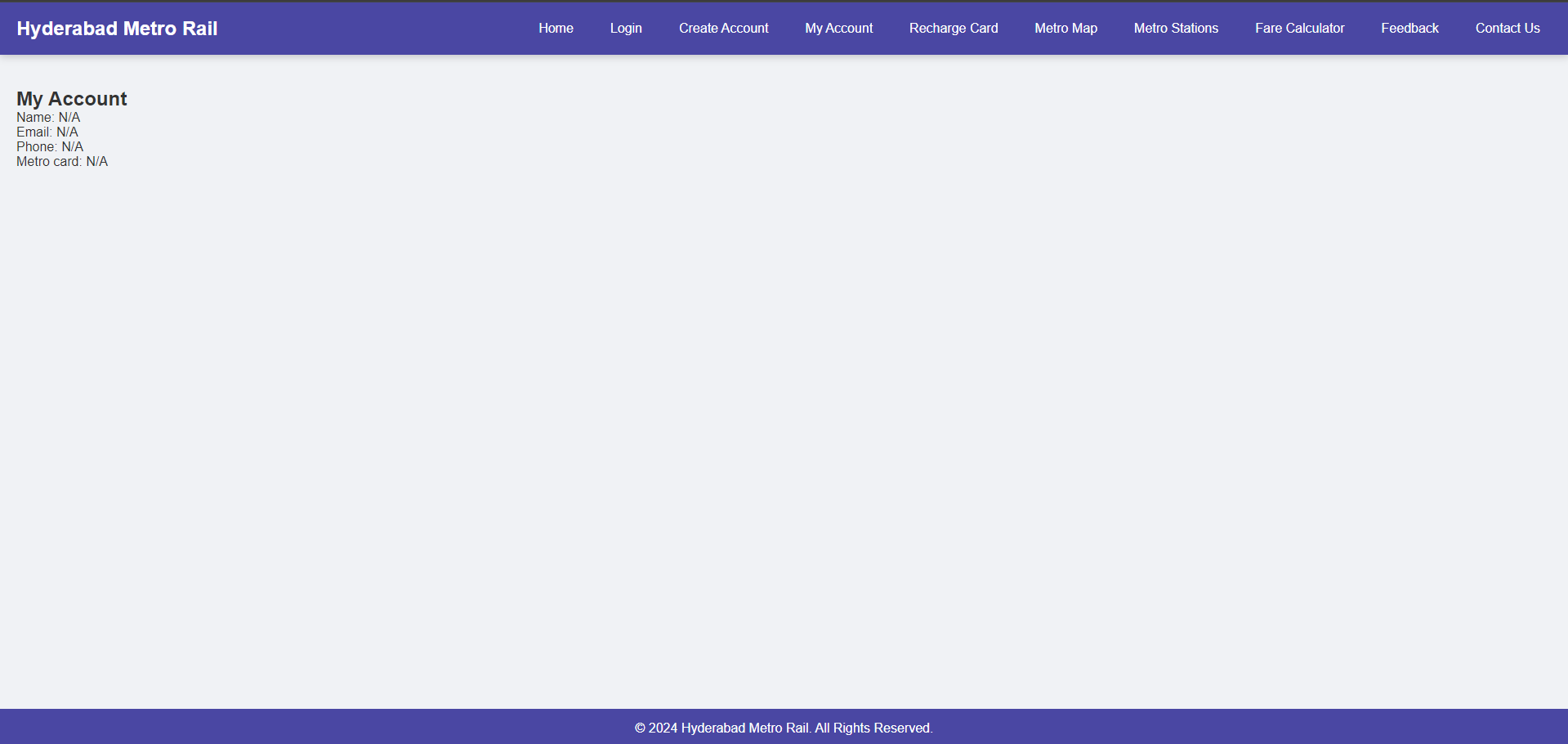
Login



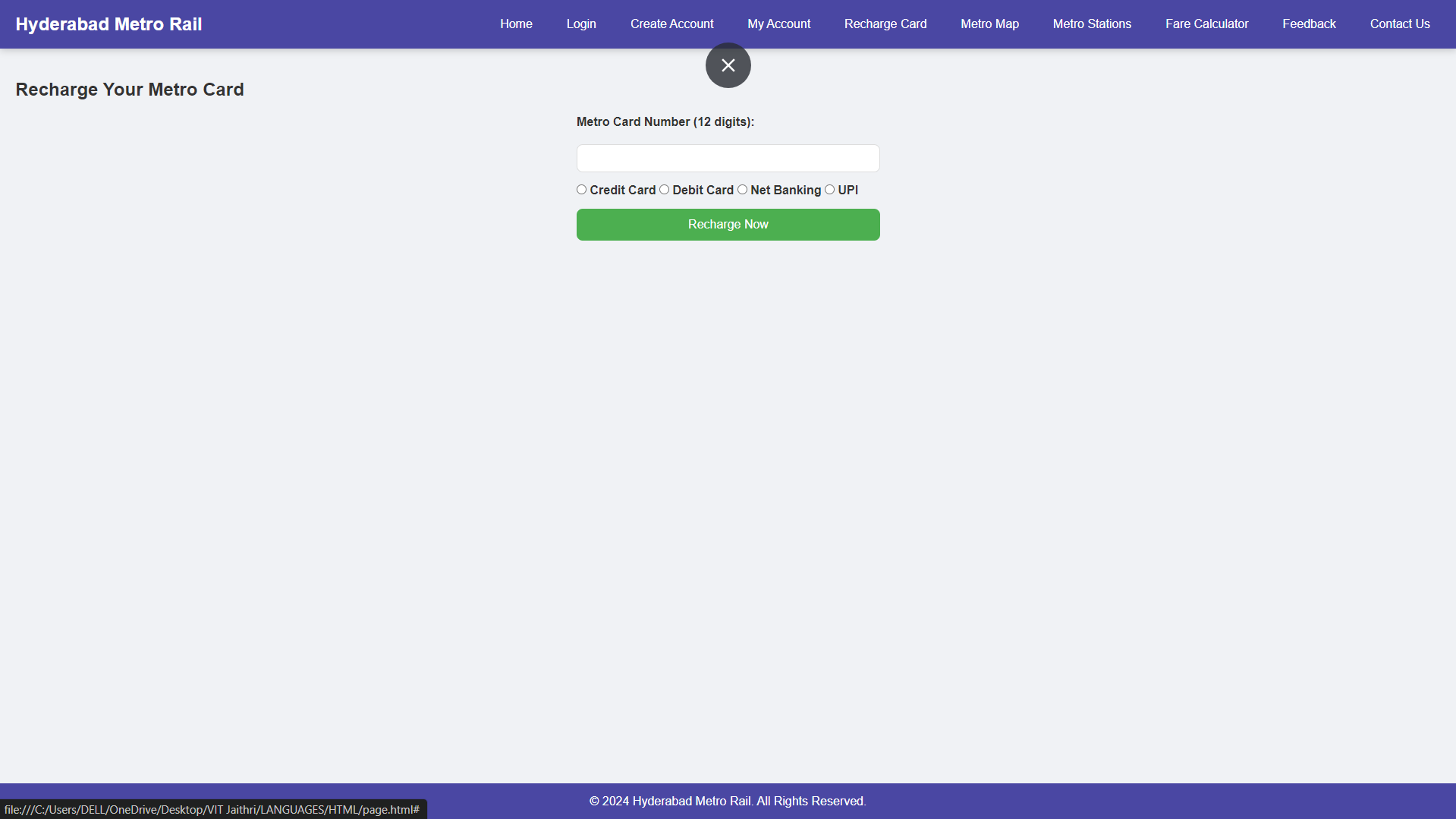
Create Account



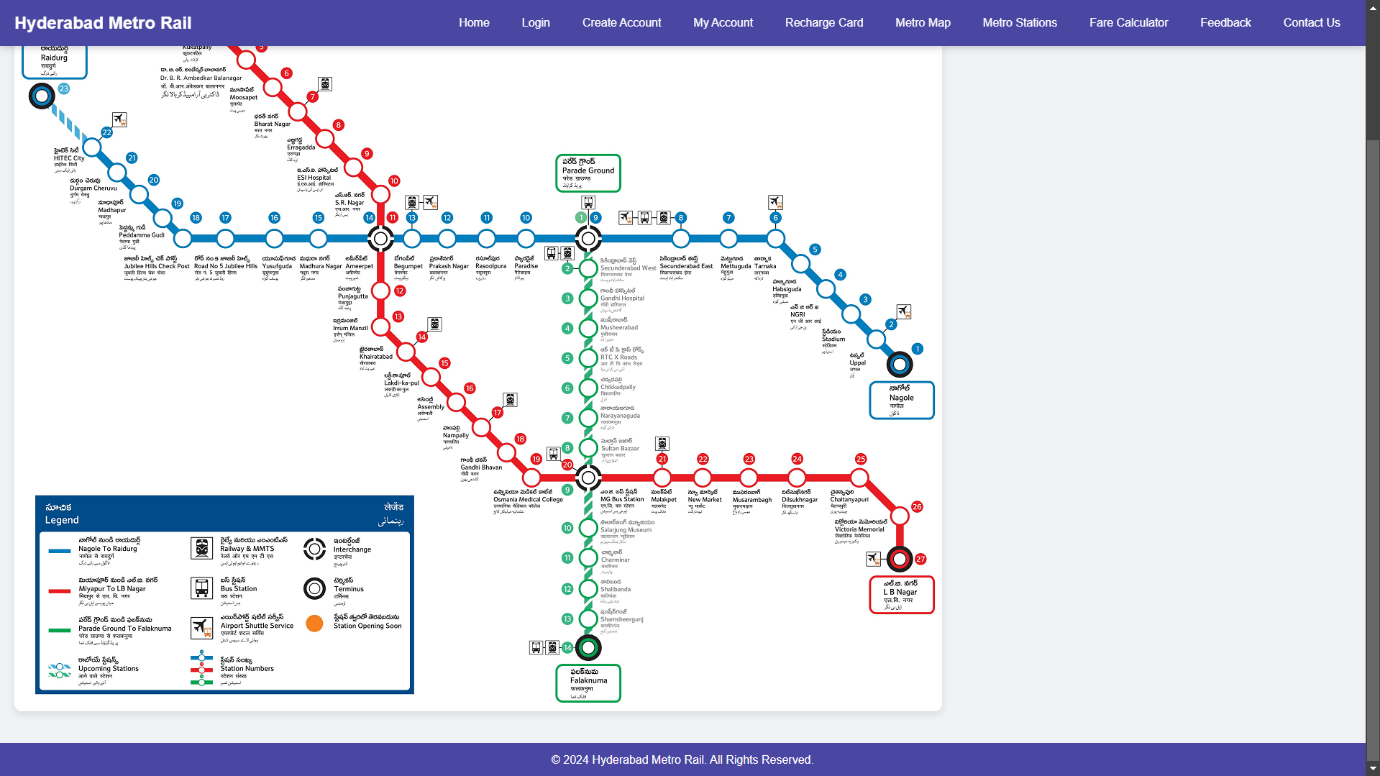
My Account



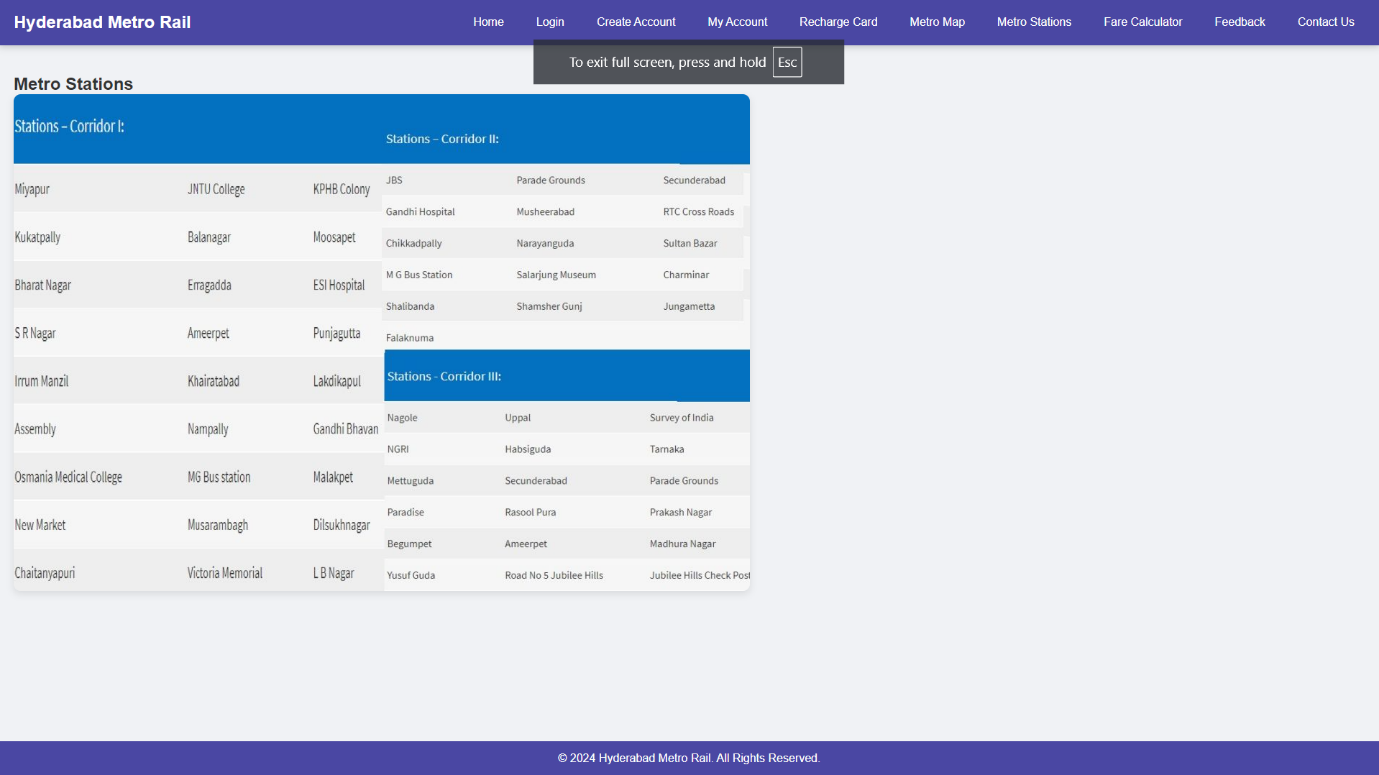
Recharge Card



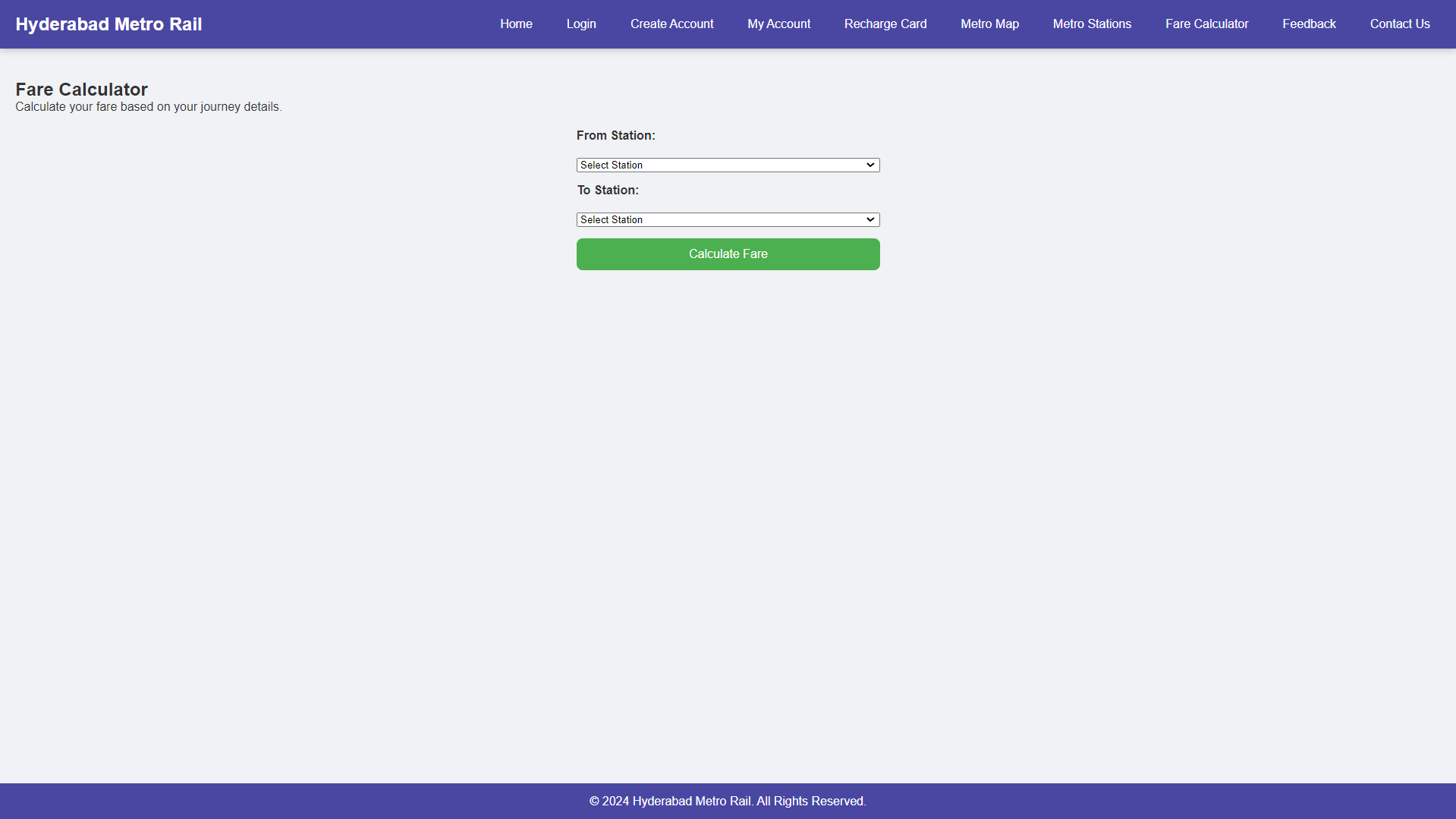
Metro Map



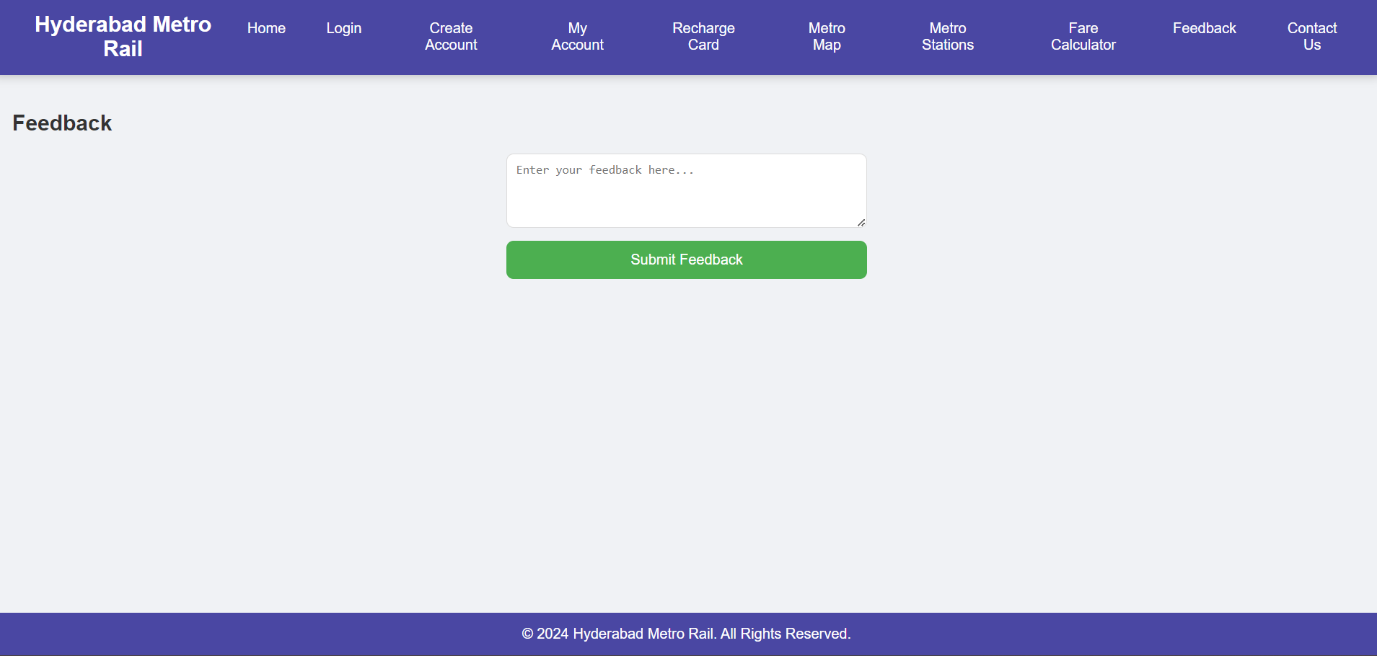
Metro Stations



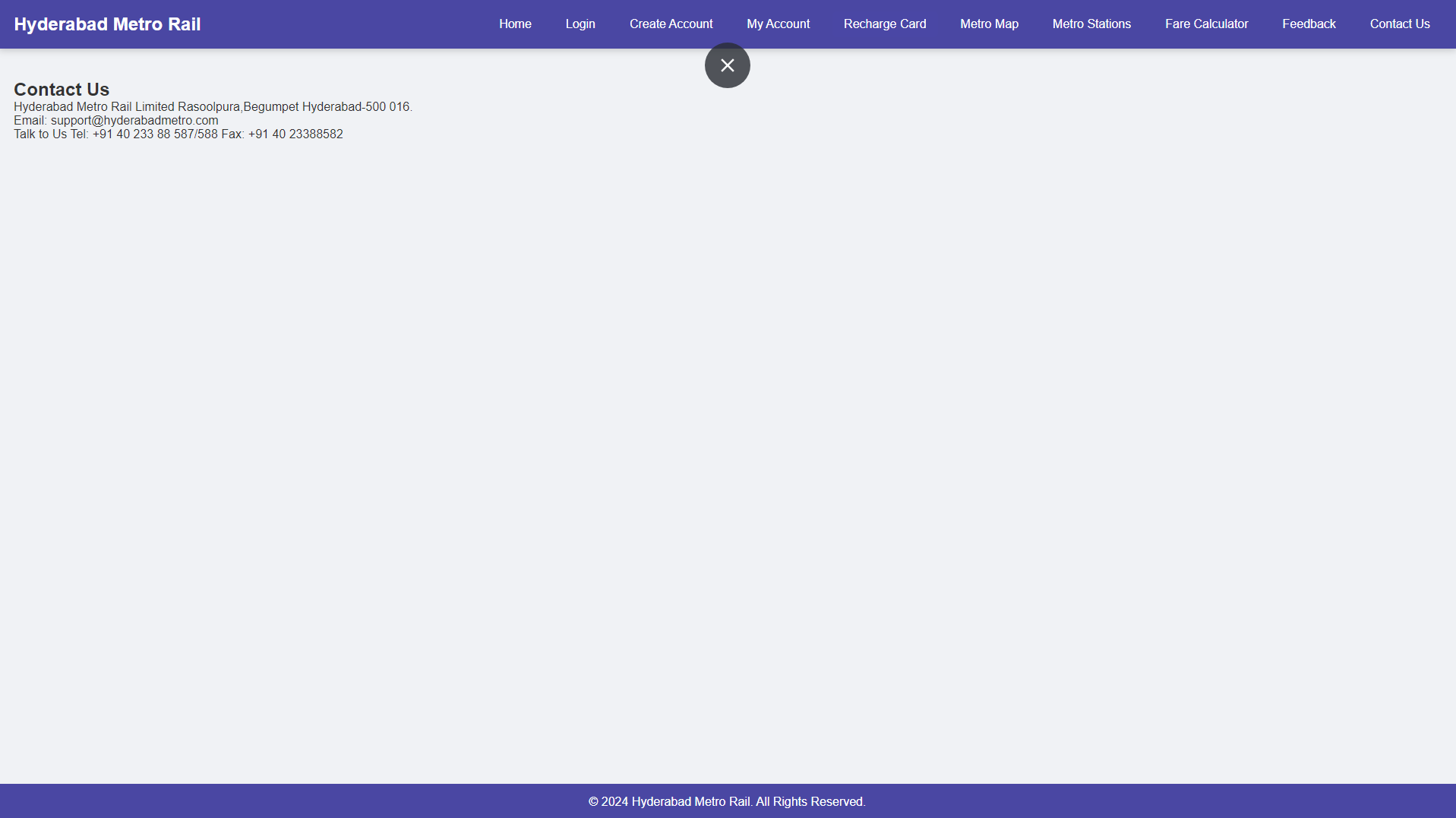
Journey Fare Calculator



Feedback



Contact Us



**CODE:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Metro Booking System</title>

    <style>

        \* {

            margin: 0;

            padding: 0;

            box-sizing: border-box;

        }

        body {

            font-family: Arial, sans-serif;

            background-color: #f0f2f5;

            color: #333;

            text-align: center;

            display: flex;

            flex-direction: column;

            min-height: 100vh;

        }

        /\* Header Styles \*/

        header {

            background-color: #4a47a3;

            color: #fff;

            padding: 15px 20px;

            display: flex;

            align-items: center;

            justify-content: space-between;

            position: sticky;

            top: 0;

            width: 100%;

            z-index: 1000;

            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.2);

        }

        header h1 {

            font-size: 24px;

            margin: 0;

        }

        nav {

            display: flex;

            gap: 15px;

        }

        nav a {

            color: #fff;

            text-decoration: none;

            font-size: 16px;

            padding: 8px 15px;

            border-radius: 5px;

            transition: background 0.3s, color 0.3s;

        }

        nav a:hover {

            background-color: #5a55c6;

            color: #e2e2ff;

        }

        /\* Section Styles \*/

        section {

            padding: 40px 20px;

            display: none;

            text-align: left;

            animation: fadeIn 0.5s ease;

        }

        .section.active {

            display: block;

        }

        /\* Form Styles \*/

        form {

            max-width: 400px;

            margin: 20px auto;

            display: flex;

            flex-direction: column;

            gap: 15px;

        }

        label {

            font-weight: bold;

            margin-bottom: 5px;

        }

        input[type="text"], input[type="email"], input[type="password"], textarea {

            padding: 10px;

            border: 1px solid #ddd;

            border-radius: 8px;

        }

        button {

            padding: 12px;

            background-color: #4caf50;

            color: white;

            border: none;

            border-radius: 8px;

            cursor: pointer;

            font-size: 16px;

            transition: background-color 0.3s;

        }

        button:hover {

            background-color: #3e8e41;

        }

        /\* Options Grid \*/

        .options {

            display: grid;

            grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));

            gap: 20px;

            margin-top: 20px;

        }

        .option {

            background-color: #fff;

            border-radius: 10px;

            padding: 20px;

            text-align: center;

            box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);

            transition: transform 0.3s, box-shadow 0.3s;

        }

        .option:hover {

            transform: scale(1.05);

            box-shadow: 0 6px 16px rgba(0, 0, 0, 0.2);

        }

        /\* Footer Styles \*/

        footer {

            background-color: #4a47a3;

            color: #fff;

            padding: 15px;

            text-align: center;

            margin-top: auto;

        }

        /\* Responsive Styles \*/

        @media (max-width: 768px) {

            header {

                flex-direction: column;

                align-items: center;

            }

            nav {

                flex-wrap: wrap;

                justify-content: center;

                gap: 10px;

            }

        }

        /\* Keyframes for Animations \*/

        @keyframes fadeIn {

            from { opacity: 0; }

            to { opacity: 1; }

        }

    </style>

    <script>

        let loggedInEmail = ''; // Variable to store the logged-in user's email

        function showSection(sectionId) {

            const sections = document.querySelectorAll('.section');

            sections.forEach(section => section.classList.remove('active'));

            document.getElementById(sectionId).classList.add('active');

        }

        // Test case functions for form submissions

        function testLoginForm(event) {

            event.preventDefault();

            const email = document.getElementById('login-email').value;

            const password = document.getElementById('login-password').value;

            console.log("Login Submitted with email:", email, "and password:", password);

            loggedInEmail = email; // Store the email of the logged-in user

            alert("Login successful!");

            showSection('account');

        }

        function testCreateAccount(event) {

            event.preventDefault();

            const name = document.getElementById('signup-name').value;

            const phone = document.getElementById('signup-phone').value;

            const email = document.getElementById('signup-email').value;

            const password = document.getElementById('signup-password').value;

            console.log("Account Created with Name:", name, "Phone:", phone, "Email:", email);

            alert("Account created successfully!");

        }

        function testRecharge(event) {

            event.preventDefault();

            const cardNumber = document.getElementById('card-number').value;

            const paymentMethod = document.querySelector('input[name="payment"]:checked').value;

            if (cardNumber.length !== 12) {

                alert("Metro Card Number must be 12 digits.");

                return;

            }

            console.log("Recharge with card number:", cardNumber, "using payment method:", paymentMethod);

            alert("Recharge successful!");

        }

        // Feedback Form Test Case

        function testFeedbackForm(event) {

            event.preventDefault();

            const feedbackMessage = document.getElementById('feedback-message').value;

            console.log("Feedback Submitted:", feedbackMessage);

            alert("Feedback submitted successfully!");

        }

        // Contact Us Form Test Case

        function testContactForm(event) {

            event.preventDefault();

            const name = document.getElementById('contact-name').value;

            const email = document.getElementById('contact-email').value;

            const message = document.getElementById('contact-message').value;

            console.log("Contact Form Submitted with Name:", name, "Email:", email, "Message:", message);

            alert("Message sent successfully!");

        }

        // Function to update My Account page with the logged-in email

        function updateAccountPage() {

            const accountEmailElement = document.getElementById('account-email');

            if (loggedInEmail) {

                accountEmailElement.textContent = `Email: ${loggedInEmail}`;

            }

        }

    </script>

</head>

<body>

    <header>

        <h1>Hyderabad Metro Rail</h1>

        <nav>

            <a href="#" onclick="showSection('home')">Home</a>

            <a href="#" onclick="showSection('login')">Login</a>

            <a href="#" onclick="showSection('create-account')">Create Account</a>

            <a href="#" onclick="showSection('account'); updateAccountPage();">My Account</a>

            <a href="#" onclick="showSection('recharge')">Recharge Card</a>

            <a href="#" onclick="showSection('metro-map')">Metro Map</a>

            <a href="#" onclick="showSection('metro-stations')">Metro Stations</a>

            <a href="#" onclick="showSection('journey')">Fare Calculator</a>

            <a href="#" onclick="showSection('feedback')">Feedback</a>

            <a href="#" onclick="showSection('contact')">Contact Us</a>

        </nav>

    </header>

    <!-- Home Section -->

    <section id="home" class="section active">

        <h2>Welcome to Hyderabad Metro</h2>

        <p>Your smart solution for urban travel. Enjoy a safe, fast, and convenient way to move around the city.</p>

        <div class="options">

            <div class="option"><a href="#" onclick="showSection('account')">My Account</a></div>

            <div class="option"><a href="#" onclick="showSection('recharge')">Recharge Card</a></div>

            <div class="option"><a href="#" onclick="showSection('metro-map')">Metro Map</a></div>

            <div class="option"><a href="#" onclick="showSection('metro-stations')">Metro Stations</a></div>

            <div class="option"><a href="#" onclick="showSection('journey')">Fare Calculator</a></div>

            <div class="option"><a href="#" onclick="showSection('feedback')">Feedback</a></div>

            <div class="option"><a href="#" onclick="showSection('contact')">Contact Us</a></div>

        </div>

    </section>

    <!-- Login Section -->

    <section id="login" class="section">

        <h2>Login</h2>

        <form id="login-form">

            <label for="login-email">Email:</label>

            <input type="email" id="login-email" name="email" required>

            <label for="login-password">Password:</label>

            <input type="password" id="login-password" name="password" required>

            <!-- Error Message Container -->

            <div id="login-password-error"></div>

            <button type="submit">Login</button>

        </form>

    </section>

    <!-- Create Account Section -->

    <section id="create-account" class="section">

        <h2>Create Account</h2>

        <form id="signup-form">

            <label for="signup-name">Name:</label>

            <input type="text" id="signup-name" name="name" required>

            <label for="signup-phone">Phone:</label>

            <input type="text" id="signup-phone" name="phone" required>

            <label for="signup-email">Email:</label>

            <input type="email" id="signup-email" name="email" required>

            <label for="signup-password">Password:</label>

            <input type="password" id="signup-password" name="password" required>

            <label for="confirm-password">Confirm Password:</label>

            <input type="password" id="confirm-password" name="confirm-password" required>

            <!-- Error Message Container -->

            <div id="create-account-password-error"></div>

            <button type="submit">Create Account</button>

        </form>

    </section>

    <!-- Recharge Card Section -->

    <section id="recharge" class="section">

        <h2>Recharge Your Metro Card</h2>

        <form onsubmit="testRecharge(event)">

            <label for="card-number">Metro Card Number (12 digits):</label>

            <input type="text" id="card-number" name="card-number" maxlength="12" required>

            <div>

                <label>

                    <input type="radio" name="payment" value="credit-card" required> Credit Card

                </label>

                <label>

                    <input type="radio" name="payment" value="debit-card" required> Debit Card

                </label>

                <label>

                    <input type="radio" name="payment" value="net-banking" required> Net Banking

                </label>

                <label>

                    <input type="radio" name="payment" value="upi" required> UPI

                </label>

            </div>

            <button type="submit">Recharge Now</button>

        </form>

    </section>

    <!-- Metro Map Section -->

    <section id="metro-map" class="section">

        <h2>Hyderabad Metro Map</h2>

        <img src="https://routemaps.info/images/maps/hyderabad-metro-map.jpg" alt="Hyderabad Metro Map" style="max-width: 100%; height: auto; border-radius: 10px; box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);">

    </section>

    <!-- Metro Stations Section -->

    <section id="metro-stations" class="section">

        <h2>Metro Stations</h2>

        <img src="https://timesalert.com/wp-content/uploads/2017/12/Hyderabad-Metro-Rail-Stations-List-1024x691.jpg" alt="Hyderabad Metro Stations List" style="max-width: 100%; height: auto; border-radius: 10px; box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);">

    </section>

    <!-- Fare Calculator Section -->

<section id="journey" class="section">

    <h2>Fare Calculator</h2>

    <p>Calculate your fare based on your journey details.</p>

    <!-- Fare Calculator Form -->

    <form onsubmit="calculateFare(event)">

        <label for="from-station">From Station:</label>

        <select id="from-station" required>

            <option value="">Select Station</option>

            <option value="Ameerpet">Ameerpet</option>

            <option value="MG Bus-Station">MG Bus Station</option>

            <option value="Hi-Tech-City">Hi-Tech City</option>

            <option value="Miyapur">Miyapur</option>

            <option value="Raidurg">Raidurg</option>

            <option value="Balanagar">Balanagar</option>

            <option value="Moosapet">Moosapet</option>

            <option value="JBS">JBS</option>

            <option value="Parade Grounds">Parade Grounds</option>

            <option value="Secunderbad">Secunderbad</option>

            <option value="Bharat Nagar">Bharat Nagar</option>

            <option value="SR Nagar">SR Nagar</option>

            <option value="Nampally">Nampally</option>

            <option value="Paradise">Paradise</option>

            <option value="Madhura Nagar">Madhura Nagar</option>

            <option value="Tarnaka">Tarnaka</option>

            <option value="Malakpet">Malakpet</option>

            <option value="ESI Hospital">ESI Hospital</option>

            <option value="Assembly">Assembly</option>

            <option value="Dilsukhnagar">Dilsukhnagar</option>

            <option value="New Market">New Market</option>

            <option value="Kukatpally">Kukatpally</option>

            <option value="Erragadda">Erragadda</option>

            <option value="Khairatabad">Khairatabad</option>

            <option value="Musarambagh">Musarambagh</option>

            <option value="Osmania Medical College">Osmania Medical College</option>

            <option value="Irrum Manzil">Irrum Manzil</option>

            <option value="Chaitanyapuri">Chaitanyapuri</option>

            <option value="Victoria Memorial">Victoria Memorial</option>

            <option value="JNTU College">JNTU College</option>

            <option value="KPHB Colony">KPHB Colony</option>

            <option value="Punjaguuta">Punjagutta</option>

            <option value="Falaknuma">Falaknuma</option>

            <option value="Charminar">Charminar</option>

            <option value="Lakdikapul">Lakdikapul</option>

            <option value="Uppal">Uppal</option>

            <option value="Prakash Nagar">Prakash Nagar</option>

            <option value="Yusuf Guda">Yusuf Guda</option>

            <option value="RTC Cross Roads">RTC Cross Roads</option>

            <option value="NGRI">NGRI</option>

            <option value="Mettuguda">Mettuguda</option>

            <option value="Shamsher Gunj">Shamsher Gunj</option>

            <option value="Shalibanda">Shalibanda</option>

            <option value="Sultan Bazar">Sultan Bazar</option>

            <option value="Gandhi Hospital">Gandhi Hospital</option>

            <option value="Road No 5 Jubllie Hills">Road No 5 Jubllie Hills</option>

            <option value="Jubllie Hills Check Post">Jubllie Hills Check Post</option>

        </select>

        <label for="to-station">To Station:</label>

        <select id="to-station" required>

            <option value="">Select Station</option>

            <option value="Ameerpet">Ameerpet</option>

            <option value="MG Bus-Station">MG Bus Station</option>

            <option value="Hi-Tech-City">Hi-Tech City</option>

            <option value="Miyapur">Miyapur</option>

            <option value="Raidurg">Raidurg</option>

            <option value="Balanagar">Balanagar</option>

            <option value="Moosapet">Moosapet</option>

            <option value="JBS">JBS</option>

            <option value="Parade Grounds">Parade Grounds</option>

            <option value="Secunderbad">Secunderbad</option>

            <option value="Bharat Nagar">Bharat Nagar</option>

            <option value="SR Nagar">SR Nagar</option>

            <option value="Nampally">Nampally</option>

            <option value="Paradise">Paradise</option>

            <option value="Madhura Nagar">Madhura Nagar</option>

            <option value="Tarnaka">Tarnaka</option>

            <option value="Malakpet">Malakpet</option>

            <option value="ESI Hospital">ESI Hospital</option>

            <option value="Assembly">Assembly</option>

            <option value="Dilsukhnagar">Dilsukhnagar</option>

            <option value="New Market">New Market</option>

            <option value="Kukatpally">Kukatpally</option>

            <option value="Erragadda">Erragadda</option>

            <option value="Khairatabad">Khairatabad</option>

            <option value="Musarambagh">Musarambagh</option>

            <option value="Osmania Medical College">Osmania Medical College</option>

            <option value="Irrum Manzil">Irrum Manzil</option>

            <option value="Chaitanyapuri">Chaitanyapuri</option>

            <option value="Victoria Memorial">Victoria Memorial</option>

            <option value="JNTU College">JNTU College</option>

            <option value="KPHB Colony">KPHB Colony</option>

            <option value="Punjaguuta">Punjagutta</option>

            <option value="Falaknuma">Falaknuma</option>

            <option value="Charminar">Charminar</option>

            <option value="Lakdikapul">Lakdikapul</option>

            <option value="Uppal">Uppal</option>

            <option value="Prakash Nagar">Prakash Nagar</option>

            <option value="Yusuf Guda">Yusuf Guda</option>

            <option value="RTC Cross Roads">RTC Cross Roads</option>

            <option value="NGRI">NGRI</option>

            <option value="Mettuguda">Mettuguda</option>

            <option value="Shamsher Gunj">Shamsher Gunj</option>

            <option value="Shalibanda">Shalibanda</option>

            <option value="Sultan Bazar">Sultan Bazar</option>

            <option value="Gandhi Hospital">Gandhi Hospital</option>

            <option value="Road No 5 Jubllie Hills">Road No 5 Jubllie Hills</option>

            <option value="Jubllie Hills Check Post">Jubllie Hills Check Post</option>

        </select>

        <button type="submit">Calculate Fare</button>

    </form>

    <div id="fare-result" style="margin-top: 20px; display: none;">

        <h3>Fare Result</h3>

        <p id="fare-message"></p>

    </div>

</section>

<script>

    function calculateFare(event) {

        event.preventDefault();

        const fromStation = document.getElementById('from-station').value;

        const toStation = document.getElementById('to-station').value;

        if (!fromStation || !toStation || fromStation === toStation) {

            alert("Please select valid stations for your journey.");

            return;

        }

        // Simple fare calculation (demo purposes)

        let fare = 0;

        // Example of fare calculation (just a basic demo logic for the sake of this example)

        if (fromStation === "Ameerpet" && toStation === "MG-Bus-Station") {

            fare = 30; // Example fare in INR

        } else if (fromStation === "Hi-Tech-City" && toStation === "Raidurg") {

            fare = 40;

        } else {

            fare = 25;

        }

        // Show fare result

        const fareResultDiv = document.getElementById('fare-result');

        const fareMessage = document.getElementById('fare-message');

        fareMessage.textContent = `Your fare from ${fromStation} to ${toStation} is ₹${fare}.`;

        fareResultDiv.style.display = "block";

        alert("Fare calculated successfully!");

    }

</script>

    <!-- Feedback Section -->

    <section id="feedback" class="section">

        <h2>Feedback</h2>

        <form onsubmit="testFeedbackForm(event)">

            <textarea id="feedback-message" rows="4" placeholder="Enter your feedback here..." required></textarea>

            <button type="submit">Submit Feedback</button>

        </form>

    </section>

    <!-- Contact Section -->

    <section id="contact" class="section">

        <h2>Contact Us</h2>

        <p> Hyderabad Metro Rail Limited

            Rasoolpura,Begumpet

            Hyderabad-500 016.

        </p>

        <p>Email: support@hyderabadmetro.com</p>

        <p>Talk to Us

            Tel: +91 40 233 88 587/588

        Fax: +91 40 23388582</p>

    </section>

    <!-- My Account Section -->

    <section id="account" class="section">

        <h2>My Account</h2>

        <p id="account-name">Name: N/A</p>

        <p id="account-email">Email: N/A</p>

    </section>

    <!-- Footer Section -->

    <footer>

        <p>© 2024 Hyderabad Metro Rail. All Rights Reserved.</p>

    </footer>

    <script>

        // Password validation functions

        function validateLoginPassword() {

            const password = document.getElementById('login-password').value;

            const errorMessage = document.getElementById('login-password-error');

            errorMessage.textContent = '';

            // Test Case 4: Password field is required

            if (password === '') {

                errorMessage.textContent = 'Password is required.';

                return false;

            }

            // Test Case 1: Password must be at least 8 characters long

            if (password.length < 8) {

                errorMessage.textContent = 'Password must be at least 8 characters long.';

                return false;

            }

            return true;

        }

        function validateCreateAccountPassword() {

            const password = document.getElementById('signup-password').value;

            const confirmPassword = document.getElementById('confirm-password').value;

            const errorMessage = document.getElementById('create-account-password-error');

            errorMessage.textContent = '';

            // Test Case 4: Password field is required

            if (password === '') {

                errorMessage.textContent = 'Password is required.';

                return false;

            }

            // Test Case 1: Password must be at least 8 characters long

            if (password.length < 8) {

                errorMessage.textContent = 'Password must be at least 8 characters long.';

                return false;

            }

            // Test Case 2: Password must contain at least one uppercase letter, one lowercase letter, and one digit

            const passwordPattern = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)/;

            if (!passwordPattern.test(password)) {

                errorMessage.textContent = 'Password must contain at least one uppercase letter, one lowercase letter, and one digit.';

                return false;

            }

            // Test Case 3: Password and confirm password must match

            if (password !== confirmPassword) {

                errorMessage.textContent = 'Passwords do not match.';

                return false;

            }

            // Test Case 5: Confirm password field is required

            if (confirmPassword === '') {

                errorMessage.textContent = 'Confirm password is required.';

                return false;

            }

            return true;

        }

        // Event listeners for form submissions

        document.querySelector("#login-form").addEventListener("submit", function(event) {

            if (!validateLoginPassword()) {

                event.preventDefault();

            }

        });

        document.querySelector("#signup-form").addEventListener("submit", function(event) {

            if (!validateCreateAccountPassword()) {

                event.preventDefault();

            }

        });

        let userData = {}; // Object to store user details

    // Save login details and show account

    function handleLogin(event) {

        event.preventDefault();

        const email = document.getElementById('login-email').value;

        if (validateLoginPassword()) {

            userData.email = email;

            alert("Login successful!");

            showSection('account');

            updateAccountPage();

        }

    }

    // Save signup details and show account

    function handleCreateAccount(event) {

        event.preventDefault();

        const email = document.getElementById('signup-email').value;

        const name = document.getElementById('signup-name').value;

        if (validateCreateAccountPassword()) {

            userData.name = name;

            userData.email = email;

            alert("Account created successfully!");

            showSection('account');

            updateAccountPage();

        }

    }

    // Display user details on My Account page

    function updateAccountPage() {

        document.getElementById('account-name').textContent = `Name: ${userData.name || 'N/A'}`;

        document.getElementById('account-email').textContent = `Email: ${userData.email || 'N/A'}`;

    }

    // Switch between sections

    function showSection(sectionId) {

        const sections = document.querySelectorAll('.section');

        sections.forEach(section => section.classList.remove('active'));

        document.getElementById(sectionId).classList.add('active');

    }

    document.querySelector("#login-form").addEventListener("submit", handleLogin);

    document.querySelector("#signup-form").addEventListener("submit", handleCreateAccount);

    const users = {}; // Object to store multiple users' details by email

    // Save signup details and ensure the email is unique

    function handleCreateAccount(event) {

        event.preventDefault();

        const email = document.getElementById('signup-email').value;

        const name = document.getElementById('signup-name').value;

        const password = document.getElementById('signup-password').value;

        if (users[email]) {

            alert("An account with this email already exists.");

            return;

        }

        if (validateCreateAccountPassword()) {

            users[email] = { name, password };

            alert("Account created successfully!");

            showSection('login'); // Direct the user to the login page after successful sign-up

        }

    }

    // Verify login details match a created account

    function handleLogin(event) {

        event.preventDefault();

        const email = document.getElementById('login-email').value;

        const password = document.getElementById('login-password').value;

        if (!users[email]) {

            alert("No account found with this email.");

            return;

        }

        if (users[email].password !== password) {

            alert("Incorrect password.");

            return;

        }

        // Store user details for displaying on My Account page

        userData.name = users[email].name;

        userData.email = email;

        alert("Login successful!");

        showSection('account');

        updateAccountPage();

    }

    // Display user details on My Account page

    function updateAccountPage() {

        document.getElementById('account-name').textContent = `Name: ${userData.name || 'N/A'}`;

        document.getElementById('account-email').textContent = `Email: ${userData.email || 'N/A'}`;

    }

    // Switch between sections

    function showSection(sectionId) {

        const sections = document.querySelectorAll('.section');

        sections.forEach(section => section.classList.remove('active'));

        document.getElementById(sectionId).classList.add('active');

    }

    document.querySelector("#login-form").addEventListener("submit", handleLogin);

    document.querySelector("#signup-form").addEventListener("submit", handleCreateAccount);

    </script>

</body>

</html>

**TEST CASES**

Password Test Case:

Test Case 1: Password must be at least 8 characters long.

Test Case 2: Password must contain at least one uppercase letter, one lowercase letter, and one digit.

Test Case 3: Password and confirm password must match.

Test Case 4: Password field is required.

Test Case 5: Confirm password field is required and must match the password.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected output | Actual output | Status |
| 1234567 | Rejected(password must be at least 8 characters long) | Password must be at least 8 characters long. | Success |
| nivi7667 | Rejected(at least one upper case letter) | Password must contain at least one uppercase letter | Success |
| David | Rejected(at least one digit) | Password must contain at least one digit | Success |
| DAVID6768 | Rejected(at least one lower case letter) | Password must contain at least one lowercase letter | Success |
| [empty] | Rejected(must not be empty) | Password field is required | Success |
| David@1234 | Successful | Successful | Success |

Additional Test Case:

Test Case 1:Email and password field is required

Test Case 2: Confirm password field is required and must match the password.

Test Case 3: Metro card number must consists of 12 digits.

Test Case 4: Journey Fare calculator must fill the both the source and final destination to calculate the fare

Test Case 5: Verify contact information displays correctly

Test Case 6: Check links for email and phone numbers

Test Case 7: In journey fare calculator both the source and destination should be different from each other

Test Case 8: Before logging into your account we should create an account

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Status |
| Email and password | User has to fill the field compulsory | Please fill out this field | Success |
| Password Field | Two fields of password must match to each other | Passwords do not match | Success |
| Metro Card | Card number should be 12 digits | Metro Card Number must be 12 digits. | Success |
| Journey Fare Calculator | Source and destination must be filled | Please select the station in the list | Success |
| Contact Information | Information must be displayed correctly | My Account section | Success |
| Journey Fare Calculator | Both source and destination stations should be different | Please select valid stations for your journey | Success |
| Login | Account must be created before login | No account found with this email. | Success |

**CONCLUSION**

This is an issue that occurred very often at the customer care centres regarding long queues, and the Smart Metro Card Recharge System would enable metro commuters to reload the smart cards online. If integrated into the existing metro infrastructure, this system would enable the HMRL to make this experience more efficient and seamless for passengers. The user-friendly interface, options for multiple payment methods, and extra features like calculating fares, navigating through stations, and personalizing settings make the overall benefit to commuters much easier.

The online recharge portal benefits the user from not having to stand for long hours in a queue and also gets up-to-date balance update information for quick and hassle-free travel. Additionally, the scalability of the system ensures that when there are sharp increases in user traffic-perhaps at peak hours-it still manages to effectively serve users without degrading performance or reliability.

**RESULTS**

The efficiency of the whole metro system is improved due to the reduction of long queues and waiting time to recharge at metro stations with taking the recharge process online.

**Enhanced User Experience:** Features like fare calculation, navigation of stations, and personalized settings make the system easier to use, thus allowing users to have a more streamlined metro commuting experience.

**Scalability and Reliability:** The system would be able to handle high traffic loads, and it would not degrade in performance, especially at peak hours. Its reliability and fault tolerance will ensure that downtime is minimal, meaning users will receive a seamless service.

--THE END--