Zee Pay

Arhaan Bhiwandkar
Arya Chawan
Dev Poriya
Sharvil Gharkar

Table of contents

Introduction

In today's fast-paced world, convenience reigns supreme. That's where our innovative e-wallet comes in. It's a secure and digital way to manage your finances, ditching the bulk of a traditional wallet overflowing with cash and cards. Instead, our e-wallet stores your payment information securely on your phone or other device, allowing you to make instant payments to friends, stores, and online platforms with just a few taps. No more scrambling for cash or waiting for change – it's the epitome of financial ease.

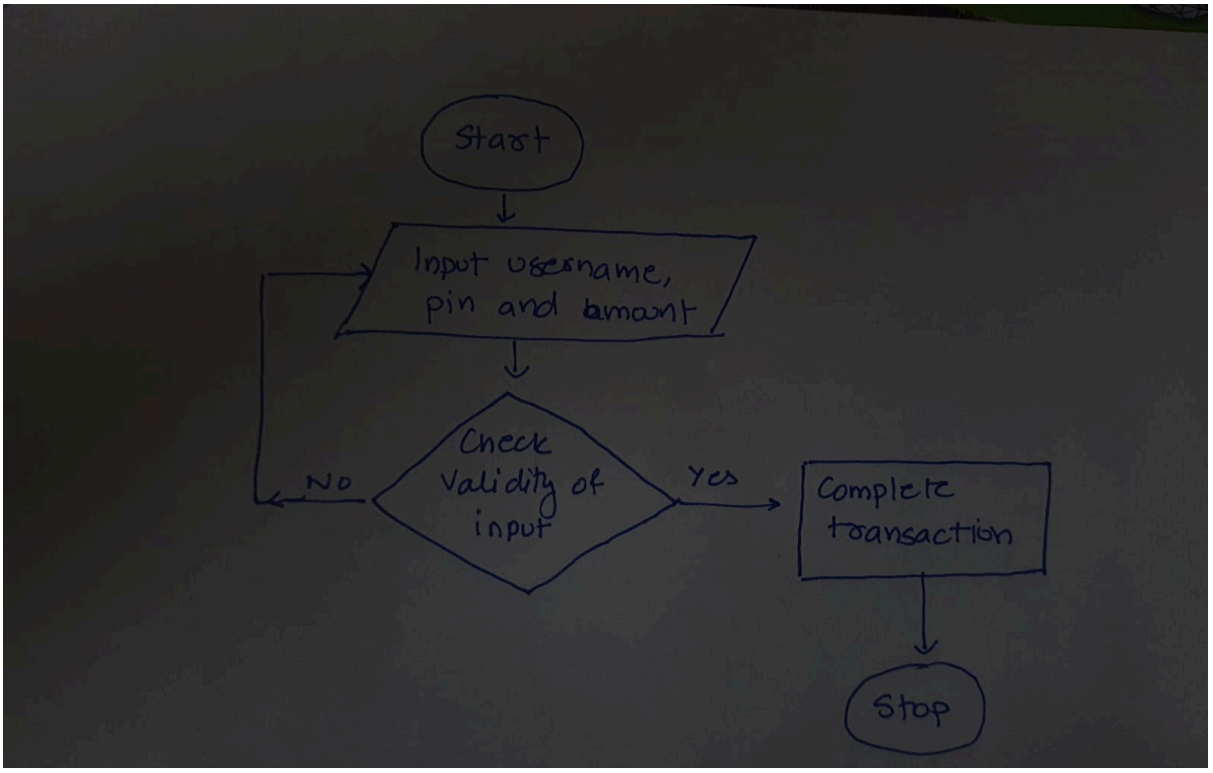
But our e-wallet goes beyond basic transactions. We offer two unique features that revolutionize how you manage money with friends and contacts. With "Quick Pay," say goodbye to scrambling for names and account details. This intelligent feature remembers your recent transactions, allowing you to send money to familiar contacts with just a few taps. No more awkward conversations about getting repaid either. The "Request Money" feature lets you send a clear, digital request to any user. They'll receive a

notification with the amount and any details you provide, making it a smooth and effortless experience to settle debts.

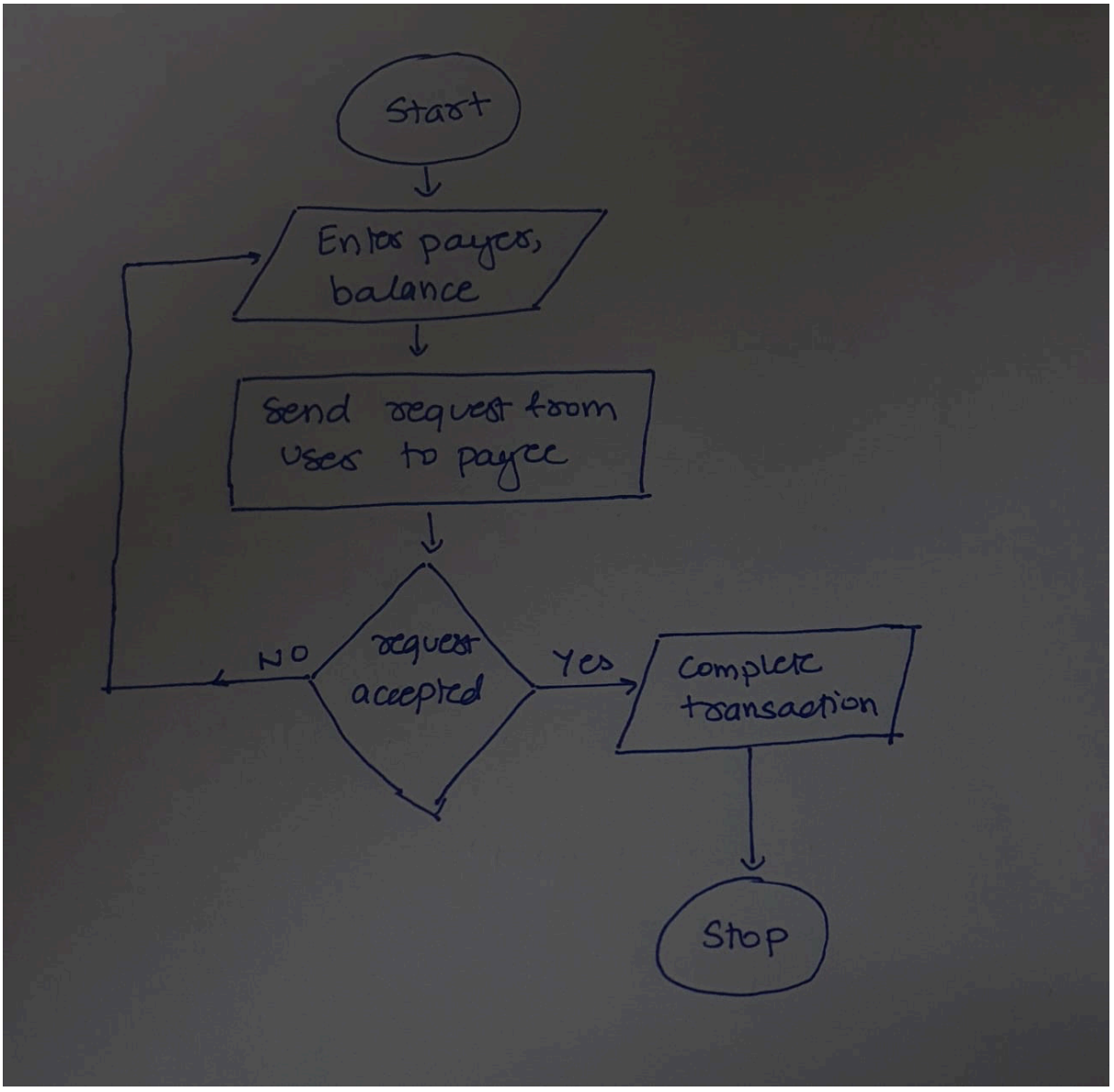
So, ditch the hassle of traditional wallets and embrace the future of financial management with our innovative e-wallet. It's secure, convenient, and packed with features that make splitting bills and settling debts with friends a breeze.

Flow Chart

E-Wallet Flow chart



“Request Payment” Flowchart



Architectural Details

Frontend (HTML/CSS & Javascript):

User Interface (UI): This is the visual layer users interact with. HTML structures the content (pages, forms, buttons), CSS styles the UI elements (colors, fonts, layouts), and Javascript adds interactivity (animations, form validation, user actions).

Frontend Logic: Javascript code handles user interactions within the browser. The buttons in the form send requests to the backend for data manipulation, and updates the UI based on responses.

Backend (Flask & Supabase):

Flask Application: This is the core backend server written in Python with Flask. It handles incoming requests from the frontend, interacts with the database (Supabase), and generates responses.

Supabase Integration: Supabase acts as the e-wallet's database. Flask interacts with Supabase using its Python libraries to perform actions like user authentication, storing transaction data, and retrieving user information.

API Endpoints: Flask defines API endpoints (specific URLs) for the frontend to interact with. These endpoints handle functionalities like user login, sending money, requesting money, and retrieving transaction history.

Data Flow:

1. User interacts with the UI (e.g., clicks a button to send money).
2. Frontend Javascript sends a request with relevant data (amount, recipient) to a specific Flask API endpoint.

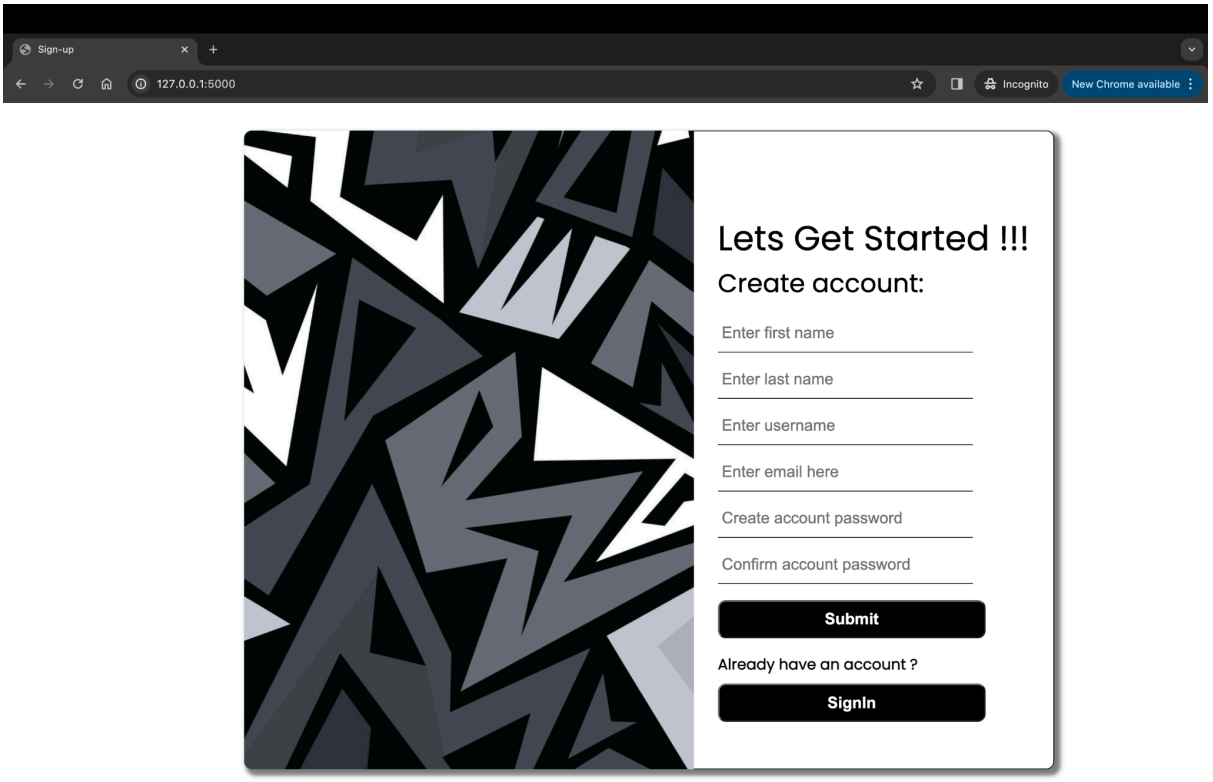
3. Flask processes the request, interacts with Supabase to transfer funds or create a request.
4. Supabase performs the database operations (update balances, create entries).
5. Flask sends a response back to the frontend with success/failure information or updated data.
6. Javascript updates the UI based on the response (e.g., displays confirmation message, updates balance).

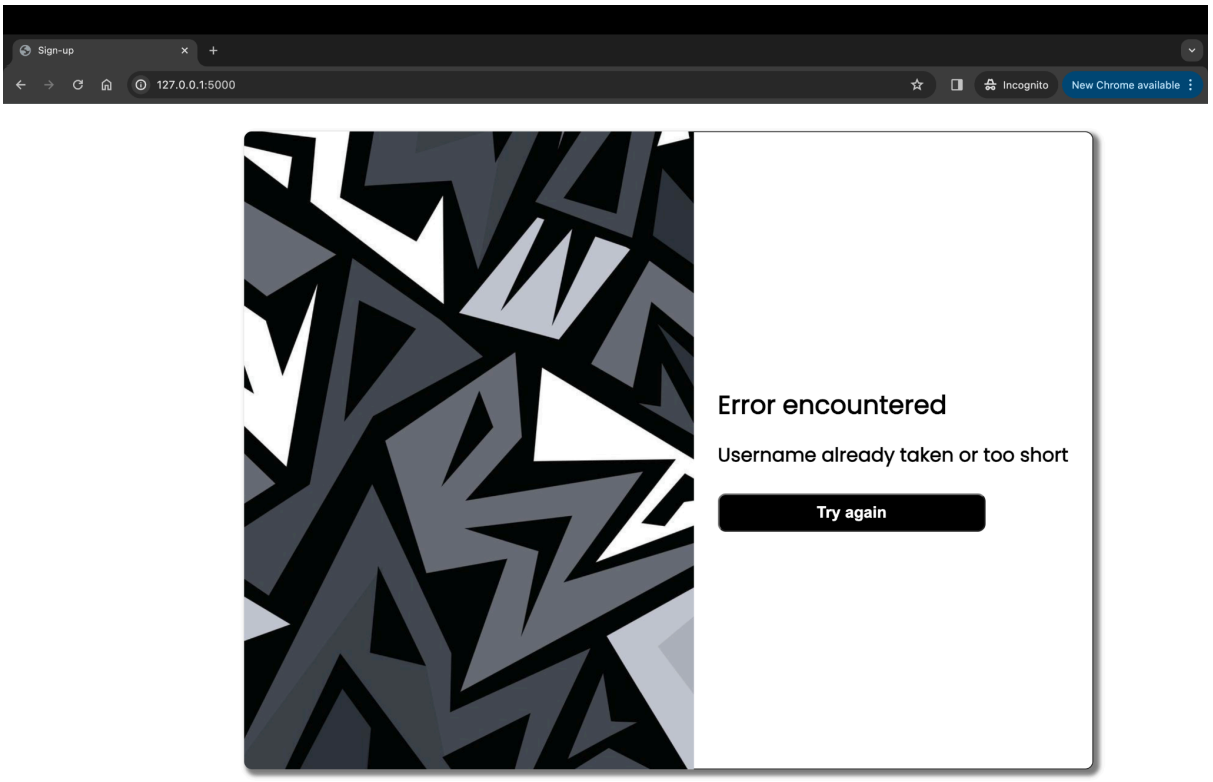
Additional Considerations:

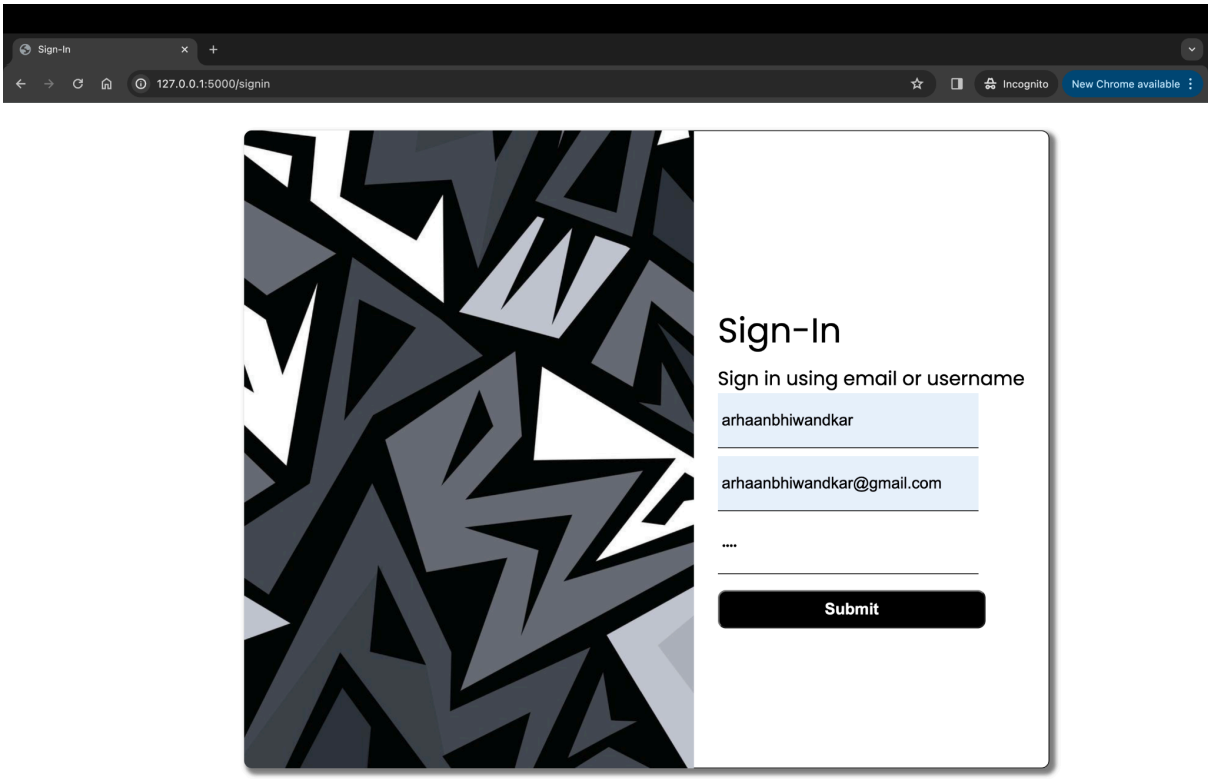
Security: Flask should implement security measures like user authentication and authorization to protect sensitive user data.

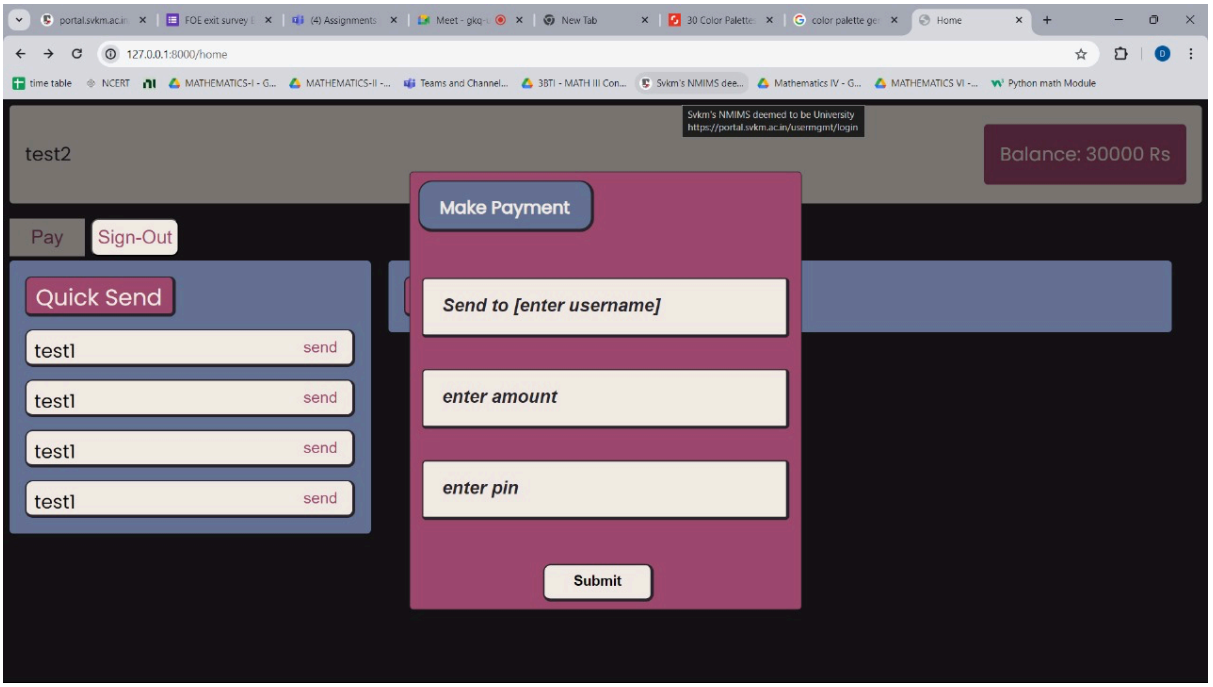
Error Handling: Both frontend and backend should handle potential errors gracefully, providing informative messages to the user.

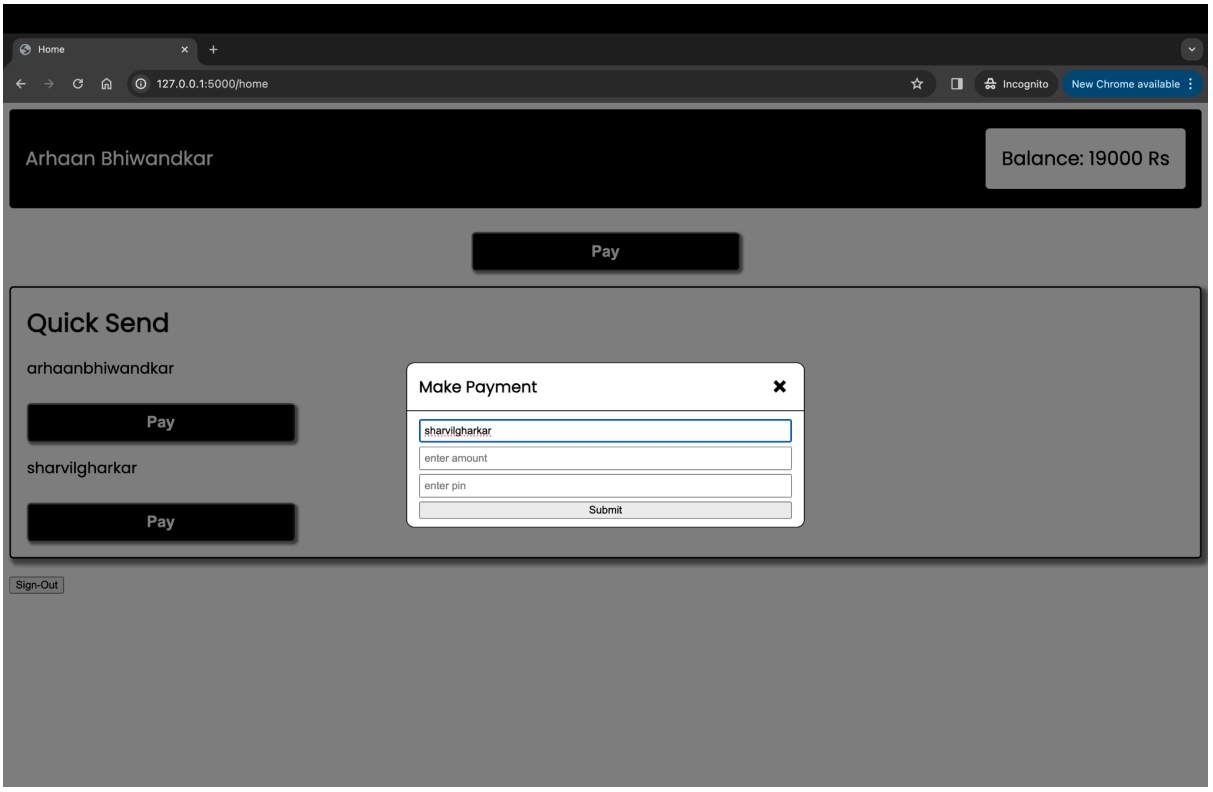
Output Screenshots






Sign-In
127.0.0.1:5000/signin
Incognito
New Chrome available
Sign-In
Sign in using email or username
arhaanbhiwandkar
arhaanbhiwandkar@gmail.com
....
Submit


127.0.0.1:8000/home
test2
Balance: 30000 Rs
Pay
Sign-Out
Quick Send
test1
send
test1
send
test1
send
test1
send
Make Payment
Send to [enter username]
enter amount
enter pin
Submit


Home
127.0.0.1:5000/home
Incognito
New Chrome available
Arhaan Bhiwandkar
Balance: 19000 Rs
Pay
Quick Send
arhaanbhiwandkar
Pay
sharvilgharkar
Pay
Make Payment
sharvilgharkar
enter amount
enter pin
Submit
Sign-Out

Conclusions

In conclusion, this e-wallet application transcends the limitations of traditional wallets, offering a future-proof solution for managing your finances. Built with a powerful combination of technologies – Flask, Javascript, HTML/CSS, and Supabase – it prioritizes both security and convenience.

Supabase, the application's backbone, acts as a secure and scalable database, ensuring the safekeeping of your financial information. The Flask framework provides a robust foundation for the backend, handling server-side operations efficiently. Meanwhile, the user interface, crafted with HTML/CSS and brought to life with Javascript, delivers an intuitive and user-friendly experience.

This isn't just another e-wallet; it's an innovation that streamlines financial interactions with friends and contacts. The "Quick Pay" feature eliminates the hassle of repeatedly entering payment details, while the "Request Money" feature gracefully handles the awkwardness of requesting repayments. With a few taps, you can send or request money, fostering a smooth and effortless financial ecosystem.

Imagine a world where settling bills with friends becomes a breeze, and requesting money back is a frictionless process. This e-wallet app has the potential to make that a reality. It's a testament to the power of technological innovation, simplifying financial management and fostering stronger financial connections with those around you. By embracing this app, you're not just managing your money electronically; you're stepping into a future of financial ease and empowerment