Report

1. Presentation Layer

The presentation layer of this application serves as the user interface through which users interact with the system. It encompasses the design and layout of various components such as web pages, forms, navigation menus, and interactive elements. In this project, the presentation layer has been developed to provide users with an intuitive and visually appealing experience.

Fully Implemented:
I. User Interface Design: The UI design follows modern design principles, with attention to usability and aesthetics. It includes intuitive layouts, clear navigation paths, and visually appealing elements.
II. Navigation Menus: The application includes navigation menus that enable users to easily move between different sections of the system. These menus provide clear labels and hierarchical structures for efficient navigation.
III. Interaction Elements: There are various interaction elements like buttons, links, input fields, and dropdown menus have been implemented to facilitate user actions and inputs. These elements are designed to provide clear cues and feedback to users.
IV. Feedback Mechanisms: The presentation layer incorporates feedback mechanisms such as error messages, success notifications, and confirmation dialogs to communicate with users effectively.

Partially Implemented:
I. Visual Enhancements: While the basic UI components and layouts were used, visual enhancements and refinements can still be done to typography, color schemes, iconography, and overall visual consistency.

2. Business Logic Layer

The business logic layer of the application contains the core logic and algorithms that govern the behavior and functionality of the system. It encompasses processes such as *data validation, calculation, and decision-making based on business rules.* In my project, the business logic layer has been developed to handle various operations related to financial transactions and business processes.

Fully Implemented:
1. Data Validation Logic: The application includes robust data validation logic to ensure the integrity and validity of user inputs. This logic validates user-submitted data against predefined rules and constraints, preventing the entry of invalid or erroneous data.
2. Calculation and Processing Algorithms: Complex algorithms for financial calculations, transaction processing, and other business operations have been implemented successfully. These algorithms ensure accurate calculations and consistent processing of data.
3. Integration with External Services: The business logic layer integrates seamlessly with external APIs, services, or databases to extend the functionality of the system. This integration allows the application to interact with external systems for tasks such as payment processing, data synchronization, or authentication.

3. Data Access Layer

The data access layer of the application is responsible for interacting with the underlying database or data storage system. It facilitates the *retrieval, insertion, updating, and deletion of data entities*, as well as the execution of database queries and transactions. In my project, the data access layer plays a crucial role in ensuring efficient and reliable access to data for various components of the system.

Fully Implemented:
1. Object-Relational Mapping (ORM): The application utilizes an ORM framework such as Django's ORM to abstract database interactions and work with high-level Python objects. This approach simplifies database access and management, allowing developers to focus on application logic rather than low-level SQL queries.
2. CRUD Operations: The data access layer supports basic CRUD (Create, Read, Update, Delete) operations for interacting with database entities. This includes methods for creating new records, retrieving existing records, updating data, and deleting records as needed.
3. Query Optimization: Database queries are optimized for performance and efficiency to minimize response times and resource utilization.

4. Security Layer

The security layer of the application is responsible for safeguarding the system against unauthorized access, data breaches, and malicious attacks. It encompasses various measures and mechanisms to protect sensitive information, authenticate users, authorize access to resources, and enforce security policies throughout the application.

Fully Implemented:
1. Authentication: The application implements robust authentication mechanisms to verify the identity of users before granting access to protected resources. This include username/password authentication.
2. Input Validation: Input validation techniques are applied to prevent common security vulnerabilities such as cross-site request forgery (CSRF). This includes validating and sanitizing user input before processing it to mitigate the risk of injection attacks and data manipulation.

5. Web Services

The web services layer of the application facilitates communication and data exchange between different systems or components over the network. It enables interoperability between heterogeneous systems by providing standardized interfaces and protocols for accessing and manipulating data. The web services layer include both RESTful APIs and SOAP-based services, depending on the requirements of the application.

Fully Implemented:
1. Representational State Transfer (REST): RESTful APIs are fully implemented to provide a lightweight, scalable, and stateless communication architecture for accessing and manipulating resources. REST endpoints are designed following REST principles, including the use of HTTP methods (GET, POST, PUT, DELETE) for CRUD operations, resource-based URLs, and stateless communication.
2. JSON (JavaScript Object Notation): JSON is used as the preferred data format for representing and transferring data between clients and servers in RESTful web services. JSON provides a lightweight and human-readable format that is widely supported by modern web applications and programming languages.

3. Authentication and Authorization: RESTful APIs support authentication and authorization mechanisms to secure access to resources and enforce access control policies. This may include token-based authentication using JWT (JSON Web Tokens), OAuth authentication with access tokens, or basic authentication with API keys.

6. Remote Procedure Call (RPC)

The RPC layer of the application enables communication between different processes or systems by allowing one program to invoke procedures or functions on another system across a network. RPC provides a mechanism for distributed computing, allowing clients to call remote procedures as if they were local functions, abstracting the underlying network communication details.

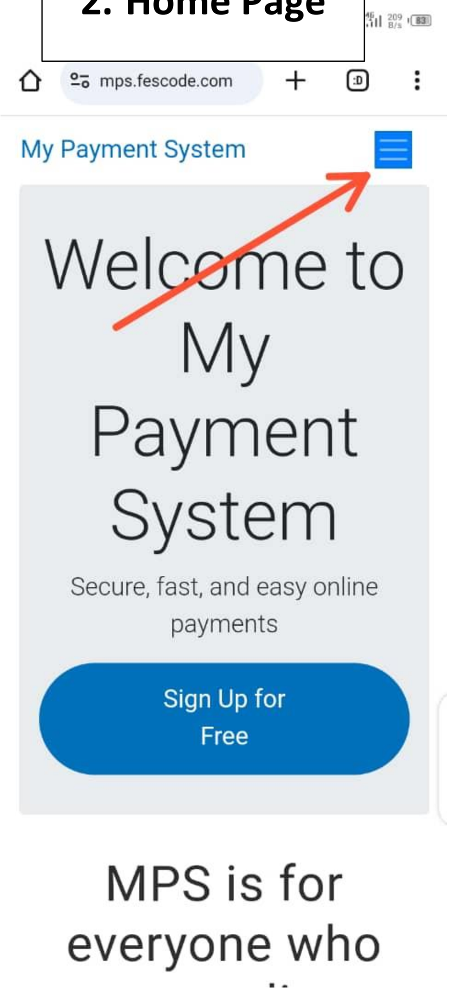
Fully Implemented:
1. RPC Framework: A robust RPC framework is fully implemented to facilitate communication between distributed components of the application.

7. Cloud

The cloud computing layer of the application leverages cloud infrastructure and services to deliver scalable, reliable, and cost-effective solutions. By utilizing cloud resources, the application gains access to a wide range of computing resources, including servers, storage, databases, and networking capabilities, without the need for upfront investment in physical hardware or infrastructure management.

Requirements Not Implemented:
1. Cloud Infrastructure: The application is yet to be deployed on a cloud platform, such as Amazon Web Services (AWS).

2. Home Page

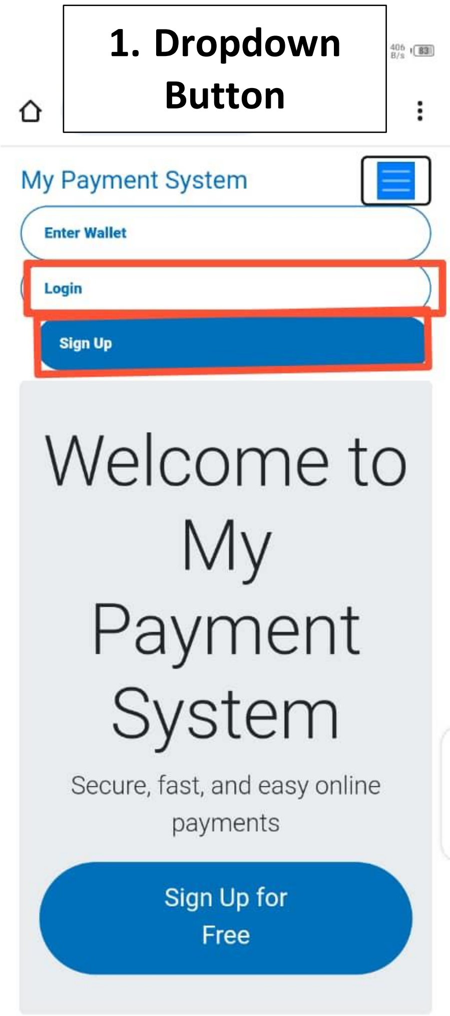


Welcome to My Payment System

Secure, fast, and easy online payments

MPS is for everyone who

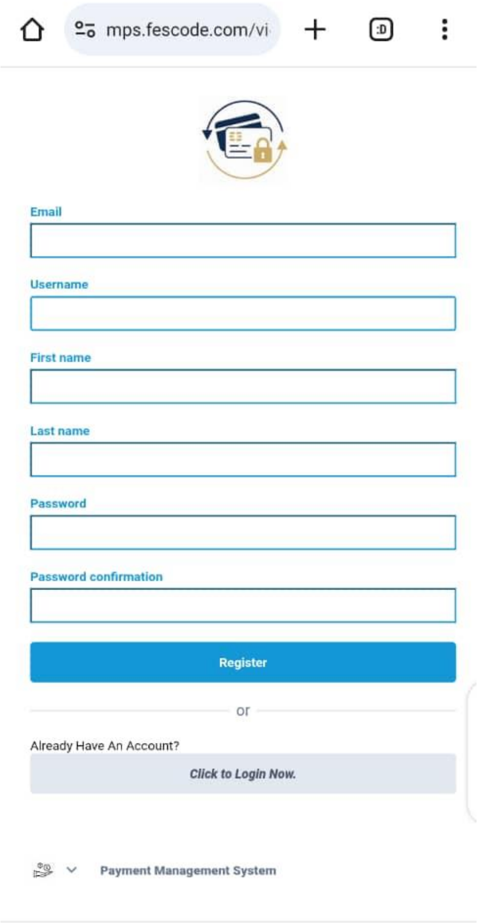
1. Dropdown Button



Welcome to My Payment System

Secure, fast, and easy online payments

4. Signup Page



mps.fescode.com/vi

Email

Username

First name

Last name

Password

Password confirmation

Register

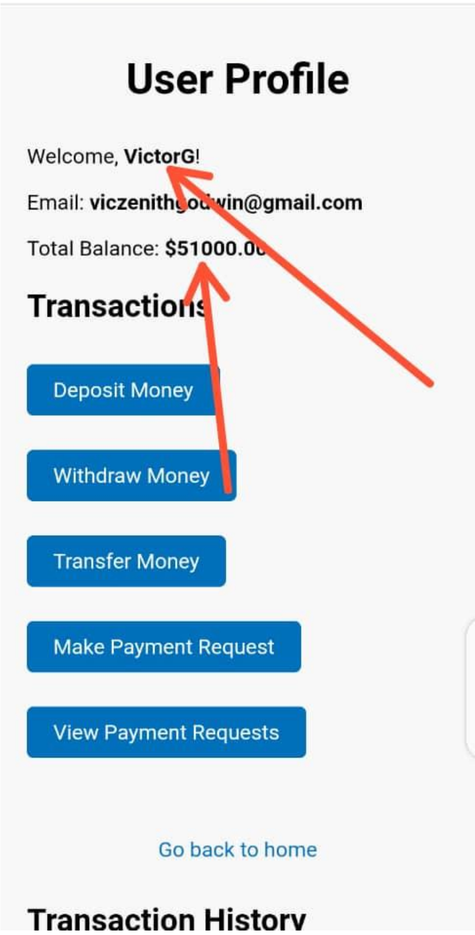
or

Already Have An Account?

Click to Login Now.

Payment Management System

3. Account Page



User Profile

Welcome, **VictorG**!

Email: **viczenithgodwin@gmail.com**

Total Balance: **$51000.00**

Transactions

Deposit Money

Withdraw Money

Transfer Money

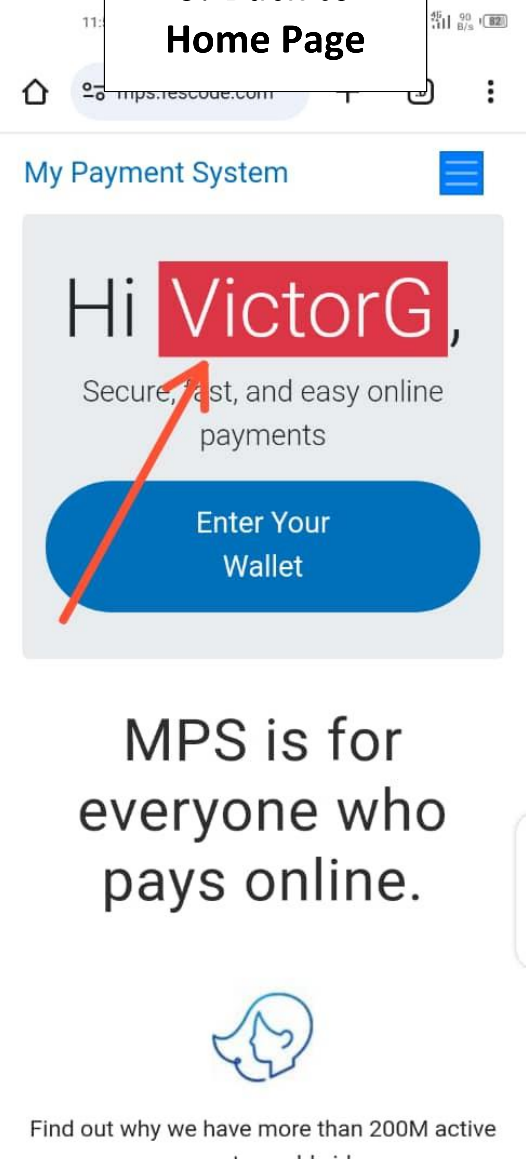
Make Payment Request

View Payment Requests

Go back to home

Transaction History

5. Back to Home Page



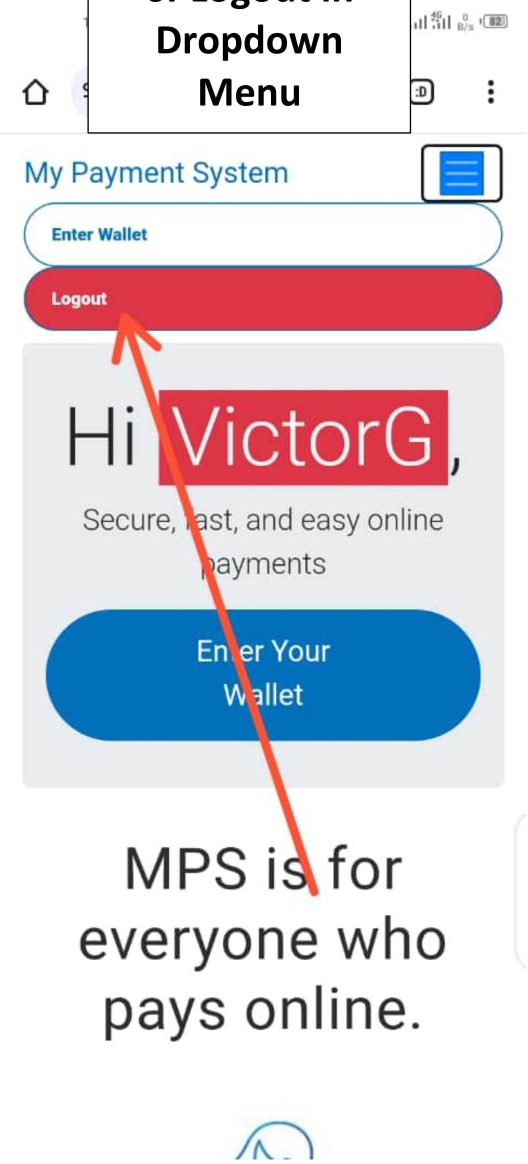
Hi VictorG,

Secure, fast, and easy online payments

MPS is for everyone who pays online.

Find out why we have more than 200M active

6. Logout in Dropdown Menu

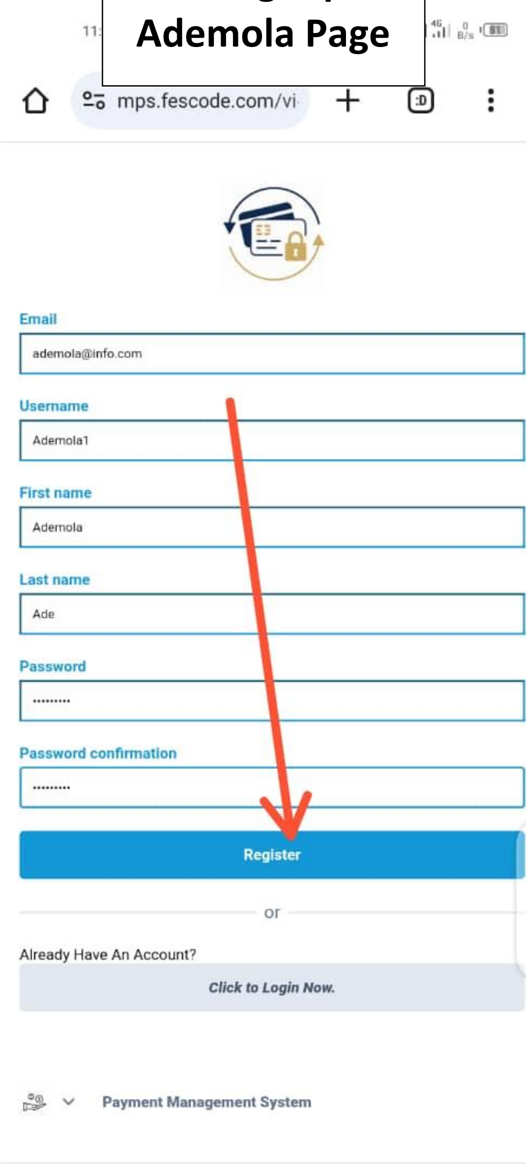


Hi VictorG,

Secure, fast, and easy online payments

MPS is for everyone who pays online.

8. Signup Ademola Page



Email: ademola@info.com

Username: Ademola1

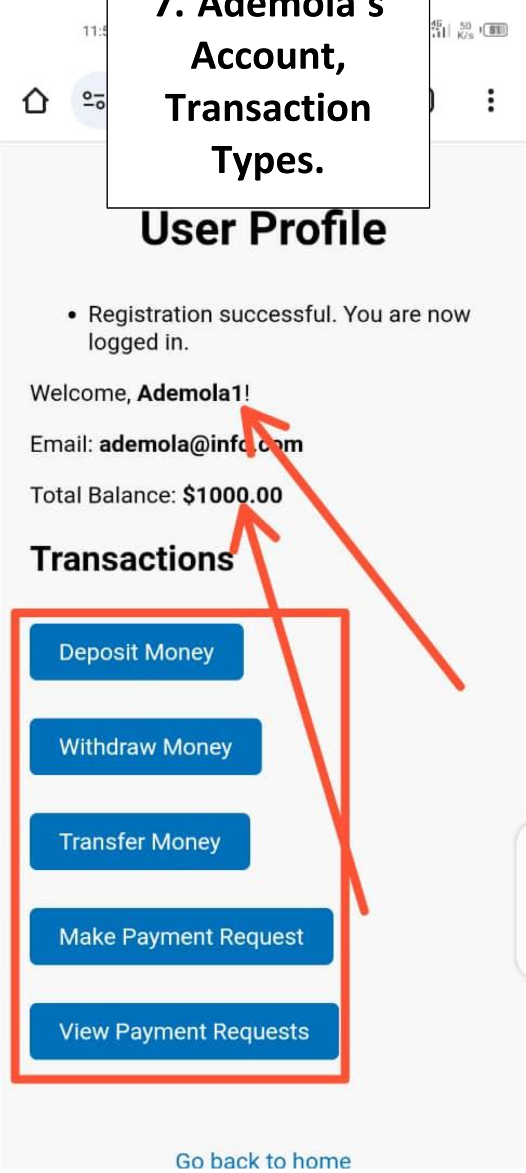
First name: Ademola

Last name: Ade

Password: ·········

Password confirmation: ·········

7. Ademola's Account, Transaction Types.



User Profile

- Registration successful. You are now logged in.

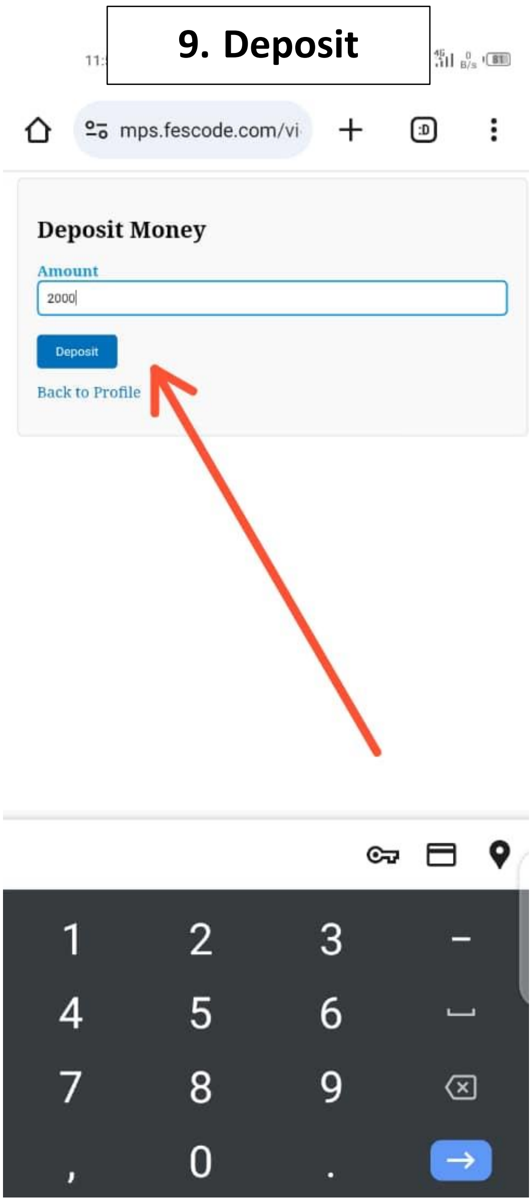
Welcome, **Ademola1**!

Email: **ademola@info.com**

Total Balance: **$1000.00**

Transactions

9. Deposit

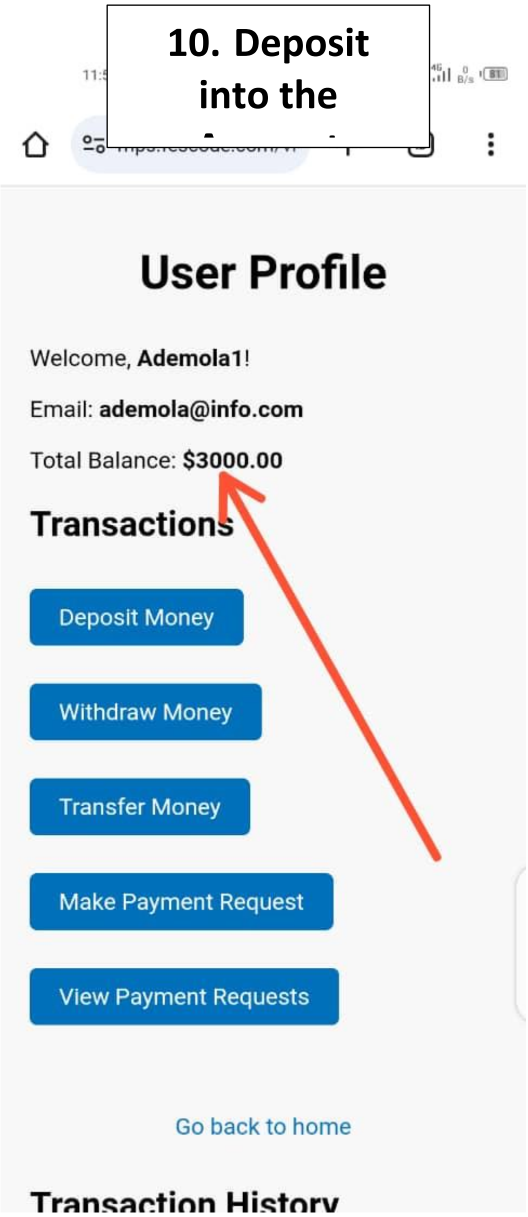


Deposit Money

Amount

2000

10. Deposit into the



User Profile

Welcome, **Ademola1**!

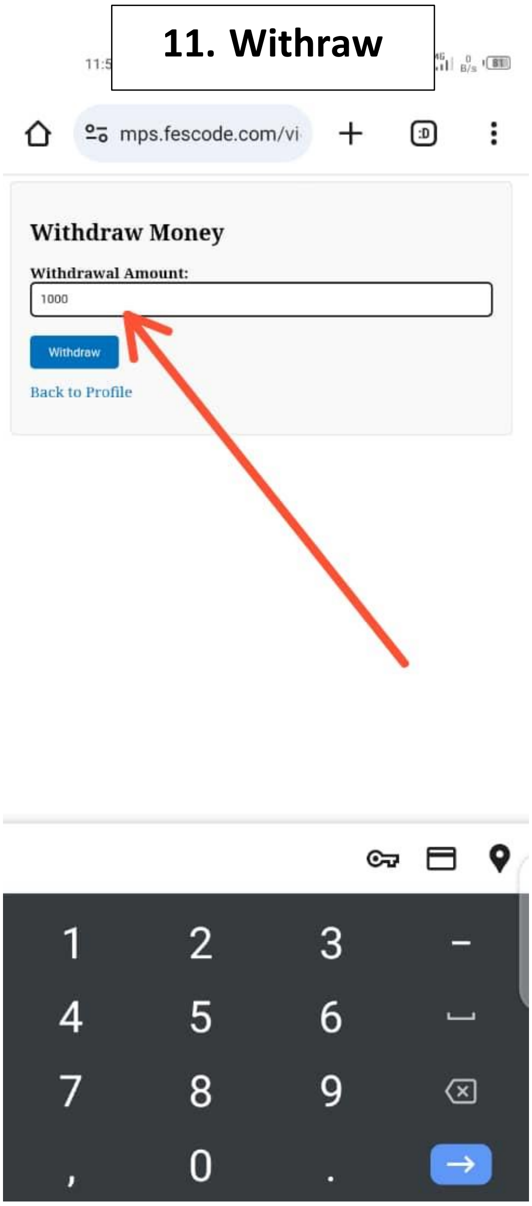
Email: **ademola@info.com**

Total Balance: **$3000.00**

Transactions

Transaction History

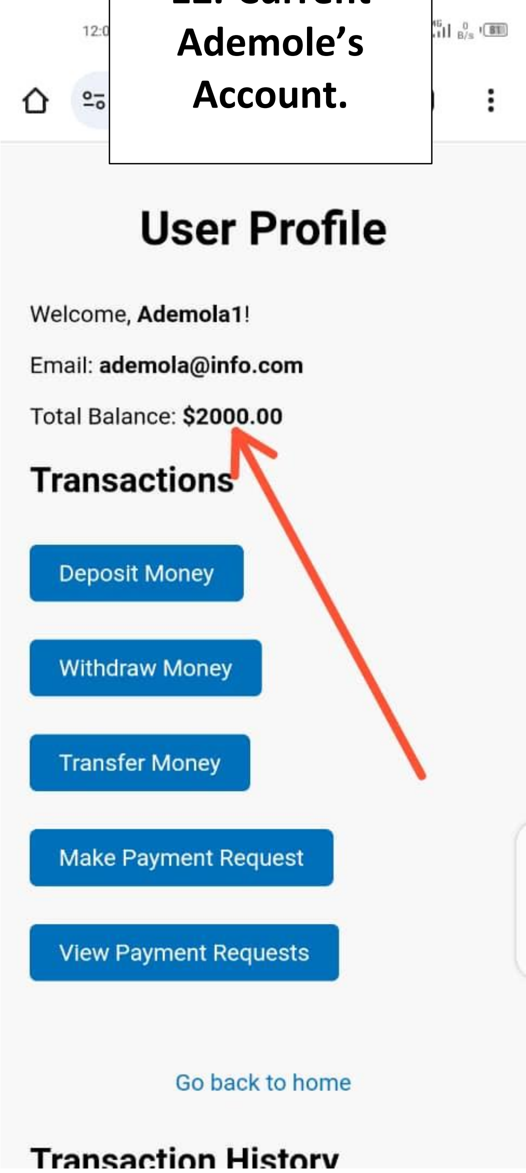
11. Withraw



Withdraw Money

Withdrawal Amount: 1000

12. Current Ademole's Account.



User Profile

Welcome, **Ademola1**!

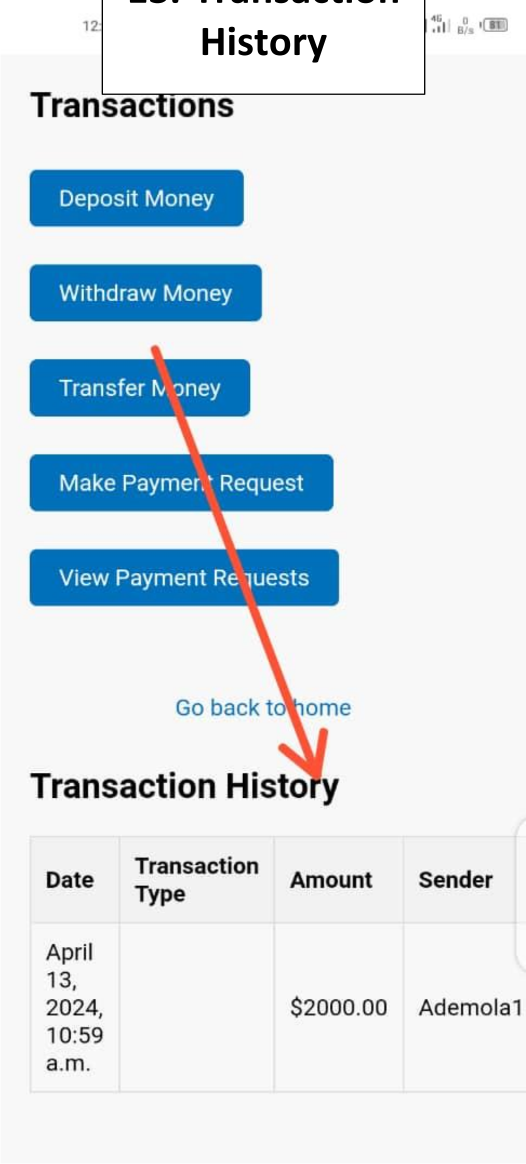
Email: **ademola@info.com**

Total Balance: **$2000.00**

Transactions

Transaction History

13. Transaction History

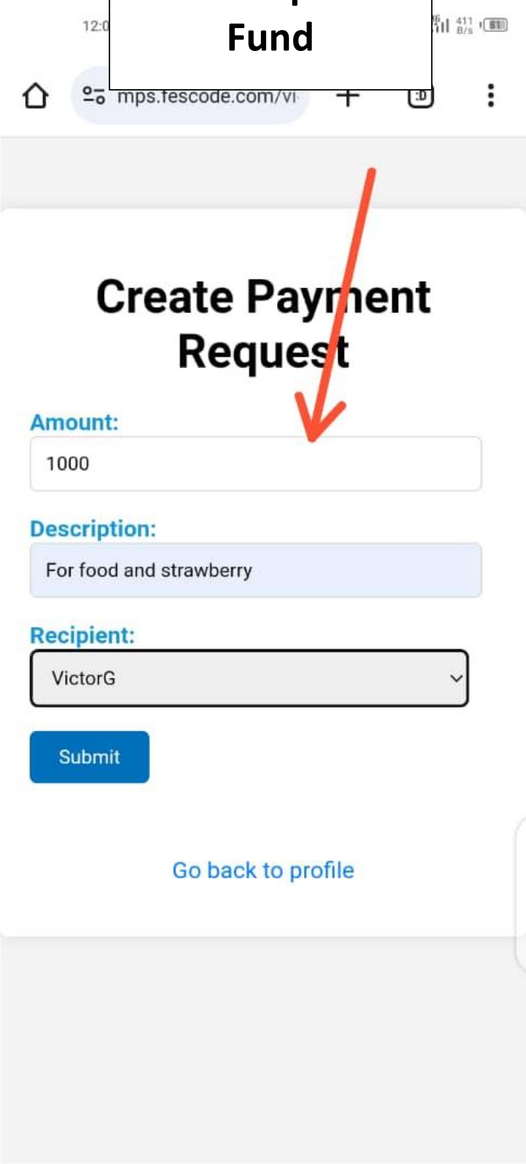


Transactions

Transaction History

Date	Transaction Type	Amount	Sender
April 13, 2024, 10:59 a.m.		$2000.00	Ademola1

14. Request Fund



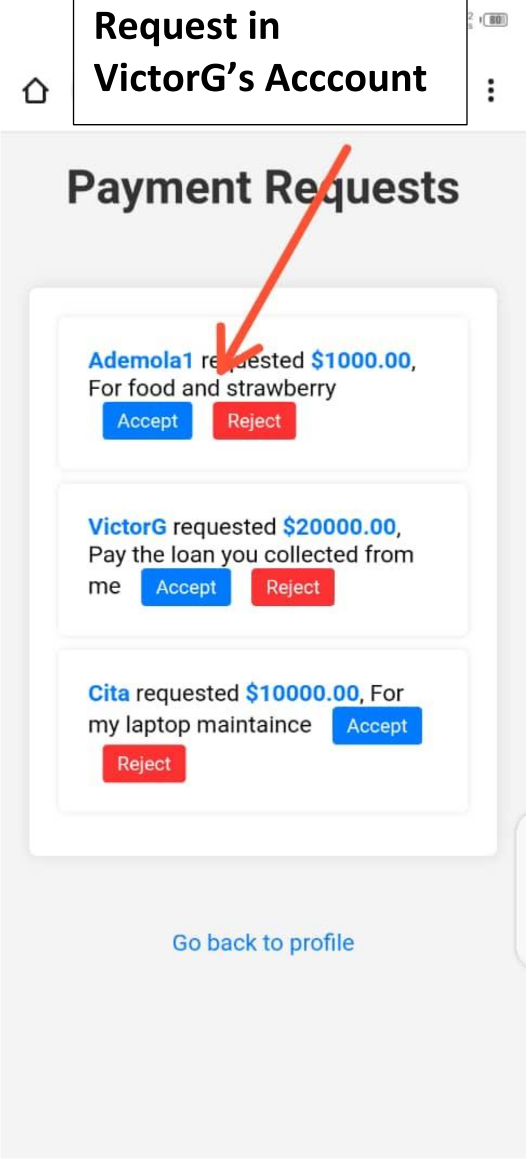
Create Payment Request

Amount: 1000

Description: For food and strawberry

Recipient: VictorG

15. Payment Request in VictorG's Acccount



Payment Requests

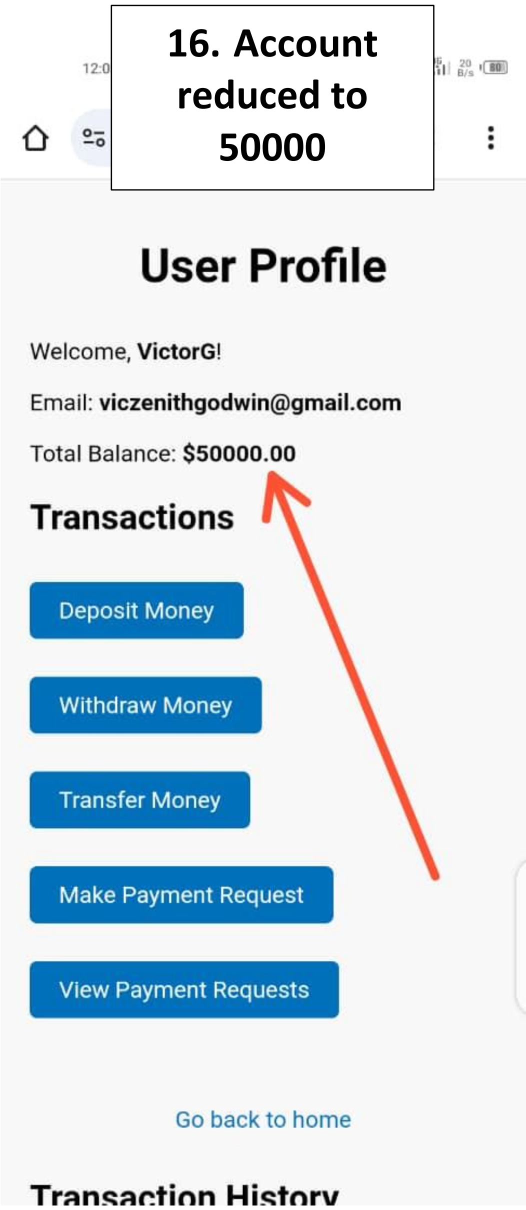
Ademola1 requested $1000.00, For food and strawberry

VictorG requested $20000.00, Pay the loan you collected from me

Cita requested $10000.00, For my laptop maintaince

Go back to profile

16. Account reduced to 50000



User Profile

Welcome, **VictorG**!

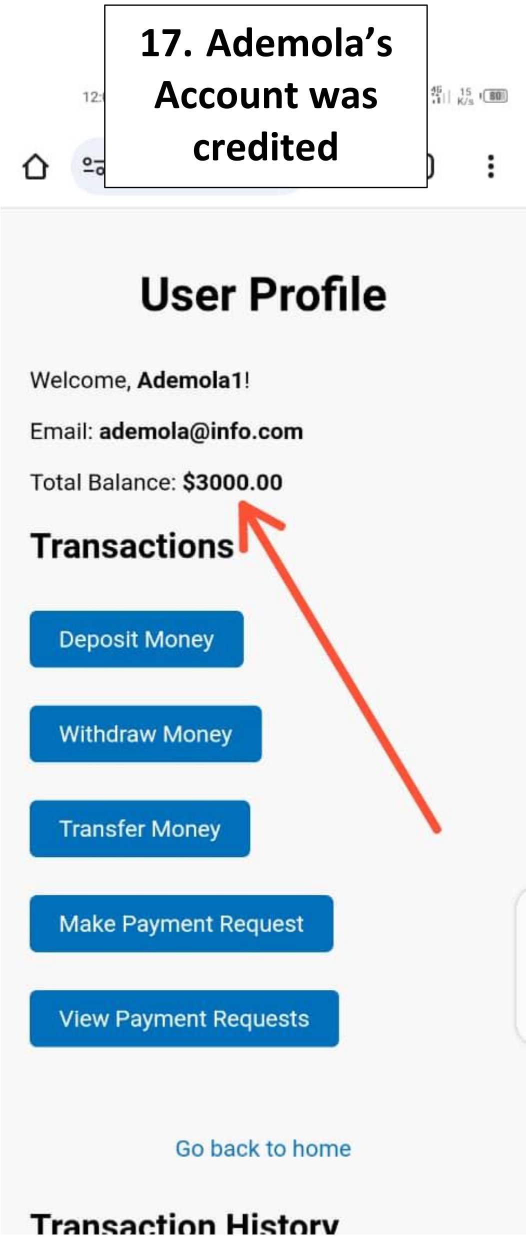
Email: **viczenithgodwin@gmail.com**

Total Balance: **$50000.00**

Transactions

Transaction History

17. Ademola’s Account was credited



User Profile

Welcome, **Ademola1**!

Email: **ademola@info.com**

Total Balance: **$3000.00**

Transactions

Transaction History