

Khalad mhamud omi
Daffodil smart city
Asulia dhada
Khalad15-4606@diu.edu.bd
0130597380
3-12-24

Monir Uddin
Dream Fill Foundation
48/AB, Bitul Khair Bhadan,
Purana Paltan, Dhaka-1000

Dear Monir Uddin,

Re: Proposal for Functional Banking Web Application

At Khalad Mhamud Omi , we understand that developing a client-oriented solution requires a combination of technical expertise and clear communication. We are committed to delivering a unique, innovative, and cost-effective solution tailored to meet your specific needs. Our goal is to provide a reliable and secure banking system that aligns with your requirements, delivered on time and within budget.

Based on the requirements you have shared, we propose the Functional Banking Web Application. This system will allow users to manage deposits, withdrawals, balance tracking, and provide secure access for both regular users and administrators. The features and functionalities are outlined in the attached proposal, which includes all the details regarding the development process.

Please review the enclosed proposal carefully, and if it satisfies your expectations, kindly sign and confirm the agreement.

We look forward to your positive response.

Yours truly,
Khalad mhamud omi
Functional Banking Web Application

1. Project Overview

The **Functional Banking Web Application** is designed to simulate core banking operations such as deposits, withdrawals, and balance management. The system ensures secure user authentication and provides a user-friendly interface for managing transactions. Key features include real-time balance tracking, a detailed transaction history, and an administrative panel for managing user accounts and system settings.

This web application uses modern technologies, including PHP for backend development, MySQL for database management, and Tailwind CSS with DaisyUI for a responsive and aesthetically pleasing frontend. The goal is to provide a seamless and efficient banking experience, emphasizing security, performance, and usability.

2. Obstacles

- Requirement Clarity: Ensuring all functional details are well-defined.
- Security: Implementing secure authentication and preventing vulnerabilities.
- Database Synchronization: Maintaining consistent, real-time transaction data.
- Validation: Handling errors and ensuring accurate inputs.
- UI Responsiveness: Providing a seamless experience across devices.
- Data Integrity: Ensuring reliable data storage and backups.

3. User Features in Details:

1. User Authentication:

Registration:

Users can create an account by providing essential details such as email, password, and basic personal information. This ensures that each user has a unique identity within the system.

Login/Logout:

Secure login functionality allows users to access their accounts using their registered email and password. Sessions are managed securely to prevent unauthorized access.

Password Recovery:

Users can recover their password through a secure process involving email verification or security questions.

2. User Dashboard:

Upon successful login, users are directed to a personalized dashboard that provides a comprehensive overview of their account. The dashboard displays critical information, including the current balance, total deposits, and total withdrawals.

Recent Transactions:

Users can view a summary of recent transactions, helping them keep track of their financial activities.

3. Deposit Functionality:

Users can input an amount they wish to deposit into their account. The system validates the input to ensure it is a positive number and processes the transaction securely.

Confirmation and Balance Update:

After a successful deposit, the system updates the user's balance in real-time and provides a confirmation message. An entry is also added to the transaction history for record-keeping.

4. Withdrawal Functionality:

Users can enter an amount they wish to withdraw from their account. The system checks if the withdrawal amount is available in the current balance.

Validation:

If sufficient funds are available, the withdrawal is processed, and the balance is updated. If not, an error message is displayed, preventing the transaction.

Transaction Records:

Each withdrawal is recorded in the transaction history, and users can review details such as the date, amount, and status of the withdrawal.

5. Transaction History:

Users have access to a detailed transaction history that includes both deposits and withdrawals. Each entry shows the date, type, amount, and status of the transaction.

Filtering Options:

Users can filter transactions by type (deposit or withdrawal) and date range, making it easier to track specific activities.

6 . Balance Inquiry:

Users can check their current balance at any time through the dashboard or a dedicated

balance modal. This feature provides an up-to-date view of their financial status, ensuring transparency and ease of access.

Balance Modal:

Clicking on the balance button triggers a modal window displaying the current balance, offering a quick and convenient way to view this information without navigating away from the current page.

7.Admin Features:

User Management:

Admins have the authority to create, edit, or delete user accounts. They can assign roles and manage access permissions, ensuring that only authorized users can perform specific actions.

Transaction Monitoring:

Admins can view a comprehensive list of all user transactions. This feature helps in monitoring account activities and ensuring the integrity of financial operations.

System Management:

The admin panel provides tools for managing system settings, viewing logs, and overseeing overall system performance.

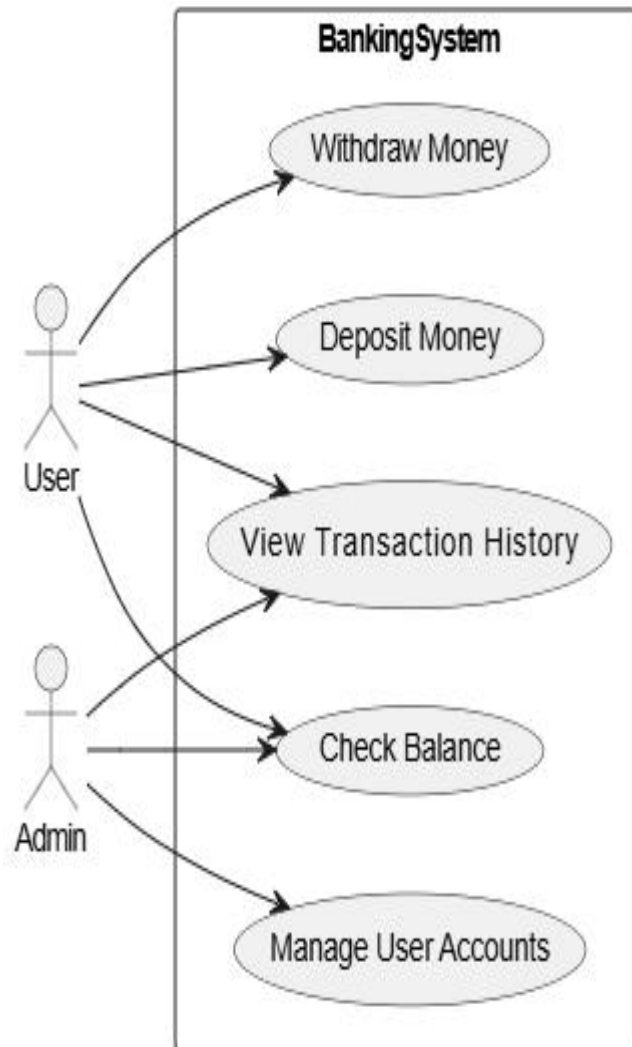
8. Responsive User Interface:

The application is designed with a responsive interface using Tailwind CSS and DaisyUI, ensuring a consistent and user-friendly experience across various devices, including desktops, tablets, and smartphones.

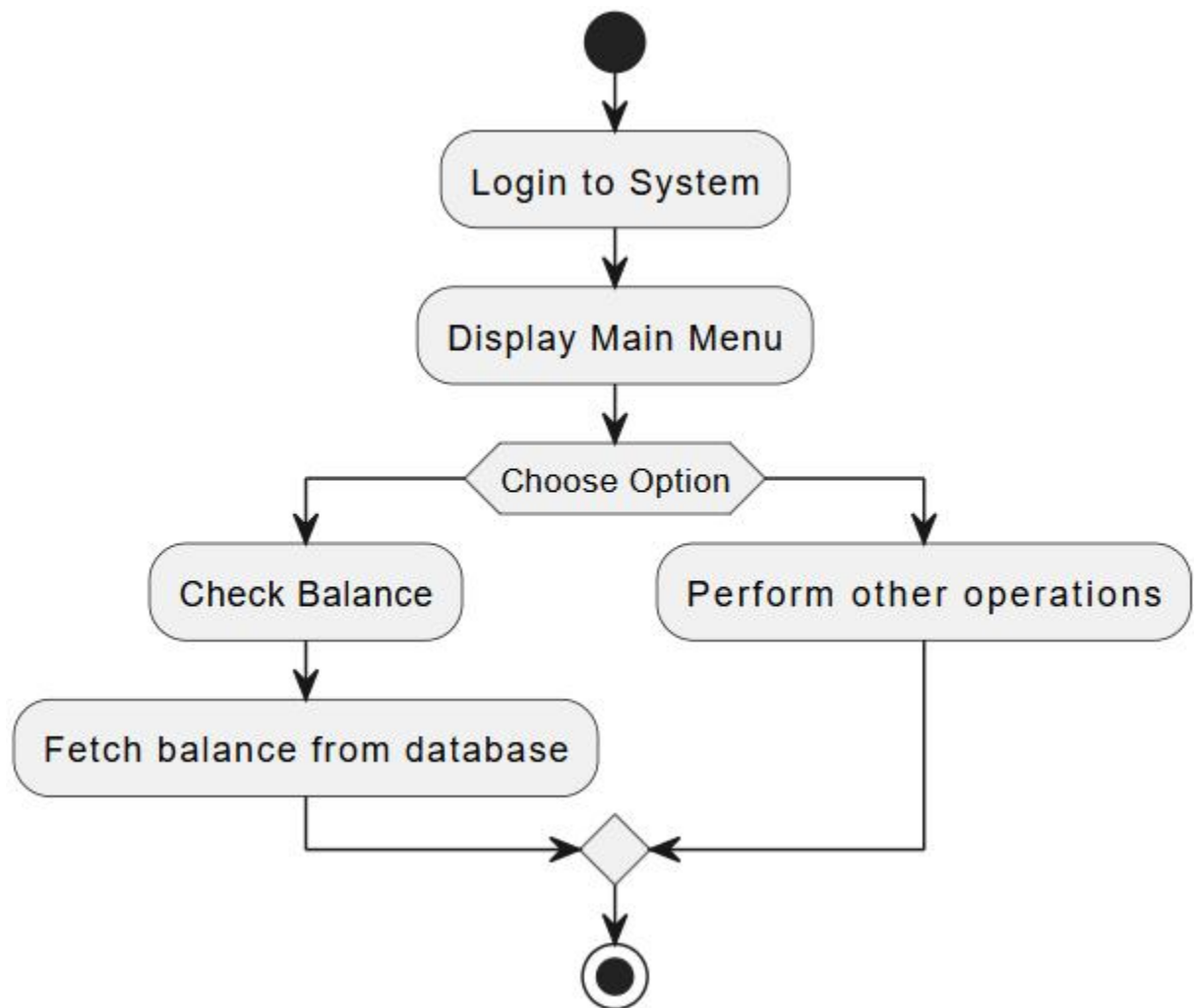
Accessibility Features:

Efforts are made to ensure that the interface is accessible to users with different needs, enhancing usability for a broader audience.

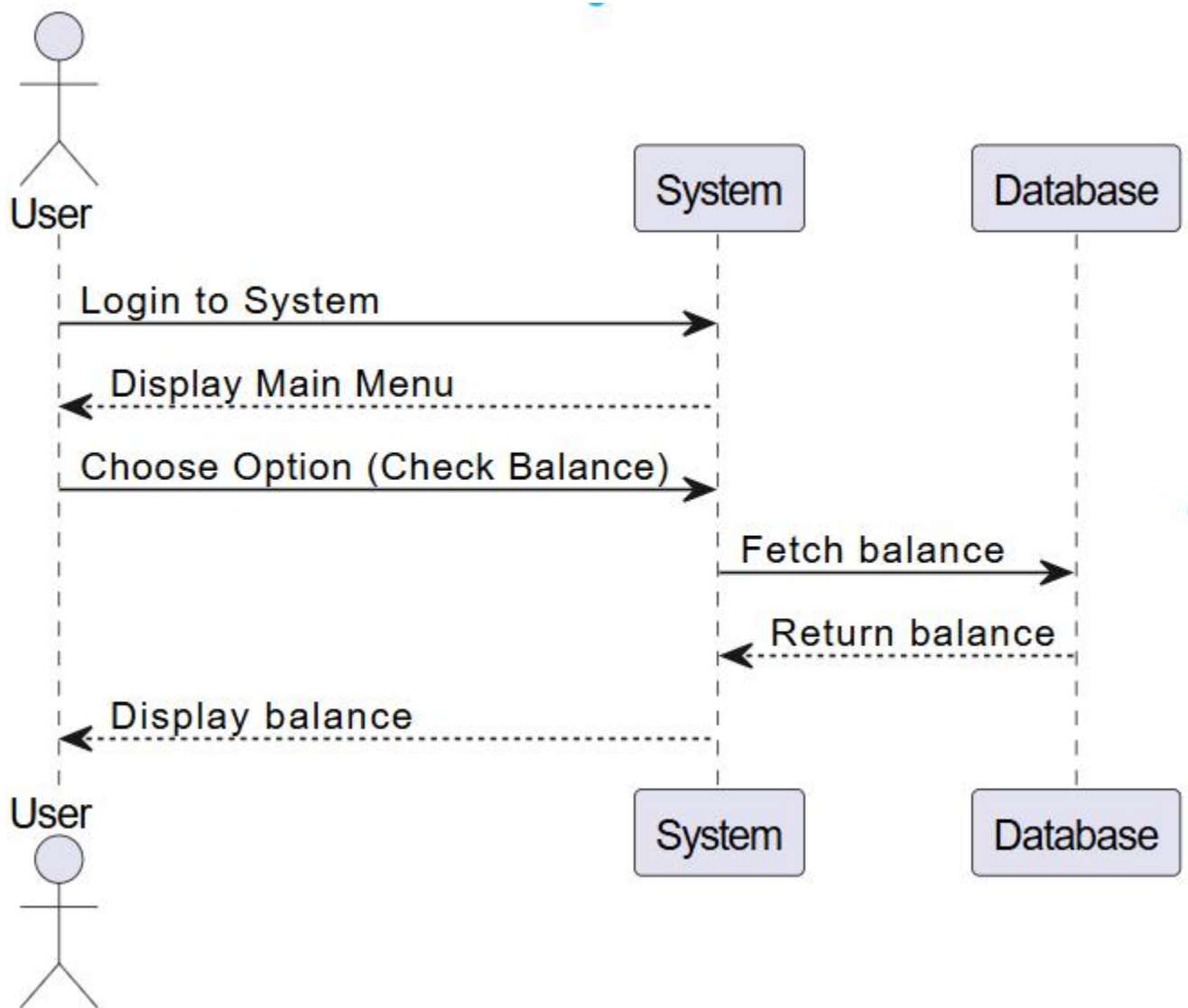
Use Case Diagram:



Activity Diagram:



Sequence Diagram:



4. Modules Details:

1. User Authentication Module:

Registration:

Users can create an account by providing details such as email, password, and personal information.

Login/Logout:

Secure login using email and password. Manages session handling to ensure secure access.

Password Recovery:

Allows users to reset their password through email verification or security questions.

2. Dashboard Module:

Overview:

Provides a summary of the user's account, including current balance, total deposits, and total withdrawals.

Recent Transactions:

Displays a list of the most recent transactions, offering a quick view of account activity.

3. Deposit Module:

Deposit Form:

Users input the deposit amount, which is validated to ensure it's a positive number.

Transaction Processing:

Updates the user's balance and records the deposit in the transaction history.

Confirmation:

Provides feedback to the user with a confirmation message upon a successful deposit.

4. Withdrawal Module:

Withdrawal Form:

Users enter the amount they wish to withdraw.

Validation and Processing:

Checks if the user has sufficient funds before processing the withdrawal. Prevents the transaction if funds are insufficient and displays an error message.

Transaction Logging:

Records each withdrawal in the transaction history for future reference.

5• Transaction History Module:

• Transaction Records:

Displays a detailed history of all transactions (deposits and withdrawals), including dates, amounts, and transaction types.

Filtering and Sorting:

Users can filter transactions by type (deposit or withdrawal) and date range to locate specific entries quickly.

6• Balance Management Module:

• Balance Inquiry:

Users can view their current balance on the dashboard or through a balance modal.

Real-Time Updates:

Ensures that balance calculations are updated in real-time after each deposit or withdrawal.

7• Admin Module:

• User Management:

Admins can create, edit, or delete user accounts and manage roles.

Transaction Monitoring:

Provides an overview of all transactions, enabling admins to monitor account activities.

System Configuration:

Allows admins to manage system settings, view logs, and ensure overall performance.

8• Frontend Interface Module:

• Responsive Design:

Ensures a consistent and user-friendly interface across various devices using Tailwind CSS and DaisyUI.

Interactive Elements:

Modals, buttons, and forms are designed for easy navigation and enhanced user experience.

5. Technology:

Frontend: HTML5, CSS3, Tailwind CSS, DaisyUI, JavaScript.

Backend: PHP for server-side logic.

Database: MySQL for data storage.

Framework: Custom PHP-based structure, likely following MVC architecture.

Security: Input validation, encryption, and prepared statements for SQL injection prevention.

Version Control: Git for code management and collaboration

6. Application

The Functional Banking Web Application is a platform designed to handle basic banking functions such as deposits, withdrawals, and balance management. It allows users to securely log in, manage their finances, and track transactions. The application features:

User Authentication: Secure registration, login, and password recovery.

Transaction Management: Users can deposit, withdraw, and view transaction history.

Balance Management: Real-time balance updates after each transaction.

Admin Panel: Admins can manage users, monitor transactions, and configure system settings.

Responsive UI: The application is designed with a user-friendly interface using Tailwind CSS and DaisyUI.

7. Milestones and Reporting

Milestone	Tasks	Reporting	Timeframe
Analysis	Requirement gathering, understanding client needs.	Report on requirement analysis and approval.	7 days
Requirements Collection & Sign-off	Collect and finalize all functional and non-functional requirements.	Deliver signed project requirements.	10 days
Development	Develop core banking features (deposit, withdrawal, balance management).	Regular development progress updates.	60 days
Testing	Perform unit testing, integration testing, and user acceptance testing (UAT).	Detailed testing reports and issues.	20 days
Deployment	Deploy the application on the server and ensure it is running smoothly.	Deployment report and final system check.	5 days
Delivery	Final project delivery and client approval.	Final project delivery report.	5 days

8. Deployment

Development Setup: Develop and test the application in a local environment.

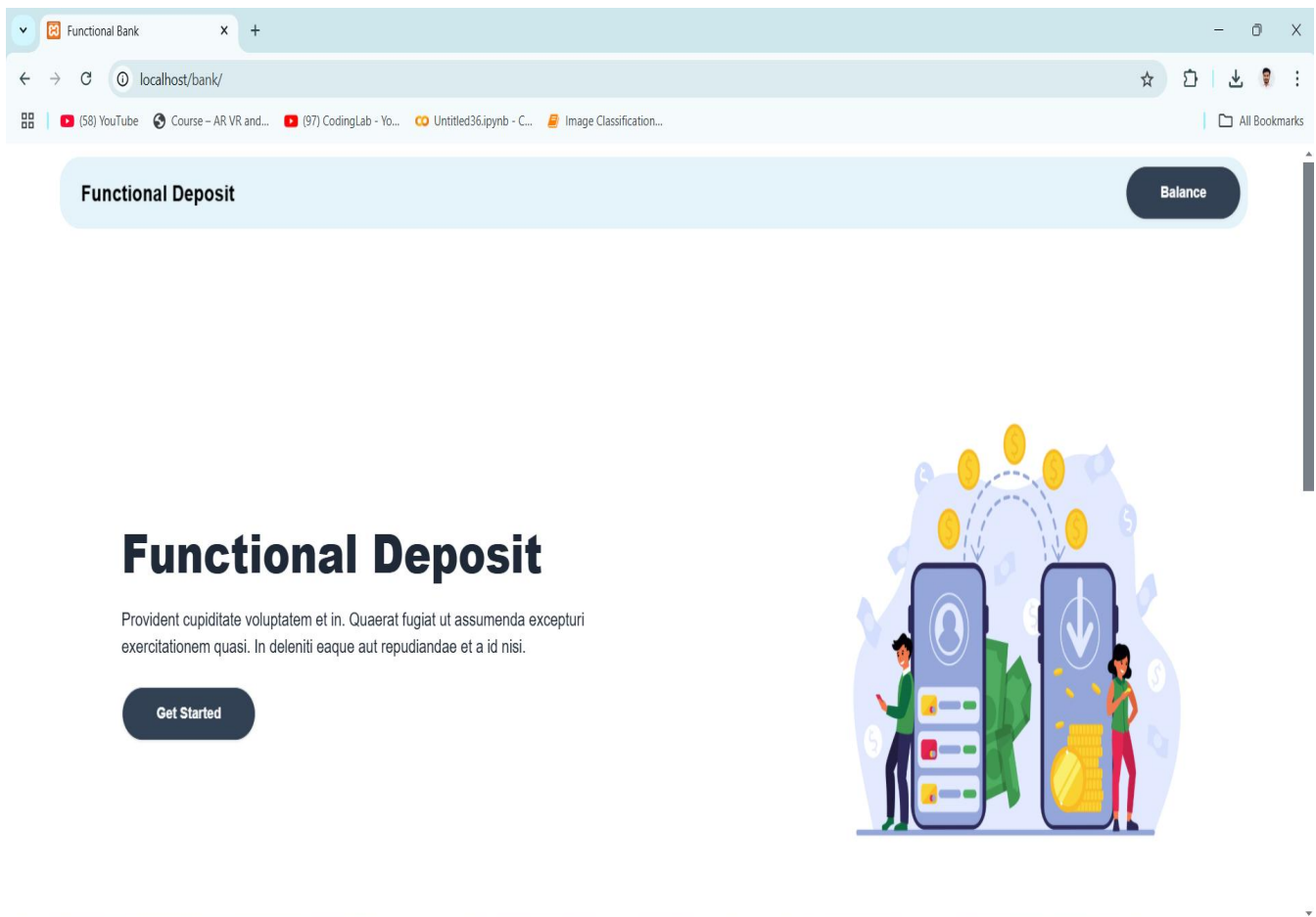
Server Configuration: Host on a cloud or local server (e.g., AWS, DigitalOcean) with PHP and MySQL support.

Database Setup: Configure MySQL database for user data and transactions.

Deployment Steps: Upload code to the server, set up environment variables, configure domain, and install SSL for HTTPS.

Post-Deployment Checks: Final testing to ensure smooth operation and performance.

Monitoring & Maintenance: Set up monitoring tools and perform regular updates as needed.



Main Game

Deposit
\$850.00

Withdraw
\$123.00

Balance
\$102,527.00

Your Deposit

Deposit Amount

Deposit

Your Withdraw

Withdraw Amount

Withdraw



SOCIAL

Your Deposit

Deposit Amount

Deposit

Your Withdraw

Withdraw Amount

Withdraw

9. Testing

Unit Testing: Test individual functions (deposit, withdrawal, balance) for correct operation.

Integration Testing: Ensure seamless interaction between frontend, backend, and database.

User Acceptance Testing (UAT): Validate system usability with real users.

Security Testing: Check for vulnerabilities like SQL injection and ensure SSL encryption.

Performance Testing: Ensure the app performs well under load.

Bug Fixing: Identify and fix any issues found during testing.

10. Support

24/7 Support: Available via email, phone, or live chat.

Bug Fixes & Updates: Ongoing fixes, performance improvements, and security patches.

User Documentation: Guides and FAQs for troubleshooting.

Technical Support: Assistance with server and backend issues.

Feature Enhancements: Regular updates based on user feedback.

11. Pricing

The total fee for developing and delivering the Functional Banking Web Application is 70,000 Taka. This includes:

Initial Payment: 60,000 Taka, due upon signing the agreement.

Remaining Payment: 10,000 Taka, due within 90 days of product delivery.

This pricing covers development, testing, deployment, and initial support, with additional charges for any new features or modifications requested after the initial deliver.

12. **Payment Terms**

10% (Initial Payment): Paid upon acceptance of the proposal.

40% (Second Payment): Paid upon signing the application development agreement.

25% (Third Payment): Paid at 70% completion of the application (demonstration phase).

25% (Final Payment): Paid upon the completion and delivery of the application.

13. **Responsibility:**

- Responsible for the full development of the Functional Banking Web Application including planning, coding, testing, deployment, and maintenance.
- Handling authentication, transaction management, balance tracking, and security.
- Ensuring the application is functional, secure, and delivered on time.
- Providing documentation and support after deployment.

14. **Contact Us :**

Khalad mhamud omi

Email: khalad15-4606@diu.edu.bd

Phone: 01305097380

Additional attachment.

Donate money:

1. Donate to the general fund
2. Donate to specific program
 - i) Helps for road accident
 - a) Daily safe
 - b) Monthly safe
 - c) Yearly safe
 - ii) Lessening the greenhouse effect
 - iii) Free legal aid
 - iv) Charitable program
 - a) Establish and maintain education
 - b) Street children
 - c) Orphan child
 - d) Training for producing human resource
 - IT
 - Nursing
 - Construction
 - Home serving
 - Hotel management
 - Agro based training

3.Input Field:

When the user clicks on any of the above donation options, an input field labeled "Amount" appears for users to enter the donation amount.

- A submit button is provided for users to complete their donation.

