PeerPay: A Blockchain-Based Transaction Visualization System

Phase 1 Report: Project Proposal

Harsh Dayal
Registration Number: 22BCE10564

Course: IBM blockchain
Institution: VIT

April 19, 2025

Contents

1 Introduction

This document outlines my proposal for the PeerPay project, undertaken as part of the IBM blockchain-course. The core goal of this project is to develop a functional web application that serves as an educational tool to demonstrate fundamental blockchain concepts. Specifically, I aim to build a system that visually represents a chain of blocks containing simple peer-to-peer transaction data, allowing users to observe the chain and add new transactions. This Phase 1 report details the problem I intend to address, my proposed solution, the objectives I aim to achieve, the technologies I plan to utilize, and the expected outcomes.

2 Problem Statement

While sophisticated blockchain platforms exist, understanding their core mechanics – such as hashing, block creation, and cryptographic linking – can be challenging for learners. Many existing examples are either too theoretical or involve complex consensus mechanisms and smart contracts that obscure the basic principles. Furthermore, traditional centralized systems for simple record-keeping often lack transparency and inherent immutability. Therefore, I identified a need for a simplified, hands-on application that:

- Clearly visualizes the structure and growth of a basic blockchain.
- Focuses on fundamental concepts like hashing and block chaining.
- Provides a practical context (peer-to-peer transactions) for learning.
- Demonstrates how a modern web application can interact with a simple blockchain backend.

3 Proposed Solution: PeerPay

To address the identified need, I propose building **PeerPay**, a full-stack web application. PeerPay will consist of:

1. **A Node.js Backend:** This server-side application will manage a simplified blockchain structure. It will handle the creation of new blocks containing transaction data, calculate cryptographic hashes (SHA-256) for each block, and ensure blocks are correctly linked via their ‘previousHash‘ field. The blockchain state will be persisted using MongoDB. It will expose a RESTful API for the frontend.
2. **A React Frontend:** This client-side application, built using React and Vite, will provide the user interface. Users will be able to view the sequence of existing blocks, inspect the data within each block (index, timestamp, transaction details, hashes), and submit new transactions (sender, recipient, amount) via a form.

The primary function of PeerPay is not to be a secure, production-ready payment system, but rather a dynamic educational tool for visualizing and interacting with core blockchain mechanics.

4 Core Objectives

My main objectives for completing Phase 1 and moving into implementation are:

- To design and implement a basic ‘Block‘ data structure in JavaScript/Node.js, including index, timestamp, transaction data, hash, and previous hash.
- To develop the backend logic for adding new blocks to the chain, ensuring correct hash calculation (SHA-256) and linkage.

- To set up a MongoDB database and integrate it with the backend using Mongoose for persistent storage of the blockchain.
- To build a RESTful API using Express.js with endpoints for retrieving the current blockchain (‘GET /blocks‘) and adding new transactions/blocks (‘POST /mine‘).
- To construct a responsive frontend using React and Chakra UI that displays the blockchain data fetched from the API.
- To create a user-friendly form using React Hook Form and Zod for submitting new transaction data, including input validation.
- To implement data fetching on the frontend using TanStack Query, enabling automatic updates (polling) to reflect new blocks added to the chain.
- To provide clear visual feedback to the user regarding the status of their transaction submissions (e.g., success or failure toasts).

5 Key Features & Functionality

Based on the objectives, the PeerPay application will offer the following key features:

- **Blockchain Visualization:** Displaying blocks sequentially, showing index, timestamp, transaction data (sender, recipient, amount), hash, and previous hash for each.
- **Transaction Submission:** A validated form for users to input sender, recipient, and amount details.
- **Block Addition:** Successful transaction submissions trigger the backend to create and add a new block to the persistent chain.
- **Immutability Demonstration:** While not strictly immutable like decentralized blockchains, the cryptographic linking (hash and previousHash) will demonstrate the principle – tampering with data in the database would invalidate subsequent hashes (verifiable via a potential validation endpoint).
- **Real-time Feeling Updates:** The frontend will periodically poll the backend to display new blocks automatically, simulating a live chain.
- **Modern User Interface:** A clean, responsive interface built with Chakra UI, including features like light/dark mode.

6 Technological Approach

I intend to utilize a modern, JavaScript-centric technology stack for this project:

- **Frontend:**
  - *Framework/Library:* React 18+
  - *Build Tool:* Vite
  - *UI Components:* Chakra UI
  - *Data Fetching/State Management:* TanStack Query (React Query) v5
  - *Forms:* React Hook Form & Zod (for validation)
  - *Animation:* Framer Motion

  - *Icons:* Lucide React

- **Backend:**
  - *Runtime:* Node.js
  - *Framework:* Express.js
  - *Hashing:* Node.js built-in ‘crypto‘ module (SHA-256)

- **Database:**
  - *Type:* MongoDB (NoSQL)
  - *ODM:* Mongoose

- **Development Environment:**
  - *Package Manager:* npm
  - *Version Control:* Git / GitHub

This stack was chosen for its strong ecosystem, my familiarity with JavaScript, and the suitability of these tools for building modern, interactive web applications and backend APIs efficiently.

7 Innovation and Educational Value

While the concept of a transaction ledger is not new, the innovation of PeerPay lies in its specific focus and implementation:

- **Educational Clarity:** It prioritizes demonstrating fundamental blockchain mechanics (hashing, chaining) without the added complexity of distributed consensus, smart contracts, or tokenomics found in full-fledged cryptocurrencies.
- **Custom Implementation:** By building the core block-linking logic from scratch in JavaScript, I gain a deeper understanding compared to using pre-built blockchain platforms.
- **Modern Full-Stack Integration:** The project showcases how a contemporary frontend application interacts with a custom blockchain-inspired backend, providing a relevant full-stack development experience.
- **Visual Learning:** The emphasis on visualizing the blocks and their connections aims to make abstract concepts more tangible.

8 Expected Outcomes

Upon successful completion of the implementation phase based on this proposal, I expect to deliver:

- A fully functional PeerPay web application hosted locally.
- A backend API service capable of managing and persisting the blockchain state in MongoDB.
- A frontend user interface allowing users to view blocks and submit new transactions.
- A working demonstration of block creation, SHA-256 hashing, and chaining within the application context.
- A codebase adhering to modern JavaScript and React development practices.
- A foundation upon which potential future extensions could be built.

9 Potential Future Extensions

While the core focus is on delivering the functionality outlined above, I recognize potential avenues for future development beyond the scope of this initial project:

- Implementing a '/validate' API endpoint to explicitly check the integrity of the entire chain stored in the database.
- Adding user authentication to associate transactions with specific users.
- Enhancing the block explorer with search and filtering capabilities.
- Simulating a basic Proof-of-Work mechanism for block addition.
- Exploring P2P concepts for sharing the chain between multiple instances (significantly more complex).

10 Conclusion

The PeerPay project presents a valuable opportunity for me to apply and deepen my understanding of both blockchain fundamentals and full-stack web development. By focusing on creating a clear, interactive, and educational tool for visualizing simple transactions on a custom blockchain structure, I aim to meet the project requirements effectively. I am confident that the proposed solution, objectives, and technology stack provide a solid plan for successful implementation.