HederaConnect - Project Documentation

1. Project Overview

Objective

HederaConnect is a decentralized knowledge-sharing platform that aggregates educational content on Hedera from various social media platforms. It allows users to access and share articles while rewarding content creators for their contributions.

Key Features

- Aggregation of educational articles from social media.
- Decentralized storage using IPFS/Arweave.
- Engagement-based rewards using Hedera Token Service (HTS).
- Transparency in content verification via Hedera Consensus Service (HCS).
- User interactions (likes, shares, comments) to boost content engagement.

2. System Architecture

Tech Stack

- **Backend:** Node.js (Express) or Rust (Actix Web)
- **Database:** PostgreSQL or MongoDB
- **Storage:** IPFS/Arweave for content storage
- **Blockchain:** Hedera Hashgraph (HTS & HCS)
- **Authentication:** Hedera DID or Firebase Auth

Microservices

1. **Content Aggregation Service** - Fetches articles from Twitter, LinkedIn, Medium.
2. **User Interaction Service** - Manages likes, shares, and engagement.
3. **Content Verification Service** - Community/admins verify content.
4. **Reward Distribution Service** - Calculates and distributes rewards.
5. **Storage Service** - Manages article storage and metadata.

3. API Endpoints

Method	Endpoint	Description

POST	`/articles/submit`	Submit a new article
GET	`/articles`	Fetch all verified articles
POST	`/articles/verify/:id`	Admin verification
POST	`/articles/like/:id`	Like an article
POST	`/articles/share/:id`	Share an article
POST	`/rewards/distribute/:id`	Distribute rewards

4. Smart Contract Integration

- **HTS (Hedera Token Service):** Distributes rewards in HBAR or custom tokens.
- **HCS (Hedera Consensus Service):** Logs article verification & engagement.
- **Mirror Node:** Fetches real-time blockchain data.

5. Reward Mechanism

- Each like: +1 point
- Each share: +2 points
- Rewards distributed based on total engagement points.

6. Setup & Deployment

Backend Setup

1. Clone the repository:
2. Install dependencies:
3. Set up environment variables (`.env` file):
4. Start the backend server:

Frontend Setup

1. Navigate to the frontend directory:
2. Start the React frontend:

7. Future Enhancements

- Implement AI-based content filtering.
- Introduce community-driven article moderation.
- Add NFT-based proof-of-authorship for articles.

8. Contributors

- **Eugene Karewa** - Project Lead & Developer
- Destinne - Project Lead & Developer