Project Ideas

- TODO: Markdown based record log
- TODO: Integrated note taking
- TODO: Supply Chain Management System
- TODO: Asset Tokenization Platform
- TODO: Decentralized Voting System
- TODO: Trade Finance Platform
- TODO: Healthcare Data Exchange
- TODO: Web and Android based TOTP generator
- TODO: RESTful API with Authentication and Authorization
- TODO: Task Scheduling and Background Jobs with a Job Queue
- TODO: Real-time Chat Application with WebSockets
- TODO: Microservices Architecture with Communication Protocols
- TODO: Scalable and Fault-Tolerant Distributed Systems
- TODO: Building a Content Delivery Network (CDN)
- TODO: Implementing a Caching Layer for Performance Optimization
- TODO: Designing and Optimizing Relational Database Schemas
- TODO: Building a Web Scraping and Data Processing Service
- TODO: Creating an Analytics Dashboard with Data Visualization
- TODO: Clone: StreamYard, Google Docs, Google Meet, Google Drive, LeetCode

Each project idea will allow you to explore different aspects of backend development, including API design, authentication, data processing, scalability, and performance optimization. Let's delve into the details of each project:
Note: You can build any of these projects in different framework or language.

- Markdown based record log

- Integrated note taking

- Supply Chain Management System: Build a decentralized supply chain management system using Hyperledger Fabric. This project will involve creating smart contracts to track and verify the movement of goods, managing identities and permissions, and implementing a user-friendly interface for participants to interact with the system.

- Asset Tokenization Platform: Develop an asset tokenization platform using Hyperledger Fabric. This project will involve creating smart contracts to represent and manage digital assets, implementing a token issuance and transfer mechanism, and building a secure and scalable infrastructure for asset tokenization.

- Decentralized Voting System: Create a decentralized voting system using Hyperledger Fabric. This project will involve designing smart contracts to handle the voting process, implementing identity management and authentication mechanisms, and building a secure and transparent platform for conducting elections.

- Trade Finance Platform: Build a trade finance platform using Hyperledger Fabric. This project will involve creating smart contracts to facilitate trade finance transactions, implementing a decentralized ledger for tracking and verifying trade documents, and integrating with external systems for payment processing and risk assessment.
- Healthcare Data Exchange: Develop a healthcare data exchange platform using Hyperledger Fabric. This project will involve designing smart contracts to handle the secure exchange of medical records, implementing privacy and consent management mechanisms, and building a scalable infrastructure for healthcare data interoperability.
- RESTful API with Authentication and Authorization: Develop a RESTful API using frameworks like Express or Django, incorporating authentication and authorization mechanisms to secure the endpoints.
- Task Scheduling and Background Jobs with a Job Queue: Implement a task scheduling system using a job queue like Bull or RabbitMQ, allowing asynchronous execution of tasks and background job processing.
- Real-time Chat Application with WebSockets: Build a real-time chat application using WebSockets and frameworks like Socket.io or Django Channels, enabling instant messaging between users.
- Microservices Architecture with Communication Protocols: Design and implement a microservices architecture using tools like Docker and Kubernetes, and establish communication between services using REST APIs or message brokers.
- Scalable and Fault-Tolerant Distributed Systems: Build a distributed system that can handle high traffic and ensure fault tolerance by leveraging technologies like Apache Kafka, Apache Cassandra, or Redis.
- Building a Content Delivery Network (CDN): Create a CDN using technologies like Nginx or Varnish Cache to efficiently deliver static assets and improve the performance of web applications.
- Implementing a Caching Layer for Performance Optimization: Integrate caching mechanisms like Redis or Memcached into your backend system to cache frequently accessed data and enhance application performance.
- Designing and Optimizing Relational Database Schemas: Develop a database schema for a complex domain, optimizing queries, relationships, and indexing strategies to ensure efficient data retrieval and storage.
- Building a Web Scraping and Data Processing Service: Create a service that collects data from web sources using tools like Scrapy or Puppeteer, and process and store the data in a structured format.
- Creating an Analytics Dashboard with Data Visualization: Build an analytics dashboard using frameworks like React or Angular, integrating with data visualization libraries like D3.js or Chart.js to provide insightful data representations.
- Clone: Repo link for requirements and design