

# MOHAMMED MINHAJ

+91 7306279647 | [minhajmuhamad@gmail.com](mailto:minhajmuhamad@gmail.com)

 [LinkedIn](#) |  [GitHub](#) |  [Leetcode](#)

## SUMMARY

- Self driven software engineer with a strong foundation in **Python, JavaScript, and Blockchain** technologies.
- **Python Developer** at Infox Technologies for **Two years**, gained deep expertise in backend frameworks, with a particular focus on **Django** and **RESTful APIs**.
- Explored **Blockchain development** as an independent developer, specializing in writing **Smart Contracts** and creating **Automated Trading Programs**.
- Exclusively designed and developed an **e-commerce** platform using Django and successfully deployed it on **AWS cloud**.

## SKILLS

- **Soft Skills:** Leadership, Proactivity, Problem-Solving, Adaptability, Self-Motivation
- **Programming Languages:** Python, JavaScript, Scilla, Solidity, SQL
- **Backend Technologies:** Django, FastAPI, Node.js, REST APIs
- **Frontend Technologies :** React, Next.js, HTML, CSS, JavaScript
- **Database Systems:** PostgreSQL, MySQL, MongoDB, Redis
- **DevOps & Version Control:** Docker, Git, GitHub, AWS EC2, Gunicorn, Nginx
- **Other Tools & Technologies:** Postman, Redis
- **Research Skills:** Algorithm Design, System Optimization

## EXPERIENCE

- **Self Employed - Full Stack and Blockchain Developer** *Sep 2022 - Present*  
Kerala
  - **Gulf Mobiles 2 (Refurbished iPhone Store)** []
    - \* **Tech Stack :** Python, Django, SQLite, Celery, Redis, AWS EC2, Gunicorn, Nginx, Razorpay
    - \* Developed Gulf Mobiles 2, a refurbished iPhone store web application using Python, Django, and SQLite, ensuring robust backend functionality and an intuitive user experience.
    - \* Implemented asynchronous processing with **Celery and Redis** to manage critical background tasks like inventory updates and order processing, significantly enhancing performance.
    - \* Integrated **Razorpay** to facilitate secure and efficient online transactions, boosting customer trust and streamlining the payment process.
    - \* Deployed and scaled the application on **AWS EC2 using Gunicorn and Nginx**, optimizing server performance and ensuring high availability in a dynamic production environment.
  - **Arbitrage Trading Bot in Zilliqa Blockchain V1**
    - \* **Tech Stack :** Javascript, Node.js, Next.js Zilliqa.js.
    - \* Developed an Arbitrage Trading Bot on the Zilliqa Blockchain leveraging JavaScript, Node.js, Next.js, and Zilliqa.js to identify and capitalize on arbitrage opportunities across decentralized exchanges.
    - \* Implemented **real-time transaction monitoring using WebSockets** to listen to blockchain transaction blocks and detect price discrepancies across trading pairs, optimizing trading strategies through advanced algorithms.
  - **Arbitrage Trading Bot in Zilliqa Blockchain V2** []
    - \* **Tech Stack :** Scilla, Javascript, Node.js, Next.js Zilliqa.js.
    - \* Engineered an enhanced Arbitrage Trading Bot on the Zilliqa Blockchain, utilizing Scilla, JavaScript, Node.js, Next.js, and Zilliqa.js to **address transaction failure** challenges inherent in decentralized exchanges.
    - \* Designed and implemented a robust smart contract capable of executing multiple trades within a single transaction. The contract **incorporated fail-safe mechanisms** that ensured the entire transaction would revert if one trade failed, effectively mitigating losses and boosting efficiency.

- \* Achieved remarkable profitability, reducing transaction failures and increasing operational success. Delivered more than a **10,000% profit margin**, showcasing the efficiency and precision of the enhanced arbitrage trading mechanism.
- **Token Smart Contract Creation** [🌐]
  - \* **Tech Stack** : Scilla, Javascript, Node.js Next.js, Zilliqa.js , Zilpay, ZRC-2.
  - \* Developed a smart contract for the Stellar Void Project on the Zilliqa blockchain, implementing a **deflationary token model** that automatically burns a portion of tokens with every transfer to reduce circulating supply.
  - \* Integrated a rewards mechanism that redistributes transaction fees to long-term holders, incentivizing loyalty and fostering sustainable community engagement.

#### • Infox Technologies - Python Developer

*October 2020 - Aug 2022  
Kochi, Kerala*

##### ◦ **Infox Core (Employee Management System)**

- \* **Tech Stack** : Python, Django, MySQL, Django Templates, Bootstrap, Celery, Redis
- \* Developed a Comprehensive Employee Management System using Django, streamlining employee data handling and task assignments.
- \* Reduced task assignment processing time by 40% through Celery optimization
- \* Improved dashboard load time from 8s to 1.2s via Redis caching

##### ◦ **FinSys - (ERP System)**

- \* **Tech Stack** : Python, Django, Javascript, React, MySQL, JWT , RESTful APIs
- \* Developed an Enterprise Resource Planning (ERP) System to manage employee data, inventory, sales, and accounting, leveraging Django for the backend, React for the frontend, RESTful APIs, and **JWT-based authentication** for secure user access. Improved operational efficiency and data accessibility across the organization
- \* Designed and implemented an Inventory Management System with **RESTful APIs** and JWT authentication to support real-time tracking of stock levels, product categorization etc. Automated stock updates upon order fulfillment, reducing inventory discrepancies and enhancing reliability.

## OTHER PROJECTS

- 
- **Realtime Messenger: (Realtime Messaging Application using websockets)** [🌐]
    - *Django, React, Psql, Redis, Django-channels, Daphne, Render*
    - Developed a real-time chat application using Django as the backend framework, React as frontend, with WebSockets for real-time communication, **Redis as the message broker**, and PostgreSQL as the database for storing messages and user data.
  - **PDF - Summarizer : (FastAPI microservices using NLP )** [🧠]
    - *FastAPI, Mongodb, PostgreSQL, Celery, Redis, HuggingFace Transformers*
    - A PDF summarization tool that uses NLP to summarize content and generate answer for questions from uploaded PDFs. Built using microservices with FastAPI, Celery, MongoDB, PostgreSQL, and React.
  - **Smart Web - Parser : (FastAPI microservices using NLP and Web scraping )** [🧠🧠]
    - *FastAPI, Sqlite, BeautifulSoup4, HuggingFace Transformers*
    - The Web Processor Service takes a URL as input, fetches and parses the webpage content using BeautifulSoup, and stores the extracted data into an SQLite database.
    - The Chat Service is a FastAPI microservice that enables users to ask questions about previously parsed webpage content. It uses HuggingFace Transformers for question answering and Sentence Transformers for embedding.

## EDUCATION

#### • Vimal Jyothi Engineering College

*June 2016 - Aug 2020  
Kannur, Kerala*

- Bachelors of Technology in Computer Science Engineering*
- GPA: 6.6/10

#### • St. Thomas HSS

*March 2016  
Kasargod, Kerala*

- Higher Secondary Education*
- Grade: 90.0%

#### • Archangels School

*March 2014*

- Secondary Education*
- GPA: 9.8/10

## ADDITIONAL INFORMATION

**Languages:** English (Fluent), Hindi (Intermediate), Malayalam (Native)