

# BASITH AHMED

+91 94009 35550 basithahmedddd@gmail.com

linkedin.com/basith-ahmed github.com/basith-ahmed basithahmed.me

## Technical Skills

**Languages:** HTML, CSS, JavaScript, TypeScript, Python, C, C++, Java, Go, SQL

**Frameworks/Libraries:** Next.js, React, Node.js, Express.js, TailwindCSS

**Databases:** MongoDB, PostgreSQL, Redis, ClickHouse

**Cloud/Platforms:** AWS, GCP, EC2, S3, ECS, ECR

**Tools & Environments:** Git, GitHub, Docker, Postman, Linux

## Projects

**Deploy-It** | AWS ECS, Fargate, ECR, S3, Docker, Kafka, ClickHouse, PostgreSQL, Next.js, Express [Live](#) — [Source](#)

- Made a containerized build system using **AWS ECS Fargate** to run on-demand **Docker** containers from **ECR** that clone and build GitHub repositories, and upload build artifacts to **S3**.
- Streamed build logs from build containers to **Kafka**, consumed by the API server and stored in **ClickHouse** for high-speed retrieval, then exposed via **REST APIs** for real time access.
- Developed a **reverse proxy** server to route requests to **S3**-hosted builds based on subdomains, while capturing web analytics and storing request logs in **PostgreSQL** for monitoring.

**Web-Builder** | Gemini, WebContainer, Next.js, React, Node.js, Express [Live](#) — [Source](#)

- Offloaded the build processes to the client using the **WebContainer API**, enabling live terminal logs, in-browser compilation, and reducing server load, infrastructure costs, and improving scalability.
- Built a frontend pipeline with **Next.js** and **React**, including a custom parser to process LLM responses and inject structured code into **WebContainer** for real-time, in-browser compilation and preview.
- Developed a backend system with **Node.js** and **Express** that uses **Gemini** to generate full-stack projects from user prompts, returning structured code based on a predefined schema.

**Git-Ask** | Langchain, Gemini, tRPC, PostgreSQL, GitHub API, Supabase, Next.js, TypeScript [Live](#) — [Source](#)

- Built a **RAG system** that fetches **GitHub** repo content, generates embeddings via **Langchain**, and stores them in **PostgreSQL** using **pgvector** on **Supabase** for semantic code search and question answering.
- Developed a fully type safe API layer using **tRPC** and **TypeScript** to manage embeddings, query processing, and user interactions with high reliability and maintainability.
- Integrated **Gemini LLM** for natural language query resolution, enabling **semantic search**, **contextual understanding**, and **automated codebase querying** directly from GitHub repositories.

**VPrint** | Firebase Cloud Storage, Firestore, Razorpay, Next.js, React, Node.js, Express [Live](#) — [Source](#)

- Designed and implemented the system architecture for a dual role web application, featuring separate user and admin interfaces with **Firebase Authentication** and secure **role-based access control**.
- Integrated real time features including live queue visualization, dynamic price calculation, shop availability status, and **payment workflows** using the **Razorpay** payment gateway.
- Developed a comprehensive **admin dashboard** with **role-specific access** to manage orders, monitor queues, update statuses in real time, and securely handle files using **Firebase Object Storage**.

## Education

**Vellore Institute of Technology**

Aug 2023 – May 2027

Bachelor of Technology in Computer Science

CGPA 8.57

**Relevant Coursework:** Object Oriented Programming, Database Management Systems, Data Structures and Algorithms, Operating Systems, Computer Networks, Theory of Computation, Computer Architecture and Organization, Machine Learning

## Certifications and Courses

**Modelling Software Systems using UML** | The Hong Kong University of Science and Technology [Certificate](#)

**AWS Fundamental Specialisation** | Amazon Web Services [Certificate](#)

**Software Engineering Experience Program** | JP Morgan Chase & Co [Certificate](#)

**Introduction to Trading, Machine Learning and GCP** | New York Institute of Finance [Certificate](#)