

# Muhammad Saim

Software Engineer

[✉️](mailto:saimrao849@gmail.com) saimrao849@gmail.com

[📞](tel:+923096854078) +923096854078

[📍](#) Islamabad, Pakitan

[LinkedIn](#)

[Github](#)

[Portfolio](#)

## Profile

I'm a Software Engineer with experience in building modern web applications using **React**, **Next.js**, **NestJS** and **AWS**. I focus on creating clean, scalable, and high-performing solutions from frontend to backend. I've worked with tools and libraries like **Tailwind**, **Ant Design**, **Material UI**, **Redux**, **Recoil**, **TanStack**, and **GraphQL** to build interactive and efficient user interfaces. On the backend, I use **NestJS** and **MySQL** to develop secure and scalable APIs with **JWT** authentication, **WebSocket** communication, and optimized database performance. I also have experience managing deployments and infrastructure using **AWS**, **Docker**, **Terraform**, and **CI/CD** pipelines through **GitHub Actions** and **CodePipeline**. I enjoy solving problems, improving performance, and collaborating with teams to deliver reliable, user-friendly applications. Always eager to learn new technologies and improve my skills to build better software.

## Professional Experience

### FutureNostics

Software Engineer

10/2024 – Present | Islamabad, Pakistan

**Construction Project Builder** – an enterprise construction management platform that unifies plan viewing, field workflows, and form automation across web and offline-capable experiences.

- Led the front-end architecture for an Nx-based React 18 **monorepo**, orchestrating **Auth0**, **Apollo GraphQL**, **Sentry**, and **RxDB** providers with lazy module loading to deliver a resilient, **offline-capable** field operations platform.
- Built a drag-and-drop form builder library (**Dnd Kit + MUI + Zustand**) that enforces layout rules, supports i18n, and is published as a reusable Nx package powering dynamic workflows across the suite.
- Implemented an **offline mutation queue** atop Apollo Link and RxDB with optimistic updates, conflict resolution, and **background sync**, ensuring uninterrupted jobsite data capture in low-connectivity environments.
- Engineered an advanced plan management UI with geospatial overlays, bulk task operations, undo stacks, and role-based access controls, backed by **Apollo cache hygiene** and service-worker broadcast updates.
- Hardened the front-end toolchain with **Nx task orchestration**, GraphQL **codegen**, **Husky hooks**, and **Playwright smoke suites**, enabling repeatable **CI/CD** and fast developer feedback loops.

### Company Official Website:

[www.futurenostics.com](http://www.futurenostics.com) ↗

- Developed and maintained the company's official website from scratch using **Next.js** with **TypeScript**, ensuring a responsive and high-performance experience.
- Implemented UI components and layouts using **ShadCN** for design consistency and usability.
- Integrated **Acernity UI** and **Magic UI** for dynamic animations, enhancing the user experience with smooth, interactive elements.
- Utilized **GraphQL** to fetch dynamic content, including articles, blogs, and other media, ensuring efficient and flexible data retrieval from **Strapi CMS**.
- Implemented API calls using **Server-Side Rendering (SSR)**, optimizing page load times, improving **SEO performance**, and ensuring dynamic content is always up-to-date.
- Delivered dynamic content with CMS-powered **Strapi** to streamline content management and enhance user experience.

### Real-time Secure Messaging System

#### Backend (NestJS, TypeORM, Socket.IO):

- **WebSocket** Gateway with custom namespaces
- JWT-based auth with **AES-GCM encryption**
- Rate limiting (60 msgs/min), 1MB message cap
- Group/direct chats, message receipts, file sharing
- Optimized DB with Many-to-Many relationships

#### Frontend (React, TypeScript, MUI):

- Real-time updates with custom hooks & Context API
- Responsive UI with group chat, read receipts, typing indicators
- File uploads with encryption & validation
- Optimistic updates, cached messages, lazy loading

#### Security & Performance:

- **End-to-end** encryption, secure token validation
- Role-based access control, **secure file handling**
- Efficient WebSocket & DB query management

**Highlights:** Built a secure, scalable, full-featured messaging platform with real-time communication, robust encryption, and advanced chat capabilities.

### BlockyFy

React Developer

06/2023 – 10/2024 | Islamabad, Pakistan

- Leveraged expertise in **React** to develop robust frontend applications, utilizing industry-standard practices and technologies.
- Implemented **Ant Design** framework to create visually appealing and responsive user interfaces, enhancing overall user experience.

- Integrated **GraphQL** for efficient data fetching and management, optimizing application performance and flexibility.
- Utilized external hooks such as **useRecoil** for **state management**, enhancing **code maintainability** and scalability.
- Demonstrated proficiency in **TypeScript** to ensure **type safety** and improve code quality throughout the development process.
- Contributed to the **technical documentation** and knowledge sharing within the team to ensure smooth project transitions and onboarding processes.

#### **Chatbot Web Application Development:**

- Developed a **chatbot web application** in **Django** that involves data storage in text files, facilitating efficient data retrieval using the **Langchain library**.

#### **Data Management:**

- Created mechanisms to store and manage **user-generated data** efficiently through text files while ensuring **data security** and **accessibility**.

#### **Natural Language Processing (NLP):**

- Utilized the **Langchain library** to implement advanced **NLP techniques**, enabling the chatbot to provide accurate and context-aware responses to user queries.

#### **Question-Answering System:**

- Established a robust question-answering system, allowing users to retrieve answers from stored data effectively.

#### **Admin Panel Development:**

- Designed and implemented a comprehensive **admin panel** using **TypeScript** and **Tailwind CSS**, ensuring a seamless integration with the backend services.
- Focused on creating an intuitive and responsive user interface to enhance the overall user experience.
- Implemented various features for **user management**, **analytics**, and reporting, providing administrators with efficient tools for managing operations.
- Ensured the application adhered to best practices in **security** and **performance optimization**.

#### **Crypto Wallet Admin Panel:**

- Created an **admin panel** for a **crypto wallet application** to monitor user activities and manage transactions, using **TypeScript**.
- Integrated **real-time data fetching**, updates, and **push notification services** to provide administrators with up-to-date information on user transactions and account activities.
- Implemented advanced **filtering** and **search capabilities**, enabling administrators to efficiently track and manage transactions.
- Ensured high standards of **security** and **data protection**, adhering to industry best practices and regulatory requirements.

#### **DevOps:**

##### **Infrastructure as Code with Terraform:**

- Designed and managed scalable and reliable cloud infrastructure using **Terraform**, implementing **IaC** practices to automate resource provisioning.
- Utilized **Terraform** for building **multi-tier architectures** on **AWS**, including networking, security, and storage components.
- Maintained remote backends for secure state management and team collaboration.

##### **Containerization with Docker and AWS ECS:**

- Developed **Docker containers** to streamline application deployment and improve portability across environments.
- Deployed containerized applications using **AWS ECS**, leveraging its managed cluster capabilities for scalable and resilient deployments.
- Configured **ECS task definitions** and integrated **Elastic Load Balancing (ELB)** for optimized traffic distribution.

##### **CI/CD Pipeline Automation with GitHub Actions & AWS CodePipeline:**

- Built and automated **CI/CD pipelines** using **GitHub Actions** and **AWS CodePipeline** to ensure smooth and reliable deployment processes.
- Integrated automated testing, build, and deployment stages, reducing manual intervention and minimizing errors.
- Configured pipelines to support deployment of containerized and serverless applications across multiple environments.

##### **AWS Cloud Management & Storage Solutions:**

- Managed and optimized **AWS resources**, including **EC2**, **EBS**, **S3**, and **VPC**, to ensure **security**, **availability**, and **cost efficiency**.
- Used **EBS** for persistent block storage, managing snapshots and optimizing volume types based on performance needs.
- Implemented **CloudWatch** and **CloudTrail** for monitoring and logging, configuring alerts for proactive infrastructure management.
- Deployed services with high availability using **Auto Scaling**, **Elastic Load Balancing (ELB)**, and **RDS**.

#### **Statefarm**

10/2022 – 06/2023 | Remote

##### **AWS Developer**

- Worked with **EC2**, **ECS**, **ELB**, **EBS**, **S3**, **VPC**, **IAM**, **SQS**, **RDS**, **Lambda**, **CloudWatch**, **Storage Gateway**, **CloudFormation**, and **Auto Scaling**.
- Worked with developer tools like **CodeDeploy**, **CodeBuild**, **CodePipeline**.
- Designed the overall **Virtual Private Cloud (VPC)** environment, including server instances, storage instances, subnets, **high availability zones**, etc.
- Worked with **CI/CD** using **GitHub Actions** and **AWS CodePipeline**.

#### **Education**

##### **International Islamic University Islamabad**

08/2018 – 08/2022 | Islamabad, Pakistan

##### *Bachelor in Computer Science*

Courses: Programming Fundamentals, Object-Oriented Programming, Data Structure & Algorithms, Analysis of Algorithms, Operating Systems, Database management system, Android Development, Web Development, Artificial Intelligence.

## Skills

- Html
- Tailwind
- Nest.js
- Ant Design
- Axios
- Docker
- RxDb
- CSS
- TypeScript
- MySQL
- Styled Components
- Micro-frontened
- TerraForm
- Zustand
- Javascript
- Redux
- Api's Integration
- Recoil
- AWS
- CI/CD
- React
- Next.Js
- GraphQL
- TanStack
- GitHub Actions
- GitHub