

## SHRUTHI M R

---

### EXPERIENCE SUMMARY

---

**Experience Level:** 4 Months

Miraфра Software Technologies Private Limited, April 2025 - Present

#### Education:

- Bachelor of Engineering (B.E.), Electronics & Telecommunication Engineering, Dr. Ambedkar Institute of Technology (Dr.AIT), Graduated: 2024

#### Technical Skills:

- Programming Languages: **JavaScript**, Typescript, **Python**
- Frontend: HTML5, CSS3, **React** (Hooks, Redux Toolkit, Context API), **Vue.js**
- Backend & APIs: Node.js, **NestJS**, JWT Authentication, Microservices, MVC Architecture
- Databases: MongoDB, MySQL, PostgreSQL
- Version Control: Git and GitHub
- Operating Systems: Windows
- Tools: VS Code, **Postman**, PyCharm

### PROJECT EXPERIENCE

Company Name: Miraфра Software Technologies Private Limited

Role: Software Engineer I

Project: **Farm2Fam – E-Commerce Web Application**

#### Description:

Developed a **full-stack e-commerce platform** for buying and selling **fresh farm products** using **Microservices Architecture** and the **MVC Design Pattern** to ensure scalability and maintainability. Implemented core features like **user authentication**, **cart management**, **order placement**, **PayPal payment integration**, and an **admin dashboard** for managing products and transactions. Included a **farmer panel** for product uploads with images and ensured a **responsive UI** with form validations and clean UX.

#### Tech Stack:

React.js, Vue.js, NestJS, MongoDB, Redux Toolkit, TailwindCSS, Zod, Axios, PayPal, Nodemailer, GitHub, Microservices.

#### Key Features & Implementation:

- **Role-Based Access:** Buyers can place orders; Farmers manage products via a dedicated dashboard.
- **Admin Dashboard (Vue.js):** Manage products, users, and transactions efficiently.
- **JWT Authentication:** Secure login system with access to user-specific orders and cart.
- **Payment Integration:** Integrated **PayPal** for safe and secure transactions.
- **Password Reset:** Email-based password reset using **Nodemailer** with secure token links.
- **State Management:** **Redux Toolkit** for user state, and **Pinia** for admin dashboard.
- **Architecture:**
  - **Microservices (NestJS):** Each service has its own **MongoDB** database for **fault isolation** and **scalability**.
  - **Strict MVC Design Pattern:** Followed across services for clean separation of concerns and maintainable codebase.
  - **Service-to-Service Communication:** Done using **TCP Protocol**.
  - **Client-to-Server Communication:** Done via standard **HTTP/REST APIs**.

## Project: **Smart Recruitment with Integrated Proctoring System**

### Description:

Built a **modular monolithic full-stack app** for secure **online assessments** with **MCQ and coding tests**, real-time **proctoring**, Designed modules for **user auth**, **test-taking**, **result tracking**, and **admin control** with **role-based access**. Enabled **question uploads** with options, **auto-evaluation**, and **dynamic delivery** by skill and difficulty. Ensured a **responsive, secure, and scalable** assessment workflow.

### Tech Stack:

React.js, TypeScript, Nest.js, PostgreSQL, Python, Redux Toolkit, Git & GitHub.

### Key Features & Implementation:

- **Role-Based Access:** Admins can upload/manage questions and view test results; Candidates can take tests
- **Admin Dashboard (React.js):** Easily manage users, upload questions by skill and difficulty, and track test attempts in real-time.
- **JWT Authentication:** Secure login for both users and admins, with protected routes and role-based access.
- **Test Timer & Auto Submission:** Countdown timer per test; automatic submission when time expires.
- **Test Link Sharing via Email:** Admin/TA can send test links to candidates using Nodemailer with secure tokenized URLs.
- **State Management:** Redux Toolkit used for managing authentication, questions, and test attempt data across components.
- **Dynamic Question Fetching:** Unique questions are dynamically fetched per test, based on selected skill and difficulty, ensuring no repetition.
- **Architecture:**
  - **Monolithic Architecture (Nest.js):** Backend contains all core modules (auth, question, test) in one service for ease of development and deployment.
  - **Strict MVC Design Pattern:** Codebase is modular and organized for scalability and maintainability.
  - **Client-to-Server Communication:** Communication between React frontend and Nest.js backend is handled via standard HTTP/REST APIs using Axios.
  - **Database (PostgreSQL):** Collections are structured for users, questions, and test attempts, with indexing for efficient queries and quick access.

## PERSONAL PROJECTS

### To-Do List Application

**Tech Stack:** React.js, TypeScript, HTML5, CSS3, JavaScript (ES6)

### Description:

Developed a **responsive task management application** using **React** and **TypeScript**, enabling users to efficiently manage their daily tasks. The application supports **task addition**, **completion toggling**, and **deletion** through an intuitive interface. Employed a **component-based architecture** and applied **TypeScript interfaces** to ensure **type safety** and **code maintainability**. Focused on **mobile responsiveness** using **Flexbox** and **media queries** to provide a consistent experience across various screen sizes.

## ACHIEVEMENTS

- Awarded **Gold Medal** in Engineering.
- Presented project at the 8th National Techno Exhibition at Dr. AIT.

## CERTIFICATION

- Internship at Karnataka Power Corporation Limited.
- Java Full Stack Course - Pentagon Space