Eugene Nunoo - Full Stack Developer Application

Eugene Nunoo - Full Stack Developer Application

1. The Basics

1. How many years of professional software development do you have?

I have 3 years of professional software development experience.

2. Are you available for a full-time role (40hrs/week)?

Yes, I'm available for a full-time role (40 hours/week).

3. What is your current city and time zone?

Accra, Ghana - GMT.

4. Provide a link to your portfolio. Which 3 projects do you want us to pay particular attention to?

Portfolio: https://eugene-nunoo.prostash.site/

Projects to highlight:

- Multi-store Inventory Management System

- E-commerce Application

- Property Management System

5. Provide a link to your GitHub profile. Which repo do you want us to pay particular attention to?

GitHub: https://github.com/Daemonlite

Highlighted Repository: https://github.com/Daemon-org/property-managment-front

6. Provide a link to your LinkedIn Profile.

LinkedIn: https://www.linkedin.com/in/eugene-nunoo-2947b8289/

2. Your Motivation (The "Why")

7. What's something you think is broken or frustrating about finding a place to rent or buy in Ghana?

How would you fix it?

One major frustration is the involvement of renting agents who inflate rental prices while offering subpar housing. This creates mistrust and inefficiency in the market.

Proposed Solution:

I would build a platform that connects renters directly with property owners. Verified listings, transparent pricing, and user reviews can help eliminate the need for middlemen. Incorporating geolocation filters and real-time availability would further improve trust and convenience.

8. Tell us about a personal project you built (outside work or education) because you were passionate or curious. What did you build, why did you build it, and what was the most challenging part?

I built a multi-tenant, multi-store inventory management system with tier-based license management and role-based access control. I created it to help businesses manage inventory across multiple locations efficiently. It's live at https://www.prostash.shop/

You can test it with:

- Email: admin@pharma.com

- Password: pass@123

The most challenging part was implementing dynamic tenant isolation for data, especially in scenarios where permissions and roles varied per organization.

9. Coziza is still in the early days. This means uncertainty, shifting priorities, and wearing many hats.

How do you feel about working in that kind of environment?

I enjoy working in fast-paced, dynamic environments. I see early-stage challenges as opportunities

to grow and make a meaningful impact. Wearing multiple hats helps me develop a broader skill set and contribute in creative and flexible ways.

3. Technical Aptitude

10. Describe a time when you had to learn a new technology or framework quickly to complete a project. How did you approach it, and what was the outcome?

I had to learn and integrate the Paystack payment API for a project with tight deadlines. I started by studying the official documentation, then created a sandbox environment to simulate real payment flows. I used Django and Python's requests module to handle transactions, validate webhooks, and securely process payments. The integration worked successfully, allowing seamless user payments with real-time feedback and status updates.

11. Coziza uses a stack that includes Node.js, React, PostgreSQL, and AWS. Pick one and tell us something clever, non-obvious, or deeply useful you've learned about using it effectively.

React: One useful pattern I've adopted is co-locating state with components only where needed and lifting it up only when truly shared. This prevents unnecessary re-renders and helps maintain separation of concerns.

Also, I've found that using React Query or SWR significantly simplifies handling server state and caching, improving responsiveness and reducing boilerplate code for loading/error states.

12. What's a bug or problem that took you a long time to figure out - and what did you learn from the process?

While building a multi-tenant inventory system, I initially struggled to segregate data properly between tenants. Records from one tenant were occasionally visible to another due to incorrectly scoped queries.

After hours of debugging, I realized the issue stemmed from how the tenant context was set in

middleware. I refactored the database access layer to ensure all queries were automatically filtered by the tenant ID.

This taught me the importance of context isolation in multi-tenant applications and reinforced the value of test-driven development when dealing with complex permissions and data access.

4. Detail & Product Sense (The "How")

13. Think of a digital product (any product) that you think is beautifully crafted. What is it and what makes it delightful or effective in your opinion?

Notion is one product I find beautifully crafted. It combines clean design with powerful functionality. The drag-and-drop interface, customizable layouts, and seamless collaboration features make it extremely intuitive. What makes it stand out is how flexible it is - you can use it for anything from documentation to personal task management, without needing to learn a complex UI.