KENNEDY CHINWO

FULLSTACK DEVELOPER | GO & NEXT.JS SPECIALIST | AI ENGINEER

chinwokennedy9@gmail.com | +234 913 706 8463 | github.com/org-cyber
pixelpro001.netlify.app

PROFESSIONAL SUMMARY

Experienced Fullstack Developer with deep expertise in Go (Golang) backend development, Next.js frontend engineering, and AI systems integration. Proven track record of architecting and deploying production-grade applications including AI-powered fraud detection systems, enterprise document intelligence platforms with RAG (Retrieval-Augmented Generation), and Web3 decentralized applications.

Specialized in building scalable cloud-native solutions with Go (Gin, GORM), Firebase/Firestore, and advanced AI integrations using Groq API, Llama models, and OpenRouter. Proficient in modern JavaScript/TypeScript ecosystems including Next.js 15+, React 19, and contemporary styling approaches with Tailwind CSS and Styled Components.

Passionate about cybersecurity, clean code architecture, and solving complex real-world problems through innovative technology solutions. Strong foundation in ethical hacking and security best practices with hands-on experience using Kali Linux.

TECHNICAL SKILLS

Backend Development: Go (Golang) with Gin framework, GORM ORM, RESTful API design, Node.js, Firebase/Firestore, SQLite, PostgreSQL, Cloud Functions, Microservices architecture

Frontend Development: Next.js 15+ (App Router), React 19, TypeScript, JavaScript (ES6+), HTML5/CSS3, Tailwind CSS, Styled Components, CSS Modules, Responsive Design, Progressive Web Apps

AI Engineering: Groq API integration, Llama 3/4 models, OpenRouter API, RAG (Retrieval-Augmented Generation) architecture, AI-powered analysis, Natural Language Processing, Vector embeddings, Document intelligence

Web3 & Blockchain: Sui Blockchain development, Solidity smart contracts, Wallet integration (@mysten/dapp-kit), Decentralized storage (Storj), Web3 authentication, Cryptocurrency payments

DevOps & Tools: Git, GitHub, Docker, VS Code, Chrome DevTools, Cloudinary, Firebase Admin SDK, Postman, npm/pnpm/yarn

Security & Other: Cybersecurity fundamentals, Ethical hacking with Kali Linux, API security best practices, Authentication & Authorization (OAuth, JWT), Clean code architecture, Test-driven development

WORK EXPERIENCE

Fullstack Developer Intern | Dmiebi IT Ventures, Port Harcourt

January 2024 - October 2024

- Developed and maintained full-stack web applications using modern JavaScript frameworks, Go backend services, and cloud technologies
- Collaborated with senior developers on architecture decisions, code reviews, and implementation of best practices
- Implemented secure authentication systems, database integrations, and RESTful API endpoints
- Gained hands-on experience with production deployment, CI/CD pipelines, and DevOps practices
- Contributed to AI-powered feature development and integration with third-party APIs

FEATURED PROJECTS

Asguard - AI-Powered Fraud Detection System

GitHub: github.com/org-cyber

Go 1.25.6, Gin Web Framework, Firebase/Firestore, RESTful API, AI Integration, API Key Authentication

Asguard is a cutting-edge fraud detection system designed to analyze financial transactions in real-time and assess their risk levels using a combination of rule-based scoring and AI-powered analysis. Named after the mythical realm of the gods, Asguard stands as a guardian protecting financial systems from fraudulent activities.

- Real-time Transaction Analysis: Process and score transactions instantly with sub-second response times
- Multi-factor Risk Assessment: Evaluate transactions based on amount, currency, device fingerprints, and IP address patterns

- Weighted Scoring Algorithm: Sophisticated algorithm with configurable thresholds that balances multiple risk factors
- Three-tier Risk Classification: Automatic categorization into LOW, MEDIUM, and HIGH risk levels
- AI-Enhanced Detection: Automatic AI engagement for high-risk transactions (score >= 50) with confidence scoring and natural language summaries
- Secure API Access: API key authentication middleware protecting all endpoints with environment-based configuration
- Cloud-Native Architecture: Built for scalability with Firebase/Firestore integration and context-aware database operations
- Comprehensive Logging: Detailed transaction analysis and audit trails for compliance and debugging
- Clean Architecture: Layered architecture pattern with Middleware Layer, Routes Layer, and Services Layer (Risk Engine, AI Service, DB Service)

Azeru - Enterprise Document Intelligence Platform (RAG)

GitHub: github.com/org-cyber

Go 1.25+, Gin, GORM, SQLite, Next.js, TypeScript, Tailwind CSS, Groq AI (Llama 3/4 Models)

Azeru is a powerful, AI-driven document intelligence platform designed to transform PDFs into searchable, actionable knowledge. It uses advanced Retrieval-Augmented Generation (RAG) to provide accurate answers based on private documents. Built with a scalable Go backend and modern Next.js frontend, this enterprise-ready solution enables organizations to unlock insights from their document repositories.

- PDF Intelligence: Seamlessly upload and process PDF documents with automatic text extraction
- Intelligent Chunking: Automatic text segmentation for optimal context retrieval and embedding generation
- RAG-Powered Chat: Chat with documents using Llama-powered AI via Groq API for contextually accurate responses
- BYOK Architecture: Bring Your Own Key design for enhanced security and cost management
- Go Backend Service: High-performance backend with Gin HTTP framework, GORM ORM with SQLite, and modular service architecture
- Vector Search: Semantic document search with embedding-based retrieval for finding relevant content
- Enterprise Scalability: Designed for enterprise deployment with clean separation of concerns

CodeReviewAI - AI-Powered Code Analysis Tool

Live: aya5.vercel.app | GitHub: github.com/org-cyber

Next.js 15+, React 19, TypeScript, Groq Cloud API (Llama-3 Models), Styled Components, Vanilla CSS Modules

CodeReviewAI is a premium, AI-powered code analysis tool that provides instant feedback on code scripts. Whether writing JavaScript, TypeScript, or Python, CodeReviewAI identifies bugs, suggests performance optimizations, and provides a fully corrected version of code in seconds. Built with cutting-edge technologies and powered by Groq's high-speed inference engine.

- Deep Error Detection: Catches syntax errors, logical bugs, and potential runtime crashes across multiple languages
- Smart Warnings: Identifies anti-patterns, security risks, and performance bottlenecks with senior-level insights
- Actionable Suggestions: Provides clear, actionable advice on improving code readability and maintainability
- Full Code Correction: Generates refined, production-ready versions of code snippets automatically
- Lightning Fast Analysis: Powered by Groq's high-speed inference engine for near-instant results
- Modern Interface: Sleek, dark-themed UI with glassmorphism effects and smooth animations
- Multi-Language Support: JavaScript, TypeScript, and Python code analysis in a single platform
- One-Click Copy: Copy corrected code with a single click for immediate use

Vaught - Insurance Dashboard Application

GitHub: github.com/org-cyber

Next.js 16 (App Directory), TypeScript, Firebase (Auth, Firestore, Admin SDK), Tailwind CSS, pnpm

Vaught is a modern insurance dashboard application built with Next.js, allowing users to manage policies, file claims, and track their status in real-time. This comprehensive platform streamlines insurance operations with secure authentication and intuitive user interfaces.

- Secure Authentication: Google Sign-In integration with Firebase Authentication for seamless user access
- Comprehensive Dashboard: Overview of active policies, claims, and notifications in a unified interface
- Claim Submission: Easy multi-step form to file new insurance claims with file upload capabilities
- Real-time Claim Tracking: Live status updates and timeline visualization for filed claims
- Policy Management: View and manage insurance policies with detailed coverage information
- Firestore Integration: Real-time database synchronization for instant data updates
- Modern App Router: Built with Next.js 16 App Directory for optimal performance

Hauser - Real Estate Listing Platform

Live: hauserhomes.xyz | GitHub: github.com/org-cyber

Next.js, React, JavaScript, Firebase, Firestore, Firebase Storage, Cloud Functions

Hauser is a full real-estate listing platform designed to allow agents and landlords to publish properties, manage listings, and track user interactions. Built as a full-stack web application with a strong focus on automation, clean UI, and scalable property-management features.

- Dynamic Agent and Landlord Dashboard: Role-based dashboards with custom PIN-based login system instead of traditional user accounts
- Auto-Detection & Routing: User roles (agent, landlord, visitor) are auto-detected and routed to different dashboards for seamless experience
- Interactive Property Listings: Dynamic property listing pages displaying properties for sale and rent with advanced filtering
- Premium Agent Highlighting: Priority highlighting for premium agents including badges like 'Verified' and 'Priority'
- Smart Description Handling: 'Read More' dynamic dropdown for long property descriptions with smooth animations
- Posting Limit Enforcement: Automatic hiding of the 'Add Property' button for free users after one listing
- Firestore Integration: Comprehensive database management for agent/landlord accounts, property uploads (title, price, description, images, location), and user-specific filtering
- Automatic Image System: Image upload using Firebase Storage with automatic display on website - no manual URL entry needed
- Property Analytics: Automatic click tracking visible only to premium users for market insights
- Persistent Limits: Deleted listings do not reset free-user upload limits to prevent abuse
- Premium Sorting: Public property pages fetch listings from Firestore with premium properties appearing first
- Real-time Updates: Live data synchronization from Firestore without page reloads

AVO (DApp) - Decentralized File Storage Application

GitHub: github.com/org-cyber

React, JavaScript, Sui Blockchain, Storj, REST API, Web3, Cryptography

Avo is a hybrid decentralized application (dapp) designed to provide secure file storage and streamlined communication using Web3 technologies. The platform integrates blockchain-based authentication, off-chain distributed file storage, and modern React-based UI to deliver a fast, privacy-focused user experience.

- Blockchain Authentication: Serverless-style authentication via wallet signatures on the Sui blockchain
- Encrypted File Management: Encrypted file uploads and retrieval with user-controlled data ownership
- Decentralized Storage: Storj integration for efficient, distributed large file storage
- Secure REST API: Backend service exposing secure REST endpoints that interact with Sui blockchain for action authorization
- Full-Stack Web3 Integration: Complete blockchain integration from frontend to backend
- Privacy-First Design: No reliance on traditional centralized storage - complete user data ownership
- Modern React Engineering: Clean component workflows with contemporary React patterns
- Production-Grade Architecture: Scalable, secure Web3 application architecture from scratch

Pixel AI - Direct AI Model Access Platform

GitHub: github.com/org-cyber

Next.js, React, JavaScript, OpenRouter API, Claude 3.5, Environment Management

Pixel AI is a web app that allows users to plug in their OpenRouter API key and get direct access to the Claude 3.5 model from a simple, clean interface - no backend setup or local environment needed.

- Secure API Key Management: Input and manage OpenRouter API keys with secure storage
- Dynamic Key Switching: Ability to switch and update API keys anytime without restart
- Smooth Chat Interface: Responsive UI for seamless conversation with Claude AI
- Modern Next.js Architecture: Built with contemporary Next.js routing, API handling, and environment logic
- Developer-Friendly: Lightweight way to test and use advanced AI models without complex setup
- Zero Configuration: No backend setup required - works entirely in the browser

DSPACE - Decentralized Collaboration Platform

Live: dspace1.netlify.app | GitHub: github.com/org-cyber

React, JavaScript, Sui Blockchain, Firebase Realtime Database, @mysten/dapp-kit, Vanilla CSS

DSPACE is a decentralized collaboration and hybrid chat platform built on the Sui Blockchain and Firebase, featuring real-time messaging, pod-based group management, and seamless crypto payments.

- Wallet Integration: Seamless sign-in using any standard Sui wallet (Suiet, Ethos, etc.) via @mysten/dapp-kit
- On-Chain Verification: Pod membership verified transparently on the blockchain for security
- Pod Creation & Management: Launch collaborative spaces (Pods) with single transactions and manage members by wallet address
- Glassmorphic UI: Premium, modern interface fully styled with advanced Vanilla CSS without Tailwind dependency
- Live Messaging: Real-time messaging powered by Firebase Realtime Database for instant communication
- Secure Context: Only active pods and members can interact with content
- History Management: Clear chat history or delete entire pods with admin controls
- Direct Crypto Transfers: Send SUI tokens directly to pod members without leaving the chat interface
- Integrated Payment Controls: Manage funds and payments from the Pod Dashboard

EDUCATION

Bachelor of Cybersecurity

Federal University of Technology Owerri (FUTO)

INTERESTS

Cybersecurity, Ethical Hacking, AI Integration & Automation, Machine Learning, Blockchain Technology, Clean Code Architecture, Open Source Contribution, Continuous Learning