

Ayodele Olayinka

Lagos, Nigeria | +234 816 593 0089 | olayinkacodes@gmail.com
olayinka.tech | github.com/theactualdev | linkedin.com/in/theactualdev

EDUCATION

University of Lagos

Bachelor of Science in Computer Science

Expected 2028

Lagos, Nigeria

Relevant coursework: Algorithms and Complexity, Data Structures, Concurrent Programming, Operating Systems

EXPERIENCE

Nevo Learning

Founding Frontend Engineer

Mar 2026 – Present

Lagos, Nigeria

- Built the frontend of an adaptive learning platform (nevolearning.com) from the first commit in Next.js and TypeScript
- Designed the core lesson interface (multi-learner lesson panel, question input, progress state) with the founding team, translating Figma designs into production React components
- Cut initial bundle size with route-level code splitting and lazy loading, targeting the low-end Android phones students use
- Established the component library, folder conventions and PR review process used by all frontend contributors

Soft-Web Digital

Software Engineer, Frontend

Nov 2024 – Jan 2025

Lagos, Nigeria

- Implemented 30+ Figma specifications as responsive React and Tailwind CSS screens, reviewed weekly with the design team
- Reduced initial load time on the main client app by code splitting and lazy-loading routes
- Onboarded junior frontend developers through pair programming and code review; documented shared component patterns

MicroBytes IT

Frontend Engineer Intern

Mar 2023 – May 2023

Lagos, Nigeria

- Shipped 5 user-facing features in vanilla JavaScript, HTML and CSS, refactoring repeated UI into reusable modules
- Fixed 15+ UI and layout defects in the first release cycle, working with senior developers through daily code review

PROJECTS

MSE LUX | *Next.js, React, TypeScript, Prisma, PostgreSQL, Paystack, Vitest* | github.com/theactualdev/MSE-LUX Live

- Built a production e-commerce storefront and role-gated admin for a Lagos jewellery brand: catalogue, cart, checkout, orders, transactional email (Resend) and live courier rates (ShipBubble)
- Integrated Paystack with server-side webhook signature verification; every charged amount is derived on the server in integer minor units (NGN and USD), never trusted from the client
- Signed shipping quotes with HMAC tokens, added Redis-backed rate limiting and a cron job that expires unpaid orders
- Shipped a deny-by-default pre-launch gate whose HMAC session cookie is keyed to the password, so rotating it revokes every session; deployed on Vercel with hourly ISR

Bleachers | *NestJS, Next.js, TypeScript, Prisma, PostgreSQL, WebSockets, Playwright* | github.com/theactualdev/bleachers Live

- Designed an event-sourced REST API for live grassroots sports scoring: every action is an immutable event, corrections void and append, and all statistics are derived by replay, never stored
- Built the offline-first PWA client with an IndexedDB event queue, optimistic UI, UUID idempotency keys and background sync so scorers can record matches without signal
- Wrote a framework-free scoring engine where each sport (football, basketball, volleyball) is a JSON config of periods, event types and derived stats; adding a sport adds no code
- Structured as a Turborepo monorepo with shared Zod schemas as the API contract, organisation and role authorisation (owner, scorer, viewer), Docker Compose, GitHub Actions CI and Playwright end-to-end tests

FaceBlur | *React, TypeScript, OpenCV.js (WebAssembly), Web Workers, Vitest* | github.com/theactualdev/faceblur Live

- Runs YuNet face detection in the browser through OpenCV.js compiled to WebAssembly; images never leave the device, and the app installs as an offline PWA with the model cached
- Moved model loading and inference into a Web Worker so the UI stays responsive while the 10 MB WebAssembly module compiles
- Hand-decoded YuNet's anchor-free outputs across strides 8, 16 and 32 with non-maximum suppression in a pure, unit-tested module; added a second detection pass to recover close-up faces the model misses
- Chose the 512 px input cap and 0.6 confidence threshold by measuring recall on group photos and false positives on negatives, cutting inference compute about 5x versus 1024 px with no loss on ordinary photos

SKILLS

Languages: TypeScript, JavaScript, Python, Java, C, SQL

Frameworks and web: React, Next.js, Node.js, NestJS, Tailwind CSS, GSAP, HTML, CSS

Backend and data: PostgreSQL, Prisma, Supabase, MongoDB, Firebase, Redis, REST APIs, WebSockets

Testing and tooling: Vitest, Playwright, GitHub Actions, Docker, Git, Vite, Turborepo, Figma, Postman