

JOHNNY L DALE Jr.

Marion, AR 72364 | (870)635-0843 | johnnydalejr@gmail.com

Portfolio: www.johnnydalejr.online | GitHub: github.com/midas78 |

LinkedIn: linkedin.com/in/johnny-l-d-17a643306

PROFESSIONAL SUMMARY

Full-Stack Software Engineer with 10+ years of hobbyist coding and a unique background in industrial automation and controls. Recent graduate bringing a mechanic's systematic approach to software development—debugging complex systems and solving problems from the ground up.

Built projects across native Android (Java/Room), mobile cross-platform (React Native + computer vision), full-stack web (Java Spring Boot + PostgreSQL), and modern frontend (Angular 19). Seeking mission-driven organizations where practical problem-solving and clean, maintainable code can drive meaningful impact.

TECHNICAL SKILLS

Mobile (Native): Android, Java, Room SQLite, Material3, RecyclerView, Fragments, Intents

Mobile (Cross-Platform): React Native 0.86 (Fabric/Hermes), Kotlin, VisionCamera 5

Frontend: Angular 19, React, TypeScript, JavaScript, HTML5, CSS3, Responsive Design

Backend: Java, Spring Boot, Node.js, Python, REST APIs, JPA/Hibernate

Database: PostgreSQL, SQLite, Room, H2, Supabase, IndexedDB, AsyncStorage, MongoDB

Computer Vision: Google ML Kit, face detection, landmark extraction, frame processing

DevOps: Docker, Render, GitLab Pages, Maven, JDK 17, Nginx, Git

Industrial: PLC Programming, Control Systems, Electrical Schematics

RELEVANT BACKGROUND

INDUSTRIAL MAINTENANCE & AUTOMATION | Various Employers

Arkansas, Tennessee, & Mississippi | Dates: 2001 – Current

Held progressive roles including Mechanic, E&I Technician, and Automation Controls Technician.

- Diagnosed and resolved complex electrical and mechanical faults in industrial systems, minimizing downtime
- Programmed and maintained PLCs and HMIs to optimize production workflows
- Developed systematic troubleshooting protocols and translated technical requirements across teams
- Built a foundation in logic, systems thinking, and high-stakes problem-solving that directly translates to software engineering.

PROJECTS

Triplet — *Vacation & Excursion Planner*

Android, Java, Room SQLite, Material3, RecyclerView, Fragments | 04/2026

Overview: A native Android app for planning vacations and excursions with full CRUD operations, business-rule validation, and SMS sharing. Built as a single-activity, multi-fragment application with offline-first data persistence.

Technical Highlights:

- Built a single-activity Android app with 6 fragments for creating, viewing, editing, and deleting vacations and excursions
- Integrated Room SQLite ORM for all CRUD operations across two entities (Vacation, Excursion) with type-safe DAOs
- Implemented date-picker dialogs with business-rule constraints: start date must be today or later, end date must be after start date, and excursions must fall within the vacation window
- Added SMS sharing via implicit Intent and real-time date-based alert toasts for user feedback
- Designed UI with RecyclerView and CardView for efficient list displays, Material3 theme for modern Android aesthetics
- Used manual fragment transactions for fine-grained navigation control
- Compiled to a signed APK and published via a custom landing page hosted on GitLab Pages
- Offline-first — all data stored locally; no network required

CheckMate — *Post-Drinking Impairment Estimator*

React Native, TypeScript, VisionCamera 5, Google ML Kit, Kotlin, Jest | 08/12/2026

Overview: Privacy-first mobile app that estimates post-drinking impairment using the device's front camera as a sensor—no data ever leaves the device. Users complete a one-time sober calibration, then future checks measure three facial signals against their baseline.

Technical Highlights:

- Engineered a 60-second calibration protocol capturing three signals: motor steadiness, pupil tracking accuracy (smooth-pursuit eye movement), and blink pattern analysis

- Implemented real-time on-device computer vision using VisionCamera v5 with native frame processors, streaming frames directly into Google ML Kit for face geometry extraction at frame rate—zero server latency, zero image uploads
- Designed a custom TypeScript scoring model blending three metrics against a per-user baseline (persisted with AsyncStorage) into a single impairment indicator, with 100% unit-tested scoring functions (Jest)
- Integrated video recording and playback aligned to Media3 1.9 via a targeted patch-package fix
- Shipped a production-ready Android app with custom Kotlin native modules, portrait-locked UI, device-density-aware layout, and universal release APK (arm64 + arm32) built with JDK 17
- Leveraged React Native 0.86 with the New Architecture (Fabric) and Hermes for high-performance native rendering
- Privacy by architecture — camera stream, analysis, and recorded video never leave the device; no backend by design

Wordie — *Full-Stack Word Puzzle Game*

Java, Spring Boot, PostgreSQL, Docker, Render, HTML/CSS/JS | 06/2026

Overview: A full-stack Wordle-like word game with a REST API backend, responsive frontend, and cloud deployment. Features a curated dictionary of ~2,300 5-letter words, user authentication, and persistent game stats.

Technical Highlights:

- Built a RESTful API with Spring Boot using JPA/Hibernate for data persistence and PostgreSQL via Supabase for stateful storage, with H2 in-memory fallback for local development
- Designed a responsive, mobile-first frontend with HTML/CSS/JS, featuring dark mode support and an iframe-based menu overlay for seamless navigation
- Implemented word seed loader processing an XLS file containing ~2,300 curated 5-letter words from the official Wordle answer bank
- Developed REST endpoints for random word selection, word validation, game statistics tracking, and user authentication
- Deployed to Render Web Service with containerized Docker build and environment-driven PostgreSQL connection configuration
- Implemented localStorage-based client-side session management with guest mode support for instant play without authentication
- Managed dependencies and builds with Maven, version control with Git

Interactive World Map Application

Angular 19, TypeScript, Docker, World Bank API, IndexedDB | 3/2026

Overview: A single-page interactive world map application featuring click-to-zoom SVG navigation with animated transitions, real-time country metadata display, and offline-capable data caching.

Technical Highlights:

- Built a single-page application with Angular 19 featuring click-to-zoom SVG navigation with animated transitions for intuitive map exploration
- Implemented IndexedDB-backed client-side data cache for persistent storage, enabling offline access and reduced API calls
- Integrated World Bank API via Angular HTTP services to fetch real-time country metadata including capital, region, population, area, income level, and flag
- Designed a responsive side panel UI with loading skeletons for smooth UX during data fetching
- Implemented accessibility features including keyboard navigation (ESC to close modals) and tooltip hover states
- Containerized with a multi-stage Docker build using Node 22 to nginx Alpine for optimized production deployment
- Leveraged Angular's dependency injection and reactive patterns for clean, maintainable service architecture

EDUCATION

BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

Western Governors University, Salt Lake City, Utah | 2026

- Relevant Coursework: Data Structures & Algorithms, Database Design, Web Development, Operating Systems, Software Testing, Mobile Development
- Capstone: Triplet mobile Android application

CODING JOURNEY

- 10+ years of hobbyist programming—started building small scripts, evolved into full-stack web, mobile, and modern frontend applications
- Active learner: constantly exploring new frameworks and building side projects
- Passionate about bridging hardware and software—bringing a unique "mechanic's perspective" to every codebase
-