Prajwal Navada G P

prajwalnavada74@gmail.com | +91 8660228838 | Bengaluru, Karnataka | LinkedIn | GitHub | Portfolio

SUMMARY

Second-year CSE student (CGPA 9.2) building toward systems and embedded software engineering. Designs reliable, resource-conscious systems — from a local-first client architecture handling persistence, indexing, and state without a backend dependency, to C++ firmware running TinyML inference on a single ESP32 within strict memory and compute budgets. Strong foundation in Operating Systems, Computer Networks, DBMS, and DSA. Interested in Linux systems, embedded software, real-time systems, and distributed software.

EDUCATION

Dr. H.N. National College of Engineering, Bengaluru 2024 – 2028
B.E. Computer Science & Engineering · Visvesvaraya Technological University **CGPA: 9.2**

TECHNICAL SKILLS

Programming:	C, C++, Java, Python, JavaScript
Core CS:	DSA, OOP, DBMS, Computer Networks, Operating Systems, Real-Time Systems, Distributed Systems
Linux & Systems:	Linux (Ubuntu), POSIX fundamentals, Git, GitHub, Shell scripting
Embedded:	ESP32, Arduino (C++), MQTT, Sensor Fusion, Signal Processing, ThingsBoard, DHT11, MPU6050
Software Engineering:	Node.js, Express.js, REST APIs, Socket.io, JWT, MySQL, React, PWA
Local-First Systems:	IndexedDB, Dexie, Zustand, Client-Side Indexing (Fuse.js), Offline-First Architecture
AI / TinyML:	TinyML (Edge Impulse), K-Means Anomaly Detection, Linear Regression, Predictive Analytics

PROJECTS

EdgeAI-Predict — TinyML Predictive Maintenance System GitHub
C++ · ESP32 · TinyML (Edge Impulse) · Signal Processing · MQTT · ThingsBoard

- Chose a single-MCU architecture over a Raspberry Pi gateway to cut cost and latency — required fitting sensor fusion, signal processing, and ML inference inside ESP32's RAM/flash budget.
- Built a hybrid decision pipeline (moving-average filtering → linear regression forecasting → rule-based thresholds → TinyML inference) so failures are caught even when no single signal alone exceeds a threshold.
- Resolved memory-constrained TinyML inference failures (Guru Meditation crashes) by restructuring buffer allocation and sensor sampling rate — stabilized continuous on-device inference.
- Implemented MQTT telemetry with bidirectional RPC to ThingsBoard, trading off update frequency against radio power draw for sustained real-time monitoring.

Base — Local-First Student Workspace GitHub | Live
React · IndexedDB · Dexie · PostgreSQL · Google Drive API · Fuse.js · PWA

- Designed a local-first architecture using IndexedDB and Dexie so the app remains fully functional with zero network dependency — data lives on-device first, sync is optional, not load-bearing.
- Implemented a hybrid sync model — PostgreSQL handles metadata while user-owned Google Drive folders serve as Knowledge Sources, eliminating dependency on centralized file storage without duplicating user files.
- Built client-side indexing with Fuse.js for sub-100ms search across workspaces, tasks, ideas, and linked resources without a server round-trip — trades a larger client bundle for lower latency and offline reliability.
- Architected the Knowledge Sources model to index selected Drive folders by reference rather than copy, preserving user ownership and privacy while still surfacing external files inside a unified workspace.

QuizLive — Distributed Real-Time Assessment Platform GitHub | Live
React · Node.js · Socket.io · MySQL · JWT

- Built synchronized multiplayer quiz sessions over Socket.io achieving sub-100ms score propagation across concurrent clients — required careful event ordering to avoid race conditions on simultaneous answers.
- Designed a 10-table normalized schema separating session state from persistent records, enabling concurrent real-time writes without locking contention; added JWT auth and anti-cheat session validation.

Auditorium Booking System — Multi-Role Backend Platform GitHub
Node.js · Express · MySQL · Firebase FCM · React · PWA

- Architected a decoupled controller-service-middleware backend handling role-based authorization (Admin/Staff/Principal) and approval workflows, designed to scale past the initial 3-college deployment without schema changes.
- Built automated conflict-detection logic at the booking layer rather than the UI, ensuring consistency regardless of client; added scheduled cron jobs and notification pipelines as independent background workers.

CERTIFICATIONS & LEADERSHIP

- **Programming in Java** — NPTEL (IIT Kharagpur) · Elite + Gold
- Led 3+ project teams, coordinating planning, implementation, testing, and delivery across software and embedded engineering initiatives.
- Active contributor to technical communities and peer-learning initiatives at college.