Interactive 3D Portfolio

GitHub: https://github.com/190-785/Interactive-Resume-Maker

- **Role: Frontend / Graphics Engineer**
  • Engineered an interactive 3D driving experience and vehicle physics using Three.js for responsive Ackermann-like steering and robust collision handling.
  • Built the WebGL scene, asset orchestration, and a drag-and-drop CMS that binds backend resume JSON to 3D surfaces for live edits.
  • Delivered an immersive portfolio UI with reliable local/demo fallback, enabling recruiter walkthroughs and consistent interview demos.

- **Role: Full-Stack Engineer**
  • Designed the end-to-end system connecting a Spring Boot REST API and MongoDB to the Three.js frontend with JWT-secured CRUD and ownership checks.
  • Implemented public default templates and deterministic error handling to support unauthenticated demos and safe live editing.
  • Simplified secure deployments and integration testing for recruiter-facing demos.

- **Role: Developer Experience / Systems Engineer**
  • Orchestrated asset loading and startup gating and built a Node.js auto-starter to ensure reproducible local development demos.
  • Optimized critical asset delivery and startup sequencing to avoid partial renders and failed demos.
  • Reduced developer friction and improved demo reliability for interviews and live showcases.

Real-time Community Finance Tracker

GitHub: https://github.com/190-785/Daily_Ledger

- **Role: Frontend Engineer (PWA)**
  • Built a React PWA with service worker, persistent storage, and Tailwind CSS to provide an offline-first, responsive financial dashboard.
  • Implemented realtime synchronization via Firestore onSnapshot listeners and a client-side stats cache to avoid redundant reads.
  • Improved offline resilience and reduced client read costs while maintaining instant cross-client updates.

- **Role: Backend / Serverless Engineer**
  • Designed Firestore schemas and authored granular security rules enforcing RBAC, ownership, and server-time validations at the database layer.
  • Implemented precomputed daily and monthly aggregations with staleness-aware caching to minimize Firestore reads and query costs.
  • Hardened production data integrity and enabled scalable, real-time financial reporting for community workflows.

- **Role: Data / Business Logic Engineer**
  • Modeled rolling outstanding balances as idempotent ledger entries to support partial payments, historical backfills, and soft-deleted members.
  • Built reconciliation paths, validation hooks, and an in-browser SheetJS export engine for member statements and month-wise audits.
  • Ensured auditability and prevented double counting to deliver accurate month-over-month financial reports.

Wikipedia Philosophy Crawler

Personal / Hobby Project · Backend, Automation
GitHub: https://github.com/190-785/First_Link_Backend

Role: Backend / Python Engineer

• **Engineered** a recursive web-crawling engine in Python using Selenium WebDriver to automate the Wikipedia “Getting to Philosophy” traversal rule.
• **Implemented** DOM-accurate link extraction via XPath selectors with per-request loop detection to prevent infinite traversals.
• **Optimized** crawl latency by introducing memoized jump paths using precomputed JSON routes for high-traffic articles.

Role: Systems / Reliability Engineering

• **Designed** deterministic session management with guaranteed browser teardown to avoid memory leaks during long-running crawls.
• **Built** a containerized headless Chrome runtime using Docker with all required Linux dependencies for stable execution.
• **Ensured** resilient execution of recursive automation workflows under partial failures and timeouts.

Role: API / Integration Engineering

• **Developed** a RESTful Flask API to expose traversal results for frontend consumption.
• **Handled** CORS preflight requests and cross-origin access to support a React/Vercel frontend.
• **Deployed** the service behind Gunicorn with extended worker timeouts suitable for stateful scraping tasks.

Ultra-compressed version (if you need to cut space)

WikiCrawler — Personal Project
Backend / Automation Engineer — Built a Python + Selenium crawler to automate Wikipedia Philosophy traversal; implemented XPath-based extraction, loop detection, memoized paths, and a Dockerized headless Chrome API.

Final Year Project Management Portal

Capstone / Product · Deployed (Aug 2025–Present) · Tech: Next.js 15, React 19, Firebase (Firestore/Auth), Tailwind CSS v4, ShadCN UI, Radix UI

Role: Full-Stack / Product Engineer
• Built the end-to-end portal for final-year project management, centralizing team formation, mentorship requests, multi-phase submissions, and panel evaluations.
• Implemented reactive UI with Next.js + React 19 and a component system (ShadCN + Radix) to support role-specific views (student, faculty, admin, panelist, external) and mobile responsiveness.
• Deployed a production UX that supports 1,600+ users (scales toward ~4,000), enabling reliable workflows for submissions, meetings, and bulk admin operations.

Role: Backend / Data Integrity Engineer
• Architected Firestore schemas and rules enforcing role-based access, audit trails, immutable revision/submission history, and atomic team creation via transactions.
• Implemented PanelEvaluationService and MentorshipService with batch writes and transaction-backed ops to prevent race conditions (no dual-mentor conflicts, safe assignment semantics).
• Delivered auditability and integrity: immutable revision_history/submission_history, effective deadline enforcement, and consistent aggregated grading across partial/absent evaluations.

Role: Systems / Reliability & Scale
• Engineered real-time sync using Firestore `onSnapshot` with mandatory cleanup patterns and efficient listeners to keep the dashboard live and minimize read amplification.
• Built admin capabilities: CSV import/export, bulk user/team creation, Excel/SheetJS exports for marks and reports, and workload balancing tools for faculty panels.
• Hardening and scale readiness for campus-wide adoption (1600+ users) with optimized client polling, batched notifications, and bulk Firestore write strategies.

Role: Product Safety, Compliance & UX
• Implemented conflict detection (UI + service layer + Firestore rules) to block faculty from evaluating teams they mentor and prevent accidental grade tampering.
• Built versioned submission flows and a PhaseSubmissionModal that archives previous submissions to `submission_history` before updating the active document, simplifying audits and rollbacks.
• Added meeting scheduling, temporary external panelist IDs with scoped access, role-based feedback collection, and downloadable CSVs for accreditation or external review.

Full Stack Developer Intern — Arterns Technologies Pvt. Ltd.

Yash_Agarwal_Arterns_Internship_Certificate.pdf - Google Drive

· Remote/Dehradun· **May 2025 – July 2025**.
Tech: React / Next.js, Firebase, Tailwind, i18n, Node.js, Socket.io, Realtime systems, Vercel

• Implemented production frontends for multiple client websites (Vedichealings, Motortb, OpulenceByAnchal, Stimulus Advertising, Arenax, Leela, Vedichealings-vercel and others), building responsive UIs from scratch and curating visual assets (Instagram sourcing + AI generation) to match brand needs.
• Built a real-time chat system and UX improvements for Menticate (live messaging, dark mode, accessibility enhancements), integrating realtime sync and client-side state management to improve user engagement and responsiveness.
• Implemented multi-language support (i18n) for selected sites, performed end-to-end frontend work (layout, accessibility, responsive behaviour), and owned deployment/CI tasks—delivering client-ready sites and cross-browser polished UI components.

Frontend-focused variant (use for UI / Frontend roles)

Tech: React, Next.js, Tailwind CSS, i18n, Recharts, Vercel

• Delivered end-to-end frontends for client sites (listed in portfolio), creating responsive, accessible UIs and designing brand-aligned visual content sourced from social media and AI when required.
• Led UX improvements and dark mode implementation for Menticate, plus a realtime chat integration to enable low-latency messaging and improved session continuity.
• Implemented localization (i18n) for multi-language support, performed cross-device testing, and shipped production deployments with Vercel and standard CI pipelines.

Full-Stack / SDE variant (use for backend / full-stack roles)

Tech: Next.js, Node.js, Firebase (Auth/Firestore), Socket.io/Realtime, Tailwind CSS, Vercel

• Built and integrated full-stack features across multiple client sites including a realtime chat system for Menticate and backend endpoints to support messaging, presence, and command flows.
• Owned end-to-end delivery: frontend implementation, lightweight backend glue, Firebase integration (Auth + Firestore) and deployments—ensuring consistent state, auth flows, and client ownership mapping.
• Implemented i18n, dark mode, and UX polish; curated and processed imagery for production assets; collaborated with designers and clients to deliver launch-ready websites.

Software Engineering Intern (IoT & Full-Stack)

DataCulture Technologies Private Limited (DCT) · New Delhi, India
May 2025 – November 2025

[Yash_Agarwal_DataCulture_Internship_Certificate.pdf - Google Drive]

Role Focus: Software Engineering / Systems

• **Engineered** end-to-end software for an IoT-based automated irrigation system, spanning embedded C++ firmware, cloud synchronization, and a production web dashboard.
• **Designed** deterministic state-driven control logic with fault handling, retries, and persistence to ensure correctness under partial failures and restarts.
• **Delivered** a reliable production system, earning **"Exceeded Expectations"** ratings across all technical and delivery metrics.

Role Focus: Backend / Distributed Systems

• **Built** a cloud-backed control and telemetry layer using **Firebase Firestore**, enabling real-time device state sync, historical logging, and remote command execution.
• **Implemented** asynchronous command handling and idempotent updates to safely coordinate remote actions with intermittently connected devices.
• **Enabled** scalable multi-device monitoring and remote management suitable for agricultural automation deployments.

Role Focus: Full-Stack / Product Engineering

• **Developed** the **BloomWatch** web application using **Next.js and React**, providing live dashboards, historical charts, and remote configuration for connected devices.
• **Implemented** secure user authentication, device ownership mapping, configuration UIs (thresholds, cooldowns), and safety controls to prevent pump misuse.
• **Co-authored and filed a patent** for the system's novel remote irrigation architecture and produced comprehensive technical documentation.

Ultra-condensed version (for very tight resumes)

Software Engineering Intern — DataCulture Technologies (May–Nov 2025)
Built an end-to-end IoT irrigation platform spanning embedded firmware, cloud synchronization, and a Next.js web dashboard; implemented fault-tolerant control logic, real-time telemetry, and remote configuration; co-authored a patent and received "Exceeded Expectations" ratings.

Technical Skills

Programming Languages

- **JavaScript (ES6+)**, **TypeScript**, **Python**, **Java**, **C++ (Embedded)**

Frontend Engineering

- **React 19**, **Next.js 15 (App Router)**, **Vite**
- **Tailwind CSS v4**, **ShadCN UI**, **Radix UI**
- Responsive UI, Accessibility (ARIA), Dark Mode
- Client-side state management, custom hooks
- Data visualization (**Recharts**)

Backend & Serverless

- **Node.js**, **Express**
- **Firebase** (Auth, Firestore, Security Rules)
- RESTful API design, async workflows
- Real-time systems (polling, listeners, event-driven updates)

Databases & Data Modeling

- **Firestore (NoSQL)**, **MongoDB**
- Schema design, role-based access control (RBAC)
- Transactions, batch writes, audit/version history
- Caching strategies, read-cost optimization

Systems & IoT

- **ESP8266 (NodeMCU)**, **PlatformIO**
- Embedded C++ firmware, finite state machines (FSM)
- Sensor-driven control logic, fault detection
- Cloud-connected IoT architectures

DevOps & Tooling

- **Docker**, **Git**, **GitHub**
- **Vercel** deployment
- **Gunicorn**
- CI/CD basics, environment configuration

Automation & Scraping

- **Selenium WebDriver**
- Headless Chrome, XPath-based DOM parsing
- Loop detection, memoization, containerized scraping

Other

- **i18n / Localization**
- Excel & CSV export (**SheetJS**)
- Technical documentation & system design