KARTIKEYA KUMARIA

SANTA CLARA, CA 95054 | 408-963-7480 | kumaria@ucsc.edu | LinkedIn | GitHub | Portfolio

Summary

Results-driven Computer Science student with proven skills in software development and analytical problem-solving. Adept at contributing to collaborative projects and delivering effective, user-centered solutions. Brings adaptability, persistence, and a focus on innovation to fast-paced development environments.

Education and Training

BACHELOR'S DEGREE: COMPUTER SCIENCE **Expected in 12/2025**
University of California, Santa Cruz

Skills

- JavaScript / TypeScript
- React.js / Next.js
- Django / Django REST Framework
- Node.js
- SQL / MySQL / PostgreSQL
- Git / GitHub / GitLab
- Agile / Scrum Methodologies
- Cross-functional Collaboration
- Debugging & Problem Solving
- Communication
- Time Management
- Mentorship & Onboarding

Experience

Software Developer **02/2024 to Current**
UCSC BLUEPRINT

- Built and deployed a volunteer registration mobile app for the Santa Cruz Mountain Arts Center (SCMAC) using Expo, React Native, and CSS, reducing manual sign-up time by 60% and increasing volunteer conversions by 40% within the first 3 months.
- Translated over 12 Figma components into fully responsive UI for both iOS and Android platforms, ensuring consistent performance across various screen sizes including tablets, improving cross-device compatibility by 95%.
- Integrated Firebase Firestore to fetch and render user-specific data (name, date of birth, email) on profile views, enhancing data accessibility and improving profile load times by 30%.
- Implemented custom error-handling for login and profile update workflows using Firebase Authentication, resulting in a 50% decrease in failed login attempts and improved user feedback during credential validation.
- Designed and launched YFIOB, a survey-based web app for K–12 career discovery using React.js, Next.js, Tailwind CSS, and Firebase, achieving 100% functional test coverage with React Testing Library, and onboarding 5+ contributors via documented Notion workflows and pair programming sessions..

Backend Developer Intern (Django & REST) **04/2024 to 08/2024**
SKYIT (SUBSIDIARY OF GBCS GROUP)

- Updated and cleaned client onboarding data by modifying null entries in MySQL and correcting CSV inputs, resolving data mismatches that were causing API ingestion failures and restoring full client onboarding functionality within 2 days.
- Onboarded 3 backend developer interns by setting up local environments with Python virtualenv, resolving MySQL port and credential issues, and configuring dependency installs using pip, enabling full backend run-time setup in under 1 day.
- Developed a custom Django REST Framework API endpoint to validate user authentication flow, tested POST and GET HTTP response statuses in Postman, and verified accurate field mappings in the relational database schema.
- Managed API enhancement tasks using GBCS University's internal workflow system, identifying and escalating deprecated endpoint logic to the executive director, and coordinating updates that reduced internal bug reports by 20%.
- Participated in cross-team sprint planning to ensure backend timelines aligned with frontend delivery schedules, helping reduce integration rework by 15% and ensuring milestone completeness on time.

Software Engineer Intern **10/2023 to 12/2023**
CODEDAY

- Extended Open Energy Dashboard (OED) by implementing feature modules to meet federal compliance requirements for agencies focused on climate analytics, contributing to dashboard readiness for real-time sustainability data reporting.
- Built a custom JavaScript-based conversion engine to process uploaded CSV files and transform kWh energy readings to BTUs, integrating unit logic directly into OED's backend workflow to support dual-unit reporting with zero data loss.
- Wrote and validated unit and integration tests using Mocha.js for energy conversion functions, CSV parsers, and edge cases (e.g., null/zero energy values), achieving full validation coverage for all energy data types.
- Used Git version control, configured Docker containers, and resolved merge conflicts for multi-developer collaboration, reducing environment-related setup issues by 40% and ensuring consistent builds across local and containerized setups.
- Debugged and deployed Node.js features using Mac/Linux terminal utilities, diagnosing cross-platform dependency issues and improving deployment reliability across development and staging environments.

Projects

Evil Number Guessing Game, 08/22/24 - 08/27/24, GitHub Repo, Link

- Developed a full-stack CRUD number guessing game with a randomly generated number between -1,000,000 and 1,000,000. Integrated
- background music, sound effects, and dynamic button visibility based on game states. Optimized user experience with interactive elements and real-time feedback.
- Utilized HTML, CSS and vanilla JavaScript for the front end to toggle button visibility and to send the user's guess.
- Used NodeJS and MongoDB for the backend to store the game logic, separate the logic into api routes for each gamestate, takes in the data from the user's guess in the front end and let's the game logic decide the response for that guess (Too High, Low, or Exact) and delivers a response message accordingly. Also delivers a response if the user decides to give up at any time

Full stack task manager application 02/21/2025-05/23/2025, GitHub Repo, Link

- Developed a full stack task manager application using React for the frontend, Redux for state management, HTML, CSS, JavaScript, Node.js, Express.js for the backend, and PostgreSQL17 with PgAdmin4 for the database.
- Implemented user authentication with JSON Web Tokens (JWT), created RESTful APIs, using Express.js for CRUD operations for adding, updating and deleting tasks once the user logs into their account, and optimized backend performance to handle high concurrency.
- Utilized libraries and tools such as Axios for HTTP requests, Sequelize as the ORM for PostgreSQL, and React Router for client-side navigation, enhancing task management efficiency and providing an intuitive user interface.

LogBlog - AI-Powered Full-Stack Blog Platform, 06/04/2025 - 07/10/25, GitHub Repo, Link

- Developed a comprehensive full-stack blog platform with AI tutorial generation featuring user authentication, password reset functionality, blog CRUD operations, and machine learning-powered content creation. Implemented JWT token-

based authentication with secure session management, automated email notifications, and role-based access control using Django REST Framework serializers and ViewSets.

- Built responsive frontend with React 18, JavaScript ES6+, JSX components, React Router for SPA navigation, Context API for state management, and custom hooks for API integration. Styled with Tailwind CSS utility classes, responsive design patterns, and Vite build tool for optimized bundling with hot module replacement and fast build times.
- Engineered advanced machine learning pipeline with SentenceTransformer (all-MiniLM-L6-v2) for high-dimensional text embeddings, PyTorch neural networks with encoder-decoder architecture, scikit-learn TfidfVectorizer for feature extraction, and NLTK for comprehensive text preprocessing. Implemented cosine similarity matching for content relevance and template-based generation with customizable tutorial structures.
- Trained custom ML models on extensive tutorial templates with automatic vectorization, feature extraction using pandas and numpy for data manipulation, and Hugging Face transformers for pre-trained model integration. Optimized model inference for efficient processing, implemented CPU-based inference for cost-effective deployment, and achieved high tutorial generation success rates with fallback mechanisms.
- Architected production-ready microservices deployment: Django REST Framework backend with Python deployed on Railway with comprehensive error handling, PostgreSQL database on Supabase with automated migrations and data integrity, React frontend with Vite deployed on Vercel with custom domain configuration and global CDN distribution.
- Configured advanced infrastructure: Custom gunicorn WSGI server with extended timeout for ML processing, optimized frontend Axios timeout handling, Django CORS middleware for secure cross-origin requests, whitenoise static file serving, and automated CI/CD pipelines with GitHub integration for seamless deployments.