NISHIT GROVER

+1 (602)-815-3837 | grovernt@mail.uc.edu | https://www.linkedin.com/in/grovernishit/ | nishitgrover.com

EDUCATION

Bachelor of Science, Computer Science Expected Graduation: May 2025

University of Cincinnati, Cincinnati, OH

- GPA*: 3.8*
- Honors and Awards: 4x Dean's List, CEAS International Outreach Scholarship, UC Global Scholarship, Transfer Ambassador @CEAS

SKILLS

- Programming Languages**: Python, C/C++, Java, JavaScript, TypeScript**
- Frameworks/Libraries: **Express.js, React, Next.js, Node.js**
- Front-End Technologies: **HTML, CSS, Bootstrap, Tailwind, Semantic UI**
- Databases: **MySQL, MongoDB, PostgreSQL**
- Operating Systems/Tools: **Visual Studio, Unix/Linux, Git/GitHub**
- Network and System Engineering: **AWS Certified Solutions Architect – Associate, Docker, Kubernetes**

EXPERIENCE

Software Engineer Intern @Intel May'24-Aug'24

- Developed significant features that impacted both internal and external stakeholders, while working independently and within teams
- Specialized in **Git** version control, deploying **Python** automation scripts for **RESTful APIs**, and creating **front-end applications** for hardware test suites. Utilized **Postman** for API testing and debugging to ensure robust and reliable integrations
- Designed and implemented **UI** enhancements, collaborated with **UX** designers and product leads to engineer a system that reduced testing time by over **8x** through parallel testing across multiple device generations

PDK Technical Intern @Intel May'23 -Aug'23

- Administered **JIRA**, developing customized workflows and dashboards while conducting extensive training and onboarding sessions for the PDK team. Empowered team members with proficient JIRA skills for effective issue tracking and project management
- Created a real-time visibility dashboard using **Power BI** and internal Intel tools, offering comprehensive insights into the progress and status of issues within the PDK team
- Collaborated on the development of a robust 5000+ line **Python** Scorecard script for the **Power BI** Dashboard, showcasing rapid learning and the ability to deliver production-ready code soon after joining

Software Developer @UC-CEAS Mar'23 -June'23

- Orchestrated and partnered with a student software developer to execute the **Software Development Life Cycle (SDLC)** to devise a scalable and interoperable **Python/PowerShell** script using **FFmpeg** and **Beautiful Soup** libraries within a two-week deadline
- Conducted extensive research on various **Python** modules to implement custom scripts utilizing the **VLC** media player, achieving significant efficiency in **data-scraping and frame-extraction** for over 1000 movies in 2 months

PROJECTS & EXTRACURRICULAR

MAKEUC(Hackathon)

- Developed robust backend solutions using **KeystoneJS** to manage data models and provide a **GraphQL API**, while leveraging **Prisma** for efficient and type-safe database interactions
- Engineered a dynamic and responsive **UI** using **Next.js 13+** and **Tailwind CSS**, while ensuring a seamless user experience with modern front-end technologies
- Enhanced data fetching and manipulation through **GraphQL**, allowing precise and flexible client-server communication, reducing over-fetching, and improving application performance

Real Estate Value Prediction Model

- Developed a predictive model using **linear regression** to analyze over 10,000 property listings, accurately estimating real estate values with 95% precision. Provided valuable market insights for investors
- Optimized the model using **NumPy**, **pandas**, and **matplotlib** libraries for enhanced efficiency. Refined data training and testing processes by leveraging **Scikit-learn**, optimizing model parameters, and reducing runtime by **25%**

BookStore – MERN Full Stack

- Developed a full-stack application using **Node.js**, **Express.js**, **MongoDB**, and **React**, demonstrating expertise in both backend and front-end development
- Implemented **RESTful APIs** and **CRUD** operations to manage book records, ensuring robust functionality for create, read, update, and delete operations
- Created a responsive **single-page application (SPA)** using **React**, **Vite**, and **Tailwind CSS**, incorporating reusable components and client-side routing with **React Router DOM**
- Enhanced the application's user interface with visually appealing alerts, modal dialogs, and a card-based layout for displaying book lists, improving usability and aesthetics