Alek Schmierer

(209)-327-1110 | alekschmierer@gmail.com | linkedin.com/in/alekschmierer | github.com/alekschmierer | alekschmierer.com

Education

Arizona State University Tempe, AZ
Bachelor of Science in Computer Science, GPA: 3.93 *Aug. 2022 – May 2025*

Technical Skills

Languages: Python, C/C++, SQL (Postgres), Java, JavaScript, HTML/CSS, Swift
Frameworks: Flask, Pandas, NumPy, Matplotlib, Chart.js, D3.js
Developer Tools: Git, VS Code, Visual Studio, PyCharm, Eclipse, XCode
Technologies: REST APIs, NASA API, JSON

Experience

Software Engineering Intern Aug. 2024 – May 2025
Living in Silico *Remote*

- Collaborated in an Agile Scrum environment, completing biweekly sprints with sponsor feedback to iteratively refine product design and features
- Designed the application's GUI using Figma and worked closely with teammates for frontend implementation and iterative UI updates
- Integrated third-party scientific tools (RxDock, RDKit) for immune genomics analysis, resolving compatibility issues with legacy command-line software using Conda environments
- Contributed to backend development using Python, focusing on frontend-backend integration and robust data processing for docking simulations

Projects

League Stat Tracker | *Python, Flask, Riot API, PostgreSQL, Scikit-learn* July. 2025 – Present

- Conceptualizing a web application to help friend groups track League of Legends stats, highlight synergy trends, and team-based performance insights.
- Designing architecture to integrate Riot Games API for fetching match history, champion stats, and player stats.
- Planning a PostgreSQL database schema to persist player profiles, group match data, and computed metrics.
- Prototyping a predictive model to estimate game outcomes based on team comp, champion matchups, and historical performance.

RxDock Portal | *Python, Flask, HTML, CSS, JavaScript* Aug. 2024 – May 2025

- Implemented a full-stack web application integrating RxDock for immune receptor-ligand docking simulations.
- Built backend services in Python and Flask for file handling, ligand library processing, and RxDock integration.
- Integrated RDKit to convert 2D molecular structures into 3D conformers to ensure compatibility with RxDock.
- Implemented parallel processing for ligand docking to improve performance on large ligand libraries up to 100%.

AstroTerra | *Swift, XCode, NASA APOD & LandSat API* Jan. 2025 – Feb. 2025

- Developed an iOS app in Swift using NASA's APOD and LandSat 8 APIs to display daily space imagery and satellite views of the user's current location.
- Integrated CoreLocation to fetch real-time geolocation and retrieve corresponding LandSat 8 satellite imagery.
- Built a TableView interface to browse and revisit historical APOD entries and LandSat captures by date.
- Designed an intuitive UI using SwiftUI to render dynamic content and enhance the user experience.

Weighted Graph Analyzer | *C++* Sept. 2023

- Implemented Dijkstra's algorithm in both C++ and C to solve shortest path problems on weighted graphs.
- Used efficient data structures such as min-heaps, stacks, and adjacency lists to optimize graph traversal.
- Computed shortest paths between arbitrary nodes with improved runtime, reinforcing understanding of algorithmic optimization and memory management.