Edson S. Cortes Rivera

767 Sunnyfield LN. Brooklyn, MD, 21225 | (443) 766 – 6576 | edcortes@terpmail.umd.edu

Education:

University of Maryland, College Park, MD
Bachelor of Science, Computer Science **Anticipated: December 2022**
Minor: General Business
UG Cumulative GPA: 3.49

Skills:

Coding and Software Experience: Worked on multiple projects using different languages and tools including Java, C++, C, JavaScript, Ruby, Python, OCAML, Rust, Android Studio, Kotlin, SQL, PostgreSQL, Peewee, Hadoop (Map Reduce), Android Studio, MIPS pipelining, Git and SVN version control, React, HTML, CSS and Javascript. Worked with FASTAPI in python to implement multiple endpoints for a RESTful API using HTTP requests. Worked on designing multiple efficient algorithms to solve complex problems.

Relevant Projects:

GitHub: *https://github.com/Edsoncortes40*, https://github.com/gutama8787/Rank-My-Store

MIPS Pipeline Simulator | *MIPS simulator* | *C*

- C project that simulates the pipelining of MIPS instructions. A file containing MIPS instructions are taken in and processed, simulating a MIPS architecture that implements instruction pipelining. The pipeline supports stalling and forwarding in case of any hazards and/or dependencies within the registers and the values stored.

Rank My Store | *Android Phone Application* **|** Kotlin

- Android application made using the coding language, Kotlin in Android Studio. Collaborated with two other Kotlin developers to create a social media app that allowed users to rate, rank, and share pictures of grocery stores in their area. Google Firebase was used to handle user authentication and for Database storage. Google Places API was used to get the user's location and nearby grocery stores. The ranking and store rating is based on healthy fruits and vegetables available at the given grocery store.

LAVI | *Language Agnostic Vulnerability Identifier* | Python

- CLI tool that allowed users to pass in a package along with its dependencies and output the package's vulnerabilities and exposures (CVE'S) associated with those dependencies and sub-dependencies. The CLI would communicate to an API via HTTP endpoints. The API would then query the LAVI database for CVE data associated with passed in package dependencies. This also included a LAVA (Language Agnostic Vulnerability Analysis) feature to help users get statistics on CVE's and Repositories such as Pip, NPM, and Golang. The queries that LAVA answered were such as, "how many CVE's exist in a certain repo".