Hamza Carew

Fullstack Developer

☎ *+1 (470) 637 7551*
✉ *hamzacarew@gmail.com*
Atlanta, GA

Education

2017–2020	**Computer Science BSc Hons (1st Class)**, University of Greenwich, London, United Kingdom.

Knowledge Area

Frontend	**HTML, CSS, TailwindCSS, Javascript, VueJS**.
Backend	**Golang, Python (Flask), ExpressJS**.
Database	**PostgreSQL**.
Other	**Docker, Kubernetes, Git, TravisCI, REST, Agile, Postman, CLI, Vim**.

Experience

2020 **Software Developer @Combine AI**, *Developing products with exposure to the entire stack. Hands-on usage with Vue.JS, SASS, Docker and PostgreSQL.*

- **Implemented multiple features and applied bug fixes across the UI of a production application**
- **Developed UI templates from mock-up wire frames**
- **Collaborated in a team to research and design the architecture of a new product that was delivered to production**
- **Completed tasks assigned to me within sprint timelines whilst leveraging agile methodologies**

My Projects - github.com/6ixBit

- **xStat**, *Docker, Micro-services, TravisCI, MongoDB, Flask, Vue.js, Pandas* ,
A data visualization platform to view and compare the statistics of football players from Europe's Top 5 leagues.

- Services containerized with Docker-compose
- Micro-services built with synchronous architecture
- Continuous integration for tests implemented with TravisCI
- Football data pulled from Third-party API on RapidAPI
- Endpoint responses are formatted with Pandas data analysis library
- Football data is visualized on front end service with radar and scatter charts
- Vue.js router, state management and AJAX requests implemented

- **FoxDos**, *GoLang, TCP, concurrency, worker pool, channels*,
  A denial of service CLI tool that uses threads (go routines) from a worker pool to open a plethora of TCP connections to a given host/IP address, built for educational purposes.
  - Developed using GoLang for speed and built-in concurrency
  - Uses multiple channels to communicate and track state of workers threads
  - Worker pool integration means go routines have a quicker start time and are constantly reused until jobs are complete
  - HTTP and HTTPs targets supported

- **University Intranet App**, *Express JS, MySQL, Mocha, JSON*,
  A full stack web application that served as an internal intranet for university staff and students. During the project I worked in a team of 7 utilising agile methodologies and mostly contributed as a back end developer and tester.
  - Wrote various unit tests for Express.js back end using Mocha
  - Collaborated in an Agile team to complete tasks on a weekly sprint basis
  - Developed SQL queries using JOINS and Sub queries to deliver data for endpoints
  - Implemented JWT authentication for back end during pair programming
  - Collaborated with team to develop user management for admin users
  - Developed endpoints and SQL queries to return formatted data on user data; most active users, departments with most posts, most liked posts etc.