Juan Vargas

(916) 289-6670 | juan@jvargas.dev | jvargas.dev | linkedin.com/in/jcvj/ | github.com/jcvargas1

EDUCATION

B.S., Computer Science January 2020
California State University, Fullerton *Fullerton, CA*

WORK EXPERIENCE

Student Information Technology Consultant July 2017 – January 2020
California State University, Fullerton *Fullerton, CA*
- Provided technical support for over 4,000 faculty and staff members on Windows- and Mac-based systems.
- Collaborated with team members and contacted various departments to assist in resolving niche technical issues.
- Assisted other I.T. departments in completing their respective goals.
- Utilized ServiceNow to manage tasks.

SKILLS

Languages: Python, C++, JavaScript, HTML/CSS, SQL
Frameworks and Libraries: React, Node.js/npm, Express.js, Styled Components, Material UI, Scrapy, Selenium
Developer Tools: AWS(EC2, RDS, S3, etc.), Google Cloud, Cloudflare, DigitalOcean, MongoDB Cloud, PostgreSQL, Nginx, CRUD/REST, APIs, Git
Other: Linux, Virtual Environments, Bash/Shell Script, Postman, VS Code, PyCharm, PgAdmin, Self-Hosting, Open-Source, Spanish (Native Fluency)

PROJECTS

TollCal.com | *React, Node.js/npm, Express.js, Nginx, AWS (EC2, RDS), Cloudflare, Postgres, MUI, Styled Components*
- Developed a full-stack web application that displays toll prices and traffic live feeds using React, Material UI, and Styled Components for the front-end and Postgres, Express.js, and Nginx for the back-end.
- Implemented a RESTful API to retrieve data from AWS RDS and save it in browser local storage.
- Utilized Nginx to provide a reverse proxy for the front-end and back-end.
- Hosted the website on AWS EC2 Ubuntu instance, managed by PM2.
- Utilized Cerbot for HTTPS and Cloudflare for SSL, DNS, Email Alias, Proxy, Firewall, allowing only U.S. IPs

Collectible Figures Scrapers | *Python, MongoDB (Cloud), Scrapy Framework, Selenium, Discord API, Self-Hosted*
- Developed two separate website scrapers using Python, Scrapy, and Selenium that monitor and scrape product information by crawling product pages, CRUDing the data to a MongoDB Cloud.
- Users are notified of new product additions and updated information (e.g., stock status, price) via Discord by using a Discord API webhook.
- Developed a separate bot to complement, allowing users to provide a product link with which the bot can calculate the product stock available and output the result.
- MongoDB cloud database contains over 20,000 unique entries so far.
- Script and bot is run on self-hosted Ubuntu server and triggered hourly by Cron Job/Shell Script.

Apex Legends Bot | *Javascript, Node.js/npm, MongoDB (Cloud), Discord JS, APIs, Google Compute Engine*
- Developed a Discord bot that uses a reverse engineered API for the game Apex Legends to track users' live play sessions and perform various commands for better overall engagement.
- Allows the user to quickly see unique data about their account that otherwise would not be available through the main game.
- Various CRUD functions responsible for manipulating JSON data and retrieving/uploading it to MongoDB Cloud.
- Hosted on GCP Compute Engine Ubuntu instance and managed with PM2.

Encrypted Image Cloud Storage | *AWS (EC2, S3), React, Bootstrap, Node.js/npm, Express.js, MongoDB, OAuth*
- Academic group project creating a full-stack application where users can upload images into the cloud to be stored and encrypted.
- Implementation of Google OAuth endpoints for user authentication and sharing capability.
- AWS S3 bucket for storing images and 4 EC2 instances with load balancing for other services.
- Utilized MongoDB as well as RESTful APIs with Node and Express.
- React and Bootstrap front-end provided the UI.
- AES-256 encryption/decryption for the images.