Sufyan Ayaz

(403) – 620 – 1920 | sufyan.raja263@gmail.com | www.linkedin.com/in/sufyan-ayaz-263 | github.com/SufyanAyaz

Education

University Of Calgary — Sep 2021 – Present

BSc. Software Engineering (Major) — *Calgary, Canada*

- Coursework: Full Stack Web Dev, Data Structures, Algorithms, OOP, Computer Organization, Databases
- In-Progress: OS, Embedded Software and Hardware, Software Architecture, Software Design Network Systems

Udemy — May 2022 – Aug 2022

The Web Developer Bootcamp — *Calgary, Canada*

Technical Experience

Projects

Wordle 2.0 | HTML, CSS, JS

- Built a front-end application using the popular game Wordle as inspiration, with this clone replicating the functionality of the original game.
- Fetches a dictionary of words and hints from an API endpoint, once per page refresh, and then uses key events from the user to populate the game board, remove letters, check answers, etc.
- In addition to being a clone of Wordle, the application has additional features such as icons the user may utilize to view game rules, get hints, and transitions between light and dark mode.

Lotion | HTML, CSS, JS (React.js), Terraform, Python, SQL, AWS

- Conceptualized a full-stack application, inspired by the widely used Notion software, that allows users to make personal notes and save, edit, or remove them at will.
- The app makes use of the react-oauth/google library, enabling each user to log in with Google on any device and access their private notes.
- The front-end of the application is built using React.js, HTML, and CSS, and the back end is built using Terraform, which connects to AWS services like DynamoDB, Lambda, and CloudWatch.
- All notes' data is saved as a SQL database in DynamoDB on the backend, with Python based Lambda Functions being used to fetch and manipulate the data.

The Last Show | HTML, CSS, JS (React.js), Terraform, SQL, Python, AWS

- A full-stack program users can use to write and preserve obituaries for their loved ones.
- The front-end of the application is built using React.js, HTML, and CSS, while the back end is built using Terraform, which connects to AWS services such as DynamoDB, Lambda, AWS Polly, Parameter Store, and CloudWatch.
- The program writes an obituary for each loved one, whenever a user creates an obituary, using the ChatGPT AI integrated into the Python based Lambda functions, which can subsequently be played to the user using AWS Polly.
- All obituary data is kept in DynamoDB as a SQL database in the backend, with Lambda Functions being used to fetch the data.

Scheduling Application | Java, MySQL

- Designed a Java computer application in a team of 4 that provides scheduling capabilities to a Wildlife Rescue Center, such that they can generate the most optimal employee schedule with the tap of a button.
- Implemented a scheduling algorithm that accurately analyses task data in the database and creates a schedule that is 100% more effective than supervisors, saving 100+ hours of annual administrative work.
- Used GUI library to create an application front-end, object-oriented Java to write the classes and algorithms, the Junit library to test application components, and utilized MySQL DBMS for the database.

Data Structure Library | Python

- Used Python to conceptualize class implementations of various Linear and Tree Data Structures, such that they could be compiled to create a data structure library.
- Used principles such as inheritance to reduce redundant code within functions written for each data structure, allowing them to run as efficiently as possible.
- Tested each function of the data structures using pytest framework, ensuring they work in accordance with the principles of each data structure.

Skills

Technical Skills: HTML5, CSS3, JS, Python, Java, C/C++, React, Terraform, SQL, AWS, Scrum-Agile

Soft Skills: Problem Solving, Critical Thinking, Communication, Teamwork, Adaptability, Leadership