Sufyan Ayaz

+1 (403)-620-1920 | sufyan.raja263@gmail.com | https://ca.linkedin.com/in/sufyan-ayaz-12064822a
https://github.com/SufyanAyaz | https://sufyanayaz.vercel.app/

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
- Consecutively achieved Jason Lang Scholarship

Udemy **May 2022 – Aug 2022**

The Web Developer Bootcamp *Calgary, Canada*

Skills

Technical Skills: HTML5, CSS3, JS, Python, Java, C/C++, React, Terraform, SQL, AWS, Assembly, Scrum-Agile, Git, VS Code, Netlify, Vercel
Soft Skills: Problem Solving, Critical Thinking, Communication, Teamwork, Adaptability, Leadership

Technical Experience

Projects

Wordle 2.0 | HTML, CSS, JS | https://github.com/SufyanAyaz/Wordle_Clone

- Created a front-end application inspired by the popular game Wordle, the functionality of which replicates the original game.
- The application 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, delete letters, check answers, etc.
- Additional features include icons that users can interact with to view game rules, receive hints, and change between light and dark modes.

Lotion | HTML, CSS, JS (React.js), Terraform, Python, SQL, AWS, Netlify | https://641de600f911e31baaf371f9--jolly-cactus-3b9395.netlify.app/

- Created a full-stack application, deployed through Netlify, inspired by the widely used Notion software, where users can create personal notes and save them, modify them, or delete them at will.
- The application utilizes the react-oauth/google library that allows every user to sign-in to the application, using Google, on any device and view their personal notes.
- Reacts.js, HTML, and CSS are used to create the front-end of the application, while Terraform is being used to connect with AWS services (DynamoDB, Lambda, and Cloudwatch) and forms the backend.
- In the backend, information of all notes is stored in DynamoDB as an SQL DB, and the Lambda Functions that are used to retrieve/manipulate the information are written in Python.

The Last Show | HTML, CSS, JS (React.js), Terraform, SQL, Python, AWS | https://github.com/SufyanAyaz/The_Last_Show_Project

- A full-stack application that users can use to create and store obituaries of their loved ones.
- Reacts.js, HTML, and CSS are used to create the front-end of the application, while Terraform is being used to connect with AWS services (DynamoDB, Lambda, AWS Polly, Parameter Store, and Cloudwatch) and forms the backend.
- Every time a user creates an obituary, the application uses the ChatGPT AI integrated within the Lambda functions to write an obituary for each loved one, which can then be played to the user using AWS Polly.
- In the backend, information of all obituaries is stored in DynamoDB as an SQL DB, and the Lambda Functions that are used to retrieve the information are written using Python.
- Modular structuring of application allows for content to be revised by changing the prompt used in the AI functionality.

Scheduling Application | Java, MySQL | https://github.com/SufyanAyaz/Animal_Schedule_Generator_Project

- 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 in the application that accurately analyzes task information within the database and generates a schedule 100% more efficient than supervisors, reducing 100+ hours of yearly administrative tasks.
- 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.

Art Museum Database | Python, MySQL | https://github.com/SufyanAyaz/Art_Museum_DB

- Using a provided narrative, implemented a Python command line database application for managing events, artifacts, and information.
- Integrated interfaces with separate access keys and authorities for the Admin, Data Entry user, and End user.
- Wrote SQL scripts that can be used in MySQL to create appropriate databases for the Arts Museum. Also wrote queries to evaluate the database's functionality.
- Linked application to database, enabling users to maintain, update, and lookup information based on degree of access.

Data Structure Library | Python | https://github.com/SufyanAyaz/Data_Structures_Library_Project

- Used Python to write 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 the functions written for each data structure, allowing them to run as efficiently as possible.
- Tested functions of each data structure using pytest framework, ensuring they work in accordance with the principles of each data structure.