Sufyan Ayaz

+1 (403)-620-1920 | sufyan.raja263@gmail.com | linkedin.com/in/sufyanayaz | github.com/SufyanAyaz | 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*

Technical Experience

Projects

Lotion | HTML, CSS, JS (React.js), Terraform, Python, Non-SQL DB, AWS, Netlify |

- **Full-stack application**, deployed through Netlify, inspired by the widely used Notion software.
- Allows users to create personal notes and save them, modify them, or delete them at will.
- Utilizes the **react-oauth/google library** to allow every user to sign-in to the application using Google, on any device, and view their personal notes.
- Front-end created using **Reacts.js, HTML, and CSS**, while **Terraform** connects with **AWS** services (DynamoDB, Lambda, IAM, and Cloudwatch) to form the backend.
- In the backend, **DynamoDB** stores information of all notes as a **Non-SQL DB**, and the **Lambda Functions** written in **Python** are used to retrieve/manipulate the information.

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

- A **full-stack application** that users can use to create and store obituaries.
- Front-end created using **Reacts.js, HTML, and CSS**, while **Terraform** connects with **AWS** services (DynamoDB, Lambda, AWS Polly, Parameter Store, IAM, and Cloudwatch) to form the backend.
- Application uses the **ChatGPT AI** integrated within the **Lambda functions** and **Cloudinary** to write an obituary, which can then be played to the user using **AWS Polly**.
- In the backend, **DynamoDB** stores information of all obituaries as a **Non-SQL DB**, and the **Lambda Functions** written in **Python** are used to retrieve the information.
- **Adaptable structuring** of the application code allows for content to be revised by changing the prompt used in the **AI functionality**, creating a whole new application.

Scheduling Application | Java, SQL DB |

- Designed a **Java** computer application in a team of 4 that provides scheduling capabilities to a Wildlife Rescue Center, generating the **most optimal** employee schedule with the tap of a button.
- Implemented a **scheduling algorithm** that accurately analyzes task information within the given database and generates a schedule **40% 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 classes and algorithms, the **Junit library** to test application components, and utilized **MySQL DBMS** for the database.

Wordle 2.0 | HTML, CSS, JS |

- **Front-end application** inspired by the popular game Wordle.
- 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.

Art Museum Database | Python, SQL DB |

- Using a given 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 in **MySQL Workbench** 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.

Skills

Technical Skills: HTML5, CSS3, JS, Python, Java, C/C++, React, Terraform, SQL DB, Non-SQL DB, AWS, Assembly, Scrum-Agile, Git, VS Code, Netlify, Vercel, MySQL Workbench, Cloudinary

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