



Faculty of Computer Science and Information Technology

***SQL\_DataDetective: Learn Standard Query Language (SQL) With a Webapp***

**Final Year Project 2**

Mcarthur Balang anak Mabong

Bachelor of Computer Science with Honours (Software Engineering)

2024

SQL\_DataDetective: Learn Standard Query Language (SQL) With a Webapp

MCARTHUR BALANG ANAK MABONG

This project is submitted in partial fulfilment of the requirements for the degree of Bachelor  
of Computer Science with Honours (Software Engineering)

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK 2024

## DECLARATION

I hereby declare that this project is my original work. I have not copied from any other student's work or from any other sources except where the due reference or acknowledgement is not made explicit in the test, nor has any part has been written form by another person.

  
.....  
(MCARTHUR BALANG ANAK MABONG)

MATRIC NO: 72327

.....31<sup>st</sup> DECEMBER 2023.....  
( DATE )

## **ACKNOWLEDGMENT**

Firstly, I would like to thank my supervisor, Dr Bong Chih How for the guidance, comments, and advice for my final year project. The fruition of the report come to realization due to his supervision to make sure the project and the report are handled up to a high quality standard.

Secondly, I would like to express my gratitude to Faculty of Computer Science and Information Technology (FCSIT) for providing me essential resource for the completion of the project and report.

Lastly, I would express my greatest appreciation to my family and friends for the support and encouragement to motivate me complete this report and project.

## Table of Contents

|   |    |
|---|----|
| <b>DECLARATION</b> .....  | 3  |
| <b>ACKNOWLEDGMENT</b> .....   | 4  |
| <b>List of Figures</b> .....  | 7  |
| <b>ABSTRACT</b> .....   | 9  |
| <b>ABSTRAK</b> .....  | 10 |
| <b>CHAPTER 1: INTRODUCTION</b> .....  | 11 |
| <b>1.1 Introduction</b> .....   | 11 |
| <b>1.2 Problem Statement</b> .....  | 12 |
| <b>1.3 Scope</b> .....  | 13 |
| <b>1.4 Aims and Objective</b> .....   | 13 |
| <b>1.5 Brief Methodology</b> .....  | 14 |
| <b>1.6 Phases in Agile Methodology</b> .....                                      | 14 |
| <b>1.6.1 Requirement Analysis</b> .....   | 14 |
| <b>1.6.2 System and Software Design</b> .....                                     | 15 |
| <b>1.6.3 Implementation</b> .....   | 15 |
| <b>1.6.4 System Testing</b> .....   | 15 |
| <b>1.6.5 Deployment</b> .....   | 15 |
| <b>1.6.6 Review</b> .....   | 15 |
| <b>1.7 Significance of Project</b> .....  | 15 |
| <b>1.8 Expected Outcome</b> .....   | 16 |
| <b>1.9 Project Outline</b> .....  | 17 |
| <b>1.10 Summary</b> .....   | 18 |
| <b>CHAPTER 2: LITERATURE REVIEW</b> .....   | 19 |
| <b>2.1 Overview</b> .....   | 19 |
| <b>2.2 Introduction</b> .....   | 19 |
| <b>2.3 Relevant Research</b> .....  | 20 |
| <b>2.4 Review of Literature</b> .....   | 20 |
| <b>2.5 Gaps and Limitation in Research</b> .....                                  | 22 |
| <b>2.6 Improvement on the Proposed Webapp Compared to Existing platform</b> ..... | 23 |
| 2.6.1 Increase the variety and complexity of puzzles and scenarios offered.....   | 23 |
| 2.6.2 Provide more in-depth explanations and feedback.....                        | 24 |
| <b>2.7 Conclusion</b> .....   | 24 |
| <b>CHAPTER 3: METHODOLOGY</b> .....   | 26 |
| <b>3.1 Introduction</b> .....   | 26 |

|            |   |           |
|------------|---|-----------|
| <b>3.2</b> | <b>Overview .....</b>                                 | <b>26</b> |
| <b>3.3</b> | <b>Requirement Analysis.....</b>                      | <b>27</b> |
| <b>3.4</b> | <b>System and Software Design .....</b>               | <b>28</b> |
| 3.4.2      | Entity Relationship Diagram .....                     | 29        |
| 3.4.3      | User Interface .....                                  | 30        |
| 3.4.3      | Storyboard .....                                      | 31        |
| <b>3.5</b> | <b>Implementation .....</b>                           | <b>32</b> |
| 3.5.1      | Database Management System .....                      | 32        |
| 3.5.2      | Programming and Scripting Language .....              | 33        |
| 3.5.3      | Web Server .....                                      | 33        |
| <b>3.6</b> | <b>System Testing.....</b>                            | <b>34</b> |
| <b>3.7</b> | <b>Deployment and Review .....</b>                    | <b>35</b> |
| <b>3.7</b> | <b>Conclusion .....</b>                               | <b>35</b> |
|            | <b>CHAPTER 4: IMPLEMENTATION .....</b>                | <b>36</b> |
| <b>4.1</b> | <b>Introduction .....</b>                             | <b>36</b> |
| <b>4.2</b> | <b>System Development .....</b>                       | <b>36</b> |
| <b>4.3</b> | <b>SQL_DataDetective Frontend Implementation.....</b> | <b>36</b> |
| 4.3.1      | User Interface .....                                  | 36        |
| <b>4.4</b> | <b>SQL_DataDetective Backend Implementation.....</b>  | <b>42</b> |
| 4.4.1      | Logic Management.....                                 | 42        |
| 4.4.2      | Building the Database.....                            | 43        |
|            | <b>CHAPTER 5: TESTING .....</b>                       | <b>46</b> |
| <b>5.1</b> | <b>Introduction .....</b>                             | <b>46</b> |
| 5.1.1      | Functional Testing .....                              | 46        |
| 5.1.2      | Test Cases for Functional Testing .....               | 48        |
| <b>5.2</b> | <b>Summary .....</b>                                  | <b>51</b> |
|            | <b>CHAPTER 6: CONCLUSION AND FUTURE WORKS .....</b>   | <b>52</b> |
| <b>6.1</b> | <b>Introduction .....</b>                             | <b>52</b> |
| <b>6.2</b> | <b>Project Achievement .....</b>                      | <b>52</b> |
| <b>6.3</b> | <b>Project limitations .....</b>                      | <b>52</b> |
| <b>6.4</b> | <b>Future works.....</b>                              | <b>53</b> |
| <b>6.5</b> | <b>Summary .....</b>                                  | <b>53</b> |
|            | <b>REFERENCES.....</b>                                | <b>54</b> |
|            | <b>APPENDICES .....</b>                               | <b>55</b> |
|            | <b>Project Planning .....</b>                         | <b>55</b> |

## List of Figures

|  |    |
|--|----|
| Figure 1. 1 Agile Methodology .....  | 14 |
| Figure 3. 1 Agile Methodology .....  | 27 |
| Figure 3. 2 Use case diagram .....   | 28 |
| Figure 3. 3 Entity Relationship Diagram.....   | 29 |
| Figure 3. 4 Beginner's User Interface.....   | 30 |
| Figure 3. 5 Input Answer User Interface .....  | 31 |
| Figure 4. 1 Homepage Interface .....   | 37 |
| Figure 4. 2 Difficulty Selection example.....  | 37 |
| Figure 4. 3 CSS that allow the description to transition on the button .....               | 38 |
| Figure 4. 4 Body of the HTML Structure .....   | 38 |
| Figure 4. 5 Beginner SQL_DataDetective Interface .....                                     | 39 |
| Figure 4. 6 HTML for tutorial slide content example .....                                  | 39 |
| Figure 4. 7 function for automating typing in text area .....                              | 40 |
| Figure 4. 8 A minimalist interface .....   | 40 |
| Figure 4. 9 simple JavaScript allows mission tab open on default.....                      | 40 |
| Figure 4. 10 Entering SQL command from Keywords tab .....                                  | 41 |
| Figure 4. 11 Entering Table Names from the Tab .....                                       | 41 |
| Figure 4. 12 Submit Answer Interface.....  | 41 |
| Figure 4. 13 JavaScript responsible for sending a user-entered SQL query to a server ..... | 42 |
| Figure 4. 14 server-side implementation using Node.js with Express and SQLite.....         | 43 |
| Figure 4. 15 SQLiteStudio Interface .....  | 43 |
| Figure 4. 16 100 Sample data of residents in SQL City .....                                | 44 |
| Figure 4. 17 JavaScript that enable connection between database and webpage .....          | 45 |
| Figure 5. 1 Installing Selenium Package .....  | 47 |
| Figure 5. 2 database configuration for testing.....  | 47 |
| Figure 5. 3 test server configuration.....   | 47 |
| Figure 5. 4 line for exporting the app for testing.....                                    | 47 |

|   |    |
|---|----|
| Table 5.2 functionalities to be tested on ..... | 46 |
| Table 5.3 Test Cases of Functionalities .....   | 48 |
| Table 4 Objectives and Achievement.....         | 52 |



## **ABSTRACT**

SQL\_DataDetective is the web app that teaches user to learn SQL in a more interactive and enriched learning experience approach. To achieve that, SQL\_DataDetective teaches user through 3 stages of learning difficulty that teach and implement the use of SQL queries and syntax through scenarios. SQL\_DataDetective use scenario to motivate users to find clue and use it as syntax for SQL input. The interface is done in a beginner's friendly manner, that have tips and guide for each scenario. To enrich the learning experience, data exploration through relational database is essential to make the learning experience feels more rewarding, especially on higher difficulty, where the data extracted previously may be crucial for the next scenario that chained from the previous scenario. With this, user learning SQL experience can be enhanced with motivation for reward are in mind as well using human nature of curiosity to explore the database using SQL. The outcome of this project is that user able to be more proficient in SQL and exposed to the nature of basics data extraction within the SQL database.

## **ABSTRAK**

*SQL\_DataDetective* adalah aplikasi web yang memperkayakan pengajaran SQL dengan pembelajaran yang lebih interaktif. Untuk mencapai itu, *SQL\_DataDetective* mengajar pengguna melalui 3 tahap kesukaran pembelajaran yang mengajar dan menerapkan penggunaan SQL dan sintaks melalui senario. *SQL\_DataDetective* menggunakan senario untuk memotivasi pengguna untuk mencari petunjuk dan menggunakannya sebagai sintaks untuk dimasukan sebagai SQL. Antara muka dilakukan dengan cara yang ramah bagi pemula, yang memiliki tips dan panduan untuk setiap skenario. Untuk memperkayakan proses pembelajaran, eksplorasi data melalui pangkalan data hubungan sangat penting untuk membuat pengalaman belajar lebih berkesan, terutama pada tahap kesukaran yang lebih tinggi, di mana data yang diekstrak sebelumnya mungkin sangat penting untuk senario berikutnya yang terkait dengan senario sebelumnya. Dengan ini, pengalaman belajar SQL pengguna dapat ditingkatkan dengan motivasi untuk ganjaran di dalam minda, yang juga menggunakan sifat manusia untuk mengeksplorasi basis data menggunakan SQL. Hasil dari projek ini adalah pengguna dapat lebih cekap dalam SQL dan menededahkan diri dengan sifat asas ekstraksi data di dalam data SQL.

## **CHAPTER 1: INTRODUCTION**

### **1.1 Introduction**

SQL query is a standard query language that is designed to navigate through SQL relational database. This query language has aged for five decades old, and yet, its relevancy and contribution to the modern information technological advancement is one of the mankind greatest accomplishments. SQL may have changed the past five decades, but mostly it is the same as it was, to querying, updating, changing, adding, and deleting data in relational database (Custer, 2021).

Due to SQL query capabilities to access, store and manipulate data in relational database, it is widely used by many sectors in many organizations. Most organization that has growing businesses rely on data warehousing by SQL database. These databases may contain data of their employees, clientele and important data and information that keep the organization running efficiently. Most data collected are valuable resources to chart up statistics, organizing a business model and be well informed of the state of the company from the first-year data was collected to the current.

As more organization are growing, and population of human keep increasing, the reliant of SQL databases also increasing. Every day, more data is collected, added, removed, and updated to keep the organization well informed for the sake of the businesses. This leads to huge database is made and accessing it without prior knowledge of SQL is overwhelming and impossible. Most organization are good at collecting data but are unable to effectively use it for the benefit of the organization (Editorial Team, 2018). Hence, there are data scientist who are up to the task, who are expert in extracting and identifying specific data from overwhelming database. Data scientist will use SQL to extract and identify specific data from the database based on the required condition set by the business model requirement and usually accompanied by deeper analysis using Python or R programming language.

To navigate through the database, prior knowledge to SQL is important. Hence, the aim of this project is to enrich the learning experience for SQL, especially among beginners. Started by generally introduce the basics of SQL, user will teach various queries which also can be applied in the webapp. The learning experience will be through a webapp interface that allow interactive learning, where user can input queries to find solution for the given situation or question. This approach reward user, by using the correct queries along with the correct naming and navigation through the tables in relational database, they will find the correct solution for

given situation or question. This approach is used by a webapp interactive SQL puzzle, The SQL Murder mystery by Knight Lab fellows, Joon Park and Cathy He.

To enrich user SQL learning experience even further, the webapp will provide a more complex situation to be solve. These situations usually required user to aggregate queries learnt briefly from the simple situation and questions. The complexity of the situation required user to extract and identify data even deeper into the relational database. Extracted data may be useful to correlate the findings of other data. The user may encounter patterns and association among patters that will be useful to solve the situation. The purpose of this criteria is to teach user the fundamental of filtering, identifying, and extracting relevant data from big database, which is the basics element of data mining.

By enriching the SQL learning experience and enforcing the skills and technique of basics data mining, user capability to use SQL on a complex scale on big database will be more efficient. With all the basics at hands, user who are invested in this webapp will find themselves capable of applying complex SQL and extracting data that will be useful for data preparation, which is the early steps of data mining.

## **1.2 Problem Statement**

Learning SQL query are easy due to its minimal coding requirement as it relies heavily on English sentences with minimal special characters (Leverage Edu, 2021). However, with just basic SQL and failing to understand the correlation usage of queries can be challenging. Fortunately, there are existing webapp that teaches user to learn SQL interactively. By means of using data exploration and scenarios as a core game mechanics of the webapp, user are able to utilize the SQL capability that they learn. Similarly, the proposed webapp aim to achieve those feats and improving the existing problem with existing system.

Interactive learning is one of the most efficient ways to teach a lesson, hence, the use of interactive web-based interface to teach user to learn SQL can be enriching as it involve more interaction and engagement during the learning process. However, there are only a few existing systems implement learning SQL with interactive learning, such as SQL Murder Mystery and SQLZoo. SQL game-based learning are not a polarized and popular compared to others such as code.org which are widely used by most academic institutions to teach and introduce younger generation to coding. Hence, the addition of this webapp will add more variation for beginners to learn SQL as well popularize the existence of SQL game-based learning media as SQL is one of the most valuable skills in most industry.

Despite the existence of SQL game-based interactive webapp, the learning approach still using the same conventional method, that is tutorials of list of SQL syntax such as in SQLZoo. This may discourage new learner to learn about SQL, hence the proposed webapp will provide the necessary SQL tutorials related to the problem prompted by the webapp.

Existing webapp also have lack of replayability and flexibility in the game-based scenarios. Scenarios are one of the best ways to anticipate user engagement with the webapp. The scenario needs to have more story path to provide more flexibility and variation for each variation for user gameplay experience. Flexibility of the story path of scenario will increase replayability for the user making it more entertaining as the proposed system acknowledge user ability to solve a scenario in a different approach.

### **1.3 Scope**

The scope of this project is to introduce SQL query, usage and how to facilitate with SQL queries through relational database by identifying relevant data:

- i. To introduce and teach basic SQL.
- ii. Usage of webapp interface to provide interactive learning experience.
- iii. Usage of relational database to for user to implement SQL.
- iv. Encourage data exploration and data extraction as process for learning SQL.

### **1.4 Aims and Objective**

The objective of this project is focusing on the learning of SQL and data identification with a sequential approach of these objectives.

- i. To develop a web application with an interactive interface for introducing SQL queries.
- ii. To design a learning curriculum that starts from basic and gradually increase in complexity of learning SQL.
- iii. To evaluate users' ability to identify and extract data by testing their proficiency in applying the SQL learned.
- iv. To assess the users' understanding of data quality by measuring their ability to identify relevant data relationships within a relational database through a series of data analysis simulations.

## 1.5 Brief Methodology

To accomplish the development of the project, Agile methodology approach is used. This model respects the requirement of Software Development Life Cycle (SDLC). This model allows feedback path from the previous phase that allows correcting error for the later phase of the development (Kumar, 2021). This methodology consists of 6 phases.

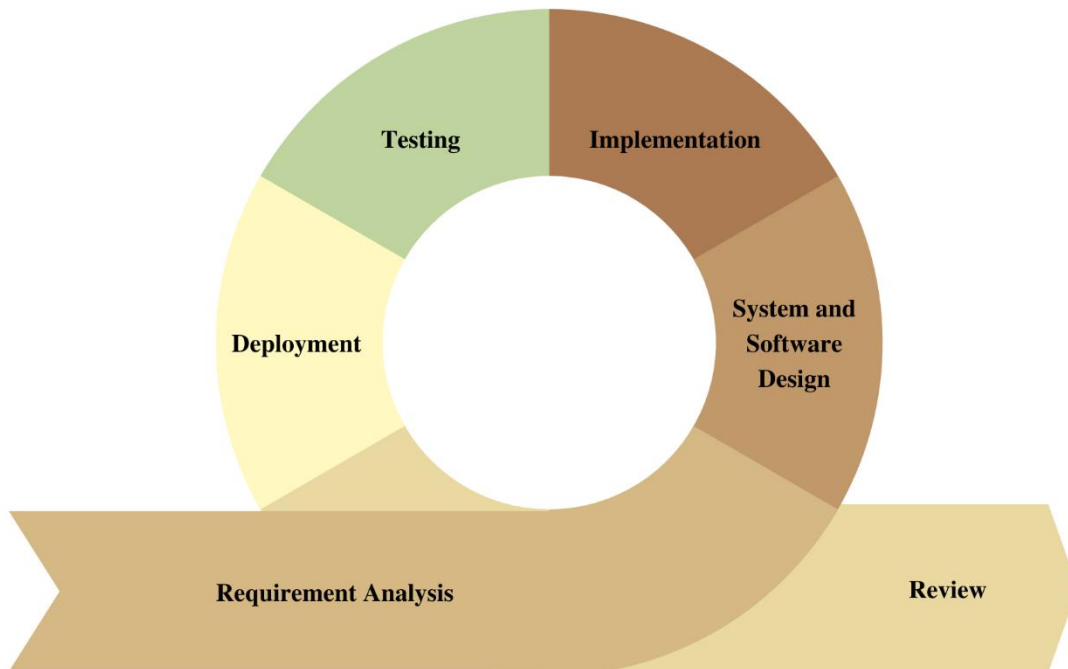


Figure 1. 1 Agile Methodology

## 1.6 Phases in Agile Methodology

### 1.6.1 Requirement Analysis

Requirement analysis that needs to be done is identify project objectives, outcomes, required software and hardware tools. The objective required is interactive teaching of SQL and using SQL to solve complex situation to simulate data extraction exercise. This required research on interactive learning interface and chained scenario to mimic data extracting simulation that fully utilise relational database. The required software tools are SQLite for relational database, storyboard software to create scenario and Sublime Text 3 or any similar tool to create a HTML webapp. Required hardware are any device that can access to the internet to load webpages, such as phone or laptop.

### 1.6.2 System and Software Design

The design focus is on beginner's friendly interface, with a command prompt like interface with guided prompt for learning SQL. The learning is designed to be less verbose, more interactive, and visual.

### 1.6.3 Implementation

At this phase, the development of the webapp began, based on the design and requirement researched by previous phase. The decision here is made on which requirement and design seems feasible to the objective requirement. The development of the interface and webapp will be using HTML, CSS, and JavaScript. The creation of relational database will be using SQLite with SQL, C#, and PHP. The scenario will be created using Canva, Adobe Photoshop and Storyboarder software by wonder unit.

### 1.6.4 System Testing

In this phase, using feedback path is important. Detecting error or bugs from different implementation from each phase will enable the fixing of error to be traced. It could also be from requirement from each phase are underdelivered or fail to be delivered. This is to ensure to achieve the objective of the project meanwhile accommodating for an excellent user experience.

### 1.6.5 Deployment

In this phase, the webapp is ready to be release and be tested to real user. The purpose of this phase is to get user to use the webapp to test the user experience. Any review, comments and critics done in this phase are highly valuable for the improvement of the webapp.

### 1.6.6 Review

This phase is to fix more bugs, improve the user experience, improve performance, adding more feature to improve interactive learning, or more content, such as adding more scenario for users to play. In this phase, redundancy of the scenario is also tested, to prevent loophole and repetitiveness on scenarios.

## 1.7 Significance of Project

A tool for user to learn Standard Query Language (SQL)

## **1.8 Expected Outcome**

The expected outcome of this project is to enrich the SQL learning experience with a webapp by applying the basic of SQL queries to navigate through the overwhelming amount of data in the database where user is given situation or question to identify and present specific data in the database. The experience of identifying specific data in the database also introduce the early stage of data preparation in data mining, for the usage of analysis to achieve the objective of the situation.



## **1.9 Project Outline**

Final Year Project 1 consists of 3 chapters. This chapter are to describe the brief proposal for the development of educating SQL with a webapp.

Chapter 1: Introduction introduces the development of webapp for learning SQL. This chapter discuss the background of the topic, the objectives and outcome needed to be accomplished, and the scope this topic will be covering. the methodology is also discussed as a way an approach taken to accomplish the objective of this project.

Chapter 2: Literature Review will be discussing the background in relation of this topic by evaluating existing research or development topic that will set an inspiration for innovation for this project.

Chapter 3: Methodology, Requirement Analysis and Design will be discussing in detail of the methodology used for this project, which is Agile methodology. In detail, it will address the requirement and design required to achieve the project objective.

Chapter 4: Implementation details the actual development process, explaining the coding and construction of the SQL\_DataDetective. It discusses the development phase and solutions employed during the implementation phase.

Chapter 5: Testing focuses on the rigorous evaluation of the web app, examining its functionality, usability, and performance. Web app is tested using testing tool to simulate the web app process to evaluate its performance and logics. The web app also tested with selected individual by undergoing comprehensive guided test cases to evaluate its functionality.

Chapter 6: Conclusion and Future Works provide a comprehensive summary of the project's outcomes and findings. It discusses the achievements, limitations, and potential areas for future improvements. This chapter serves as a reflection on the project's overall success and outlines potential directions for continued research and development in the realm of SQL education through web applications.

### **1.10 Summary**

In summary, this proposal provides an introduction of this project. This proposal outlines the background, problem and objective that innovate the development of the webapp. The methodology is briefly discussed to assure the development of this project is done properly to achieve its objective. The Gantt chart also provides the Final Year Project 1 timeline so the proposal for the development of the project is done as schedule to avoid delays. The expected outcome of this project is in hope that it will enrich SQL learning among beginners and motivate them to learn more by exploring through relational database.

## **CHAPTER 2: LITERATURE REVIEW**

### **2.1 Overview**

The purpose of this literature review is to summarize and evaluate the existing research on teaching and learning SQL, with a focus on interactive and engaging approaches such as game-based learning and scenario-based learning. The review begins by discussing the importance of SQL for accessing and managing data in relational databases, and the growing demand for individuals with SQL skills in various industries. The review then summarizes the research on teaching and learning SQL, including the strengths and limitations of different approaches, such as traditional methods such as lectures and textbooks, and more interactive methods such as game-based learning and scenario-based learning.

The review also discusses the potential benefits of interactive and engaging approaches to teaching SQL, such as motivating students and providing a relevant and engaging context for learning. The review then looks at specific examples of resources that use interactive and engaging approaches to teach SQL, such as the SQL Murder Mystery game and the SQLZoo website. The review concludes by summarizing the main findings of the literature on teaching and learning SQL and discussing the implications of these findings for future research and practice.

### **2.2 Introduction**

SQL (Structured Query Language) is a standard query language that is widely used to access, manipulate, and manage data in relational databases. As the demand for individuals with SQL skills grows in industries such as data science, business analysis, and IT, there is a need for effective approaches to teaching and learning SQL. This literature review aims to summarize and evaluate the existing research on teaching and learning SQL, with a focus on interactive and engaging approaches such as game-based learning and scenario-based learning.

Gros (2007) suggests that game-based learning can be an effective way to motivate students and provide a context for learning that is both relevant and engaging. Kapp (2012) argues that scenario-based learning can be an effective way to teach complex skills and concepts by providing a context for learning that is relevant and engaging. Custer (2021) reviews the existing research on teaching and learning SQL, and discusses the strengths and limitations of different approaches, including traditional methods such as lectures and textbooks, as well as more interactive methods such as game-based learning and scenario-based learning.

The SQL Murder Mystery game and SQLZoo website are examples of resources that use scenario-based puzzles and challenges to teach SQL to beginners. These resources demonstrate the potential of interactive and engaging approaches to teaching SQL. This literature review aims to evaluate the research on teaching and learning SQL, and to explore the potential of interactive and engaging approaches for improving the learning experience and outcomes for SQL beginners.

The main objective of this literature review is to:

- I. To review relevant research.
- II. Reviewing literature based on relevant research.
- III. Identify any gaps and limitations in the existing research on teaching and learning SQL.
- IV. Proposing improvement on future development based on existing platform.

### **2.3 Relevant Research**

Interactive and engaging approaches to teaching SQL, such as game-based learning and scenario-based learning, can be effective in motivating students and providing a relevant and engaging context for learning. Gros (2007) suggests that game-based learning can be an effective way to motivate students and provide a context for learning that is both relevant and engaging, while Kapp (2012) argues that scenario-based learning can be an effective way to teach complex skills and concepts by providing a context for learning that is relevant and engaging. By using scenario-based puzzles and challenges to teach SQL and data mining fundamentals through a web app, your project could potentially leverage the benefits of both game-based learning and scenario-based learning to improve the learning experience and outcomes for SQL beginners. Custer (2021) reviews the existing research on teaching and learning SQL, and discusses the strengths and limitations of different approaches, which can help inform the design and evaluation of your project. Overall, the relevant research suggests that interactive and engaging approaches to teaching SQL, such as those proposed in your project, have the potential to be effective in improving the learning experience and outcomes for SQL beginners.

### **2.4 Review of Literature**

One strength of the research on teaching and learning SQL is the diversity of approaches that have been studied. Custer (2021) reviews a range of approaches, including traditional methods such as lectures and textbooks, as well as more interactive methods such as game-based learning and scenario-based learning. The SQL Murder Mystery game and the SQLZoo website

are examples of resources that use interactive and engaging approaches to teach SQL, specifically scenario-based puzzles, and challenges. These resources demonstrate the potential of these approaches for improving the learning experience and outcomes for SQL beginners.

Another strength of the research is the emphasis on evaluating the effectiveness of different teaching and learning approaches. Custer (2021) discusses the importance of evaluating the effectiveness of different approaches to teaching SQL and reviews the existing research on this topic. The SQL Murder Mystery game and the SQLZoo website have both been widely used as teaching resources, and there is some evidence to suggest that they are effective in teaching SQL to beginners (Park & He, n.d.; SQLZoo, n.d.). However, more research is needed to understand the effectiveness of these resources more fully and to identify best practices in using them to teach SQL.

One limitation of the research on teaching and learning SQL is that the focus is primarily on traditional approaches such as lectures and textbooks. While these approaches are important and have been widely studied, there is a need for more research on interactive and engaging approaches such as game-based learning and scenario-based learning. These approaches may have the potential to improve the learning experience and outcomes for SQL beginners, but they have not been as extensively studied as traditional approaches.

Another limitation is the lack of research on the generalizability of the findings. Many of the studies on teaching and learning SQL have been conducted in specific contexts, such as in universities or countries, and it is not always clear how well the findings from these studies can be applied to other contexts. There is also a lack of research on the generalizability of the findings from the SQL Murder Mystery game and the SQLZoo website, as these resources have been primarily used in self-guided learning rather than in formal educational settings. There is a need for more research on the generalizability of the findings from different studies, and on the effectiveness of different teaching and learning approaches in different contexts.

Overall, the existing research on teaching and learning SQL has several strengths, including a diversity of approaches and an emphasis on evaluating the effectiveness of different approaches. The SQL Murder Mystery game and the SQLZoo website are examples of resources that use interactive and engaging approaches to teach SQL, and there is some evidence to suggest that they

## **2.5 Gaps and Limitation in Research**

The use of SQL to access and manipulate relational databases is an important skill for many professionals in a range of industries. As such, there is a need for effective approaches to teaching and learning SQL, particularly for beginners. While there is a significant body of research on teaching and learning SQL, there are also several gaps in this research that need to be addressed to better understand the strengths and limitations of different approaches and to identify best practices in teaching and learning SQL.

One gap in the existing research is the lack of research on the generalizability of the findings. Many studies on teaching and learning SQL have been conducted in specific contexts, such as in universities or countries, and it is not always clear how well the findings from these studies can be applied to other contexts. There is also a lack of research on the generalizability of the findings from resources such as the SQL Murder Mystery game and the SQLZoo website, which have been primarily used in self-guided learning rather than in formal educational settings. To better understand the generalizability of the findings from different studies and resources, it will be important for future research to study the effectiveness of these resources in different educational settings and in different countries.

Another gap in the existing research is the lack of research on the effectiveness of interactive and engaging approaches to teaching SQL, such as game-based learning and scenario-based learning. While there is some evidence to suggest that these approaches can be effective in improving the learning experience and outcomes for SQL beginners (Gros, 2007; Kapp, 2012), more research is needed to understand the strengths and limitations of these approaches more fully and to identify best practices in using them to teach SQL. This research could involve studying the effectiveness of these approaches in different educational settings and in different countries, to better understand the generalizability of the findings.

A third gap in the existing research is the lack of research on the effectiveness of web-based resources in teaching SQL to beginners. While the SQL Murder Mystery game and the SQLZoo website have been widely used and have demonstrated some effectiveness in teaching SQL, more research is needed to understand the strengths and limitations of these resources more fully and to identify best practices in using them to teach SQL. This research could involve studying the effectiveness of these resources in different educational settings and in different countries, to better understand the generalizability of the findings.

Finally, there is a gap in the existing research on the impact of different teaching and learning approaches on student motivation and engagement. While it is generally thought that interactive and engaging approaches such as game-based learning and scenario-based learning can improve motivation and engagement (Gros, 2007; Kapp, 2012), more research is needed to more fully understand the mechanisms through which these approaches impact motivation and engagement, and to identify best practices in using these approaches to teach SQL. This research could involve studying the impact of different teaching and learning approaches on student motivation and engagement in different educational settings and in different countries, to better understand how these approaches can be used effectively to teach SQL.

In conclusion, the existing research on teaching and learning SQL has a few strengths, including a diversity of approaches and an emphasis on evaluating the effectiveness of different approaches. However, there are also several gaps in this research that need to be addressed in order to better understand the strengths and limitations of different approaches and to identify best practices in teaching and learning SQL. These gaps include a lack of research on the generalizability of the findings, a lack of research on the effectiveness of interactive and engaging approaches, a lack of research on the effectiveness of web-based resources, and a lack of research on the impact of different teaching and learning approaches on student motivation and engagement. To address these gaps, it will be important for future research to study the effectiveness of different approaches in different educational settings and in different countries, and to understand the mechanisms more fully through which different approaches impact student motivation and engagement.

## **2.6 Improvement on the Proposed Webapp Compared to Existing platform.**

### **2.6.1 Increase the variety and complexity of puzzles and scenarios offered.**

One potential limitation of the SQL Murder Mystery is that it may not offer a wide enough range of puzzles and scenarios to fully challenge and engage learners. While the platform does provide a variety of puzzles and scenarios, some learners may find that they are able to complete all the available content relatively quickly, which could lead to a lack of motivation and engagement. To address this limitation, the SQL Murder Mystery could consider adding more varied and complex puzzles and scenarios to keep learners engaged and motivated and to better test their skills and knowledge. This could include puzzles that require learners to use more advanced SQL techniques or that involve more complex data sets and scenarios.

To add more varied content, the proposed webapp will make learners face three sequential levels. From the first to third level, the difficulty increases where the first level is mostly the basic prompt of SQL command. On the second level, is the intermediate level where learner faces complex scenario that require aggregation of SQL syntax. It has many sets of scenario testing learner ability of aggregate multiple SQL command and syntax. The final one is the advanced level. The learner will face a general scenario that its complexity can be built up the more time player spend to navigate through the database. Even though it is only scenario, through thorough planning on the story creation during development, the story face by each learner may be unique to their decisions, including the outcomes. This level tests the ability to decide to extract the correct data from the database.

#### 2.6.2 Provide more in-depth explanations and feedback.

Some learners may find it difficult to fully understand the concepts and techniques being taught through the SQL Murder Mystery, particularly if they are struggling with certain puzzles. While the platform does provide some explanations and feedback to help learners understand the material, this may not be sufficient for all learners, particularly those who are struggling or who have prior knowledge gaps. To address this limitation, the SQL Murder Mystery could consider providing more in-depth explanations and feedback to help learners better understand the material and overcome any challenges they may be facing. This could include more detailed explanations of SQL concepts and techniques, as well as more personalized feedback based on learners' performance and progress.

With the aim to enrich SQL learning experience, the webapp interface design will be in beginner friendly in nature. Filled with guide that only prompted based on the current scenario they are facing. SQL Murder Mystery also lacks the ability to highlight errors when user enters the wrong SQL command or syntax. Mistakes are an important step for a more enriching learning experience.

## 2.7 Conclusion

Overall, the existing research on teaching and learning SQL has several strengths, including a diversity of approaches and an emphasis on evaluating the effectiveness of different approaches. However, there are also several gaps in this research that need to be addressed to better understand the strengths and limitations of different approaches and to identify best practices in teaching and learning SQL. These gaps include a lack of research on the generalizability of the findings, a lack of research on the effectiveness of interactive and



engaging approaches, and a lack of research on the effectiveness of web-based resources such as SQL Murder Mystery and SQLZoo. To address these gaps, it will be important for future research to study the effectiveness of different approaches in different educational settings and in different countries, and to understand the mechanisms more fully through which different approaches impact student motivation and engagement. This research will be valuable for educators, students, and professionals seeking to improve the learning experience and outcomes for SQL beginners, and for those involved in the design and development of teaching and learning resources.

## **CHAPTER 3: METHODOLOGY**

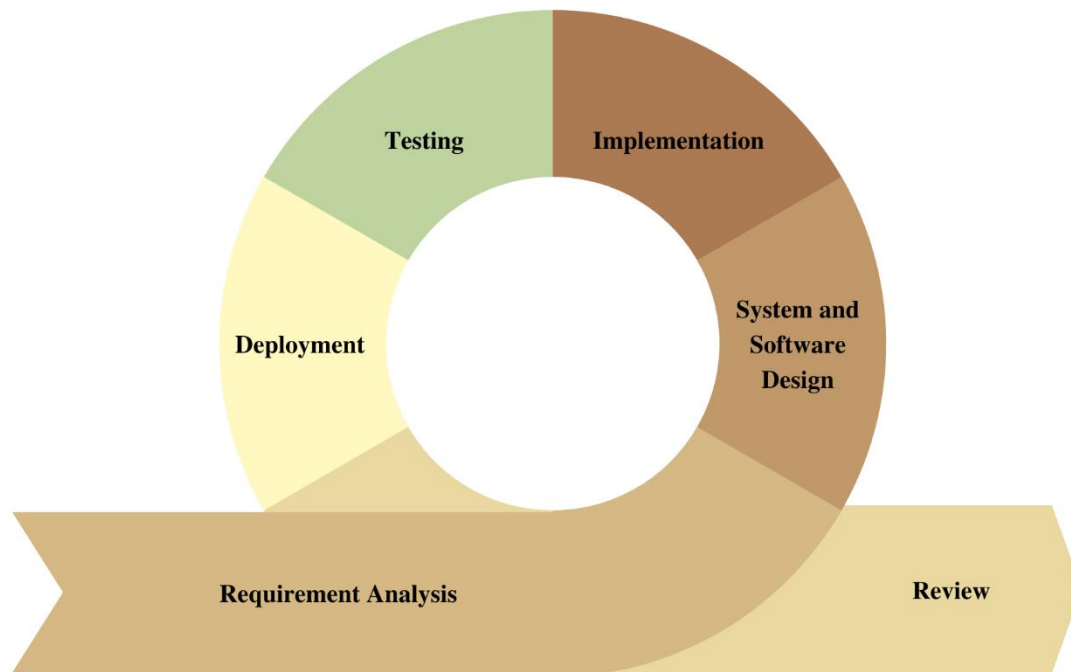
### **3.1 Introduction**

The methodology of this project aims to provide a comprehensive and interactive learning experience for beginners to learn SQL and data mining skills. With interactive exercises and scenarios, users will have the opportunity to apply their knowledge and improve their skills in navigating and manipulating large databases using SQL. The webapp interface will allow users to input and execute SQL queries, view, and interact with the results of their queries, and receive feedback on their progress. The methods and approaches used in this project include the design and development of interactive exercises and scenarios, as well as the creation of a webapp interface for users to access and engage with the learning materials. The success of this project will be evaluated based on the effectiveness of the exercises and scenarios in teaching SQL and data mining skills, as well as the engagement and satisfaction of users. The results of this project will be shared with stakeholders through the webapp and accompanying documentation and resources.

### **3.2 Overview**

The methodology of this project involves the use of a webapp and SQLite to provide an interactive learning experience for beginners to learn SQL and data mining skills. The webapp will include a range of interactive exercises and scenarios that cover important concepts and techniques in SQL and data mining and will allow users to input and execute SQL queries to solve these exercises and scenarios.

To accomplish the development of the project, a methodology approach of Agile methodology is used.



*Figure 3. 1 Agile Methodology*

In addition, this project will describe Entity Relationship Diagram (ERD) in System and Software Design phase to depict the relationships between the different entities in the webapp's database. This will help us to understand the structure of the data and how it is organized within the database.

### **3.3 Requirement Analysis**

Major requirement that needs to be considered when developing the webapp is to ensure the direction of the development of the webapp will fulfil the objective proposed and solve the problem statement stated in Chapter 1: Introduction. It also important to improved and attain features based on existing system, that is SQL Murder Mystery. This part is discussed within literature review of what limitation that can be improved for the webapp.

Technical requirements is also important, to consider the technical requirements of the web application, such as the programming languages and technologies that will be used, the hosting environment and infrastructure that will be needed, and the performance and scalability requirements of the application. This will be discussed more on this chapter in the implementation phase.

The project will consider the importance of data requirements, especially its involvement to enrich user learning experience. It is important to consider the types of data that will be stored and manipulated in the SQLite database and the ways in which this data will be used. This can

include identifying the data structures and relationships that will be needed, as well as the data input, output, and processing requirements of the application. This part will be discussed in System and Software Design phase in detail in this chapter.

### 3.4 System and Software Design

#### 3.4.1 Use Case Diagram

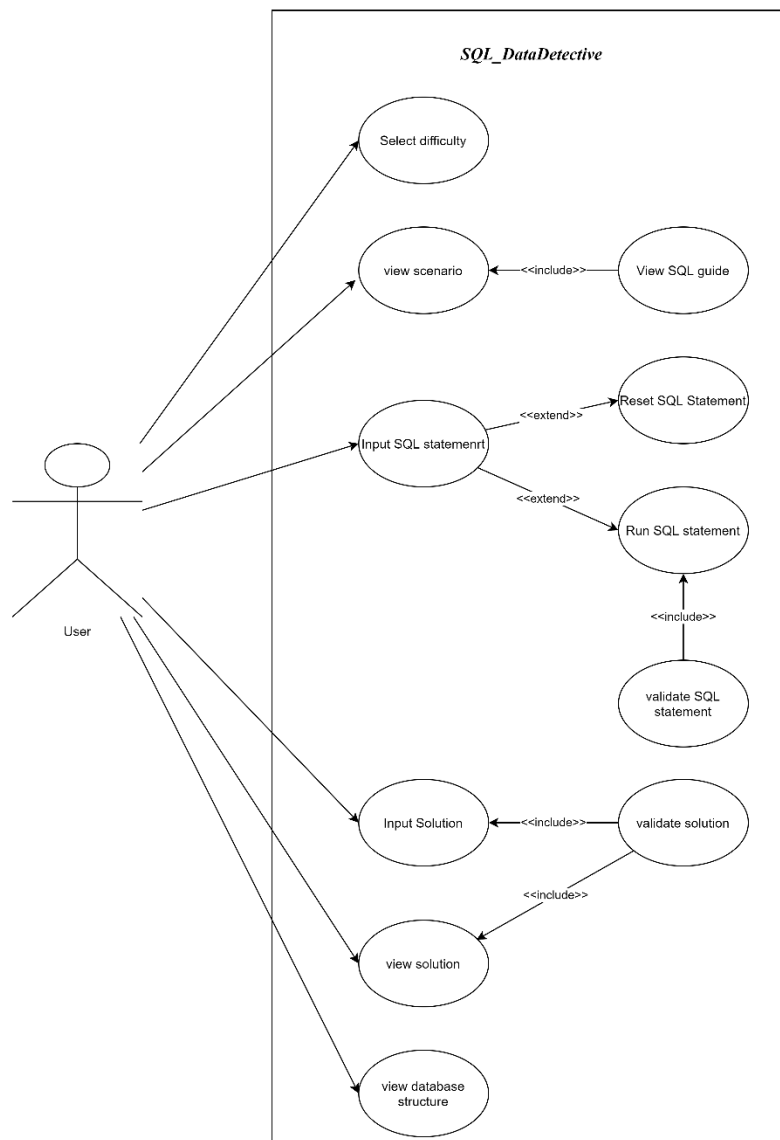


Figure 3. 2 Use case diagram

Use case diagram above describe the interaction of user with the proposed webapp, SQL\_DataDetective. The user can access the level of difficulty of learning SQL. Each difficulty follows the same interface, they can view scenario, input SQL statement, input solution and view their solution. When the user input SQL statement, the webapp will run and validate the statement given. As a result, the webapp will showcase the solution, in form of

tables that resemble the tables in database. The user is also given option to view the database structure, that is entity relational diagram in the database which is shown in 3.4.2. This help user to navigate the database easier and understand the correlation of data within the table.

### 3.4.2 Entity Relationship Diagram

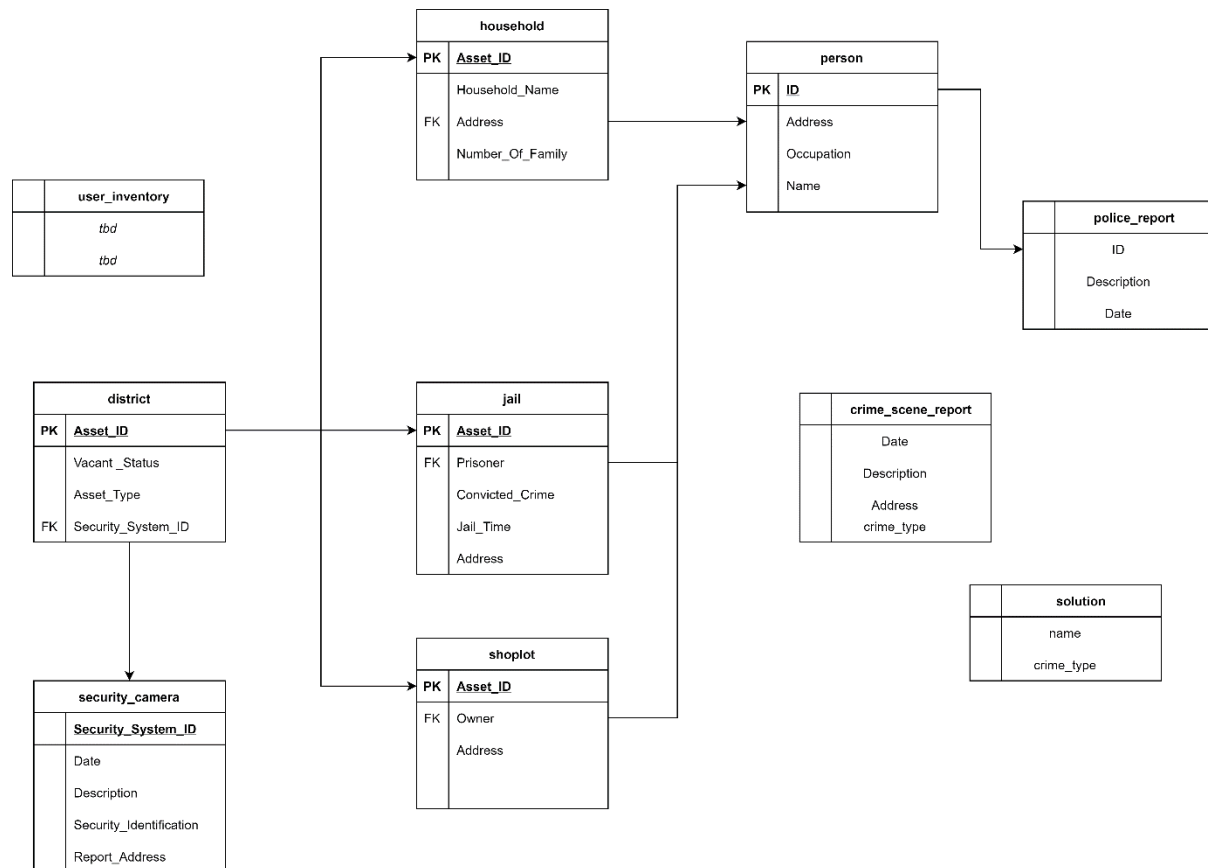


Figure 3. 3 Entity Relationship Diagram

Entity Relationship Diagram, also known as ERD, is the visualization of how relationships of each table in the database could be represented. From Figure 3, usage of primary key and foreign key heavily affects the relationship between table. This approach is used so that the user, which is also the player able to find cohesive correlation between data across within the database.

To elaborate further on the ERD flow, this ERD only works when user given scenario or question to solve. The user job is to find and extract data in either of the table. At the beginner level, the scope is only usage of one table. In the intermediate level, user start to face multiple tables at once with given scenario. This is to encourage user to user complex aggregation of SQL command and syntax. At the advanced level, user will have to use all the table to their arsenal, including *user\_inventoy* table. This table allow user to collect relevant data or item

from each chained scenario. Such data or item may be useful for their next or previous scenario if they wish to visit it back. Depend on the skill of the user and how much effort of detailed data foraging in the database they spent, the scenario may lead to unique paths or ending. This is to encourage user freedom and creativity as well making the webapp non-repetitive for a few plays.

With the right SQL command and syntax, player can access any table in the database. If they solve each scenario, they can input their solution with empty text area provided within the webapp. On advanced level may provide advantage such as clue for random sources or item rewards.

### 3.4.3 User Interface

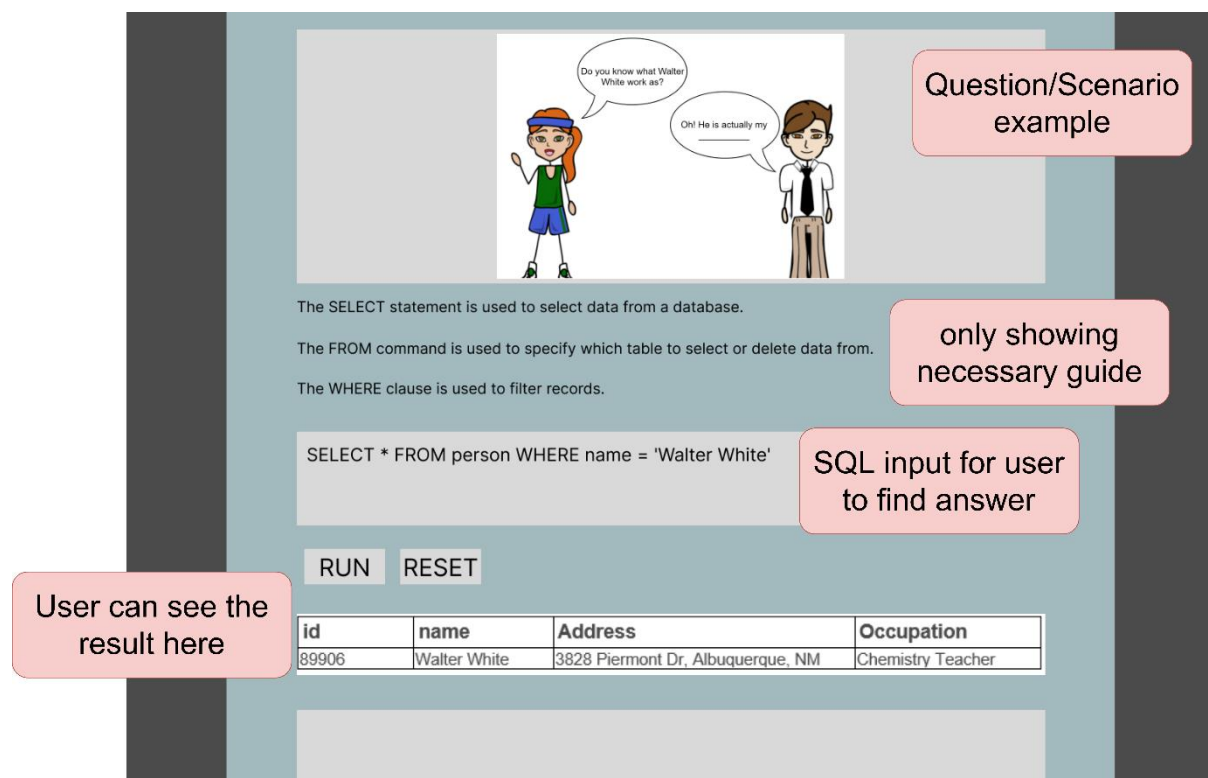


Figure 3. 4 Beginner's User Interface

User Interface above is the simple webapp for beginners' difficulty of interactive learning SQL. The question or scenario section will show what objective need to be achieved by the users. Usually for beginners are dialog with straightforward question. Depending on the scenario, the page only shows necessary guide for user to learn to get the objective. User can input their SQL command and syntax in the text area based on what they learn, then run it. Below the text area is the result of the SQL command. On this end, user need to be pay attention to what the need of the scenario to get the correct answer.

The SELECT statement is used to select data from a database.

The FROM command is used to specify which table to select or delete data from.

The WHERE clause is used to filter records.

```
SELECT * FROM person WHERE name = 'Walter White'
```

**RUN** **RESET**

| id    | name         | Address                           | Occupation        |
|-------|--------------|-----------------------------------|-------------------|
| 89906 | Walter White | 3828 Piermont Dr, Albuquerque, NM | Chemistry Teacher |

Solution:

Chemistry Teacher

**CONFIRM** **RESET** Your answer correct!

User can input the answer here

Figure 3. 5 Input Answer User Interface

In the solution text area, if user able to identify the answer, they can input it in the solution text area and confirm the answer. Using JavaScript, the webapp will confirm the answer based on pre-existing answer that related to the question/scenario.

### 3.4.3 Storyboard

To enrich user SQL learning experience at the advanced level, it is important to have a more engaging story that gives multiple scenario that can be chained to each other. this is also to improve the existing scenario learning approach applied by SQL Murder Mystery. By using similar approach, but the scenarios are more meaningful and have visuals to aid the user to find clues. Hence, storyboarding is used to describe the story path of the webapp and also visual representation of the story and the prototype of the webapp interface. On every problem solved, new related scenario may come up to reward the player that they have made substantial progress in the storyline.

However, this is just the introduction of the storyline, at this point, player are given clue that can be used to explore the robust database. Each choice of clue may affect user's progress, majorly or minorly. Of course, the user can use every single clue to solve every possible scenario, that can lead to more advantage for the user at the end.

This is how the storyline flexibility can be done on the SQL\_DataDetective webapp. This also encourage replayability that curious user can try different approach to solve each scenario.

### **3.5 Implementation**

In the implementation segment, this chapter will explain the tools used to develop the webapp. Tool that will be needed within the implementation phase are database management system, programming language, code editor and web server.

#### **3.5.1 Database Management System**

SQLite is a popular database management system that is widely used for its simplicity and ease of use. It is a lightweight database that does not require a separate server to run, making it well-suited for use in web applications and other types of software.

In this project, SQLite will be used as a development tool to store and manage data within the webapp. The webapp will include a range of interactive exercises and scenarios that teach SQL and data mining skills to users, and SQLite will be used to store and manage the data used in these exercises and scenarios.

To use SQLite as a development tool, the following steps will be followed:

- I. Design and create the database schema using SQLite. This will involve defining the entities and attributes that will be stored in the database, as well as the relationships between these entities.
- II. Populate the database with data. This will involve inserting rows of data into the various tables in the database using SQL commands.
- III. Connect the webapp to the SQLite database. This will involve writing code to establish a connection to the database and execute SQL queries to retrieve and manipulate data.
- IV. Use SQLite to store and manage data within the webapp. This will involve using SQL commands to insert, update, delete, and retrieve data from the database as needed.

Overall, SQLite will be used as a development tool to store and manage data within the webapp and will be used in conjunction with a programming language and other tools to build the webapp and interactive exercises and scenarios. By using SQLite as a development tool, the webapp will be able to store and manage data in a structured and organized manner and allow users to access and manipulate this data using SQL queries.



### 3.5.2 Programming and Scripting Language

For this project, the chosen programming and scripting languages encompass HTML, CSS, SQL, PHP, and JavaScript. The project focuses on utilizing the robust capabilities of these languages to build the web application. JavaScript, in particular, will play a crucial role in directing user actions based on predefined scenarios.

The development process involves the following steps:

- I. Determine the requirements and goals of the web application. Identify the features and functionality that the application should have, as well as any constraints or limitations influencing the development process.
- II. Choose appropriate compilers and libraries or frameworks for building the web application. Evaluate available options and select tools that align with the project's requirements and goals.
- III. Write code to build the web application, implementing the user interface, business logic, and database connectivity using the selected languages and frameworks.
- IV. Test and debug the web application by running it and addressing any errors or issues that arise during testing.
- V. Deploy the web application by hosting it on a web server or another hosting solution, making it accessible to users over the internet.
- VI. Maintain and update the web application by periodically reviewing and updating the code and functionality to ensure it meets user and stakeholder needs.

The development process follows the Agile Model described in the brief methodology in Chapter 1: Introduction. The chosen code editor for this project is Sublime Text, known for its speed, ease of use, and customization options. It supports a wide range of programming languages, including HTML, CSS, SQL, PHP, and JavaScript, and provides features for editing, debugging, and testing code.

### 3.5.3 Web Server

While during the development of this project, no hosting is needed since SQLite required no hosting. However, to make it able to use by other user or publicly, Apache open source we

server will be used for hosting. Reason why Apache is chosen for this project is due Compatibility with a wide range of programming languages and technologies: Apache is compatible with a wide range of programming languages and technologies, including C, PHP, Python, and Ruby. This makes it a good choice for web applications that use a variety of technologies and frameworks.

### **3.6 System Testing**

After completing the implementation phase of the development, testing will be done. Testing will be done automatically using Selenium Testing Tool. This phase is a mix of both testing and maintenance phase as during testing. Here are a few ways in which Selenium can support the maintenance phase of the SDLC:

1. Regression testing: During the maintenance phase, it is important to ensure that changes or updates to the web application do not introduce new bugs or regressions. Selenium can be used to automate regression testing by running a suite of tests that verify the functionality of the web application before and after changes are made.
2. Performance testing: Selenium can also be used to test the performance of the web application during the maintenance phase. This can include testing the application's response times, capacity, and scalability under different load conditions.
3. Security testing: Selenium can be used to test the security of the web application during the maintenance phase. This can include testing the application's authentication and access control mechanisms, as well as its handling of sensitive data and its ability to resist security threats.
4. User acceptance testing: Selenium can be used to automate user acceptance testing during the maintenance phase. This can include simulating user actions and interactions with the web application and verifying that the application meets the needs and expectations of its users.

Overall, Selenium can be an effective tool for testing and maintaining a web application that uses SQLite during the maintenance phase of the SDLC. Its ability to automate testing and verify the functionality, performance, security, and user acceptance of the application can help ensure that the application continues to meet the needs of its users and remain stable and reliable over time.

### **3.7 Deployment and Review**

In this phase, the complete development of the webapp will be released among selected users through purposive sampling. Purposive sampling is reaching to a target audience that match a certain criterion which are mostly students from the age of 15 – 25, which should cover for secondary school level students, university foundation and degree level students, and possibly fresh graduate. The purpose of targeting user with such criteria is to measure the efficiency of the learning experience done through the webapp by they are encouraged to gives critics, comment or feedback to their experience using the webapp. This step is important for the review phase, to improve the webapp fluidity on its user interface as well fixing the errors that are experienced by the user when using SQL\_DataDetective.

The expected outcome for this phase is user are parallel to the objective of this project, that is introducing SQL, enrich student learning experience through planned curriculum and user able to understand, assess and evaluate data extracted from the webapp.

### **3.7 Conclusion**

Overall, this chapter provide detailed methodology on how to develop the interactive SQL learning using webapp. This chapter explain the requirement need to be analyse before the development. This chapter also provide tools that will be use during implementation phase, followed up by system testing. Agile models allow development to be done rapidly towards the completion of the webapp, meanwhile fixing bugs and error, as well as improving the system.

## **CHAPTER 4: IMPLEMENTATION**

### **4.1 Introduction**

This chapter details the comprehensive implementation of SQL\_DataDetective, employing the Agile Methodology to construct a dynamic web application from frontend to backend. The discussion encompasses the system development environment, outlining the frontend and backend construction processes essential for creating an intuitive and effective SQL learning platform.

### **4.2 System Development**

The system development encompassed a diverse set of technologies. For frontend design, HTML and CSS were employed to structure and style webpages, while JavaScript handled logic, SQL requests, and database connections. The database management system utilized was SQLite, and SQLiteStudio aided in the creation and development of the database. On the server side, Node.js served as the hosting environment, and Express, a web application framework for Node.js, streamlined the handling of HTTP requests, route definition, and middleware integration. This comprehensive tech stack, inclusive of Express, contributed to the creation of a dynamic and responsive web application with an organized frontend and a robust server-side architecture.

### **4.3 SQL\_DataDetective Frontend Implementation**

#### **4.3.1 User Interface**

A user-friendly frontend is pivotal for an optimal user experience. HTML and CSS were instrumental in crafting the webpages, with CSS stylization playing a crucial role in creating a visually appealing and comfortable user interface.

##### *4.3.1.1 Homepage*

The homepage serves as the starting point, offering users a welcoming introduction to SQL\_DataDetective. It briefly explains the difficulty selection of each level, providing users with a clear understanding of the learning path.

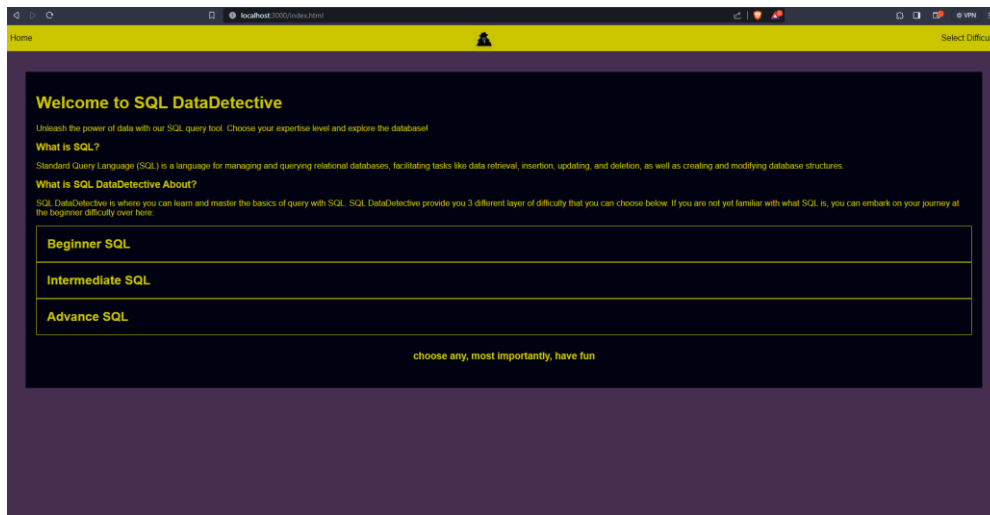


Figure 4. 1 Homepage Interface

From the homepage interface, user can choose to start their SQL learning experience by choosing any of the following difficulties listed. User can choose to learn as a beginner, take up slightly more complex challenges on intermediate mode, or explore the narrative-based SQL puzzle. Upon hovering mouse on either of the difficulty selection, user is briefed about the difficulty as shown below: to the right is the CSS that enable the description of the button to transition.



Figure 4. 2 Difficulty Selection example

```

.fancybutton {
  position: relative;
  padding: 20px;
  border: 1px solid #CCC600;
  background-color: #000211; /* Updated color */
  color: #000211; /* Updated color */
  cursor: pointer;
  overflow: hidden;
}

.fancybutton h2 {
  text-align: left;
  margin: 0;
  transition: opacity 0.3s ease-in-out;
  color: #CCC600; /* Updated color */
}

.fancybutton-description {
  position: absolute;
  top: 0;
  left: 0;
  width: calc(100% - 20px);
  height: calc(100% - 20px);
  padding: 20px;
  background-color: #CCC600; /* Updated color */
  opacity: 0;
  transition: opacity 0.3s ease-in-out;
  word-wrap: pre-line;
  overflow: hidden;
  word-wrap: break-word;
}

.fancybutton:hover h2 {
  opacity: 0;
}

.fancybutton:hover .fancybutton-description {
  opacity: 1;
}

```

Figure 4. 3 CSS that allow the description to transition on the button

```

<body>
  <div class="ribbon">
    <div class="home-tab" onclick="goHome()">Home</div>
    <div class="web-icon">
      
    </div>
    <div class="dropdown">
      <div class="dropbtn">Select Difficulty</div>
      <div class="dropdown-content">
        <a href="beginner.html">Beginner</a>
        <a href="intermediate.html">Intermediate</a>
        <a href="story.html">Advanced</a>
      </div>
    </div>
  </div>

  <div id="content" class="container">
    <div class="homepage-content">
      <div>
        <h1>Welcome to SQL DataDetective</h1>
        <p>Unleash the power of data with our SQL query tool. Choose your expertise level and explore the database!</p>
      </div>
      <div>
        <h3>What is SQL?</h3>
        <p>Standard Query Language (SQL) is a language for managing and querying relational databases, facilitating tasks like data retrieval, insertion, updating, and deletion, as well as creating and modifying database structures.</p>
      </div>
      <div>
        <h3>What is SQL DataDetective About?</h3>
        <p>SQL DataDetective is where you can learn and master the basics of query with SQL. SQL DataDetective provide you 3 different layer of difficulty that you can choose below.
        <p>If you are not yet familiar with what SQL is, you can embark on your journey at the beginner difficulty over here:</p>
        <div class="fancybutton" onclick="location.href='beginner.html'">
          <h2>Beginner SQL</h2>
          <div class="fancybutton-description">Learn the basic of SQL through guided walkthrough</div>
        </div>
        <div class="fancybutton" onclick="location.href='intermediate.html'">
          <h2>Intermediate SQL</h2>
          <div class="fancybutton-description">Got the basics right? challenge yourself with more complex SQL queries</div>
        </div>
        <div class="fancybutton" onclick="location.href='beginner.html'">
          <h2>Advance SQL</h2>
          <div class="fancybutton-description">want more challenge but exploring database seems broing? try the SQL-Based puzzle narrative</div>
        </div>
      </div>
      <div>
        <h3 style="text-align: center; padding: 30px;">choose any, most importantly, have fun</h3>
      </div>
    </div>
  </div>
  <script src="script.js"></script>
</body>

```

Figure 4. 4 Body of the HTML Structure

#### 4.3.1.2 Building Up the SQL Puzzle Experience

The SQL puzzle experience is divided into three levels: Beginner, Intermediate, and Advanced. Each level is represented by a dedicated page. The Beginner level provides SQL tutorials, while the Intermediate and Advanced levels present progressively complex puzzles, requiring users to apply their SQL knowledge. This section explains the basic of the web page, based on beginners' mode. More details on advanced mode, which is similar to intermediate mode will be explained in 4.3.1.3 Quality of Life Addition.

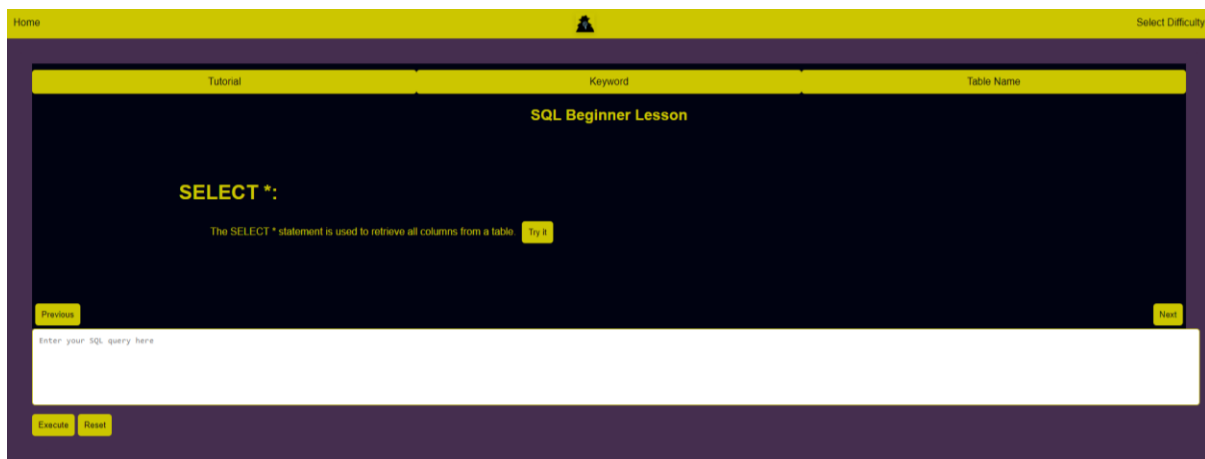


Figure 4. 5 Beginner SQL\_DataDetective Interface

In beginner mode, users are given tutorials to help user understand the basics of SQL. User can slide to next or previous slide to learn more about SQL queries. User can type their own queries or use 'Try it' button to generate pre-made queries onto the text area.

```
38
39
40 <div class="slide" style="background-color: #000211;">
41   <h1> SELECT statement:</h1>
42   <p>The SELECT statement is used to retrieve data from one or more tables. <
43     button onclick="appendCommand('SELECT First_Name, Last_Name FROM Prisoners')'"
44   >Try it</button></p>
45 </div>
46
47 <div class="slide" style="background-color: #000211;">
48   <h1> SELECT *:</h1>
49   <p>The SELECT * statement is used to retrieve all columns from a table. <
50     button onclick="appendCommand('SELECT * FROM Cases')">Try it</button></p>
51 </div>
```

Figure 4. 6 HTML for tutorial slide content example

```

8 function appendCommand(command) {
9   const sqlQueryTextarea = document.getElementById('sqlQuery');
10
11   // Append the selected command to the existing content in the textarea
12   const currentContent = sqlQueryTextarea.value.trim();
13   const separator = currentContent.length > 0 ? ' ' : ''; // Add a space separator if needed
14   sqlQueryTextarea.value = currentContent + separator + command;
15 }

```

Figure 4. 7 function for automating typing in text area

The JavaScript function above is responsible for filling the text area if user chose to use the provided button. This concept is used in filling the text area using ‘Try it’ button, which also similarly used in figure 10 and figure 11, in Quality-of-Life Addition section.

#### 4.3.1.3 Quality of Life Addition

To enhance user experience, a minimalist one-page design was implemented throughout the puzzle-solving process. Tabs were introduced to allow users to seamlessly navigate between the Mission, Keyword, Table Name, and Submit Answer sections.

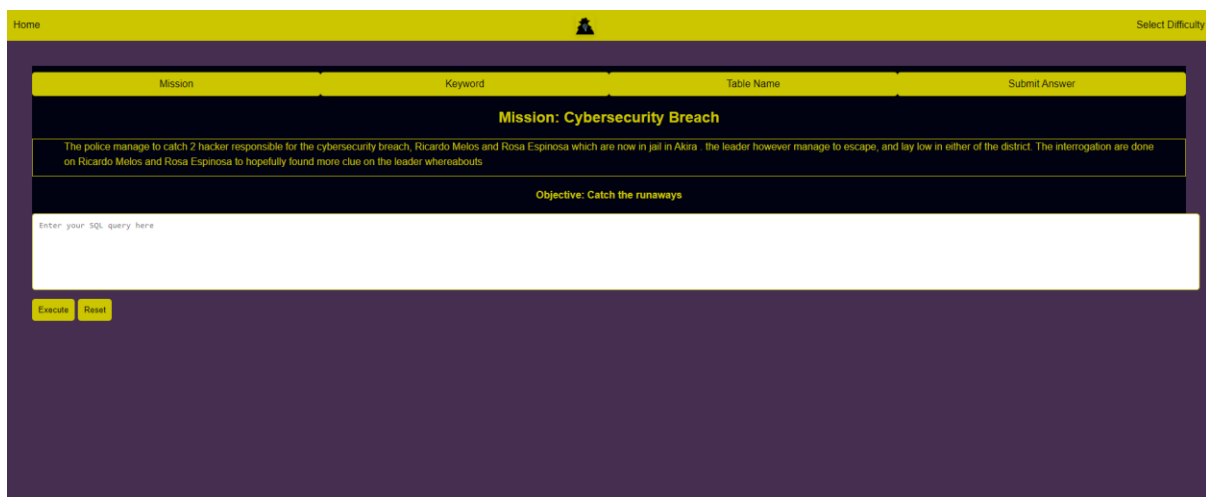


Figure 4. 8 A minimalist interface

Despite taking little space, the interface provides sufficient information for player, by default, user is greeted with Mission Tab open that gives the user the description of the narrative or tutorial is users are in beginners or intermediate SQL page.

```

1 showSection('story-section');
2

```

Figure 4. 9 simple JavaScript allows mission tab open on default

From this page, user allow to use their query by typing, reset or execute the query written on the text area. If user having trouble memorising every single SQL command or tables, the tab also provide keywords and table name user can use to input command by just a click of a button.





Figure 4. 10 Entering SQL command from Keywords tab



Figure 4. 11 Entering Table Names from the Tab

When user manage to solve the puzzle or the narrative, user can submit their answer though the ‘Submit Answer’ Tab. User can input more than one answer before submitting, as some puzzle may require more than one answer.

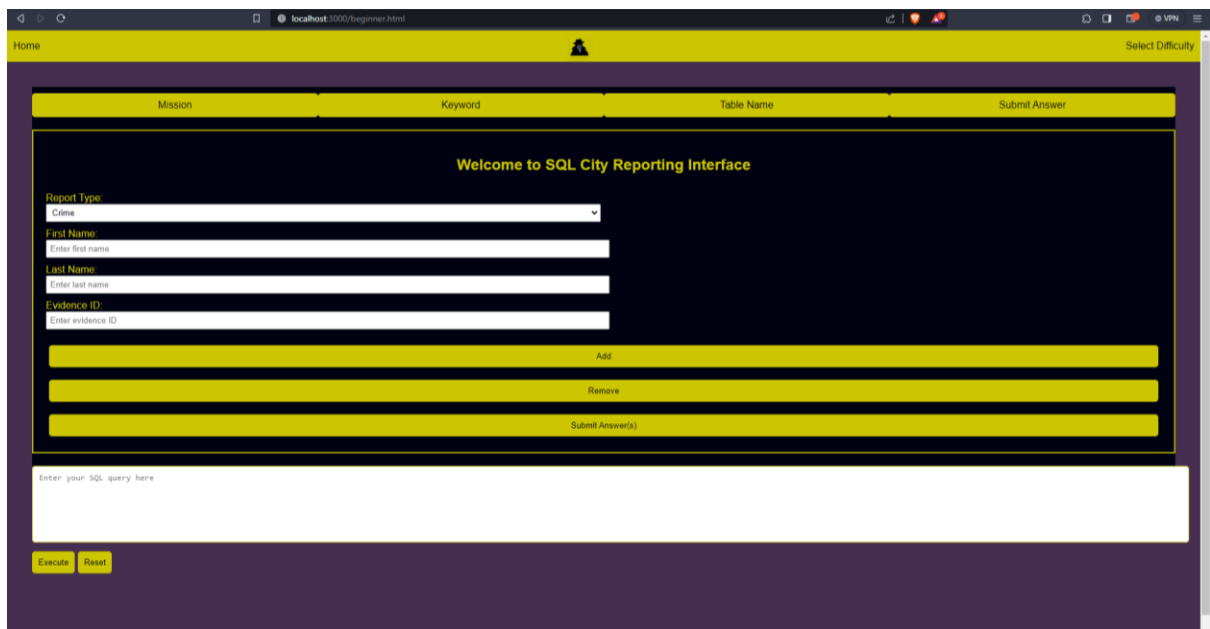


Figure 4. 12 Submit Answer Interface

The validation of answer is run by JavaScript instead of cross referencing from the table for security reason so user cannot have access to the answer and cheat the narrative, and from the developer side, it encourages more flexibility and creativity when pushing the narrative.

## 4.4 SQL\_DataDetective Backend Implementation

### 4.4.1 Logic Management

JavaScript was employed to handle logic throughout the backend development. This included managing user interactions, validating inputs, and executing SQL queries. The Significance of JavaScript started when user execute their query. On the client side, JavaScript code, particularly the **executeQuery** function, facilitates user interaction by sending SQL queries to the server upon specified actions. The server, built with Node.js, Express, and SQLite, listens for requests at the **/query** endpoint. It establishes a connection to an SQLite database, executes incoming SQL queries, and responds with the results or an error message. This collaboration allows users to input SQL queries on the webpage, triggering server-side processing and displaying outcomes back to the user, effectively creating a functional and interactive web-based database interface.

```
65
66 function executeQuery() {
67     const sqlQuery = document.getElementById('sqlQuery').value;
68
69     fetch(`/query?sql=${encodeURIComponent(sqlQuery)}`)
70     .then(response => {
71         if (!response.ok) {
72             throw new Error(`Server error: ${response.status} - ${response.statusText}`);
73         }
74         return response.json();
75     })
76     .then(data => {
77         if (data.error) {
78             displayErrorMessage(data.error);
79         } else {
80             displayQueryResults(data);
81         }
82     })
83     .catch(error => {
84         console.error('Error:', error);
85         displayErrorMessage('Something wrong with your SQL Statement :(');
86     });
87 }
88
89 function displayErrorMessage(message) {
90     alert(`Error: ${message}`);
91 }
92
```

Figure 4. 13 JavaScript responsible for sending a user-entered SQL query to a server

```

// Handle SQL query requests
app.get('/query', (req, res) => {
  const sqlQuery = req.query.sql;

  // Execute the SQL query
  db.all(sqlQuery, [], (err, rows) => {
    if (err) {
      res.status(400).json({ error: err.message });
      return;
    }

    // Send the query result as JSON response
    res.json(rows);
  });
});

```

Figure 4. 14 server-side implementation using Node.js with Express and SQLite

#### 4.4.2 Building the Database

The database was constructed using SQLiteStudio, a robust tool for designing and managing SQLite databases. This facilitated the creation of a structured database to support SQL\_DataDetective's functionalities. Below are the SQLiteStudio interface, and Detective () is the database name used for this SQL\_DataDetective

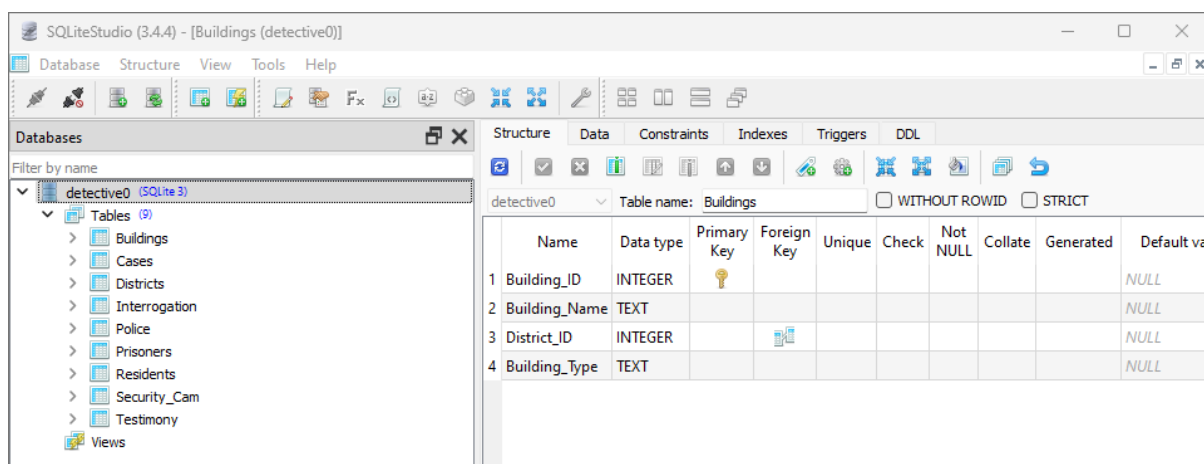


Figure 4. 15 SQLiteStudio Interface

The Database is built with sample data that are generated with the help of Chat GPT 3.5 to populate the data, as SQL\_DataDetective need to simulate a small-scale district life that are filled with residents, prisoners, buildings and more. The sample data are later modified to fits the narrative, especially on advanced SQL page in SQL\_DataDetective.

| Person_ID | First_Name | Last_Name | Building_ID | Occupation        |
|-----------|------------|-----------|-------------|-------------------|
| 1         | John       | Silva     | 25          | Software Engineer |
| 2         | Jane       | Smith     | 47          | Student           |
| 3         | David      | Johnson   | 47          | Police Officer    |
| 4         | Emily      | Davis     | 47          | Doctor            |
| 5         | Michael    | Wilson    | 47          | Bartender         |
| 6         | Sarah      | Taylor    | 47          | Lawyer            |
| 7         | James      | Brown     | 47          | Teacher           |
| 8         | Rogue      | Amenda    | 47          | Unemployed        |
| 9         | Robert     | Lee       | 47          | Detective         |
| 10        | Jennifer   | Harris    | 47          | Journalist        |
| 11        | William    | Clark     | 1           | Chef              |
| 12        | Mary       | Lopez     | 2           | Police Officer    |
| 13        | Charles    | Young     | 3           | Artist            |
| 14        | Karen      | Hall      | 4           | Nurse             |
| 15        | David      | Martin    | 6           | Unemployed        |
| 16        | Patricia   | Gonzalez  | 6           | Engineer          |
| 17        | Joseph     | Lewis     | 7           | Bartender         |
| 18        | Nancy      | Rodriguez | 8           | Student           |
| 19        | Thomas     | Hernandez | 9           | Taxi Driver       |
| 20        | Betty      | Moore     | 10          | Journalist        |
| 21        | Kevin      | King      | 11          | Doctor            |
| 22        | Dorothy    | Perez     | 12          | Unemployed        |
| 23        | Richard    | Wright    | 13          | Detective         |
| 24        | Susan      | Turner    | 14          | Teacher           |
| 25        | Kenneth    | Scott     | 15          | Lawyer            |
| 26        | Lucy       | Kushina   | 10          | Artist            |
| 27        | Mark       | Baker     | 18          | Police Officer    |
| 28        | Donna      | Adams     | 19          | Chef              |
| 29        | George     | Nelson    | 20          | Nurse             |
| 30        | Debra      | Murphy    | 21          | Engineer          |
| 31        | Edward     | White     | 22          | Suspect           |
| 32        | Margaret   | Monst     | 23          | Bartender         |
| 33        | Brian      | Williams  | 24          | Software Engineer |
| 34        | Lisa       | Hall      | 25          | Teacher           |
| 35        | Aleks      | Torres    | 26          | Unemployed        |
| 36        | Lori       | Hill      | 27          | Journalist        |
| 37        | Anthony    | Sanchez   | 28          | Doctor            |
| 38        | Sandra     | Clark     | 29          | Artist            |

Figure 4. 16 100 Sample data of residents in SQL City

For SQL\_DataDetective to be able to access the detective () database, Node.js and Express script serves as the backbone for connecting a website to an SQLite database. The code sets up a server, hosts static files, and establishes a read-only connection to the SQLite database using the **sqlite3** module. The '/query' endpoint handles incoming SQL queries, executes them on the database, and responds with the results or error messages in JSON format. This setup streamlines the interaction between the website's frontend, coded in HTML, CSS, and JavaScript, and the backend server, allowing seamless integration of database functionality. With this infrastructure, users can input SQL queries through the website, triggering server-side execution and enabling dynamic data retrieval for a responsive and interactive web experience.

```

1  const express = require('express');
2  const path = require('path');
3  const sqlite3 = require('sqlite3').verbose();
4
5  const app = express();
6  const port = 3000;
7
8  const dbPath = path.join(__dirname, 'detective.db');
9
10
11  const db = new sqlite3.Database(dbPath, sqlite3.OPEN_READONLY, (err) => {
12    if (err) {
13      console.error('Database opening error:', err.message);
14    } else {
15      console.log('Connected to the database.');

```

Figure 4. 17 JavaScript that enable connection between database and webpage

## 4.5 Conclusion

The implementation of SQL\_DataDetective successfully integrated frontend and backend technologies to create an interactive and educational platform for learning SQL. The application's architecture, from the user interface to the database management, ensures a seamless and engaging experience for users at varying skill levels. The use of Agile Methodology facilitated an iterative and adaptive development process, resulting in a well-rounded and user-centric product.

## CHAPTER 5: TESTING

### 5.1 Introduction

This chapter emphasizes on the testing methods on SQL\_DataDetective webapp. The testing will focus on the webapp functionality and response time. A specific functionality has been picked to conduct testing which will be included in this chapter.

#### 5.1.1 Functional Testing

These functions are selected to be tested for the SQL\_DataDetective webapp functional requirements assessment. These functions are the main features of the web app to achieve the objective of the project.

Table 5.1 functionalities to be tested on

| No. | Functionalities            | Pass Rate |                  |
|-----|----------------------------|-----------|------------------|
|     |                            | Pass/Fail | Total Test Cases |
| 1   | Selecting difficulty level | 3         | /3               |
| 2   | tutorial                   | 2         | /2               |
| 3   | Test Story selection       | 2         | /2               |
| 4   | Test SQL execution         | 1         | /1               |
| 5   | Select SQL statements      | 1         | /1               |
| 6   | Select Table Name          | 1         | /1               |
| 7   | Submit Answer              | 2         | /2               |

To test these functionalities, 7 testers are given to test each of the functionalities. Each tester will give instructions according to the test case table in table 2. They will give the grade of their testing, pass or fail based on the expected results provided.

In addition to that, the web app will also undergo logical testing using test scripts. The test scripts are possible with the deployment of Selenium testing tools. To install selenium, simply Install the selenium-webdriver package using npm, on the command prompt or terminal where the directory of the project is.

```
C:\FYP>npm install selenium-webdriver

added 14 packages, and audited 179 packages in 3s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 9.8.1 -> 10.2.5
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.2.5
npm notice Run npm install -g npm@10.2.5 to update!
npm notice
```

Figure 5. 1 Installing Selenium Package

When the Selenium package is ready, it will be hosted on a different host as well as an exact copy of the SQL Data Detective database for testing as well as backups. For hosting, while using the same structure as previous server.js, a minor change is added for the testing. The changes are as follows.

#### I. Database configuration for testing

```
const dbPath = path.join(__dirname, 'test.db'); // Use a different database for testing
const db = new sqlite3.Database(dbPath, sqlite3.OPEN_READWRITE | sqlite3.OPEN_CREATE, (err) => { /* ... */ });
```

Figure 5. 2 database configuration for testing

The database file path is changed to test.db, which is the copy of detective.db. the database is also opened in read-write mode for testing purposes.

#### II. Tests sever configuration.

```
const port = 3001; // Use a different port for testing
const db = new sqlite3.Database(dbPath, sqlite3.OPEN_READWRITE | sqlite3.OPEN_CREATE, (err) => { /* ... */ });
```

Figure 5. 3 test server configuration

The port for hosting is change from 3000 to 3001 to avoid conflicts when the server running during test case testing.

#### III. Export for testing

```
module.exports = app; // Export the app for testing
```

Figure 5. 4 line for exporting the app for testing.

Added new line code at the end that export the 'app' object which allows it to be used in testing scripts.

### 5.1.2 Test Cases for Functional Testing

Table 5.2 Test Cases of Functionalities

| Test Case ID | Description                             | Procedure   | Expected Results   | Pass/Fail |
|--------------|---|---|--|-----------|
| TC01         | Selecting Beginner difficulty level     | I. Go to web homepage.<br>II. Hover cursor on difficulty selection<br>III. Click on beginner difficulty   | User will be redirected to Beginner SQL page that shows SQL tutorials  | pass      |
| TC02         | Selecting Intermediate difficulty level | I. Go to web homepage.<br>II. Hover cursor on difficulty selection<br>III. Click on intermediate difficulty   | User will be redirected to intermediate SQL page that shows puzzle question  | pass      |
| TC03         | Selecting advanced difficulty level     | I. Go to web homepage.<br>II. Hover cursor on difficulty selection<br>III. Click on advanced difficulty   | User will be redirected to a page where they can select a mission/narrative type   | pass      |
| TC04         | Tutorial slides                         | I. Go to website, click on beginners' difficulty selection.<br>II. On the tutorial tab, click "NEXT" on right side for next tutorial.<br>III. Click "PREV" on the left for previous tutorial. | The tutorial tab content will slide to next slide with new tutorial content, previous slide with previous tutorial content | pass      |
| TC05         | Tutorial walkthrough                    | I. on the beginner's page, click on "tutorial" tab.   | The clicking "try it", the text area should be filled with   | pass      |



|      |  |  |  |      |
|------|--|--|--|------|
|      |  | II. Click “try it” on the content of tutorial page   | corresponding SQL statements from the tutorial   |      |
| TC06 | Test Story selection                   | I. Go to website, click on advanced beginner’s difficulty selection.<br>II. Hover to either the “Mission type”<br>III. Click on the “Mission Type”   | User will be redirected to the advanced SQL page that correspond to the Mission type selected                      | pass |
| TC07 | Test SQL execution                     | I. On either beginner, intermediate or advanced SQL page, type “SELECT * FROM Residents”.<br>II. Click button “execute”  | User will be shown a table named “Residents” under the text area   | Pass |
| TC08 | Selecting and Resetting SQL statements | I. On the website, click on advanced beginner’s difficulty page.<br>II. On the page, click on “Keywords” tab.<br>III. Click “SELECT”, “*” and “FROM” button in the keywords content.<br>IV. Click “Reset” button | The textarea should be fill with text accordingly to the clicked Keywords that can be reset to empty the text area | pass |
| TC09 | Select Table Name                      | I. On the website, click on advanced beginner’s difficulty page.   | The text area should display the text correspond to the button input   | Pass |

|      |                            |   |  |      |
|------|----------------------------|---|--|------|
|      |                            | II. On the page, click on “Table Name” tab.<br>III. Click button name “Buildings”   |  |      |
| TC10 | Submit correct Answer      | I. On the advanced SQL page, click on “Submit Answer” tab.<br>II. Fill the form as follows <ul style="list-style-type: none"> <li>• Report Type: Crime</li> <li>• First name: Aleks</li> <li>• Last name: Tanaka</li> <li>• Evidence ID: 20</li> </ul> III. Click “Submit Answer(s)” button | the user will be greeted with a pop out window that display a text-based narrative     | Pass |
| TC11 | Submit wrong answer        | I. On the advanced SQL page, click on “Submit Answer” tab.<br>II. Fill the form with wrong/random input.<br>III. Click “Submit Answer(s)” button  | The user will get an alert for the wrong answer and the “Submit Answer” tab will reset | pass |
| TC12 | Add more answer submission | I. On “Submit Answer” tab, click “Add”  | The “submit answer” tab will add additional form                                       | pass |

These 12 test cases are tested to verify the functioning of the web app features. The procedure described that process that can be done to achieve the feature functionalities, which

are also the user-end experience will face when using the web app. These processes will yield and expecting results, which all test case met all the expectation. The test case purpose is also to oversee the implementation of the features are done correctly, as well as a note for further improvement on the web app for future works.

## **5.2 Summary**

In conclusion all the 7 functionalities are tested and met the expected results based on the 12 test cases are done. Since the functionalities focus on user-end experience, each of the test cases are given steps as show in the table to the tester and passed the results. In addition to make a more detailed testing, especially from the backend, Selenium and test scripts are done to flesh out more it flaws for improvement. The testing overall is a success to deliver the functionality of the web app.

## CHAPTER 6: CONCLUSION AND FUTURE WORKS

### 6.1 Introduction

This chapter will focus on the objectives that is achieved through the development of SQL\_DataDetective. The chapter also explore the limitations of the product and potential improvements that for future works involving the product.

### 6.2 Project Achievement

*Table 3 Objectives and Achievement*

| Objectives   | Achievements   |
|--|--|
| To develop a web application with an interactive interface for introducing SQL queries   | The objective was achieved by building a web app that focus on introducing SQL to user, as well as mastering it by solving narrative driven puzzle   |
| To design a learning curriculum that starts from basic and gradually increase in complexity of learning SQ   | The objective is achieved by designing 3 different difficulty level, beginner, intermediate and advanced SQL challenges that user can choose to face   |
| To evaluate users' ability to identify and extract data by testing their proficiency in applying the SQL learned   | User's ability is tested on their ability to write SQL and execute it properly based on their learning on beginner's level SQL   |
| To assess the users' understanding of data quality by measuring their ability to identify relevant data relationships within a relational database through a series of data analysis simulations | The objective is achieved by causing the information they extracted from the database to be important in their problem-solving session, which is prominent feature that drives Advance's level SQL |

### 6.3 Project limitations

The limitation SQL\_DataDetective: Learn Standard Query Language (SQL) With a Webapp is as below:

- I. The database used for SQL\_DataDetective is static, hence the replay ability is limited once user understand the full relation of the database.

- II. The interface lacks appeal in graphics and animation that can push the potential more for narrative driven puzzle.
- III. Throughout the playthrough, user progress is not saved, hence reset or restarting the web app page will reset users progress along the narrative.
- IV. Usage of SQL commands is limited for safety reasons, such as DELETE, TRUNCATE and DROP commands are inaccessible on user-end to protect the database.

## **6.4 Future works**

The possibility of future works are as follows:

- I. Adding more narrative to increase replay ability.
- II. Adding a save by adding checkpoints for progression in the narrative.
- III. With addition of checkpoints, addition of slides for graphics to dictate the narrative is possible.
- IV. Take advantage of JavaScript ability to improve the puzzle and narrative experience.
- V. Improve user interface to make aesthetically pleasing and have more prominent theme.

## **6.5 Summary**

SQL\_DataDetective successfully achieved its learning objectives by building a web app with interactive puzzles and a multi-level curriculum. However, limitations like static database, unappealing visuals, and unsaved progress restrict its replay ability and user experience. Future works aim to address these by adding more narrative, save features, graphical storytelling, and enhanced UI/UX elements. This will improve user engagement and learning through a more immersive and visually appealing experience.

## REFERENCES

- Stedman, C., & Hughes, A. (2021, September 7). *data mining*. SearchBusinessAnalytics.  
<https://www.techtarget.com/searchbusinessanalytics/definition/data-mining>
- Linoff, G. S. (2015). *Data Analysis Using SQL and Excel* (2nd ed.). Wiley.
- Custer, C. (2021, April 29). *Why You Should Learn SQL if You Want a Data Science Job*. freeCodeCamp.org. <https://www.freecodecamp.org/news/learn-sql-to-work-in-data-science/>
- Park, J., & He, C. (2018). *GitHub - NUKnightLab/sql-mysteries: Inspired by @veltman's command-line mystery, use SQL to research clues and find out whodunit!* GitHub.  
<https://github.com/NUKnightLab/sql-mysteries>
- Brooks, C. (2022, June 29). *What Is SQL?* Business News Daily.  
<https://www.businessnewsdaily.com/5804-what-is-sql.html>
- Twin, A. (2022, August 2). *What Is Data Mining? How It Works, Benefits, Techniques, and Examples*. Investopedia. <https://www.investopedia.com/terms/d/datamining.asp>
- Leverage Edu. (2021, July 24). *Uses of SQL in Data Science & Businesses*.  
<https://leverageedu.com/blog/uses-of-sql/>
- Panoho, K. (2019, October 1). *The Age Of Analytics And The Importance Of Data Quality*. Forbes. <https://www.forbes.com/sites/forbesagencycouncil/2019/10/01/the-age-of-analytics-and-the-importance-of-data-quality/?sh=67eba4495c3c>
- The Analytics Advantage*. (2016, October 12). Deloitte.  
<https://www2.deloitte.com/global/en/pages/deloitte-analytics/articles/the-analytics-advantage.html>
- Davies, N. (2021, October 5). *Why You Should Learn SQL – Even If You're Not a Developer*. freeCodeCamp.org. <https://www.freecodecamp.org/news/why-learn-sql/>
- Singhal, P. (2022, August 29). *What are the Advantages of SQL?* Scaler Topics.  
<https://www.scaler.com/topics/advantages-of-sql/>
- Gros, B. (2007). Digital Games in Education. *Journal of Research on Technology in Education*, 40(1), 23–38. <https://doi.org/10.1080/15391523.2007.10782494>
- Kapp, K. M. (2012). *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons.

## APPENDICES

### Project Planning

| ID | Task  | Begin date | Start date | Progress | Duration |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
|----|---|------------|------------|----------|----------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|---------|---------|---------|
|    |   |            |            |          | WEEK 1   | WEEK 2 | WEEK 3 | WEEK 4 | WEEK 5 | WEEK 6 | WEEK 7 | WEEK 8 | WEEK 9 | WEEK 10 | WEEK 11 | WEEK 12 | WEEK 13 | WEEK 14 | WEEK 15 |
| 1  | Final Year Project 1                                    | 20/10/2022 | 12/1/2023  | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 2  | Brief Proposal  | 20/10/2022 | 28/10/2022 | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 3  | Full Proposal   | 29/10/2022 | 14/11/2022 | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 4  | Chapter 1: Introduction                                 | 15/11/2022 | 21/11/2022 | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 5  | Chapter 2: : Literature Review                          | 22/11/2022 | 9/12/2022  | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 6  | Chapter 3: Methodology, Requirement Analysis and Design | 10/12/2022 | 30/12/2022 | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 7  | Final Year Project 1 Final Report                       | 31/12/2022 | 12/1/2023  | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |

Appendices 1 FYP 1 Gantt Chart

| ID | Task                                   | Begin date | Finish date | Progress | Duration |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
|----|--|------------|-------------|----------|----------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|---------|---------|---------|
|    |  |            |             |          | WEEK 1   | WEEK 2 | WEEK 3 | WEEK 4 | WEEK 5 | WEEK 6 | WEEK 7 | WEEK 8 | WEEK 9 | WEEK 10 | WEEK 11 | WEEK 12 | WEEK 13 | WEEK 14 | WEEK 15 |
| 1  | Final Year Project 1 Revised Structure | 2/10/2022  | 22/10/2023  | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 2  | Chapter 4: Implementation              | 22/10/2022 | 3/12/2023   | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 3  | Chapter 5: Testing                     | 3/12/2023  | 21/12/2023  | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 4  | Chapter 6: Conclusion                  | 3/12/2023  | 21/12/2023  | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 5  | FYP 2 Final Report and Papers          | 21/12/2023 | 9/1/2024    | 100%     |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 6  | Product Deployment & Symposium         | 19/1/2024  | 19/1/2024   | 70%      |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| 7  | Final Report                           | 19/1/2024  | 19/2/2024   | 80%      |          |        |        |        |        |        |        |        |        |         |         |         |         |         |         |

Appendices 2 FYP 2 Gantt Chart