Min Thu

SPRINT 1 – DOCUMENTATION VISUAL PROTOTYPING, DATA BASE DESIGN AND DATA STRUCTURE SELECTION

Contents

[Goals and Deliverables 2](#_Toc184039132)

[Visual Prototyping 2](#_Toc184039133)

[Tools and Methods 3](#_Toc184039134)

[Styling Choices 3](#_Toc184039135)

[Design Iterations 4](#_Toc184039136)

[Database Schema 6](#_Toc184039137)

[Validation Testing 7](#_Toc184039138)

[Afterthought 10](#_Toc184039139)

# Goals and Deliverables

The objective of the sprint was to create a visual prototype of the web application, aswell as a documented database schema, the initial data structure and algorithms plans to support efficient local data storage and retrieval

Planned Work

* Visual Prototyping, creating the necessary user flow diagrams, visual prototype of the included pages
* Database Schema Design – define entities, relationships and indexing strategies for optimal query searching
* Algorithm Selection – By recognising the function of the query search and storage, we shortlist any standard data structure and algorithms for storage and retrieval that are commonly used.
* Prototype Validation – testing the database schema and selected data structure with sample data for performance

Actual Deliverables

* Produced a high fidelity prototype
* Established a structured ERD Diagram to define the entities, relationships involved
* Shortlisted common data structures and algorithms
* Created a test database and ran a benchmark to check performance of the schema

# Visual Prototyping

Before implementing visual designs of the web app, It was necessary to investigate similar applications that aligned with the web application’s goals and motivation. This was necessary to conceptualise the design in order to create a clean and intuitive design, addressing the goals of the project outline.

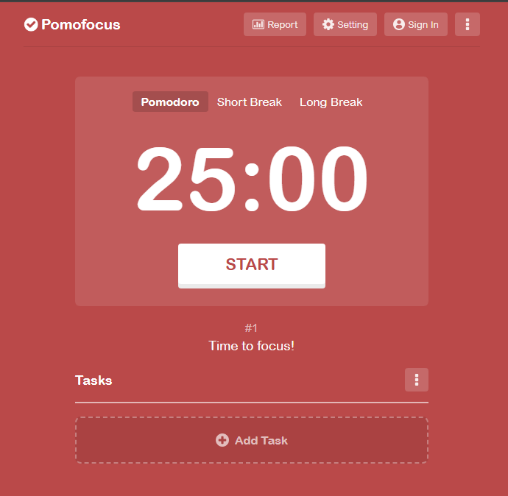
The figures shown are the applications that inspired the web app development, by using simple colour schematics and thematic of nature elements, it connotates serenity and mindfulness. The minimalistic design is visually appealing and not overwhelming which reinforces this idea of mindfulness. Through this investigation, I adopted neutral and lighter tones to create a calming visual experience. Additionally I incorporated nature imagery to further reinforce the idea of mindfulness and well-being. This minimalistic approach ensures the design remains visually appealing without overwhelming the user, aligning with the core principles of mindfulness and focus.

Figure 2 Forest APP design

Figure 1 Pomofocus.io design

## Tools and Methods

To design a visual prototype, I used Figma which offered browser based design tools to create user interfaces.

Using Figma made designing the prototype more streamline by utilising Figma’s design systems, components and styles.

The Frame tool acted as a foundational structure for the app screen, I was able to differentiate and design individual screens such as the timer and journal entry interfaces.

As it was a component based design system, creating reusable elements streamlined the design process (An example would be the backdrop elements used, aswell as the header, titles)

The Auto layout simplified resizing and modification as they adjusted dynamically.

## Styling Choices

To ensure consistency across interfaces, I had developed a colour palette and complementary fonts to ensure user interface was seamless and transitional upon observation.



Figure 3 Colour Palette

****

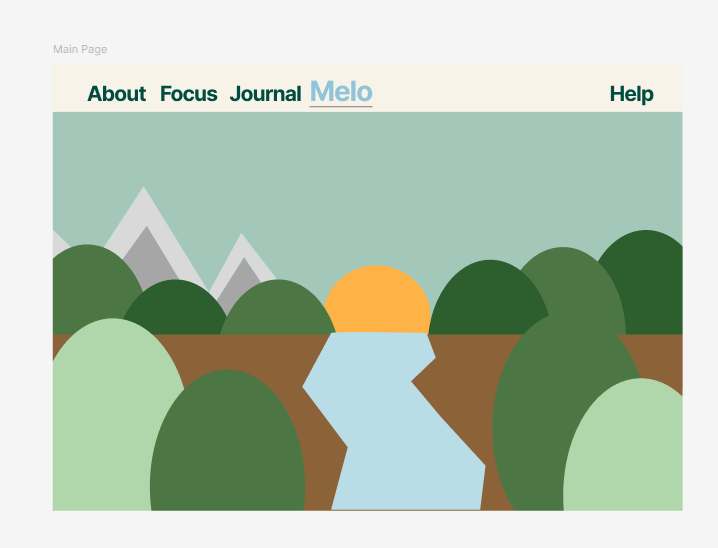
Figure 4 Colour Palette

## Design Iterations

 The initial design, however the colour was too sharp and contrasting so another choice was adopted.

Figure 5 Home screen Design

A warmer and washed colour was adopted to make it easier to look at, the lighter contrast to the text opted for a visually appealing interface



I then began adding themes of nature aswell as a change in the header to indicate which page the user was currently on

Figure 6 Home Screen Design Final

Reusing assets of the main page, a mountain scenery was creating using the simple shape tool provided by Figma, The font Playfair Display was used to compliment Inter, which was the main font I used for bolder and bigger titles.

Figure 6 About screen Design

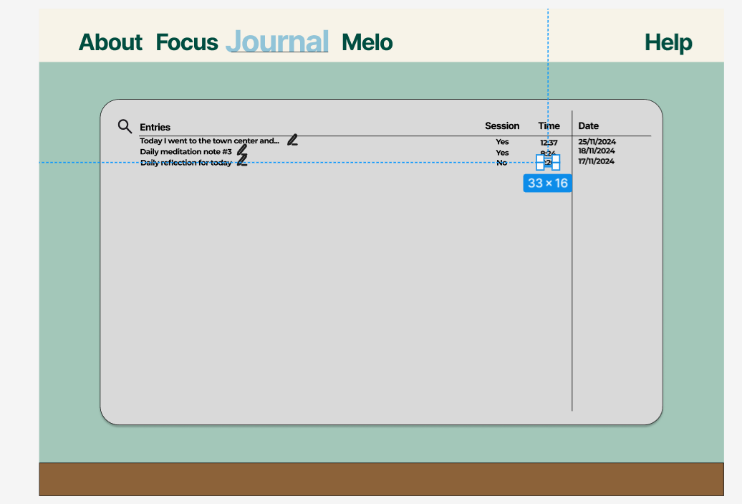
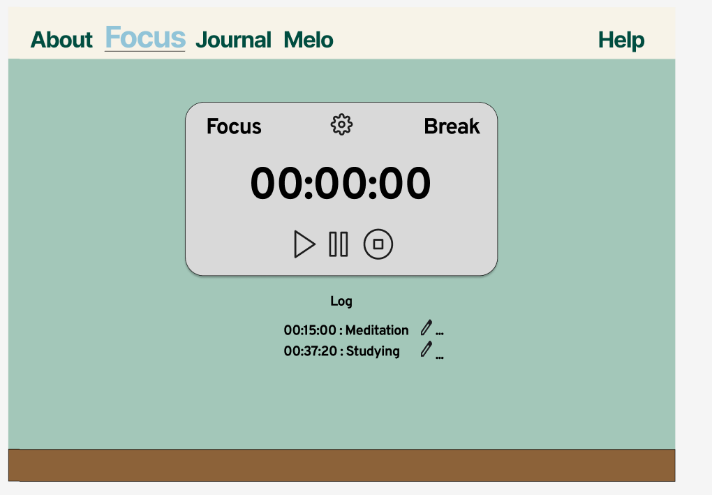


Figure 8 Focusl screen Design

Figure 7 Journal screen Design



Overall, the use of lighter tones and neutral colours and nature imagery effectively reflects the app’s purpose. The designs given serves as a foundation for the app’s front end development, ensuring consistent visual identity throughout the implementation process.

I was happy with the outcome given that Figma was a software I was unfamiliar with, I hope the next time I use Figma I can use more of the features available to me and dynamically add depth to the page, by giving some form of interactivity (such as the button asset provided by Figma)

Given the time frame I am given to complete the sprint, I would like to invest more time in wire framing before creating high fidelity prototypes to clarify layout and functionality and allocate more time to refining designs by incorporating a feedback session at each design stage.

# Database Schema

The objective of this task was to conceptualise the database structure by identifying key entities and their relationships to ensure efficient data management for the application

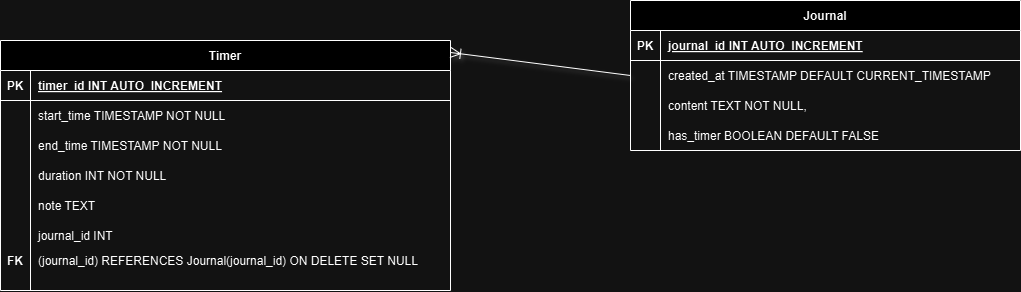


Figure 9 Database Schema

* Timer

The primary key used to identify each session was timer\_id

Start\_time and end\_time was used to track session startand end, duration calculated the duration of the session.

Note is an optional text field to add any additional details

Journal\_id is a foreign key that links the timer to a journal entry

* Journal

Similar to the timer entity, journal\_id is a unique identifier for each journal entry

Created\_at is a timestamp to mark when the entry was created

Content contains the text content of the journal entry

Has\_timer is a Boolean indicating if the journal is linked to a timer

The relationship between the entities is described to be “one to many”, as a single journal entry can have multiple associated timers but a timer can be optionally linked to a journal entry.

**ON DELETE SET NULL** ensures referential intregity by setting the journal\_id null in the Timer table if a journal entry that is associated to the timer is deleted.

This simple schema design allows efficient retrieval and storage by allowing timers and journals to exist independently with an associated unique identifier, integrity is ensured through foreign key constraints.

## Validation Testing

For the validation testing, I have decided to use MySQL to benchmark performance due to its integrated workbench with built-in performance tools. The graphical dashboard provides activity and performance statistics, including essential metrics for crud operations such as response times and transaction throughput. The testing focuses on the database’s ability to handle concurrent operations effectively.

MySQL supports automated scripting tools that allows Python scripts to be written for creating and performing query tests. Leveraging libraries such as mysql-connector-python to validate database constraints, which includes primary and foreign keys, constraints and cascading updates/deletes.

By automating these processes, It ensures consistent testing and intuitive validation of relationships and security measures within the database.

Figure 10 shows a script that is designed to setup and initialise a MySQL database, create sample data and test basic functionality

* Automates creation
* Defines two tables (journal and timer)
* Populates them with randomized test data
* Validates the setup through basic query.

The Script is intended for development, testing and educational purpose to help simulate and test a relational database system in Python.

Currently, MySQL uses B trees to store and manage indexes for efficient look ups, ‘order by’ uses quicksort.

So the existing algorithm and data structure alone is efficient by itself.

The libraries used :

Pymysql – interaction with MySQL database from python

Datatime, timedelta – manipulates data and times aswell as generates random time stamp

Random - is used to generate random numbers from the sample data

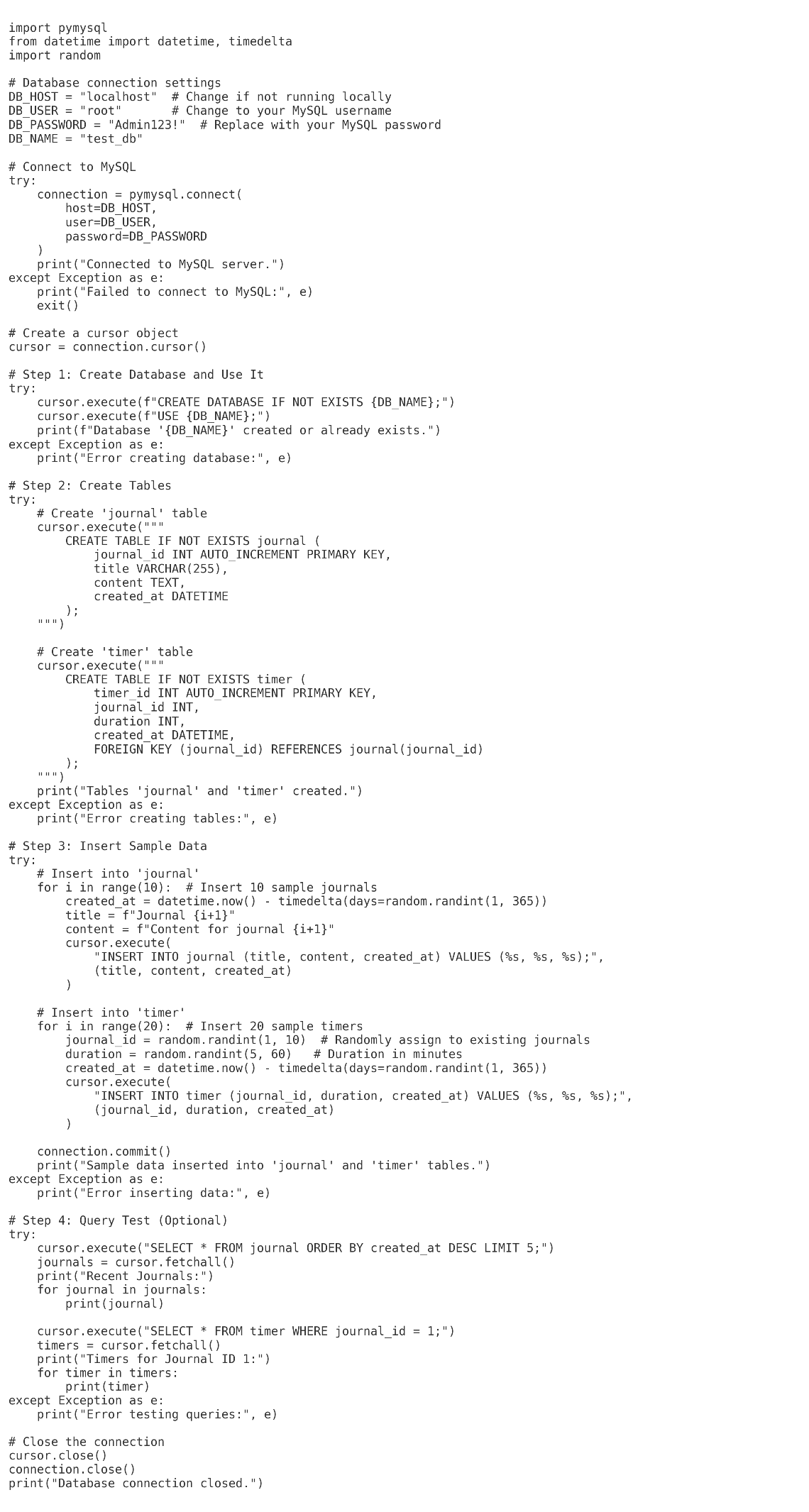


Figure 10 Database creation script

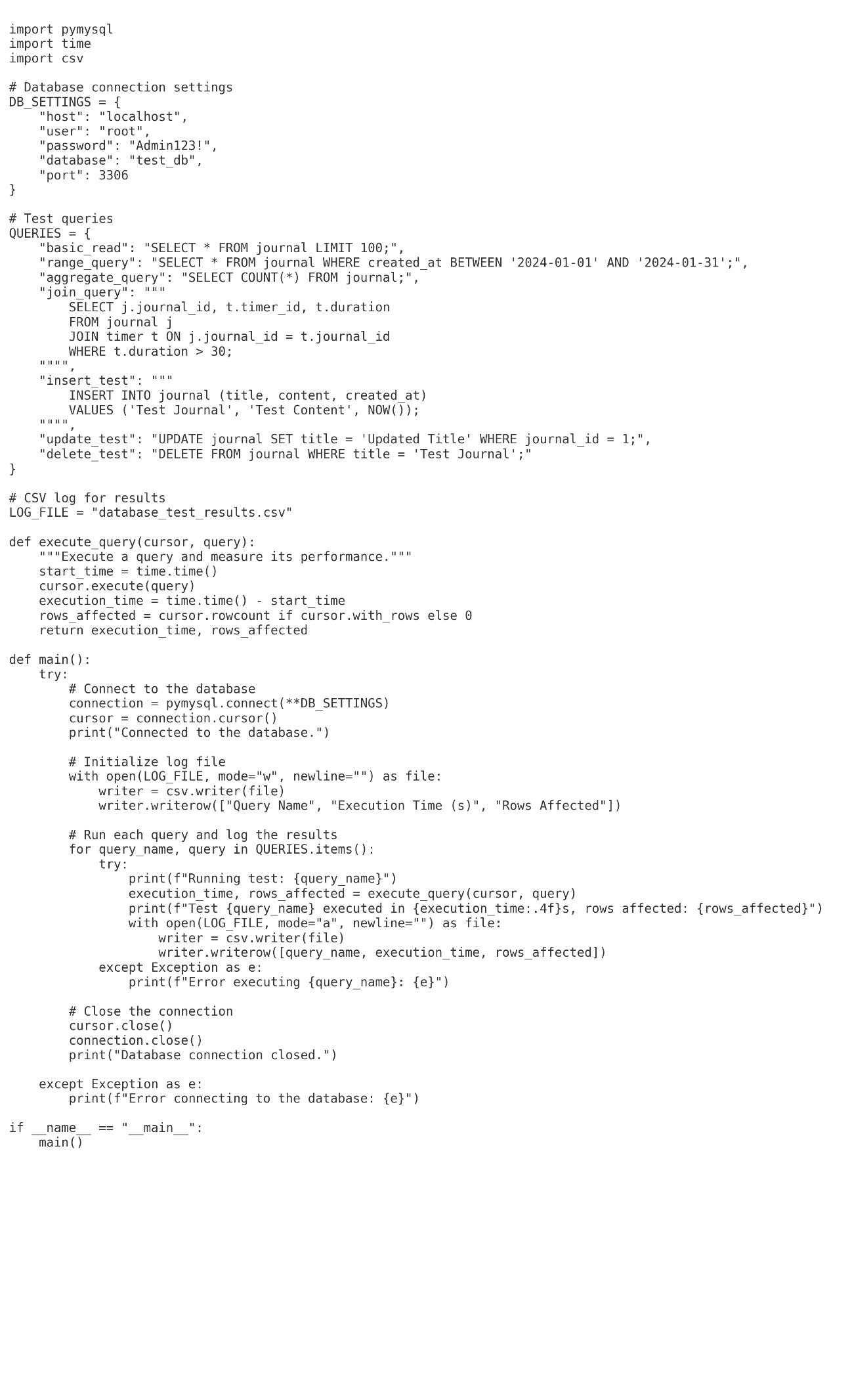


Figure 11 Automated query test script

The script that automates testing has the objective of

* Validating Schema, by verifying relationships (primary and foreign keys) and that data types and constraints are aligned with expected data

This is accomplished by checking appropriate indexes

* Performance benchmarking, measuring query execution time (mainly SELECT, INSERT, UPDATE and DELETE queries)

# Afterthought

After completing the 1st sprint which consisted of prototyping and scripting, it was an exciting overlook to how a developer may approach how they would develop a full stack application. A lot of the libraries used I was very unfamiliar with and I had to use a lot of tutorial and documentation code to help me assist in creating the script.

I’m very happy I did create the script considering I would have to create 1000 entries to benchmark performance.

As for other software’s such as Figma, It was quite confusing at first, but the UI and available resources made it very easy to navigate through the tools. As for future development It’ll definitely be one of the software’s I will pick up again whenever I am creating a high fidelity prototype.

Overall it was an insightful glimpse of the importance of adaptability and resourcefulness in software development, challenges from learning new libraries to using unfamiliar software, It pushed me to grow as a developer and I am very excited to apply these skills in the next stages of development.