HABIT TRACKER

AARON ALEX SEQUEIRA

INTRODUCTION

- Brief overview of the habit tracker program
  - The habit tracker application is an application used to track users habits and analyze them to provide an accurate statistic on the habit using different functions from the class
- Importance of habit tracking for personal development
  - Habit tracking can be used for personal gain such as breaking a habit where you bite your nails or it can be used to develop useful habits such as practicing your coding skills by doing more problem solving and the habit tracker reminds you of these habits that you must complete
- Objectives of the presentation
  - To achieve greater understanding of the project

PROGRAM OVERVIEW

- Description of the main functionalities provided by the program
  - The main function in the program denoted by the __main__ is the heart of the project as it initializes the object habit which allows the main function to access the function of the habit class. Features such as habit creation, removal, checking off, and analysis. The main function is the main CLI which uses questionary as its cli as it is clean
- Importance of each feature in habit tracking
  - Each function has its own importance as some functions are used for other functions to operate such as checkoff() function needs analyze function to be accessed and how timer_function() is used when checkoff() is executed. The viewAllHabits() function is used to view all the current habits in the database and the variations of the function such as longestStreak(), latestStreak(), most_Struggling_habit(), least_Struggling_habit(), Daily_habits()
  - Weekly_habits()

IMPLEMENTATION DETAILS

- Explanation of the main components of the program
  - The main components of this programs are create habit, remove habit and analyze habit each function has its own uses as create habit is used for the user to create a new habit by inputing the required information/data.The remove habit is used for the user to remove an existing habit in the database by inputing the name to the program. Finally the analyze function is used to manage and view the existing habits
- How Python's threading module is used for implementing timers
  - The implementation of threads in the project is used to uncheck and break habits that have been checked after the user inserted the name of the habit in the checkoff() function that sets off a timer.
- Use of questionary library for user input
  - The use of questionary is to create a clean CLI instead of inserting numbers to navigate throughout the program and to avoid potential errors in the program

CLASS STRUCTURE

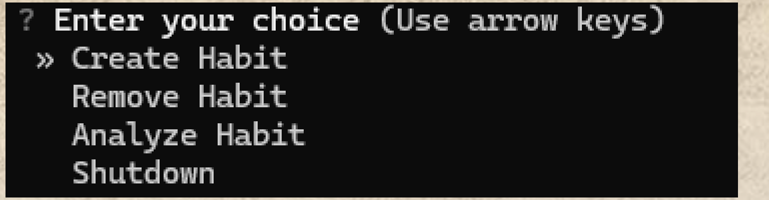
The project revolves around one class and it is called the habit class, in this class there are various functions needed to track the habit and analyze it and to view them. The class provides various functions for sorted view and to modify the habits. These habits can be removed by the user by inserting their name and can create habits by inputing the data required for the program to analyze the habit. The class uses a default constructor for now and more functions are yet to be added to the project.

DATABASE SCHEMA

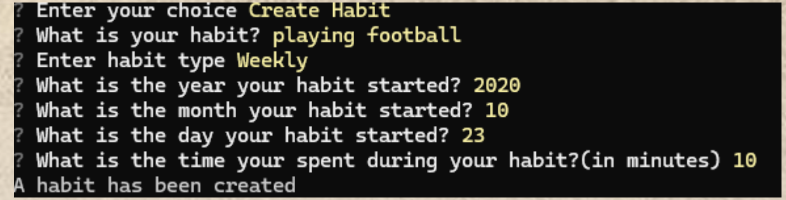
- Overview of the MySQL database schema used for storing habit data
  - The main database used called habittrackerdb contains the table habits where the data is stored
  - The database will have multiple tables for different users after they create an account
- Table and its attributes:
  - 1,name,habittrackerdb,habits,VARCHAR.
  - 2,time_Since_Creation,habittrackerdb,habits,VARCHAR.
  - 3,Total_hours,habittrackerdb,habits,INT.
  - 4,start_Date,habittrackerdb,habits,DATE.
  - 5,minutes_Saved,habittrackerdb,habits,INT.
  - 6,checked,habittrackerdb,habits,TINYINT.
  - 7,streak,habittrackerdb,habits,INT.
  - 9,Check_date,habittrackerdb,habits,DATE.
  - 8,Habit_Type,habittrackerdb,habits,VARCHAR.
- Importance of data organization for efficient habit tracking
  - The data stored in the table is used for the calculation of the time saved and time since creation

DEMONSTRATION

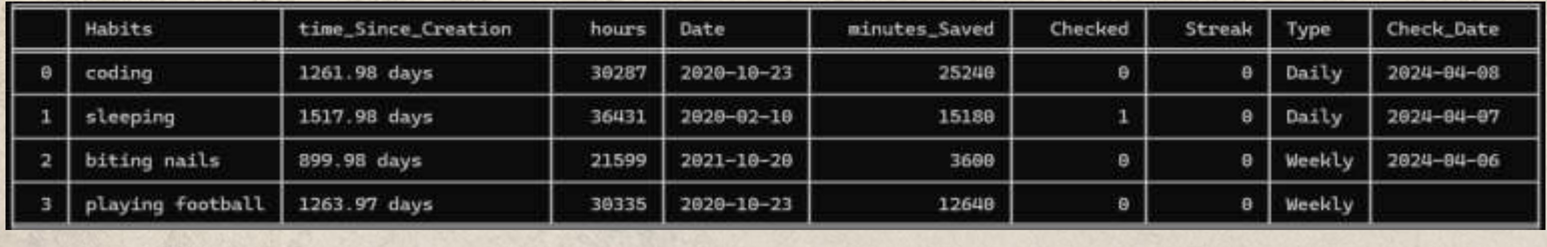
- Screenshots or code snippets showcasing the program in action



```
? Enter your choice (Use arrow keys)
 » Create Habit
   Remove Habit
   Analyze Habit
   Shutdown
```

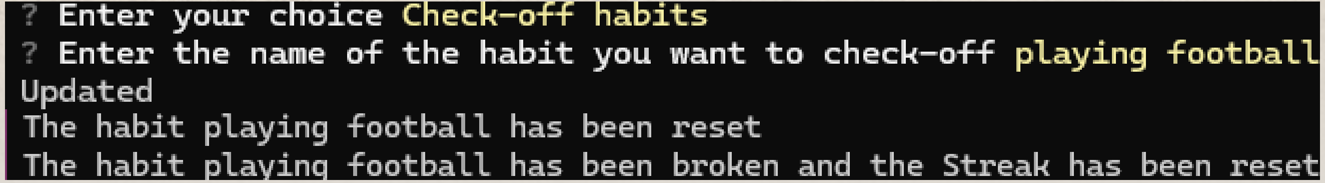


```
? Enter your choice Create Habit
? What is your habit? playing football
? Enter habit type Weekly
? What is the year your habit started? 2020
? What is the month your habit started? 10
? What is the day your habit started? 23
? What is the time your spent during your habit?(in minutes) 10
A habit has been created
```



	Habits	time_Since_Creation	hours	Date	minutes_Saved	Checked	Streak	Type	Check_Date
0	coding	1261.98 days	30287	2020-10-23	25240	0	0	Daily	2024-04-08
1	sleeping	1517.98 days	36431	2020-02-10	15180	1	0	Daily	2024-04-07
2	biting nails	899.98 days	21599	2021-10-20	3600	0	0	Weekly	2024-04-06
3	playing football	1263.97 days	30335	2020-10-23	12640	0	0	Weekly	

- Example scenarios of creating, checking off, and analyzing habits



```
? Enter your choice Check-off habits
? Enter the name of the habit you want to check-off playing football
Updated
The habit playing football has been reset
The habit playing football has been broken and the Streak has been reset
```

PERFORMANCE AND SCALABILITY

- performance of the program with large datasets
  - The program has passed all the test and the components have been tested and work without errors
- Potential optimizations for improving performance
  - The program needs to be optimized for creating a new account and handling more users to the database

CONCLUSION

- Summary of key points covered in the presentation
  - Habit class: -The core functionality is encapsulated within the Habit class.
  - Database Interaction:-The program interacts with a MySQL database to store habit data.
  - Functionality: -Habit creation involves specifying details such as name, start date, time spent, and habit type (daily or weekly).
- Reflection on the design and implementation process
  - **Design Patterns**: The code follows a procedural programming style with a class (**Habit**)
  - Database Interaction: Interaction with the MySQL database is implemented using the “mysql.connector” library.
  - User Interaction:The program utilizes the questionary library for user input.

- Future enhancements and potential areas for further development
    - Adding authentication to the project to have more than one user
    - Better habit management system to better the user experience
    - Using a real user interface(UI) framework to make the app more user friendly
    - Adding more statistics on the habit to give more information to the user

- Therefore, this is the brief overview of the projects and how it works, the key components and the main functions