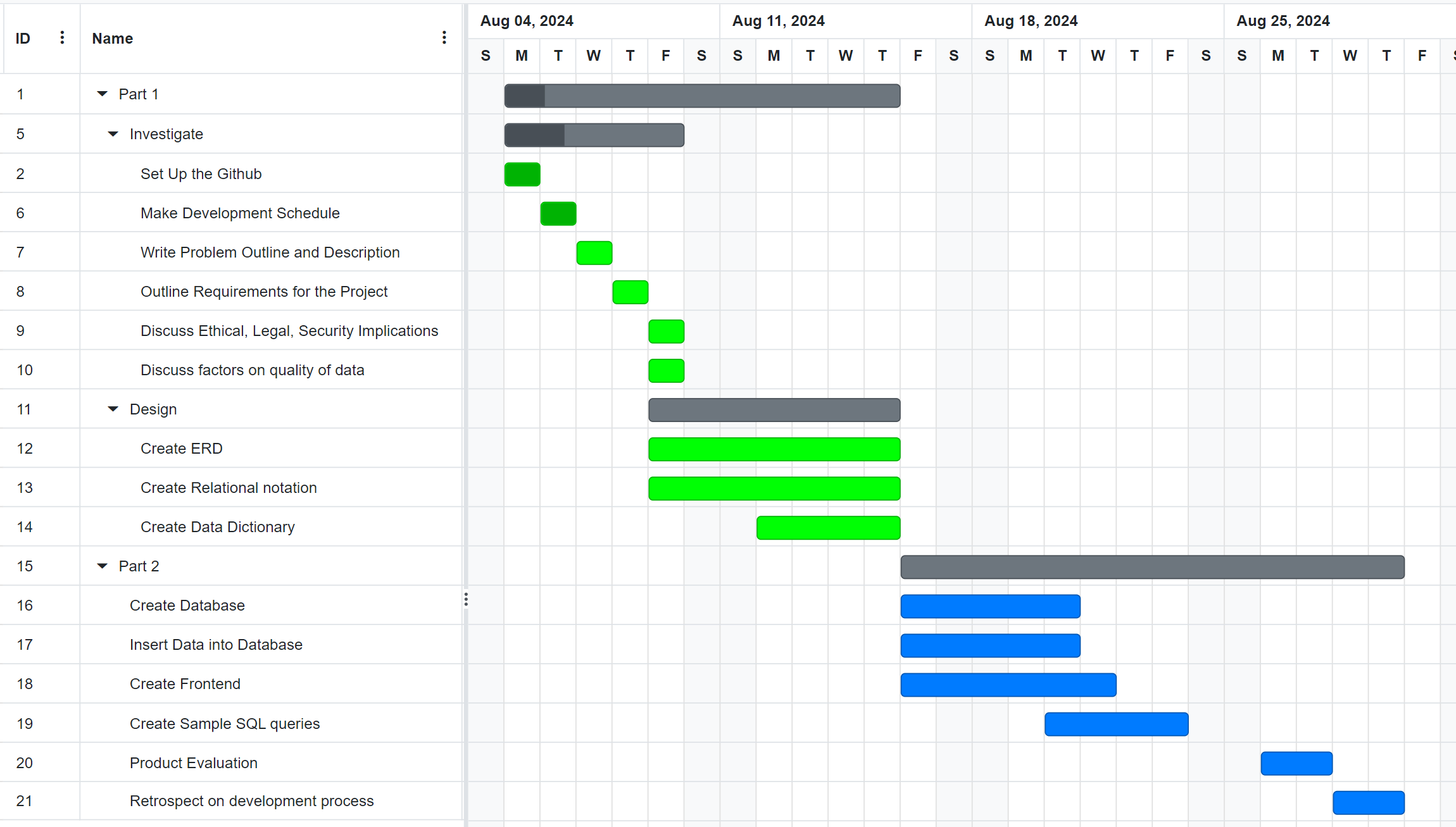
**Timeline Gantt Chart**



*(Made using* [*https://www.onlinegantt.com/#/gantt*](https://www.onlinegantt.com/#/gantt)*)*

Steps:

1. Set up GitHub
2. Make development Schedule
3. Write problem Outline and Description
4. Outline Requirements for the Project
5. Discuss Ethical, Legal, and Security Implications
6. Discuss factors on qualities of data
7. Create ERD
8. Create Relational notation
9. Create Data Dictionary
10. Create Database
11. Insert Data into Database
12. Create Frontend
13. Create Sample SQL Queries
14. Product Evaluation
15. Retrospect on development process

**Problem Outline**

A family member is starting a new online business and needs a software solution to manage customers and record sales information for an online music streaming service. They require a backend for storing music data, and a basic frontend for user and administrator use.

This includes, creating a database in SQLite and filing it up with some sample data, alongside creating a python frontend that can interact with and query this database. The backend must track customer data, sales information, and follow legal requirements such as the adequately safeguarding customer data.

**Problem Description**

I have been tasked with developing a front and backend for an online music streaming service by a family member. The backend database built in SQLite of this software, must be able to manage customer data, record sales information, and store music track and album data, along with keeping user data confidential. The frontend must be able to make SQL queries to the database and display the outputs of said queries to the user. Ethical, legal, and security issues must be considered when creating this software solution, to take protect customer data.

**Problem Scenario**

A family member has asked me to assist them in the creation of their business by developing a software solution in the form of a frontend and backend for their online music streaming service. Customers can subscribe to the service, and through their account, access all the music stored within the database through an intuitive frontend, just by logging in, and searching up the song title, or genre. They want the backend to be able to securely store customer data and accounts for the customers, for ease of use. They want to be able to access customer data, and record subscription figures with ease.

**Requirements for the Solution**

**Frontend –**

**Functional:**

* Users need to be able to create accounts and sign in to said accounts
* The user must be able to change and/or delete their customer data
* Customers can browse songs based on certain input criteria
* Results of any queries made to the database must be displayed to the user

**Non-Functional:**

* An intuitive interface for users to use
* SQL code should never be shown to the user
* The user should never see a non-custom error message
* Users can add songs to playlists and these playlists are stored and the songs they contain can be edited at any time
* The program should never crash on the user
* Fast response time on the frontend

**Backend –**

**Functional:**

* This account data must be stored, and accessible to the user
* All of the songs must be stored in the database, along with their respective attributes
* The database must be in third normal form to reduce data anomalies
* The frontend python code must be able to query the backend SQLite database
* Must store all subscription data, on customers
* Data in the database must be validated using appropriate constraints

**Non-Functional:**

* All inputs should be validated, so there are no SQL injections
* User data in the database is updated every once in a while, by asking them if it is up to date.
* User data in the database is encrypted
* User passwords are stored as hashes
* The code is clean, and easy to read