Team 3

Administrator Manual

Hasan Ali, Chloe Farrell, Talha Hussain, Riley Sheehy, Thomas Nguyen, Coby Ziv

5-23-2023

Table of Content

Contents

1. Introduction 2

2. System Overview 2

2.1 Background 2

2.2 Hardware & Software Requirements 2

3. Administrative Procedures 4

3.1 Installation 4

3.2 Routine Tasks 6

3.3 Periodic Administration 7

4. Troubleshooting 8

4.1 Dealing with Error Messages and Failures 8

4.2 Known Bugs and Limitations 9

1. Introduction

This project aimed to develop a comprehensive heat map visualization of crime data within the city of Baltimore, while examining its correlation with the availability of food sources. Leveraging the Python server library Flask and a MySQL database for data management on the backend and employing the JavaScript library Leaflet.js to enable heatmap functionality on the frontend, our study successfully generated an interactive visual representation of crime hotspots and food access points within the city. This paper presents the methodology employed, data collection and processing techniques, as well as the findings pertaining to the relationship between crime incidents and food availability.

2. System Overview

2.1 Background

The system administrator overseeing the software web application holds the primary responsibility of ensuring the system's efficient operation and maintenance. This encompasses a range of tasks from the initial installation to day-to-day management, with the goal of maintaining the application's functionality and protecting its data.

The first task of the administrator is to carry out the installation and configuration of the software web application. This process involves setting up the necessary hardware and software components as outlined in Section 2.2 of the manual. Once the software is successfully installed, the administrator assumes the responsibility of monitoring and maintaining the application's performance and availability. Routine maintenance activities are a key aspect of the administrator's role, including tasks such as applying software updates and patches, optimizing database performance, and promptly resolving any system errors or crashes that may arise.

The administrator's overall role is focused on ensuring the efficient operation, security, and maintenance of the system. This encompasses implementing measures to protect sensitive data, conducting regular security audits, and taking proactive steps to mitigate potential vulnerabilities. By fulfilling these responsibilities diligently, the administrator provides users with a reliable and secure platform to effectively utilize the software web application.

2.2 Hardware & Software Requirements

Hardware

In order to determine the necessary hardware and software for the project, the team conducted a thorough analysis of the project requirements and considerations. Regarding hardware requirements, it was determined that a CPU with a 64-bit x86 architecture would be suitable to handle the processing demands of the project. Additionally, a minimum of 8GB of RAM was deemed necessary to ensure smooth execution. As for disk space, a conservative estimate of 100MB was considered sufficient for storing the project files and data.

Required Hardware

- CPU: 64 bit x86
- RAM: 8GB
- Disk Space: 100MB

Software

In terms of software, the team recognized the need for a compatible operating system that would support the project's development and deployment. The options identified included Windows, MacOS, and Linux, enabling flexibility for team members with different operating system preferences. To handle the data management aspect, the team decided to utilize a MySql 8.0 database or an equivalent relational database system to efficiently store and retrieve project data.

For implementing the project's functionalities, the team selected Python 3.10 or a more recent version as the programming language. Python's versatility, extensive libraries, and ease of use made it an ideal choice for data processing and analysis tasks. To facilitate seamless integration with the MySql database, the team identified specific pip packages, including pandas, numpy, mysql-connector-python, and python-dotenv, which were essential for data manipulation, numerical computations, and establishing a connection with the database.

Required Software

- Operating System: Windows/MacOs/Linux
- Database: MySql 8.0 database or equivalent
- Coding Language: Python 3.10 or greater
- pip packages:
    - pandas
    - numpy
    - mysql-connector-python
    - python-dotenv

By collectively considering the hardware and software requirements, the team ensured that the project would be supported by an appropriate technical environment capable of effectively executing the desired functionalities and achieving the project's objectives.

3. Administrative Procedures

3.1 Installation

Download the software by clicking here: https://github.com/Team-3-CMSC-447-Spring-2023/CrimeDatabase

Steps for setting up the database:

1. First, open your browser and go to the URL website https://dev.mysql.com/downloads/installer/. Download the latest version of MySQL Installer from the website https://dev.mysql.com/downloads/installer/ by clicking the first blue "**Download**" button in the tab "**General Availability (GA) Releases**".

2. Second, your browser will automatically take you to the website https://dev.mysql.com/downloads/file/?id=518834 where you will officially download the current version of MySQL Installer as a result of clicking on the hyperlink message "**No thanks, just start my download.**" at the bottom of the website https://dev.mysql.com/downloads/file/?id=518834. Wait for the download to finish loading as it pops up at the bottom of the web page which you will then click to begin installing the MySQL Installer.

3. Third, within either the Microsoft Windows Command Prompt, Visual Studio Code, PyCharm, or any other possible source-code editor that can run Python code, type the following commands in your terminal to successfully install those commands:

    a. pip install mysql-connector-python
    b. py -m pip install install python-dotenv
    c. pip install pandas
    d. pip install numpy
        i. NOTE: Make sure that the installation commands are installed in the correct "C:...." File Path where your Python File will be located.

4. In Microsoft Windows Command Prompt or Source-Code Editor that you chose, open the file "**database.py**" and make edits to the values that are assigned to the configuration variables in lines 6 to 10 within the "**mydb = mysql.connector.connect()**" changing all the configuration values to match the mysql server you are running:

    a. **host=** <Name of Your Host>
    b. **user=** <Your Database User Name>
    c. **password=** <The Connection Port>
    d. **database=** <Name of the Database>

5. Next, in the Source Code Editor of your choosing, either Windows Command Prompt, Visual Studio Code, PyCharm, etc., run the "**database.py**" Python code in a terminal based on the version of Python you have installed on your computer by running the command "**python database.py**".

    a. WARNING: Do not press "Ctrl + C" on your computer keyboard or laptop keyboard to stop the Python File "database.py" from running, otherwise, the data tables coded in the "database.py" Python File will not be generated in your MySQL Workbench.

6. Then, if you are using Microsoft Windows, click on the Windows "**Start**" menu located on the bottom left-corner of your computer screen and then in the Search Bar next to the Windows "**Start**" menu, type "**mysql**" to search for "**MySQL 8.0 Command Line Client**" App on your computer. The "**MySQL 8.0 Command Line Client**" Application will be used to ensure that name of the database that the user has given in line 10 of the "**database.py**" Python Code is recognized by the "**MySQL Workbench 8.0 CE**" App. This will be done by first checking that the name of the database assigned to "**database_name=**" in line 10 of the "**database.py**" Python Code exists by running the command "**show DATABASES;**" in "**MySQL 8.0 Command Line Client**" to list all databases that exist for running in MySQL. If one of the listed database names matches the assigned value that you typed earlier in the single quotation marks '' after "**database_name=**" in line 10 of "**database.py**" Python code, then skip Step 7 and go onto Step 8, otherwise, continue to Step 7 if "**MySQL 8.0 Command Line Client**" did not list the name of your database that you manually typed earlier in the single quotation marks '' after "**database_name=**" in line 10 of the Python code "**database.py**".

7. In "**MySQL 8.0 Command Line Client**" type "**CREATE DATABASE**", then the name of the database that you assigned to "**database_name=**" in line 10 of "**database.py**" Python code before you ran it, and then a semicolon "**;**". For example, if you assigned the value of the name of your database within the single quotation marks '' after "**database_name=**" in line 10 of the "**database.py**" Python code to the default name "**crime_database**" then the full command that you would type and then execute in "**MySQL 8.0 Command Line Client**" would be "**CREATE DATABASE crime_database;**". As a result of rerunning the command "**show DATABASES;**", you will now see that one of the databases listed in "**MySQL 8.0 Command Line Client**" matches up with the name of your database that you manually typed earlier within the single quotation marks '' after "**database_name=**" in line 10 of the Python Code "**database.py**".

8. Then, if you are using Microsoft Windows, click on the Windows "**Start**" menu located on the bottom left-corner of your computer screen and then in the Search Bar next to the Windows "**Start**" menu, type "**mysql workbench**" to search for "**MySQL Workbench 8.0 CE**" App on your computer. The "**MySQL Workbench 8.0 CE**" Application will be the tool to use for generating all possible Structured Query Language SQL expressions after running the Python Code "**database.py**" in your terminal.

9. Click on the "**MySQL Workbench 8.0 CE**" App and it will automatically open the "**MySQL Workbench**" Window. When it opens, you should see the large font message saying "**Welcome to MySQL Workbench**" and then under the "**MySQL Connections**" section below "**Local instance MySQL80**", you should see the value for your MySQL User Name that should be typed between the single quotes '' in line 7 after "**user=**" in the Python File "**database.py**" and in the second line you should see the value of your port number that should be typed between the single

quotes " in line 9 after "**port=**" in the "**database.py**" Python File.

10. Click on "**Local instance MySQL80**" under the "**MySQL Connections**" section in the "**MySQL Workbench**" Window. After you click "**Local instance MySQL80**", a small "**Connect to MySQL Server**" Window will pop up on your computer screen that will ask you for the password you created for your MySQL Account. Therefore, type your password in the rectangular entry box after "**Password:**" to successfully connect to your Local Host and then at the bottom of the "**Connnect to MySQL Server**" Window, press the "**OK**" button which will automatically open the SQL Editor in the "**MySQL Workbench**" Window.

11. Then in "**Navigator**" on the left side of your SQL Editor in "**MySQL Workbench**" Window, under the "**Filter objects**" Search Bar after "**SCHEMAS**", double left-click on the right triangle icon before the cylinder icon that shows the name of the database that you typed before running the "**database.py**" Python File that is the assigned value for your database name "**database=**" within the single quotes on line 10 of "**database.py**" Python code.

12. Under the "**Tables**" section, Right-Click "**crime**" and then Left-Click "**Select Rows - Limit 1000**" that will display the generated output data for the "**crime**" data table. Right-Click "**crime_type**" under the "**Tables**" section and then Left-Click "**Select Rows - Limit 1000**" that will display the generated output data for the "**crime_type**" data table. Under the "**Tables**" section, Right-Click "**neighborhood**" and then Left-Click "**Select Rows - Limit 1000**" that will display the generated output data for "**neighborhood**" data table. Last but not least, Right-Click "**weapon**" under the "**Tables**" section and then Left-Click "**Select Rows - Limit 1000**" that will display the generated output data for "**weapon**" data table.

To run the web application, follow these steps:

1. Navigate to the "Crime_Database" folder inside the "CrimeDatabase" directory
2. Open the file "app.py" in your source-code editor, terminal, or command prompt
3. Running "app.py" will provide you with a local URL
4. Copy the local URL and paste it into any web browser of your choice
5. You can now start using the web application through the provided URL

3.2 Routine Tasks

Routine tasks are crucial for sustaining optimal performance levels and ensuring the success of the project. These routine tasks encompass updating data, handling edge cases, and conducting comprehensive testing. Additionally, the team aims to enhance the user experience by incorporating new features, such as user account creation and facilitating data exports to aid users in their research efforts. The following are periodic routine tasks that need to be carried out:

- **Data Updates**: Regularly updating the project's data ensures that the information available for analysis and visualization remains current and accurate. This includes retrieving the latest crime data from reliable sources and incorporating any relevant updates or additions into the system.
- **Handling Edge Cases**: Identifying and addressing edge cases is essential to maintain the project's reliability and robustness. By analyzing unusual scenarios or uncommon data patterns, the team can develop strategies to handle such cases and prevent unexpected system behavior or errors.
- **Test Case Maintenance**: Continuous monitoring and updating of test cases are necessary to validate the functionality and performance of the project. Creating new black box and white box test cases to ensure that the system consistently operates as expected and remains free of issues or bugs.
- **System Documentation and Knowledge Base Maintenance**: Updating and maintaining comprehensive system documentation, user guides, and knowledge base resources ensure that accurate and up-to-date information is available for users and system administrators.

At present, the team's web application lacks the functionality to establish and manage user accounts. However, these features are part of the team's future development plans, aimed at enhancing the web application experience and personalizing it for users. The team intends to explore various updates, including webpage customization and research saving, to provide a more tailored experience. While the specific details of these features are yet to be determined, the team anticipates incorporating the following routines into the future updates:

- **User Registration**: The web application will provide a user registration process for new users to create a new account. This will involve collecting information such as name, email and password. This will then need to be validated to ensure there is no duplication or a weak password is being to provide user security.
- **User Authentication**: After creating an account, the user must proceed with the authentication process. Typically, this involves using the information provided during the sign-up phase. The web application will send an email to the user, requesting verification. The user can verify their account by either clicking on the URL provided in the email or entering the passcode included in the email.
- **User Account Management**: These tasks involve providing users with the ability to oversee and control their accounts, encompassing various features such as modifying personal details like usernames, email addresses, and passwords.
- **Account Deactivation and Removal**: If the user wants to deactivate or delete their account, the application should provide a means of way to do so and securely.

Conducting these routine tasks on a periodical basis ensures the project remains efficient, reliable, and responsive to user needs. By regularly updating data, addressing edge cases, maintaining test cases, and introducing new user-friendly features, the team can continuously enhance the project's performance and usability, delivering a high-quality experience to users.

3.3 Periodic Administration

Performing periodic tasks plays a vital role in the smooth operation and maintenance of a system. These tasks involve recurring activities that need to be performed at regular intervals to ensure the system reliability, security, and optimal performance. Below the team will list periodic tasks that will be needed to complete periodically and followed by future features periodic tasks.

System Backups: Regular system backups are essential for safeguarding data and minimizing the risk of data loss in the event of hardware failures and software errors. Periodically performing system backups helps create a snapshot of the system's current state, including data, configurations, and settings. Key aspects of system backups include:

- **Backing up the database**
  - mysqldump command "mysqldump <database name> > backup-<current date>. sql"
- **Data Integrity**
  - This verifies the integrity of the backed up data and ensures all the data was captured and can be restored when needed
- **Restoring the database**
  - Open the Mysql terminal application, if you are using Microsoft Windows, click on the Windows "Start" menu located on the bottom left-corner of of your computer screen and then in the Search Bar next to the Windows "Start" menu, type "mysql" to search for "MySQL 8.0 Command Line Client".
  - Once in the MySql terminal app type "use <database name>;" and hit enter to select the database.
  - Use the "Source" command to select a ".sql" file to use. For example, "source backup-2023-05-11.sql" and hit enter.

As mentioned in section 3.2, the current team's web application lacks the functionality of creating users, however, it is something the team will be developing in the near future. Some of the periodic tasks for user accounts will be:

- **Cleaning up user accounts**: Over time the, users will become inactive, or they might delete their account. It is important to review and clean up these accounts to ensure the privacy of data, security and efficient database management are maintain

Regularly performing these tasks will not only help maintain the system but also contribute to the efficient management of users and resources. By conducting these tasks periodically, the team can lower risks, protect the data and ensure ideal performance.

4. Troubleshooting

4.1 Dealing with Error Messages and Failures

For administrators, effectively handling serious system messages and failures is a crucial aspect of their role. Here are some recommended approaches:

- **Code Analysis**: Maintain composure when encountering errors or failures and conduct a thorough analysis of the code. Gather relevant information and conduct research to understand the root cause of the issue.
- **Documentation**: Document the details of errors or failures, including their occurrence time, location, and how they manifested. Such documentation provides valuable information for future reference and aids in identifying patterns or recurring issues. Additionally, document the solutions implemented to address these issues to facilitate efficient resolution in case of their recurrence.
- **System Log Examination**: Review system logs to identify specific error messages and pinpoint the source of the error. If necessary, reach out to the software team for assistance or conduct research to find potential solutions to the identified issues.
- **Research**: When confronted with errors or failures, ensure that you are using the correct commands and following instructions accurately. Engage in research to understand if the error or failure stems from your actions or if it is a known issue with established resolutions.
- **Seek Support**: Remember that assistance is available when dealing with errors or failures. Reach out to the software team to report persistent issues and seek their guidance. Additionally, explore online communities or forums where other users may have encountered similar errors or failures, providing an opportunity to learn from their experiences and potential solutions.

By following these guidelines, administrators can effectively address and resolve errors and failures, ensuring the smooth operation of the system.

4.2 Known Bugs and Limitations

No bugs have been founded in the software as of right now. The team is working on creating new test cases to ensure the software work as planned without any errors.