BEFORE RUNNING THE APPLICATION…

In order to run the application, you MUST FIRST install the software requirements. In doing so, follow the steps indicated under the succeeding subsections.

The prerequisites

- Before anything else, make sure that you have the following:
    - Mkcert
        - Click this link to install Chocolatey
        - Run **choco install mkcert** on the terminal as an administrator
        - Follow up with this command: **mkcert -install**
    - Git and GitHub Desktop
        - This will be used in cloning the repository. As for its installation, kindly follow these:
            - For Git: Open this & Download
            - For GitHub Desktop: Open this
    - Python
        - You may download the package in this link and follow the installation instructions via this guide.
    - Visual Studio Code
        - Because of its flexibility, it is recommended to use Visual Studio Code as the IDE for the project.
    - Devices for testing
        - Desktop with Google Chrome (Windows)
    - MySQL Workbench
        - You may download this in this link and follow the installation instructions

Clone the GitHub repository

- Use the repository's URL for cloning:
    - https://github.com/Kirito1456/SECWB
- Clone the repository via:
    - GitHub Desktop
        - Open GitHub Desktop
        - Click on 'File' -> 'Clone Repository'
        - Click the tab indicating ‘URL’
        - Paste the URL on the first input field and select the path where you want to store the clone repository.

Creating Django Python Environment and Installing Django packages and dependencies

- While inside the folder, together with the project directory, open a terminal.
- Type and enter this command: **py -m venv <name of env>**
    - This will create a django environment
- Next, use this command **<name of env>\scripts\activate** to activate the environment
- Navigate inside the project folder where requirements.txt is located
    - Use the command: **cd <path to folder>**
- Install all dependencies using this command:
    - **pip install -r requirements.txt**
        - This command will fetch and download the packages and dependencies. Take note that this step is crucial as most components of the system rely on these.

Creating Database Schema in MySQL Workbench

- Launch MySQL Workbench on your computer.
- Connect to the Database Server:
    - In the MySQL Workbench main window, click on the database server connection you want to use or create a new connection if you haven't done so already.
- Navigate to the menu and select Server > Data Import.
- In the "Data Import/Restore" dialog, select the option "Import from Self-Contained File".
    - Click on the ... button to browse and select your .sql file containing the schema.
- Click on the "Start Import" button to begin the import process.
    - Once the import is complete, you should see a confirmation message indicating that the import was successful.
- To Verify the Imported Schema
    - Go back to the "Schemas" tab on the left pane.
    - Right-click on the schemas area and select "Refresh All" to refresh the list of schemas.
    - Expand the database schema you imported to verify that all tables and other objects have been imported correctly.

Connect Database in the Source Code

- Open Visual Studio Code and Open the folder containing the source code
- Navigate inside the project folder settings.py is located
- Open **settings.py** and scroll down under **DATABASES**
- Configure your mysql workbench database credentials

HOW TO RUN THE APPLICATION…

Running the Application on HTTP

- While running the django environment, navigate inside the project folder where the **manage.py** is located
- Type and run this command:
    - **python manage.py makemigrations**
    - **python manage.py migrate**
        - These commands will make sure that all necessary migrations are completed to run the application
    - **python manage.py createsuperuser**
        - This command is for creating the admin user
    - Use this command to run the application: **python manage.py runserver**
- Open a web browser and go to **http://127.0.0.1:8000/** to view the application.

Running the Application on HTTPS

- While running the django environment, navigate inside the project folder where the **manage.py** is located
- Generate your own certificate with this command:
    - **mkcert -cert-file cert.pem -key-file key.pem localhost 127.0.0.1**
- Type and run this command:
    - **python manage.py makemigrations**
    - **python manage.py migrate**
        - These commands will make sure that all necessary migrations are completed to run the application
    - **python manage.py createsuperuser**
        - This command is for creating the admin user
    - **python manage.py runserver_plus --cert-file cert.pem --key-file key.pem --insecure**
        - Use this command to run the application
- Open a web browser and go to **https://127.0.0.1:8000/** to view the application.

Bonus: Running logging through syslog

- Download and install Syslog Watcher Manager through this: link
- After setting everything up, go to configurations and change the IP address to your local IPv4 address.
- Go to settings.py and find the syslog handler and input your Syslog server IP address.

You're all set!