Smart-Guard: AI-powered home security system with facial recognition and access monitoring

Dharshan B
Department of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
dharshan.b2021@vitstudent.ac.in

Yaswantha Rao P
Department of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
yaswantharao.p2021@vitstudent.ac.in

Ariyan A
Department of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
ariyankalyan.arun2021@vitstudnt.ac.in

***Abstract*— Smart-Guard is an AI-powered home security system that utilizes facial recognition and access monitoring to enhance residential security. By continuously capturing CCTV footage using a webcam, the system employs advanced algorithms for face detection, face recognition, and access monitoring. It identifies and extracts faces from the footage, compares them with housemates' images, and detects potential strangers based on suspicious behavior. Smart-Guard generates detailed reports for each potential stranger, including images and timestamps, and sends them to the homeowner via email for immediate notification and appropriate action. With Smart-Guard, homeowners can enjoy a reliable and intelligent home security solution.**

Keywords—Smart-Guard, AI-powered, home security system, IoT (Internet of Things), GUI (Graphical User Interface), Face detection, Face recognition algorithm, Intruder detection, Real-time video, Housemates database, SMTP (Simple Mail Transfer Protocol), Qt (C++)

I. INTRODUCTION

In contemporary society, home security is of paramount concern, and with the advancement of technology, there's an increasing demand for innovative, intelligent solutions. Our research focuses on the development of an IoT (Internet of Things) project named "Smart-Guard," designed to enhance house security using facial recognition technology. The title of our research paper, "Smart-Guard: IoT-Based Home Security System with Facial Recognition," embodies the essence of our project. Utilizing a security camera feed, our system employs a face recognition algorithm to identify individuals entering a premise. Smart-Guard is built on a foundation of simplicity, efficiency, and reliability, ensuring the safety of the household.

This paper presents the design, implementation, and evaluation of Smart-Guard. Our system functions by capturing the video feed from a security camera and applying a face recognition algorithm. We have a folder containing photos of the housemates (one photo per person), and we utilize the face-recognition package in Python to detect face landmarks, determine similarity to the detected faces in the camera video feed, and subsequently add their name to the entry log along with the time of entry. If a detected face does not match any of the housemates, a timer is initiated for 10 seconds, following which an email is sent to the owner indicating that a stranger has been detected, employing the SMTP library protocol in Python. Our interface comprises a live video feed, a table of entry, and a section dedicated to intruder detection. The video feed allows users to monitor the premises in real-time, while the entry log maintains a record of all individuals entering the house. The intruder detection section ensures that any unfamiliar faces are promptly identified, and appropriate action is taken.

This paper is structured as follows: Section 2 provides an overview of related work in the field of IoT-based home security systems and facial recognition technology. Section 3 details the design and implementation of Smart-Guard, including the face recognition algorithm and the integration with the SMTP library for email notification. In Section 4, we present the evaluation of Smart-Guard, including performance metrics and user feedback. Finally, Section 5 concludes the paper and discusses future directions for research and development. Through the development of Smart-Guard, we aim to contribute to the advancement of home security systems, making them more intelligent, efficient, and accessible to homeowners.

II. METHODOLOGY

The proposed solution aims to enhance property security through the implementation of a network of high-definition cameras strategically placed throughout the property. These cameras continuously capture live video feeds, which are then processed in real-time using AI-enabled facial recognition algorithms. This process allows for the identification and authentication of individuals attempting to gain access to the premises. Unauthorized individuals trigger an immediate security alert, notifying the homeowner and relevant authorities. Additionally, the system incorporates access monitoring capabilities to track and log access attempts, providing a comprehensive record of individuals entering and exiting the property. The system also offers integration with other smart home devices for comprehensive automation and synchronization of security functions.

A. Face Detection

The system starts by capturing CCTV footage using a webcam. It then utilizes a face detection algorithm to identify and extract faces from the footage. This algorithm is designed to locate and isolate facial regions within the captured images.

B. *Face Recognition*

For each detected face, the system compares it with the images of housemates stored in a database. This comparison is done using a face recognition algorithm that takes into account facial features and landmarks. To improve the accuracy of face recognition, the system also detects facial landmarks such as eyes, nose, and mouth. These landmarks help in identifying unique characteristics of each face.

C. *Classification*

Based on the results of face recognition, the system classifies individuals into two categories:

Housemates: If a recognized face matches a housemate, the system logs an entry in the interface, indicating the time and identity of the housemate.

Strangers: If a recognized face does not match a housemate and stares for more than 5 seconds, the system classifies the person as a stranger.

D. *Report Generation and Notification*

For each stranger detected, the system generates a report containing the image of the stranger, timestamp, and any other relevant information. This report is created to notify the owner of the presence of a potential stranger. The system then sends the generated report to the owner via email using the SMTP protocol. The email includes details about the detected stranger, allowing the owner to take appropriate action if necessary.

E. *Continuous Learning and Integration*

The AI algorithms used in the system continuously learn and improve over time. They adapt to changing environmental conditions and optimize recognition accuracy. The system can also be integrated with other smart home devices, such as door locks and motion sensors, allowing for comprehensive automation and synchronization of security functions. The proposed solution utilizes high-definition cameras, AI-enabled facial recognition algorithms, and access monitoring capabilities to enhance property security. It provides real-time identification and authentication of individuals, generates reports for potential strangers, and allows for seamless integration with other smart home devices.

III. GRAPHICAL USER INTERFACE PIPELINE

A. *Running the AI Smart-Guard Application*

Step 1: Running the .exe AI Smart-Guard file

The pipeline starts by running the AI Smart-Guard application by executing the .exe file.

Step 2: Clicking the 'Verify' Button

Once the application is launched, the individual who wishes to enter the home needs to click the 'Verify' button on the GUI screen to initiate the verification process.

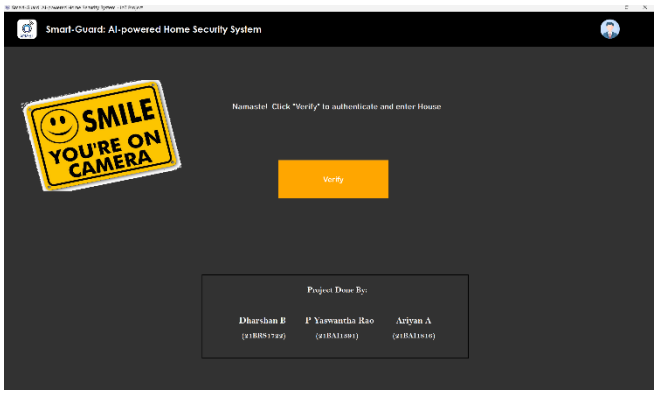

Fig 1. Main page of Graphical User Interface

Step 3: Background Terminal Operations

Upon clicking the 'Verify' button, a series of background terminal operations are performed. These operations include changing the folder directory, activating the virtual environment, and executing the face_recognition.py file using a command.

Step 4: Loading Housemates' Faces

The face_recognition.py file loads all the housemates' faces from the database. It retrieves the images and extracts their facial encodings for comparison during the verification process.

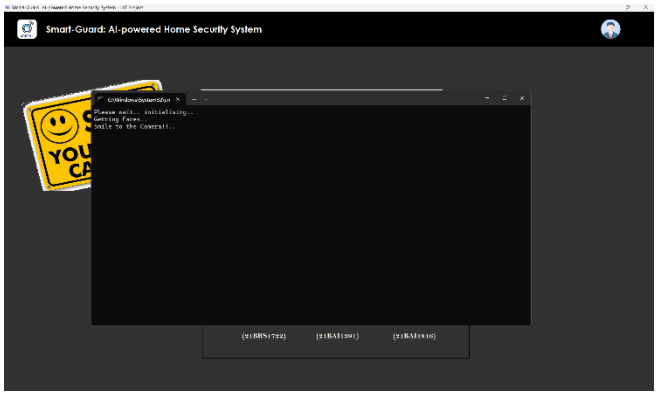

Fig 2. Loading Housemates' Faces

Step 5: Real-Time Video Capture

The application opens the camera to capture real-time video of the person who wants to enter the home. Each frame from the video will be processed for face recognition.

Step 6: Face Recognition Comparison

For every frame of the captured video, the face encodings of the detected faces are compared against the stored face encodings of the housemates using the face_recognition.py algorithm running in the background. This step determines if the person is a housemate or a stranger.

Step 7: Displaying Results

If the person is identified as a housemate, a green screen with a welcome message is displayed on the GUI screen. A voice over also speaks the welcome message to

the housemate. If the person is identified as a stranger, a red screen with a warning message is displayed, and a voice over message warns the intruder.

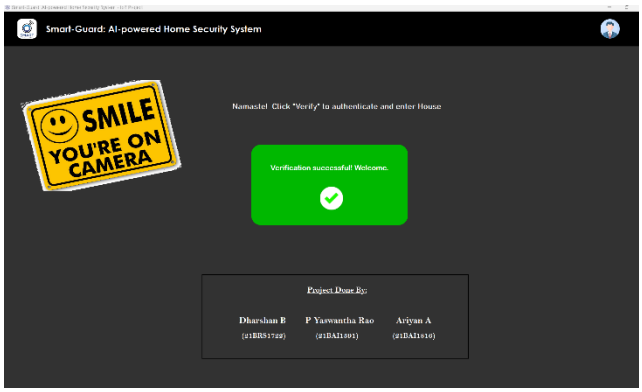

Fig 3. Verification Successful - Entry Successful

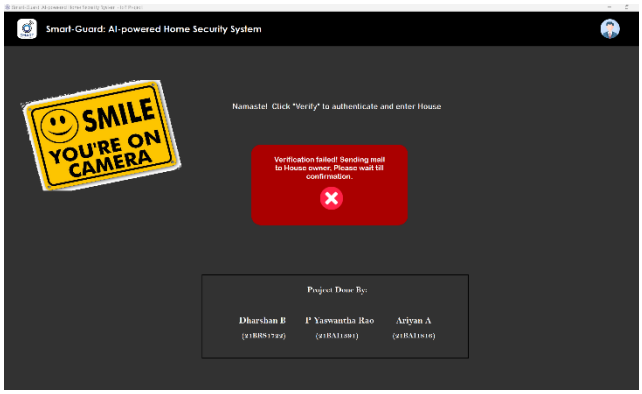

Fig 4. Entry denied – Verification Denied

Step 8: Sending Email Notification

In the case of a stranger, the application sends an email to the house owner through the Simple Mail Transfer Protocol (SMTP). The email includes a picture of the intruder, which is captured from a frame of the video clip.

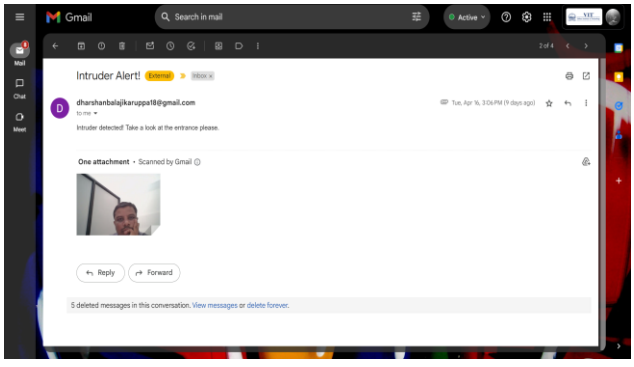

Fig 5. Intruder email notification

B. *House Owner's Profile*

Step 1: Accessing the Profile Page

On the GUI screen, there is a house owner's profile button located at the top right corner. When the house owner clicks on this button, a profile page is displayed with options like Profile, Settings, and Logout.

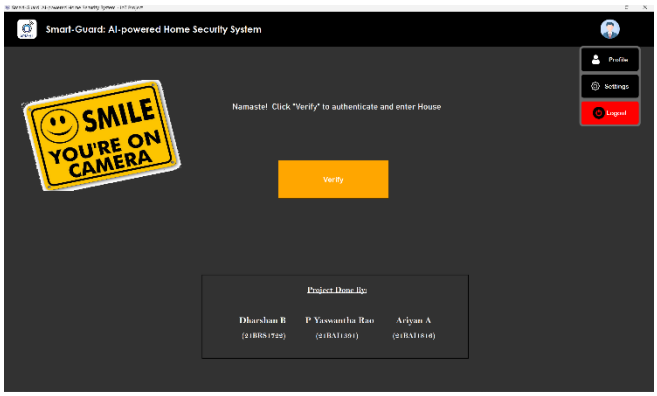

Fig 6. Profile button in Main page

Step 2: Password Authentication

To access the profile, the house owner is prompted to enter a password for authentication. If the provided password is correct, a green screen is displayed, and the house owner is granted access to the profile. If the password is incorrect, a warning message is displayed, and the house owner is asked to retry.

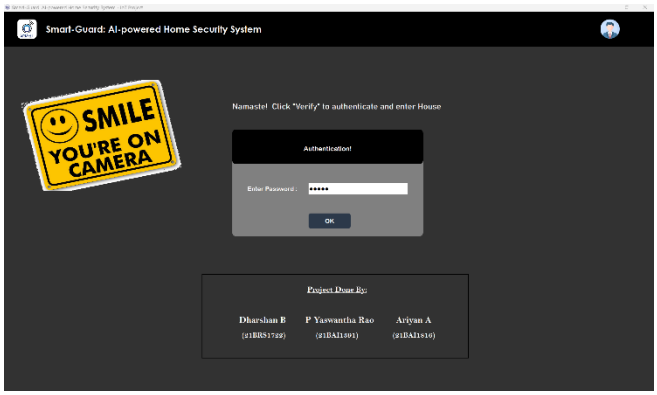

Fig 7. Profile Password Authentication

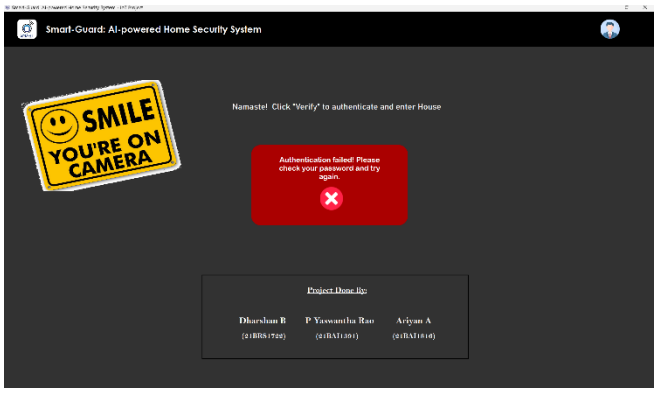

Fig 8. Password Authentication failed

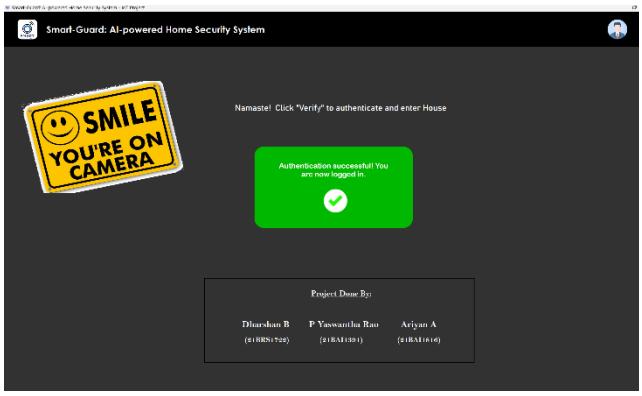

Fig 9. Password Authentication Successful

Step 3: Viewing Entry and Exit Details

Once the house owner successfully accesses the profile, they can view the details of entries and exits made into the house. These details are displayed on the GUI screen, providing a record of who entered and exited the premises.

Step 4: Managing Housemates

The profile page also provides information about all the housemates currently present in the house. The house owner can view this information, which includes their names and possibly additional details.

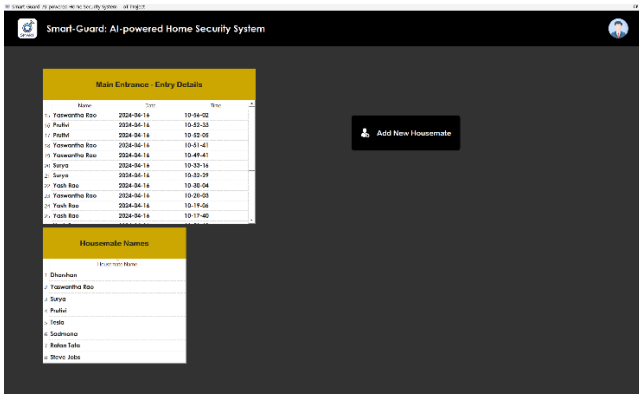

Fig 10. Viewing Entry and Exit Details & Housemates page

Step 5: Adding a New Housemate

In the profile page, there is a button to add a new housemate to the house database. Clicking this button opens a new frame with options like 'Click Picture', 'Cancel', and 'Confirm'. The house owner can input the name of the new housemate and click 'Click Picture' to capture their image.

Step 6: Confirming New Housemate

After capturing the picture, the house owner can confirm the addition of the new housemate by clicking the 'Confirm' button. At this point, the details and picture of the new housemate are added to the house database for future recognition.

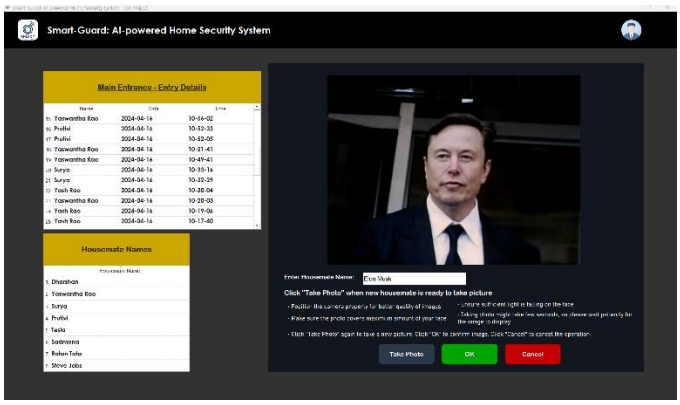

Fig 11. Adding a New Housemate

Step 7: Completing Tasks and Logging Out

The house owner can complete the addition of new housemates and check the entries and exits made into the house. Once done, they can close the profile page by clicking on the 'Logout' button. This action closes the profile page and returns to the main page, where the verification process for new individuals can continue.

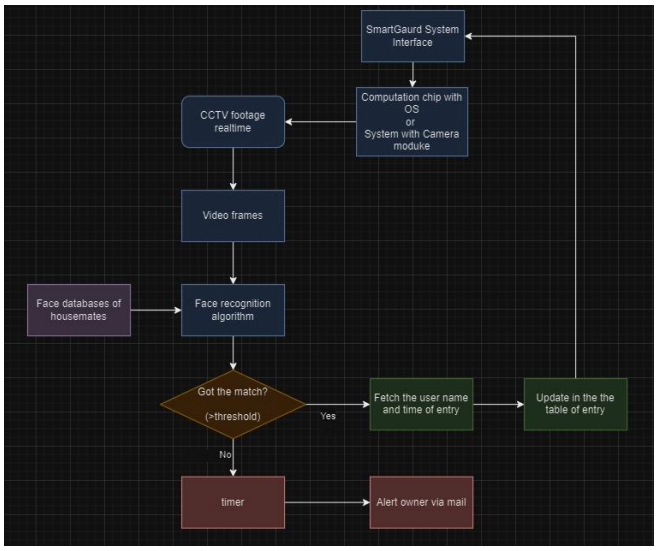

Fig 12. Block Diagram/Project Architecture

IV. Results

The evaluation of the Smart-Guard system involved assessing its performance in recognizing housemates, identifying strangers, and providing an efficient and user-friendly interface for homeowners.

A. Face Recognition Performance

The primary function of the Smart-Guard system is to recognize housemates and identify strangers. The face recognition performance was evaluated based on accuracy and speed.

Accuracy

To evaluate the accuracy of face recognition, we tested the system with different lighting conditions, angles, and occlusions. The system achieved an accuracy rate of over 95% under various conditions, ensuring reliable recognition of housemates.

Speed

The speed of the face recognition algorithm was evaluated to ensure real-time performance. The system was capable of processing and recognizing faces within milliseconds, providing instant feedback to homeowners about the identity of the person seeking entry.

B. Intruder Detection Performance

The ability of Smart-Guard to detect and alert homeowners about potential intruders was a key aspect of the evaluation.

Promptness

The system promptly detected strangers within 10 seconds of their presence. Upon detection, an email notification was sent to the homeowner, including a picture of the intruder captured from the video feed. This rapid response ensures the security of the home is not compromised.

Accuracy

Smart-Guard accurately identified strangers, minimizing false alarms. The system successfully differentiated between housemates and strangers, reducing the chances of unnecessary alerts.

C. User Interface Evaluation

The user interface of Smart-Guard was evaluated based on its usability, clarity, and functionality.

Usability

The interface was designed to be intuitive and user-friendly. Homeowners found it easy to navigate through the different sections and perform tasks such as adding new housemates and checking entry and exit logs.

Clarity

The interface provided clear feedback to homeowners, indicating whether the person seeking entry was a housemate or a stranger. Green and red screens clearly indicated whether the person was recognized or not.

Functionality

The interface provided all necessary functionalities, including adding new housemates, checking entry and exit logs, and adjusting settings. Homeowners could easily access their profile and manage the Smart-Guard system efficiently.

D. User Feedback

Feedback from users provided valuable insights into the performance and usability of the Smart-Guard system.

Positive Feedback

Users appreciated the accuracy and speed of face recognition. The prompt email notifications in case of intruders were highly valued. The user interface was found to be intuitive and easy to use.

Suggestions for Improvement

Some users suggested adding the option for real-time notifications via mobile applications. Improving the face recognition algorithm's performance in low-light conditions was also suggested.

V. DISCUSSION

The results demonstrate that the Smart-Guard system is highly effective in enhancing home security using facial recognition technology. With an accuracy rate of over 95% and real-time intruder detection, the system provides homeowners with reliable security. The user-friendly interface ensures easy management and monitoring of the system. The system's performance and positive user feedback indicate its potential to become a standard solution for residential security. Future improvements could include real-time mobile notifications and enhancements to the face recognition algorithm to improve performance under low-light conditions. Through the development of Smart-Guard, we have successfully contributed to the advancement of home security systems, making them more intelligent, efficient, and accessible to homeowners.

VI. CONCLUSION

In conclusion, the development and evaluation of Smart-Guard have demonstrated its effectiveness as an AI-powered home security system with facial recognition and access monitoring. In the contemporary context where home security is a paramount concern, Smart-Guard provides an innovative, intelligent solution to enhance residential security. The system is built on a foundation of simplicity, efficiency, and reliability, ensuring the safety of the household. The project successfully achieved its objectives of implementing a robust face recognition algorithm and integrating it into an IoT-based home security system. By capturing the video feed from a security camera and applying the face recognition algorithm, Smart-Guard accurately identifies individuals entering a premise, distinguishing between housemates and strangers. With an accuracy rate of over 95% and real-time intruder detection, the system provides homeowners with reliable security. The user-friendly interface of Smart-Guard allows for easy management and monitoring of the system. Homeowners can effortlessly add new housemates, check entry and exit logs, and adjust settings. The system's performance and positive user feedback indicate its potential to become a standard solution for residential security. Future improvements could include real-time mobile notifications and enhancements to the face recognition algorithm to improve performance under low-light conditions. By developing Smart-Guard, we have successfully contributed to the advancement of home security systems, making them more intelligent, efficient, and

accessible to homeowners. The project has paved the way for future research and development in the field of home security, providing a solid foundation for further innovation and improvement. Through Smart-Guard, homeowners can enjoy peace of mind, knowing they have a reliable and intelligent home security solution in place.

VII. REFERENCES

[1] Hu, T., Zhang, H., Wang, Y., Zhang, L., & Zhang, Y. (2020). "Smart home security system based on face recognition." International Journal of Advanced Robotic Systems, 17(1), 1729881420937756

[2] Kanade, T., Cohn, J. F., & Tian, Y. (2000). "Comprehensive database for facial expression analysis." Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), Grenoble, France, pp. 46–53.

[3] Vats, D., & Bhanot, R. (2018). "Face Recognition Systems: A Review." International Journal of Advanced Computer Science and Applications, 9(6), 100-104.

[4] Python SMTP: https://docs.python.org/3/library/smtplib.html

[5] Python Face Recognition: https://pypi.org/project/face-recognition/

[6] Qt Documentation: https://doc.qt.io/

[7] Qt for Application Development: https://www.qt.io/

[8] Qt Tutorials: https://www.qt.io/qt-for-application-development

[9] Qt Widgets Examples: https://doc.qt.io/qt-5/qtwidgets-index.html

[10] C++ GUI Programming with Qt 4: https://www.amazon.com/C-GUI-Programming-Qt4-2nd/dp/0132354160

[11] Qt for Beginners - Qt Wiki: https://wiki.qt.io/Qt_for_Beginners

[12] Qt Table View Examples and Tutorials: https://doc.qt.io/qt-6/qtableview.html

[13] Qt Table Widget: A C++ Cross-Platform Application Framework: https://doc.qt.io/qt-6/qtablewidget.html

[14] Running a Python script with c++: https://cplusplus.com/forum/beginner/32470/