

Secure PI Final Project Report

By:Anton Issaev, Joseph Margulis



Table Of Contents

Table Of Contents

Executive Summary

Introduction

Functional Features of the Product

Specifications of the Product

Operating Instructions

Product Design, Implementation, and Operation of the System

- a) System block diagram and software UML diagram
- b) Component images and component description
- c) Captures of the major GUIs that you created in your project (or those of third parties)
- d) Theory of operation of the entire system

Maintenance Requirements

Conclusion and Further Developments

Appendix

References

Contact Information

Executive Summary

SecurePI is an innovative project that utilizes a Raspberry Pi and Flask server to stream video from a camera onto an HTML website. The website features a secure login page and displays the camera output with facial recognition. Additionally, users can control the camera's movements with buttons on the HTML website, thanks to a Servo Motor. When a face is recognized, the system will automatically send an email with a picture of the person to the user.

The SecurePI project is designed to offer a comprehensive security solution for individuals. The Raspberry Pi's low cost and high performance make it an ideal choice for running the Flask server and streaming the camera footage. Flask is a lightweight and flexible web application framework that simplifies web application development. The facial recognition feature adds an extra layer of security to the system by allowing users to identify and track individuals who appear on the camera feed. This feature can be especially helpful for individuals who need to keep track of traffic in and out of a location.

Moreover, the Servo Motor control feature on the HTML website enables users to adjust the camera's viewing angle, making it easy to monitor different areas. The email alert system is another essential feature that provides users with instant notification when a face is recognized on the camera feed. In conclusion, SecurePI is an innovative and reliable security solution that offers users a user-friendly and cost-effective way to enhance their security measures. The project's features, such as facial recognition, servo motor control, and email alerts, provide an all-in-one security solution that is ideal for cost-conscious individuals looking to improve their security systems.

Introduction

SecurePI solves several security-related problems for individuals, including:

1. **Monitoring:** SecurePI enables individuals to monitor their surroundings remotely, providing an extra level of security and peace of mind.
2. **Facial recognition:** The system's facial recognition feature allows users to identify individuals appearing on the camera feed, providing an added layer of security.
3. **Control:** The ability to control the camera's movements with buttons on the HTML website and adjust its viewing angle with the Servo Motor control feature enhances the system's versatility, making it easier to monitor different areas and minimize blind spots.
4. **Email system:** The email system provides users with instant notification to their Gmail account when a face is recognized or unknown on the camera feed, ensuring that they can take quick action if necessary.

The viability of the product depends on several factors, including:

1. **Market demand:** There must be a demand for the product, and users must be willing to pay for it.
2. **Competitive landscape:** There may be similar products on the market, so SecurePI must offer a unique value proposition to stand out.
3. **Technical feasibility:** The product must be technically feasible to build and operate, and any regulatory or legal requirements must be met.
4. **Cost:** The product must be priced appropriately, taking into account the cost of production, marketing, and distribution.

We believe SecurePI meets the factors aforementioned and in today's market is indeed a viable surveillance solution for individuals.

Functional Features of the Product

SecurePI has several functional features that make it a comprehensive security solution for individuals. Some of its notable features include:

Camera and Video Streaming: SecurePI uses a camera module to stream high-definition video to a Flask server. The Flask server then serves the video stream to an HTML website, which can be accessed from any device. The camera's video quality allows users to view their surroundings in high definition, providing greater detail and clarity.

Facial Recognition: SecurePI's facial recognition feature uses a facial recognition library such as OpenCV to detect and recognize faces in the camera feed. When a face is recognized, the system automatically captures a picture of the individual and sends it to the user via email. This feature provides an added layer of security by allowing users to identify and track individuals appearing on the camera feed.

Servo Motor Control: The system's servo motor control feature allows users to adjust the camera's movements with buttons on the HTML website. The servo motor can be used to change the camera's viewing angle, providing users with greater flexibility in monitoring different areas. The camera can be rotated horizontally by 90 degrees, allowing users to cover a wide range of angles.

Email Alert System: The system's email alert system provides users with instant notification when a face is recognized on the camera feed. This feature ensures that users can take quick action if necessary, such as calling the police or investigating further. The email notification includes a picture of the person's face, allowing users to identify the individual quickly.

Secure Login Page: The system's secure login page requires users to enter a username and password to access the camera feed. This feature ensures that only authorized individuals can access the camera feed and prevents unauthorized access.

In conclusion, SecurePI's functional features make it an excellent security solution for individuals seeking a cost-effective and user-friendly way to enhance their security measures. The combination of high-definition video streaming, facial recognition, servo motor control, email alerts, and a secure login page provides users with a robust and versatile security solution that can be customized to meet their specific needs.

Specifications of the Product

Hardware:

- Raspberry Pi 3 Model B+
- Raspberry Pi Camera V2
- SG90 Servo Motor

Software:

- Raspbian operating system(Buster 32-Bit Legacy OS)
- Python3 programming language
- OpenCV computer vision library
- Pandas library
- XIRS and XLWS libraries (Excel sheet reading and writing)
- MailGun API (Email Sending Client)

Network:

- WiFi connection to the internet
- Email account for alerts (Gmail only)
- Flask server

Functionality:

- Video recording
- Facial recognition using OpenCV
- Email alert system for face recognition events
- HTML-based web interface for camera control and live video feed viewing
- Database for storing sensitive information such as usernames and passwords.
- Servo motor control

These technical specifications allow SecurePI to provide users with a comprehensive security solution that is capable of monitoring and detecting activity, identifying individuals, and notifying users via email.

Operating Instructions

Setting up SecurePI involves several steps:

Set up the Raspberry Pi:

Begin by setting up the Raspberry Pi and installing the necessary software, including the Flask server and facial recognition software. This process involves downloading an older version of Buster OS and writing it to an SD card. Afterward, the Raspberry Pi is configured with basic settings such as SSH, password, and keyboard layout.

Connect the camera:

Connect the camera module to the Raspberry Pi. The Raspberry Pi has a camera connector that is compatible with a range of camera modules.

Set up the Servo Motor:

Connect the Servo Motor to the Raspberry Pi and adjust the camera's viewing angle. The Servo Motor can be connected to one of the Raspberry Pi's GPIO pins, which allows it to control the camera's movements.

Set up the email alert system:

Create a Gmail account and a MailGun API account. Configure the email alert system by providing the necessary email address, API, and Domain name located on the MailGun API. The email alert system sends an email notification to the user when a face is recognized in the camera feed.

Launch headshots.py and train.py:

Launch headshots.py to take pictures of the faces you want to be recognized in the database. Store the faces within a folder in the same directory. Then run train.py to “train” the facial recognition system.

Port Forward the Raspberry Pi:

Port Forward the Raspberry Pi to allow access outside the local network. This involves configuring the router to forward incoming requests to the Raspberry Pi.

Launch the Flask server:

Launch the Flask server to start the video streaming. This is done by running the flask server script using the Python3 command in the terminal.

Access the website:

Open the website in a web browser by copying the Flask URL or the public port forwarded IP and enter the login credentials.

Change username/password:

Use the designated button on the main surveillance page called “Change User/Pass” to change your credentials. On the webpage, Enter your desired credentials. Click the “Change User/Pass” button once again when finished.

Set Up Email:

Use the designated button on the main surveillance page called “Setup Email” to set up the Email notification system. On the webpage, Enter your Email address, API, and Domain name provided by the MailGun Client (refer to step 4). Click the “Setup Email” button once again when finished.

Monitor the video feed:

Once logged in, you will see the live video feed from the camera. Use the buttons on the website to adjust the camera's viewing angle.

Monitoring facial recognition:

The system will automatically recognize faces in the camera feed and send an email notification to the user.

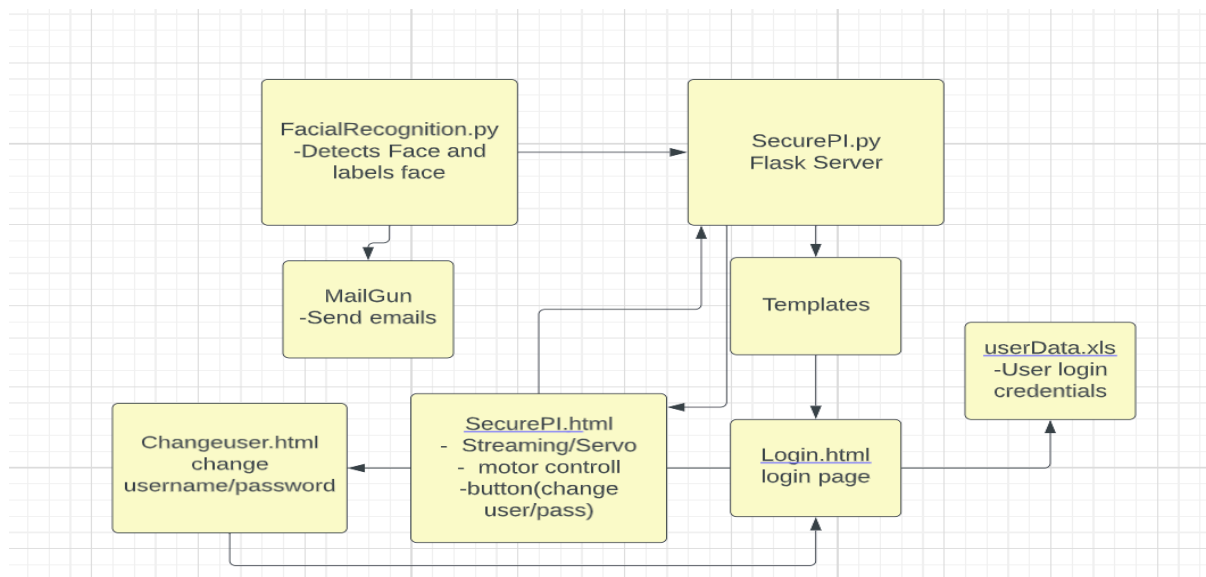
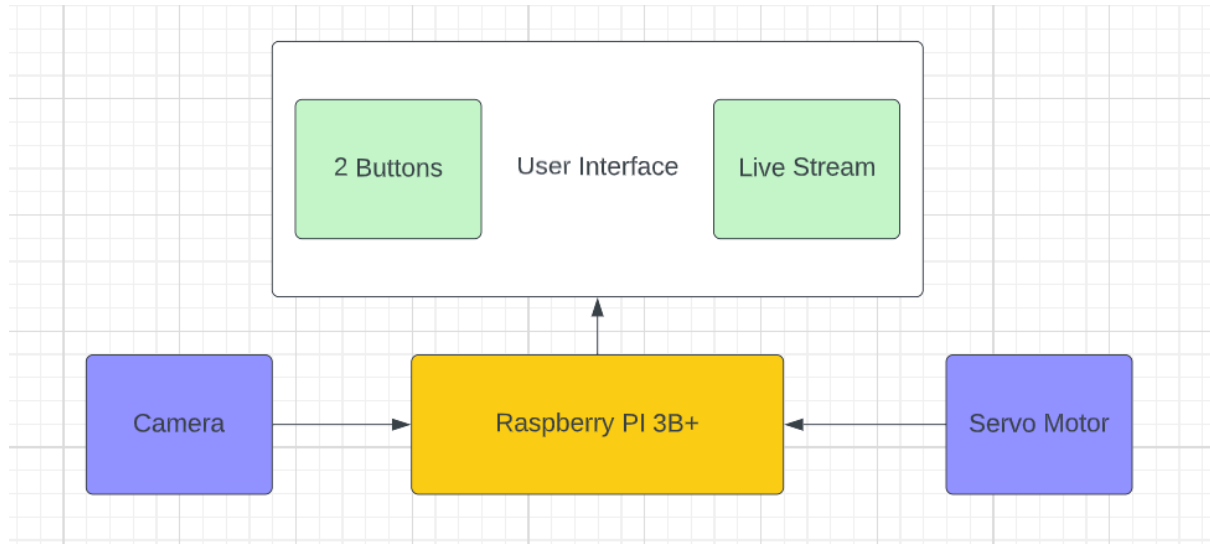
Log out:

When finished, log out of the website to secure your system.

It is essential to ensure that the system is always kept up to date with the latest security patches and updates. Additionally, it is important to follow best practices when it comes to password management, keeping login credentials secure, and restricting access to the system to authorized users only.

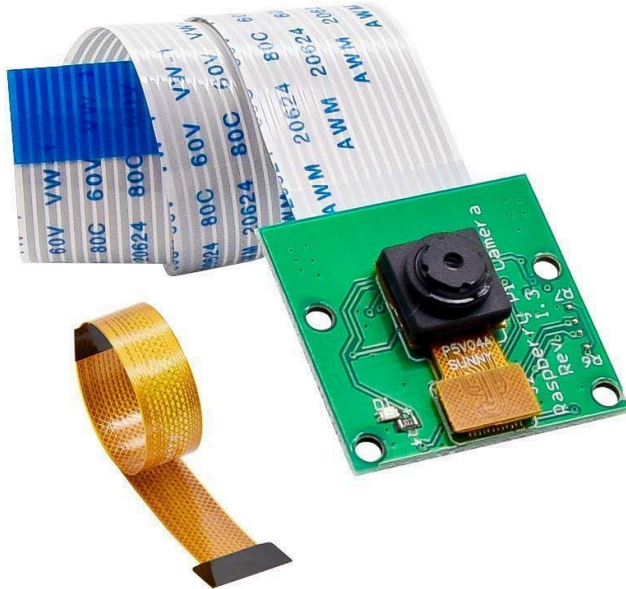
Product Design, Implementation, and Operation of the System

a) System block diagram and software UML diagram



b) Component images and component description

A. Raspberry Pi Camera



- 5MP Webcam Video 1080p Camera
- Low Light
- 30 fps

B. SG90 Micro Servo Motor



- Operating voltage: 4.8v
- Temperature range: 0°C - 55°C
- Power Supply: Through External Adapter
- Weight: 9g

C. Raspberry PI 3B+

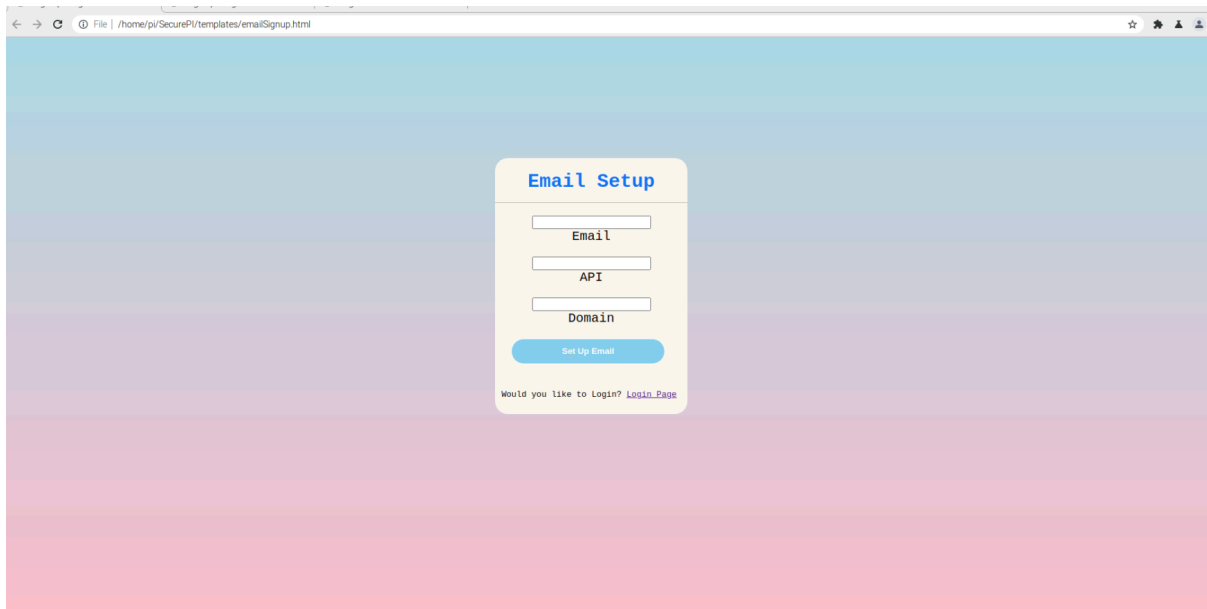
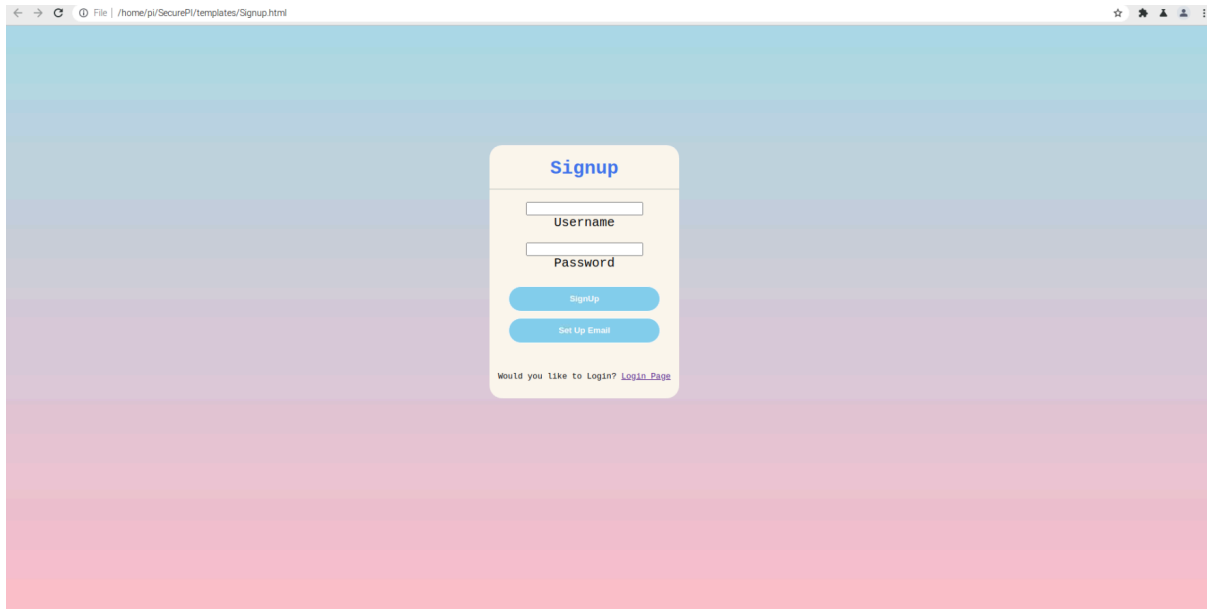


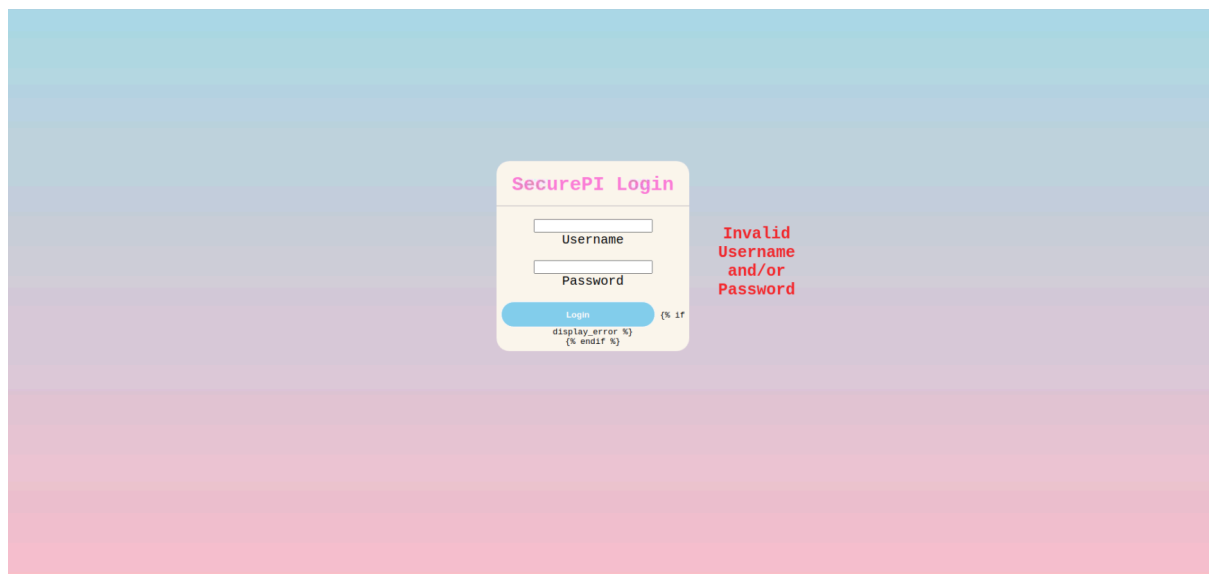
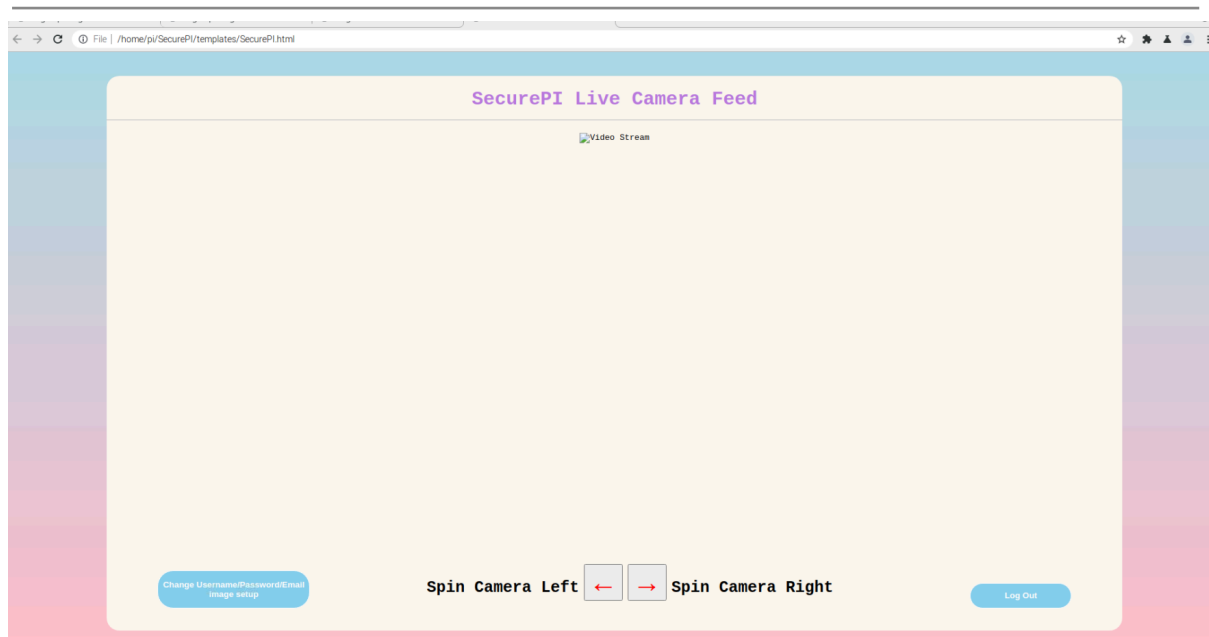
- 1.4GHz 64-bit quad-core processor
- dual-band wireless LAN
- Bluetooth 4.2/BLE, faster Ethernet
- Power-over-Ethernet support
- 1GB LPDDR2 SDRAM
- CSI camera port for connecting a Raspberry Pi camera

D. Camera Enclosure



c) Captures of the major GUIs that you created in your project (or those of third parties)





d)Theory of operation

Hardware setup:

First, connect the camera module and servo motor to the Raspberry Pi using appropriate connectors. The camera module and servo motor should come with their connectors, which should be compatible with the Raspberry Pi. Ensure that the Raspberry Pi is connected to power, and has an SD card with an old version of Buster OS installed. A guide on how to install Buster OS on a Raspberry Pi can be found at <https://core-electronics.com.au/guides/raspberry-pi/flash-buster-os-pi/>.

Install OpenCV:

To install OpenCV on the Raspberry Pi, follow the instructions provided in this article: <https://core-electronics.com.au/guides/hand-identification-raspberry-pi/#Set>. If any errors arise during installation, try running the command "sudo apt-get install python3-opencv".

Install Flask:

To install Flask on the Raspberry Pi, use the command prompt. Type in "sudo apt-get install python3-flask" to install Flask. Flask is a lightweight and flexible web application framework that simplifies web application development.

Write the code:

After installing OpenCV and Flask, write the code for the Flask server and HTML website. The Flask server should have endpoints for streaming the camera feed onto the website and for controlling the servo motor. The HTML website should have a secure login page and display the camera output with facial recognition. Buttons should be added to the website to control the servo motor's movements. This code can be written in a code editor such as Visual Studio Code or Sublime Text.

Setup resolution scaling:

To ensure that the HTML website can be viewed on mobile devices, the website must be scaled using CSS. The code can be adjusted to fit the screen size of any device.

Set up email alert system:

To set up an email alert system that sends an email with a picture of the person when a face is recognized on the camera feed, use a third-party API client such as MailGun. Follow the instructions provided by the API client to set up the email alert system.

Set up Servo Motor Control:

To set up servo motor control, import the GPIO libraries as well as set up the PWM signal as global variables. Create designated flask functions for movement left, right, and center, as well as a function to calculate the angle at which the servo motor is pointing towards. Use HTML buttons to link the functions to their appropriate movement angle. Set the angle in those functions using a variable and call the function that calculates angles while also passing the appropriate variable. Do this for each movement function. Remember to put a sleep function in the change angle function to prevent unwanted servo motor movement (after the angle calculation code inside a function). Change the duty cycle before moving the servo motor and change it back to 0 after the movement is completed to stop the servo motor once it has reached the appropriate angle (inside angle function).

Set up website encryption:

To set up website encryption, Open the flask code and set a global variable that initializes a boolean variable to false. In the login function on Flask set a condition that checks if the username and password were entered correctly, and set the boolean variable to true. In every function, set a condition that checks if the boolean variable is set to true. If not, write a redirect code that would redirect the user back to the login page. This will prevent users from bypassing the login page.

Set up port forwarding:

To allow the Flask server to be accessible from any device anywhere, set up port forwarding by accessing your router and configuring the port forward. Ensure the Pi's address is static, as this will prevent the IP address from changing. The process for setting up port forwarding may differ depending on the type of router being used.

Test and debug:

Before deploying the project, test the system to ensure that it is working correctly. Debug any issues that arise during testing. This will ensure that the project runs smoothly once it is deployed.

Deploy the project:

Finally, deploy the project by running the Flask server on the Raspberry Pi. The website should be accessible from any device anywhere. This can be done by running the code on the Raspberry Pi.

Maintenance Requirements

1. Software updates: Ensure that the software is up-to-date with the latest version to avoid any potential vulnerabilities.
2. Hardware checks: Regularly check the camera, servo motor, and other hardware components to ensure they are functioning properly.
3. Clean the camera lens: The camera lens should be cleaned regularly to avoid any image distortion or blurriness.
4. Backup data: Regularly back up the data on the system to ensure that it is not lost in case of any system failures.
5. Check network connection: Ensure that the network connection is stable and secure.

By following these maintenance requirements, the system will operate optimally and provide the necessary security and peace of mind to users.

Conclusion and Further Developments

In conclusion, SecurePI is an impressive security solution that utilizes the Raspberry Pi and Flask server to provide a cost-effective, user-friendly, and comprehensive security system. The project's features, such as facial recognition, Servo Motor control, and email notifications, make it an all-in-one security solution that is ideal for individuals looking to enhance their security measures.

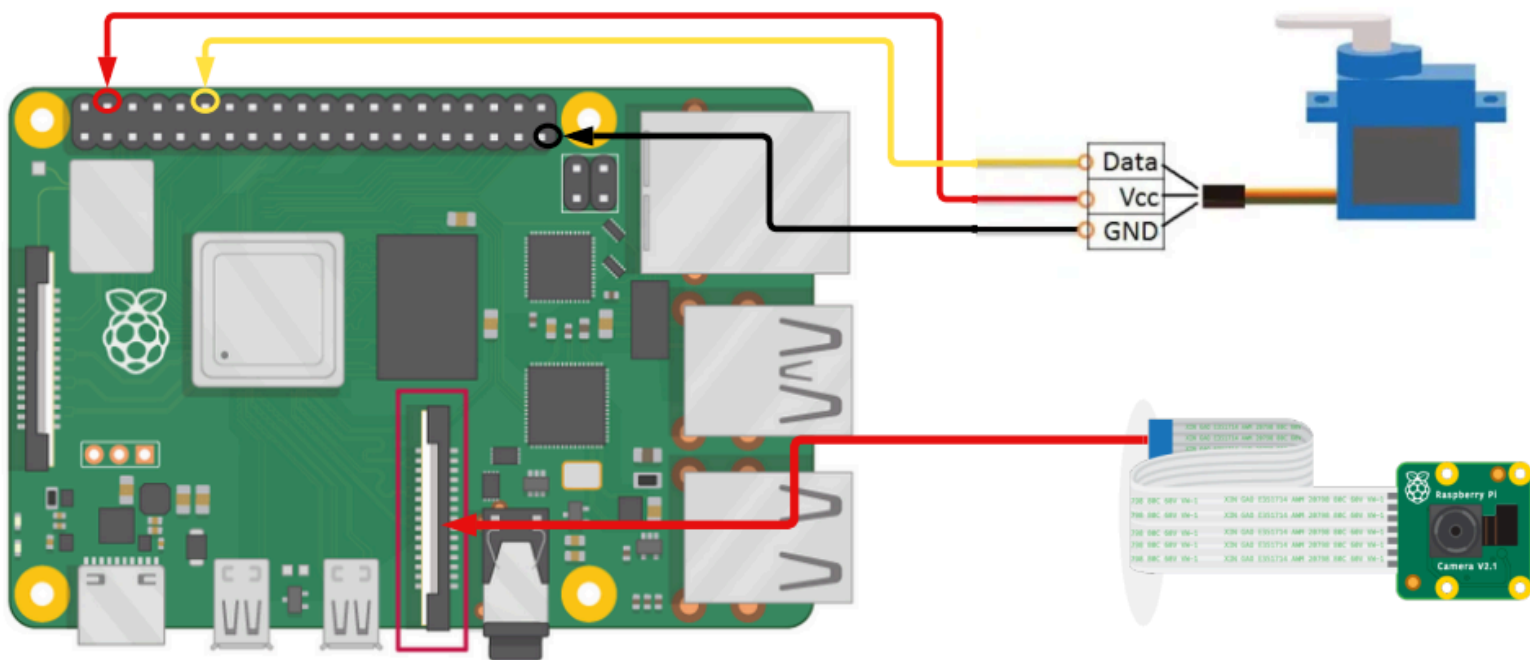
In terms of further developments, one potential area of improvement could be to add machine learning algorithms to the facial recognition feature, which could improve the accuracy of the system's facial recognition capabilities. Additionally, the system could be expanded to include more advanced features, such as motion detection or sound detection, to provide a more comprehensive “all in one” security solution.

Furthermore, the project could be developed to include integration with smart home devices or other security systems, allowing users to monitor their security system from a single platform. This integration could provide additional convenience and functionality, further enhancing the project's value to users.

Overall, SecurePI is an impressive project that offers a solid foundation for further developments and enhancements in the field of security systems. With its low cost, user-friendly design, and essential features, SecurePI provides an innovative and reliable security solution that is accessible to a wide range of users.

Appendix

Electrical Schematics



Parts List

Components	Quantity	Cost/Unit (CAD)	Total Cost (CAD)	LINKS
Raspberry Pi model 3B+	1	\$48.95	\$48.95	https://www.pishop.ca/product/raspberry-pi-3-model-b-plus/
Raspberry Pi Power Supply	1	\$9.95	\$9.95	https://www.pishop.ca/product/wall-adapter-power-supply-5-25v-dc-2-4a-usb-micro-b/
Raspberry Pi Camera Module v2	1	\$15.95	\$15.95	https://www.amazon.ca/Freenove-Camera-Raspberry-Adjustable-Holder/dp/B08Q34FKFY/ref=sr_1_12?crid=1W8EAHZ5HAQV8&keywords=raspberry+pi+camera&qid=1679098074&srefix=raspberry+pi+camera+%2Caps%2C270&sr=8-12
Camera Enclosure	1	\$14.19	\$14.19	https://www.amazon.ca/Etopar-Security-Waterproof-Flashing-Surveillance/dp/B07Q71FR58/ref=sr_1_24?crid=22C5AYTI16STU&keywords=blupont+cctv&qid=1674437013&s=electronics&srefix=blupont+cctv%2Celectronics%2C119&sr=1-24
Micro SD Card	1	\$9.00	\$9.00	https://www.amazon.ca/QUMOX-Micro-Flash-Memory-Class/dp/B07F81QTPP/ref=sr_1_8?crid=1PDMGT9HLSJ10&keywords=micro+sd+card+8gb&qid=1675391384&srefix=micro+sd+card+8g%2Caps%2C89&sr=8-8
Servo Motor SER0006	1	\$5.22	5.22	https://www.digikey.ca/en/products/detail/dfrobot/SER0006/7597224

Total: CAD: \$103.26

List of all Usernames and passwords used in your project

The username and passwords are all adjustable by the user. However, the default username is “SecurePI” and the password is “root”. Other than these two, there are no other login credentials used.

References

1. “For Raspberry Pi Camera, KEYESTUDIO 5MP 1080p Camera Module with OV5647 Sensor Video Webcam for Raspberry Pi Model A/b/B+, Pi 2, Raspberry Pi 3, 3b+, Pi 4b Ras Pi 4 Zero W Camera with Flex Cable.” *Amazon.ca: Electronics*,
https://www.amazon.ca/Keyestudio-Camera-Module-5MP-Raspberry/dp/B073RCXGQS/ref=sr_1_1_sspa?gclid=CjwKCAjw_MqgBhAGEiwAnYOAeuvYK-WlvHq5g103h3NBjTNaNxs_tQba8tg_H417gOWrTix-egmX8vBoCMDMQAvD_BwE&hvadid=208348502534&hvdev=c&hvlocphy=9000793&hvnetw=g&hvqmt=e&hvrnd=14155587320921138090&hvtargid=kwd-296167128860&hydadcr=1534_9454502&keywords=raspberry%2Bpi%2Bcamera%2Bmodule&qid=1679003164&sr=8-1-spons&psc=1&smid=A26TCVWBQE4D9T&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEuXQINFSVRVUzg0TlhlJmVuY3J5cHRlZElkPUEwMDc1ODEyM08zTk1UNkVCWDZaTiZlbnNyeXB0ZWRBZElkPUEwMzgyMDk0Q0tEUkpCV0VKN1hNjndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbGlja1JlZGlyZWNoJmRvTm90TG9nQ2xpY2s9dHJ1ZQ.
2. “Jooan Fake/Dummy Dome Security Camera Hemisphere Type Home/Store Surveillance Equipment with Twinkle Red Led.” *Amazon.ca: Electronics*,
https://www.amazon.ca/JOOAN-Security-Hemisphere-Surveillance-Equipment/dp/B01F37DKVA/ref=sr_1_6?crid=3LMTX9KFP6TI9&keywords=fake%2Bcamera&qid=1679003292&prefix=fake%2Bcamer%2Caps%2C272&sr=8-6.
3. “MMOBIEL SG90 9G Micro Servo Motor Kit for RC Drone/Car/Airplane/Helicopter/Vehicles/Robots/Compatible with Arduino Projects and Raspberry Pi Incl. Screws and 3 Servo Arms.” *Amazon.ca: Toys & Games*,
https://www.amazon.ca/MMOBIEL-Airplane-Helicopter-Compatible-Raspberry/dp/B097RD8RB7/ref=sr_1_10?gclid=CjwKCAjw_MqgBhAGEiwAnYOAepnskRCKgxLZQOUBRPTxCNalgyTn4RuUoDtU3Gg33ymQEKbMK4TERBoCTE8QAvD_BwE&hvadid=224680300718&hvdev=c&hvlocphy=9000793&hvnetw=g&hvqmt=e&hvrnd=11869733783773735554&hvtargid=kwd-298271520188&hydadcr=1503_9454474&keywords=sg90%2Bmicro%2Bservo&qid=1679003399&sr=8-10.
4. Raspberry Pi. “Buy A Raspberry Pi 3 Model B+.” *Raspberry Pi*,
[https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/.](https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/)

Contact Information



Anton Issaev

Engineer/Co-Owner

Developed and configured facial recognition for this project. Developed web server and UI of the web server. Configured email and API client.



Joseph Margulis

Engineer/Co-Owner

Developed and configured database I/O for this project. Developed and configured facial recognition for this project. Developed web server implementation. Configured Servo Motor control.



647-270-2934

josephmargulis@gmail.com