FPGA BASED UAV FOR SEARCH AND RESCUE OPERATIONS

Capstone Group 3

OVERVIEW

- Problem Statement
- Solutions
- Project Block Diagram
- Completed Blocks
- Development Hurdles
- Work Plan
- References

PROBLEM STATEMENT

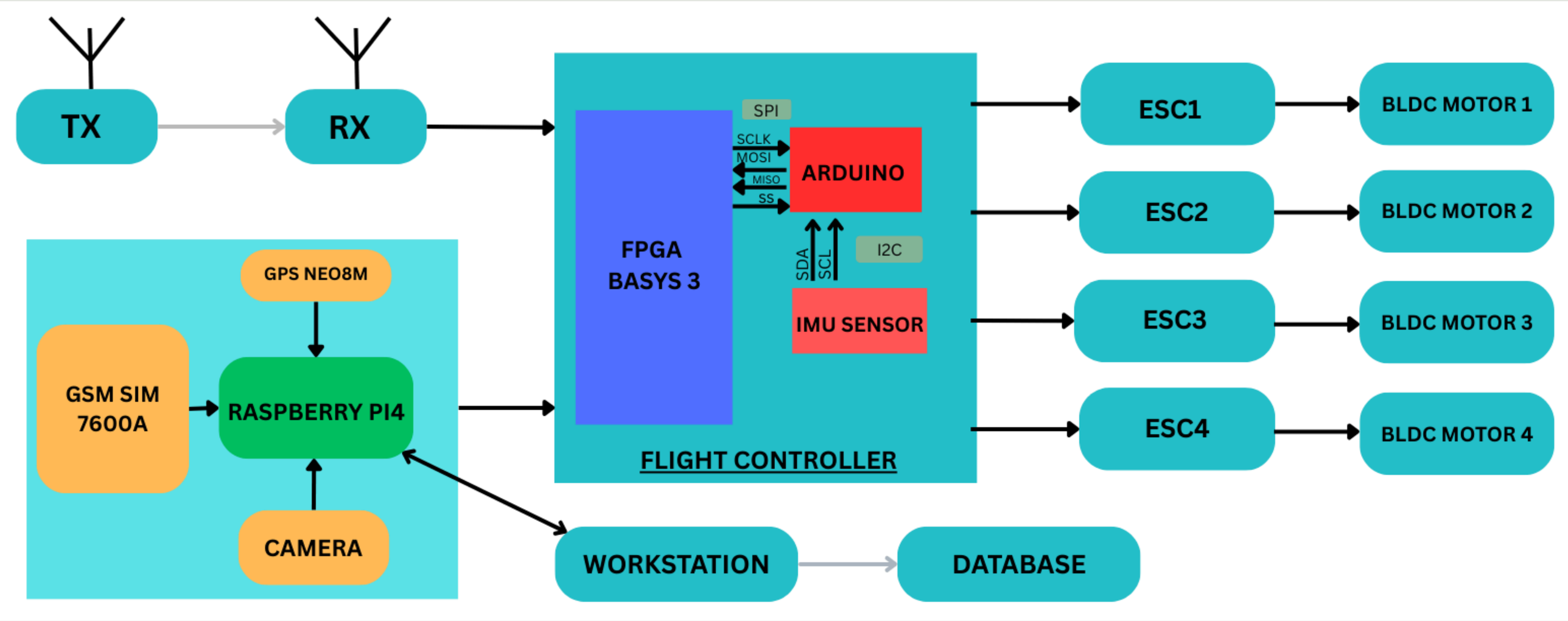
Policing in India is heavily constrained by an imbalanced personnel-to-civilian ratio, with only 152.80 personnel for every 100,000 civilians, far below the UN-recommended 222.[1] This shortage strains law enforcement agencies, hindering their ability to manage large-scale events, conduct surveillance, and respond to emergencies effectively. Traditional policing methods face additional challenges, including delayed response in search and rescue operations, limited situational awareness in large-scale missions, inadequate crowd control, and logistical constraints in accessing critical areas.

While India's defense services have made progress in UAV and UGV development, there remains a significant dependency on foreign components, including from China. This reliance exposes national security to risks, as recent incidents have shown vulnerabilities due to foreign breaches.[2] To address these challenges, there is an urgent need for the indigenous development of reliable, self-sufficient technologies to enhance the safety and effectiveness of India's police forces. Our group aims to resolve this issue by creating an indigenously developed flight controller for UAVs, ensuring secure, efficient, and scalable law enforcement operations.

SOLUTIONS

- Drones equipped with AI based recognition assist in locating individual in forests, urban areas or disaster zones, improving search efficiency.
- Drones can navigate dangerous environments, providing real-time visual imaging without risking human lives.
- Law enforcement requires real time surveillance for crime prevention and rapid response without deploying large number of personnel.
- Drones provide high altitude monitoring, allowing officers to track suspicious activities, monitor crime hotspots and respond to incidents more efficiently.

PROJECT BLOCK DIAGRAM



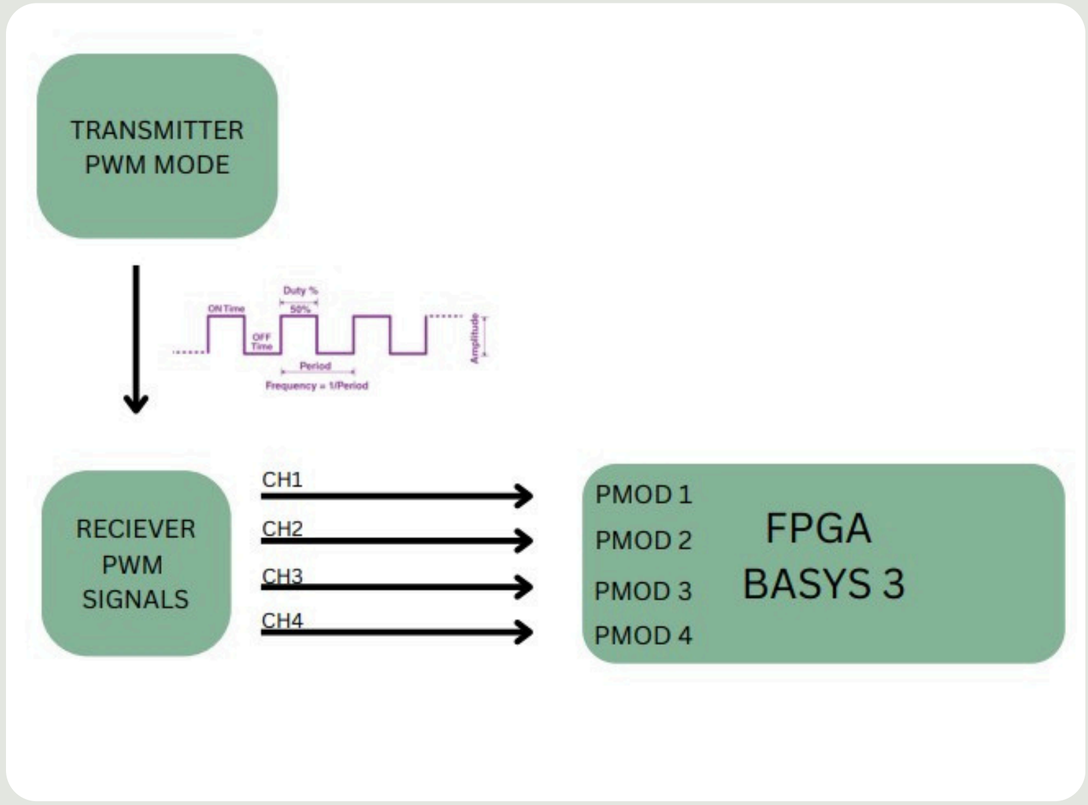
COMPLETED BLOCKS

- PWM Capture
- Motor Mixing
- PID (ONGOING)
- GPS Integration
- Facial Recognition Algorithms
- Person Detection Algorithms
- Frontend Design
- Backend Design

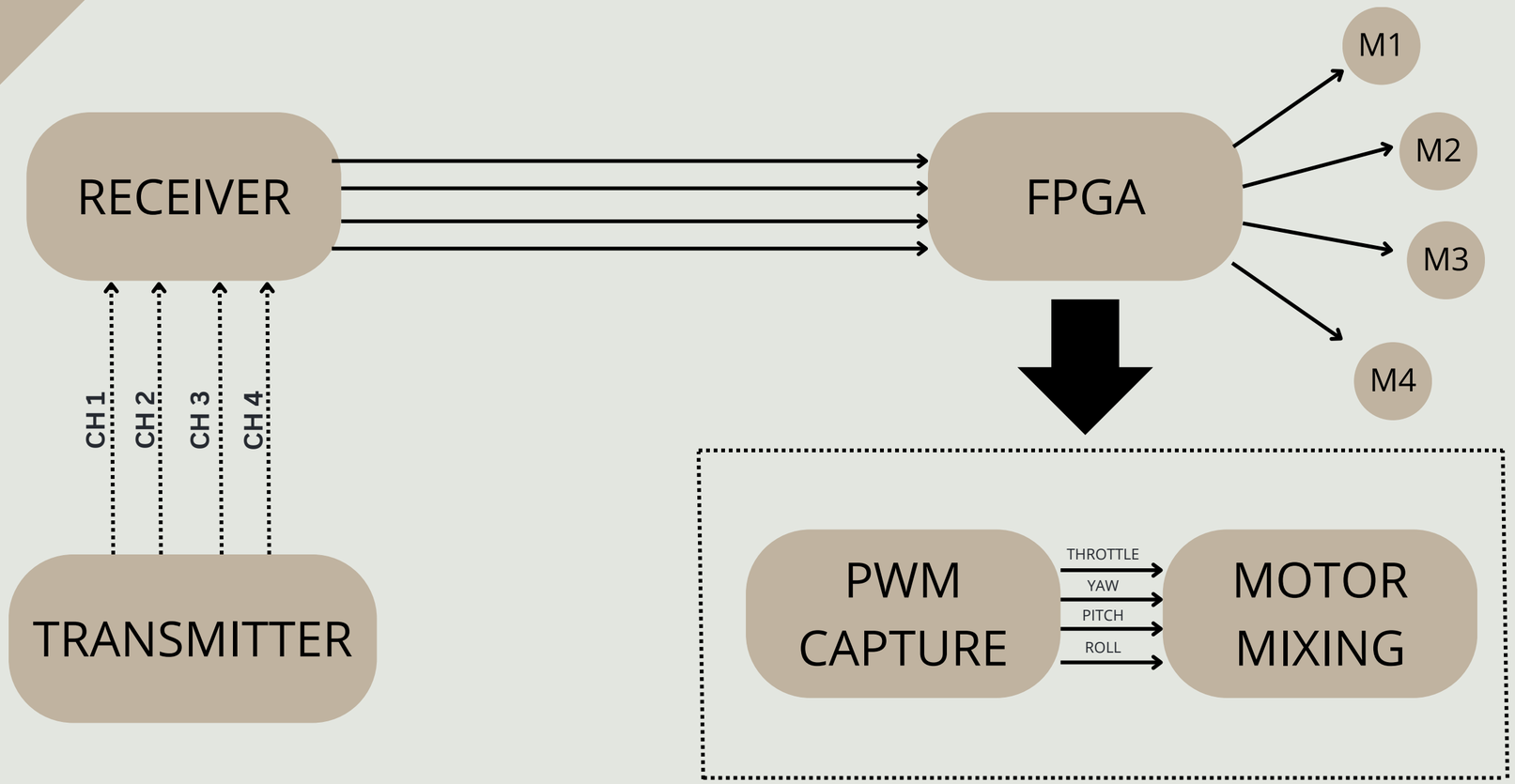
PWM CAPTURE

Working:

- This setup captures PWM signals from an RC transmitter operating in PWM mode.
- The receiver outputs individual PWM signals (CH1–CH4) corresponding to each control channel. These signals are then fed into the PMOD ports of the Basys 3 FPGA board.
- The FPGA captures the pulse width of each signal to decode control inputs such as throttle, pitch, yaw, and roll.



MOTOR MIXING



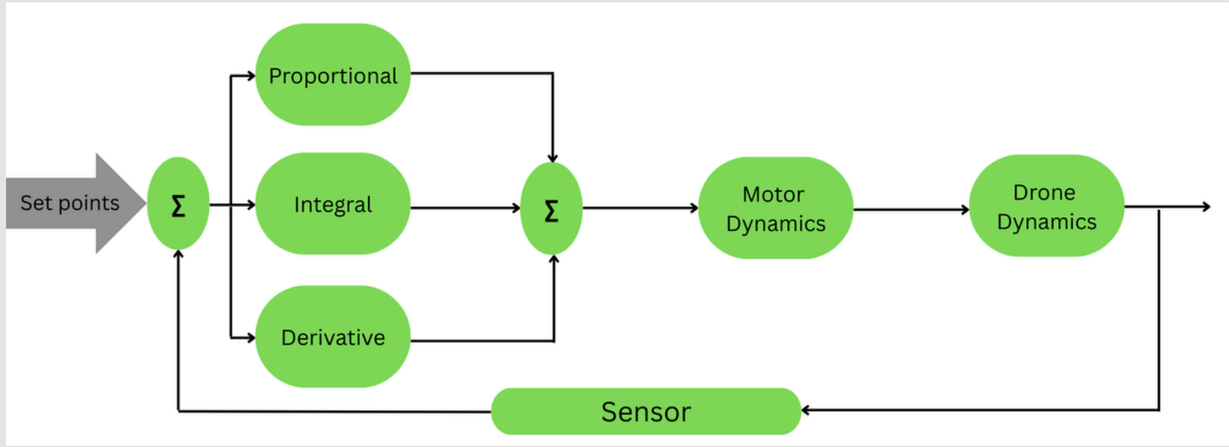
Working:

- Motor Mixing is the process of converting the pilot's control inputs — Throttle, Yaw, Pitch, and Roll — into individual speed commands for each of the quadco
- Throttle controls overall altitude.
- Yaw rotates the drone left/right (Z-axis).
- Pitch tilts it forward/backward (X-axis).
- Roll tilts it left/right (Y-axis).
- Each motor contributes differently based on its position and spin direction.
- The results are then constrained to PWM limits (e.g., 1000–2000 µs) and used to generate PWM signals that control motor speed.

PID

WORKING:

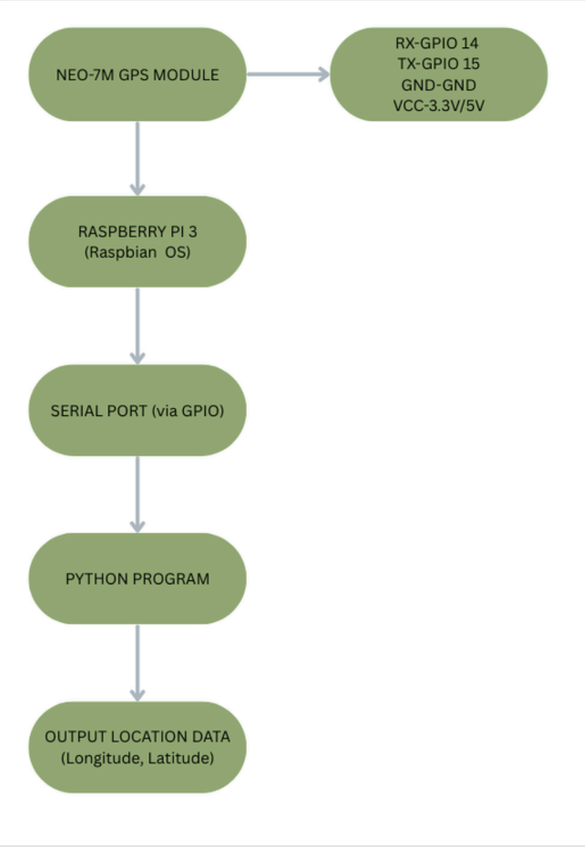
- Proportional (P): Corrects errors in direct proportion to their magnitude. If the error is large, the controller responds strongly; if the error is small, the controller responds gently.
- Integral (I): Accumulates past errors and counters persistent offsets. This term grows as long as a steady error remains, helping the system eliminate residual biases.
- Derivative (D): Reacts to the rate of change of the error. It predicts how the error is moving, providing a stabilizing “damping” effect that helps prevent overshoot and oscillations.
- By combining these three terms, PID controllers we can achieve precise, stable control for flight stabilization in drones. The main task is to tune the three gains(kp,ki,kd)so that we can control output responds quickly to changes without becoming unstable

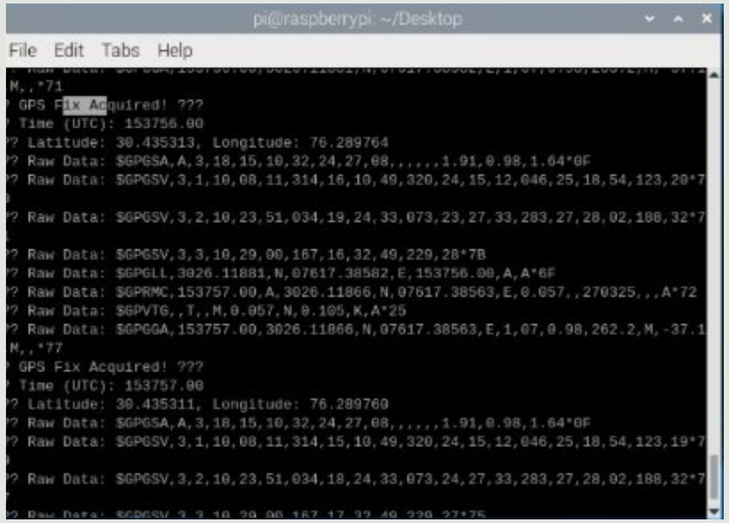


GPS INTEGRATION

Working:

- The GPS module receives signals from satellites to determine the device's location, providing accurate latitude and longitude coordinates.
- It communicates with the Raspberry Pi via serial communication.
- Python code running on the Pi reads and processes this data, allowing the location to be displayed and utilized in real-time applications.





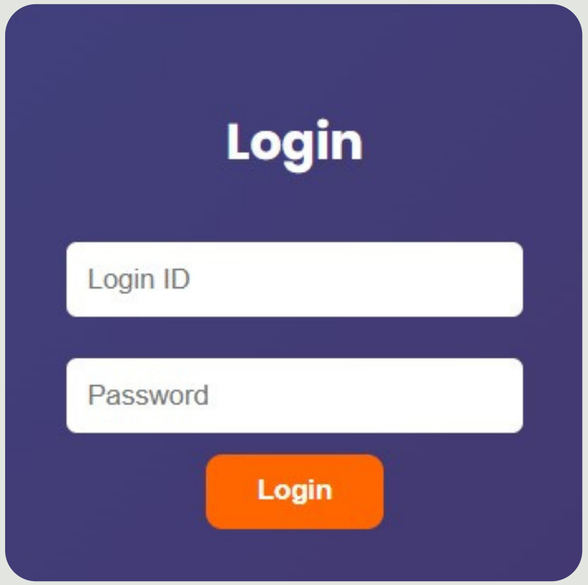
SOFTWARE

Frontend:

- Integrated a secure login system to restrict access.
- The main dashboard allows users to dynamically upload images, toggle between facial recognition and person detection features, and view a preview gallery of stored faces from the database.

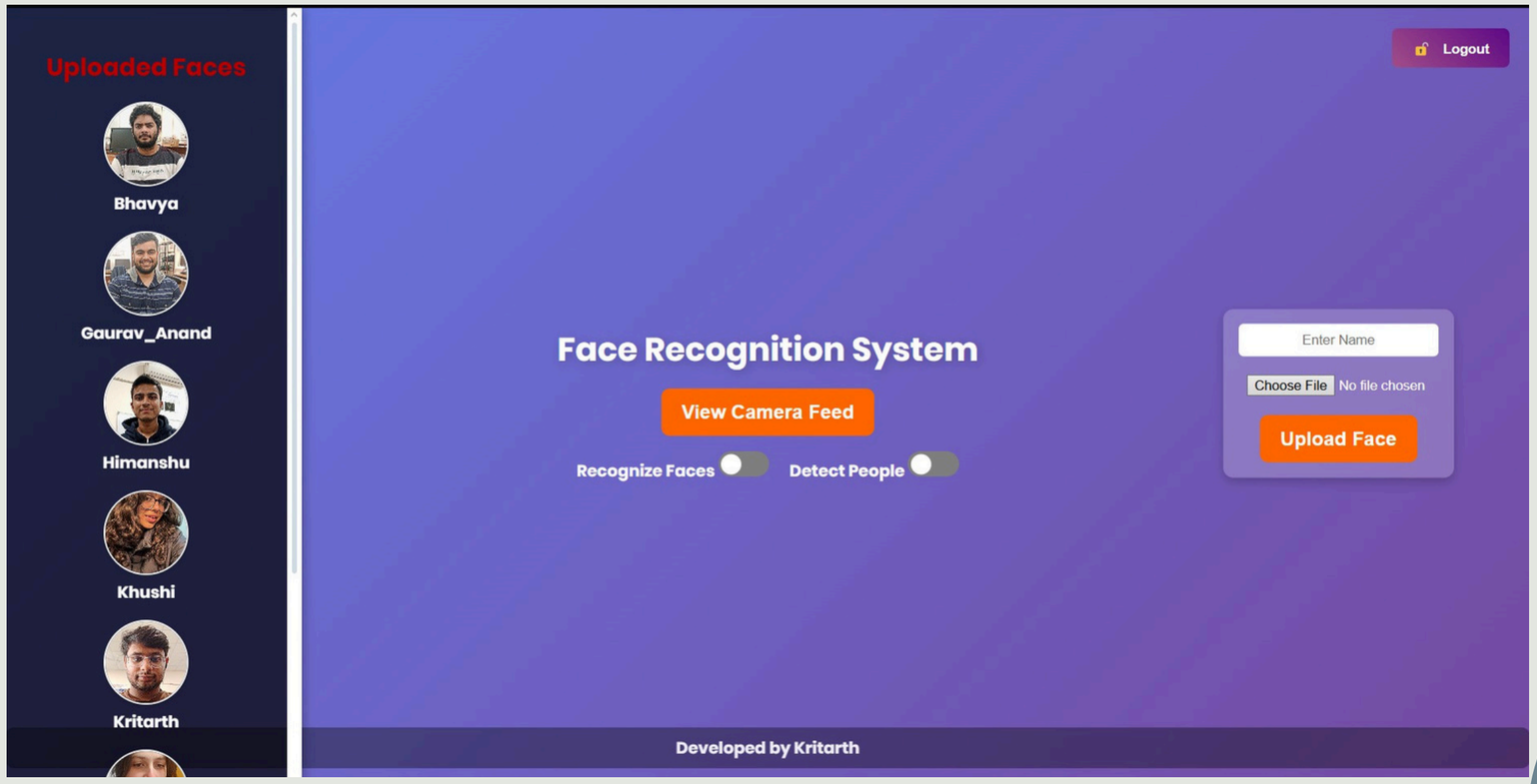
Backend:

- The backend is built using Python's Flask for robust and scalable development.
- User login credentials are securely stored in a MySQL database.
- To enable access across devices, the application is exposed to the internet using Ngrok tunneling.



Algorithms:

- We used the face_recognition library to detect and recognize faces in real-time by encoding known faces and comparing them with live video feed using deep learning-based face embeddings.
- For person detection, we used OpenCV’s HOG + SVM model, which identifies human figures in the video stream using pre-trained descriptors.



DEVELOPMENT HURDLES

Hardware

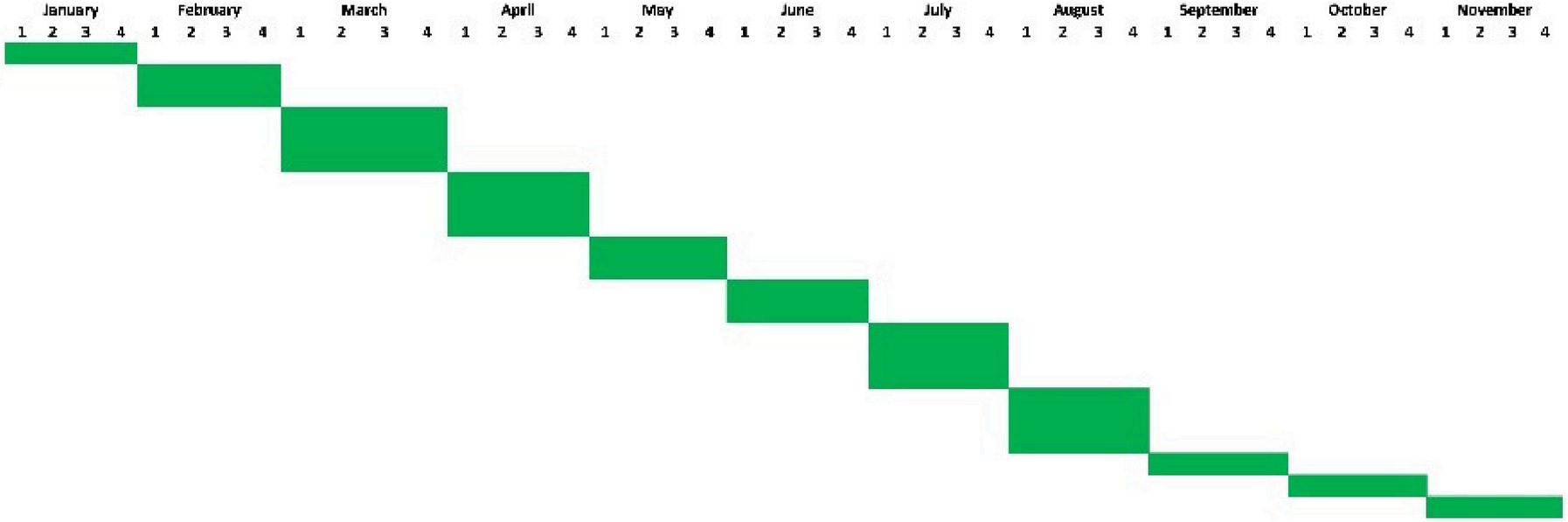
- Experienced delays in capturing PWM signals accurately, affecting response time and control precision.
- Mapping the correct PWM signals to their respective control channels was initially inconsistent, leading to misinterpretation of input commands.
- Fine-tuning PID parameters and converting their outputs into appropriate motor control signals required iterative adjustments for stable flight behavior.

Software

- Integrating toggle switches to control features like face/person detection in real-time required seamless backend coordination.
- Handling Flask session management and secure login authentication took careful implementation.
- Simultaneous access to the webcam by different processes sometimes caused camera initialization failures.

Work Plan

Sr. No.	Activity
1	Project Planning and Discussion with Mentors
2	Study of Hardware and Software Requirements, Familiarization with the FPGA board and related components
3	Study of various Communication processes and Protocools, Design of I2C Master Module, Design of Facial recognition algorithm.
4	Familiarization and Testing of Hardware Components, Design of Object detection and tracking algorithm.
5	Assembling the Quadcopter, Study of frontend and backend requirements for Web Application.
6	Design of Low Pass filter module, Designing Backend and Database for Web Application
7	Design of PWM Encoder Module, Design of MPU Driver Module, Designing Frontend for Web Application
8	Theoretical Study of PID and Calculation of desired Coefficients, Implementing automatic tracking in the drone
9	Design of Main Controller Unit Module
10	PID Coefficients Tuning, Verification
11	Optimizations and Modification, Finalizing Report



REFERENCE

[1] https://www.hindustantimes.com/india-news/policepublic-ratio-stands-at-152-80-per-lakh-person-govt-informs-parliament-101680162971094.html

[2] https://www.indiatoday.in/magazine/defence/story/20250210-chinese-threat-in-indian-drones-2672839-2025-01-31

Thank You