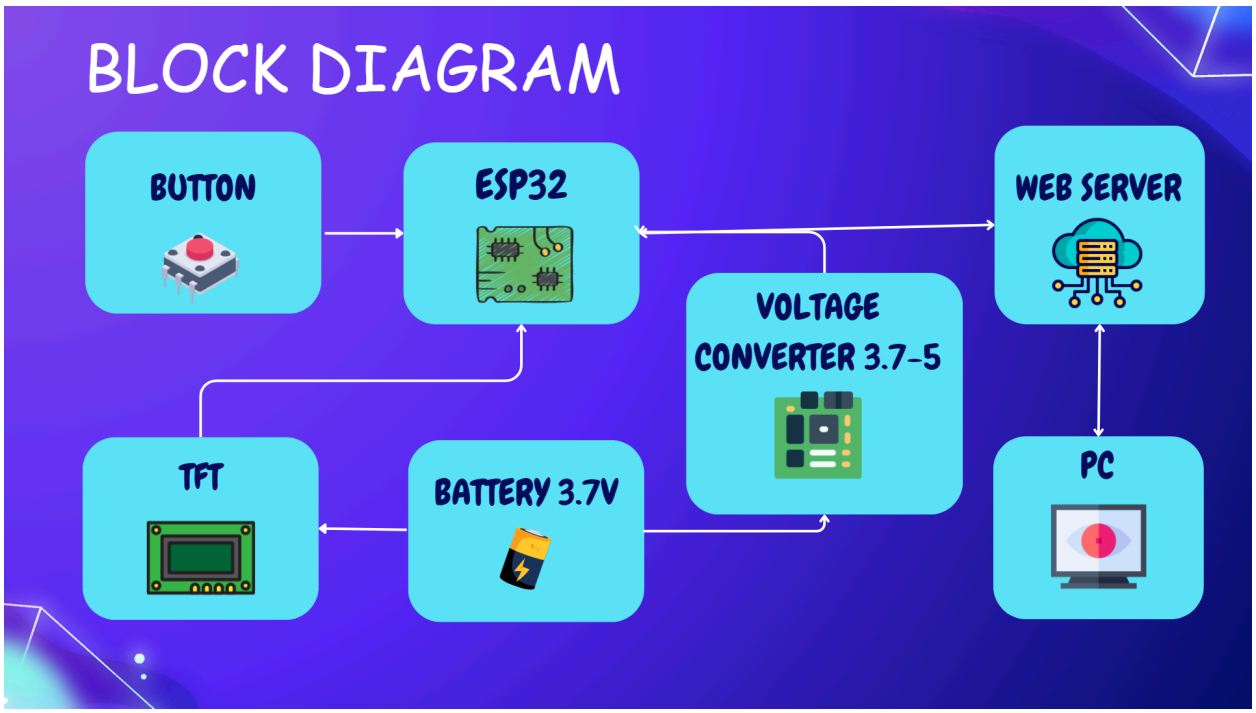
Methodology

The methodology section of our project documentation outlines the structured approach our team will take to achieve the goals set forth in developing the Smart Pen system. Our methodology encompasses several key stages, each designed to ensure efficient collaboration, effective problem-solving, and successful implementation of the project objectives. We use Waterfall Methodology for our project Methodology

1)Planning and Requirements Gathering
2)Hardware and Software Selection
3)System Design and Architecture
4)Development and Implementation
5)Testing and Quality Assurance

System Architecture

The various components that are being used in our project 'Smart Pen' can be seen in the diagram below.



Details about each component and its usability are as follows-
*1)**Button -**The button here is used for clicking pictures of the required text by long pressing it .There are 3 modes of operation in our project i.e. Display Meaning,calculate some mathematical operation and lastly display the graph of some function. The button here is used for switching the mode of operation by double clicking it.*

*2)**ESP32** -The ESP32 here is taking input from the button like which mode is it currently working on and when the button is long pressed it uploads the image clicked onto its web server. It communicated with TFT through SPI protocol to display the respective meanings of the words clicked etc.*

*3)**Web Server -**The Web Server is just the medium through which ESP32 and the PC communicate with each other.When a picture is clicked,ESP32 uploads it to its web server and the laptop then extracts the image from the web server.*

*4)**TFT** -TFT is used for displaying things across the whole working set like showing which mode is the device currently working on and later displays the required output on it.*

*5)**Battery 3.7V** -It is used to give voltage directly to the TFT and the voltage regulator.*

*6)**Voltage Regulator** -When it takes 3.7V input from the battery,it converts it into 5V and then provides it to the ESP32 device.*

*7)**PC** -It is used for handling the AI part of the scenario and also contains the ESP32 code .It extracts the picture uploaded by ESP32 from the web server and processes it and sends back the required output like final recognised word and its meaning(if any).*

Implementation Details

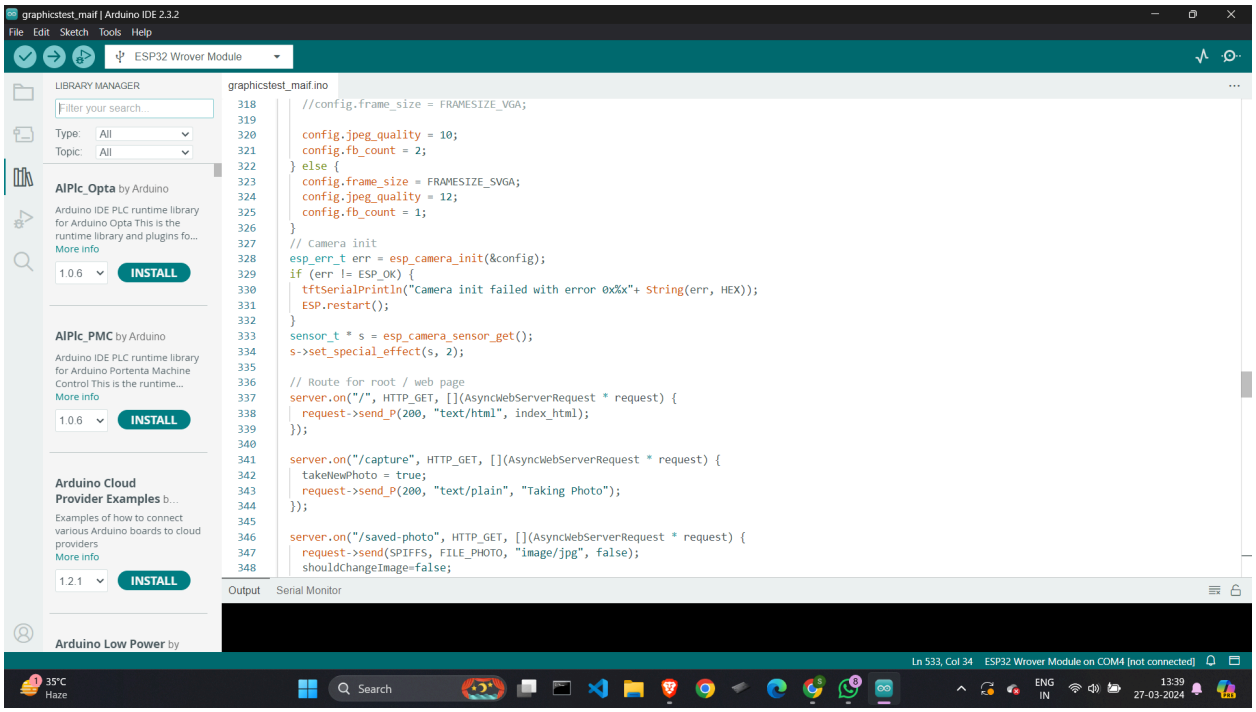
The project “Smart Pen” was developed using a combination of technologies to achieve its objectives:

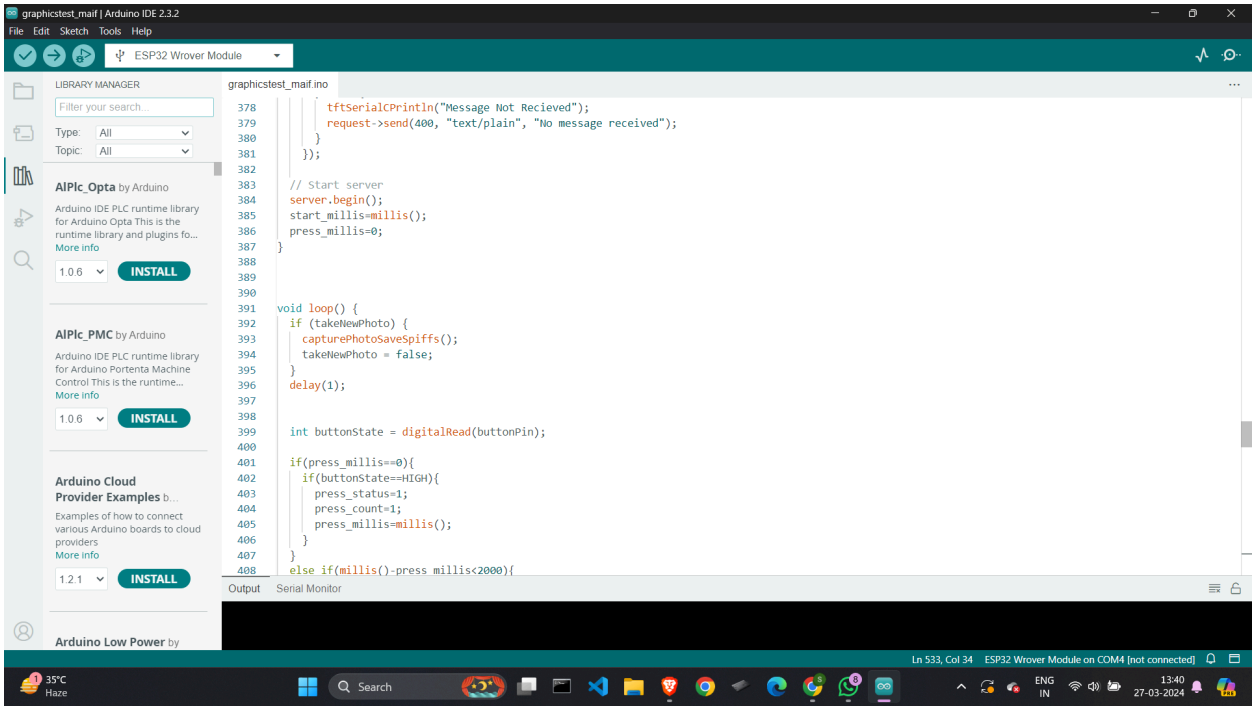
1)We programmed ESP32 devices according to our needs on the Arduino IDE.

2)Libraries such as PyTesseract and easyOCR were used by us for detecting word form a given image,

3)WordNet API was used in order to extract the meaning of the recognised word.

4)SPI protocol was used in order to display various things on the TFT display.





An overview of the ESP32 code is provided below:

1. ***Setup Function***:
   - *Initializes components like TFT display, SPIFFS, WiFi, and camera.*
   - *Sets up routes for the web server.*
   - *Handles button pin setup and configuration.*

2. ***Main Loop***:
   - *Checks if a new photo capture is requested.*
   - *Monitors button state and triggers actions based on press duration.*
   - *Manages TFT display updates based on the current pen mode.*
   - *Handles communication with the web server.*

3. ***Capture Photo Function***:
   - *Captures an image using the ESP32-CAM module.*
   - *Saves the captured photo to SPIFFS.*

4. ***Display Functions***:

- Includes functions for initializing the TFT display and displaying information based on the current pen mode.

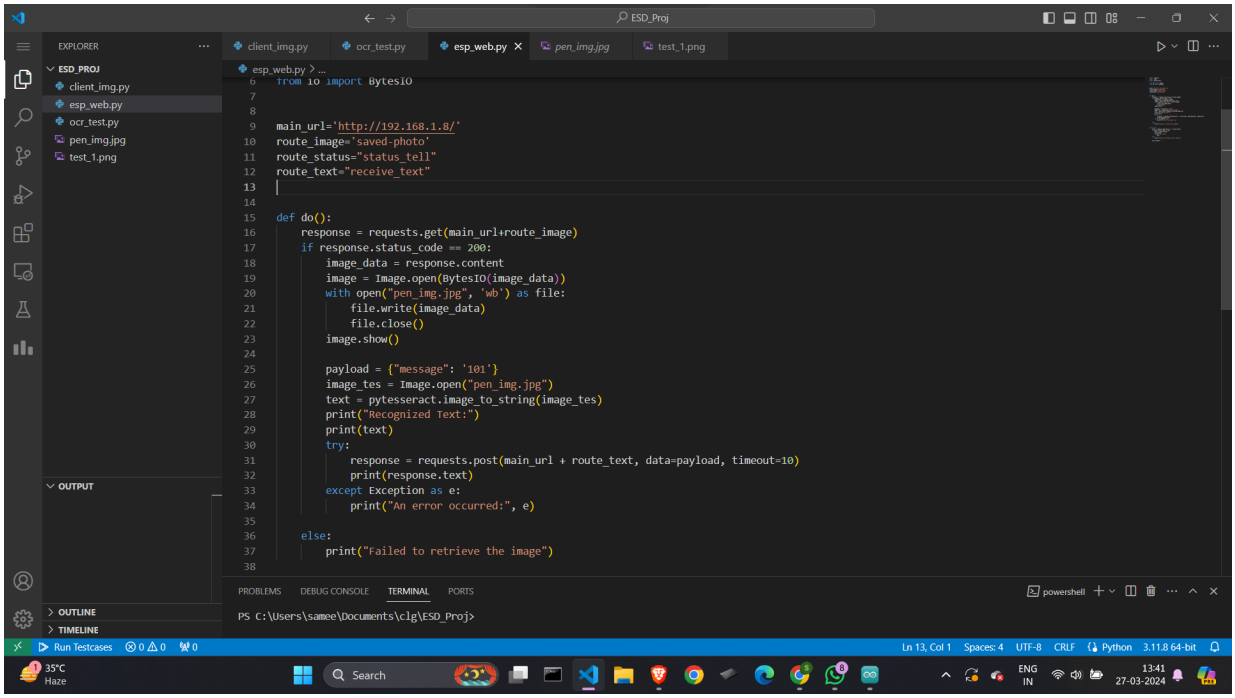
*5. **Web Server Handlers**:*
- Defines routes and handlers for serving HTML pages, capturing photos, and serving saved photos.

*6. **Button Handling**:*
- Monitors button presses and distinguishes between short and long presses.
- Triggers different actions based on press duration and current pen mode.

*7. **SPIFFS Functions**:*
- Includes functions for checking if a photo was successfully saved and saving captured photos to SPIFFS.

*8. **Web Interface**:*
- Provides a web interface for controlling the pen, capturing photos, and viewing captured images.

Overall, the code manages the interaction between hardware components, user input, and the web interface, providing a comprehensive system for controlling the smart pen and capturing images.



This Python script interfaces with a web server running on an ESP32-based device, controlling a smart pen's functionality. Here's a concise explanation:

*1. **Importing Libraries**: Imports necessary libraries including `requests` for HTTP requests, `easyocr` for optical character recognition, `autocorrect` for spell checking, `PIL` for image manipulation, and `nltk` for natural language processing.*

*2. **Setting Up Variables**: Defines URLs and routes for interacting with the web server.*

*3. **Initializing OCR Reader**: Sets up the OCR reader for English text recognition.*

*4. **Main Function (do)**:*

- Sends a request to retrieve the captured image and its type (pen mode) from the ESP32 device.
- If the request is successful (status code 200), it:
- Processes the received image, rotates it, and saves it locally.
- Performs OCR on the image to extract text.
- Corrects recognized text using spell check.
- Retrieves word meanings using WordNet.
- Sends the extracted information back to the ESP32 server.
- If the request fails, it prints an error message.

*5. **Main Loop**:*
- Continuously check the server status.
- If the server indicates that an image is available, it calls the main function to process it.
- If there's an error retrieving server status, it prints an error message.

*6. **Continuous Execution**: The main loop runs continuously, periodically checking for image availability and processing it if necessary.*

This script acts as a bridge between the ESP32 server and the OCR processing, allowing the smart pen to recognize text and perform actions based on user input.

Testing and Evaluation

We tried testing our SmartPen with different kinds of spellings like cursive and only capital words etc. We also tested our AI part by writing wrong spelling and then testing whether it autocorrects the words in a correct manner or not.
After a sufficient amount of testing we found out that our device works perfectly for capital letters words and for cursive words it is doing pretty well but in some cases it misunderstood some letters which leads to inaccuracy on which we are working right now. Another case is where if two or more

words appear in an image it combines the words and tries to find a word in the dictionary accordingly.

Future Enhancements

Expanding on our initial concept, we can incorporate additional features to make the pen even more versatile and practical:

1. By integrating translation capabilities with the assistance of libraries, users can scan words in any language and receive their meanings. This enhancement will be particularly beneficial for tourists and students seeking to understand foreign texts effortlessly.

2. Utilizing modern mathematical engines, the pen can graph functions and present them on its screen. Through a Bluetooth module, these graphs and functions can be transmitted to a larger display, enhancing the learning experience and making complex mathematical concepts more accessible.

3. Rather than displaying results on a screen, we can transmit them to Bluetooth headphones, allowing visually impaired individuals to listen to books or any text in real-time, providing them with a seamless reading experience.

4. Leveraging specifically trained models, the pen can become an invaluable tool for research or legal studies. By scanning legal propositions, it can retrieve relevant previous cases or research papers, aiding in comprehensive analysis and understanding.

5. While the current functionality allows for the detection and display of single words, future advancements aim to refine the technology to enable scanning of entire sentences and multiple words, further enhancing its utility and practicality.