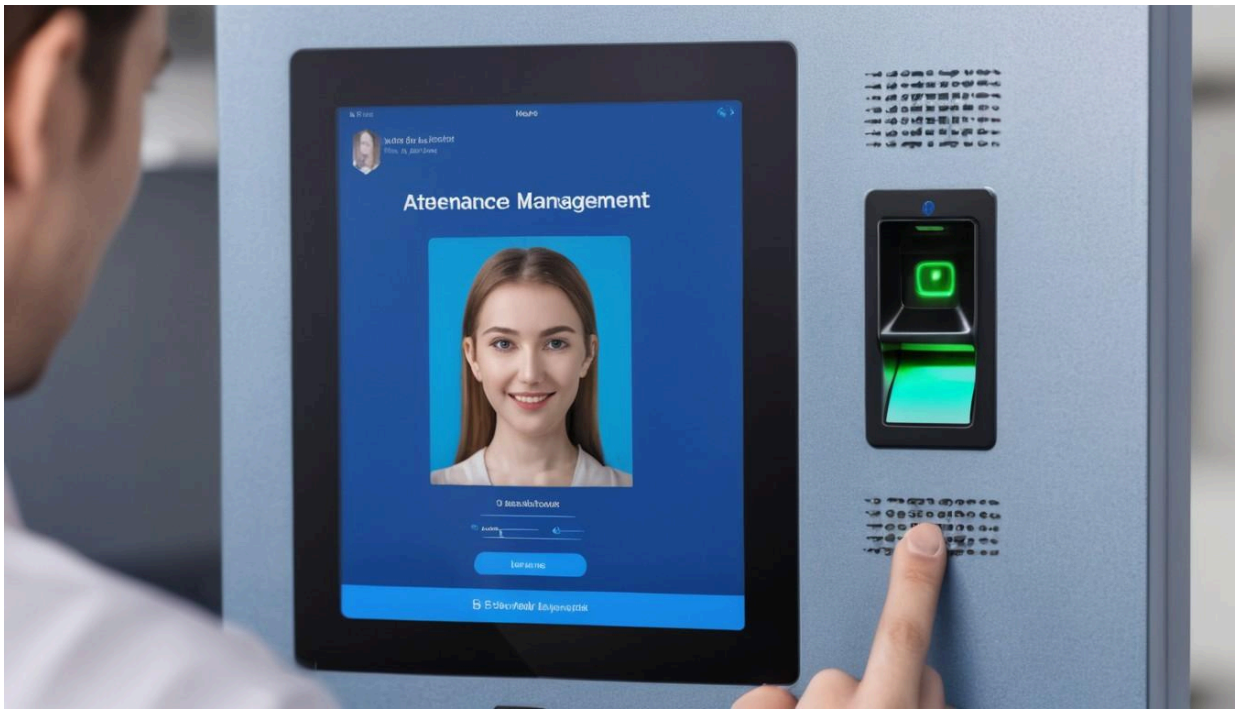
3YP

FACE-SECURE

Attendance Monitoring and Access Authentication with Face Recognition



Design - Manual

Introduction

Welcome to our cutting-edge access control project, where security meets seamless authentication! Our system combines facial recognition, fingerprint scanning, and PIN authentication to redefine access control. This multi-factor approach prioritizes security without compromising user convenience.

Traditional manual attendance management systems suffer from various drawbacks, including time-consuming processes, error-prone data entry, a lack of real-time information, bureaucratic hurdles, limited scalability, and security concerns. These issues necessitate a modernized and technologically advanced solution.

We propose a comprehensive attendance management system called **FACE SECURE** that incorporates facial recognition, fingerprint scanning, and PIN authentication. This multi-factor authentication approach provides seamless, contactless access, precision, uniqueness, and familiarity, ensuring both security and user convenience. The system comprises three main tiers: the attendance management device, backend server, and frontend dashboards for users and administrators. Admins can add/remove employees, view employee details, and receive alerts for unauthorized personnel. Employees can mark attendance through face detection or fingerprint scanning, with access to their attendance profiles.
Our system has lots of features such as Integration with Other Systems, Advanced Reporting Modules, Accurate & consistent performance, Power switching mode, Integration with Other Systems, Error Handling, Easily scalable, User friendly UI design, Less Space Consuming, User Manuals, Keypads provide an additional authentication method, Displaying relevant information.

Who will need this?

Face-Secure is an ideal solution for individuals and companies seeking advanced access control and attendance management systems with multi-level security features. This product caters to a diverse range of users and organizations, offering enhanced security and efficiency in various settings.

Hardware Overview

In the Face-Secure system, the Raspberry Pi board serves as the central control unit for managing various hardware functionalities. Each sensor connected to the Raspberry Pi is dedicated to a specific task within the Face-Secure ecosystem. Notably, Face-Secure stands out due to its three distinct security levels, each associated with different authorization processes.

Security Levels

- ***Level 1****: This mode is exclusively designed for attendance marking, with no authorization required.*
- ***Level 2****: Authorization in this mode is contingent upon correct fingerprint recognition. Upon successful verification, the system authorizes and unlocks the door.*
- ***Level 3****: Authorization in this mode involves both fingerprint and keypad input. Successful verification of both fingerprint and pin-code results in the system authorizing and unlocking the door.*

Modes of Operation

Configuration Mode: *This mode is responsible for setting up various functionalities of the device, including:*

- Changing subscribed topics for communication.
- Adjusting the device's security level based on the desired mode.
- Capturing and storing face data of employees for model training.
- Capturing fingerprint data of employees.
- Capturing and storing individual pin-codes in the backend.

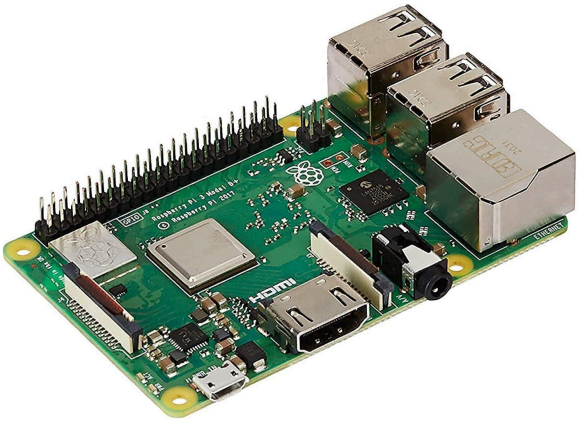
Active Mode: *This mode handles real-time operations, including:*

- Waiting for motion in front of the device.
- Capturing face images and sending them to the backend for recognition, subsequently storing attendance details.
- Capturing fingerprint and pin-code data based on the security levels and executing door unlock logic accordingly.

The robust integration of these hardware functionalities, combined with the flexibility of security levels and distinct modes of operation, sets Face-Secure apart in its approach to access control and attendance management. The Raspberry Pi's pivotal role in orchestrating these operations ensures a seamless and secure experience in diverse scenarios.

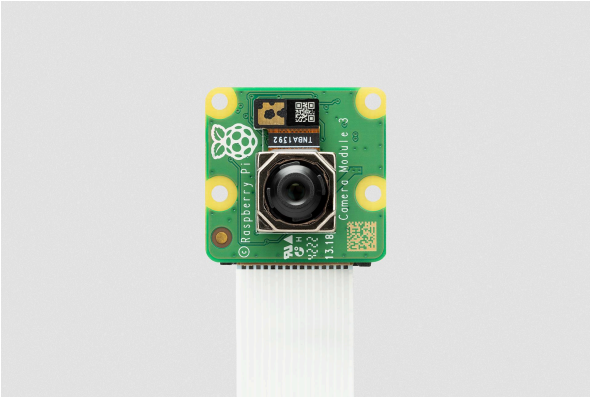
Components

Raspberry Pi 3 B +



5V/2.5A DC power input
Built-in wifi module
Micro SD port
CSI camera port
Full-size HDMI

Raspberry Pi Camera Module V 1.3



5 megapixel camera
Pixel Count 2592 x 1944
Capture video at 1080p30 with H.264 encoding
Angle of View 54 x 41 degrees
Field of View 2.0 x 1.33 m at 2 m

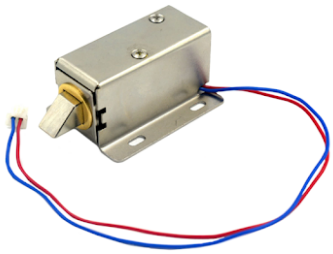
R307 Fingerprint Sensor

Supply voltage: DC 4.2 ~ 6.0V
Working current: 50mA (typical)
Peak current: 80mA
Fingerprint image input time:0.3 seconds
Window area: 14x18 mm

4 x 4 Matrix Keypad

Numeric Keys (0-9)
Programming Support

Solenoid door lock



9-12V Operation
1.7A / 20.5W power draw
1kg pull force
Locking latch can be rotated to all 4 different directions

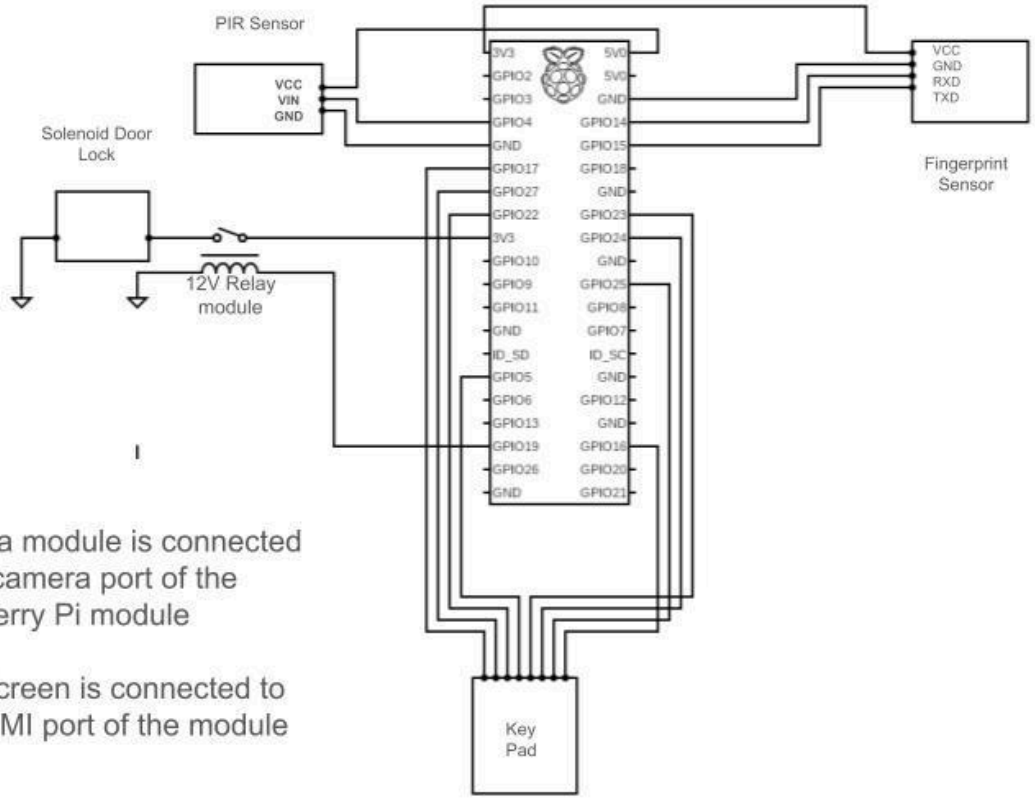
LCD Touch Screen

800 x 480 HD resolution
HDMI interface for displaying
USB interface for touch control
Supports the backlight control to save the electricity

PIR Sensor Module

Detection Technology: Passive Infrared (PIR) technology
Detects motion within its field of view
Allows customization of detection sensitivity
Adjustable time delay between motion detection and output signal

Circuit Diagram



- Camera module is connected to the camera port of the Raspberry Pi module
- LCD Screen is connected to the HDMI port of the module

System Overview

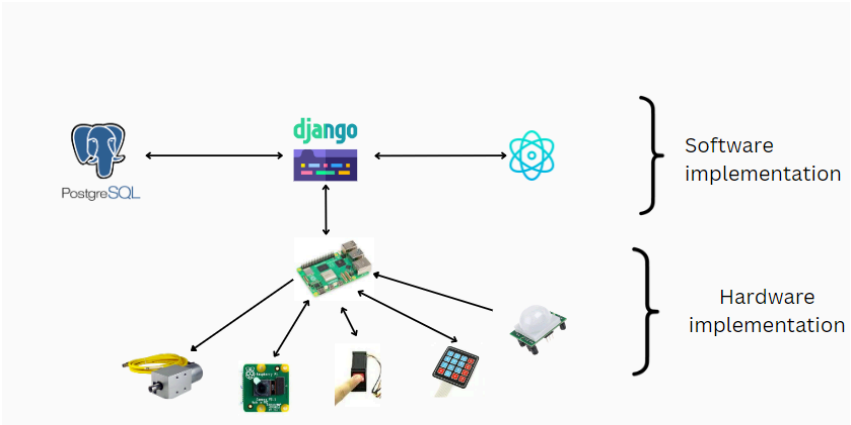


Figure 1: High Level System Overview of Face Secure

Technology Stack

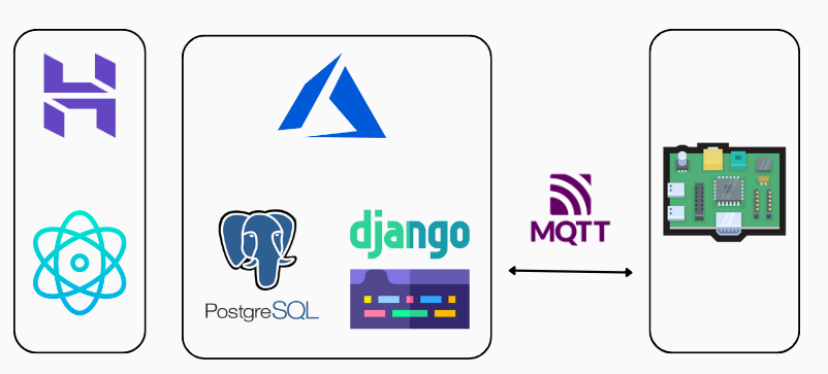


Figure 2: Technology Stack

Backend Framework

Our backend infrastructure relies on Django, a robust framework that facilitates the seamless management of administrative and user-related tasks. This includes following methods for the main functionalities of the Face-Secure.

- Admin & User handle
- Face Detection & Recognition
- Attendance Marking & Management
- Finger-print, Pin-code checking and authorization
- Communication between raspberry pi and the backend

Our system incorporates advanced face detection and recognition functionalities, leveraging specialized libraries such as dlib and face_recognition. This enables accurate identification and authentication based on facial features.

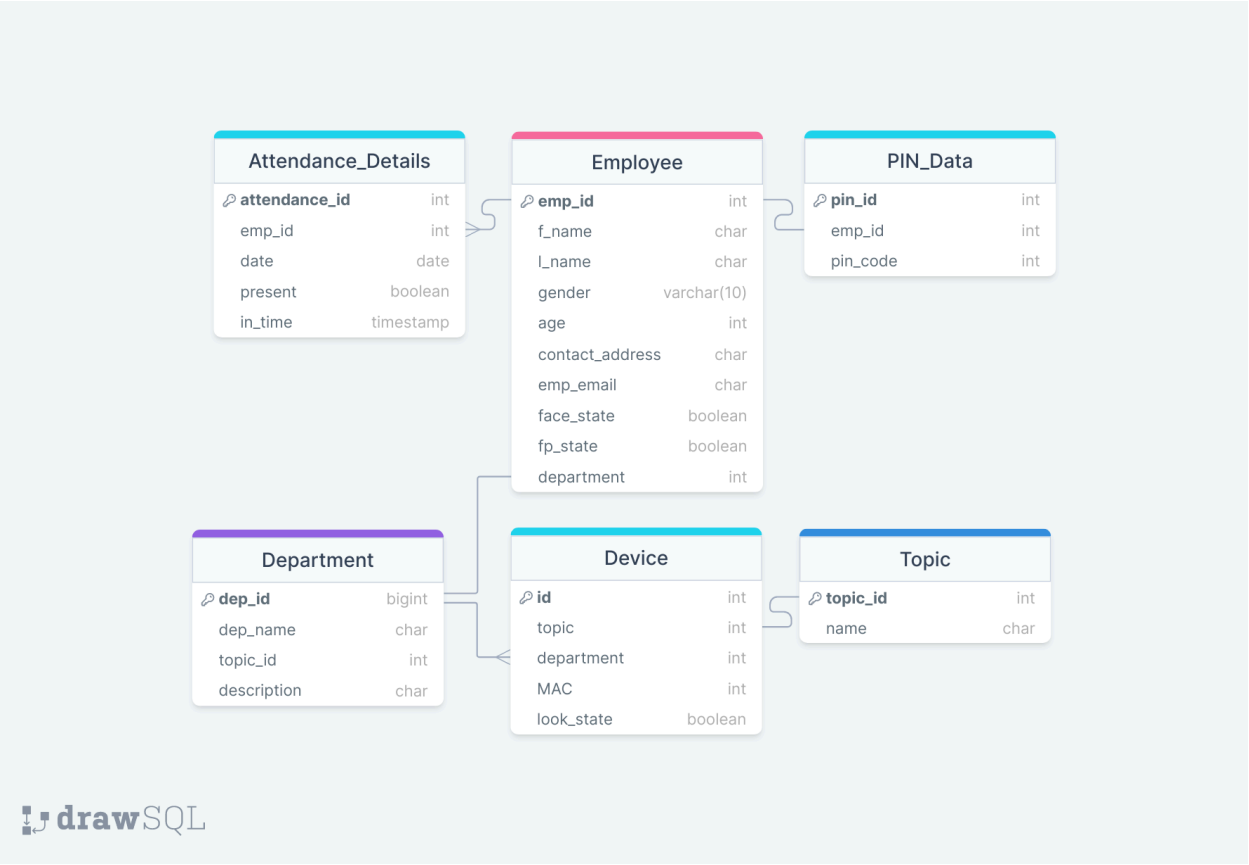
Admin & User management are handled by Django, utilizing tools like reset_framework to ensure efficient and secure processes.

Communication between the Raspberry Pi devices and the backend is facilitated by the implementation of the MQTT (Message Queuing Telemetry Transport) protocol, utilizing the paho-mqtt library. This ensures reliable and real-time data exchange, enabling effective coordination between the physical devices and the backend.

The system supports multiple authentication methods for the door lock, including fingerprint and pin-code verification. These access control mechanisms are also handled by Django using some specific functions.

Database Service

To maintain a robust and integrated database service, our system utilizes PostgreSQL in conjunction with Django. This combination enhances data management, storage, and retrieval capabilities, contributing to the overall efficiency and reliability of the system.



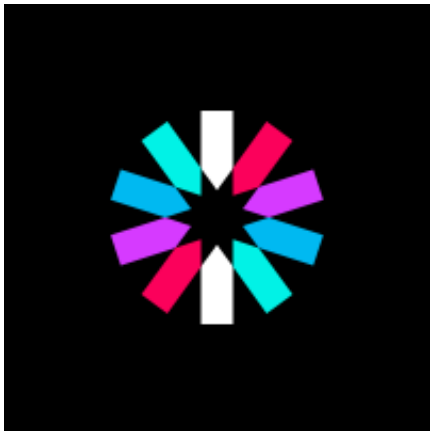
Frontend Framework

Our project utilizes React as the primary front end framework. React is a powerful JavaScript library for building user interfaces, known for its component-based architecture

and efficient rendering capabilities. By employing React, we ensure a modular and scalable codebase, enabling seamless development and maintenance of complex web applications.

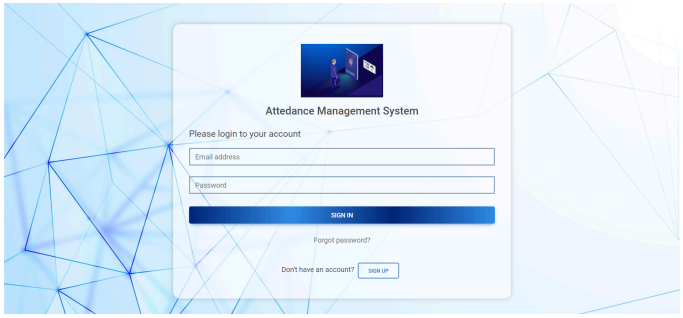
To enhance the visual appeal and user experience of our application, we have integrated the MDB Bootstrap styling library and Font Awesome for icon styling. MDB Bootstrap provides a comprehensive set of components and styles, allowing us to create sleek and responsive user interfaces effortlessly. Leveraging MDB Bootstrap's pre-designed components, we maintain consistency across the application while customizing the design to align with our project's aesthetic preferences.

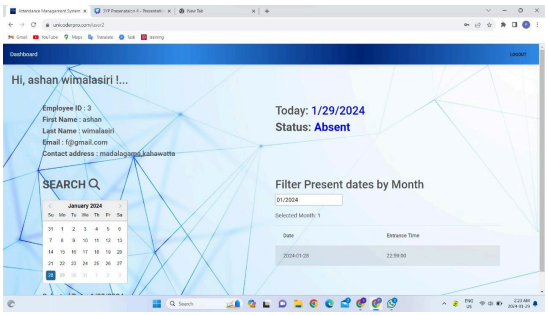
Communication between the frontend and backend of our application is facilitated through the Axios library. Axios is a popular JavaScript library for making HTTP requests, offering a simple and intuitive interface for fetching data from RESTful APIs. With Axios, we efficiently retrieve and manipulate data from our Django backend, ensuring seamless integration and robust data management.

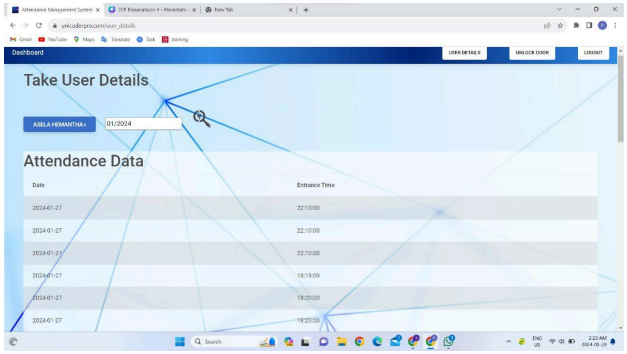


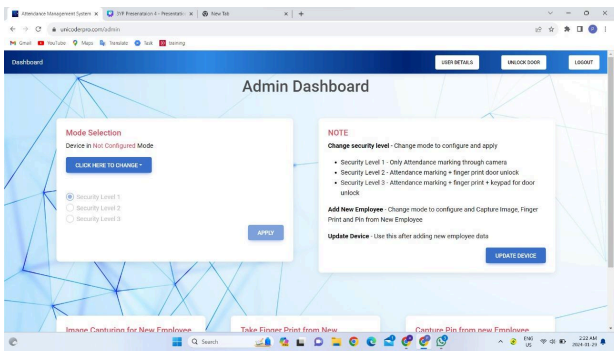
Our project employs JWT (JSON Web Token) authentication for both admin and user authentication. Admin authentication is managed by the Django backend, while user authentication is handled using Firebase Authentication. By implementing JWT authentication, we ensure secure and stateless communication between the client and server, enhancing the overall security of our application. The token refresh mechanism is seamlessly integrated into our application using an interceptor library, in tandem with Axios, for handling HTTP requests. This ensures that authentication tokens are automatically refreshed when expired, maintaining uninterrupted access to protected resources.

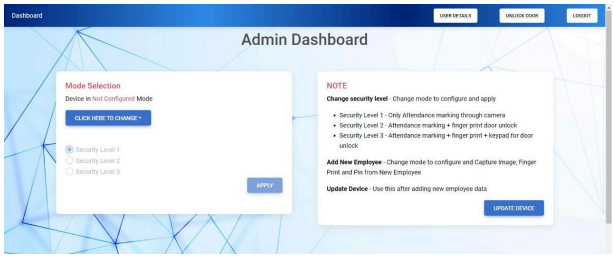
To streamline user interactions and improve overall usability, we integrate React Date Time Picker library for capturing user inputs related to date and time. This library provides intuitive date and time selection components, enhancing the user experience by simplifying complex input tasks. By leveraging React Date Time Picker, we optimize user engagement and foster a seamless interaction flow within our application.









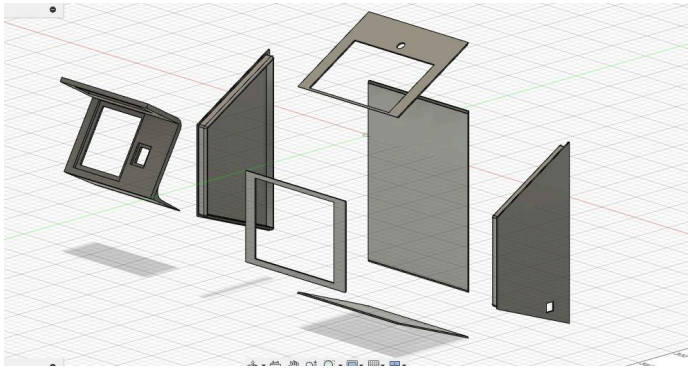


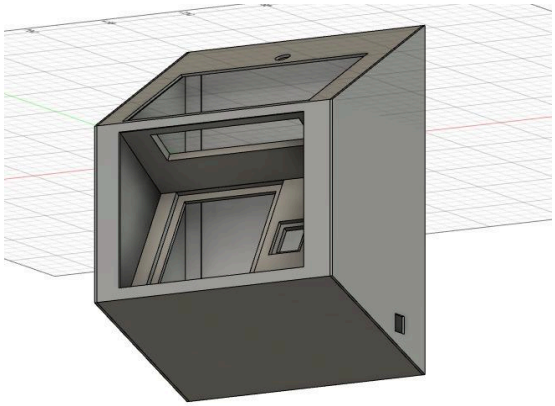
Communication

The communication between the backend and the Raspberry Pi in your project is facilitated through MQTT (Message Queuing Telemetry Transport), and the MQTT broker employed is EMQX. EMQX is utilized as the central hub for managing and routing messages between the backend server and the Raspberry Pi devices.

Design Details

CAD Model

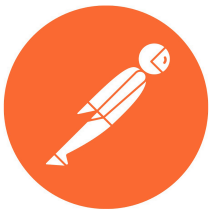




FInal Design

Testing

Django REST API Testing



Postman API is employed for testing all the REST endpoints of the Django backend.

Users and developers can access detailed information about the API testing scenarios through this link.

https://face-secure.postman.co/workspace/New-Team-Workspace~77be0894-aa2f-4a04-9bc3-5ea37f8cde25/collection/27880539-36f84500-4771-4e70-8697-f299a7bc5002?action=share&creator=27880539

Front-End Testing with Cypress

For comprehensive front-end testing, the project utilizes Cypress as the testing framework. Cypress is known for its ease of use, speed, and capability to perform end-to-end testing of web applications.

User Interface Validation - Cypress allows for the creation of robust test cases that simulate user interactions and validate the behavior of the user interface. This includes testing various scenarios such as user authentication, form submissions, and dynamic content rendering.

Automated UI Testing - Automated Cypress test scripts are designed to run through critical user flows, ensuring that the front-end components seamlessly interact with the Django backend and other hardware elements.

Hardware Component Testing

Before assembly, individual hardware components undergo unit testing to validate their functionality in isolation.

Each sensor, Raspberry Pi functionality, and communication module is systematically tested to ensure accuracy, responsiveness, and compatibility with the overall system architecture.

Servers Details

Backend Server

Django Backend with Postgresql is hosted in azure app service. You can access the backend using this url.

https://facesecure.azurewebsites.net/

Frontend Server

React Front end is hosted in hostinger. You can access it using this url.

https://unicoderpro.com/

Thank You!