

SECURITY CAMERA AND DOOR UNLOCK SYSTEM

A Project Report

Submitted in partial fulfilment of the
requirements for the

Science Honors Program

By

Pravin Kumar Mahato (25)

Under the esteemed guidance of

Ms. Pragati Thawani

Assistant Professor



DEPARTMENT OF INFORMATION TECHNOLOGY

KISHINCHAND CHELLARAM COLLEGE

(Affiliated to University of Mumbai)

CHURCHGATE, MUMBAI-400020

MAHARASHTRA

2021-22

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PRN No.: 2019016400534535

Roll No.: 25

1. **Name of the Student:** Pravin Kumar Mahato
2. **Title of the Project:** Security Camera & Door Unlock System
3. **Name of the Guide:** Ms. Pragati Thawani
4. **Teaching experience of the Guide:** 15 Years
5. **Is this your first submission?** Yes

Signature of Student

Date: _____

Signature of Guide

Date: _____

Signature of Coordinator

Date: _____

KISHINCHAND CHELLARAM COLLEGE

(Affiliated to University of Mumbai)

CHURCHGATE, MUMBAI-400020

MAHARASHTRA

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the project entitled, “**Security Camera & Door Unlock System**”, is bonafide work of **Pravin Kumar Mahato** bearing Roll. No. **25** of class **T. Y. BSc. (I.T.)** contained in this project file have been periodically examined and signed and that of course of work in **INFORMATION TECHNOLOGY** for the **Science Honors Program** has been satisfactorily carried out for the year **2021- 2022**.

Guided Teacher

Co-ordinator

Date:

College Seal

ABSTRACT

Problem Statement: Security Camera & Door Unlock System

In today's era, we are facing many security challenges on every level. We must use current technologies to solve these issues in order to overcome them. As a result, we came up with the idea of a security camera and door unlocking system. The security camera and door unlock system is implemented using Internet of Things (IoT) technology. The main goal is to secure our home by using Python OpenCV face recognition on a Raspberry Pi 3. When someone approaches the door, the camera catches their face, and the face recognition system running on the Raspberry Pi 3 recognises the person's face, compares it to the faces present in the database, and sends an email to the administrator stating that "someone is waiting at the door." The administrator can then use the door unlock system web app to unlock the door. We'll be using face recognition methods from the OpenCV Python package.

Objectives:

- To come up with a cost-effective solution which will increase the security of our house/office etc.
- To develop a cost-effective and easy-to-use solution that can be used anywhere.
- To develop a solution that can be used by anyone for security purposes.
- To help in preventing thieves from our house/office etc.

ACKNOWLEDGEMENT

I take this opportunity to express a deep sense of gratitude to **Dr. Hemlata Bagla (Principal of K. C. College)**, **Dr. Tejashree Shanbhag (Vice Principal - Science and Mentor of SHP)**, **Dr. Sagarika Damle (Convenor of Jigyasa SHP)**, **Ms. Geeta Brijwani (Coordinator of Jigyasa SHP)**, **Dr. Shalini Rai (Co-Coordinator of Jigyasa SHP)** and **Dr. Rakhi Gupta (HOD of IT)** for giving this golden opportunity of doing this project, and for their cordial support, valuable information, and guidance, which helped us in completing this task through various stages.

I take this opportunity to express my profound gratitude and deep regards to my guide **Ms. Pragati Thawani** for her exemplary guidance, monitoring, and constant encouragement throughout the course of this thesis. The blessing, help, and guidance given by her time-to-time shell care us a long way in the journey of life on which I am about to embark.

I am obliged to staff members of K. C. College, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my project.

Lastly, I thank almighty, my parents, brothers, sisters, and friends for their constant encouragement without which this project would not be possible.

Pravin Kumar Mahato

DECLARATION

We solemnly declare that the project Security Camera & Door Unllock System is based on my own work carried out during the course of my study under the supervision of Ms. Pragati Thawani.

We assert the statements made and conclusions drawn are an outcome of our research work.

We further certify that the work contained in the report is original and has been done by us under the general supervision of our supervisor.

The work has not been submitted to any other Institution for any other degree/ diploma/ certificate in this university or any other University of India or abroad.

We have followed the guidelines provided by the university in writing the report.

Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and given their details in the references.

Pravin Kumar Mahato

TABLE OF CONTENTS

| Chapter | | Topic | Page No. |
|----------|-----|------------------------------------|----------|
| | | Table of Contents | iv |
| | | Table of Figures | vi |
| | | Table of Abbreviations | vii |
| 1 | | Introduction | 1 |
| | 1.1 | Background | 2 |
| | 1.2 | Objectives | 2 |
| | 1.3 | Purpose, Scope and Applicability | 2 |
| | 1.4 | Achievements | 3 |
| 2 | | Survey of Technologies | 4 |
| | 2.1 | Frontend Technologies | 5 |
| | 2.2 | Backend Technologies | 6 |
| | 2.3 | Comparative Study of Technologies | 7 |
| 3 | | Requirements and Analysis | 8 |
| | 3.1 | Problem Definition | 9 |
| | 3.2 | Requirements Specification | 9 |
| | 3.3 | Software and Hardware Requirements | 9 |
| | 3.4 | Planning & Scheduling | 10 |
| | 3.5 | Preliminary Product Description | 10 |
| 4 | | System Design | 11 |
| | 4.1 | Basic Modules | 12 |
| | 4.2 | Data Design | 13 |
| 5 | | Implementation and Testing | 21 |
| | 5.1 | Implementation Approach | 22 |

| | | | |
|---|-----|------------------------------------|----|
| | 5.2 | Code Details | 25 |
| | 5.3 | Testing Approach | 32 |
| 6 | | Result and Discussion | 35 |
| 7 | | Conclusion and Future Scope | 52 |
| | 7.1 | Conclusion | 53 |
| | 7.2 | Limitation | 53 |
| | 7.3 | Future Scope | 53 |
| | | REFERENCE | 54 |

TABLE OF FIGURES

| Figure No. | Topic | Page No. |
|------------|---|----------|
| 3.1 | Gantt Chart | 10 |
| 4.1 | System Design | 12 |
| 4.2 | System Architecture | 13 |
| 4.3 | Schema Structure | 14 |
| 4.4 | Entity Relationship Diagram | 14 |
| 4.5 | Level 1 Data Flow Diagram of the System | 17 |
| 4.6 | Flow Chart of the System | 20 |

TABLE OF ABBREVIATIONS

| S. No. | Abbreviation | Full-Form |
|--------|--------------|-----------------------------------|
| 1 | RFID | Radio Frequency Identification |
| 2 | ATM | Automated teller machine |
| 3 | RAM | Random-access memory |
| 4 | ROM | Read-only memory |
| 5 | API | Application Programming Interface |
| 6 | REST | Representational state transfer |
| 7 | HTML | Hypertext Markup Language |
| 8 | CSS | Cascading Style Sheets |
| 9 | UAT | User Acceptance Testing |
| 10 | ST | System Testing |
| 11 | UT | Unit Testing |
| 12 | IT | Integration Testing |
| 13 | OOP | Object-oriented programming |
| 14 | SQL | Structured Query Language |
| 15 | MySQL | My Structured Query Language |
| 16 | HTTP | HyperText Transfer Protocol |
| 17 | DFD | Data Flow Diagram |
| 18 | USB | Universal Serial Bus |
| 19 | ER | Entity Relationship |

CHAPTER 1

Introduction

1.1 Background

As we know one of the main reasons for the breach of security is unauthorized access to strangers.

The old door security systems use keys, locks, and chains. These locks can be easily broken.

The use of keys to unlock the doors is not efficient sometimes, because the keys maybe sometimes used by the wrong person, or keys can get stolen or can be duplicated.

It is observed that countries such as US, China, Japan, etc. have switched from traditional door lock systems to smart door lock systems like Biometric door lock systems, password-based door lock systems, RFID-based door lock systems, etc. A face recognition door lock system is used in only some places.

In our country most people use the old traditional lock-key door lock system, only some high-security area is protected using smart door lock systems. The Face recognition door lock system is used in only some places

1.2 Objectives

To come up with a cost-effective solution which will increase the security of our house / office etc.

To develop a cost-effective and easy-to-use solution that can be used anywhere.

To develop a solution that can be used by anyone for security purposes.

To help in preventing thieves from our house/office etc.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

The currently traditionally used lock and key security system are not that secure, anyone can break or bypass the traditional door lock system. This is also one of the reasons for theft in our society. In order to secure our homes from the thief, we need to take some necessary action for the same.

Hence the purpose of this project is to develop a cost-effective solution that can be used to secure our homes from thieves. To develop a security system with add-on features like live streaming of the security camera etc. The solution not only is portable and cost-effective but should also recognize faces and alert the owner of the house about the same.

1.3.2 Scope

Statistics reveal that there are around 2.5 million burglaries happening each year, 66% of which are home break-ins. Despite its popularity, burglars are not deterred by neighborhood watch programs.

Break-in statistics reveals that only 22% of burglars consider a neighborhood watch sign as a deterrent when selecting a property to rob. Only 17% of US citizens have security systems in their homes, despite the fact that such systems can cost approx. \$10 per month.

Research has found that homes without a security system increase their chances of experiencing a home invasion more than those with a security system in place. In fact, most burglars immediately move on (to another house) if a home has a security system.

Every 3 minutes a burglary, robbery, or break-in is taking place in India. Research has found that 70% of all thefts in India are home thefts while only 30% are digital thefts.

1.3.3 Applicability

Using our solution anyone can implement home security cost-effectively and monitor it from anywhere in the world using the flask web app.

This project can be used anywhere like offices, high-security areas, and high-security rooms where only specific people are permitted, etc. where there is a need for security and monitoring of space/place.

1.4 Achievements

The Face recognition-based door unlock system is now cost-effective.

Security of the home will increase by implementing this project at our home.

Using the proposed solution, the owner can keep his eye on his home from anywhere in the world.

Smart Home Security Systems can be implemented cost-effectively anywhere.

This Face Recognition-based Smart Home Security Systems can be implemented anywhere like home, office, bank, ATM, etc.

CHAPTER 2

Survey of Technologies

2.1 Front-end Technologies

2.1.1 HTML

- HTML helps to build the structure of a website and is a widely used Markup language.
- Every browser supports HTML Language.
- HTML is light weighted and fast to load.
- HTML is simple to edit as being plain text.
- It integrates easily with other languages such as JavaScript, CSS etc.
- HTML also allows the utilization of templates, which makes designing a webpage easy.

2.1.2 CSS

- CSS plays an important role, by using CSS you simply got to specify a repeated style for an element once & use it multiple times because CSS will automatically apply the required styles.
- The main advantage of CSS is that style is applied consistently across a variety of sites. One instruction can control several areas which are advantageous.
- Web designers need to use a few lines of programming for every page improving site speed.
- Cascading sheet not only simplifies website development but also simplifies the maintenance as a change of one line of code affects the whole website and maintenance time.
- It is less complex therefore the effort is significantly reduced.
- Easy for the user to customize the online page.

2.1.3 Bootstrap

- Bootstrap is very easy and quite simple to use for designing and development.
- Bootstrap is a framework which is easily integrated with bulk frameworks uninterrupted with existing sites or the new ones.
- Bootstrap is an agile framework which is quite faster than other frameworks. It saves time due to its standard ready-made coding blocks, responsiveness, and cross-browser capabilities.
- One more recommendable feature is that bootstrap holds 12 column grid styles and supports responsiveness, counterbalance, and embedded elements.
- Bootstrap is a framework which holds base styling HTML elements like tables, typography, buttons, forms, images, lists, icons, etc.
- Bootstrap supports Javascript Components so one doesn't require knowledge of scripting. It comprises JavaScript components like tooltips, modal windows, alerts, etc. to add functionalities easily.

2.1.4 jQuery

- jQuery is easy to learn because it supports the same JavaScript style coding.
- jQuery provides a rich set of features that increase developers' productivity by writing less and readable code.
- jQuery provides excellent online API documentation.
- jQuery provides excellent cross-browser support without writing extra code.
- jQuery is unobtrusive which allows separation of concerns by separating HTML and jQuery code.

2.2 Backend Technologies

2.2.1 MySQL

- MySQL is a Relational Database Management System or RDBMS which means that it stores and presents data in tabular form, organized in rows and columns.
- MySQL is more secure as it consists of a solid data security layer to protect sensitive data from intruders and passwords in MySQL are encrypted.
- MySQL is available for free to download and use from the official site of MySQL.
- MySQL is compatible with most of the operating systems, including Windows, Linux, NetWare, Novell, Solaris, and other variations of UNIX.
- MySQL provides the facility to run the clients and the server on the same computer or on different computers, via the internet or local network.
- MySQL has a unique storage engine architecture which makes it faster, cheaper and more reliable.

2.2.2 Flask

- Built-in development server and fast debugger.
- Integrated support for unit testing.
- Restful request dispatching.
- Uses a Jinja2 template engine.
- HTTP request handling function.
- Support for secure cookies (client-side sessions).
- Extensive documentation.
- Google app engine compatibility.
- APIs are nicely shaped and coherent.
- It is easy to deploy the flask in production.

2.3 Comparative Study of Technologies

On the Web Development platform, HTML, CSS, and JavaScript is the most popular language for creating web apps. HTML is light weighted and fast. Every browser supports HTML Language. HTML integrates easily with other languages such as JavaScript, CSS etc. Using CSS we can add styles to the web pages. CSS is less complex therefore the effort is significantly reduced. It has the power for re-positioning. It helps us to determine the changes within the position of web elements that are there on the page. Bootstrap is very easy and quite simple to use for designing and development. Bootstrap is a framework that is easily integrated with bulk frameworks uninterrupted with existing sites or new ones. Bootstrap contains HTML, CSS, and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. jQuery provides a rich set of features that increase developers' productivity by writing less and readable code. jQuery provides excellent cross-browser support without writing extra code.

For data storage, we are using MySQL over several options available such as Mongo DB, Oracle, Microsoft SQL, etc. This is because MySQL is an SQL database. MySQL Stores and presents data in tabular form, organized in rows and columns. Also, MySQL is more secure as it consists of a solid data security layer to protect sensitive data from intruders, and passwords in MySQL are encrypted. And it is available for free to download and use from the official site of MySQL

Flask is technically open source. Flask is a microframework for Python based on Werkzeug, Jinja 2, and good intentions. Flask is a great framework for modern development. Development using Flask is very easy, that is why it is best for beginners. With Flask, we have the ability to create multiple Flask applications or servers, distributed across a large network of servers, each with specific purposes. This creates more efficiency, better testability, and better performance.

These are the features that attract me to choose these technologies and frameworks for this project.

CHAPTER 3

Requirements and Analysis

3.1 Problem Definition

- To come up with a cost-effective solution which will increase the security of our house/office etc.
- To develop a cost-effective and easy-to-use solution that can be used anywhere.
- To develop a solution that can be used by anyone for security purposes.
- To help in preventing thieves from our house/office etc.

3.2 Requirements Specification

Since the dawn of time, home security has been a major issue. In effort to discover a solution to this widespread problem, a large number of systems have been developed.

One of the most widely used home security system is lock and key which is not as effective. And CCTV camera is also one of the most used for security, major drawback of this systems is that it doesn't notify the admin about the theft on time and this system is quite expensive also.

Hence the requirements should be minimal, mostly on the hardware or software side to ensure that the solution is easy to setup, portable and cost effective. The solution should be dynamic which can be implemented and tested in various environments with as less manual requirements as possible to ensure a complete solution for Security Camera and Door Unlock System.

3.3 Software and Hardware Requirements

3.3.1 Software Requirements

- Operating System: Raspberry pi OS.
- Frontend: HTML, CSS, JavaScript, jQuery, Bootstrap
- Backend: Python, Flask
- Database: MySQL

3.3.2 Hardware Requirements

- Raspberry Pi 3B+ (Minimum):
RAM: 1GB
- Raspberry Pi Camera
- 5V USB Adapter (for Raspberry pi)
- Solenoid Lock
- 12 Volt Adapter
- 1 channel 5V Relay

3.4 Planning and Scheduling

| Phase | Task | 2021 | | | | | | | | 2022 | | | | | |
|-------|-----------------------------------|------|-----|-----|-----|-----|-----|-----|-----|------|-----|-----|-----|-----|-----|
| | | MAY | JUN | JUL | AUG | SEP | OCT | NOV | DEC | JAN | FEB | MAR | APR | MAY | JUN |
| 1 | Project Topic Selection | | | | | | | | | | | | | | |
| 1 | Problem Definition | | | | | | | | | | | | | | |
| | Rigorous Study & Analysis | | | | | | | | | | | | | | |
| 2 | Project Planning | | | | | | | | | | | | | | |
| | Designing | | | | | | | | | | | | | | |
| | Data Collection | | | | | | | | | | | | | | |
| 3 | Implementation of Proposed System | | | | | | | | | | | | | | |
| 4 | Testing | | | | | | | | | | | | | | |
| | Modifications | | | | | | | | | | | | | | |
| 5 | Deployment | | | | | | | | | | | | | | |

Figure 3.1. Gantt Chart

3.5 Preliminary Product Description

Home security being the major problem in today's era. And the main reasons for breach of security is the unauthorized access to strangers. The old security system includes locking door using traditional lock with keys. These locks can be easily broken. The use of traditional locks to secure the house is not effective most of the times because the keys may be stolen or duplicated by any wrong person. So we come up with the idea of security camera and door unlock system, when someone come in front of the door camera immediately captures the photo of that person recognizes the face and send an email to the admin that 'someone is there in front of your door'. Then admin checks the live video stream on website and unlocks the door for that person.

CHAPTER 4

System Design

4.1 Basic Modules

The System can be divided into four basic modules i.e. Video Streaming Server (Primary Server), Face Recognition Server (Catch Server), Door Lock Control Server, and the Flask Web Application and Database. We have three servers Primary Server is used to Stream Video and send this video input to the Catch Server, the Catch Server Recognizes the face in the video, Records the video, sends an email if any person is there in the video, and then sends this processed video to the client setup in web application, Web Application is used to view the processed video and view registered user, add user and can unlock the door by clicking the 'door unlock' button which is in the web application, and then after clicking the button web application sends a signal to the Door Lock Control Server to open the door and then the Door Lock Control Server will unlock the door.

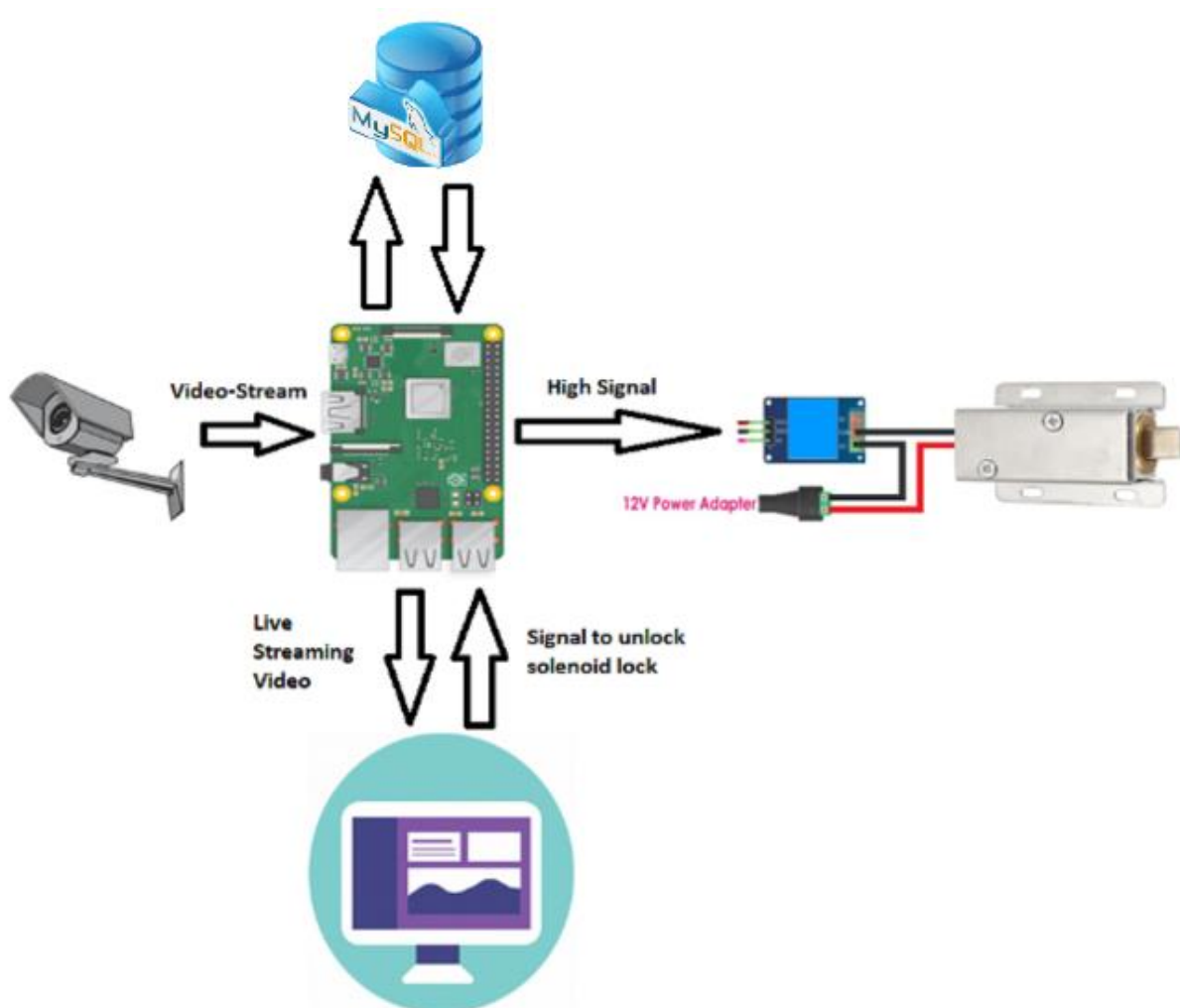


Figure 4.1 System Design

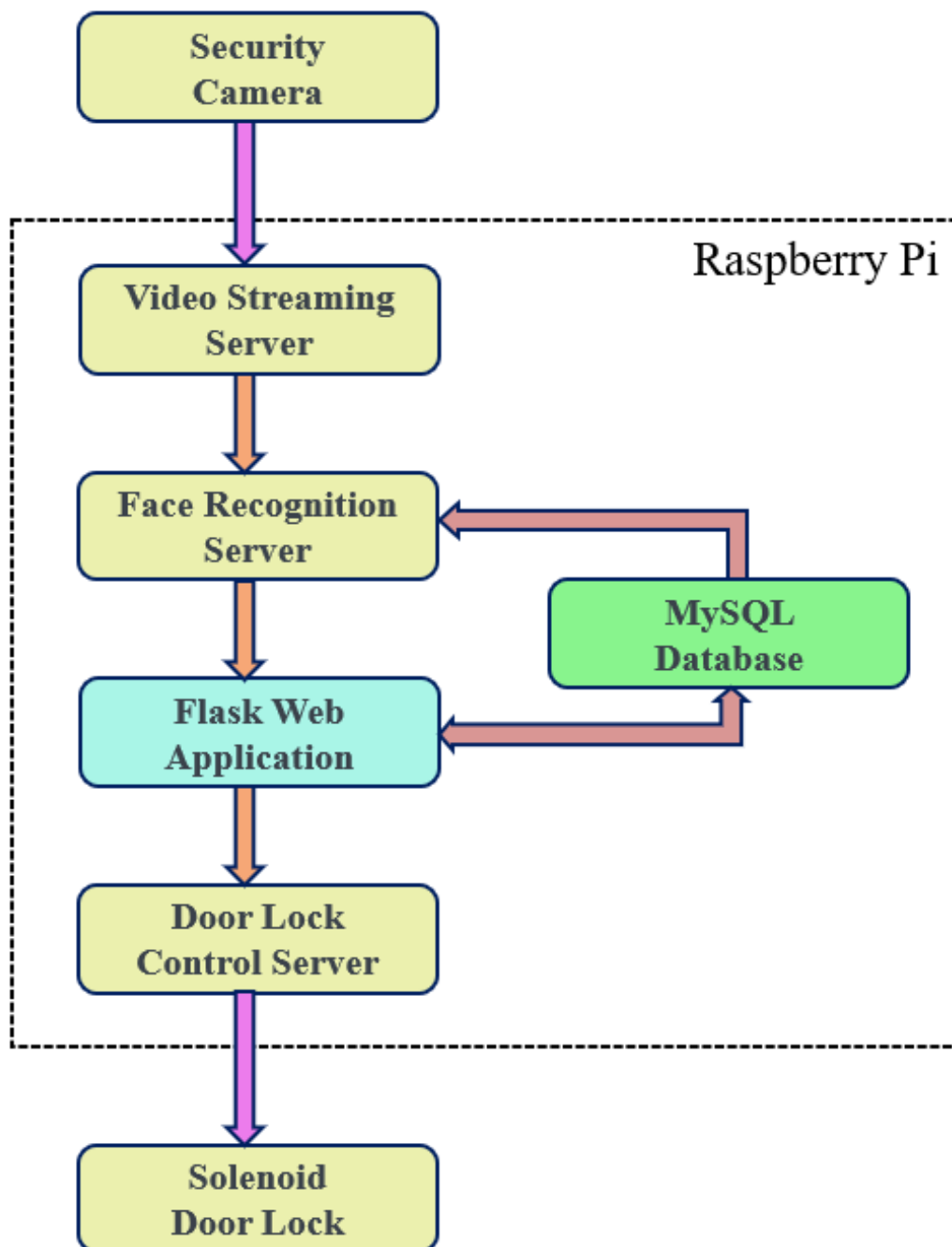


Figure 4.2 System Architecture

4.2 Data Design

The Data of the user and admin will be stored in MySQL Database. MySQL Database is an open-source relational database management system. Classified as a SQL database system, MySQL Database stores and presents data in tabular form, organized in rows and columns.

4.2.1 Schema Design

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.

| Schema | Primary key | Foreign key | Other fields |
|--------|-------------|-------------|--|
| admin | id | | name, phone, email, password |
| setup | id | aid | raspberrypiip, psport, csport, unlockport |
| user | id | aid | name, phone, email, address, photo |

Figure 4.3 Schema Structure

4.2.2 Entity Relationship Diagram

An Entity-Relationship (ER) Diagram is a type of flowchart that shows how "entities" within a system, such as people, things, or concepts, interact with one another. In the disciplines of software engineering, business information systems, education, and research, ER Diagrams are most commonly used to build or troubleshoot relational databases. They use a predetermined collection of symbols such as rectangles, diamonds, ovals, and connecting lines to illustrate the interconnectedness of entities, relationships, and their properties, and are also known as ERDs or ER Models.

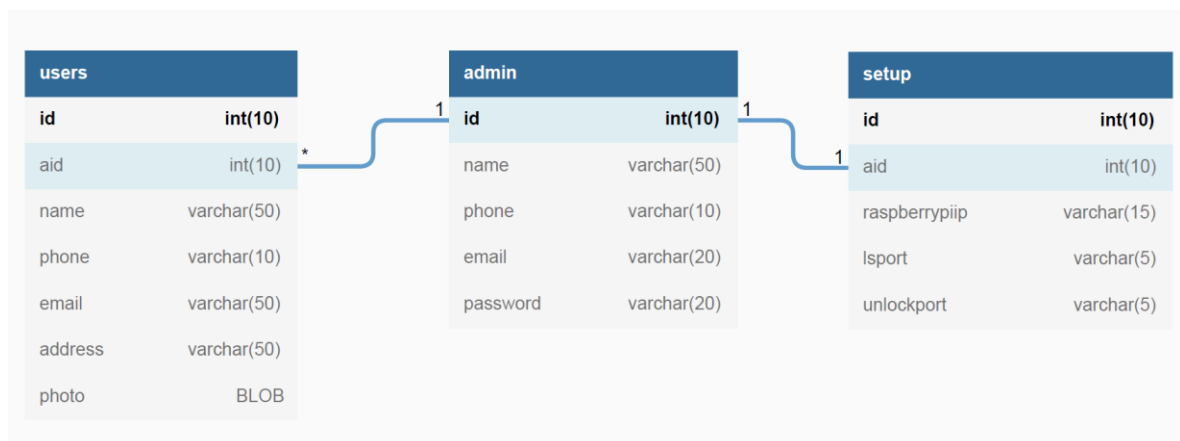


Figure 4.4 Entity Relationship Diagram

4.2.3 Data Flow Diagram

DFD is the abbreviation for **Data Flow Diagram**. DFD is a diagram that depicts the data flow of a system or process. It also provides information about each entity's inputs and outputs, as well as the process itself. There is no control flow in DFD, and there are no loops or decision rules.

The Data Flow Diagram has the following components:

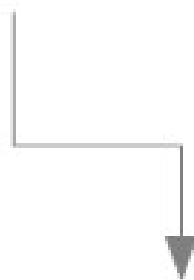
- **Process**

The process function causes input to output transformation in a system. A process can be represented by a rectangular shape with rounded corners, an oval, a rectangle, or a circle. The process is named after a brief sentence that expresses its essence in one word or phrase.



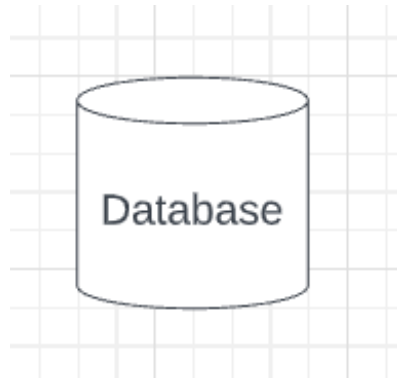
- **Data Flow**

The transport of information between different portions of a system is referred to as data flow. The arrow symbol represents data flow. To determine the information that is being moved, the flow should be given a name that is relatable. Material and information that are being transferred are also represented by the data flow. Material shifts are modeled in non-merely informative systems. A single type of data should be transferred by each flow. The arrow, which can alternatively be bi-directional, represents the flow direction.



- **Warehouse**

The information is kept in a warehouse for later use. The store's symbol is represented by two horizontal lines. The warehouse isn't limited to being a data file; it can also be a folder containing documents, an optical disc, or a filing cabinet. The data warehouse can be seen regardless of how it was built. Data reading occurs when data flows from the warehouse, while data entry or data updation occurs when data flows to the warehouse.



- **Terminator**

The Terminator is an external entity that communicates with the system while standing outside of it. Organizations such as banks, groups of people such as consumers, or various departments of the same corporation, for example, are examples of external entities that are not part of the model system. Terminator can also communicate with modelled systems.



Data Flow Diagram of the System

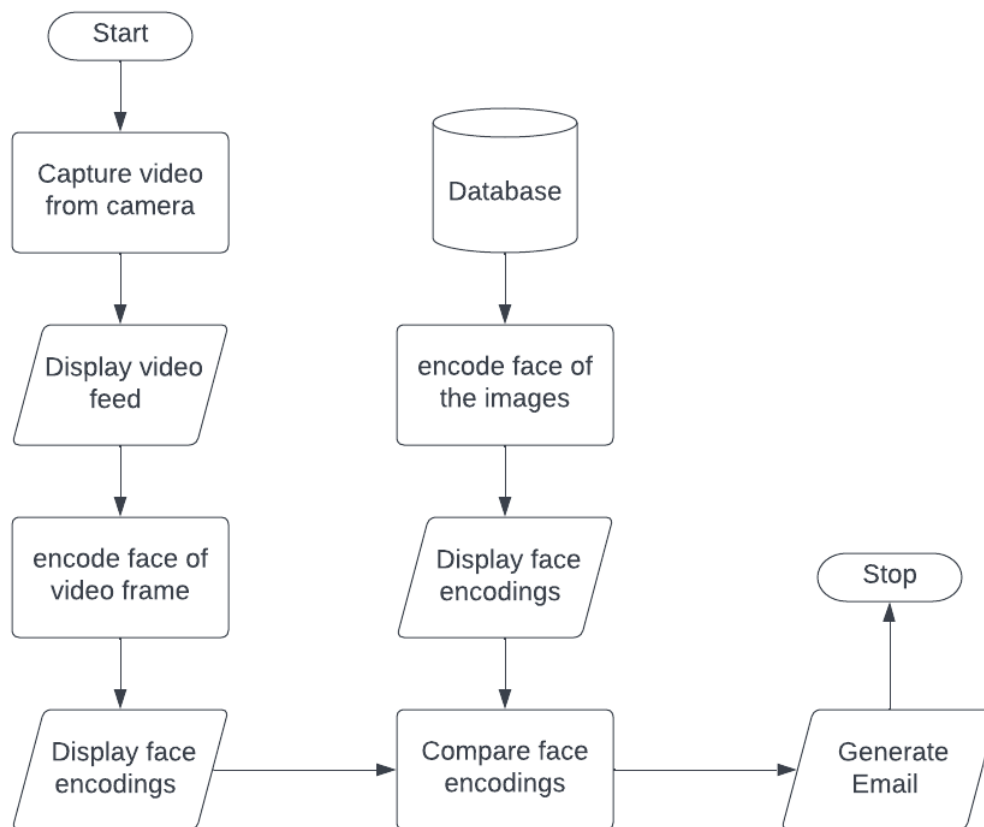


Figure 4.5 Level 1 Data Flow Diagram of the System

Given above is the Data Flow Diagram for the System. After the start of the process the external entity, in this case, the patient needs to input the information which will be sent to the API for further processing after which the data will be sent to the MongoDB Database. In the second stage after the visual acuity testing is done and the validated results are sent from the remote application to the API server which will again process this information and send it to the MongoDB server and the previous data will be appended with the Visual Acuity Score. Finally, the API will trigger the mail function which will send the final results to the patient's email address.

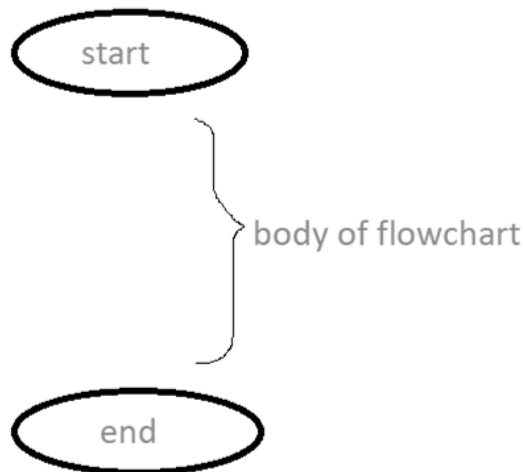
4.2.4 Flow Chart

A flowchart is a visual representation of the sequence of steps and decisions needed to perform a process. Each step in the sequence is noted within a diagram shape. Steps are linked by connecting lines and directional arrows. This allows anyone to view the flowchart and logically follow the process from beginning to end.

The Use Case Diagram has following components:

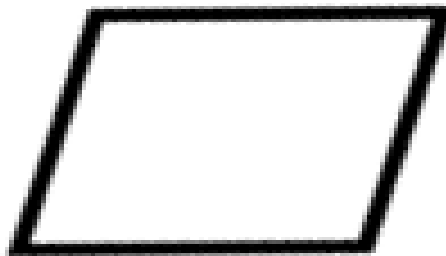
- **Terminal**

This box is of an oval shape which is used to indicate the start or end of the program. Every flowchart diagram has this oval shape that depicts the start of an algorithm and another oval shape that depicts the end of an algorithm.



- **Data**

This is a parallelogram-shaped box inside which the inputs or outputs are written. This basically depicts the information that is entering the system or algorithm and the information that is leaving the system or algorithm.



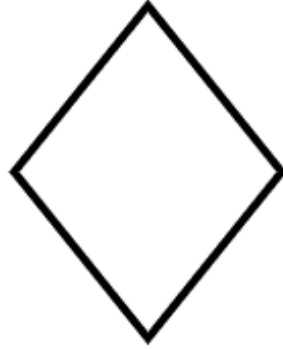
- **Process**

This is a rectangular box inside which a programmer writes the main course of action of the algorithm or the main logic of the program. This is the crux of the flowchart as the main processing codes is written inside this box.



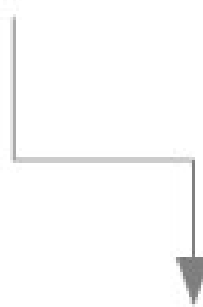
- **Decision**

This is a rhombus-shaped box, control statements like if, or condition like $a > 0$, etc are written inside this box. There are 2 paths from this one which is “yes” and the other one is “no”. Like every decision has either yes or no as an option, similarly, this box to have these as options.



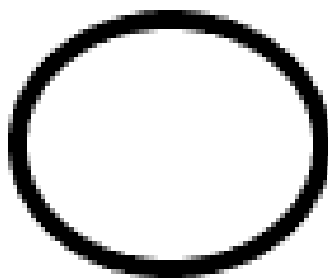
- **Flow**

This arrow line represents the flow of the algorithm or process. It represents the direction of the process flow.



- **On-Page Reference**

This circular figure is used to depict that the flowchart is in continuation with the further steps. This figure comes into use when the space is less and the flowchart is long. Any numerical symbol is present inside this circle and that same numerical symbol will be depicted before the continuation to make the user understand the continuation.



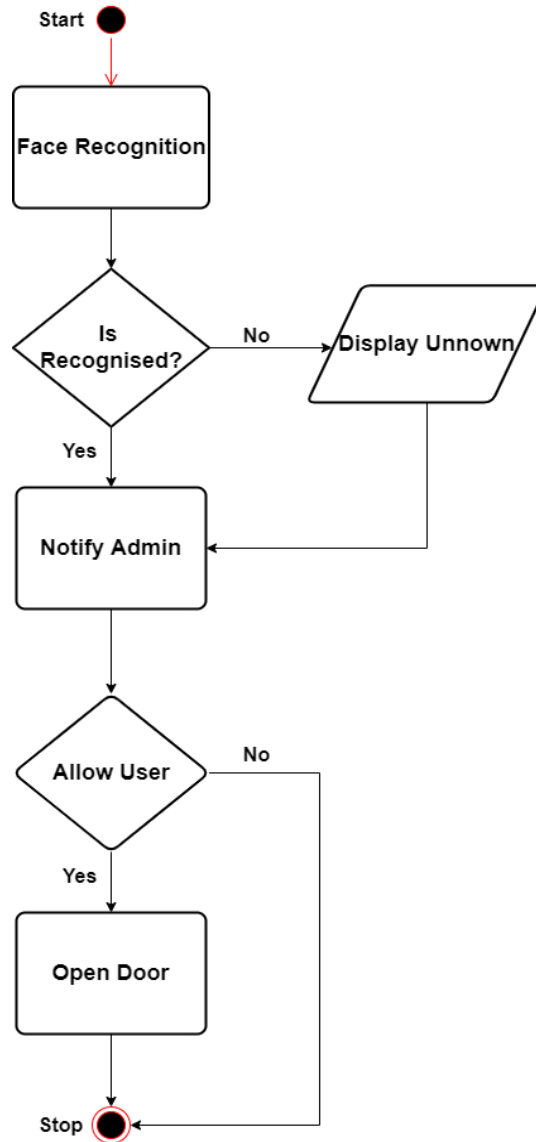


Figure 4.6 Flow Chart of the System

Given above is the Flow Chart for the System. First system takes picture of the person compares the face with the images present in the database if the face is recognized it sends email to the admin stating the name of the person and if the system cant able to recognize the face then also it sends the email to the admin stating someone is there at the door, then admin allows the user to open the door.

CHAPTER 5

Implementation and Testing

5.1 Implementation Approaches

- A planned approach for a successful project is of utmost importance as the saying goes like
- “Plans are of little importance, but planning is essential.”
- The planning approach used in this project is SCRUM METHODOLOGY.

Agile Methodology

- **Agile Methodology** means a practice that promotes **continuous iteration** of development and testing throughout the software development lifecycle of the project. In the Agile model in software testing, both development and testing activities are concurrent, unlike the Waterfall model.
- The **Agile software development** methodology is one of the simplest and most effective processes to turn a vision for a business need into software solutions.
- Agile is a term used to describe software development approaches that employ continual planning, learning, improvement, team collaboration, evolutionary development, and early delivery. It encourages flexible responses to change.

Scrum Methodology

- Scrum is an agile development methodology used in the development of Software based on an iterative and incremental process.
- Scrum is an adaptable, fast, flexible, and effective agile framework that is designed to deliver value to the customer throughout the development of the project.
- The primary objective of Scrum is to satisfy the customer’s needs through an environment of transparency in communication, collective responsibility, and continuous progress.
- The development starts from a general idea of what needs to be built, elaborating a list of characteristics ordered by priority (product backlog) that the owner of the product wants to obtain.

Phases of Scrum Methodology

1. **Initiate** - This phase includes the processes related to the initiation of a project: Create Project Vision, Identify Scrum Master and Stakeholder(s), Form Scrum Team, Develop Epic(s), Create Prioritized Product Backlog, and Conduct Release Planning.
2. **Plan and Estimate** - This phase consists of processes related to planning and estimating tasks, which include Create User Stories, Approve, Estimate, and Commit User Stories, Create Tasks, Estimate Tasks, and Create Sprint Backlog.
3. **Implement** - This phase is related to the execution of the tasks and activities to create a project's product. These activities include creating the various deliverables, conducting Daily Standup Meetings, and grooming (i.e., reviewing, fine-tuning, and regularly updating) the Product Backlog at regular intervals.
4. **Review and Retrospect** - This phase is concerned with reviewing the deliverables and the work that has been done and determining ways to improve the practices and methods used to do project work.
5. **Release** - This phase emphasizes delivering the Accepted Deliverables to the customer and identifying, documenting, and internalizing the lessons learned during the project.

Project Implementation Approach

The project consists of the following components:

1. Video Streaming Server (Primary Server)
2. Face Recognition Server (Catch Server)
3. Door Lock Control Server
4. Web Application

The process starts with the connection of the Video Streaming Server (Primary Server) to the Face Recognition Server (Catch Server), the Face Recognition Server (Catch Server) to the Web Application video streaming client, and the Door Lock Control Server to the Web Application Door Lock Control client. When we first start running Video Streaming Server (Primary Server) it searches for the Face Recognition Server (Catch Server), and when we start running the Face Recognition Server (Catch Server) then both servers connect to each other using sockets, then after the connection, the video streaming will start and side by side video is also kept on recording.

After the connection of the Video Streaming Server (Primary Server) to the Face Recognition Server (Catch Server) then we start running the Door Lock Control Server which starts searching for the client to connect. Then we start running Flask Web Application.

After the successful connection between the server and clients, our setup is all complete the Video Streaming Server (Primary Server) is continuously taking video feed from the security camera and sending it to the Face Recognition Server (Catch Server) and the Face Recognition Server keeps on checking if someone is there in front of the security camera. now we have to wait for someone to come in front of the door.

Now whenever anyone comes in front of the door the security camera which is attached to the door sends the video feed to the Video Streaming Server (Primary Server) and it sends the video feed to the Face Recognition Server (Catch Server). And Face Recognition Server then detects the face and matches it to the face images saved in the database. The server sends an email to the admin stating that someone is at the door, and if the face matches with any person image present in the database, then in the email the system also specifies the name of the person, and if the system failed to recognize the face of the person, then it specifies the person as an unknown person.

After the email is been sent to the admin. The admin then opens the web application on any browser, signs in with his credentials then it gets connected to the Door Lock Control Server through sockets, and then navigate to the live-streaming link then the admin can check who is at the door as the live streaming video is also streaming on that page, and below that there is a button when the admin click on the button, it sends '1' as a response to the Door Lock Control Server then after receiving '1' response from the client, the server then trigger the GPIO pin of raspberry pi which trigger the relay module and the door gets open.

5.2 Code Details

```
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
host_ip = "          # Enter the Primary Server IP address
port = 9999
socket_address = (host_ip, port)
server_socket.bind(socket_address)
server_socket.listen()
```

A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.

An endpoint is a combination of an IP address and a port number. Every TCP connection can be uniquely identified by its two endpoints. That way you can have multiple connections between your host and the server.

Create a new socket using the given address family, socket type, and protocol number. The address family should be AF_INET (the default) and the next parameter should be socket type.. Server_socket.bind(host_ip,port) specifies that the socket is reachable by any address the machine happens to have. Server_socket.listen() enables to a server to accept connections. Accept a connection. The socket must be bound to an address and listening for connections. The return value is a pair (conn, address) where conn is a new socket object usable to send and receive data on the connection, and address is the address bound to the socket on the other end of the connection.

```

def start_video_stream():
    client_socket, addr = server_socket.accept()
    vid = cv2.VideoCapture(0)
    try:
        print('CLIENT {} CONNECTED!'.format(addr))
        if client_socket:
            while (vid.isOpened()):
                img, frame = vid.read()
                frame = imutils.resize(frame, width=700)
                a = pickle.dumps(frame)
                message = struct.pack("Q", len(a)) + a
                client_socket.sendall(message)

    except Exception as e:
        print(f"CACHE SERVER {addr} DISCONNECTED")

while True:
    start_video_stream()

```

After the socket connection is successfully created, the backend is then open to listen to the requests from the catch server.

When the catch server gets connected to the primary server then starts capturing video and after resizing the video frames it sends the video frames to the catch server. This process is kept on running until the catch server is not disconnected.

```

path = 'images'
personName = []
images = []
myList = os.listdir(path)

for cu_img in myList:
    current_img = cv2.imread(f'{path}/{cu_img}')
    images.append(current_img)
    personName.append(os.path.splitext(cu_img)[0])
print(personName)

def faceEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList

encodeListKnown = faceEncodings(images)

```

We have created the path variable and assigned 'image' to it, then we have created two empty lists with the name images and personName, and then we have created the myList variable and assigned the path of the image folder.

Then append the path of all the images present in the image folder to the images list, and append the name of all the images present in the image folder to the personName list.

faceEncodings function encodes all the images present in the images list, first, it creates a list with the name encodeList, then it converts BGR color to RGB color then it encodes the face and appends it in the encodeList list

```

def start_video_stream():
    global frame
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host_ip = "          # Here provide Primary Server IP
    port = 9999
    client_socket.connect((host_ip, port))
    data = b""
    payload_size = struct.calcsize("Q")
    while True:
        while len(data) < payload_size:
            packet = client_socket.recv(4 * 1024)
            if not packet: break
            data += packet
        packed_msg_size = data[:payload_size]
        data = data[payload_size:]
        msg_size = struct.unpack("Q", packed_msg_size)[0]

        while len(data) < msg_size:
            data += client_socket.recv(4 * 1024)
        frame_data = data[:msg_size]
        data = data[msg_size:]
        frame = pickle.loads(frame_data)

        faces = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        facesCurrentFrame = face_recognition.face_locations(faces)
        encodesCurrentFrame = face_recognition.face_encodings(faces, facesCurrentFrame)

```

```

for encodeFace, faceLoc in zip(encodesCurrentFrame, facesCurrentFrame):
    matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
    faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)

    matchIndex = np.argmin(faceDis)

    if matches[matchIndex]:
        name = personName[matchIndex].upper()
        message = "Hello, " + name + " is waiting in front of your door, please open the door"
        email(message)

    else:
        name = 'UNKNOWN'
        print(name)
        message = "Hello, Someone is waiting in front of your door, please open the door"
        email(message)

client_socket.close()

thread = threading.Thread(target=start_video_stream, args=())
thread.start()

```

We have used threads to call the start_video_stream function, then if the primary server is running then it gets connected to the primary server using the client_socket.connect method, after a successful connection the catch server starts taking receiving video frames send by the primary server using sockets. It decodes the frame and stores it into a frame variable.

Then using cv2 the frame is converted to RGB from BGR, Then the face is located in the frame and then it finds the face encoding of thar face, then it compares face encodings of the face in the images and the face found in the video frame. Then it finds the face distance between the face in the images and the face found in the video frame. Then it finds the minimum face distance using NumPy.

Then if the face match with any of the images then it sends an email to the admin that '<the name of that person> is waiting in front of your door, please open the door' and if the face doesn't match with any of the images then it sends an email to the admin that 'Someone is waiting in front of your door, please open the door'

```

@app.route('/senddata')
def senddata():
    try:
        cs = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        host_ip = ' ' # pi server Ip address
        port = 9090
        cs.connect((host_ip, port))

        if cs:
            try:
                code = "1"
                cs.send(str.encode(code))
                print('Data Sent')
            except:
                print('Data Not Sent')
            return redirect(url_for('live'))
    except:
        return redirect(url_for('index'))

```

This is send_data route, this is called when admin clicks on the button to open the door, then it connects to the piserver using socket connection and sends 1 as a encoded string.


```

GPIO.setmode(GPIO.BOARD)
GPIO.setup(3, GPIO.OUT)

def show_client():
    try:
        client_socket, addr = server_socket.accept()
        print('CLIENT {} CONNECTED!'.format(addr))
        if client_socket: # if a client socket exists
            data = client_socket.recv(1024)
            if data.decode("utf-8") == '1':
                GPIO.output(3, True)
                time.sleep(5)
                GPIO.output(3, False)
            else:
                print('value is not equal to 1')
    except Exception as e:
        print(f"CLINET {addr} DISCONNECTED")

while True:
    show_client()

```

First set GPIO mode of the raspberry pi to Board mode, then set pin 3 to the output pin, then it checks if the client is connected, if the client is connected then it checks if the data sent by the client is '1', If it is 1 then it turns pin 3 as high and after 5 seconds it turns pin 3 as low.

5.3 Testing Approach

5.3.1 Unit Testing

- Unit testing is a testing technique in which individual modules are checked by the developer to see if there are any flaws. It is concerned with the standalone modules' functional correctness.
- The basic goal is to isolate each component of the system in order to detect, analyze, and correct any flaws.
- Testing can be accomplished in a variety of ways. The methods available are briefly described in this chapter.

5.3.2 Black-Box Testing

- Black-box testing is a method of testing an application without knowing anything about its internal workings. The tester has no understanding of the system architecture and no access to the source code. A tester will often engage with the system by providing inputs and evaluating outputs while doing a black-box test without understanding how and where the inputs are processed.

5.3.3 White-Box Testing

- White-box testing entails a thorough examination of the code's core logic and structure. Glass testing or open-box testing are other names for white-box testing. A tester must understand the internal workings of the code in order to do white-box testing on a component.
- The tester must examine the source code to determine which unit/chunk of code is acting abnormally.

5.3.4 Grey-Box Testing

- Grey-box testing is a technique for testing produced software with only a rudimentary understanding of the application's underlying workings. A tester with extensive subject expertise always has an advantage over someone with little domain knowledge. In contrast to black-box testing, where the tester only checks the functionality of the system, grey-box testing allows the tester access to architecture documentation. With this information, a tester may better prepare test data and test scenarios when creating a test plan.

5.3.5 Verification

- During the development of a product, verification testing comprises tasks such as business requirements, system requirements, design review, and code walkthrough.
- It's also known as static testing, and it's used to determine whether or not "we're developing the proper product." It also ensures that the generated application meets all of the client's specifications.

5.3.6 Validation

- Validation testing in software engineering is in place to determine if the existing system complies with the system requirements and performs the dedicated functions for which it is designed along with meeting the goals and needs of the organisation.
- Validation testing is when a tester performs both functional and non-functional tests on a product. Non-functional testing covers User Acceptance Testing (UAT), Integration Testing (IT), and System Testing (ST), while functional testing includes Unit Testing (UT), Integration Testing (IT), and System Testing (ST) (UAT).
- Validation testing, often known as dynamic testing, is used to ensure that "the product has been developed correctly." It also ensures that the software satisfies the client's business requirements.

Importance of Validation Testing

- To ensure that customers are satisfied
- To have confidence in the product
- To meet the needs of the client until the maximum capacity is reached
- The end-acceptance users of the software

When any software is tested, the primary goal is to compare the quality to the defects discovered. The developers fix the bugs, and the software is retested to ensure that no bugs are left behind. This improves not only the product's quality but also its user acceptance.

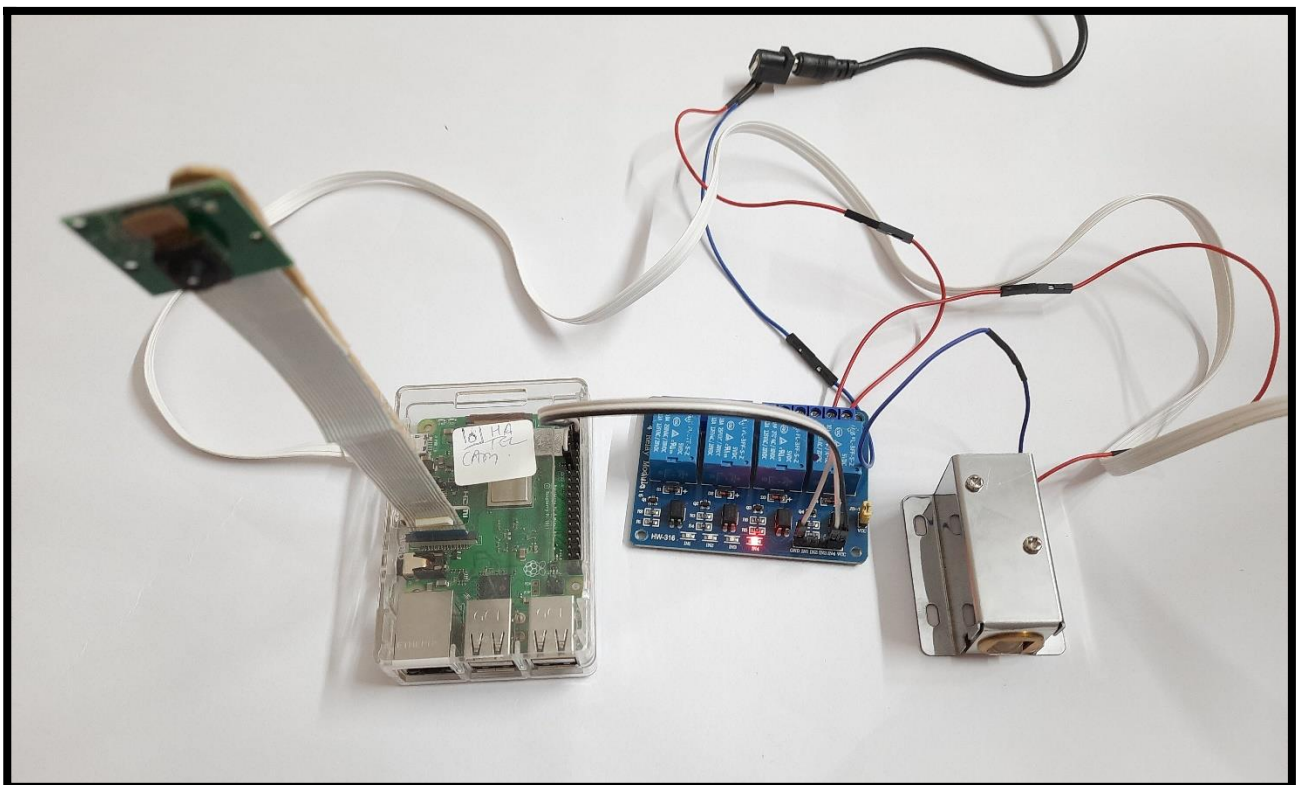
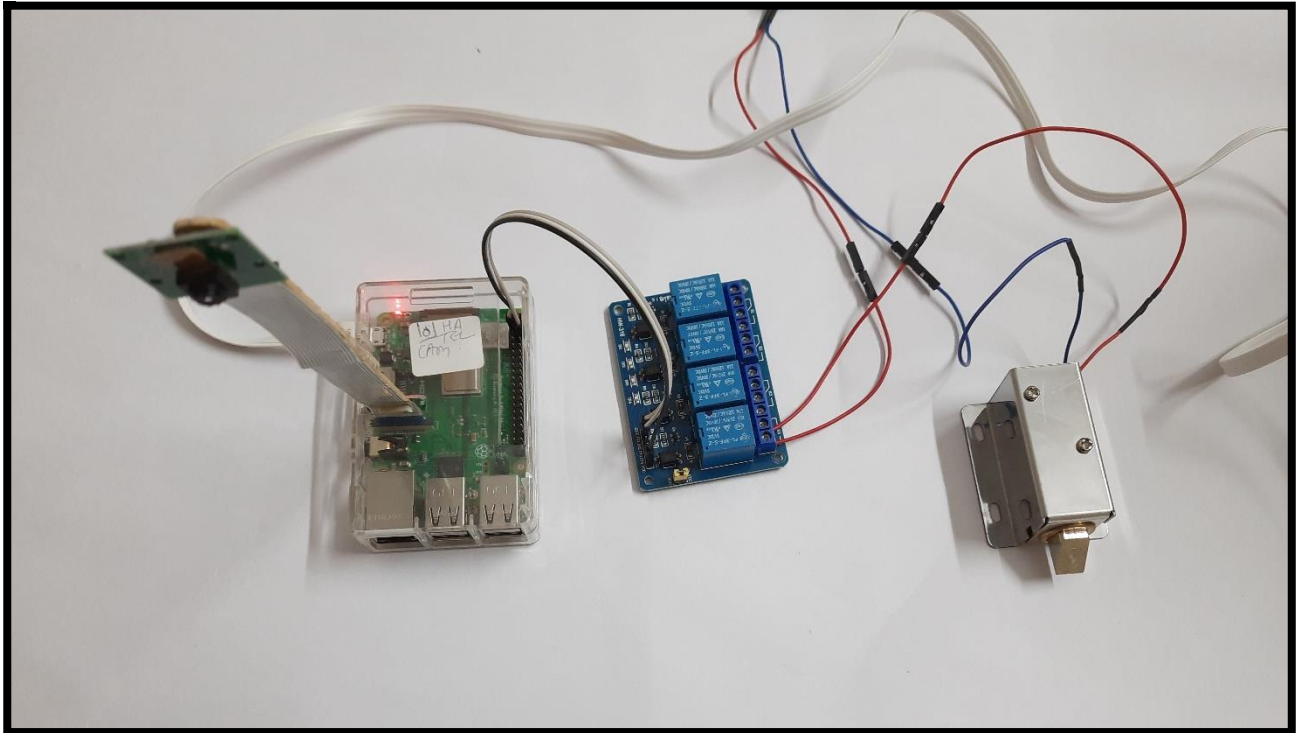
Validation user in the project

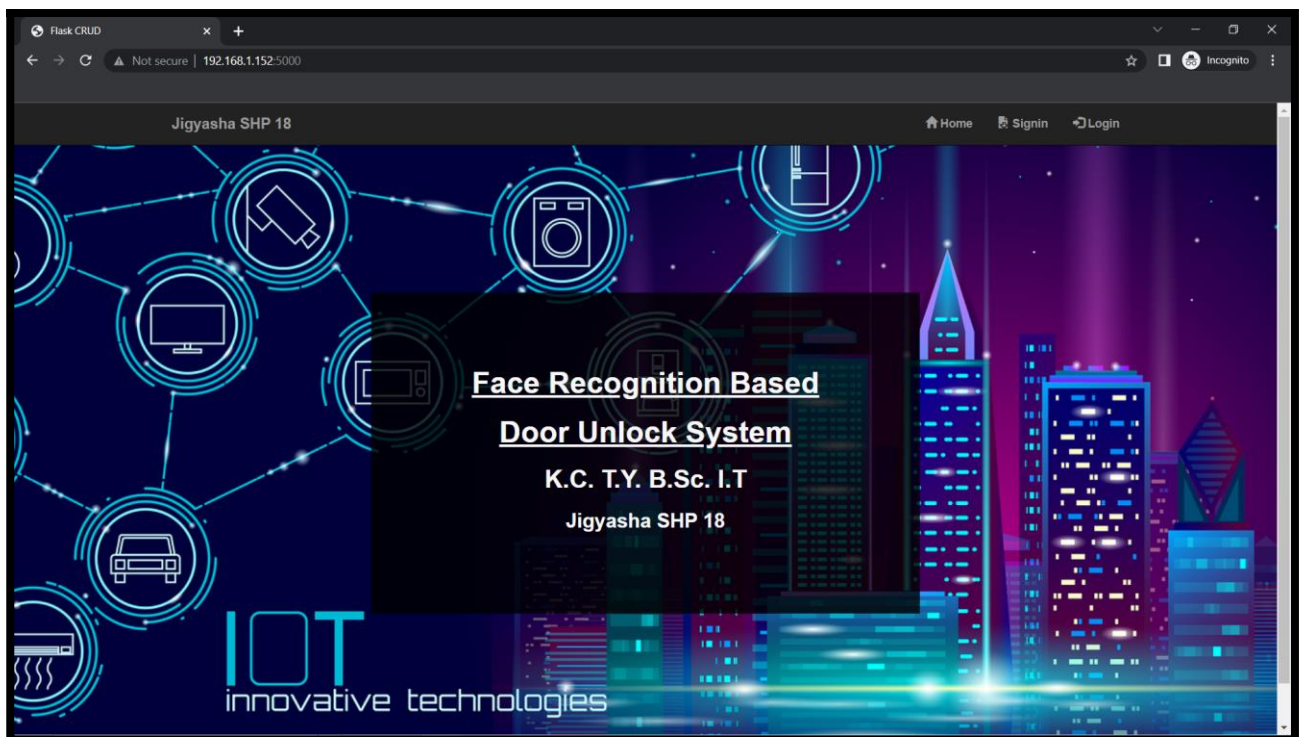
```
if passwd1 == passwd2:
    mycurser = mysql.connection.cursor()
    mycurser.execute("INSERT INTO admin (name, phone, email, password)
        VALUES (%s, %s, %s, %s)",(name, phone, email, passwd2))
    mysql.connection.commit()
    print("Data Inserted Successfully")
else:
    print("Password Doesnot Mached")

if passwd == password:
    session["id"] = id
    print("session id:", session["id"])
    return redirect(url_for('index'))
else:
    print("Wrong Password")
    return render_template("login.html")
```

CHAPTER 6

Result and Discussion





Flask CRUD x +

Not secure | 192.168.1.152:5000/signup

Jigyasha SHP 18 Home Signin Login

Signup

*Every field is mandatory **

Name

Phone Number

Email Address

Password

Confirm Password

Register

Flask CRUD x +

Not secure | 192.168.1.152:5000/signup

Jigyasha SHP 18 Home Signin Login

Signup

*Every field is mandatory **

Name

Phone Number

Email Address

Password

Confirm Password

Register

Please fill out this field.

Flask CRUD x +

Not secure | 192.168.1.152:5000/signup

Jigyasha SHP 18 Home Signin Login

Signup

*Every field is mandatory **

Name

Pravin

Phone Number

Email Address

Password

Confirm Password

Register

Please fill out this field.

Flask CRUD x +

Not secure | 192.168.1.152:5000/signup

Jigyasha SHP 18 Home Signin Login

Signup

*Every field is mandatory **

Name
Pravin

Phone Number
7584320125

Email Address
asc

Password

Confirm Password

Register

Please include an '@' in the email address. 'asc' is missing an '@'.

Flask CRUD x +

Not secure | 192.168.1.152:5000/signup

Jigyasha SHP 18 Home Signin Login

Signup

*Every field is mandatory **

Name
Pravin

Phone Number
7584320125

Email Address
asc@

Password

Confirm Password

Register

Please enter a part following '@'. 'asc@' is incomplete.

Flask CRUD x +

Not secure | 192.168.1.152:5000/signup

Jigyasha SHP 18 Home Signin Login

Signup

*Every field is mandatory **

Name

Phone Number

Email Address

Password

Confirm Password

Please fill out this field.

Register

Flask CRUD x +

Not secure | 192.168.1.152:5000/signup

Jigyasha SHP 18 Home Signin Login

Signup

*Every field is mandatory **

Name

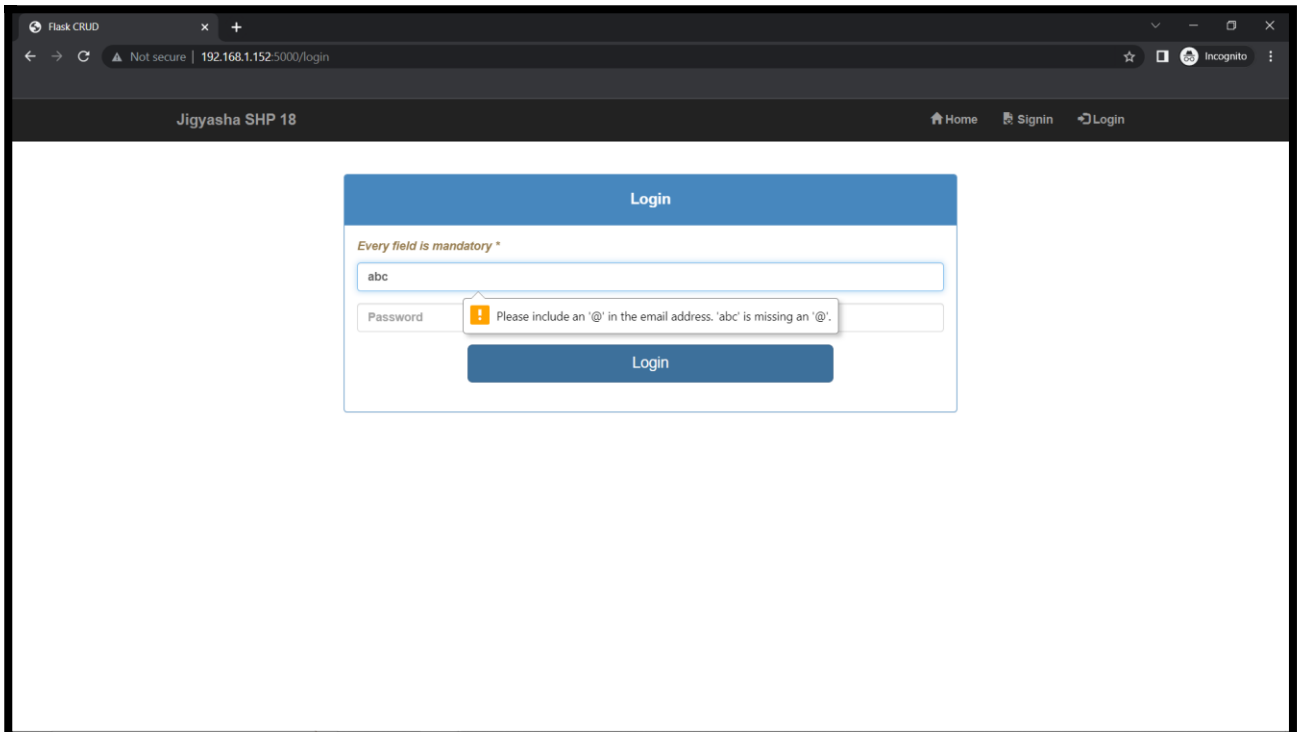
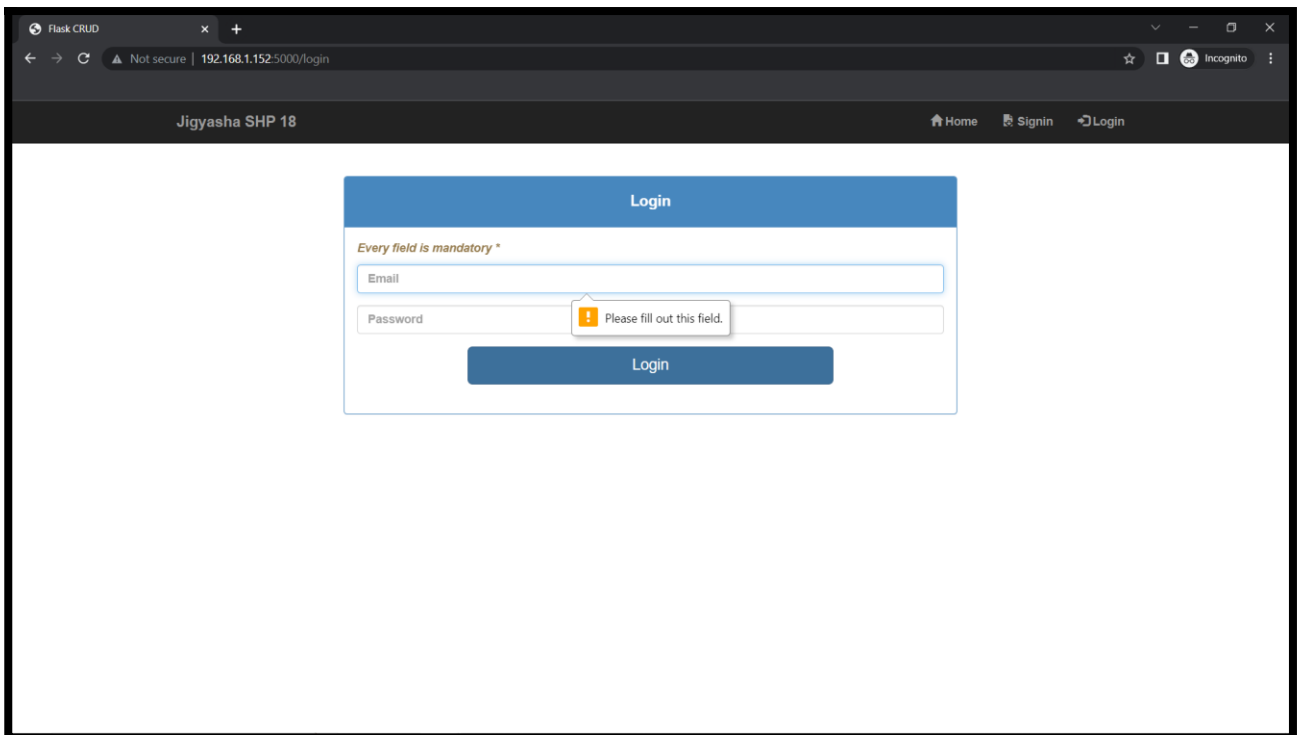
Phone Number

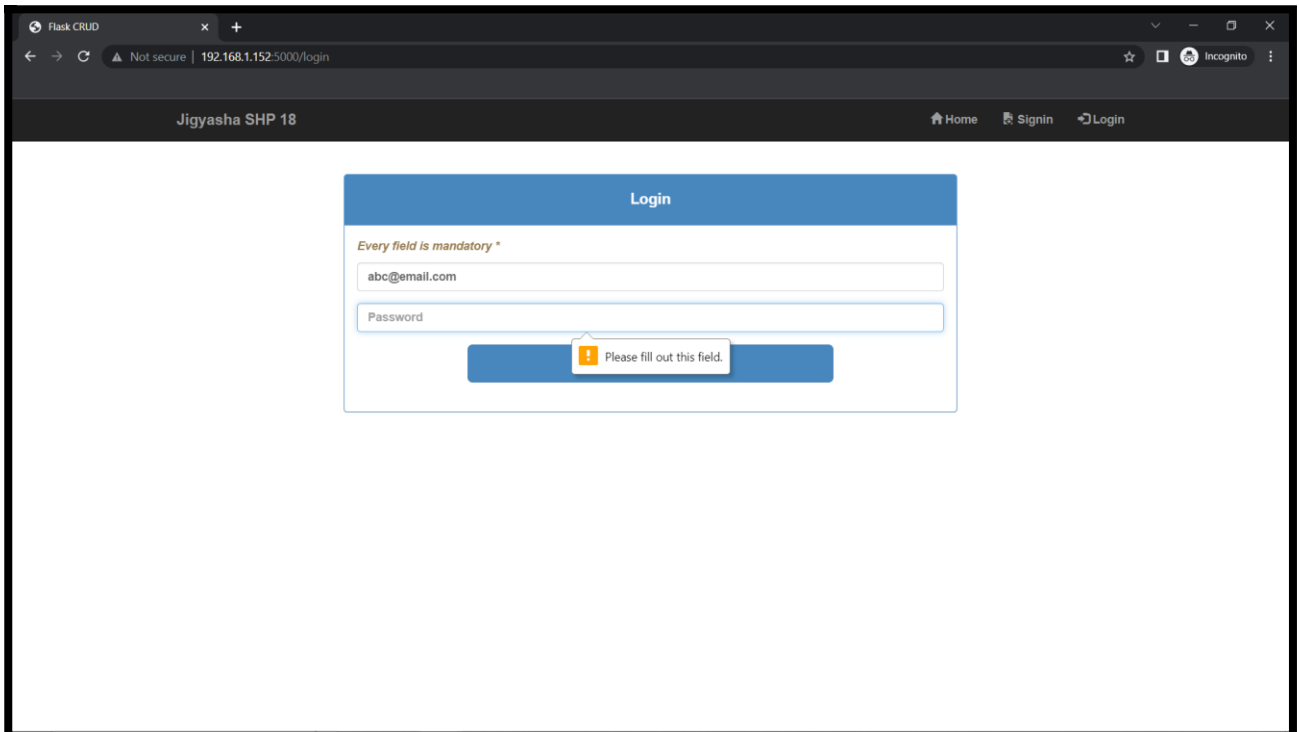
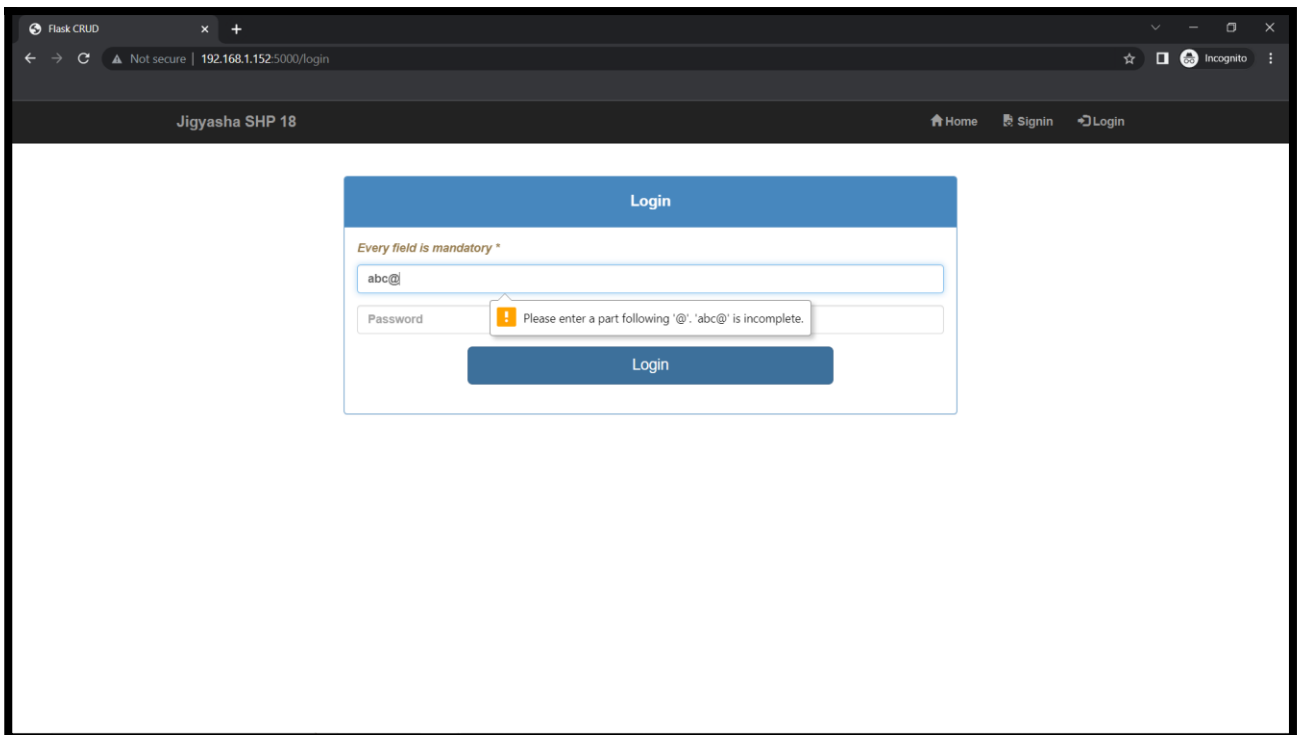
Email Address

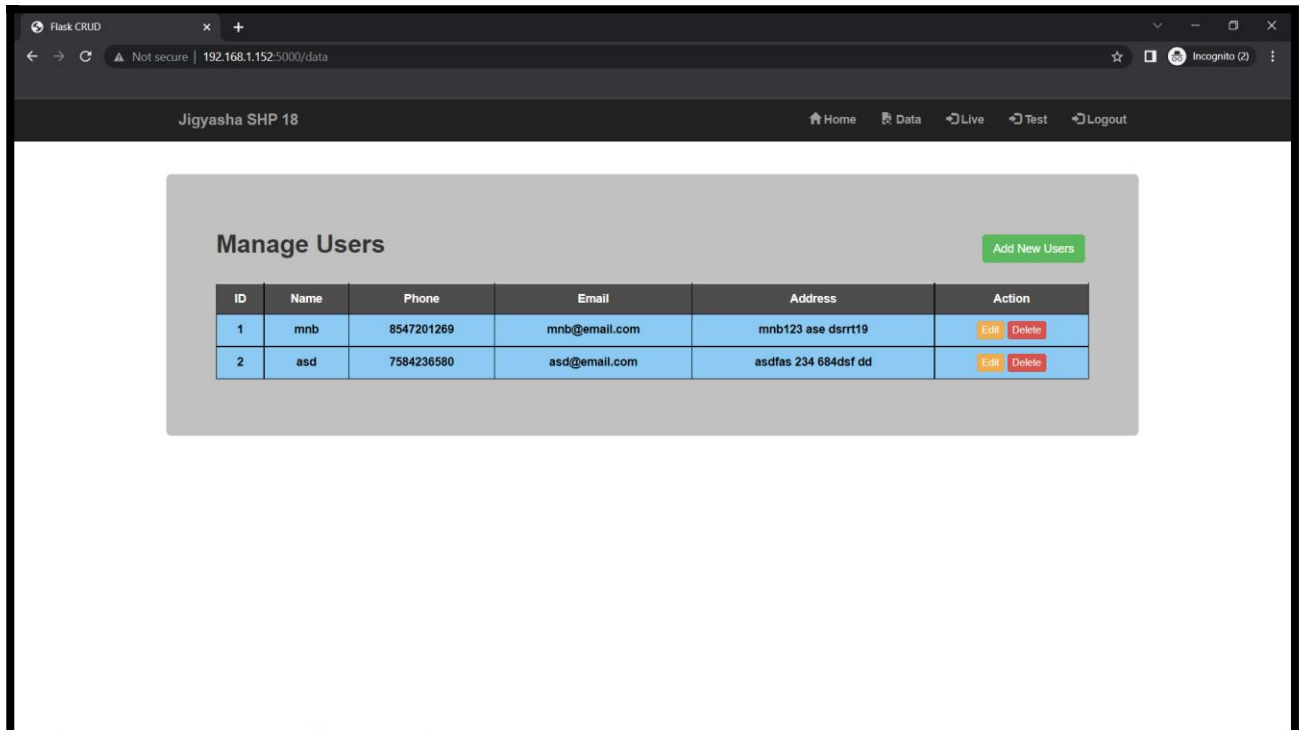
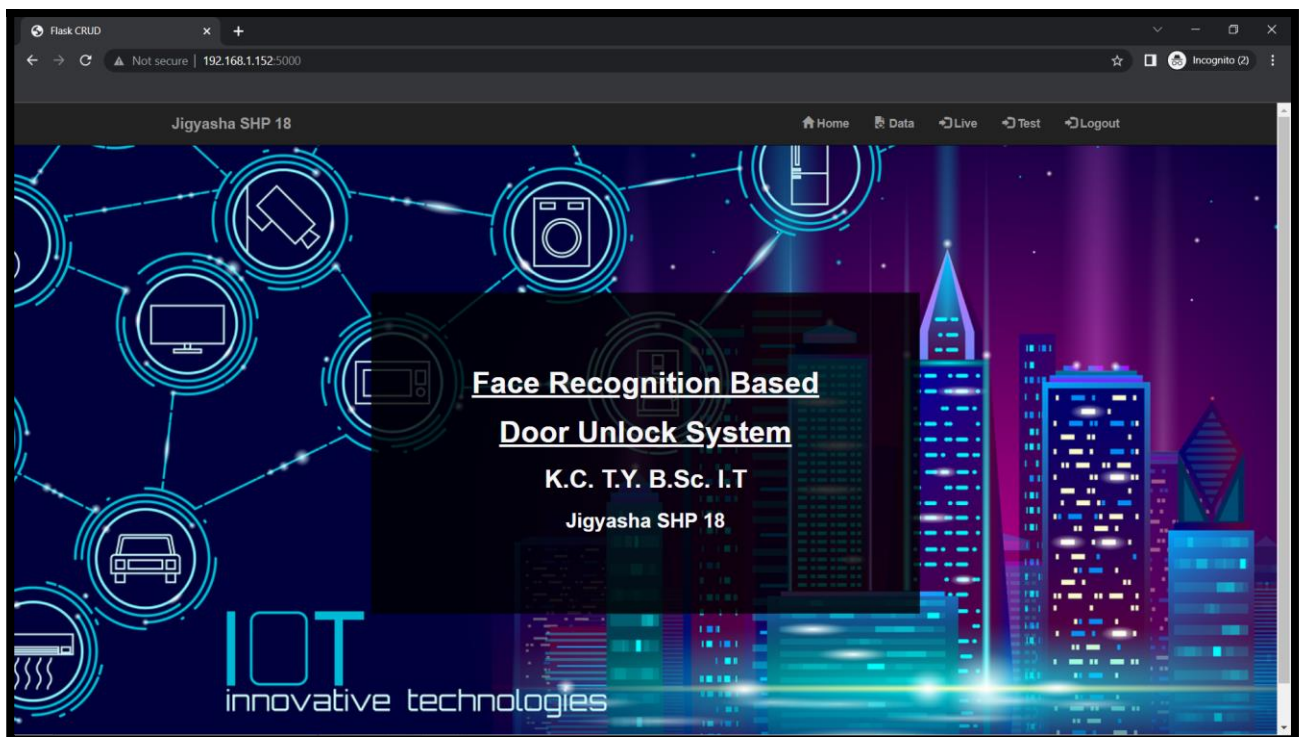
Password

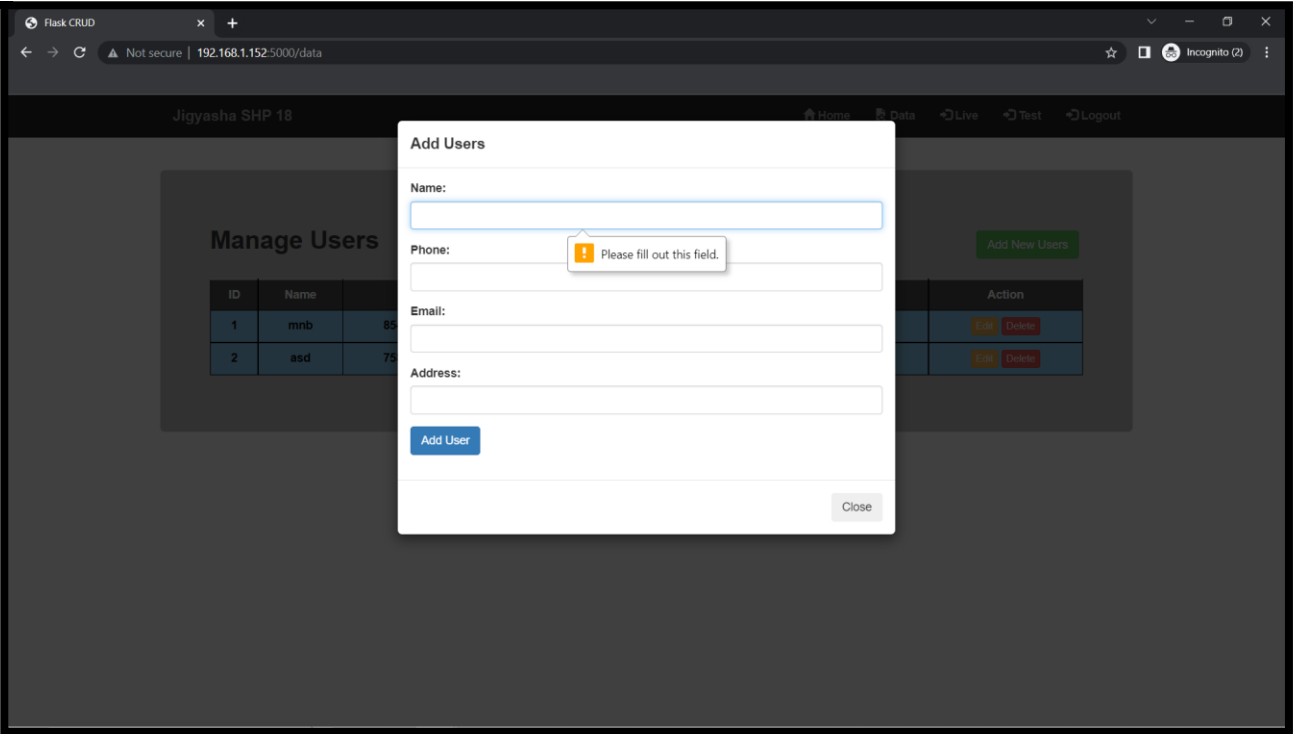
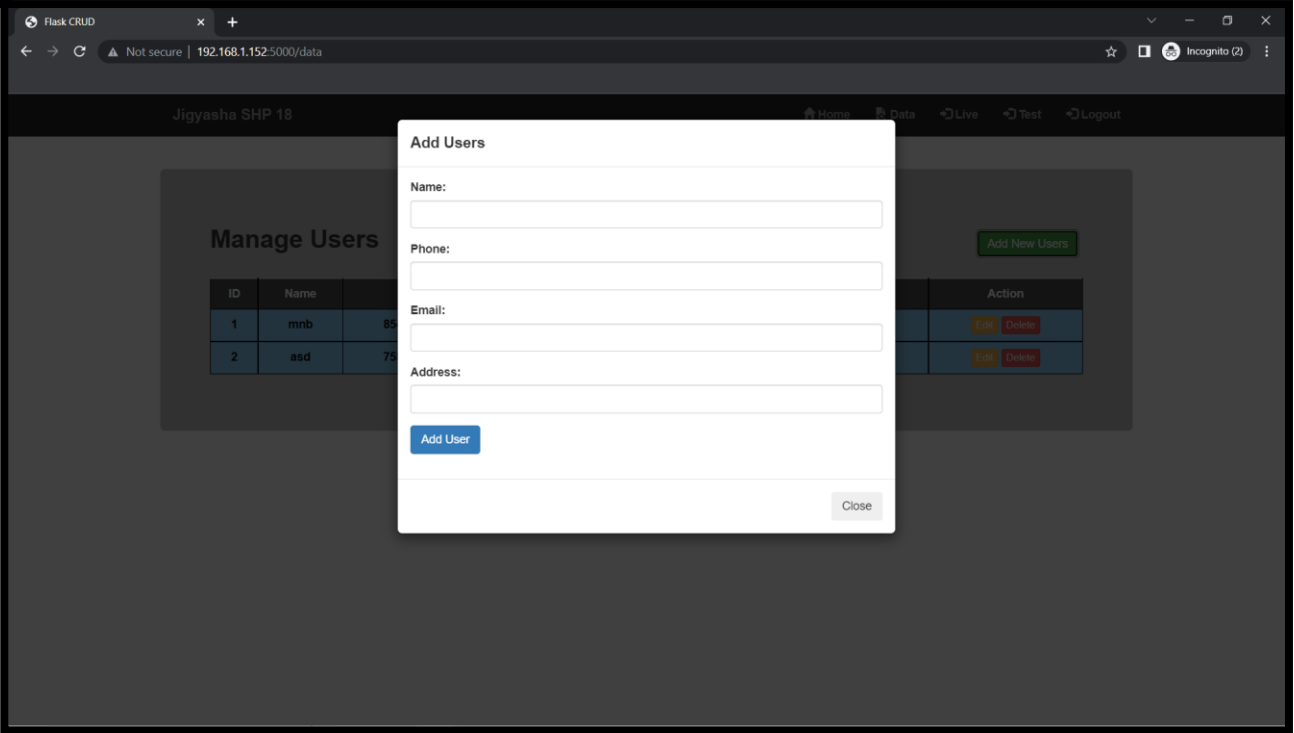
Confirm Password

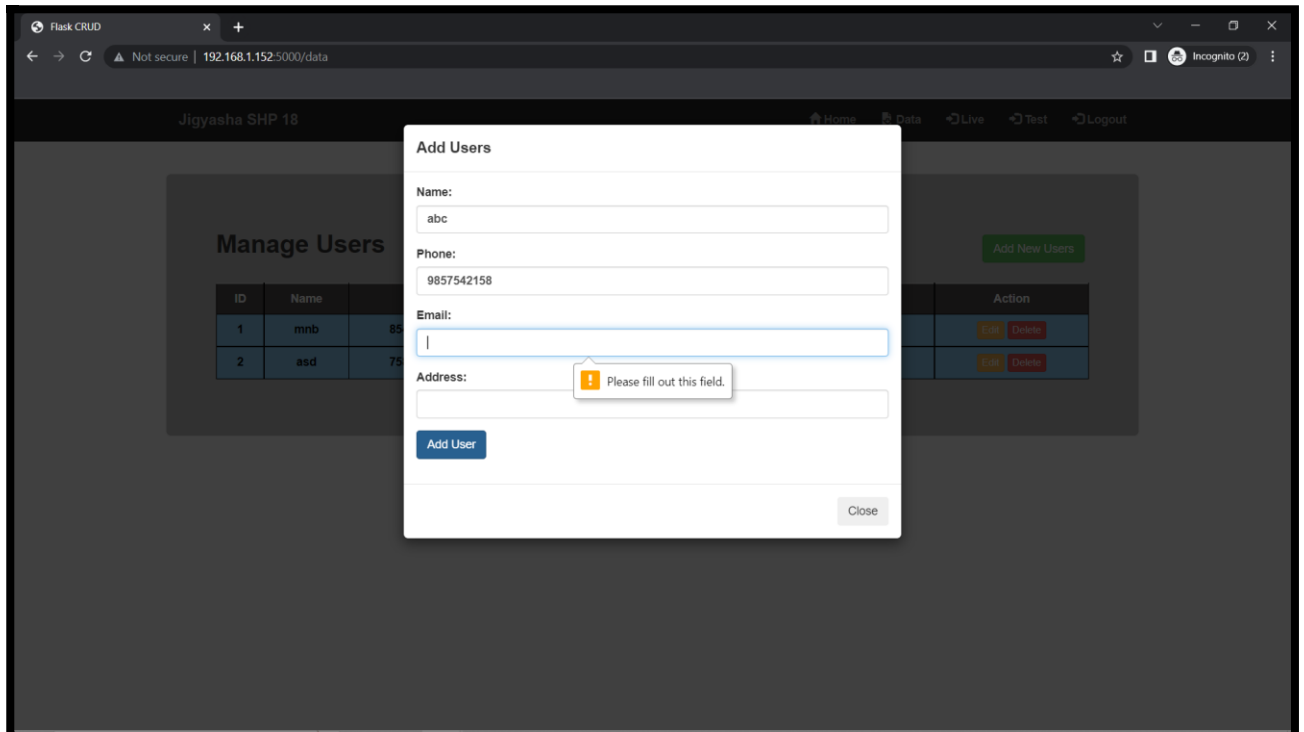
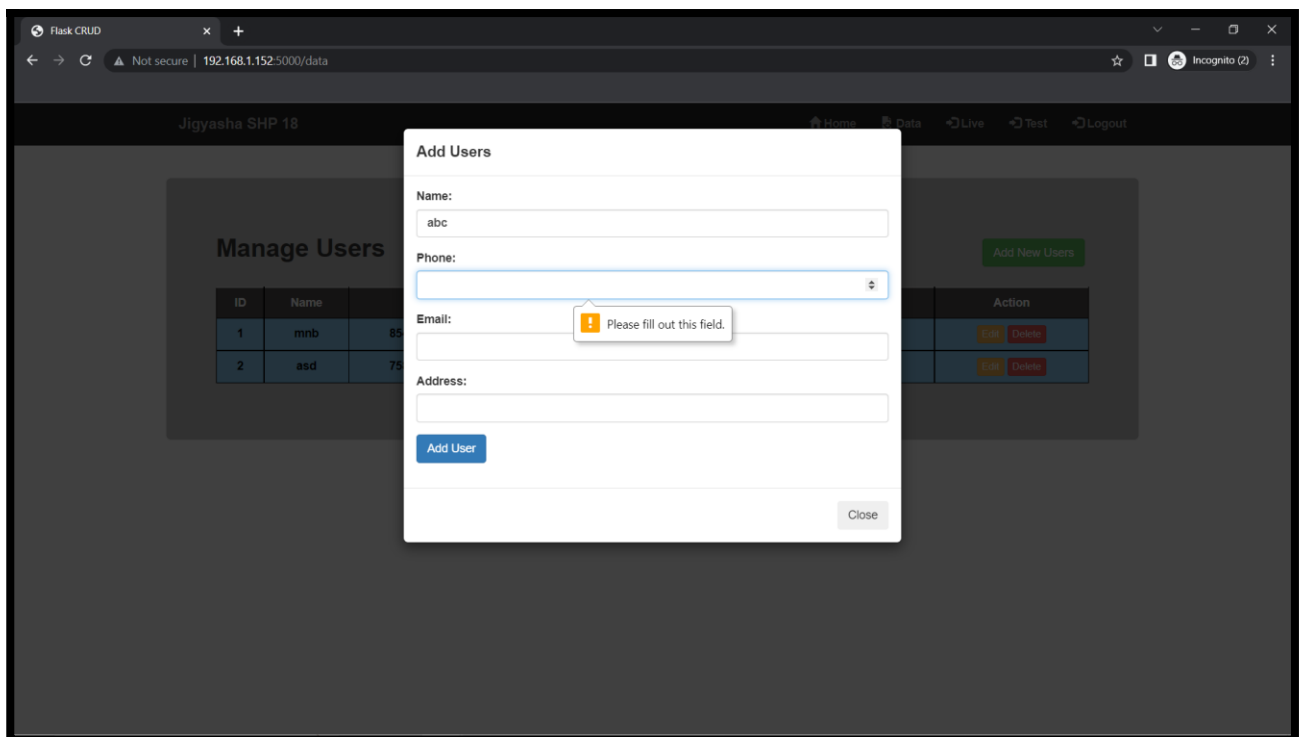
Please fill out this field.

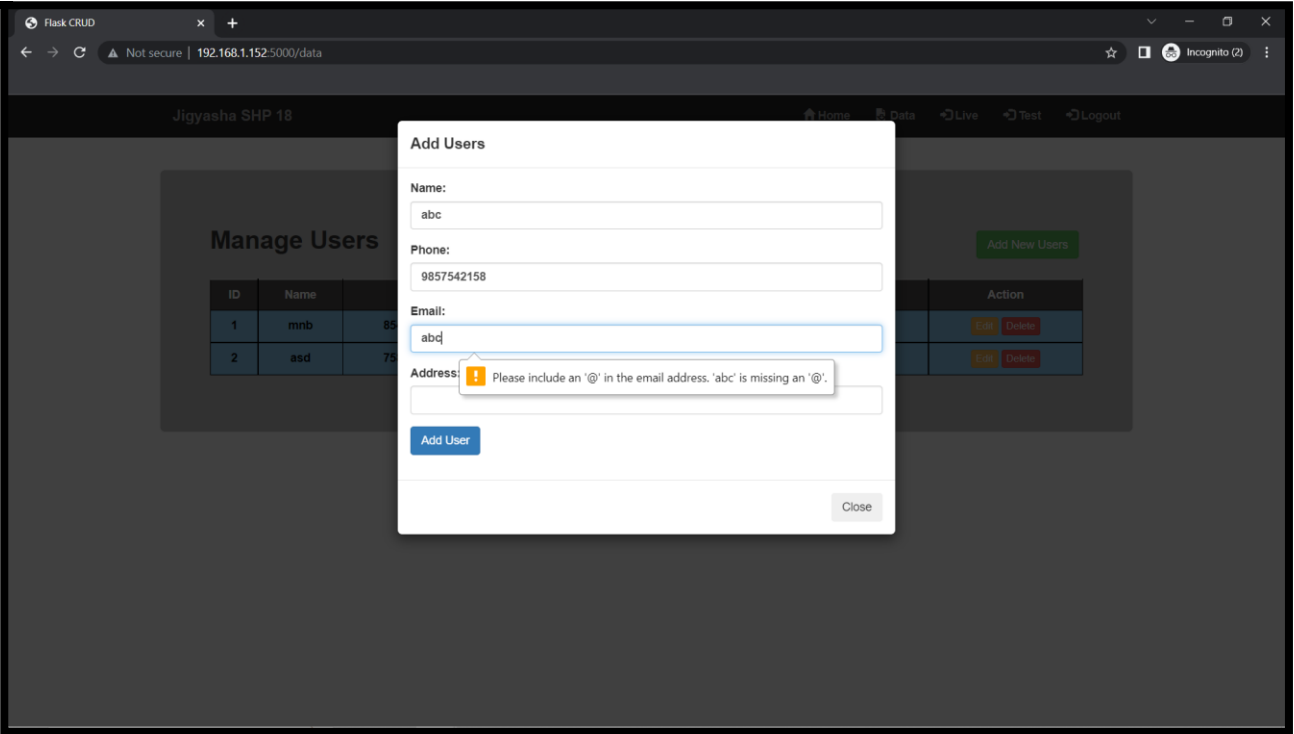
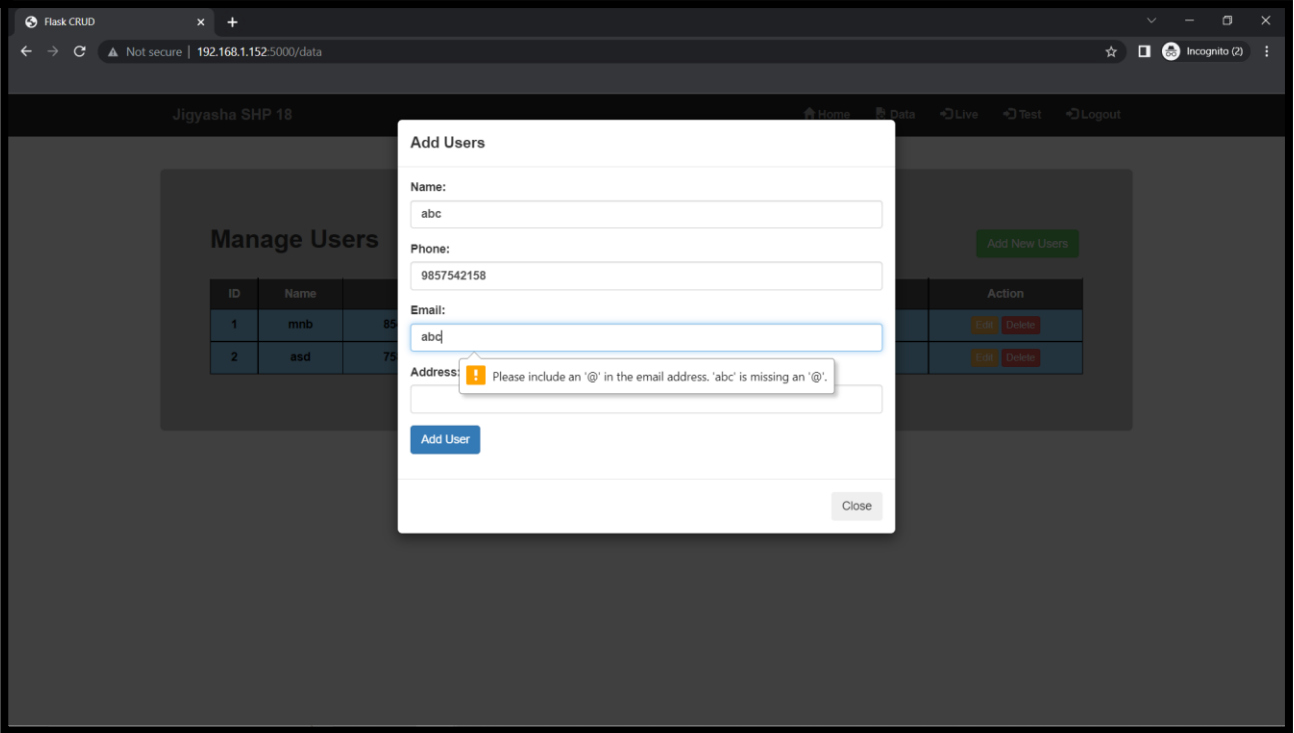


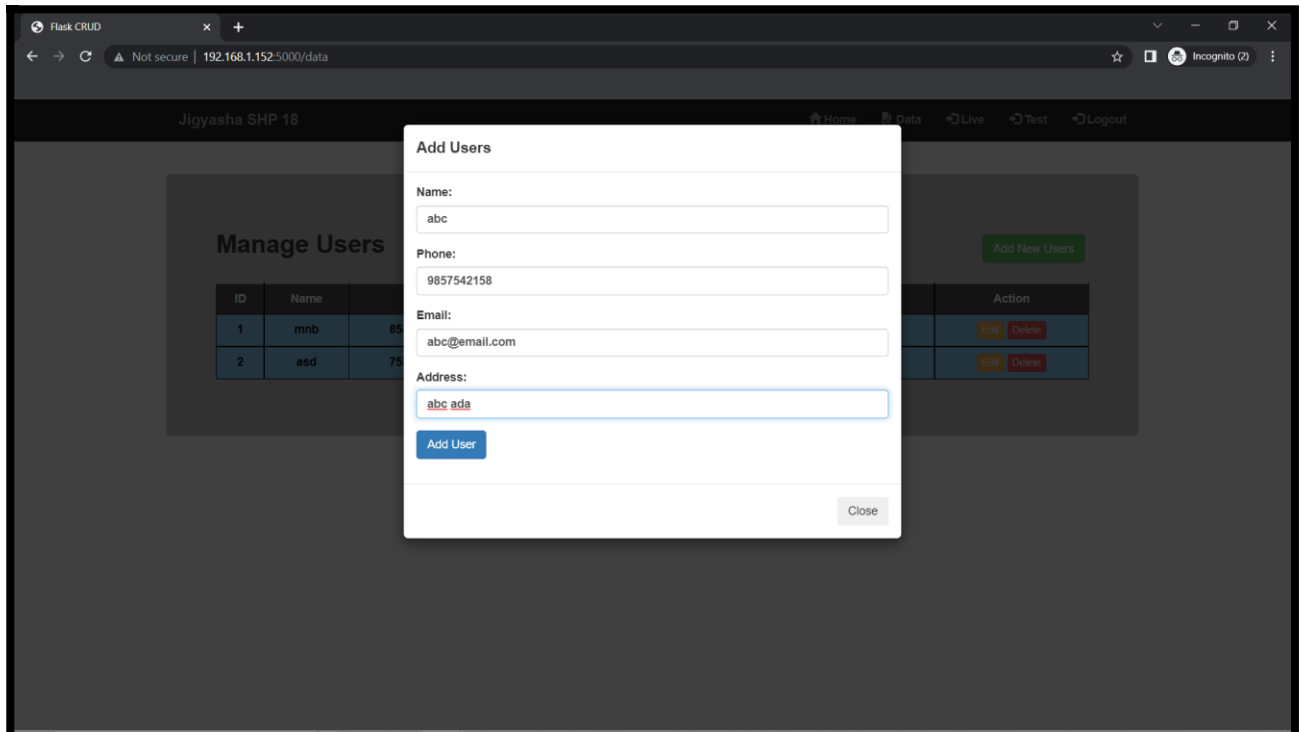
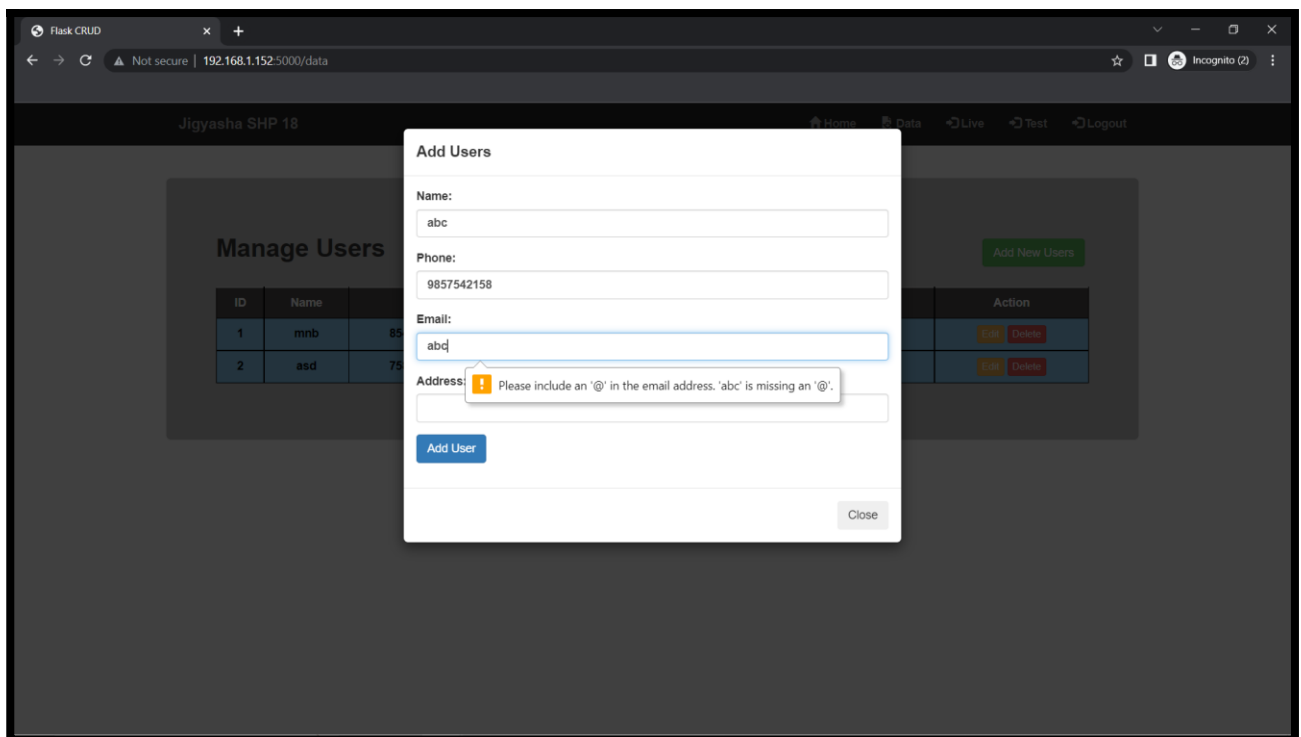


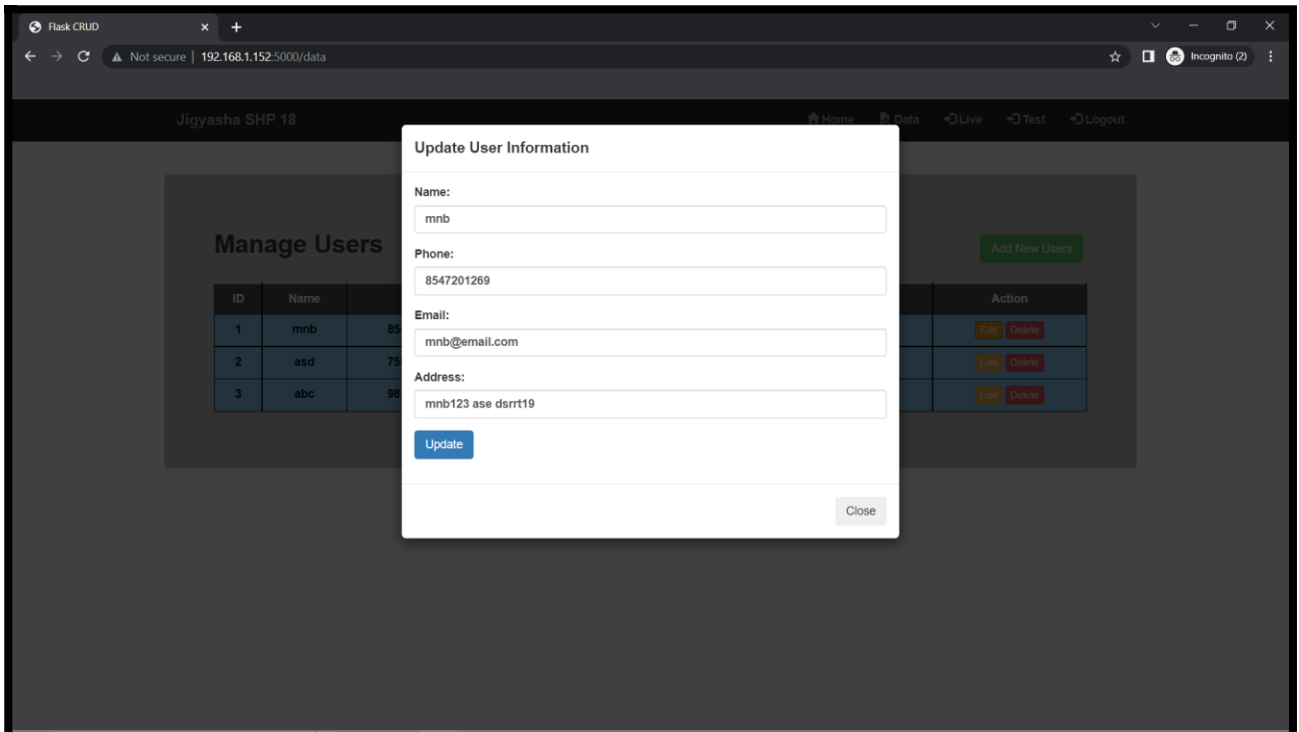
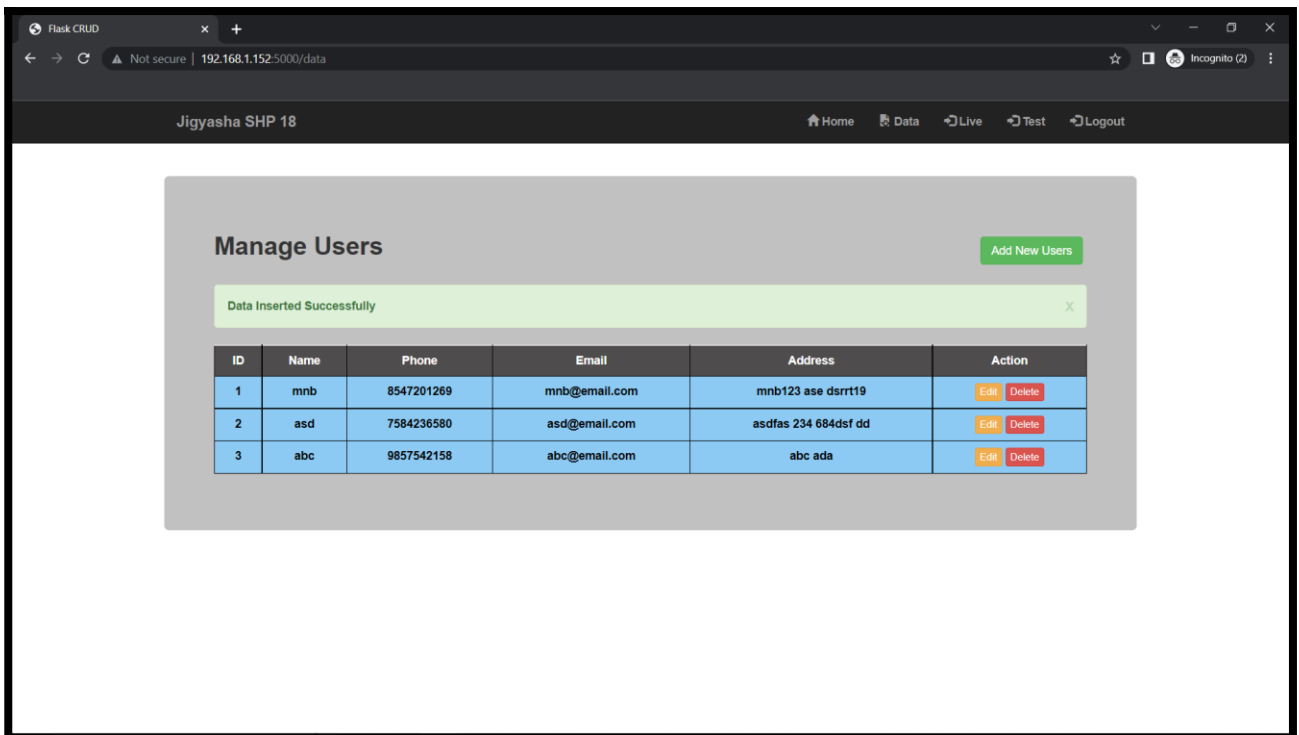


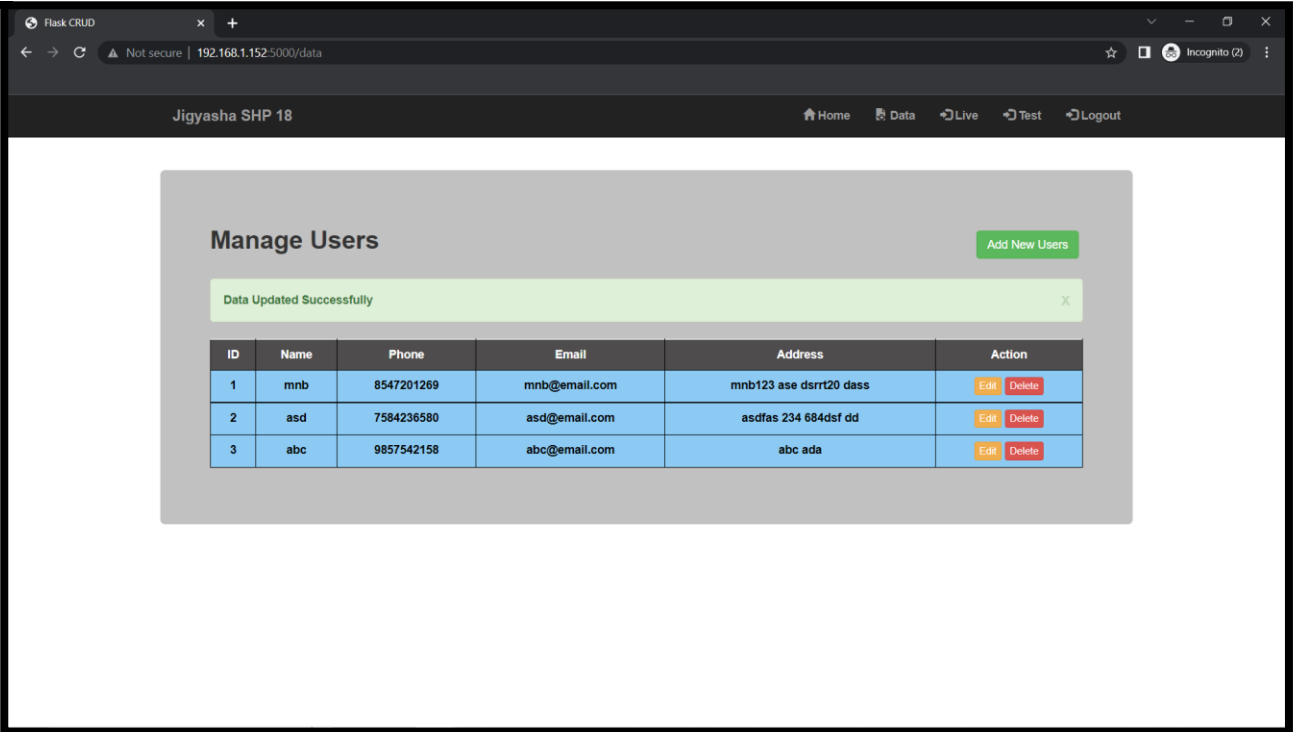
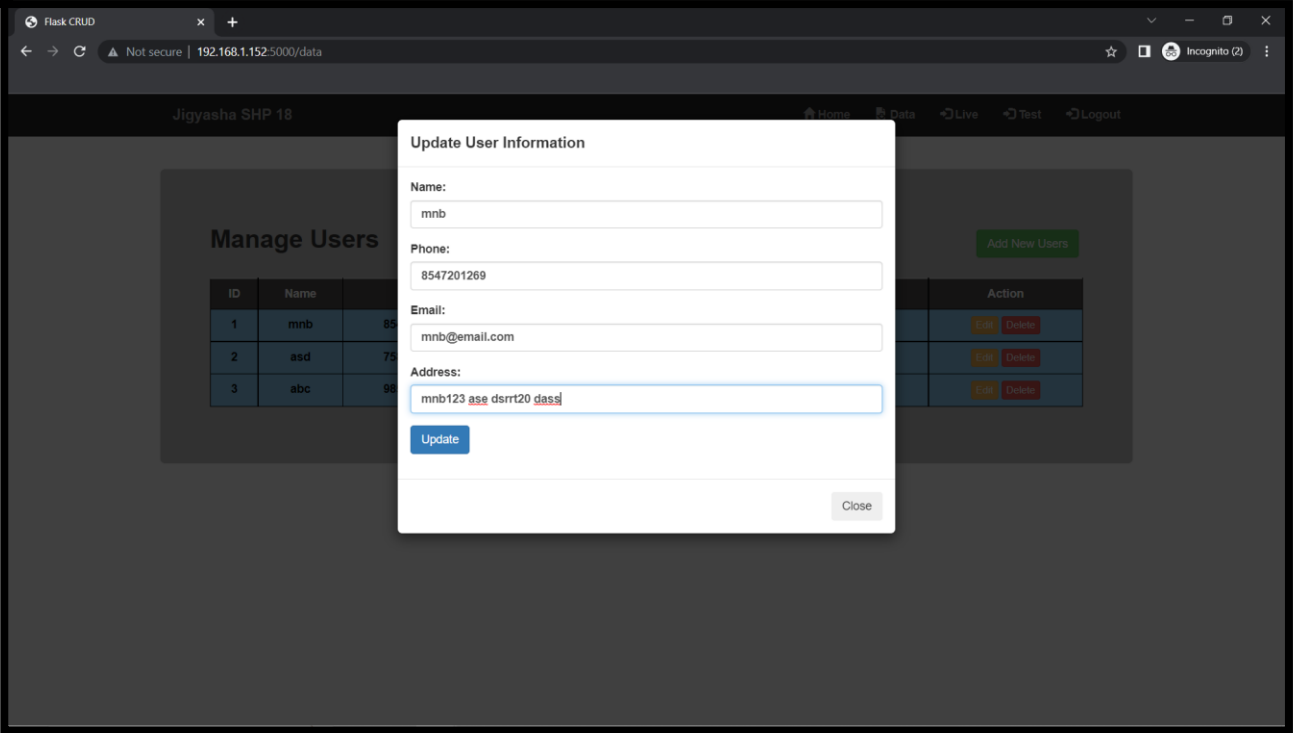


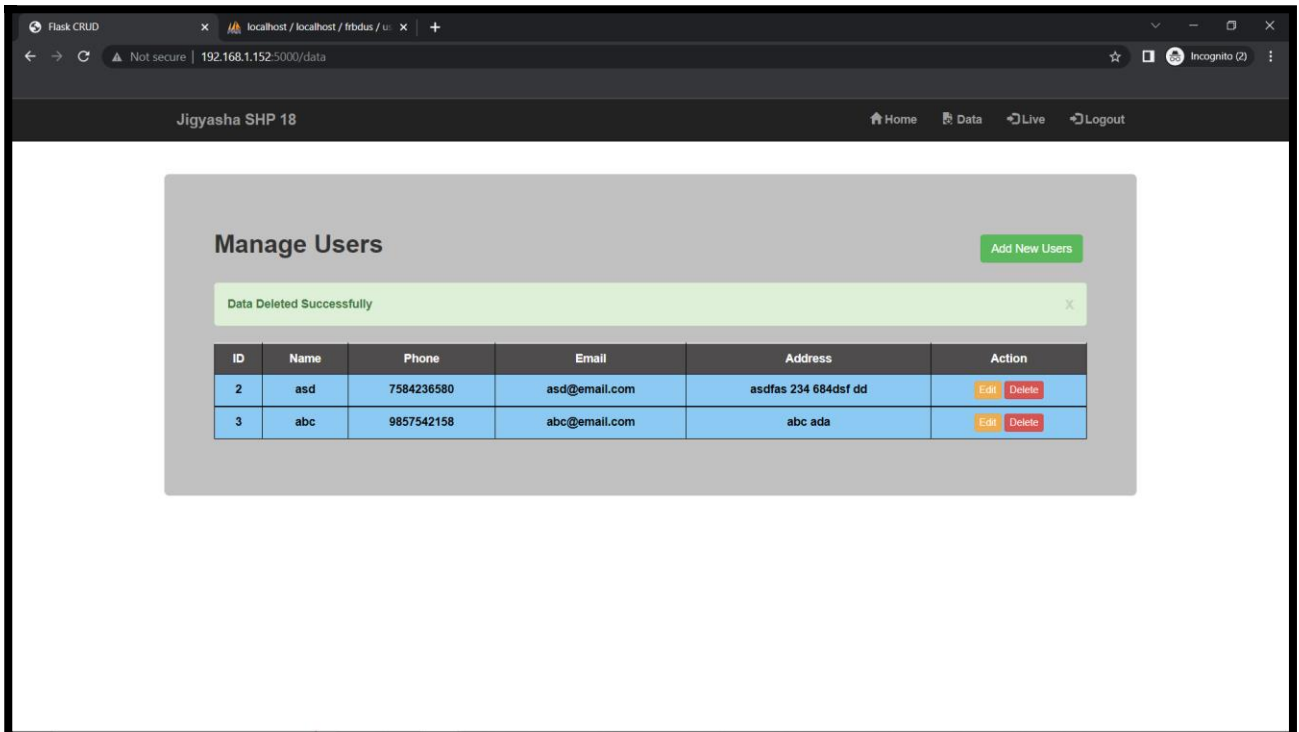
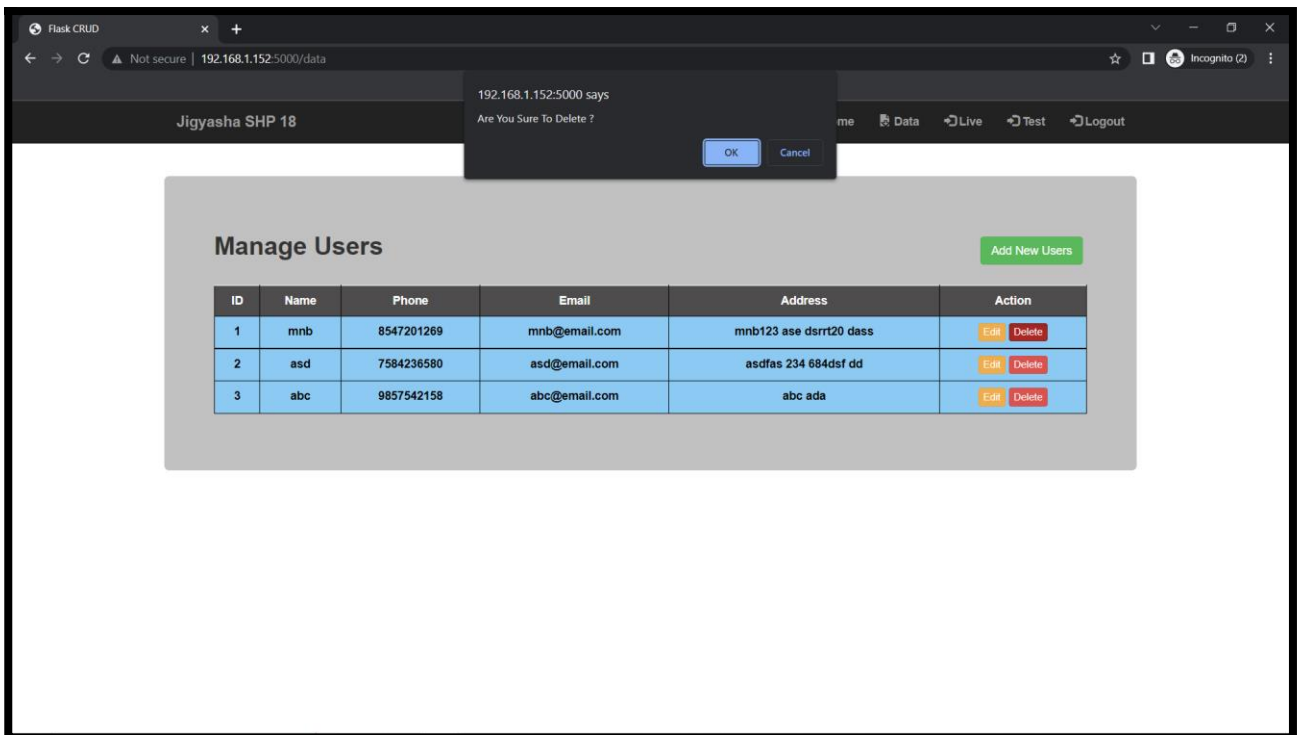


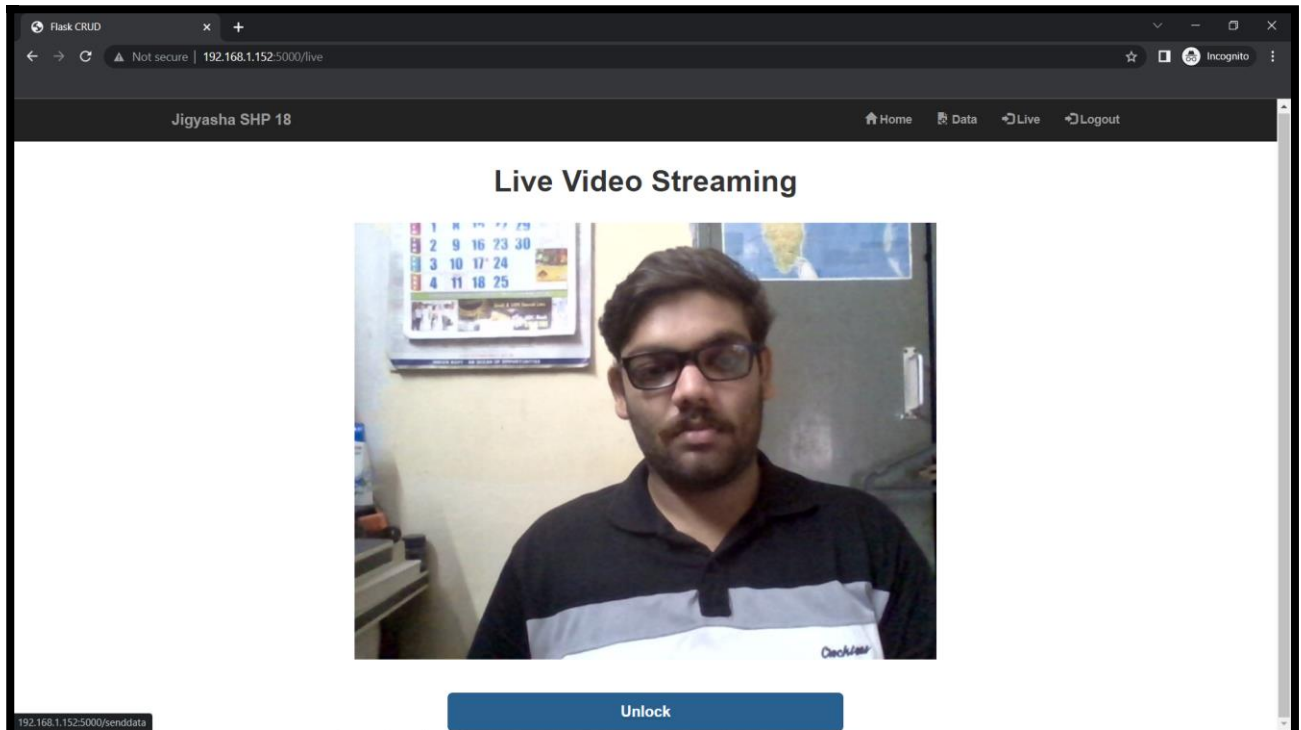
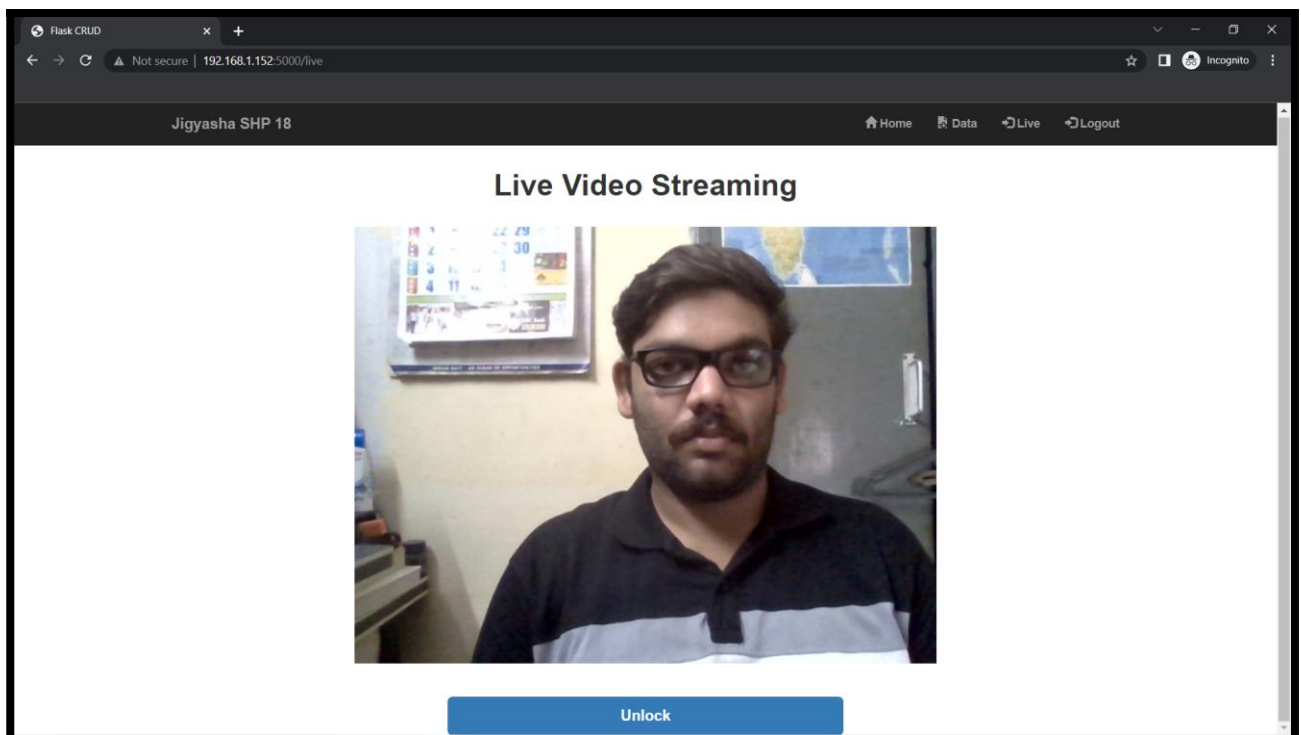












CHAPTER 7

Conclusion

Lock and key, Smart Card based lock, Password based door lock, and other door unlock systems are available on the market. However, if we have a lock and key or a Smart card based door locking mechanism and if our key or smart card is stolen by a thief, or if we have a password based door locking mechanism and a thief sees the password or can unlock it by trying it sometimes, this sort of door lock is not as secure. Face recognition-based security camera and door unlock system, on the other hand, capture the face and alert the administrator. The door cannot be opened without the authorization of the administrator.

Limitation

- The flask app does not send encrypted data to the Raspberry Pi 3, nor does the Raspberry Pi 3 send encrypted data to the flask app.
- The live video streaming is a little slow and sluggish.
- If you don't have an internet connection, it won't work.
- Face images are saved locally rather than in a MySQL database.
- If we connect to another network, we have to update the IP address and port in the code.

Future Scope

- We may integrate a GSM Module for SMS Notification so that whenever there is an Internet problem, the system will send an SMS Notification to the owner immediately.
- All data, including streaming video, will be stored on MySQL Server.
- Using the flask app GUI, we can save the IP address and port of the flask server and socket in MySQL Server.

REFERENCES

Citation:

- Emami, Shervin & Suci, Valentin. (2012). Facial Recognition using OpenCV. Journal of Mobile, Embedded and Distributed Systems. 4.
- Hasan, Ramadan & Sallow, Amira. (2021). Face Detection and Recognition Using OpenCV. Journal of Soft Computing and Data Mining. 2. 10.30880/jscdm.2021.02.02.008.
- M. Khan, S. Chakraborty, R. Astya and S. Khepra, "Face Detection and Recognition Using OpenCV," 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2019, pp. 116-119, doi: 10.1109/ICCCIS48478.2019.8974493.
- <https://pypi.org/project/face-recognition/>
- https://en.wikipedia.org/wiki/Home_security
- https://en.wikipedia.org/wiki/Internet_of_things
- https://en.wikipedia.org/wiki/Artificial_intelligence
- https://en.wikipedia.org/wiki/Machine_learning
- <https://www.iotforall.com/what-is-internet-of-things>
- <https://www.kaspersky.com/resource-center/definitions/what-is-facial-recognition>