Ram Lal Anand College

(University of Delhi)

B.Sc. (H.) Computer Science

(2020 - 2023)

Software Engineering Project Report

Submitted by :

- Harsh Jaiswal (20058570011)
- Kartik Arya (20058570018)
- Manish Choudhary (20058570020)
- Vishal (20058570046)

Submitted to:

- Dr. Vandana Gandotra

CERTIFICATE

PROUDLY PRESENTED TO

This is to certify that Software Engineering project report entitled " **Multiple Face Recognition Attendance System** " is the work carried out by Harsh Jaiswal, Kartik Arya, Manish Choudhary, Vishal Yadav students of B.Sc.(H) Computer Science 4th Semester, Ram Lal Anand College , University of Delhi under the supervision of Dr. Vandana Gandotra . This report has not been submitted to any other organisation/institution for the award of any other degree/diploma.

DR.VANDANA GANDOTRA	DR.RAKESH KUMAR GUPTA	
SUPERWISER	PRINCIPAL	EXTERNAL EXAMINER

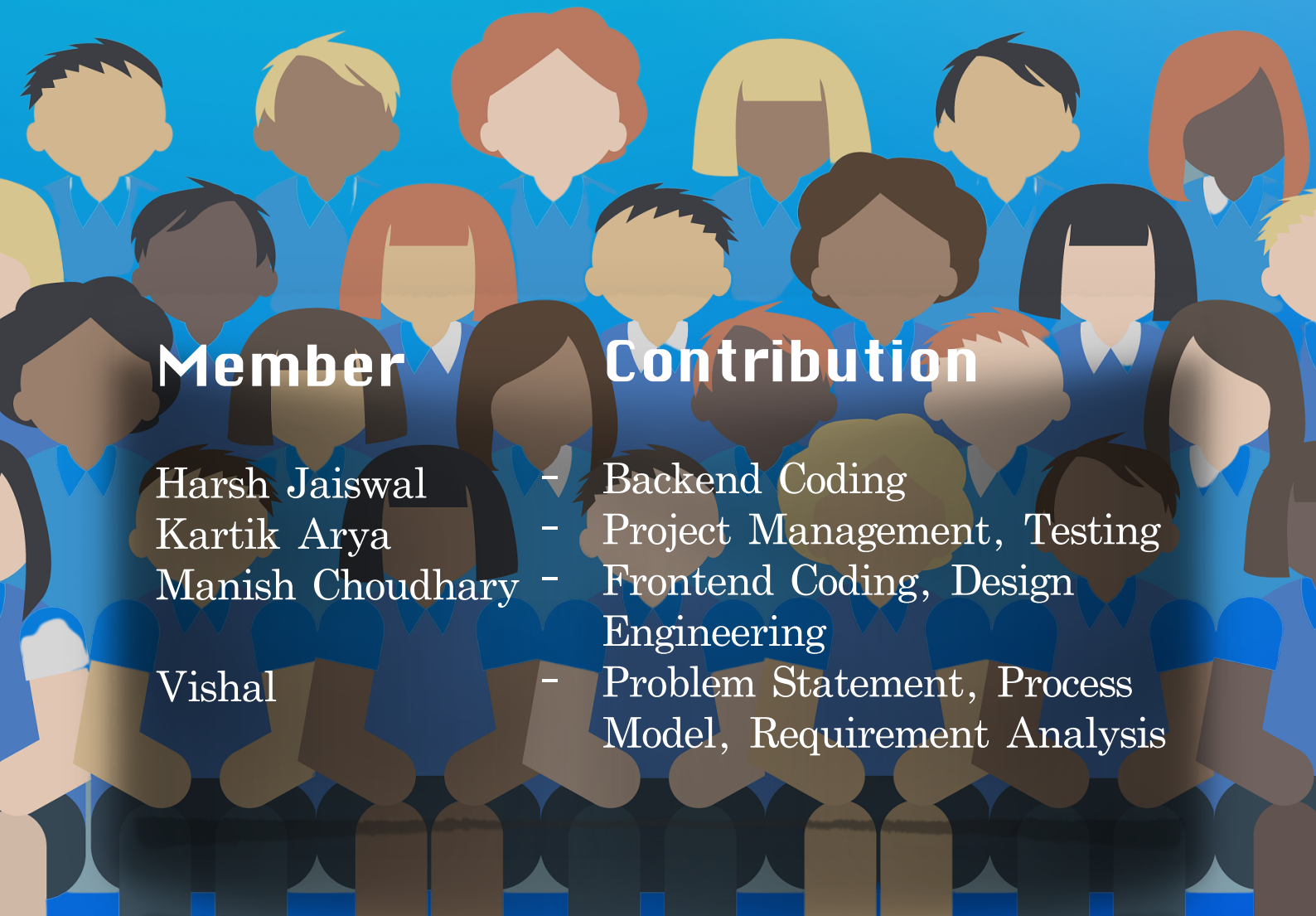
ACKNOWLEDGEMENT

In successfully completing this project, many people have helped me. I would like to thank all those who are related to this project.

Primarily, I would thank God for being able to complete this project with success. Then I will thank my Principal Dr. Rakesh Kumar Gupta and Subject teacher Dr. Vandana Gandotra under whose guidance I learned a lot about this project. Her suggestions and directions have helped in the completion of this project.

Finally, I would like to thank my parents , relatives and friends who have helped me with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

MULTIPLE FACE RECOGNITION ATTENDANCE SYSTEM



Member		Contribution
Harsh Jaiswal	-	Backend Coding
Kartik Arya	-	Project Management, Testing
Manish Choudhary	-	Frontend Coding, Design Engineering
Vishal	-	Problem Statement, Process Model, Requirement Analysis

ABSTRACT

Nowadays Educational institutions are concerned about regularity of student attendance. This is mainly due to students' overall academic performance is affected by his or her attendance in the institute. Mainly there are two conventional methods of marking attendance which are calling out the roll call or by taking student sign on paper. They both were more time consuming and difficult. Hence, there is a requirement of computer-based student attendance management system which will assist the faculty for maintaining attendance record automatically.

In this project we have implemented the automated attendance system using Python. We have projected our ideas to implement "**Multiple Face Recognition Attendance System**", in which it imbibes large web applications. The application includes face identification, which saves time and eliminates chances of proxy attendance because of the face authorization. Hence, this system can be implemented in a field where attendance plays an important role.

The system is designed using **PYTHON,HTML & EEL**. The proposed system uses Principal Component Analysis (PCA) algorithm which is based on eigenface approach. This algorithm compares the test image and training image and determines students who are present and absent. The attendance record is maintained in a database which is updated automatically in the system.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Existing System 2
1.2 Feature of projects 4

2 PROBLEM STATEMENT

2.1 Process Model 5

3 REQUIREMENT ANALYSIS

3.1 Data Flow Diagram 8
3.2 Sequence Diagram 10
3.*3 Software Requirement* 11
3.*4 Use Case Diagram* 12
3.*5 Data Dictionary* 14

4 PROJECT MANAGEMENT

4.1 Timeline Charts 15
4.*2 Computing FP* 16
4.*3 Effort* 17
4.*4 Cost* 18
4.*5 Risk management* 19

5 DESIGN ENGINEERING

5.1 Architectural Design 21
5.*2 Pseudo code* 24
5.*3 UI Design* 25

6 ALGORITHM 30
7 CODING 32
8 TESTING 33
9 LIMITATIONS AND FUTURE SCOPE 46
10 BIBLIOGRAPHY 47

1. Introduction

Uniqueness or individuality of an individual is his face and its features. In this project, face of an individual is used for the purpose of marking attendance automatically. Conventional methodology for taking attendance is by calling the name or roll number of the student and the attendance is manually recorded. The important point of concern with this methodology is Time Consumption. On an average it takes an essential part of the time allotted for the subject. Attendance is prime important for both the teacher and student of an educational organization. So, it is very important to keep record of the attendance. The problem arises when we think about the traditional process of taking attendance in class room. Calling out the name or roll number of the student for attendance is not only a problem of time consumption but also consumes energy. So, an automated attendance system can resolve all the above problems. To stay away from these losses, an automated process has been used in this project which is based on Image Processing. In this project, the two main components that have been used are face detection and face recognition.

Face detection is used to locate the position of the face region and face recognition is used for marking the understudy's attendance. The database of all the students in the class is stored and when the students' faces matches with stored faces in database then the attendance is recorded.

There exists a wide range of automatic attendance management systems which are currently being used by many institutions and organizations. One such system is the biometric based attendance system. Although it is automatic and leaps a step ahead of the traditional method it fails to meet the time constraint because the students have to wait in a queue for marking attendance, which is also a time taking process. This system eliminates classical student identification such as calling names of the students, or checking respective identification cards of the students, which can not only interfere with the ongoing teaching process, but also can be stressful for students during examination.

1.1 Existing Attendance Systems

Attendance is prime important for both the teacher and student of an educational organization. So, it is very important to keep record of the attendance. There are various attendance management systems that vary in complexity and feasibility. We have divided them into three categories namely, basic, moderate, and advanced.

1.1.1. Basic Attendance Systems

a. Manual Attendance System: The Manual Attendance System involves the process of the faculty calling out the roll calls. If the student is present in the class, the student physically acknowledges the roll call and says that he/she is present. In all other cases, the faculty marks the student absent.

b. Paper Based Attendance System: The Paper Based Attendance System is a part of the manual attendance system or could be used for any other attendance system as well. Attendance is taken in any form and it's recorded on a paper by writing either the absentees, or the presentees only. Usually faculties write the roll numbers of the students that are absent or present as per convenience.

c. Timesheet Attendance System: The Timesheet Attendance System involves recording attendance into a timesheet. A timesheet is a physical or virtual tool that allows you to record and keep track of your worked time, in this case it's number hours the student attends.

1.1.2 Moderate Attendance Systems

a. Biometric Attendance System: The biometric attendance system works on two basic principles. First, it takes an image of a finger. Then finger scanner saves characteristics of every unique finger and saved in the form of biometric key. Actually, finger print scanner never saves images of a finger only series of binary code for verification purpose. Secondly, the biometric attendance system determines whether the pattern of ridges and valleys in this image matches the pattern of ridges and valleys in pre-scanned images.

b. Swipe Card Attendance: The Swipe Card Attendance System works by the person swiping its card on entry and exit of the gate, and the attendance is recorded. A swipe card must come in contact with the corresponding card reader before any transaction can take place. The transaction becomes active when the magnetic stripe on a card is moved through a console at a gate.

1.1.3 Advanced Attendance Systems

a. Retinal Scan-based Attendance System: The Retinal Scan-based Attendance System makes uses of retinal features and marks attendance on retinal recognition. Eye scan or retinal scan is a biometric system that identifies a person by using unique patterns of the retina. Human retina contains a complex blood vessel (retinal vein) patterns through which an eye scanner device can easily identify a person and can even differentiate identical twins. To scan human retina, retinal scanner uses the reflection of light that is absorbed by retinal vein.

b. Facial Recognition Attendance System: The Facial Recognition Attendance System makes use of facial features such as distance between the eyes, width of the nose, depth of the eye sockets, the shape of the cheekbones, the length of the jaw line, etc. to recognize and mark attendance.

1.1.4 Drawbacks of Existing Systems:

a. Accuracy: With an automated system, there is no human error. When you manually track your students' time, your students typically report their hours after they've worked them. This will often increase the likelihood of inaccurate reporting. A student may not intend to 6 misrepresent his hours, he may just forget what his actual in and out times were. Or, if a student has illegible handwriting, it could make it difficult for roll list to determine actual hours attended. With manual reporting, the organization is basically relying on the honour system. This system can be abused, which can lead to time theft.

b. Increased Productivity: Organizations who use a manual roll call process spend several hours each day collecting time cards, re-entering an illegible data by hand, faxing , phoning, and processing roll-call. When you employ an automated time and attendance system, the roll-call process takes just minutes each period.

c. Savings: With an automated system, you'll save roll-call processing hours and eliminate time theft which means your bottom line will improve. Even though there exist single face recognition systems but they are less efficient as they require each students to face the recognition system individually which ultimately take more time and therefore time costly. Solution to such issue is to automatically recognize multiple faces at a time.

1.2 Salient Features of the Proposed System:

a. Face-mapping: Facial features of the student such as distance between the eyes, width of the nose, depth of the eye sockets, the shape of the cheekbones, the length of the jaw line, etc. are registered into the database. Students are recognized based on these stored facial features, and if a match is found, the student is marked present and the same is updated into the database. In all other cases, the student is recorded absent in the database.

b. Three-step Management: The entire system is efficiently split into three components. Firstly, the facial features are stored, and the model is trained. Second, the student is recognized by facial recognition algorithm. Third, the attendance is automatically updated into the database.

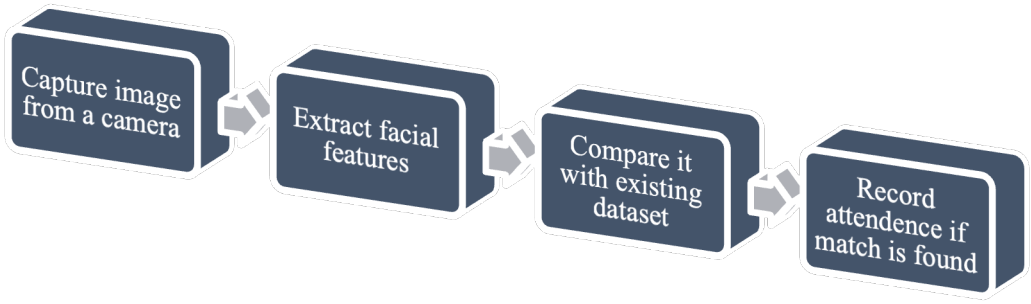
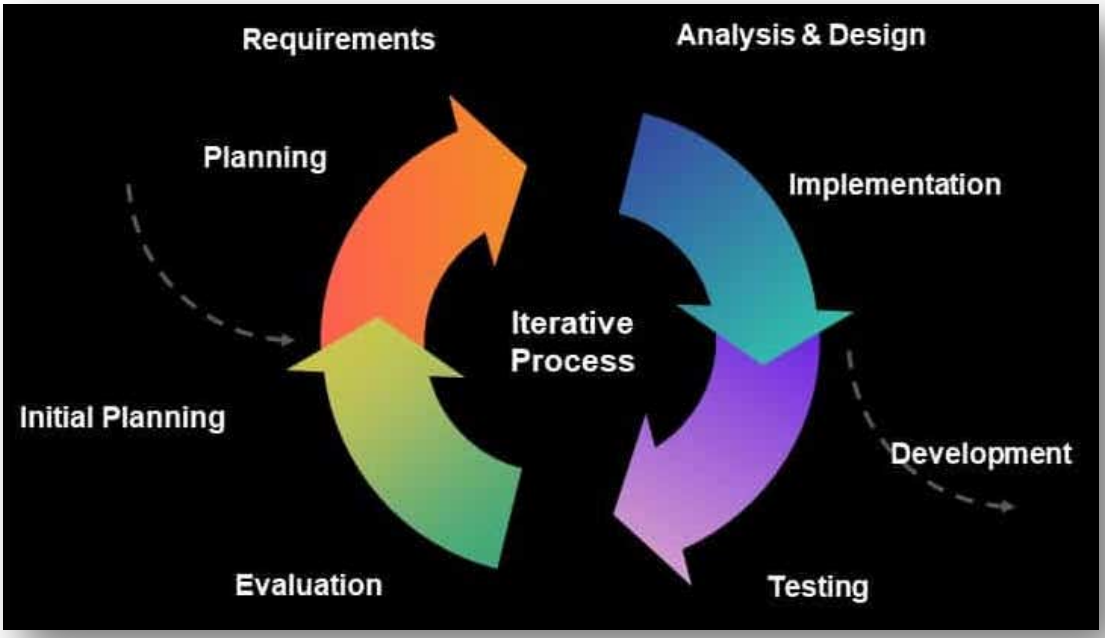


DIAGRAM OF GENERAL FRAMEWORK OF THE SYSTEM

2. Problem Statement

Attendances of every student are being maintained by every school, college and university. Empirical evidences have shown that there is a significant correlation between students' attendances and their academic performances. There was also a claim stated that the students who have poor attendance records will generally link to poor retention. Therefore, faculty has to maintain proper record for the attendance. The manual attendance record system is not efficient and requires more time to arrange record and to calculate the average attendance of each student. Hence there is a requirement of a system that will solve the problem of student record arrangement and student average attendance calculation. One alternative to make student attendance system automatic is provided by facial recognition.

2.1 Process Model



A software process model is an abstract representation of a process that presents a description of a process from some particular perspective. This project has been based on the **'Iterative model'**. The iterative model is also called an incremental model in which a particular project or software is broken down into large numbers of iterations, where each iteration is a complete development loop resulting in a release of executable product or software. A subset of the final product under development, which grows from iteration to iteration to become the final product or software.

The various phases of Iterative model are as follows:

· **Requirement gathering & analysis**– In this phase, requirements are gathered from students and check by an analyst whether requirements will fulfil or not. After all of this, the software team skips to the next phase.

· **Design** – In the design phase, team design the software by the different diagrams like Data Flow diagram, activity diagram etc.

· **Implementation** – In the implementation, requirements are written in the coding language and transformed into computer programmes which are called Software.

· **Testing**– After completing the coding phase, software testing starts using different test methods. There are many test methods, but the most common are white box, black box, and grey box test methods.

· **Deployment of system**– After completing all the phases, software is deployed to its work environment.

·**Review** − In this phase, after the product deployment, review phase is performed to check the behaviour and validity of the developed product. And if there are any error found then the process starts again from the requirement gathering.

· **Maintenance** − In the maintenance phase, after deployment of the software in the working environment there may be some bugs, some errors or new updates are required. Maintenance involves debugging and new addition options.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model".

When to use the Iterative Model?

1. When requirements are defined clearly and easy to understand.
2. When the software application is large.
3. When there is a requirement for changes in future.

Advantage(Pros) of Iterative Model:

1. Testing and debugging during smaller iterations is easy.
2. A Parallel development can plan.
3. It is easily acceptable to the ever-changing needs of the project.
4. Risks are identified and resolved during iteration.
5. Limited time spent on documentation and extra time on designing.

Disadvantage(Cons) of Iterative Model:

1. It is not suitable for smaller projects.
2. More Resources may be required.
3. Design can be changed again and again because of imperfect requirements.
4. Requirement changes can cause over budget
5. Project completion date not confirmed because of changing requirements.

1.Requirements analysis and specification phase:

The aim of this phase is to understand the exact requirements of the customer and to document them properly. Both the customer and the software developer work together so as to document all the functions , performance, and interfacing requirement of the software. It describes the "what" of the system to be produced and not "how. “In this phase, a large document called Software Requirement Specification (SRS) document is created which contained a detailed description of what the system will do in the common language.

Design Phase:

This phase aims to transform the requirements gathered in the SRS into a suitable form which permits further coding in a programming language. It defines the overall software architecture together with high level and detailed design. All this work is documented as a Software Design Document (SDD).

Implementation and unit testing:

During this phase, design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly, because all the information needed by software developers is contained in the SDD. During testing, the code is thoroughly examined and modified. Small modules are tested in isolation initially. After that these modules are tested by writing some overhead code to check the interaction between these modules and the flow of intermediate output.

Integration and System Testing:

This phase is highly crucial as the quality of the end product is determined by the effectiveness of the testing carried out. The better output will lead to satisfied customers, lower maintenance costs, and accurate results. Unit testing determines the efficiency of individual modules. However, in this phase, the modules are tested for their interactions with each other and with the system.

Operation and maintenance phase:

Maintenance is the task performed by every user once the software has been delivered to the customer, installed, and operational.

3. Requirement Analysis

3.1 Data Flow Diagrams

Data flow diagram shows the flow of data from external entities into the system, and from one process to another within the system. It is a graphical representation of flow of data through a system. There are four symbols for drawing a DFD:

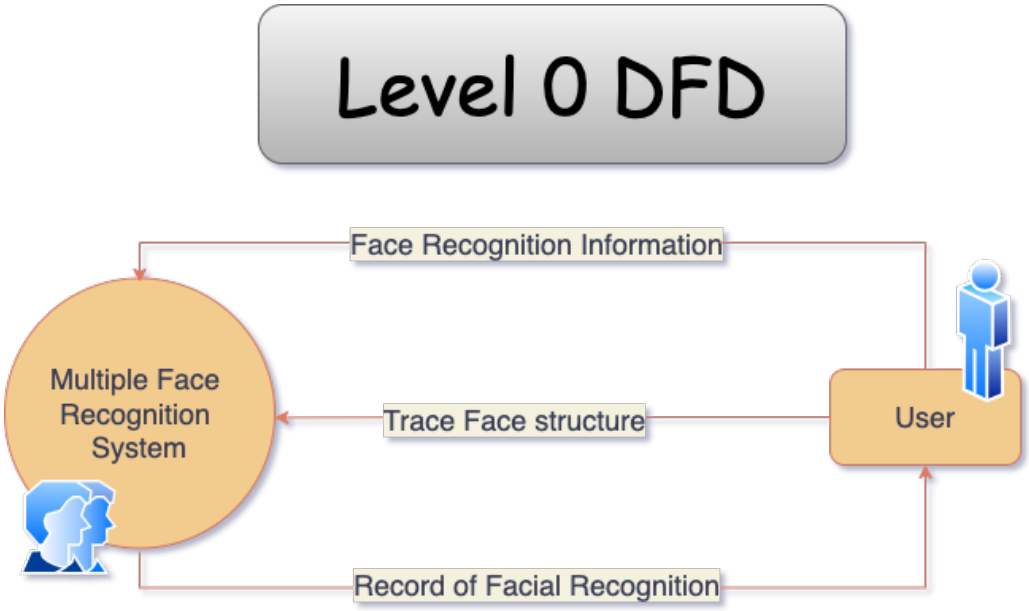
1. Rectangles representing external entities, which are sources or destination of data.

2. Ellipses representing processes, which take data input, validate and process it and output it.

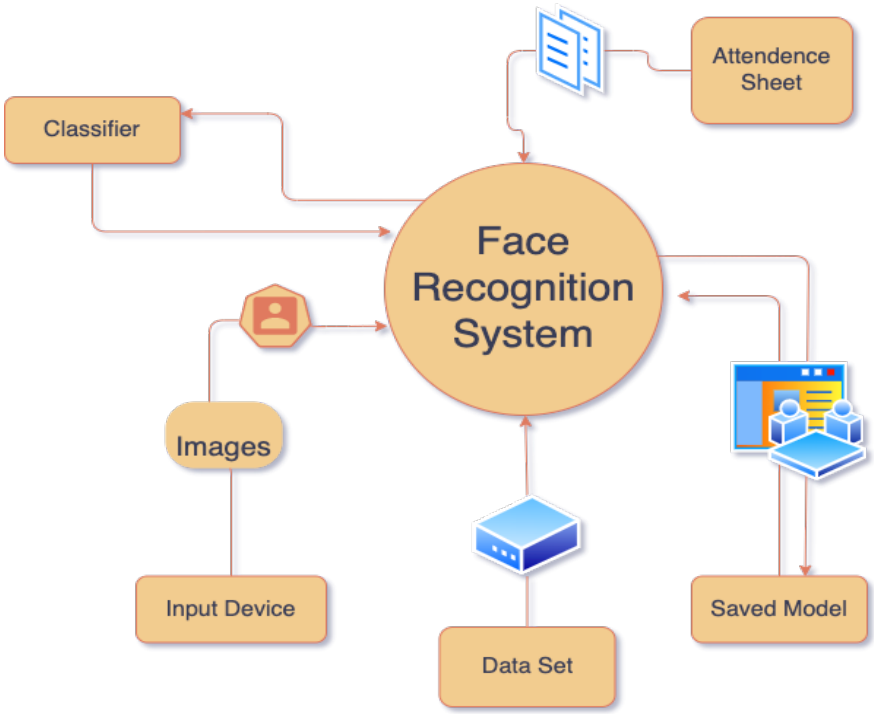
3. Arrows representing the data flows, which can either, be electronic data or physical items.

4. Open-ended rectangles representing data stores including electronic stores such as databases or XML files and physical stores such as filing cabinets or stacks of paper.

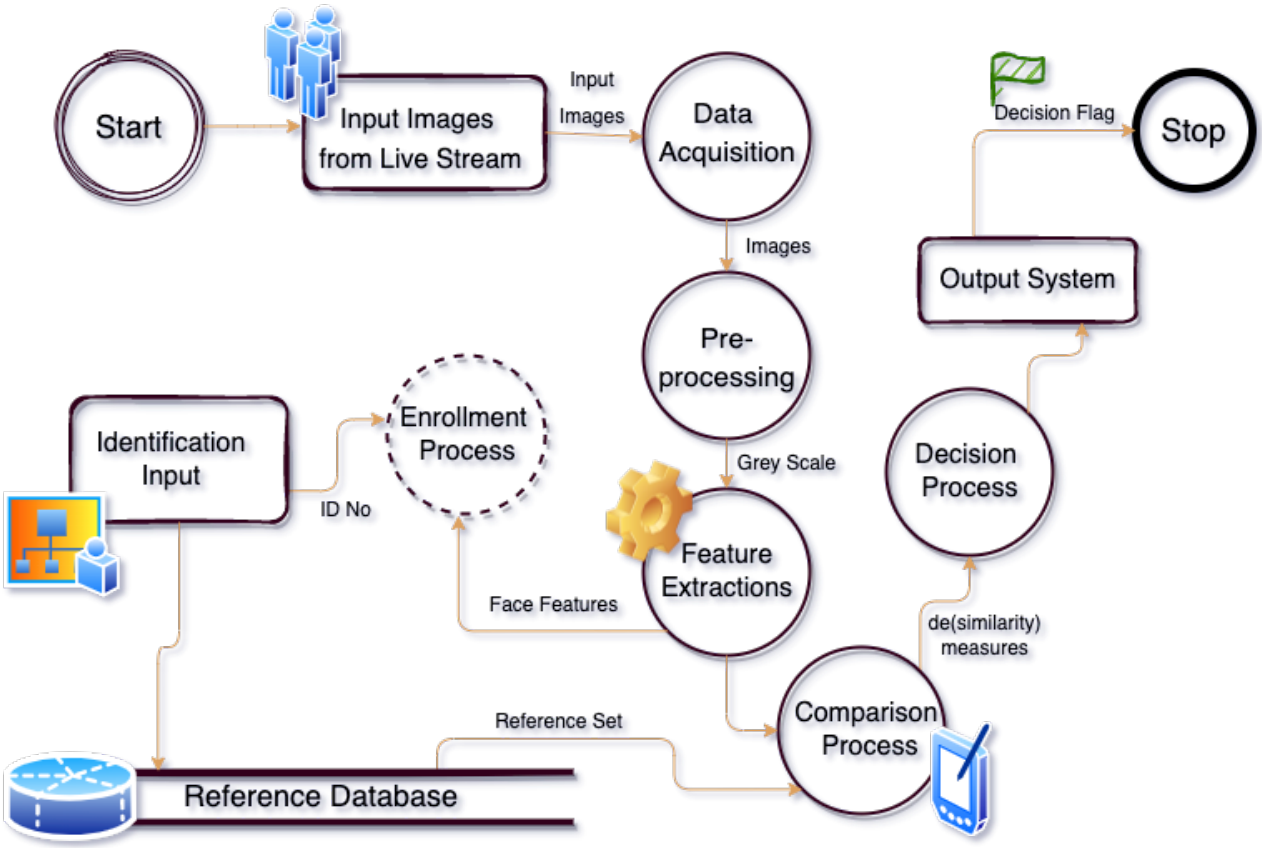
CONTEXT LEVEL DIAGRAM



Level 1 DFD



Level 2 DFD

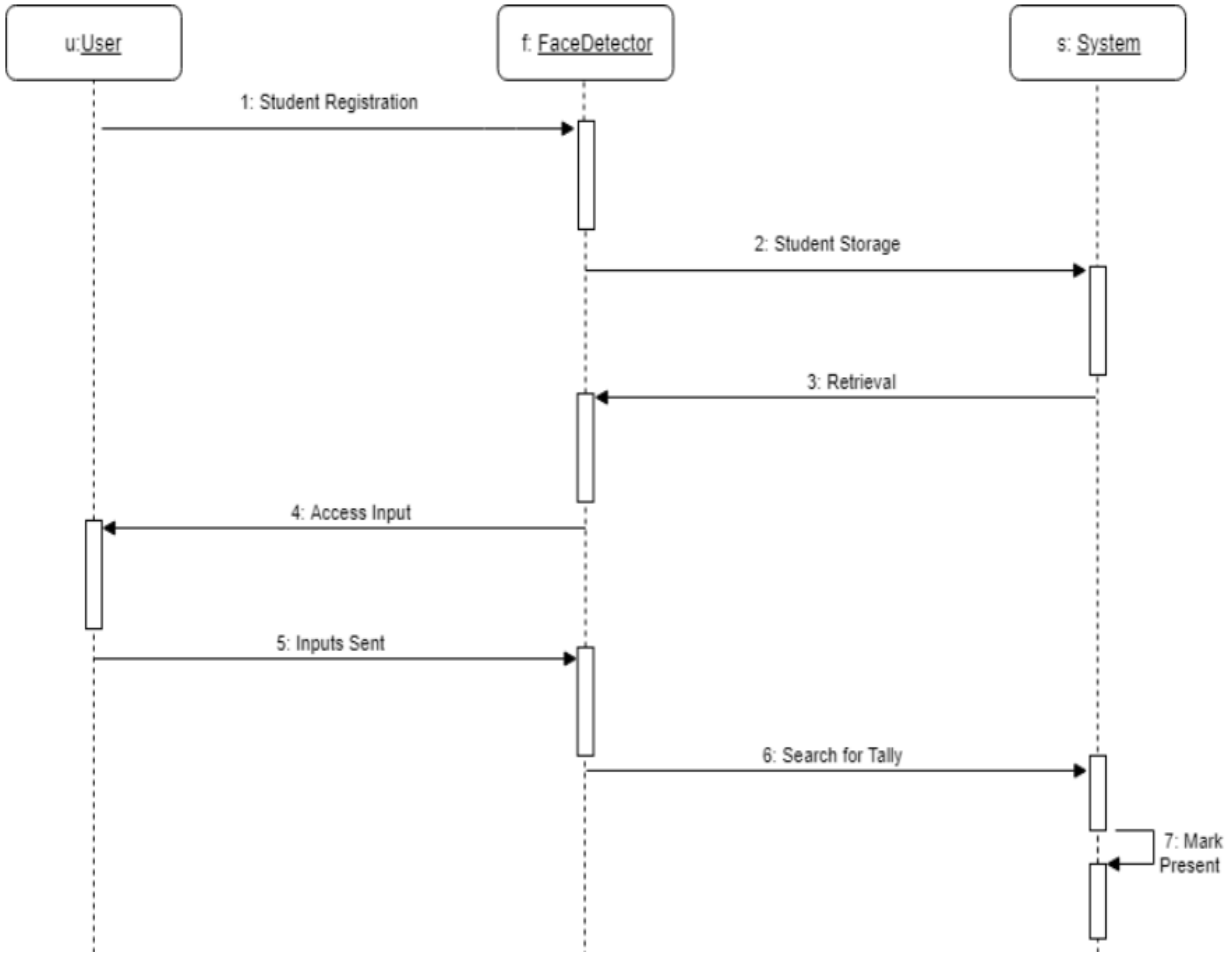


3.2 SEQUENCE DIAGRAM

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time.

Purpose of a Sequence Diagram

1. To model high-level interaction among active objects within a system.
2. To model interaction among objects inside a collaboration realizing a use case.
3. It either models generic interactions or some certain instances of interaction



Sequence Diagram

3.3 Software Requirement Specification (SRS) Document

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfil all the stakeholders' (business, users) needs. It includes the requirement definition and the requirements specification. This document aims at defining the overall software requirements for "College Enquiry Chatbot". Efforts have been made to define the requirements exhaustively and accurately. The final product will be having only features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties in developing/ testing/ implementing/ using this product. In case it is required to have some additional features, a formal change request will need to be raised and subsequently a new release of this document and/or product will be produced.

3.3.1.1 Requirements Specifications

● Operating System: Windows, Linux, MacOS

● Web-Browser - Chrome

● RAM: minimum 4 GB

Developer Requirements

● Operating System: Windows, Linux, MacOS

● Language: Python

●Libraries used : eel, opencv, numpy , os, Time, face_recognition , pickle, imutils, datetime ,multiprocessing, random, shutil, database

● IDE: Visual Studio Code

● Hardware: Webcam

Purpose

The specification document is intended to be a complete specification of what functionality the **Multiple Face Recognition Attendance System** provides. The purpose of creating this system is to . The intended audience for this system is the development team, testing team and end users of the product.

Even though there exist single face recognition systems but they are less efficient as they require each students to face the recognition system individually which ultimately take more time and therefore time costly. Solution to such issue is to automatically recognize multiple faces at a time.

Project Scope

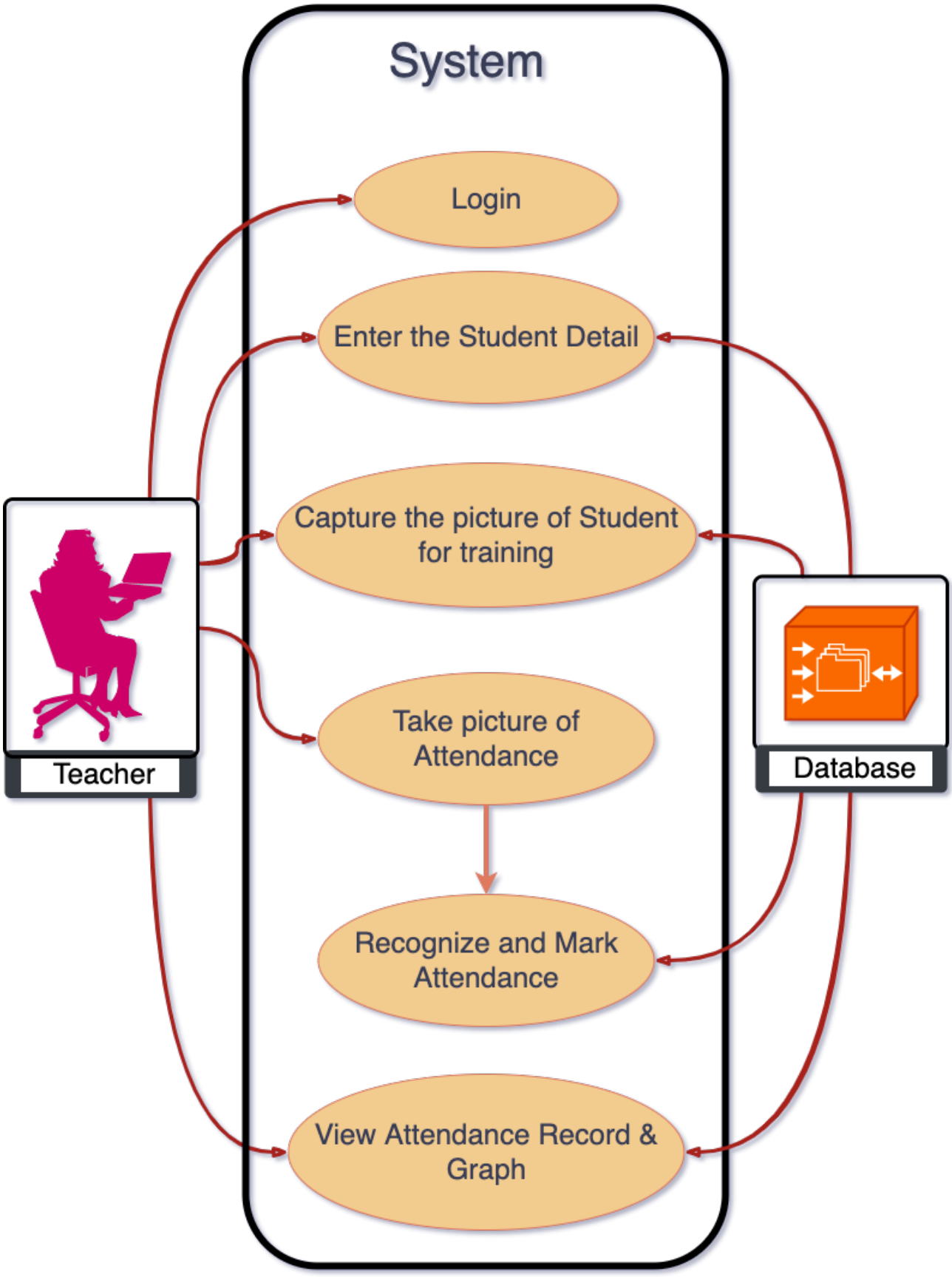
The goal of this project is to eliminate the time required to take individual attendance manually or digitally. This simplifies the cumbersome process of taking manual attendance via digital aid.

The two viable options we have are to take attendance individually or to take multiple attendances at a time. Even if the manual aspect is eliminated, the automation is partial if multiple students are not recognized in one go.

Multiple Face Recognition Model detects the faces, recognise them, validate them with present dataset and that's it. No need to maintain registers and extra 15 minutes daily, just to record the presence that's an obvious if (s)he have entered the class already.

3.4 Use Case Diagram:-

In software and systems engineering, a use case is a list of actions or event steps, typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system, to achieve a goal. The actor can be a human, an external system, or time. In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals. Another way to look at it is a use case describes a way in which a real- world actor interacts with the system. In a system use case you include high-level implementation decision.


System
Login
Enter the Student Detail
Capture the picture of Student for training
Take picture of Attendance
Recognize and Mark Attendance
View Attendance Record & Graph
Teacher
Database

3.5 Data Dictionary

A data dictionary is a collection of the names, definitions, and attributes for data elements and models. The data in a data dictionary is the metadata about the database. These elements are then used as part of a database, research project, or information system.

The data dictionary contains information about the following –

- Names of all the database tables and their schemas.
- Details about all the tables in the database, such as their owners, their security constraints, when they were created etc.
- Table constraints such as primary key attributes, foreign key information etc.
- Information about the database views

Field Name	Description	Data Type	Field Size
Student-id	Unique-id of the student (Primary Key)	Text	8
Full-Name	Name of the student	Text	20
Class	Year in which student is studying	Text	3
Session	Current session	Date	4
Date	Date of Attendance	Date	8
Time	Time of the Attendance	Time	8
Teacher-id	Unique-id of teacher (Primary Key)	Text	2
Username	Username of the teacher	Text	6
Password	Password of the account	Password	5

4. Project Management

4.1 TimeLine Chart

Tasks	Month 1				Month 2				Month 3			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Problem Statement												
Proposed Solution												
Requirement Gathering and analysis												
Software Requirement Analysis												
Schedule Estimation												
Effort Estimation												
Cost Estimation												
Risk Management												
Design												
Coding and Implementation												
Integration and Testing												
Deployment of system												
Final report generation												
Complete project execution												

Gantt Chart for working Schedule

COMPUTING FUNCTION POINTS (FP)

Function-oriented software metrics use ameasureofthe functionality delivered by the application as a normalization value. Since, "functionality" cannot be measured directly; it must be derived indirectlyusingother direct measures. Function points are computed by completing the table given below. Fiveinformationdomain characteristics are determined andcountsareprovided in the appropriate table location. Information domain values are defined inthefollowingmanner: -

Number of user inputs - The user input that provides distinct application oriented data to the software is counted. Inputs should be distinguished from inquiries, which are counted separately.

Number of user outputs - Each user output that provides application oriented information to the user is counted. In this context, output refers to compressed data file, file details, etc.

Number of user inquiries - An inquiry is defined as searching of a file which comes in the category of either frequently used file or non-frequently used file. Each distinct inquiry is counted.

Number of files - Each logical master file (i.e., a logical grouping of data that may be one part of a large database or a separate file) is counted.

Number of external interfaces - All machine readable interfaces (e.g., data files on storage media) that are used to transmit information to another system are counted.

MEASUREMENT PARAMETER	COUNT	WEIGHTING FACTOR			WEIGHTING COUNT
EI's	3	3	4	6	9
EO's	1	4	5	7	4
EQ's	2	3	4	6	6
ILF's	1	7	10	15	7
EIF's	2	5	7	10	10

TOTAL COUNT - 36

FUNCTION POINTS (FP) - Total Count * [0.65 + (0.01 * Σ (Fi))]

- The fi(i= 1 to 14) are 'Complexity Adjustment Values'; based on responses to the following questions:-

S.no.	Question	VAF(s)
1	Does the system require reliable backup and recovery?	5
2	Are specialized data communications required to transfer information to or from the application?	5
3	Are there distributed processing functions?	4
4	Is performance critical?	2
5	Will the system run in an existing, heavily utilized operational environment?	3
6	Does the system require on-line data entry?	5
7	Does the on-line data entry require the input transaction to be built overmultiple screens or operations?	4
8	Are the ILFs updated online?	4
9	Are the inputs, outputs, files, or inquiries complex?	2
10	Is the internal processing complex?	3
11	Is the code designed to be reusable?	3
12	Are conversions and installations included in the design?	3
13	Is the system designed for multiple installations in different organizations?	4
14	Is the application design to facilitate change and for ease of use by the user?	5

TOTAL COUNT - 36

FUNCTION POINTS (FP) - Total Count * [0.65 + (0.01 * Σ (Fi))]

= 36 x [0.65 + (0.01 x 52)]

= 36 x [0.65 + 0.52]

= 36 x 1.17

= 42.12

EFFORT

Barry Boehm introduced a hierarchy of software estimationmodelsbearing the name COCOMO, for COnstructiveCOstMOdel. TheoriginalCOCOMO model became one of the most widely usedanddiscussedsoftware cost estimation models in the industry. The COCOMOII application composition model uses object points. Once complexity is determined, the number of screens, reports, andcomponents are weighted. The object point count is thendeterminedby multiplying the original

number of object instances bytheweightingfactor in and summing to obtain a total object point count. Whencomponent-based development or general software reuseistobeapplied, the percent of reuse (%reuse) is estimated andtheobjectpoint count is adjusted:

NOP = (object points) x [(100 -%reuse)/100]

Where, NOP is defined as new object points.

Data used in estimating effort are-

1) Object points = 36 (taken from table. 3.2(a))

2) Estimated reuse is 35%

3) Prod is 13 (average taken)

NOP = Object Points * [(100-reuse%)/100]

= 36 * 65/100 =23.4

Estimated effort = NOP/PROD

= 23.4/13

= 1.8 ~ 2 person-months

Hence estimated effort of the project is 2 person-months

COST

To complete this project in 4 months it took 1 person. Let's say labour rate is 10,000 Rs per person per month.

Total Efforts = efforts * no. of persons * months

= 2*1*4

= 8 person-months

Cost to be paid to 1 person for 4 months = 8* 10000

= Rs.80000

Therefore, Project costing = 80,000 Rs.

4.5 Risk Management

4.5.1 Introduction

Risk Management is the process of identifying, analyzing and responding to risk factors throughout the life of a project and in the best interests of its objectives. Proper risk management will reduce not only the likelihood of an event occurring, but also the magnitude of its impact. Risk Management Systems are designed to do more than just identify the risk. The system must also quantify the risk and predict the impact on the project. The outcome is therefore a risk that is either acceptable or unacceptable. Risk Management should answer these questions

- What can go wrong?
- To what extent can something go wrong?
- What will be the plan when something goes wrong?

4.5.2 Risk Management Purpose

The purpose of risk management is to identify potential problems before they occur so that risk-handling activities may be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives. Risk management is a continuous, forward-looking process that is an important part of business and technical management processes. Risk management should address issues that could endanger achievement of critical objectives. A continuous risk management approach is applied to effectively anticipate and mitigate the risks that have critical impact on the project.

Effective risk management includes early and aggressive risk identification through the collaboration and involvement of relevant stakeholders. Strong leadership across all relevant stakeholders is needed to establish an environment for the free and open disclosure and discussion of risk. Although technical issues are a primary concern both early on and throughout all project phases, risk management must consider both internal and external sources for cost, schedule, and technical risk. Early and aggressive detection of risk is important because it is typically easier, less costly, and less disruptive to make changes and correct work efforts during the earlier, rather than the later, phases of the project.

RISK TABLE

RISK	CATEGORY	IMPACT	PROBABILITY	RMMM
Module crash	Technical Risk	1	10%	The development team should cease work on that system until the environment is made stable again, or should move to a system that is stable and continue working there.
Web design not working in small devices	Customer Characteristics	3	20%	Adding extensive CSS code will reduce such chances.
Hardware not connected (webcam)	Technical Risk	2	5%	Click multiple of students in a group of 10-15, then upload the images for attendance.
Link between frontend and database being disrupted.	Technical Risk	2	10%	Keeping a track of data links will prevent this from happening.
Face recognized incorrectly	Operational Risk	2	5%	Adding more data to the image database to provide more precise solutions.
Data privacy Leaked	Technical Risk	2	5%	Storage of dataset in hash form and using more secure system.

Impact Values	Category
1-Catastrophic	CU: Customer Characteristics
2-Critical	OP-OPERATIONAL RISK
3-Marginal	TE-TECHNICAL RISK
4-Negligible	

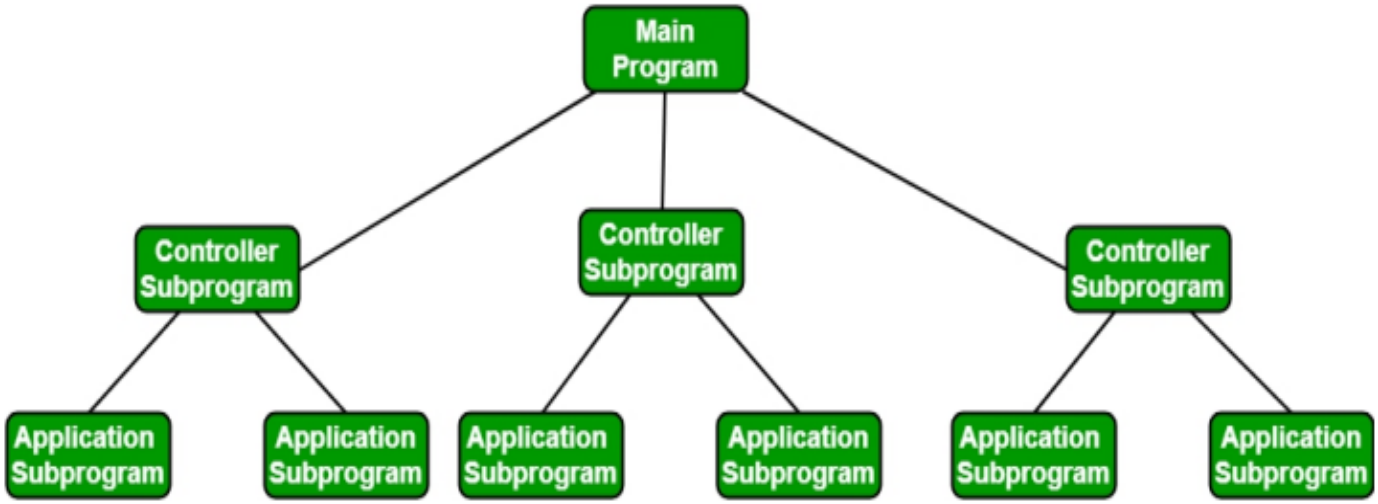
5. Design Engineering

The design phase of software development deals with transforming the customer requirements as described in the SRS documents into a form implementable using a programming language. The software design process can be divided into the following three levels of phases of design:

1. Interface Design

2. Architectural Design

3. Detailed Design



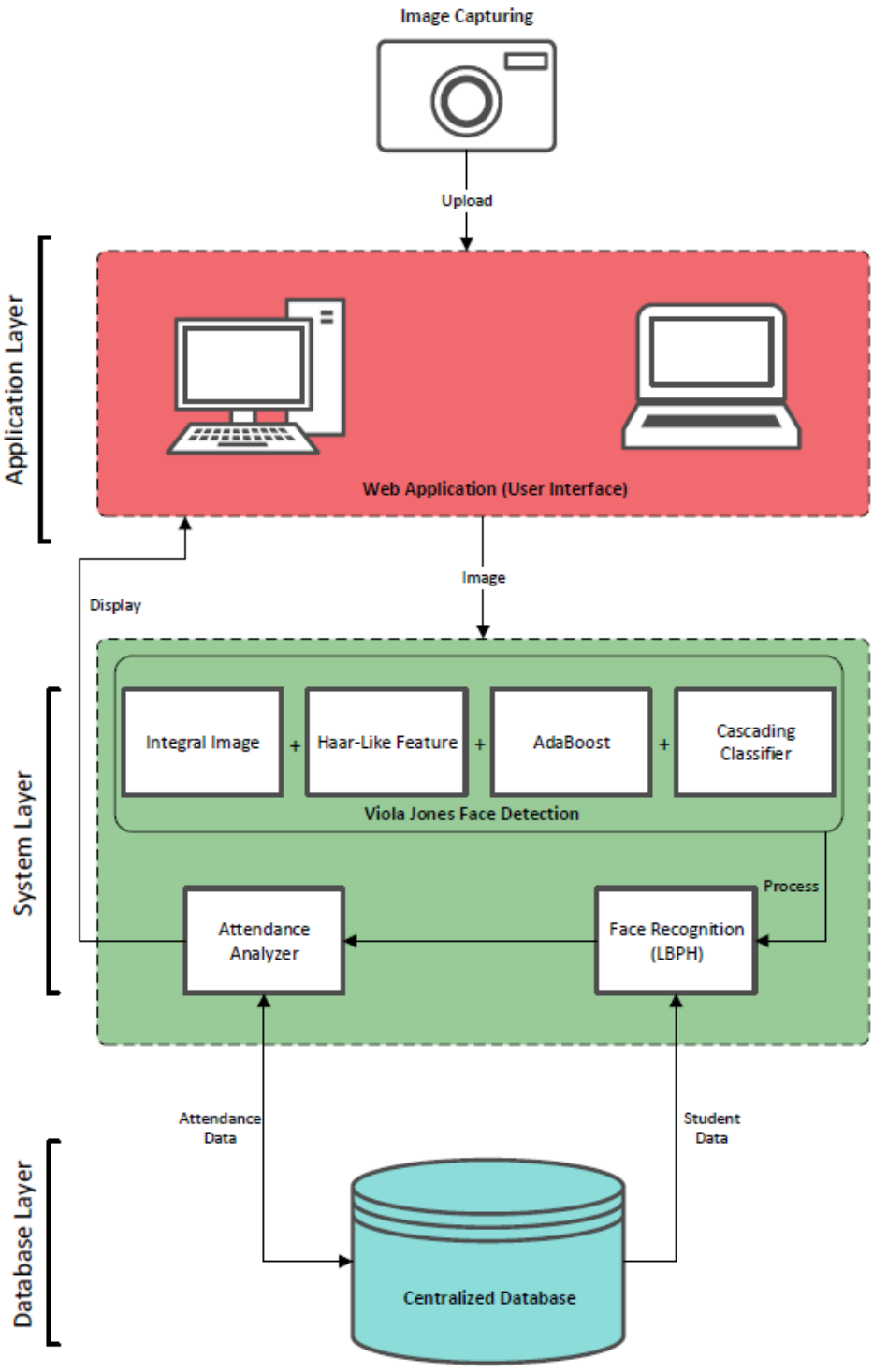
Design Engineering

5.1 Architectural Design

Architectural design is the specification of the major components of a system, their responsibilities, properties, interfaces, and the relationships and interactions between them. In architectural design, the overall structure of the system is chosen, but the internal details of major components are ignored. The architectural design adds important details ignored during the interface design. Design of the internals of the major components is ignored until the last phase of the design.

Architectural styles define a group of interlinked systems that share structural and semantic properties. In short, the objective of using architectural styles is to establish a structure for all the components present in a system. If an existing architecture is to be re-engineered, then imposition of an architectural

style results in fundamental changes in the structure of the system. This change also includes re-assignment of the functionality performed by the components.



Architectural Design

1 Application Layer:-

This is the capturing phase where the system is used to capture frames using a web application and store files on the database. This layer is secure, where authentication is provided to prevent unauthorised access to the system. Web application would be used to capture images and view the attendance reports.

2 **System Layer:-**

System layer is where detection and recognition modules are done. Viola-Jones algorithm is used to detect faces from the frames. First, integral image is generated, which assigns numbers to pixels by finding the total of the values. Furthermore, haar-like feature is generated to detect objects from the frames. Images which are extracted are sent to a classifier that is used to detect faces from the objects. LBHP algorithm is used for recognizing the images.

3 Database Layer:-

Centralized system is the database layer where the students' records and attendance are stored. During attendance, student's captured image is compared to one stored on database. Once recognition is completed successfully, attendance is recorded and updated in the database. Attendance report is available on the web application at any time.

5.2 Pseudocode Explanation For Code

Module – Face Recognition

we will be using various python libraries and modules for face recognition, face identification, saving a users image and other information also .We use *OPEN-CV(Open Source Computer Vision)* library for face recognition, identification, we use *os* package to store student information in local database, *Numpy* is used to perform the appropriate task, **Pickle** module is used to serializing and de-serializing a Python object structure, *eel* helps us to make *GUI* for better interaction with the program. In this project, we use *.db (SQL) as a* database to store the students attendance.

Psuedocode -

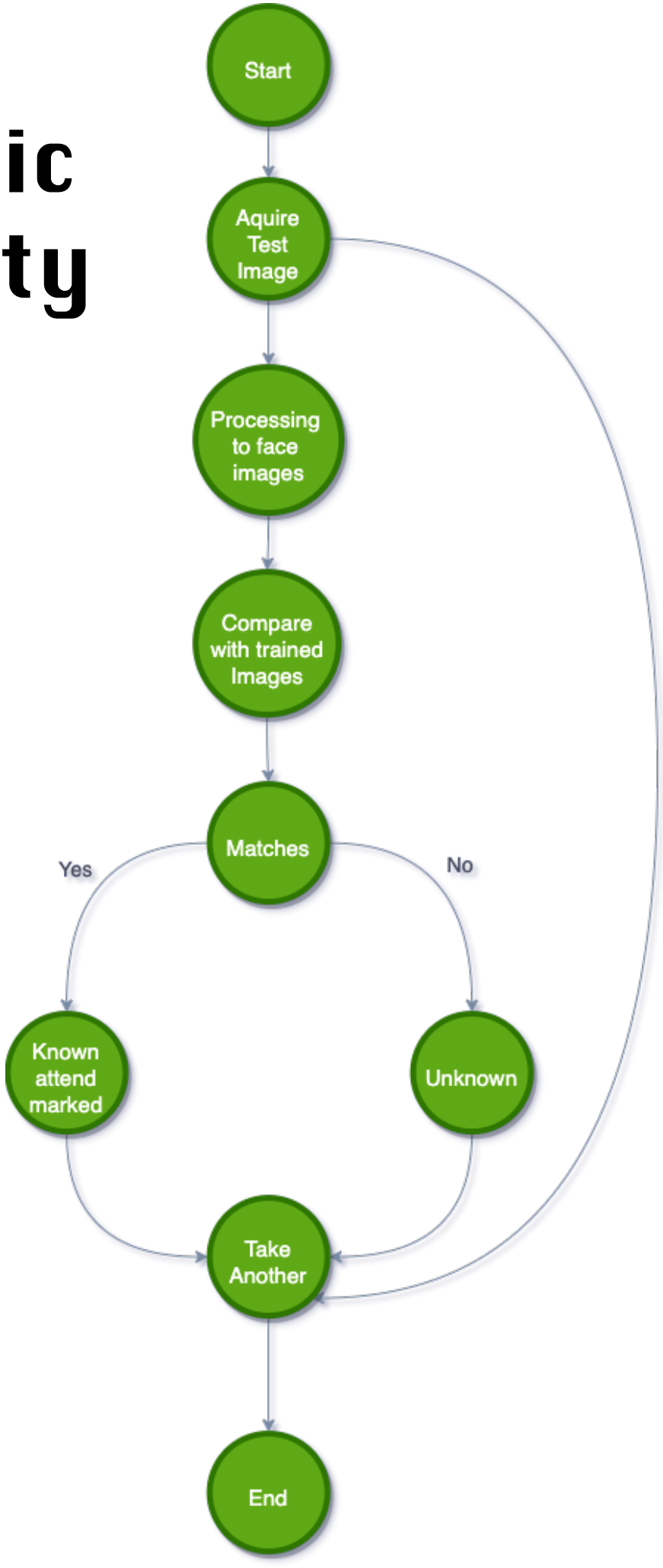
```
while(True):
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = detector.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        cv2.imwrite("TrainingImage\ "+name.lower() +"."+Id +'.'+ str(sampleNum) + ".jpg",
gray[y:y+h,x:x+w])
        cv2.imshow('frame',img)
        if cv2.waitKey(100) & 0xFF == ord('q'):
            break
        elif sampleNum>60:
            break
    cam.release()
    cv2.destroyAllWindows()
```

Cyclomatic Complexity

- M - Cyclomatic Complexity
- E - No. of Edges
- N - No. of Nodes
- R - No.of regions
- P - Predicate nodes

Calculation (M):

- E - N + 2 -> 10-9+2 = 3
- R = 3
- P+1 = 3



5.3 UI DESIGN

The following pages demonstrate some of the key functionalities of application .

FIGURE:- 5. 1 LOGIN PAGE

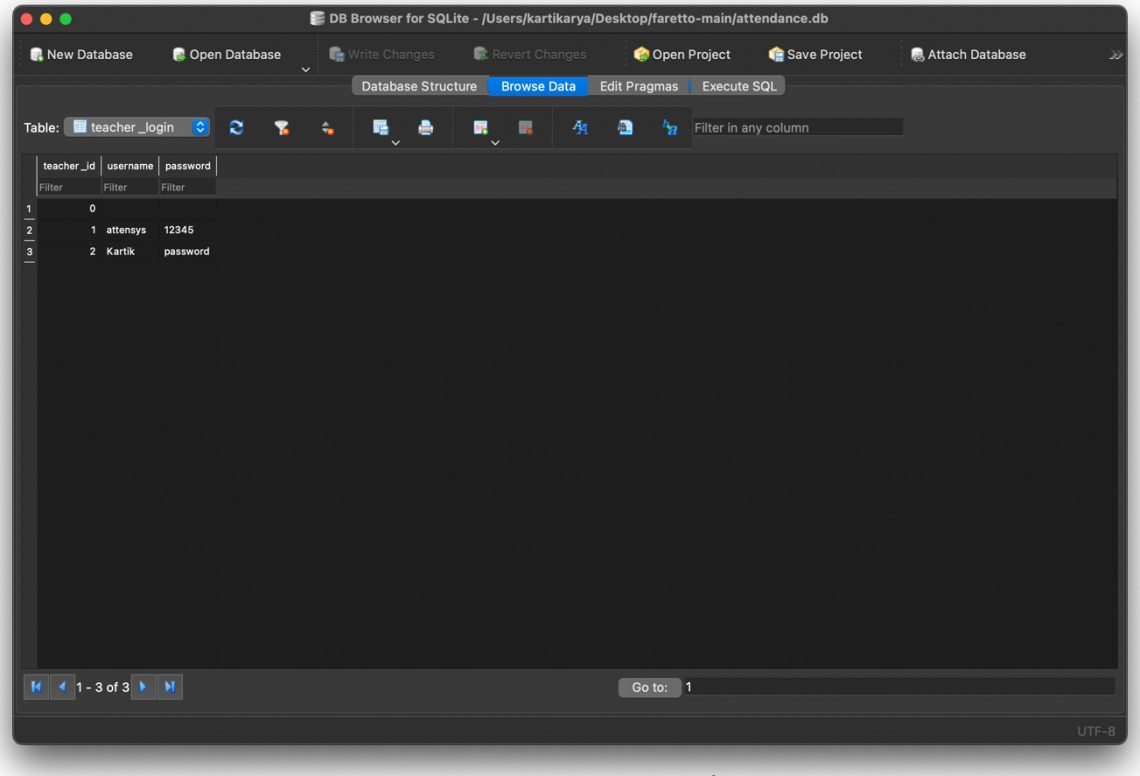


Figure:- 5. 2 Login Details

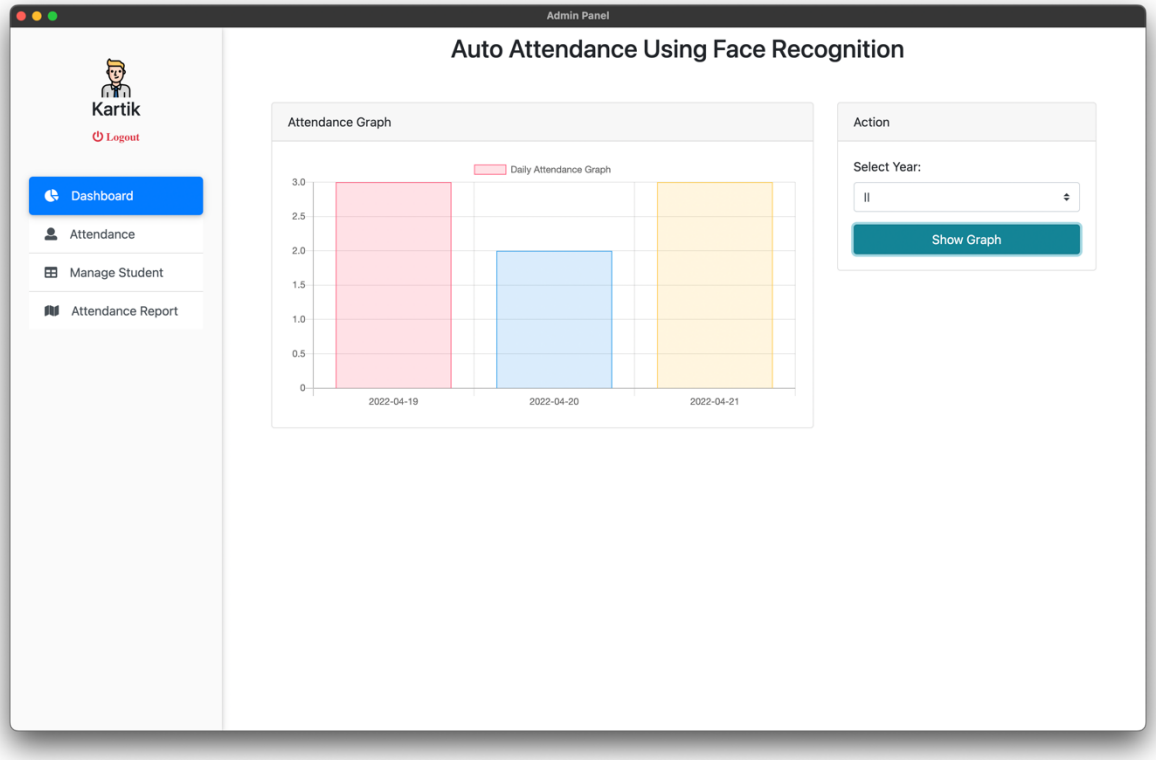


FIGURE 5.3 DASHBOARD

Fig 5.3 illustrates Application GUI developed with HTML, JavaScript and CSS. This would be used to interact with the system. It has 6 buttons to leverage features of Face Recognition Attendance

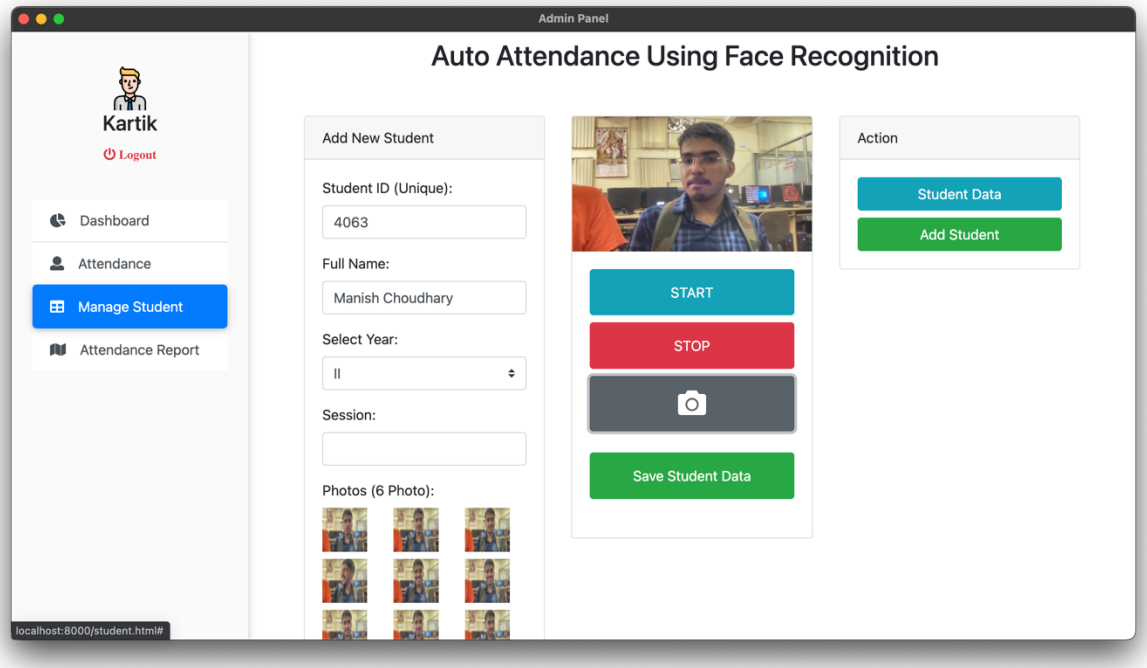


Figure:- 5. 4 Adding New Student

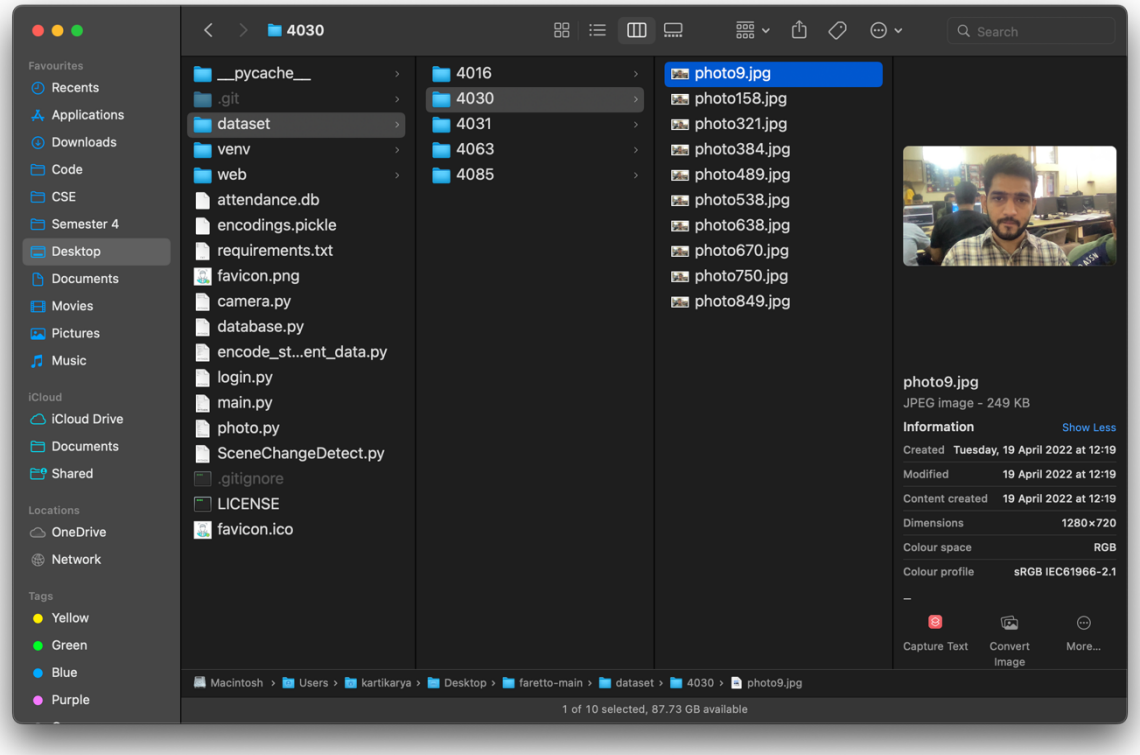


Figure :- 5.5 Dataset Folder of images

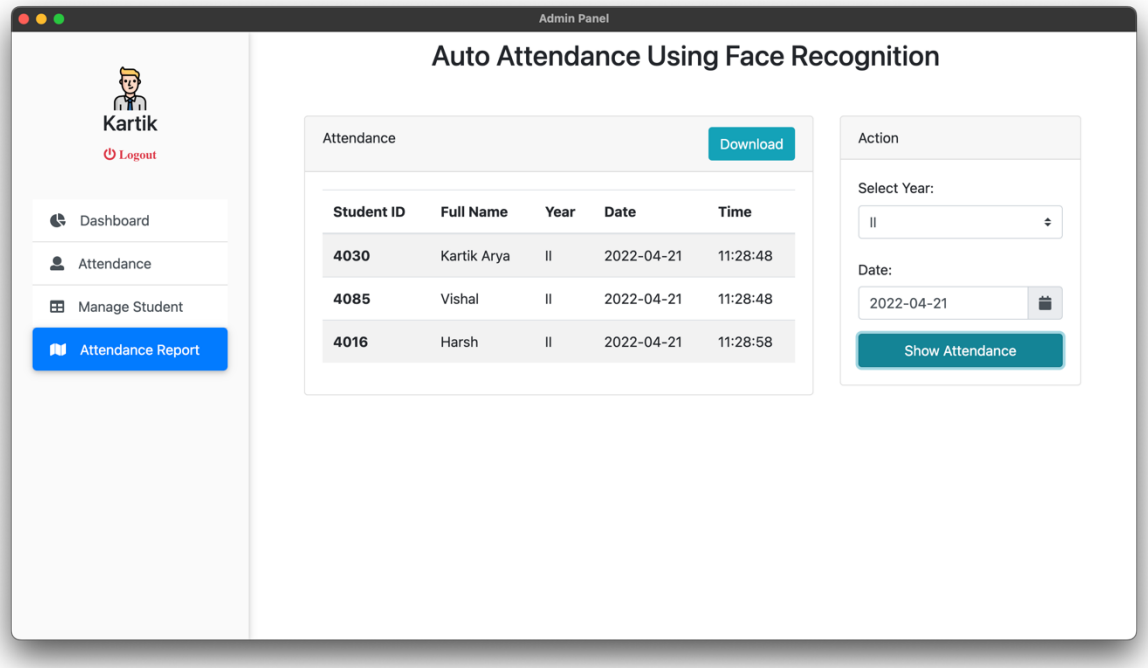


Figure:- 5. 6 Attendance Report

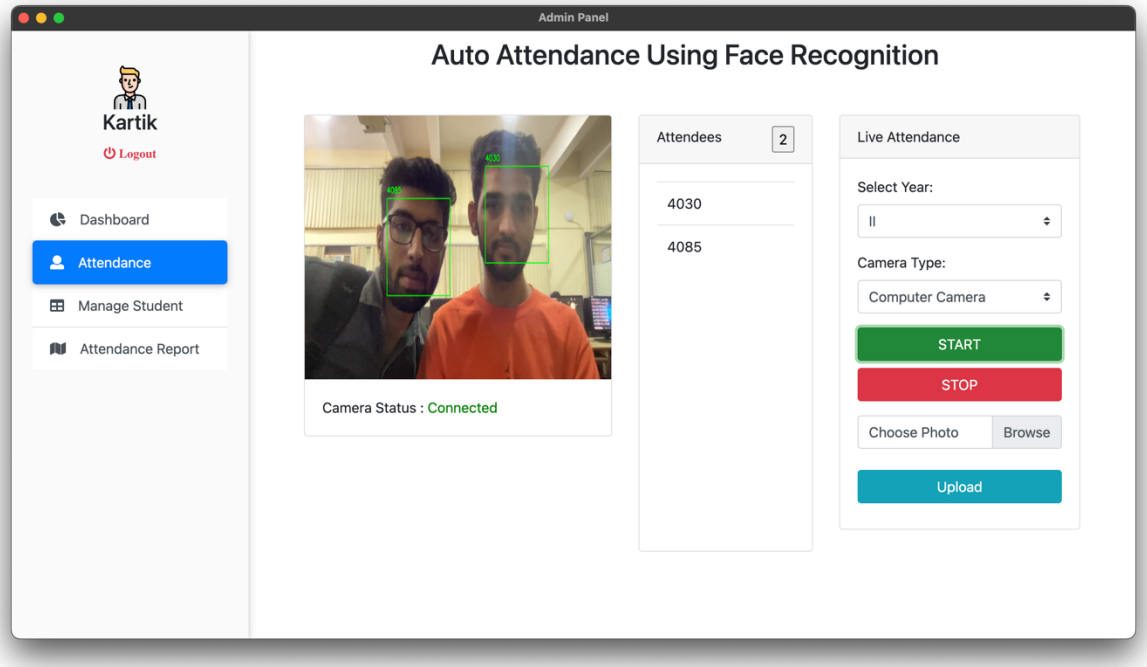


Figure:-5. 7 Taking Attendance

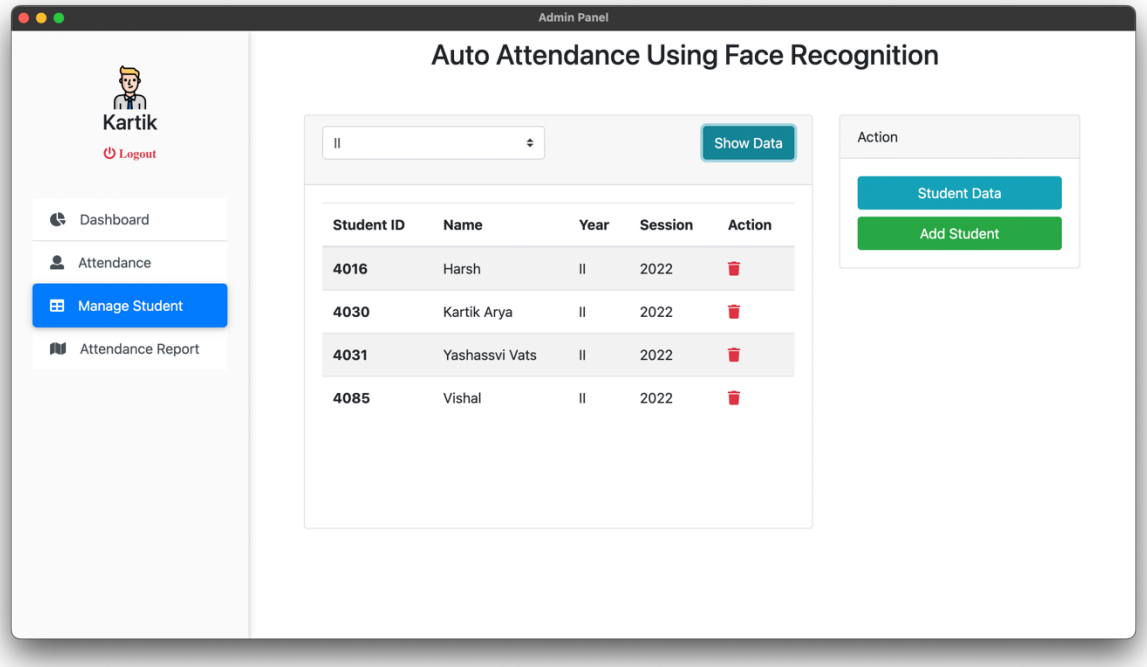


Figure:- 5.8 Showing the record of student

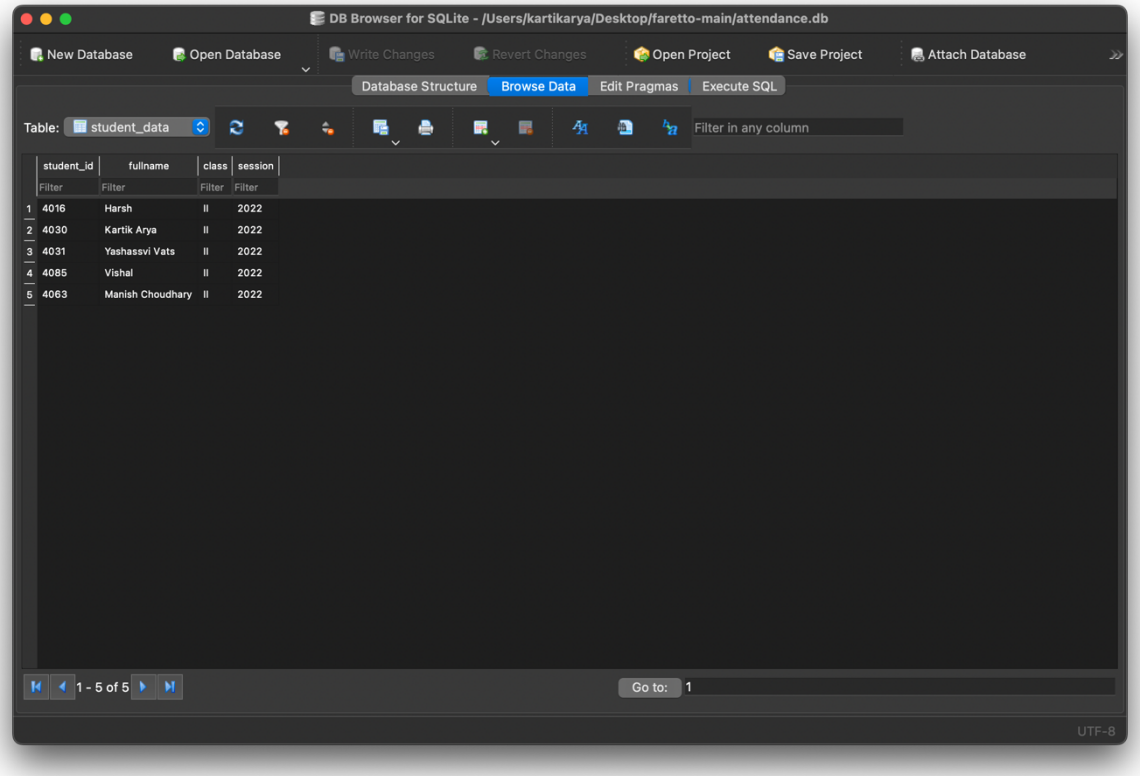


Figure:- 5.9 Record of student stored in Database

6. Algorithm

The currently available Face Recognizer Algorithms in OPEN-CV are:

- EigenFaces
- FisherFaces
- Local Binary Patterns Histograms

For our purpose, we would be using the last algorithm (Local Binary Patterns Histogram)

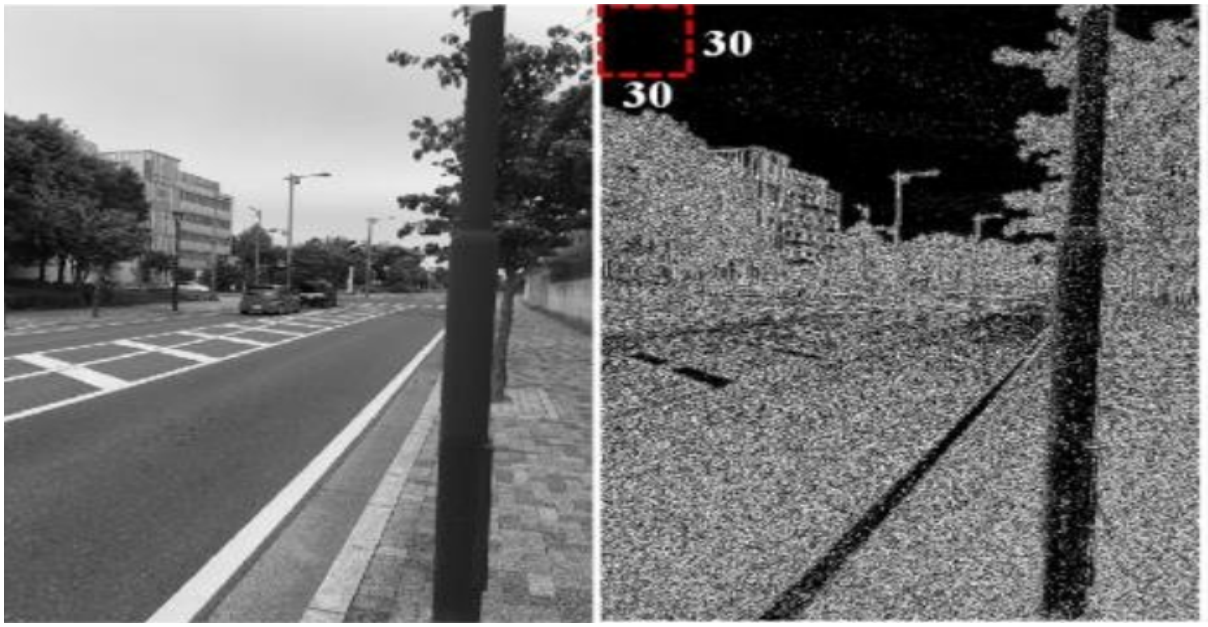
Local Binary Pattern Histogram

EigenFaces and FisherFaces take a somewhat holistic approach to face-recognition. You treat your data as a vector somewhere in a high-dimensional image space. We all know high-dimension is bad, so a lower-dimensional subspace is identified, where (probably) useful information is preserved. The EigenFaces approach maximizes the total scatter, which can lead to problems if the variance is generated by an external source, because components with a maximum variance over all classes aren't necessarily useful for classification. So to preserve some discriminative information we applied a Linear Discriminant Analysis and optimized as described in the FisherFaces method. The FisherFaces method worked great at least for the constrained scenario we've assumed in our model.

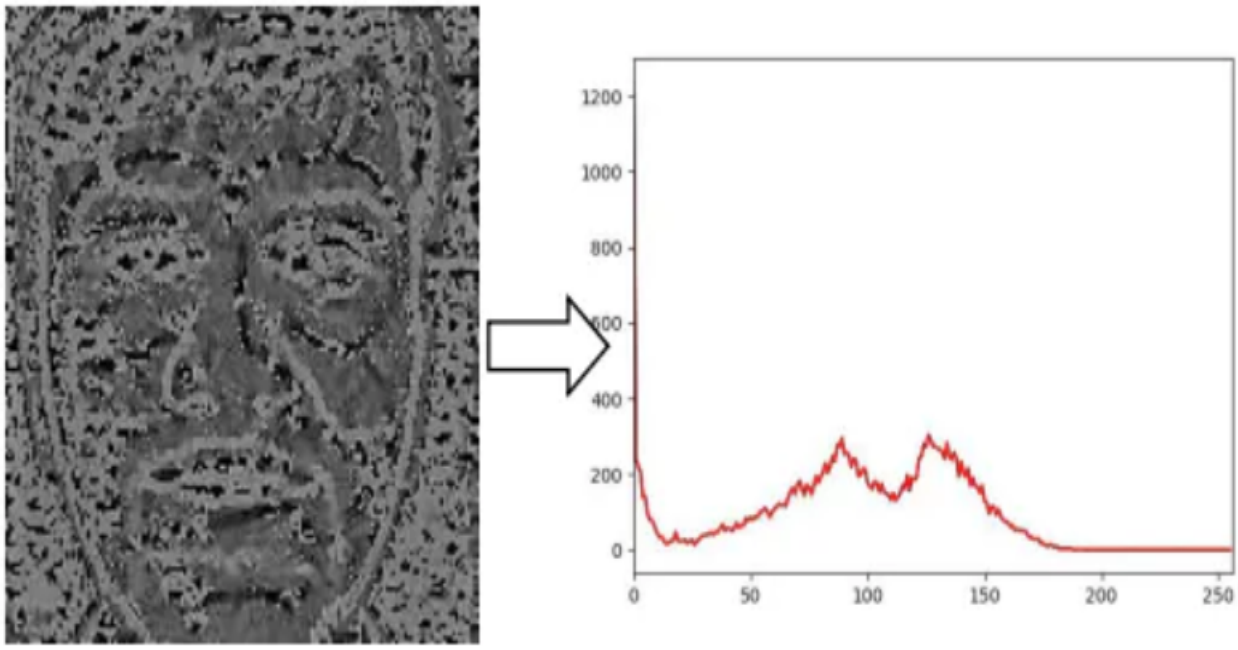
Now real life isn't perfect. You simply can't guarantee perfect light settings in your images or 10 different images of a person. So what if there's only one image for each person? Our co-variance estimates for the subspace may be horribly wrong, so will the recognition.So some research concentrated on extracting local features from images. The idea isto not look at the whole image as a high-dimensional vector, but describe only local features of an object. The features you extract this way will have a low-dimension implicitly. A fine idea! But you'll soon observe the image representation we are given doesn't only suffer from illumination variations. Think of things like scale, translation or rotation in images - your local description has to be at least a bit robust against those things. The Local Binary Patterns methodology has its roots in 2D texture analysis. The basic idea of Local Binary Patterns is to summarize the local structure in an image by comparing each pixel with its neighborhood. Take a pixel as center and threshold its neighbors against. If the intensity of the center pixel is greater-equal its neighbor, then denote it with 1 and 0 if not. You'll end up with a binary number for each pixel, just like 11001111. So with 8 surrounding pixels you'll end up with 2^8 possible combinations, called ***Local Binary Patterns*** or sometimes referred to as ***LBPcodes***. The first LBP operator described in literature actually used a fixed 3 x 3neighborhood just like this:

Matrix-representation of image

By definition the LBP operator is robust against monotonic gray scale transformations.We can easily verify this by looking at the LBP image of an artificially modified image (so you see what an LBP image looks like):



So what's left to do is how to incorporate the spatial information in the face recognition model. The representation proposed by Ahonenet. Al is to divide the LBP image into ***m*** local regions and extract a histogram from each. The spatially enhanced feature vector is then obtained by concatenating the local histograms (*not mergingthem*). These histograms are called ***Local Binary Patterns Histograms***.



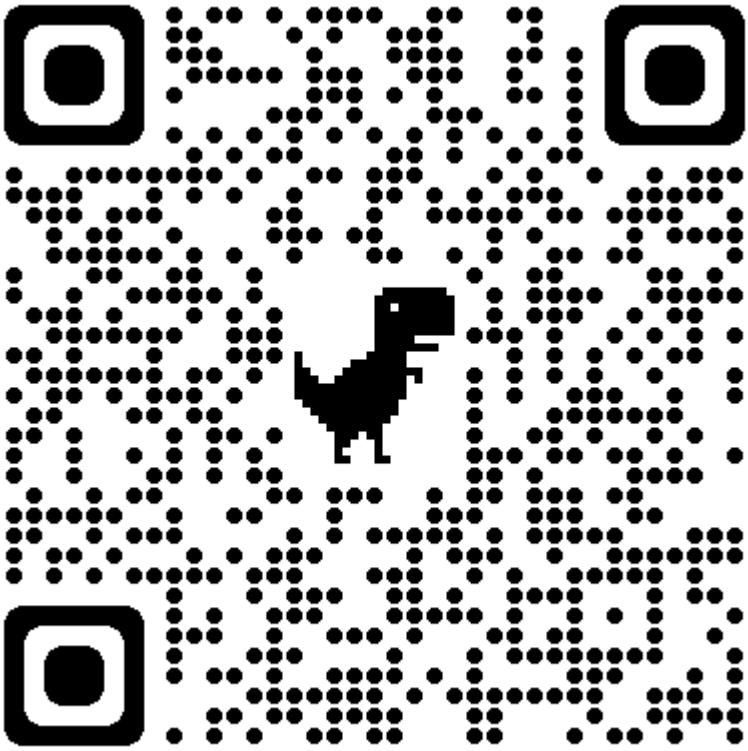
LBP image to Histogramic Graph

7. Coding

The Git-hub link & QR –code of the code is given below :-

https://github.com/ripper-18/ripper-18---Software-Engineering-Project-Private.git

OR Scan the Below QR Code



8. Testing

Testing Approach

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics. Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

Aim of Software Testing

Software is tested to uncover all those errors that were made inadvertently as per designed and constructed.

✓ Testing often accounts for more project effort than any other project action. If It is conducted in random way, whole time is wasted, unnecessary effort is expended, and even worse. It would therefore seem reasonable to establish a strategy for testing software.

✓ To perform effective testing, we should conduct effective technical reviews. Doing this will remove many errors before testing commences

✓ Testing begins at the component level and works “outward” toward the integration of the entire computer system.

✓ Testing is conducted by the developer of the software and an independent test group.

✓ Testing and debugging are different activities, but in any testing strategy debugging must be accommodated.

✓ Various testing techniques are appropriate for different software engineering approaches and at different point of time.

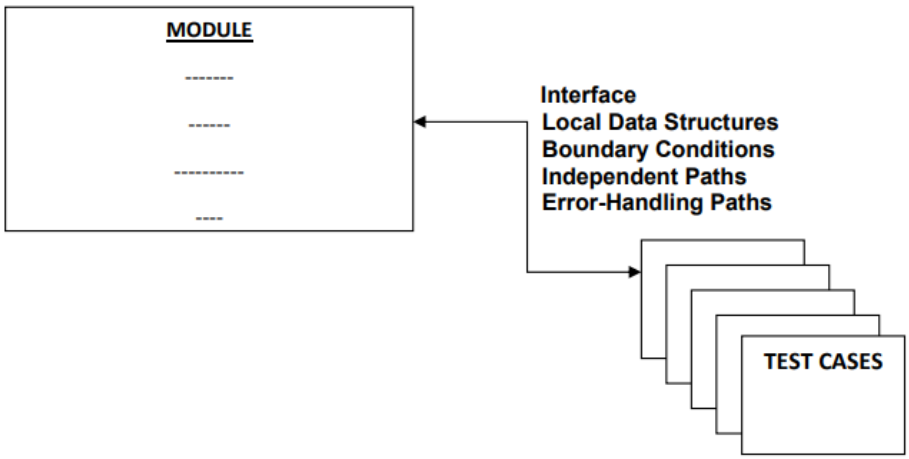
✓ A strategy should provide guidance for the practitioner and a set of milestones for the manager. Because the steps of the test strategy occur at a time when pressure of deadline begins to rise.

✓ Therefore, Progress must be measurable and problems should surface as early as possible

Unit Testing

❖ Unit test basically focuses verification effort on the smallest unit of software design – The software component or module.

❖ The relative complexity of tests and the errors those tests uncover is limited by the constrained scope established for unit testing.



❖ Generally, focuses on the internal processing logic and data structures within the boundaries of a component.

❖ This testing can be conducted in parallel for multiple components.

❖ Unit Testing is normally considered as an adjunct to the coding step.

❖ The design of unit test can occur before coding begins or after source code has been generated.

Integration Testing

❖ Integration testing is a systematic technique for constructing the software architecture while at same time conducting tests to uncover errors associated with interfacing.

❖ Basic objective of this test is to take unit-tested components and build a program structure that has been dictated by design.

❖ Incremental integration is the antithesis of the big-bang approach.

❖ The program is constructed and tested in small increments, where errors are easier to correct.

❖ There are basically two types of integration testing:

✓ Top down integration

✓ Bottom up integration

▪ TOP DOWN INTEGRATION:

❖ Top Down integration is basically an incremental approach to the construction of the software architecture.

❖ Here modules are integrated by moving downward through the control hierarchy, beginning with the main control module.

▪ BOTTOM UP INTEGRATION:

❖ Bottom Up Integration begins construction and testing with atomic modules.

❖ Since, Components are integrated from the bottom up, the functionality provided by components subordinate to a given level is always available and the need for stubs is eliminated.

❖ This can be implemented with the following steps:

1. Low-level components are combined into clusters that perform a specific software sub function.

2. A driver is written to coordinate test-case input and output.

3. The cluster is tested.

4. Drivers are removed and clusters are combined moving upward in the program structure

Functional Testing

❖ It is a type of software testing that validates the software system against the functional requirements/specifications.

❖ The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the functional requirements.

❖ It mainly involves black box testing and is not concerned about the source code of application.

❖ This testing checks User Interface, database, security, Client/Server communication and other functionality of application under test.

❖ This kind of testing can be done either manually or using automation.

Here are some of the features of functional testing:

▪ It understands the functional requirements.

▪ It identifies test input or test data based on requirements.

▪ Also computes the expected outcomes with selected test input values.

▪ It executes test cases and also compares actual and computed expected results.

No.	Resource Type	Details
1	OS Name	Windows &MacOS
2	RAM	8 GB
3	Storage	512 GB
4	Webcam	Pre-built camera in laptop
5	Source Code Editor	Visual Studio Code
6	Browser	Google Chrome

TABLE 8.1 TEST ENVIRONMENT

Test Cases

ID	TC01
Test Name	Checking the Login credential of the system
Pre-Requisite	Backend of the system is running
Test Action	Run the system Enter the correct username & password Click on login
Expected Result	Dashboard of the system is opened successfully
Status	Pass

TABLE 8.2 TEST CASE 01

ID	TC02
Test Name	Web cam integration and configuration
Pre-Requisite	Web cam is connected and integrated with the system
Test Action	1.Run the system 2. Select the year 3.Click on start to begin camera
Expected Result	System automatically turns on web cam in web browser
Status	Pass

TABLE 8.3 TEST CASE 02

ID	TC03
Test Name	Image capturing and storing in dataset
Pre-Requisite	Folder name dataset is present
Test Action	Click on Manage Student Add Student Fill the details of student Click on the start button
Expected Result	By clicking the start button, camera opens which starts capturing the image and save the image in student-id folder within the dataset folder. The details of student is stored in .db(database) file.
Status	Pass

TABLE 8.4 TEST CASE 03

ID	TC04
Test Name	Face recognition and storing in the database file
Pre-Requisite	Details and image of students are registered.
Test Action	Click on the Attendance Select the details Click on Start After completing the Attendance click on stop
Expected Result	Camera starts detecting faces and compares with stored dataset. Recognized faces are marked present and their names are recorded with date and time in .db(database) file .
Status	Pass

TABLE 8.5 TEST CASE 04

ID	TC05
Test Name	Checking the report & graph
Pre-Requisite	Attendances are marked
Test Action	Run the system Select the year on dashboard page Click on show graph Click Attendance report Select year & date Click on show Attendance
Expected Result	Graph & Attendance are displayed successfully
Status	Pass

TABLE 8.6 TEST CASE 05

ID	TC06
Test Name	Face detection and recognition in unsuitable environments
Pre-Requisite	Face is detected in real-time, recognized and attendance is marked
Test Action	Run the system Click on Attendance Select year Click on start
Expected Result	Camera starts detecting faces and compares with stored dataset. Recognized faces are marked present and their names are recorded in .db(database) file.
Status	Fail
Reason	Faces fail to detect due to bad lighting conditions

TABLE 8.7 TEST CASE 06

ID	TC07
Test Name	Login with wrong Username & password
Pre-Requisite	Backend of system is running
Test Action	Run the system Enter the wrong username & password Click on login
Expected Result	Dashboard of the system is opened successfully
Status	Fail
Reason	Invalid username & password which doesn't match with username & password stored in .db file

TABLE 8.8 TEST CASE 07

ID	TC08
Test Name	Face recognition with wrong details
Pre-Requisite	Details and image of students are registered.
Test Action	Click on the Attendance Select the wrong details Click on Start After completing the Attendance click on stop
Expected Result	Camera starts detecting faces and compares with stored dataset. Recognized faces are marked present and their names are recorded with date and time in .db(database) file .
Status	Fail
Reason	Faces of students are recognised but attendance isn't marked due to wrong details.

TABLE 8.9 TEST CASE 08

ID	TC09
Test Name	Dashboard links
Pre-Requisite	User login
Test Action	Run the system View Dashboard
Expected Result	All links work and dashboard is fully functional
Status	Pass

TABLE 8.10 TEST CASE 9

ID	TC10
Test Name	Face recognition with unregistered student
Pre-Requisite	Details and image of students are registered.
Test Action	Click on the Attendance Select the details Click on Start After completing the Attendance click on stop
Expected Result	Camera starts detecting faces and compares with stored dataset. Recognized faces are marked present and their names are recorded with date and time in .db(database) file .
Status	Fail
Reason	Face of student isn't recognised because details of student isn't registered .

TABLE 8.11 TEST CASE 10

Testing Result (Output):-

These are the results of tests we conducted in real-time. Below are all the screenshots of different functions.

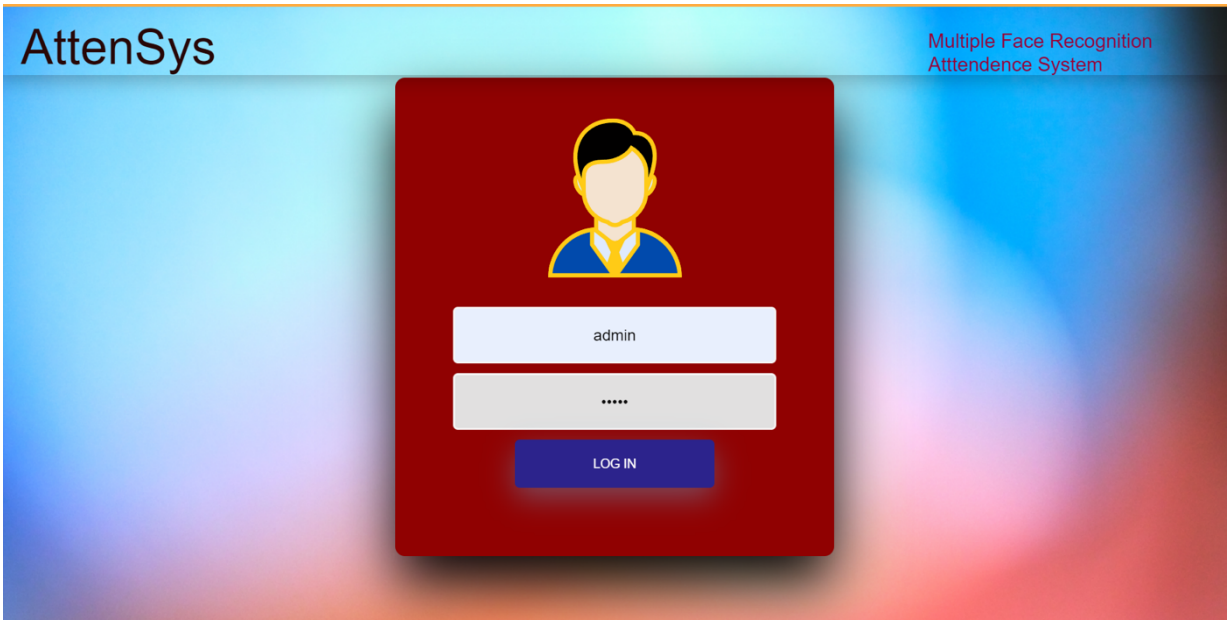


FIGURE 8.1 SUCCESSFULLY LOGIN OF THE SYSTEM

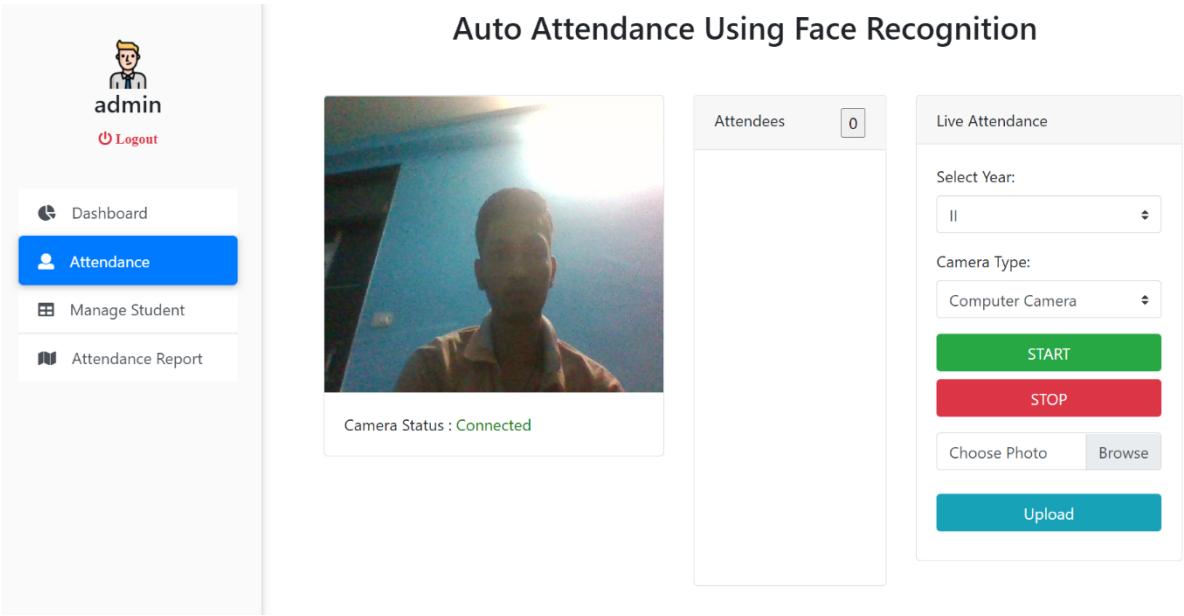


FIGURE 8.2 *SUCCESSFULLY TURN ON WEB-CAM*

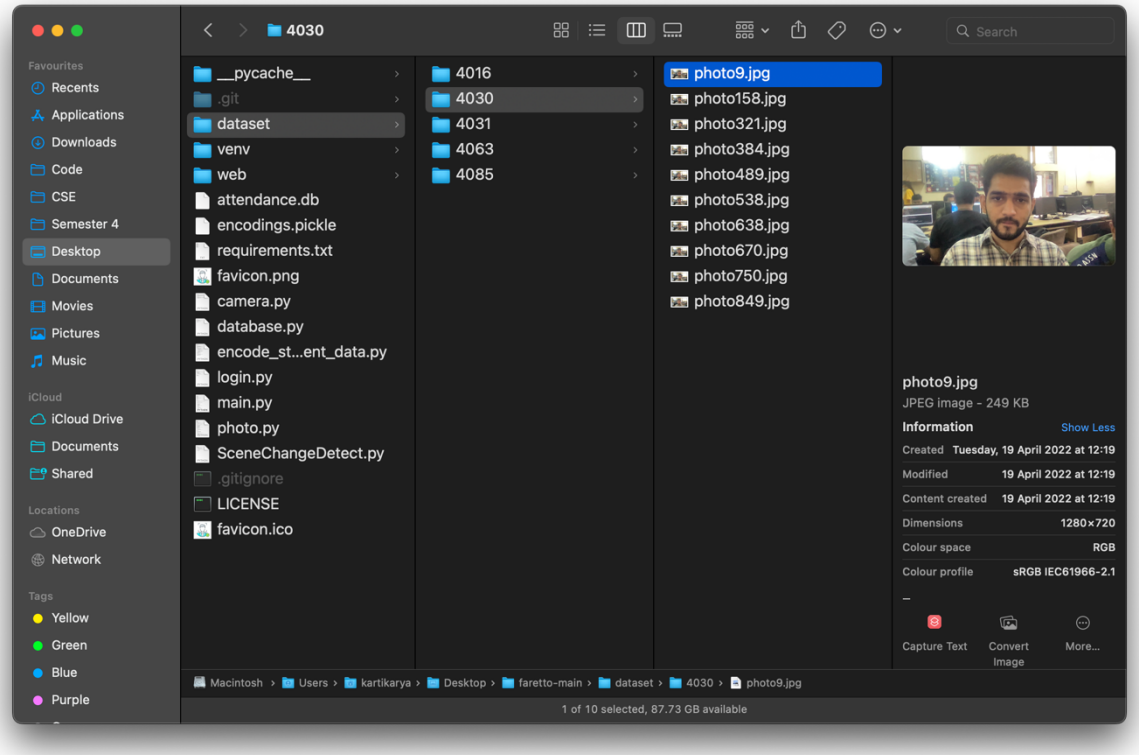


FIGURE 8.3 *SUCCESSFULLY CREATED DATASET OF STUDENT*

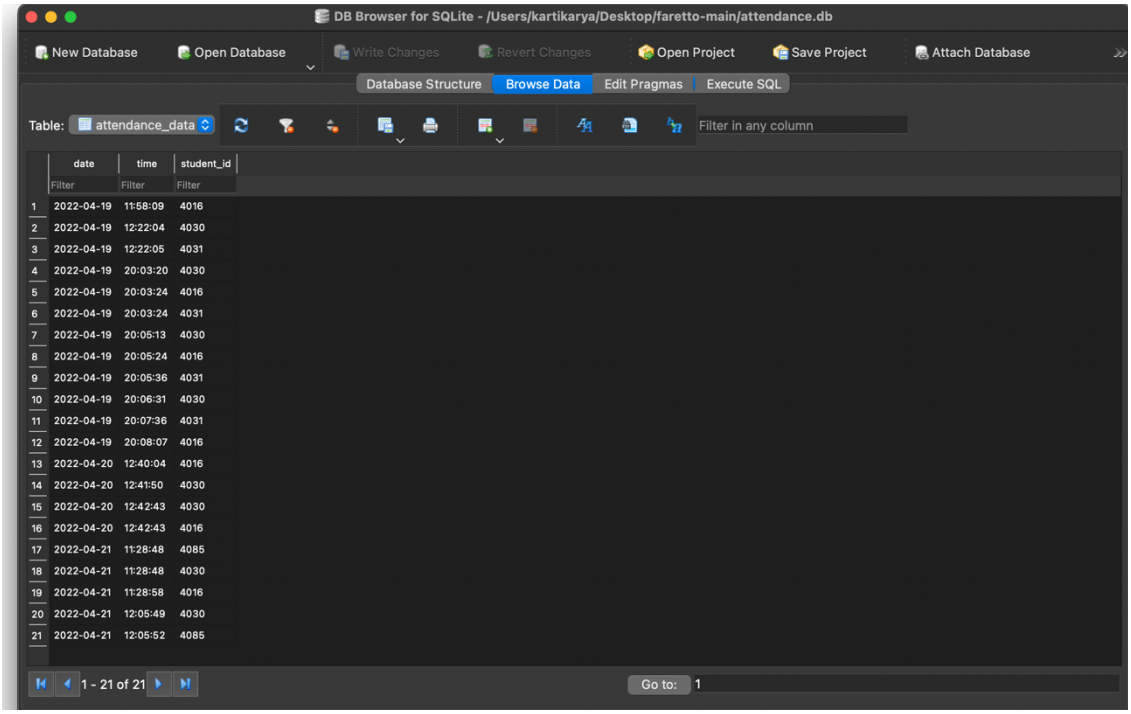


FIGURE 8.2 ATTENDANCE RECORD UPDATEDSUCCESSFULLY

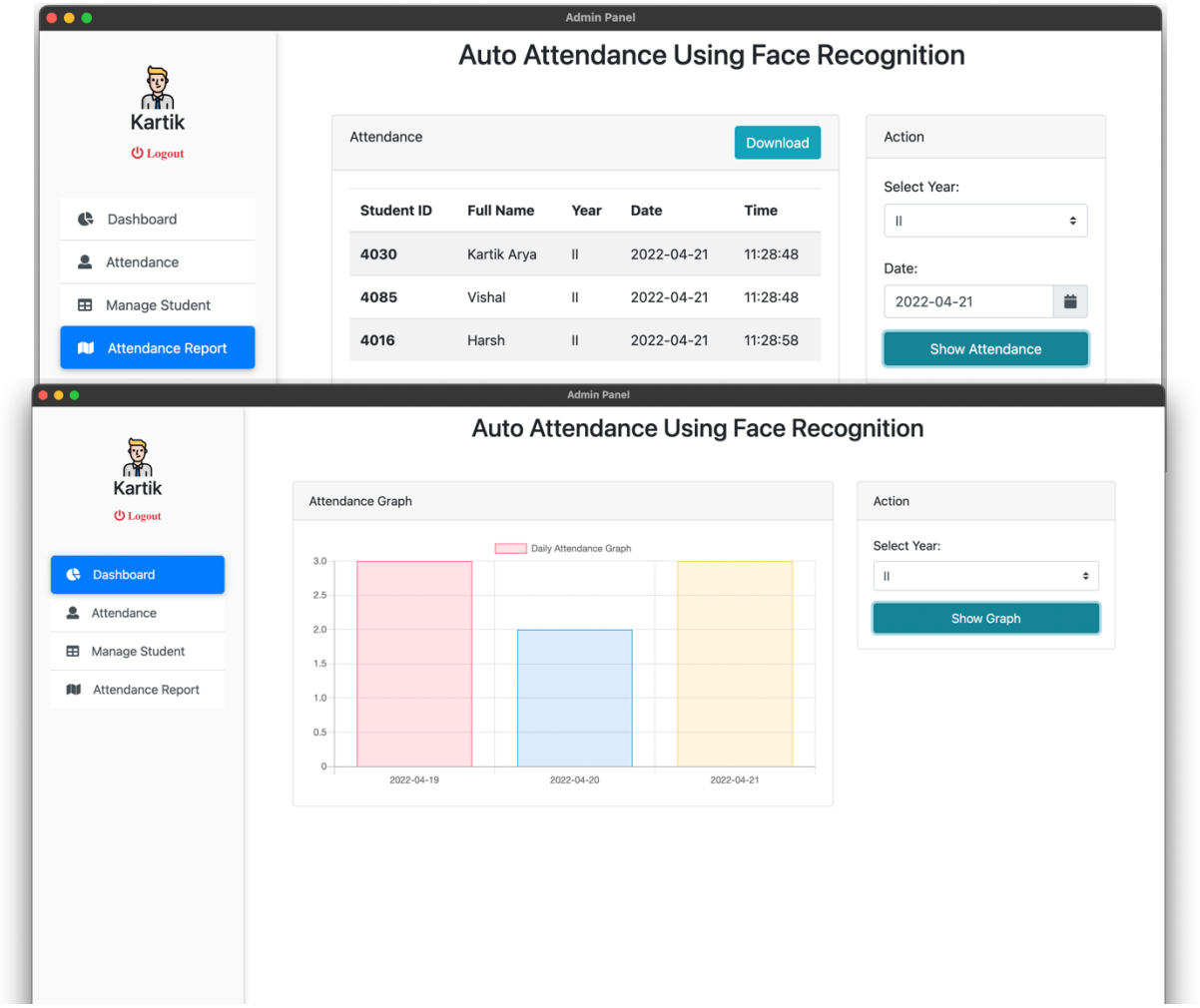


FIGURE 8.3 REPORT AND GRAPH DISPLAYED SUCCESSFULLY

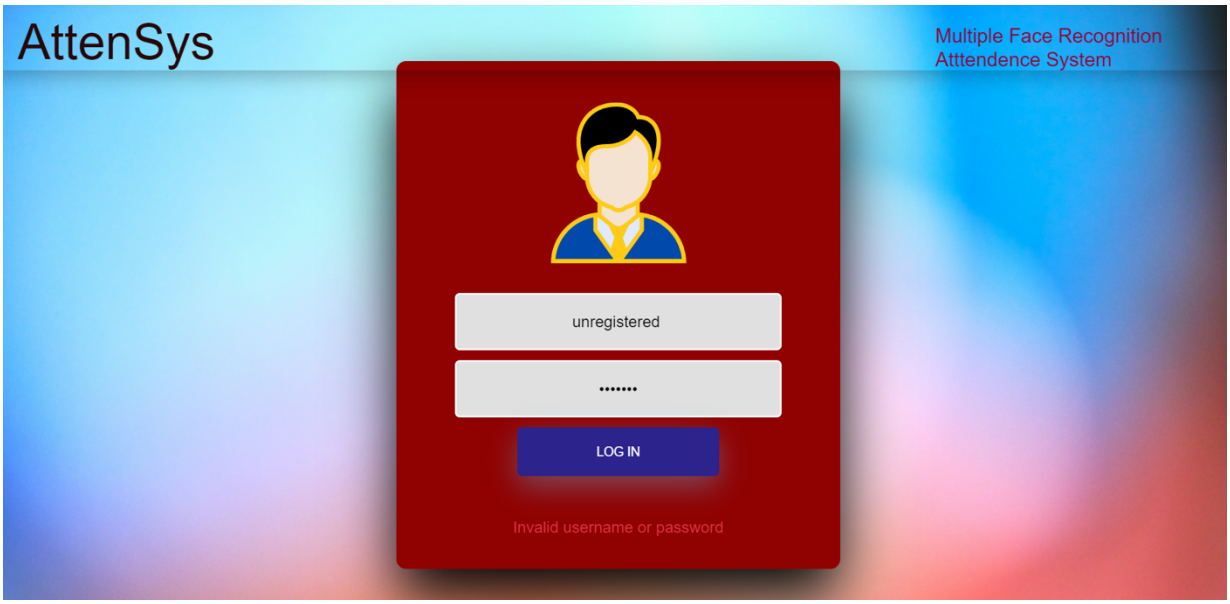


FIGURE 8.6 LOGIN WITH WRONG USERNAME & PASSWORD

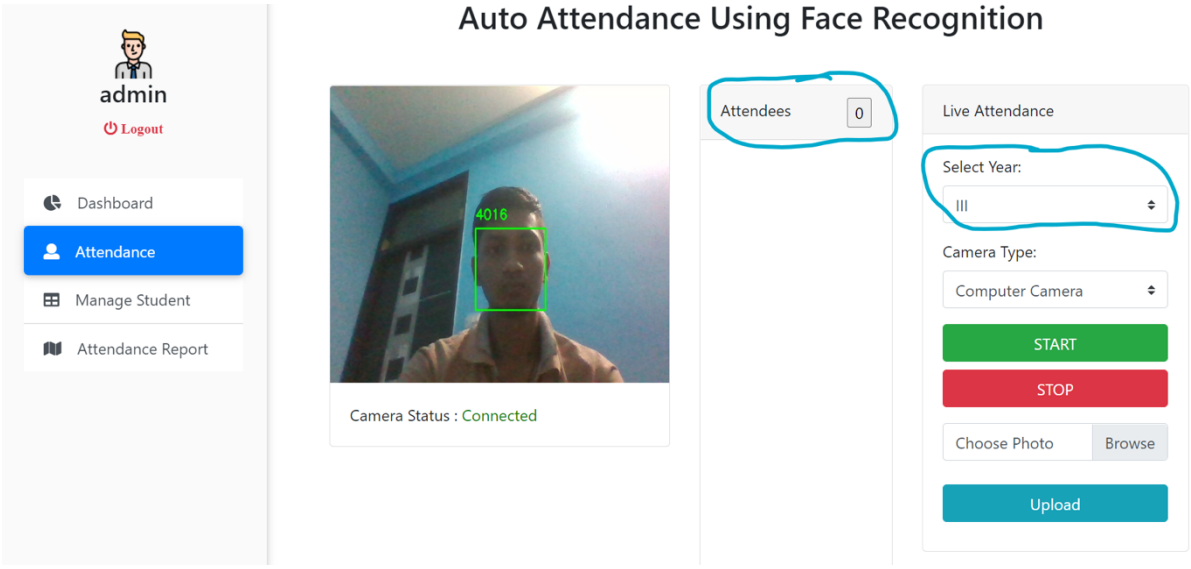


FIGURE 8.7 FACE IS RECOGNIZED BUT ATTENDANCE ISN'T MARKED DUE TO WRONG DETAILS

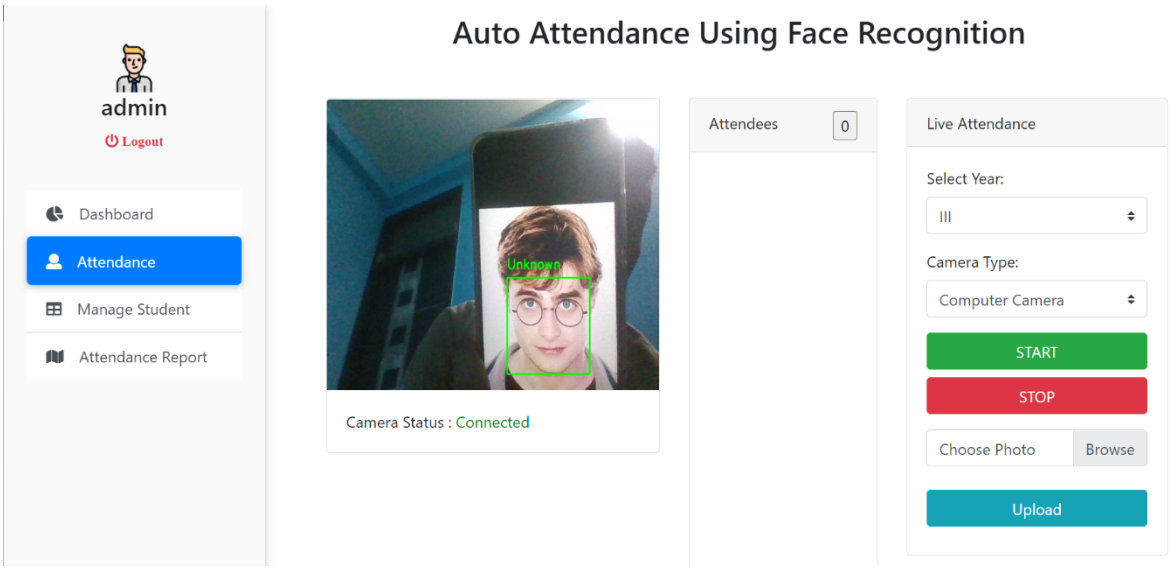


FIGURE 8.8 ATTENDANCE OF UNREGISTERED PERSON

9.Limitations and Future Scope

There are few limitations with this system .Firstly, under extreme illumination conditions, the accuracy of system will drop. Secondly, if images stored are blurred than there are chances of false recognition. Thirdly attendance of student can be marked by just showing any image of that student it doesn't required physical existence of that student. Finally, if a student applies makeup in an image than face recognition will be affected as important features would be covered.

We can eliminate illumination issue by using a good camera and start attendance process in better lighting conditions.. However, lighting source of camera varies from each device hence causes the system to be unstable.

The classifier takes time in training each image hence for large number of students, it would be time consuming. Although training the classifier only needs to be done when new student is added or when you start the system initially, it would be convenient if classifier takes lesser time in training while still maintaining accuracy of the system.

SCOPE FOR FUTURE WORK:

1. Currently, the system has reached the accuracy level up to 80% for partial and dense

images. It can further be improved to obtain higher accuracy levels.

2. Further, 2 or more IP cameras can be employed and each image can be processed

separately. The results of these can be merged to obtain better results and accuracy in

denser classrooms.

This system can be deployed for verification and attendance tracking at various government offices and corporates. For access control verification and identification of authentic users it can also be installed in bank lockers and vaults. For identification of criminals the system can be used by police force also.

9. BIBLIOGRAPHY

Websites:-

- https://github.com/ageitgey/face_recognition.git
- https://github.com/davisking/dlib.git
- GitHub: Where the world builds software · GitHub
- https://animate.style
- Share and Discover Knowledge on SlideShare
- GeeksForGeeks
- Youtube Tutorials
- en.wikipedia.org

Books :-

- Aggarwal, K. K., & Singh, Y. (2007). Software Engineering
- Software Engineering: A Practitioner's Approach. 8th edition. McGraw-Hill.