SHRI SHANKARACHARYA TECHNICAL CAMPUS

Shri Shankaracharya Group of Institutions

Faculty of Engineering & Technology

(Managed by Shri Gangajali Education Society, Bhilai)

JUNWANI, BHILAI-490 020 (CHHATTISGARH), INDIA

Phone: 0788-2291605, 2291607, Fax: 0788-2291606, E-mail: ssgi@ssgi.edu.in

Computer Science & Engineering

A project report

On

"Face Recognition System"

Submitted to

Chhattisgarh Swami Vivekanand Technical University,
Bhilai

Bachelor of Engineering

In

Computer Science & Engineering

By

AAKANKSHA RANI

BHAWANA TIWARI

GUNJAN TIWARI

UTKARSH SINGH

Under the guidance of

MRS. ADITI MISHRA, Associate Professor

Session: 2020-21

DECLARATION

We the undersigned solemnly declare the report of the project work entitled “Face Recognition System” is based our own work carried out during the course of my study under Mrs. Aditi Mishra mam, Associate Professor CSE department of SSGI, Bhilai.

I assert that the statements made and conclusion drawn are an outcome of the project work. I further declare that to the best of my knowledge and belief that the report does not contain any part of any work which has been submitted for the award of any other degree / diploma / certificate in this university or any other university.

(Signature)

Aakanksha Rani

Roll No : 301402217001

Enrollment no : BE3433

(Signature)

Bhawana Tiwari

Roll No : 301402217042

Enrollment no : BE3694

(Signature)

Gunjan Tiwari

Roll No : 301402217058

Enrollment No : BE3639

(Signature)

Utkarsh Singh

Roll No : 301402217155

Enrollment No : BE3687

CERTIFICATE

This is to certify the report of the project submitted is an outcome of the project work entitled : "**FACE RECOGNITION SYSTEM**" carried out by **UTKARSH SINGH** bearing roll no: **301402217155**, **BHAWANA TIWARI** bearing roll no: **301402217042**, **GUNJAN TIWARI** bearing roll no: **301402217058**, **AAKANKSHA RANI** bearing roll no: **301402217001** and carried out under my guidance and supervision for the award of Degree in Bachelor of Engineering in Computer Science and Engineering(FET) of Chhattisgarh Swami Vivekanand Technical University, Bhilai(CG).

To the best of my knowledge the report:

- Embodies the work of the candidates himself/herself.
- Has duly been completed.
- Fulfils the requirements of Ordinance relating to the BE degree of the University.
- Is upto the desired standards for the purpose of which it is submitted.

_____ (Signature of HOD with seal)

_____ (Signature of the Guide)

Mrs Aditi Mishra,
Computer science and Engineering
SSGI, BHILAI

The project work mentioned above hereby recommended and forwarded for examination and evaluation.

Date :

CERTIFICATE BY THE EXAMINERS

This is to certify that the project work entitled
"Face Recognition System"

Submitted by

Aakanksha Rani	Roll No : 301402217001	Enrollment no :BE3433
Bhawana Tiwari	Roll No : 301402217042	Enrollment no : BE3694
Gunjan Tiwari	Roll No : 301402217058	Enrollment No : BE3639
Utkarsh Singh	Roll No : 301402217155	Enrollment No : BE3687

Has been examined by the undersigned as a part of the examination for the award of Bachelor of Engineering degree in Computer Science & Engineering of Chhattisgarh Swami Vivekanand Technical University , Bhiai.

Internal Examiner
Date :

External Examiner
Date:

ACKNOWLEDGEMENT

Working for the project has been a great experience for us. There were moments of anxiety when we could not solve our problem for several days. But we have enjoyed every bit of process and are thankful to all the people associated with us during this period.

We convey our sincere thanks to our project guide **Mrs. Aditi Mishra**, Associate Professor for providing us all sorts of facilities. Her support and guidance helped us to carry out the project. We owe a great department of her gratitude for her constant advice, support, cooperation and encouragement throughout the project.

We would like to express our deep gratitude to our respected **Mrs. Aditi Mishra** for her help and support. We also pay special thanks for her helpful solution and comments enriched by her experience.

We would also like to express our deep gratitude to our college management **Shri I.P. Mishra, Chairman (Shri Gangajali Education Society, Bhilai), Mrs. Jaya Mishra, President (Shri Gangajali Education Society, Bhilai), Dr. P.B. Deshmukh, Director, SSGI and Mrs. Samta Gajbhiye (H.O.D. CSE Department)** for providing an education ambience. It will be our pleasure to acknowledge, utmost cooperation and valuable suggestions from time to time given by our staff members of our department to whom we owe our entire computer knowledge and also we would like to thank all those persons who have directly or indirectly helped us by providing books and computer peripherals and other necessary amenities which helped in the development of this project.

Aakanksha Rani
Bhawana Tiwari
Gunjan Tiwari
Utkarsh Singh

FACE RECOGNITION SYSTEM

Department of Computer Science & Engineering

SSTC Campus, Bhilai, India

ABSTRACT

Face Detection and Recognition System is a system which would primarily aim on performing a successful log in access operation of the concerned user on the basis of identification and recognition of their unique face provided that the user is an authorized personal (old user) within the confines of the system. This Face Recognition System is now a days is also use in education intuitions for marking the presence of the student in the curriculum. If the arriving user is completely new in the system then this Face Detection and Recognition System**,** through a non-complex and a friendly user interface, will ask for and store the required the credentials of the user in order to convert them into an authorized user for all the future purposes i.e. future log in access operations.

Further applications of this system could include a face recognition based attendance system. The system will be implemented in Python and Flask Frame work using Kairos API.

.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS USED

1	FDR	Face Recognition and Detection

2	DBMS	Database Management System
3	MIS	Management Information System
4	IP	Image processing
5	OpenCV	Library used for image processing
6	OS	Operating System

INTRODUCTION

Face recognition systems are part of facial image processing applications and their significance as a research area is increasing recently. They use biometric information of humans and are applicable easily instead of fingerprint, iris, signature etc., because these types of biometrics are not much suitable for non-collaborative people. Face recognition systems are usually applied and preferred for people and security cameras in metropolitan life. These systems can be used for crime prevention, video surveillance, person verification, and similar security activities. Face recognition system is a complex image-processing problem in real world applications with complex effects of illumination, occlusion, and imaging condition on the live images. It is a combination of face detection and recognition techniques in image analyzes.

Detection application is used to find position of the faces in a given image. Recognition algorithm is used to classify given images with known structured properties, which are used commonly in most of the computer vision applications. Recognition applications use standard images, and detection algorithms detect the faces and extract face images which include eyes, eyebrows, nose, and mouth. That makes the algorithm more complicated than single detection or recognition algorithm. The first step for face recognition system is to acquire an image from a camera. Second step is face detection from the acquired image. As a third step, face recognition that takes the face images from output of detection part. Final step is person identity as a result of recognition part. Acquiring images to computer from camera and computational medium (environment) via frame grabber is the first step in face recognition system applications. The input image, in the form of digital data, is sent to Kairos API to extracting subject ID across each face in the image. Briefly, knowledge-based methods are derived from human knowledge for features that makes a face. After faces are detected, the faces should be recognized to identify the persons in the face images. In the literature, most of the methods used images from an available face library, which is made of standard images. After faces are detected, standard images should be created with some methods. When user next time appears for

mark the attendance that particular image is identified by using the Subject Id which is provided by Kairos at the time of registration.

1.1 Overview:

Face detection and recognition module detects faces from the image captured by the camera, and the image of the face is cropped and stored. The module recognizes the images of student's face, which have been registered manually with their names and ID codes in the database. Face detection data and face recognition data are recorded into the database.

2 LITERATURE SURVEY & SCOPE

The motivation for this research is taken from recent studies which have demonstrated increased retrieval effectiveness of face detection systems. The methodology is derived from previous studies which model the impact that ambiguity and its subsequent resolution have on IR.

Human beings perform face recognition automatically every day and practically with no effort. Although it sounds like a very simple task for us, it has proven to be a complex task for a computer, as it has many variables that can impair the accuracy of the methods, for example: illumination variation, low resolution, and occlusion, amongst other.

In computer science, face recognition is basically the task of recognizing a person based on its facial image. It has become very popular in the last two decades, mainly because of the new methods developed and the high quality of the current videos/cameras. Note that face recognition is different of face detection:

Face Detection: it has the objective of finding the faces (location and size) in an image and probably extract them to be used by the face recognition algorithm. Face Recognition: with the facial images already extracted, cropped, resized and usually converted to gray scale, the face recognition algorithm is responsible for finding characteristics which best describe the image.

The face recognition systems can operate basically in two modes:

Verification or authentication of a facial image: It basically compares the input facial image with the facial image related to the user which is requiring the authentication.

Identification or facial recognition: it basically compares the input facial image with all facial images stores in the gallery of the Kairos and identifies the particular image of user with the subject Id which is provided by the Kairos at the time of the registration.

OpenCV was started at Intel in 1999 by Gary Bradsky and the first release came out in 2000. Vadim Pisarevsky joined Gary Bradsky to manage Intel's Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle who won 2005 DARPA Grand Challenge. Later its active development continued under the support of Willow Garage, with Gary Bradsky and Vadim Pisarevsky leading the project. Right now, OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning and it is expanding day-by-day.

2.2 Application of Face Recognition System:

- **Retail**—Large retailers are using facial recognition to instantly recognize customers and present offers. They can also use it to catch shoplifters augmented with camera footage. The entertainment industry, casinos, and theme parks have also caught on to its uses. Companies like NTechLab, Kairos use face recognition technology to provide customer analytics.
- **Advertising**—visual intelligence is providing not just superficial identity but it's also checking on emotions, expressions, and features to target audience accordingly. Gumgum is a facial recognition firm that can serve targeted advertising using faces.
- **Healthcare—**Analyzing faces to provide automated diagnosis of rare genetic conditions, such as Hajdu-Cheney syndrome is being explored. Recognition of expressions and emotions may give autistic people a grasp of social signals they find elusive.
- **Pandemic** — In recent case of Global Pandemic (COVID 19), face recognition system comes very handy. It can be integrated with different system to help in various fields. It can be used in institution to check whether someone is wearing mask or not. It can be integrated with an attendance system, which can come very handy as we cannot use bio-metric system in this global pandemic.

3 METHODOLOGY

3.1 Kairos API Functioning:

Face detection is a computing technology term that is used when software is used to determine the existence, location and size of a human face in a particular photo. The software is clever enough to detect the facial features, while at the same time ignoring other objects like trees, buildings and bodies.

A person's face is actually a rich source of information. It is possible to tell, simply from a human face, whether a person is male or female, approximately how old they are, and from their expression, how they are feeling.

Human beings can process faces very quickly. It only takes you a split second to determine the key attributes of anyone you look at. In the case of computers, however, it is a somewhat more complex process.

In simple terms, a computer attempting face detection will begin by examining either a still photo or a video image. It then has the task of determining if there are any faces present in that picture, distinguishing these faces from whatever else is in the background. It has to do this regardless of illumination, orientation or camera distance.

There are a number of methods which a computer can use to achieve this.

3.2 Finding Faces in Images with Controlled Backgrounds

There is a relatively straight-forward method that can be used when you simply have a frontal face image against a plain background. In this case the software can easily remove the background, leaving face boundaries. If software uses this approach it tends to have a number of different classifiers for detecting different types of front-on faces, along with

some for profile faces. It will attempt to detect eyes, a nose, a mouth, and in some cases even a whole body for pictures that show more than just a face.

3.3 Finding Faces by Color

This is a relatively simplistic method that a computer can use to look for faces. It obviously requires that the photos or video images used be color. The software scans the picture looking for areas that are of a typical skin color, then looking for face segments. A problem with this technique is that skin color varies from race to race, and this method does not work as well for all skin color. Varying lighting conditions changes the exact hue of a person's skin in the picture, and that can have a major effect on facial detection, too.

3.4 Finding Faces by Motion

When you are using video images you can use movement as a guide. Faces are usually moving in real-time videos, so one option is for the software to capture the moving area. Of course, other parts of videos also move, so the software needs to look for particular reference points to indicate that it is actually a face that is moving.

One specific face movement is blinking. If the software can determine a regular blinking pattern (two eyes blinking together, symmetrically positioned) then this is a good indication that there is a face. From this regular blinking pattern the computer can determine the area of the video image that is actually the face, using one of a number of face models.

There will be a number of face models in the software, containing the appearance, shape and motion of faces. There are actually a variety of different face shapes, roughly categorised as oval, rectangle, round, square, heart and triangle.

As well as blinking, there are various other motions that signpost to the computer that the image may contain a face. These include raised eyebrows, flared nostrils, wrinkled foreheads and opened mouths.

Once one of these actions is detected, the computer will pass their face models over the video image and try and determine a facial match.

Once a face is detected, and a particular face model matched with a particular movement, the model is laid over the face, enabling face tracking to pick up further face movements.

3.5 OpenCV Module for capturing the image:

OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems. Computer vision is concerned with modeling and replicating human vision using computer software and hardware. In this chapter, you will learn in detail about this.

Image processing studies image to image transformation. The input and output of image processing are both images.

Computer vision is the construction of explicit, meaningful descriptions of physical objects from their image. The output of computer vision is a description or an interpretation of structures in 3D scene.

Installing the package:

Run: pip install opencv-contrib-python.

3.5.1 Features of OpenCV-Python:

- Read and write image.
- Capture and save videos
- Process images (filter, transform)
- Perform feature detection
- Detect specific objects such as faces, eyes, cars, in the videos or images.
- Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.
- OpenCV was originally developed in C++. In addition to it, Python and Java bindings were provided.
- OpenCV runs on various Operating Systems such as windows, Linux, etc.
- Thresholds pixel for scaling the image in obtain the different binary images.
- OpenCV-Python contrib python module for showing, reading, sacling, writing, capturing the images

3.5.2 OpenCV Library Modules:

Following are the main library modules of the OpenCV library.

a. **Core Functionality**

   This module covers the basic data structures such as Scalar, Point, Range, etc., that are used to build OpenCV applications. In addition to these, it also includes

the multidimensional array Mat, which is used to store the images. In the Java library of OpenCV, this module is included as a package with the name org.opencv.core.

b. Image Processing

This module covers various image processing operations such as image filtering, geometrical image transformations, color space conversion, histograms, etc. In the Java library of OpenCV, this module is included as a package with the name org.opencv.imgproc.

c. Video

This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the Java library of OpenCV, this module is included as a package with the name org.opencv.videoio.

d. Video I/O

This module explains the video capturing and video codecs using OpenCV library. In the Java library of OpenCV, this module is included as a package with the name org.opencv.videoio.

e. calib3d

This module includes algorithms regarding basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence and elements of 3D reconstruction. In the Java library of OpenCV, this module is included as a package with the name org.opencv.calib3d.

f. features2d

This module includes the concepts of feature detection and description. In the Java library of OpenCV, this module is included as a package with the name org.opencv.features2d.

g. Object Detect

This module includes the detection of objects and instances of the predefined classes such as faces, eyes, mugs, people, cars, etc. In the Java library of OpenCV, this module is included as a package with the name org.opencv.objdetect.

h. Highgui

This is an easy-to-use interface with simple UI capabilities. In the Java library of OpenCV, the features of this module is included in two different packages namely, org.opencv.imgcodecs and org.opencv.videoio.

i. Haar-cascade Detection in OpenCV

OpenCV comes with a trainer as well as detector. If we want to train our own classifier for any object like car, planes etc. We can use OpenCV to create one.

j. OpenCV functions for Reading, Showing, Writing an Image File:

- **imread() function** – This is the function for reading an image. OpenCV imread() supports various image formats like PNG, JPEG, JPG, TIFF, etc.
- **imshow() function** – This is the function for showing an image in a window. The window automatically fits to the image size. OpenCV imshow() supports various image formats like PNG, JPEG, JPG, TIFF, etc.
- **imwrite() function** – This is the function for writing an image. OpenCV imwrite() supports various image formats like PNG, JPEG, JPG, TIFF, etc.

3.6 Python Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

3.6.1 Installation of Flask in Linux based version or windows:

- pip install flask
- sudo apt-get install flask

Test the Flask Installation:

```
from flask import Flask

app = Flask(__name__)

@app.route('/')

def hello_world():

return 'Hello World’

if __name__ == '__main__':

app.run(debug=True)
```

Figure: 3.1

Description:

- **host:** Hostname to listen on. Defaults to 127.0.0.1 (local host). Set to ‘0.0.0.0’ to have server available externally.

- **port:** Defaults to 5000.

- **debug:** Defaults to false. If set to true, provides a debug information.

- **options:** To be forwarded to underlying Werkzeug server.

.

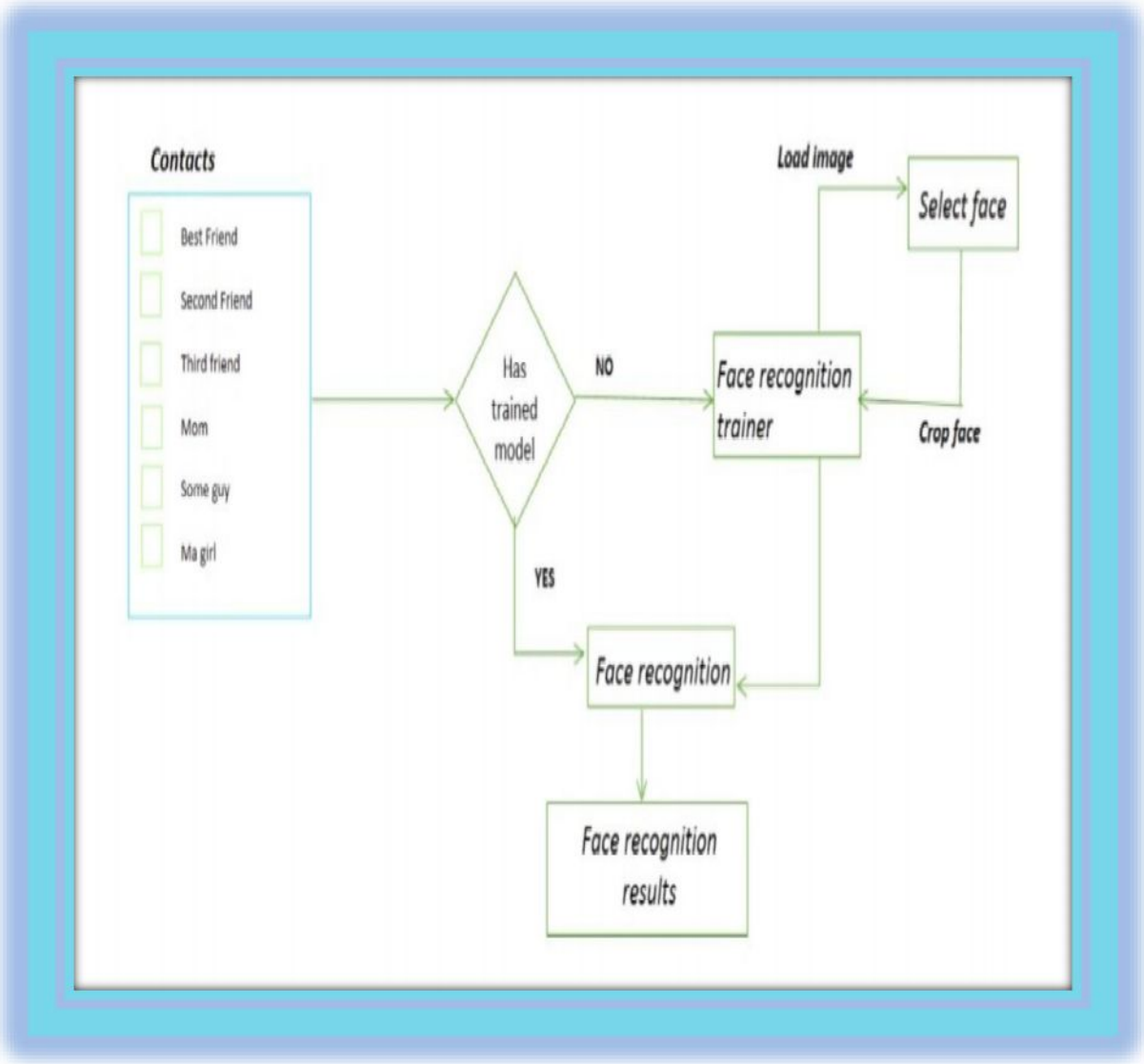

Figure 3.2: Working of Face Recognition System

We propose a method that take the attendance using face recognition based on continuous observation. Our purpose is to obtain the attendance, positions and images of students' face, which are useful information in the classroom lecture.

4 SYSTEM REQUIREMENT SPECIFICATION

4.1 Overall Description:

This section and its subsections contain the description of the project components such as interfaces, performance requirements, design constraints, assumptions and dependencies etc.

4.2 Product Perspective:

The application is Python based, Flask-Framework self-contained and independent product.

4.3 System Interfaces:

List each system interface and identify the functionality of the system (hardware and software both) to accomplish the system requirement and interface description to match the system.

4.4 User Interfaces:

The application will have a user friendly and menu based interface. Following screens will be provided:

a. A Registration/LogIn screen to enter the basic details.

b. Proceeding with SignIn option will lead you to another prompt window which would be the camera feed recognizing the user's face.

c. Proceeding with SignUp option will lead you to another prompt window to look straight into the camera.

d. After the user's face is successfully captured, system is store the image on the local system and as well as sent to Kairos gallery.

Proceeding with the exit option, program will successfully exit.

4.5 Hardware Interfaces:

- Screen resolution of at least 500 x 300 pixels is required for proper and complete viewing of screens. Higher resolutions in wide-screen mode will be better for a better view.

- Support for camera is needed for the software to work.

- Python programming language with OpenCV module is required for its working
- Other hardware interface specifications are as follows:

The following table required Minimum Hardware Interfaces at Client Side as well as for the Server Side:

HARDWARE INTERFACES - CLIENT SIDE (Minimum)	
HARDWARE	**SPECIFICATIONS**
Intel Pentium 4 and Higher Processor	
Camera (Low to High resolution)	
USB 2.0 / PS2 Mouse	
Keyboard (QWERTY)	

Table 4.1 – Minimum Client Side Hardware Interfaces

HARDWARE INTERFACES - SERVER SIDE (Minimum)		
HARDWARE	**RAM**	**DISK SPACE**
Intel Core i3 / i5 / i7 2.27 GHz and higher Or AMD 4XXX and higher	2048 MB	20 GB

Table 4.2 – Minimum Server Side Hardware Interfaces

The following table required recommended Hardware Interfaces at Client Side as well as for the Server side.

HARDWARE INTERFACES - CLIENT SIDE (Recommended)		
HARDWARE	**RAM**	**DISK SPACE**
Intel Core i3 / i5 / i7 2.27 GHz and higher Or AMD 4XXX and higher	1024 MB	1 GB
Camera (Low to high resolution)		
USB 2.0 Optical Mouse		
Keyboard (QWERTY)		

Table 4.3 – Recommended Client-Side Hardware Interfaces

HARDWARE INTERFACES - SERVER SIDE (Recommended)		
HARDWARE	**RAM**	**DISK SPACE**
Intel Xeon higher Or AMD equivalent	4096 MB	40 GB

Table 4.4 – Recommended Server-Side Hardware Interface

4.6 Software Interfaces:

- Any Microsoft Windows 7 and higher (Windows 7 / 8 / 8.1 / 10) or equivalent Linux based operating system with minimum kernel support 3.X.
- Python 3 installed and working properly.
- Working Webcam.
- Following tables contain the minimum and recommended software interfaces:

SOFTWARE INTERFACES (Minimum)		
Software Tool	**Version**	**Purpose of Use**
Operating system	Windows 7 and higher or Linux with Kernel 3.x and higher	Installation and operational platform
Language	Python version 3 or above.	Installation and operational platform.

Table 4.5 – Minimum Software Interfaces

SOFTWARE INTERFACES (Recommended)		
Software Tool	**Version**	**Purpose of Use**
Operating system	Windows 8 &higher or Linux with Kernel 4.x & higher	Installation and operational platform
Language	Python version 3 or above.	Installation and operational platform.

Table 4.6 – Recommended Software Interfaces

4.7 System Product Features:

- **Efficiency:**

  As the application is built on Python using OpenCV and Haar-cascade files, the system is relatively efficient. The images are converted and stored in a compressed manner, reducing the cost of size.

- **Reliability:**

  The application is reliable in terms of privacy (no other member/employee can access other member/employee profile), services, accessibility.

- **Economic:**

  While considering economic feasibility, it is checked in points like performance, information and outputs from the system. The cost incurred to develop the system is freeware & does not incur the cost to the project. Backend database technology is a freeware. This justifies economic feasibility of the system.

5 SYSTEM DESIGN SPECIFICATION

System design presents the schematic view of the complete system along with its major components and their connectivity.

5.1 High Level Design Diagrams

Use Case Diagrams:

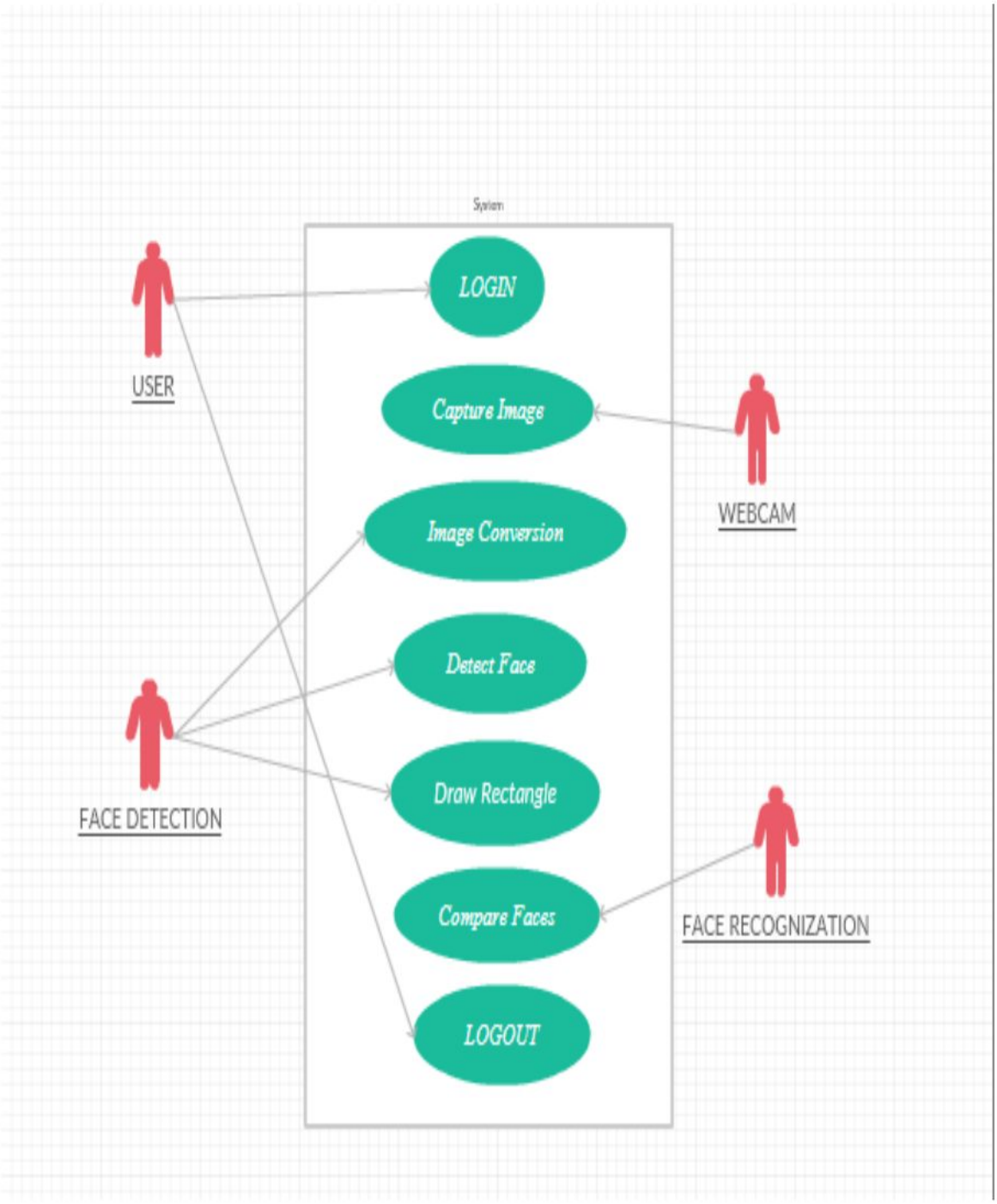

Figure 5.1: Use case Diagram

Activity Diagram

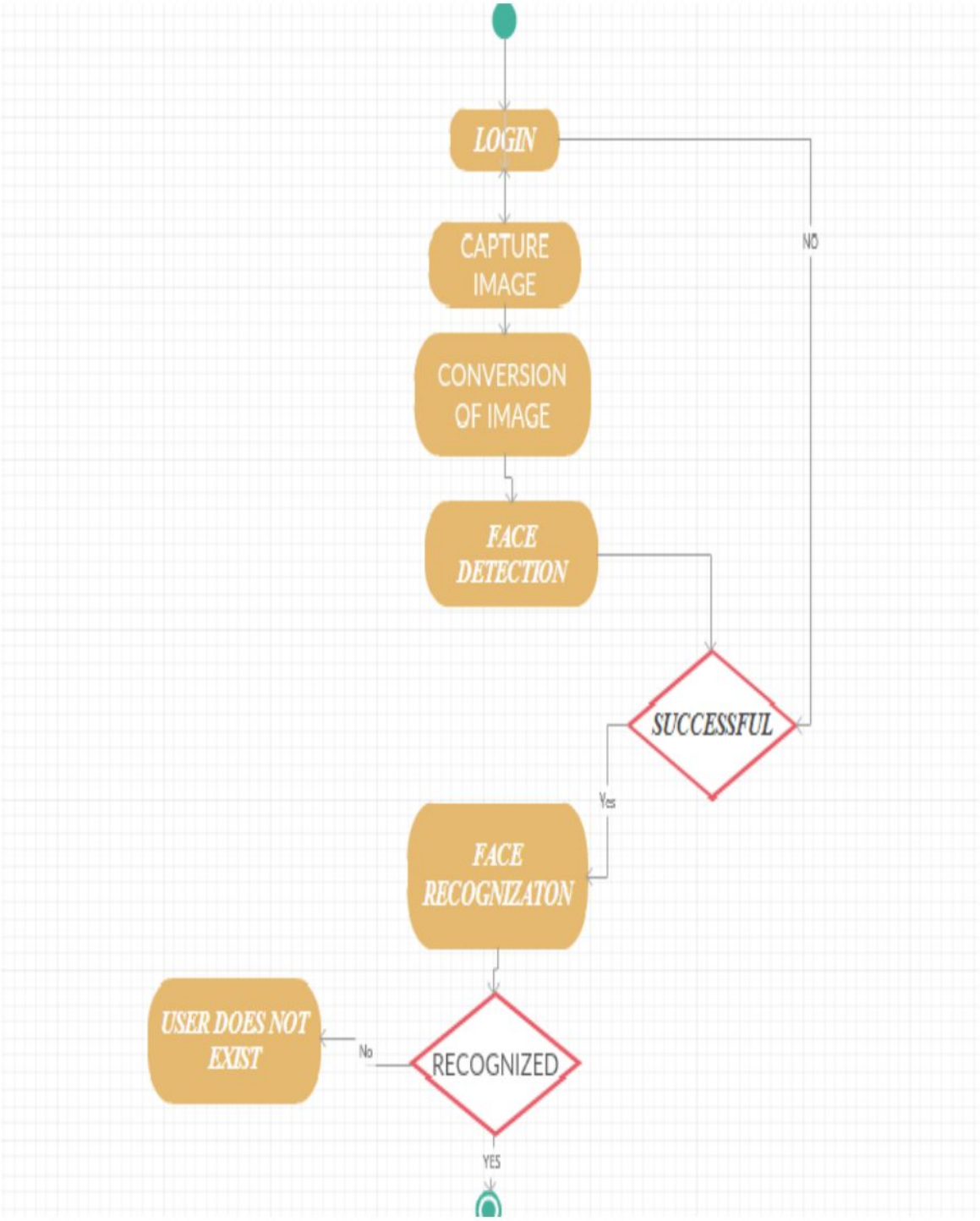

Figure 5.2: Activity Diagram

Sequence Diagram

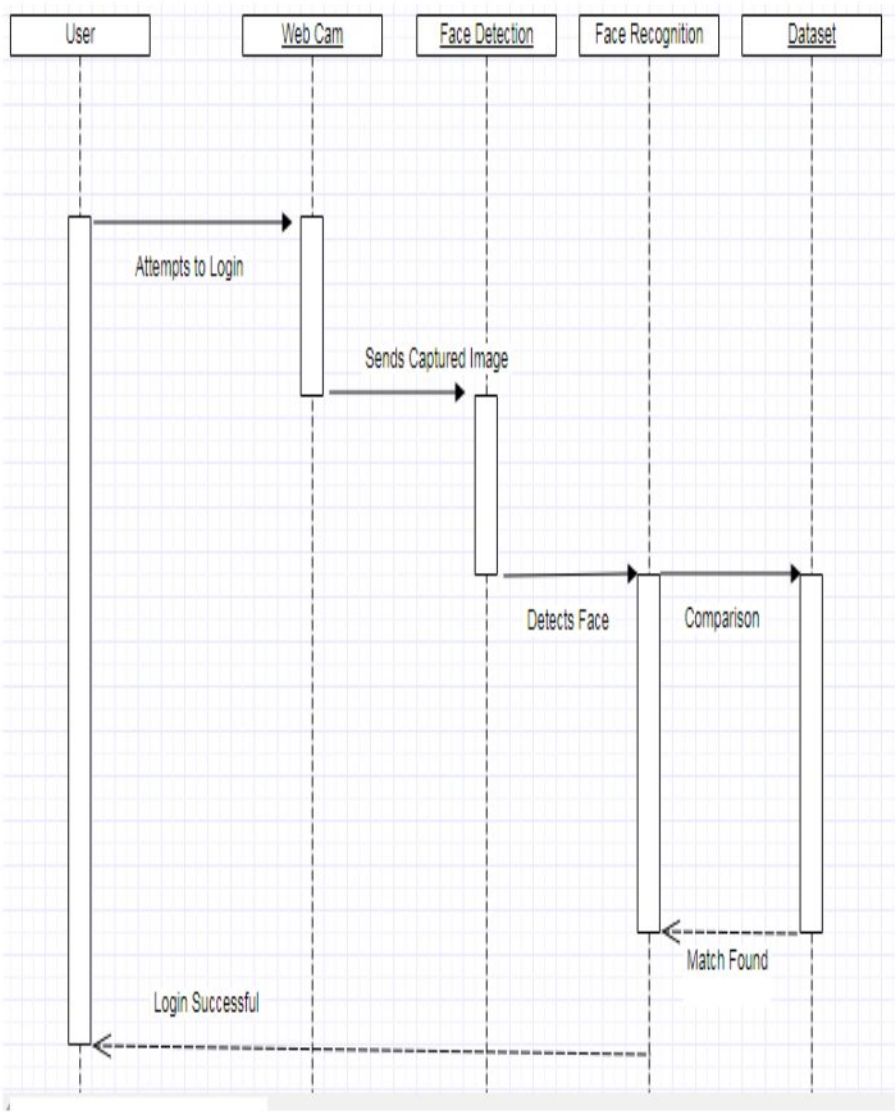

Figure 5.3: Sequence Diagram

Class Diagram

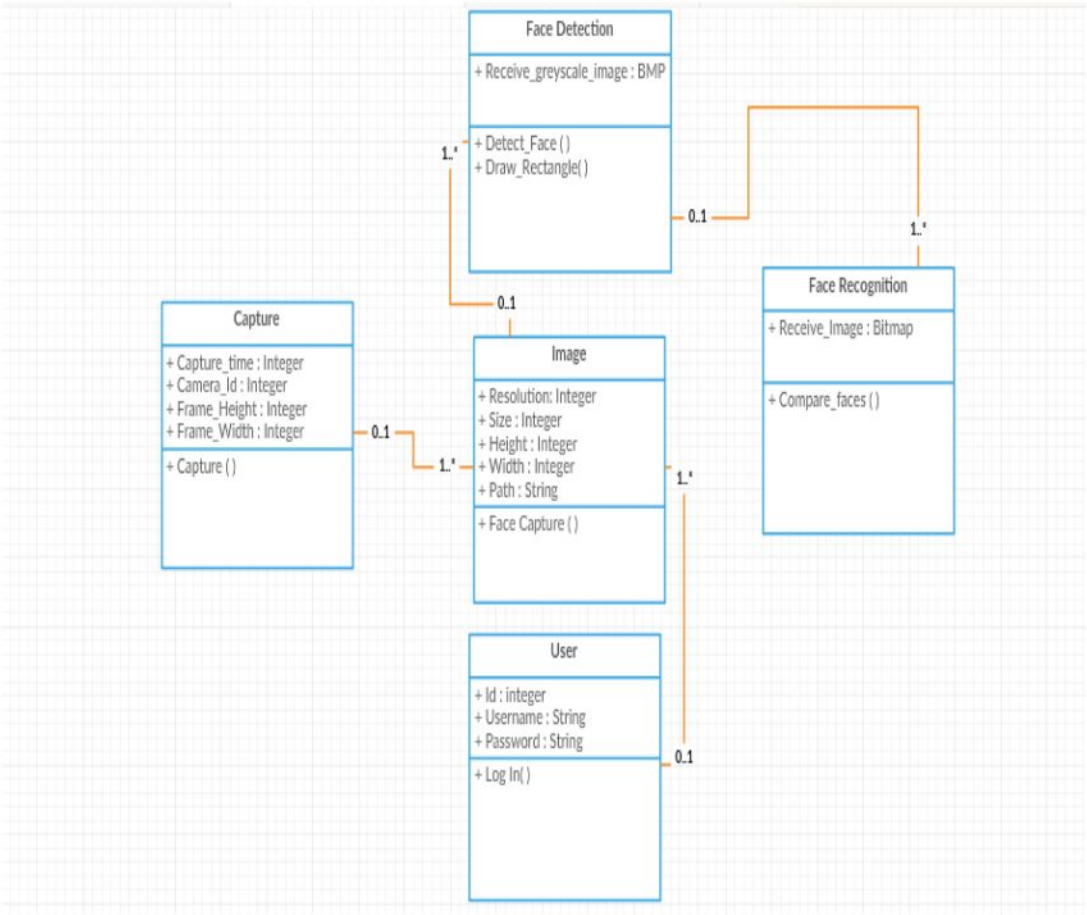

Figure 5.4: Class Diagram

Object Diagram

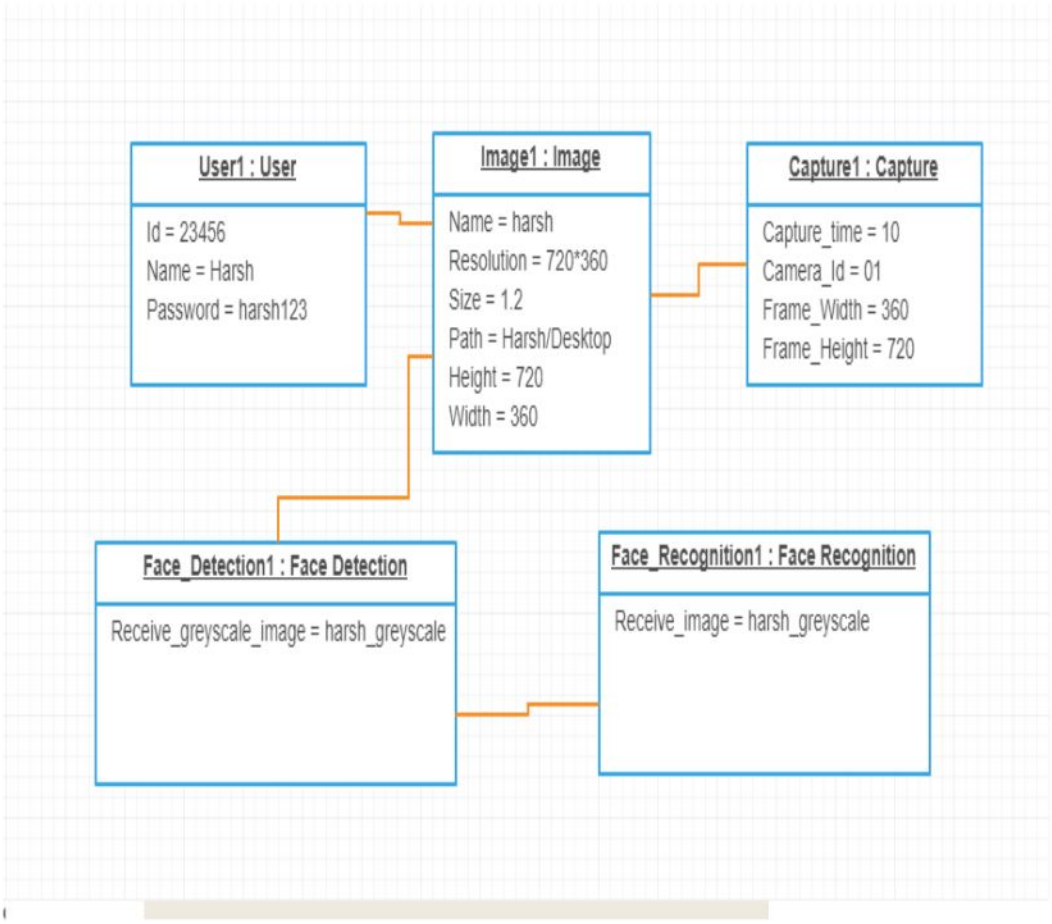

Figure 5.5: Object Diagram

6 SCREENSHOTS OF PROJECT

Image 6.1: Home Page

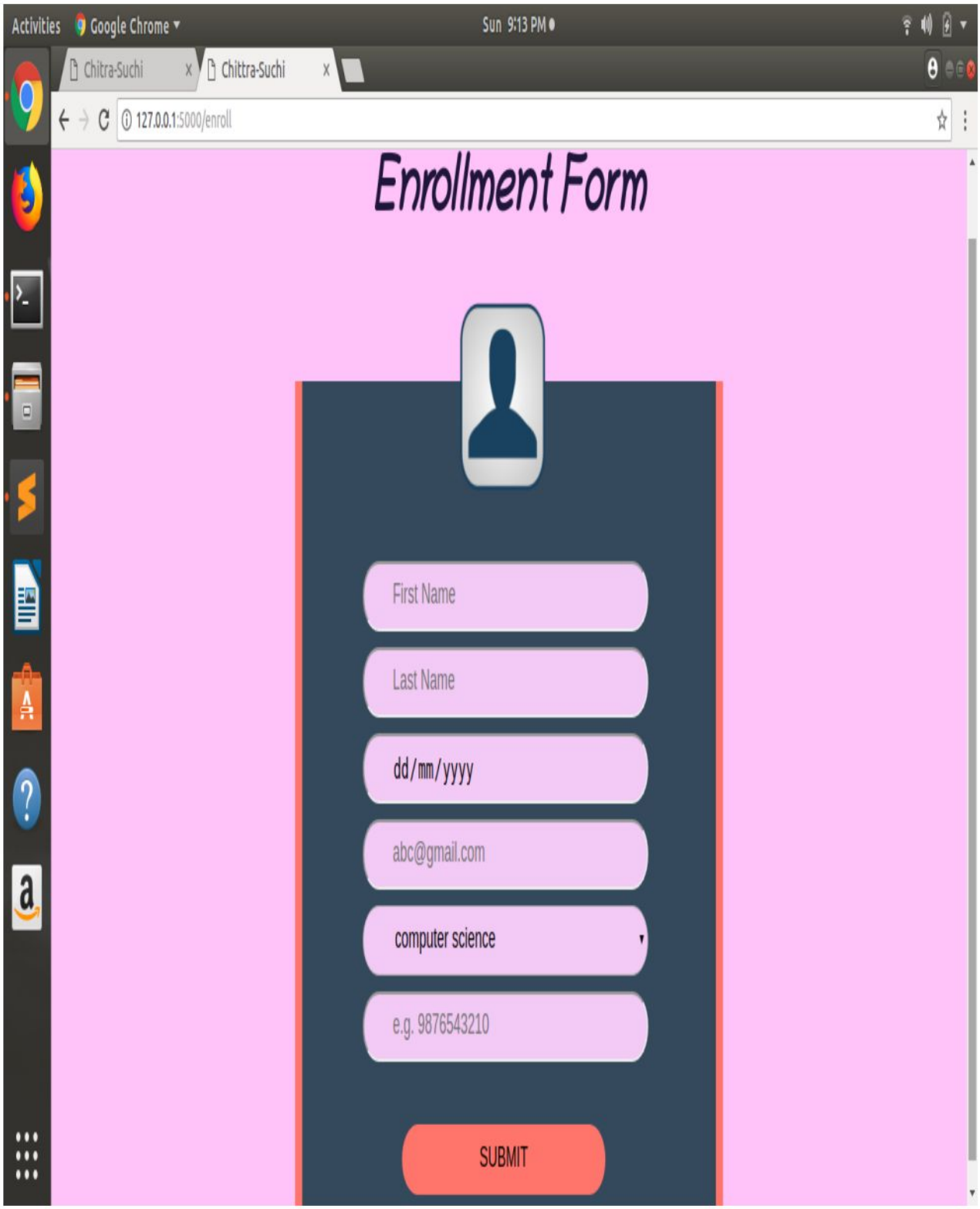

Image 6.2: Registration page

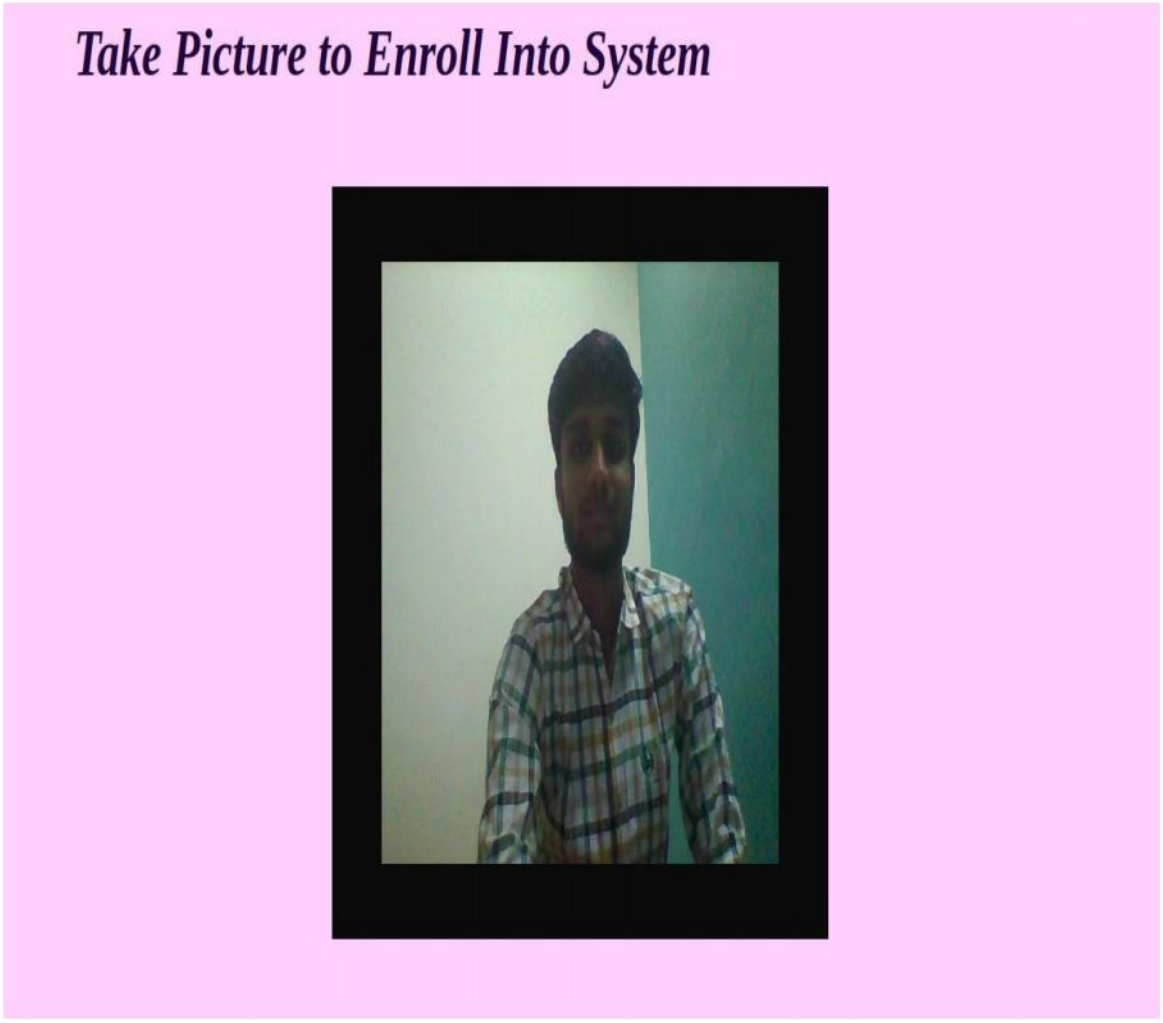

Image 6.3: Take Picture for enrolling

Image 6.4: Mark attendance

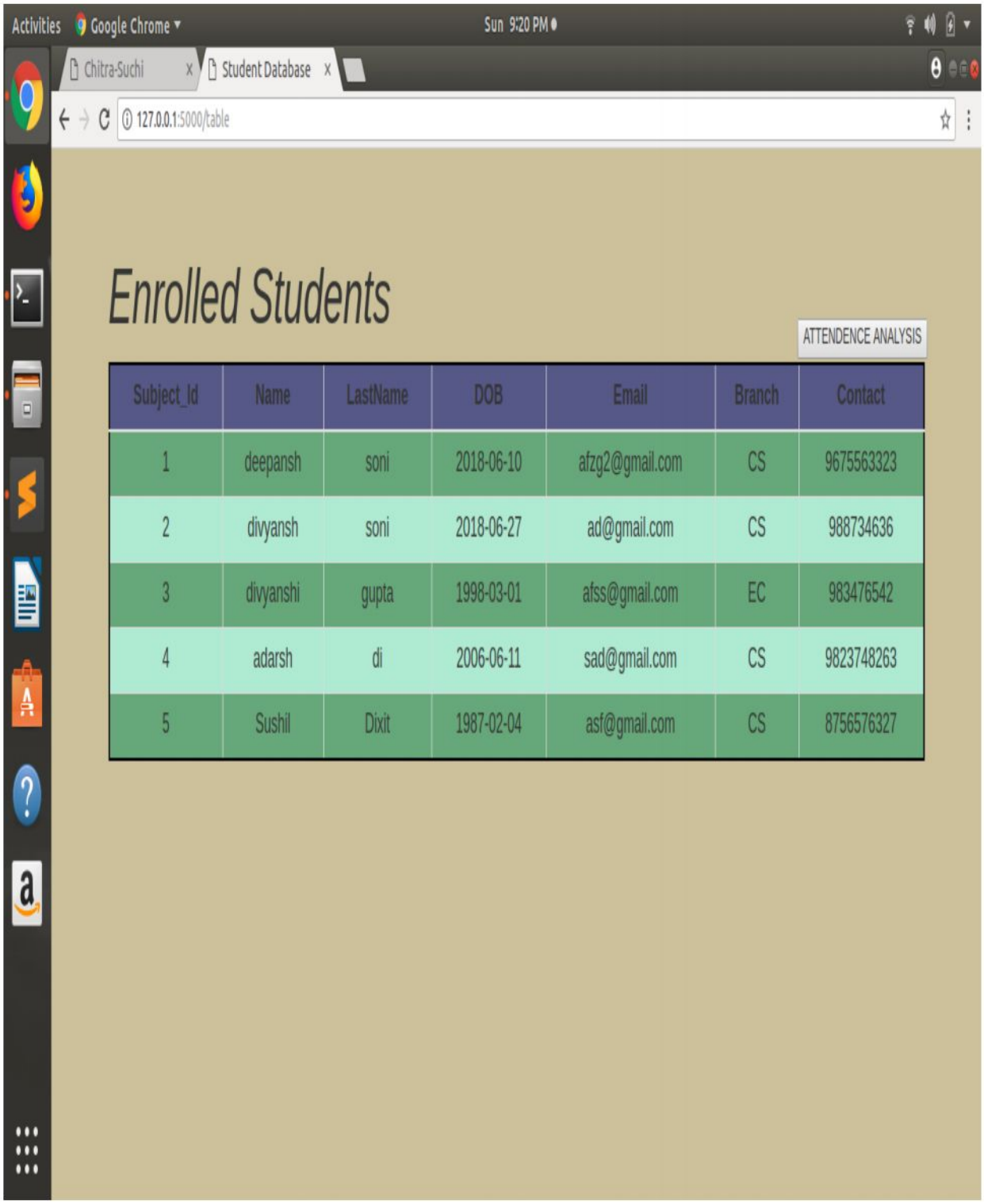

Image 6.5: Database record

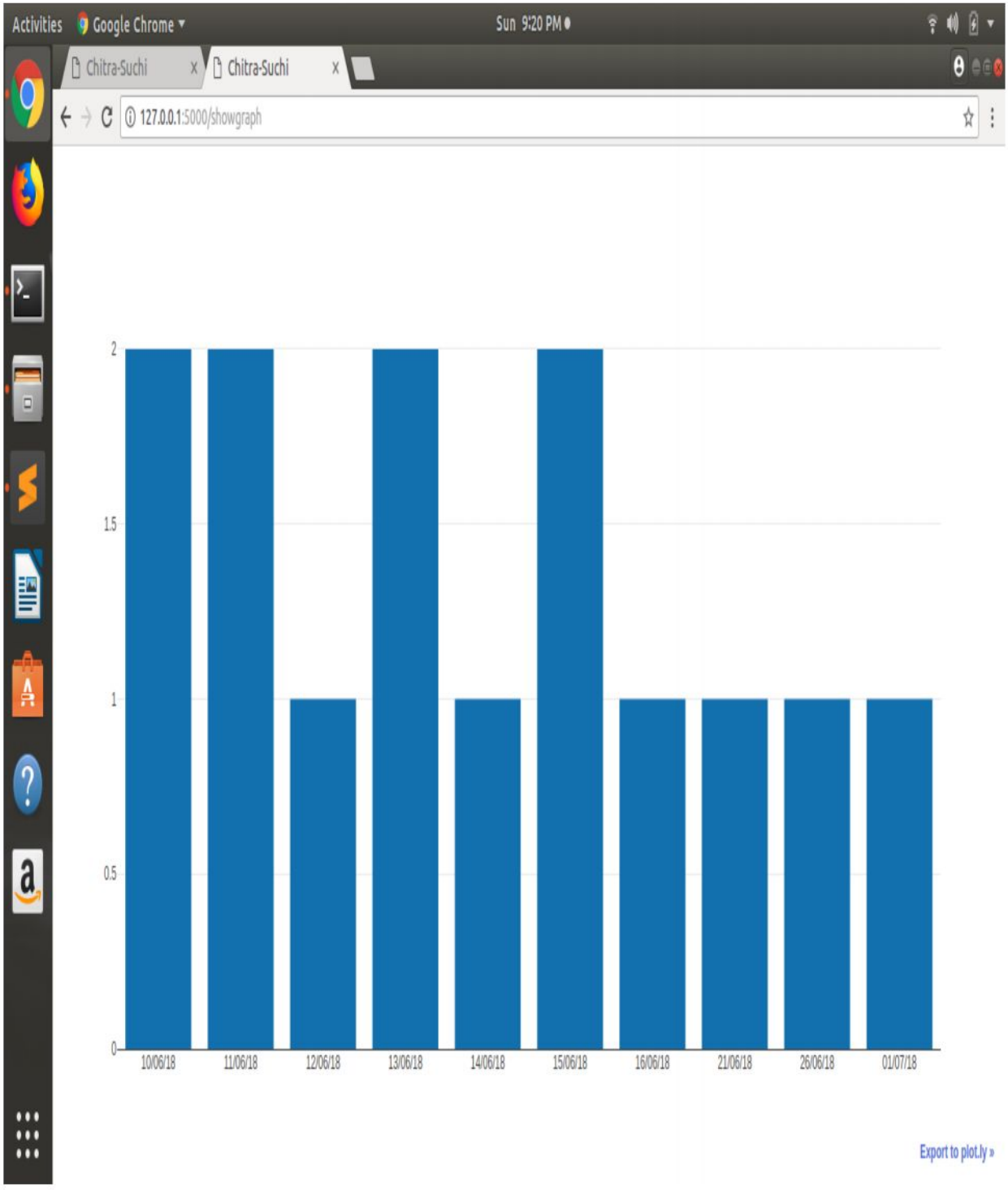

Image 6.6: Bar-graph of number of student present

7 PROJECT SUMMARY RESULT AND CONCLUSIONS

Using Kairos API as a cloud for storage the image. Which give the subject Id for particular image and provide a register name with particular image successfully.

The object of this project is to harness the power of Artificial Intelligence for our practical and potential use. This report explains to extensively cover this concept and plant a seed of inquisitiveness in the mind of users.

This system can allow users to quickly and hesitantly gain authorized access in a particular system without any requirement of heavy and time consuming credentials which not only makes a system slow but also inefficient.

Our project is currently on track with the exception of a couple of problems that were encountered. The first is that we have not yet figured out how to increase accuracy of face recognition. Implementation of this function will require further research of image processing and face recognition modules.

The second problem is determining the reason why we can only apply frontal face detection. To troubleshoot this problem we must look over the code more thoroughly. We are still confident that we will be able to find solutions to these problems, so we expect not to have to compromise on our initial project goal.

8 FUTURE SCOPE

The possible future scope of this application will be on mobile platform with following enhancements:

- Create a Self API and provide a better user interface.
- Apply machine learning for better face detection or accuracy.
- Using Maximum thresholds value so blur face also detect with accuracy.
- Working on making the system as low requirement storage wise and processing wise.

9 REFERENCES

1. https://www.tutorialspoint.com/python/python_modules.htm
2. https://www.tutorialspoint.com/python/python_environment.htm
3. https://pythonprogramming.net/loading-images-python-opencv-tutorial/
4. https://www.kairos.com/docs/getting-started-with-kairos-face-recognition
5. https://www.kairos.com/docs/api/humanface/application/appkey
6. http://flask.pocoo.org/
7. https://www.tutorialspoint.com/flask/flask_redirect_and_errors.htm
8. http://flask.pocoo.org/docs/1.0/design/
9. https://www.w3schools.com/html/html_attributes.asp
10. https://www.w3schools.com/html/html_images.asp
11. https://www.tutorialspoint.com/javascript/javascript_functions.htm
12. https://www.w3schools.com/html/html_tables.asp
13. https://www.w3schools.com/css/css_background.asp