**ATTENDANCE SYSTEM USING FACIAL RECOGNITION**

**A CAPSTONE PROJECT REPORT**

*Submitted in partial fulfillment of the*

*requirement for the award of the*

*Degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING**

*by*

**ITTA SAI ABHISHEK (18BCN7116)**

*Under the Guidance of*

**DR. PRIYADHARSHINI**



SCHOOL OF COMPUTER SCIENCE ENGINEERING

VIT-AP UNIVERSITY

AMARAVATI- 522237

*JANUARY 2022*

**CERTIFICATE**

This is to certify that the Capstone Project work titled “**ATTENDANCE SYSTEM USING FACIAL RECOGNITION**” that is being submitted by **I SAI ABHISHEK (18BCN7116)** is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Dr. PRIYADHARSHINI

Guide

**The thesis is satisfactory / unsatisfactory**

**Internal Examiner External Examiner**

**Approved by**

**PROGRAM CHAIR DEAN**

B. Tech. ECE School of Computer Science Engineering

**ACKNOWLEDGEMENTS**

**ABSTRACT**

In this digital age, where simple everyday life tasks are being automated one of such task ”Attendance marking” is still being taken manually. There should be a change in this absence to be faster and more efficient over time. We can use facial recognition to record the presence of a person. In this facial recognition, many algorithms were developed to provide recognition feature by capture images of someone's face. With the help of these algorithms the system can detect a person's face and record the presence of that person and can be more efficient and faster method.

Face recognition is among the most productive image processing applications and has a pivotal role in the technical field. Recognition of the human face is an active issue for authentication purposes specifically in the context of attendance of students. Attendance system using face recognition is a procedure of recognizing students by using face biostatistics based on the high-definition monitoring and other computer technologies. The development of this system is aimed to accomplish digitization of the traditional system of taking attendance by calling names and maintaining pen-paper records. Present strategies for taking attendance are tedious and time-consuming. Attendance records can be easily manipulated by manual recording. The traditional process of making attendance and present biometric systems are vulnerable to proxies. This paper is therefore proposed to tackle all these problems. The proposed system makes the use of Haar classifiers, KNN, CNN, SVM, Generative adversarial networks, and Gabor filters. After face recognition attendance reports will be generated and stored in excel format. The system is tested under various conditions like illumination, head movements, the variation of distance between the student and cameras. After vigorous testing overall complexity and accuracy are calculated. The Proposed system proved to be an efficient and robust device for taking attendance in a classroom without any time consumption and manual work. The system developed is cost-efficient and need less installation.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Chapter** | **Title** | **Page Number** |
| **1.** |  | **Acknowledgement** | **3** |
| **2.** |  | **Abstract** | **4** |
| **3.** |  | **List of Figures and Table** | **6** |
| **4.** | **1**  **1.1**  **1.2**  **1.3** | **Introduction**  **Objectives**  **Background and Literature Survey**  **Organization of the Report** | **8**  **9**  **10**  **11** |
| **5.** | **2**  **2.1**  **2.2**  **2.3**  **2.4**  **2.4.1**  **2.4.2** | **Vehicle Service Management and Live Monitoring System**  **Proposed System**  **Working Methodology**  **Standards**  **System Details**  **Software**  **Hardware** | **12**  **12**  **13**  **13**  **14**  **14**  **25** |
| **6.** | **3**  **3.1** | **Cost Analysis**  **List of components and their cost** | **37**  **37** |
| **7.** | **4** | **Results and Discussion** | **38** |
| **8.** | **5** | **Conclusion & Future Works** | **40** |
| **9.** | **6** | **Appendix** | **41** |
| **10.** | **7** | **References** | **50** |

**List of Tables**

**Table No. Title Page No.**

1. Cost Analysis 37

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 1 | Automobile Sales | 8 |
| 2 | System Block Diagram | 12 |
| 3 | Screen 1 | 15 |
| 4 | Screen 2 | 15 |
| 5 | Block Editor for Screen 1 | 16 |
| 6 | Block Editor for Screen 2 | 16 |
| 8  9.a  9.b  10  11    12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | Home Page  Main Page  Manage Automobile Service  Manage Automobile Service  Edit Profile  Firebase Authentication Key  Firebase Console  Garage Owner Main Page  Live Streaming Setting  Live Monitoring  Link for Live Video  Raspberry Pi  Copying to Sim Card  Raspberry Pi  Powering Raspberry Pi  Raspbian Desktop  Raspberry Pi Welcome Screen  VNC Viewer  VNC Viewer Address  VNC Server  VNC Properties Window  Arduino  Hall Effect Module  Linear Potentiometer  SIM 808  3- Axis Accelerometer  LM35  Voltage Detection Sensor  Hardware Setup  R Pi Receiving data  R Pi Receiving data | 17  18  19  19    20  21  21  22  23  23  24  25  26  27  27  28  28  29  31  31  32  33  33  34  34  35  36  36  38  39  39 |

**CHAPTER 1**

**INTRODUCTION**

A facial recognition system is a technology that can detect, and match a given human face from a digital image or a video against a database of faces stored in xml or json formats. It works by locating and measuring facial features such as distance between both the eyes, width of nose etc. from a given image and is commonly used to authenticate a person’s physical presence.

A picture containing shape

Description automatically generated

Figure 1 Measuring Facial Features

In just a few years, facial recognition technology has advanced considerably. According to tests conducted by the National Institute of Standards and Technology(NIST), the best face identification algorithm has an error rate of under 0.08 percent as of April 2020, compared to 4.1 percent for the major used algorithm in 2014.

Keeping a track of attendance is crucial for both teachers and students in a learning environment. As a result, marking attendance on daily basis manually is a hectic task. Also, the bio-metric systems which are more popularly used to avoid proxy consume a lot of time to mark attendance with huge class strengths. As a result, an automatic attendance system which can mark attendance by detecting the facial features can alleviate all these concerns.

* 1. **Objectives**

The following are the objectives of this project:

* Reducing time wastage during conventional class attendance marking methods.
* Automating the entire attendance system functionalities which requires minimal monitoring and has complete digital environment.
* Evidently narrowing the possibilities for marking proxy attendance.
* Encouraging the use of latest technology trends in daily lives.
  1. **Background and Literature Survey**

The process flow of a face detection and identification system starts with the ability to identify facial characteristics from a live or an image stored in the memory. The system analyses the taken image and determines the number of faces in the image using various system learnt patterns to filter out and compare them. This image processing employs several algorithms that consider face expressions and various facial structures and dimensions and compare them to a database of known features.

* 1. **Previous Work**

This is a project done by students as a final year project at University Tunku in 2018 The approach performs face recognition-based student attendance system. This method is also like others and begins with the input of an image either loaded from memory or from camera. Then it pre-processes the facial features and extracts it followed by subjective selecting and then the recognition of the facial images from known database. Both LBP and PCA feature extraction methods are studied in detail and computed in this approach to make comparisons. LBP is enhanced in this approach to reduce the illumination effect. An algorithm to combine enhanced LBP and PCA is also designed for subjective selection to increase the accuracy.

**Diagram

Description automatically generated**

Figure 2 Process flow Diagram of Previous Work

**1.4 Organization of the Report**

The remaining chapters of the project report are described as follows:

* Chapter 2 contains the proposed system, methodology, hardware, and software details.
* Chapter 3 gives the cost involved in the implementation of the project.
* Chapter 4 discusses the results obtained after the project was implemented.
* Chapter 5 concludes the report.
* Chapter 6 consists of codes.
* Chapter 7 gives references.

**CHAPTER 2**

**ATTENDENCE MANAGEMENT SYSTEM**

This Chapter describes the proposed system, working methodology, software, and hardware details.

**2.1 Proposed System**

The following block diagram (figure 2) shows the system architecture of this project.

Diagram

Description automatically generated

Figure 2 System Block Diagram

**2.2 Working Methodology**

The system has two sections, python-based GUI System with image recording and attendance posting options and an online web-portal for attendance view option. Both the GUI System and web portal are linked to a SQL Server where the data will be stored and processed.

Python-based GUI System is a software application that is installed on the system admin and class handing faculty systems. This GUI System allows admin to take new images of students for new registrations or delete the images of past students and re-train the existing model accordingly. Faculty on the other hand can start taking the attendance of the class automatically by triggering the system to switch on the cameras and predict the faces recorded and mark the attendance for the same. The attendance marked is stored in the local memory of faculty system and faculty can view/edit the stored data and post the marking into the database after verification.

Online Web-Portal allows students to login and view all their attendance markings. Also, faculty and admin later can login to their respective account and can view/add/edit/delete any record as per need on the website

Diagram

Description automatically generated

Figure 2 System Workflow Diagram

**2.3 Software Details**

This Section describes the software applications used for the development of system

**2.3.1 Tkinter GUI**

Tkinter (TK Interface) is a standard cross-platform package for implementing graphical user interface (GUI) using python programming language. It consists of all the required libraries as a single package and can be also used additionally with the other libraries as per requirement.

Importing Tkinter in PYTHON Ver 2.x and above

> import Tkinter as tk

> import tkFileDialog as filedialog

> import ttk

**2.3.2 OPENCV**

OpenCV is a tool for image processing and computer vision applications. It is an open-source library for tasks including face identification, objection tracking, landmark detection, and many more image recognition applications. Python, Java, and C++ are among the popular languages supported.

Installing OpenCV for WINDOWS

> pip install opencv-python

> import cv2 as cv

**2.3.3 HaarCascade Classifier**

HaarCascade Classifier is an effective approach for object detection which was proposed by Paul Viola and Michael Jones through their paper “Rapid Object Detection Using a Boosted Cascade of Simple Features” in the year 2001. This is classifier basically a machine learning based approach in which a group similar and non-similar kind of images are used to train a cascade function through which new images are detected by pattern matching. There are many huge individuals of .xml files with a lot of feature sets and each xml corresponds to a very specific type of use case or pattern to be verified. As we need to identify the facial features and detect the person in this project we will be using “**haarcascade\_frontalface\_default.xml” for the same.**

**2.3.4 XAMPP Server**

XAMPP is a popular cross-platform web server that allows programmers to write and test their code on a local web server. XAMPP is abbreviated as follows, X stands for Cross-Platform, A stands for Apache, M stands for [MY-SQL](https://www.javatpoint.com/mysql-tutorial), and the two P‘s stand for PHP and Perl respectively. It is an open-source web solution package that incorporates Apache distribution for servers and command-line executables with modules such as Apache server, [MariaDB](https://www.javatpoint.com/mariadb-tutorial), PHP, and Perl. In the current project we will be using PHP for the server-side programming and MYSQL as the database.

Graphical user interface, application

Description automatically generated

**Figure XAMPP Control Panel**

**2.4 Face Detection**

Detailed explanation of functioning of facial recognition system is done in this section

**2.4.1 LBPH Algorithm**

LBPH (Local Binary Pattern Histogram) is a Facial Detection algorithm used to identify the individual using his/her facial features. Every image is represented in the form of a matrix composed of rows and columns combining to form many individual cells. These individual cells are called as pixels. A single pixel is normally defined as least possible information of a bit of image. For every pixel in an image the value ranges from 0 to 255. Consider a 3X3 pixel matrix from a given image. We have a total of 9 cells in this matrix. We use LBPH algorithm to define a value to the mid cell with respective to the 8 cells around it.

Diagram

Description automatically generated

**Figure pixel value generation using LBPH**

For each of the neighboring cell of the central cell, we allot a new one-bit binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold. And we find the Threshold(Z) of the central cell using the formula below

Z=∑ s(In - Ic)2^n { Z >= 9 🡪 I = 1 ; Z < 9 🡪 I = 0 }

In the above formula In 🡪 pixel value of the nth cell , Ic 🡪 pixel value of the central cell

Once the one-bit binary values are set we arrange them in the binary form of i0i1i2…in and the decimal equivalence of this binary is the pixel value of the central cell. After this process(LBP procedure), we have a newly formed image which represents better the characteristics of the original image with each pixel more related to each other.

Now, using the newly generated image in the previous, we can use the X and Y grid parameters to further divide the image into multiple grids and form histograms respectively as shown in the image below.Diagram

Description automatically generated

**Figure Conversion of image to histogram**

Once the histograms are formed the data is stored and to check similarity between two images, we later compare the two histograms formed of respective images and the one close distance has high similarity. In this project we will be using Euclidean Distance to compare the histograms.

Text

Description automatically generated

**2.4.2 Training the Model**

We will be using **haarcascade\_frontalface\_default.xml to detect the face of individuals through the given image. To train the model all the images taken must be scaled down to uniform dimensions and can be done using the code below.**

import cv2

i = cv2.imread('/home/img/python.png', cv2.IMREAD\_UNCHANGED)

print('Original Dimensions : ',img.shape)

sp= 60 # percent of original size

width = int(i.shape[1]\*sp/100)

height = int(i.shape[0]\*sp/100)

dim = (width, height)

resized = cv2.resize(img, dim, interpolation = cv2.INTER\_AREA)

print('Resized Dimensions : ',resized.shape)

**Once we scale down all the images to uniform dimensions, we now use Cascade Classifier function of OpenCV to point to the path of the haarcascade\_frontal\_default.xml in the code below.**

harcascadePath = "haarcascade\_frontalface\_default.xml"

detector = cv2.CascadeClassifier(harcascadePath)

**Now we load all the images and convert them into grayscale. We perform this operation to remove the RGB (Red, Green, Blue) format in the image and bring down all the computations to 256 bits i.e., Black and White**

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

**Once the images are converted to greyscale from RGB format we try to collect all the facial features in the images using an inbuilt method called detectMultiScale as below.**

faces = detector.detectMultiScale(gray, 1.3, 5)

In the above code we are trying to use face\_classifier which is an object that is loaded with haarcascade\_frontal\_default.xml file. This function detects the facial features and record in the object assigned to.

The object from the above step (i.e., faces) in our code returns four values and they are X-Coordinate, Y-Coordinate, Width, Height of the detected face and store the processed images in the memory using below code.

for (x, y, w, h) in faces:

    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

    sampleNum = sampleNum + 1

  cv2.imwrite("TrainingImage\"+Id+'.'+str(sampleNum)+".jpg",gray[y:y+h,x:x +w])

    cv2.imshow('Taking Images', img)

    if cv2.waitKey(100) & 0xFF == ord('q'):

        break

    elif sampleNum > 100:

        break

Once the images are stored we start training the model using recognizer method and save the trained data in the format of .yml using below code.

recognizer = cv2.face\_LBPHFaceRecognizer.create()

    harcascadePath = "haarcascade\_frontalface\_default.xml"

    detector = cv2.CascadeClassifier(harcascadePath)

    faces, ID = getImagesAndLabels("TrainingImage")

    try:

        recognizer.train(faces, np.array(ID))

    except:

        print(Exception while training model!)

        return

    recognizer.save("TrainingImageLabel\Trainner.yml")

On successful execution of above code, the recorded images are processed, trained and model is saved as Trainner.yml on the local drive.

**2.4.3 Recognition of the Image**

Once the model is trained and saved in the format of .yml successfully we proceed with tracking (detection) of faces in the new images given to the system. We use read() inbuilt method with the recognizer object to read the .yml data and haarcascade\_frontalface\_default.xml file to extract new face from new image.

recognizer.read("TrainingImageLabel\Trainner.yml")

harcascadePath = "haarcascade\_frontalface\_default.xml"

faceCascade = cv2.CascadeClassifier(harcascadePath)

Later on we compare two images using predict() inbuilt method with the recognizer object to get the confidence level(similarity level) of the two files and give a threshold limit such that all the images above this threshold will be accepted to be similar and below will be rejected.

gray = cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

            faces = faceCascade.detectMultiScale(gray, 1.2, 5)

            for (x, y, w, h) in faces:

                cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)

                serial, conf = recognizer.predict(gray[y:y + h, x:x + w])

                if (conf > 50):

print(Accepted)

**2.5 Python GUI System**

Functionalities of the GUI System are explained in this section

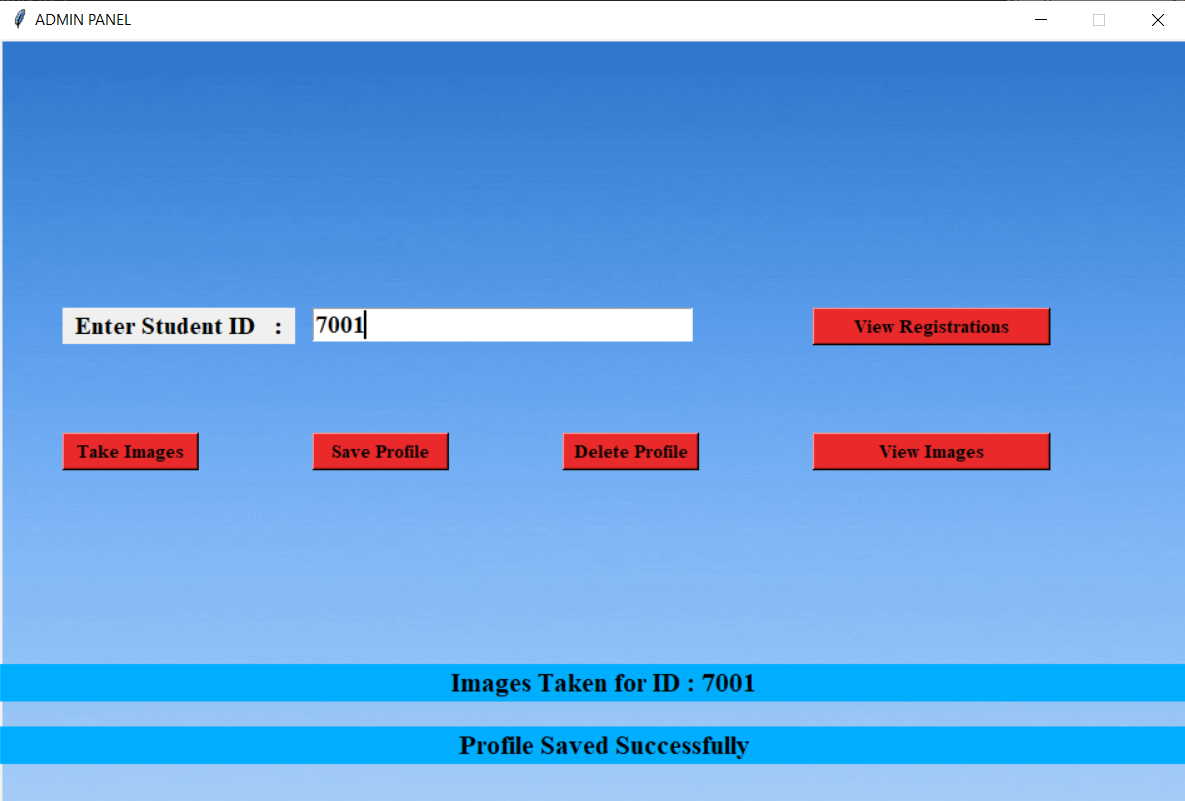
The entire graphical user interface in this project is developed using TKinter GUI Package Libraries. The first page of the system is the login page where user needs to enter his/her employee ID and password and login to their respective profiles. There are two profiles in the system, and they are Admin and Faculty. Each profile has their own functionalities in them respective to their requirements. Detailed explanation of each profile and their functionalities is given below



**Figure GUI Login Panel**

**2.5.1 Admin Panel**

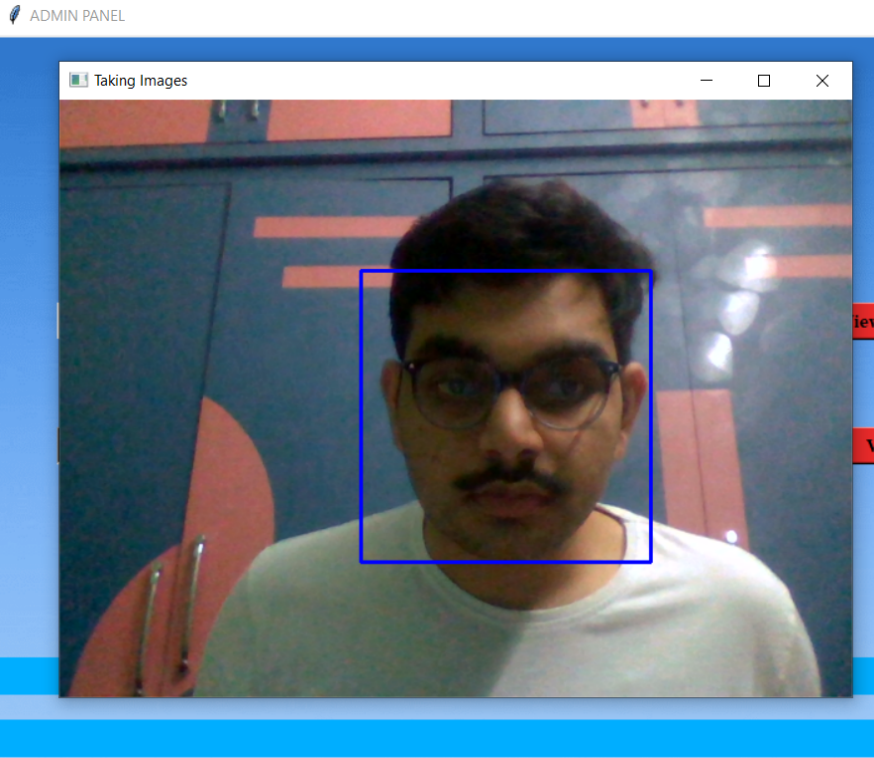
The system admin is solely responsible for maintaining class records and updating the trained model with any new changes such as addition/deletion of student images in the system. Admin can also access the images of individual student and entire registration data and trained data (.yml file) to process any necessary corrections in the system.

****

**Figure GUI Admin Panel**

Admin can enter the student ID and check if his/her images are registered and can also delete or add new registrations by using delete profile or take images option and later saving the new profile. Every time admin saves a profile the current existing .yml file gets deleted and new yml file will be generated with the current images available in the system.

To add new student details to the system Admin first selects Take Images option. A camera appears with a blue square around the detected faces. The system is now tracking the images and storing them locally.

****

**Figure New Profile Registration**

Admin can manually stop image recording by clicking on ‘q’ key on the keyboard or the system will automatically shut the camera after certain predefined time and saves the images. Once images are recorded successfully, we can proceed with the profile saving option which overrides the current existing .yml file with newly processed data.

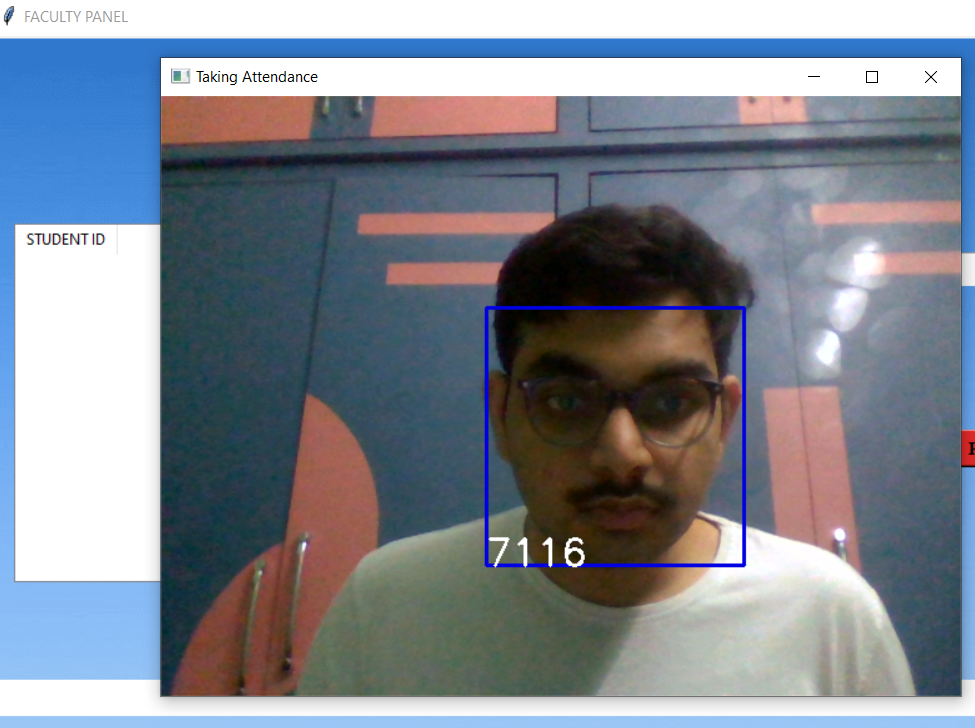
**2.5.2 Faculty panel**

Once logged into the system faculty will be directed to a panel as shown in the figure below.

****

**Figure Faculty Panel**

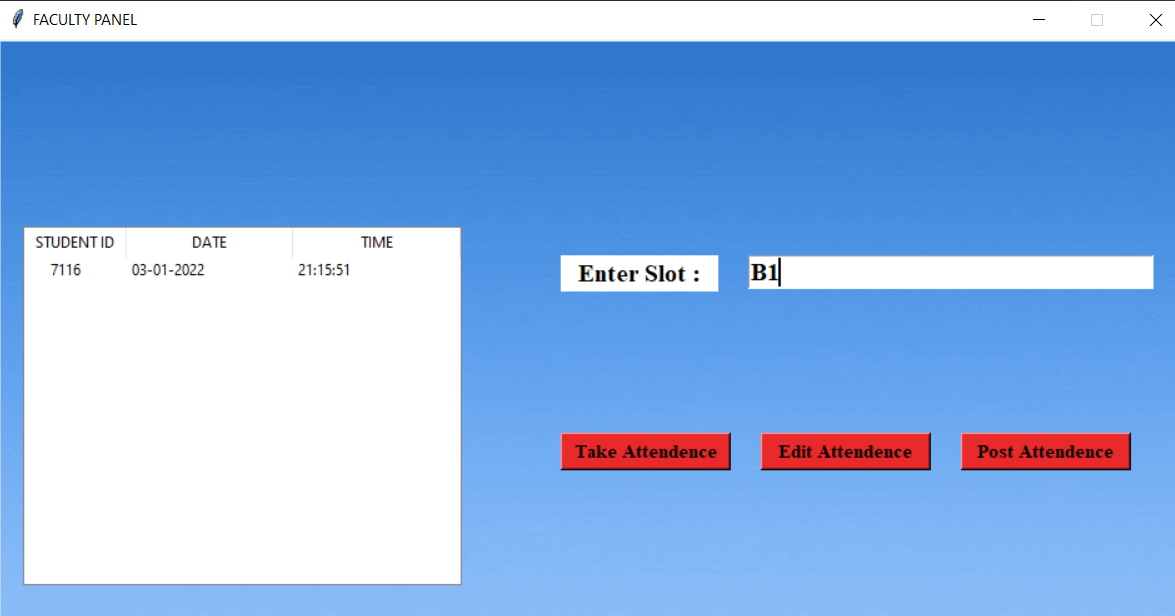
Faculty need to enter the slot code of the class he/she is wishing to proceed with attendance marking and select the option Take Attendance which switches the cameras on and detects the human face.



**Figure Attendance Tracking**

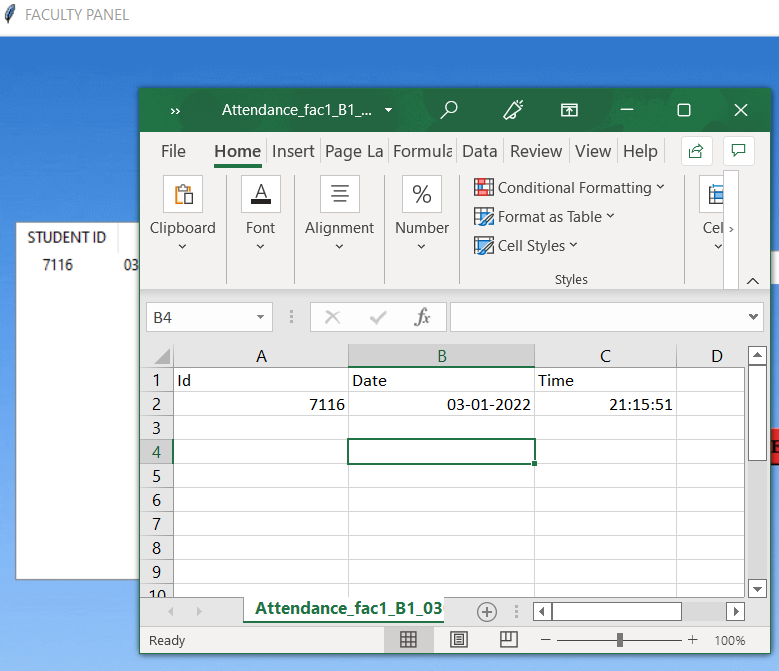
The ID of the student will be displayed at the right bottom of the blue square box if system could successfully recognize the person in the image. Faculty can use this option to check if the system is successful in recognition of student. And once the attendance marking is done faculty can close the camera by clicking ‘q’ key on the keyboard

Once faculty finishes the attendance tracking the marked attendance with data and time will be saved in the faculty local drive in the excel format and will also be displayed in the faculty panel as shown in the figure below



**Figure Marked Attendance View**

If there is any error to be rectified faculty can use Edit Attendance option which will open the excel sheet in which attendance was recorded. Faculty can manually correct any errors that occurred here. Faculty can also manually enter the class attendance in the excel sheet if automatic attendance does not work due to any error in the system.



**Figure Attendance Edit Option**

And once faculty verified the marked attendance, they can use Post Attendance option to post the recorded attendance to the SQL Database. Also, the recorded attendance is stored locally on the faculty system in the excel format which allows faculty to post data whenever they are connected to the server without any data loss.

**2.6 Database**

In this project we will be using XAMPP Server for the backend and MYSQL for the database. We are going to implement three tables called ‘adminlogin’, ‘facultylogin’, ‘studentlogin’ which stores the username and password for the respective accounts. To Store attendance records we are going to use three main tables namely ‘atd’,‘faculty\_slot’,‘std\_registration’ and the entity relationship diagram is given below for these tables

Diagram

Description automatically generated

**Figure ER Diagram**

As faculty login with his/her id and choose a slot listed in the faculty\_slot table and post the recorded attendance the data will be subsequently posted into the ‘atd’ table with respective facid(faculty id), stdid(student id), slot, date, time. Later students can view his/her respective slot wise attendance details posted to the database on the web portal by logging into their respective account.

**2.7 Web Portal**

The online web portal can be accessed by students to view all their respective class attendance records and faculty/admin have access to their class, attendance records and can also add/delete a student record into the class attendance later using the portal.

Graphical user interface, text, application

Description automatically generated

**Figure Portal Login Page**

**2.7.1 Faculty Portal**

As faculty login to the portal they will be redirected to faculty home page where all the courses they are handling will be displayed with an action button called ‘process’ as shown in the figure below.

**Graphical user interface, application, Word

Description automatically generated**

**Figure Faculty Home Page**

Faculty can later on choose a particular class in their course list and they will be redirected to class detail’s view page where they can choose date and view the students present in the class on that particular date. Faculty can also add new record to the class attendance by selecting the date and entering student id in the respective field and clicking on Add Record button. Also, for each individual record in the attendance view table there is an action button called Delete which will delete that record from the data. Faculty can use this option to delete any attendance record of a student from the class.

**Graphical user interface

Description automatically generated with low confidence**

**Figure Faculty Slot View**

**2.7.2 Student Portal**

Once student’s login to the portal they will be navigated to students home page where they can find all the classes they are registered in as shown in the figure below.

**Graphical user interface, application

Description automatically generated**

**Figure Student Home Page**

Students can later on choose a particular class to view attendance and they will be navigated to attendance view page where the date and time of the attendance markings will be displayed in order according to slot they have selected.

**Table

Description automatically generated**

**Figure Student Slot View**

**CHAPTER 3**

**COST ANALYSIS**

**3.1 List of components and their costs**

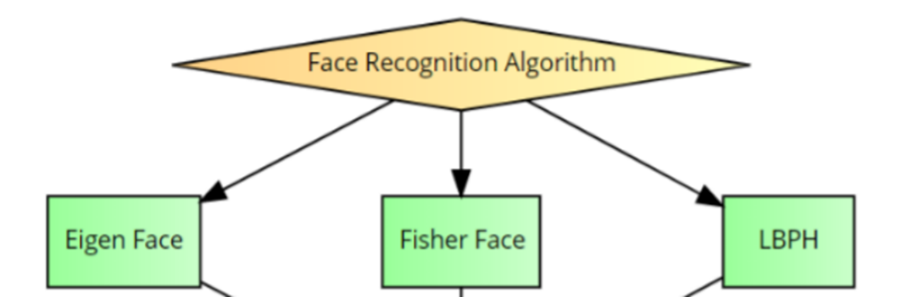
As most of the project components are based on open-source software they can be downloaded for free on the internet. The costs of the various components used in this project are given below in Table 3.1.

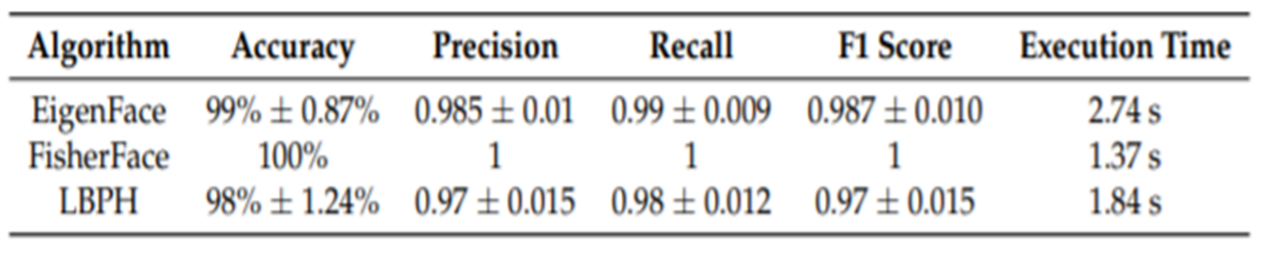
**Table 3.1 List of components and their costs**

|  |  |
| --- | --- |
| **COMPONENT** | **COST** |
| Python | ₹ 0 |
| TKinter GUI | ₹ 0 |
| OpenCV | ₹ 0 |
| XAMPP Server | ₹ 0 |
| USB Supported Web-Camera | ₹ 700 |
| TOTAL | ₹ 700 |

**CHAPTER 4**

**RESULTS AND DISCUSSIONS**

****

****

**CHAPTER 5**

**CONCLUSION AND FUTURE WORK**

**CHAPTER 6**

**APPENDIX**

**Python GUI Code**

from io import FileIO

import tkinter as tk

from tkinter import Label, PhotoImage, Tk, ttk

from tkinter import messagebox as mess

import tkinter.simpledialog as tsd

import cv2,ossni

import csv

import numpy as np

from PIL import Image

import pandas as pd

import datetime

import time

import os, glob

import mysql.connector

import MySQLdb

global key

key = ''

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')

day,month,year=date.split("-")

mont={'01':'January',

'02':'February',

'03':'March',

'04':'April',

'05':'May',

'06':'June',

'07':'July',

'08':'August',

'09':'September',

'10':'October',

'11':'November',

'12':'December'

}

def check\_haarcascadefile():

exists = os.path.isfile("haarcascade\_frontalface\_default.xml")

if exists:

pass

else:

mess.\_show(title='File Not Found', message='Cascade File Not Found!!!')

window.destroy()

def assure\_path\_exists(path):

dir = os.path.dirname(path)

if not os.path.exists(dir):

os.makedirs(dir)

def getImagesAndLabels(path):

# get the path of all the files in the folder

imagePaths = [os.path.join(path, f) for f in os.listdir(path)]

# create empth face list

faces = []

# create empty ID list

Ids = []

# now looping through all the image paths and loading the Ids and the images

for imagePath in imagePaths:

# loading the image and converting it to gray scale

pilImage = Image.open(imagePath).convert('L')

# Now we are converting the PIL image into numpy array

imageNp = np.array(pilImage, 'uint8')

# getting the Id from the image

print(imagePath)

print(os.path.split(imagePath)[-1])

print(os.path.split(imagePath)[-1].split(".")[0])

ID = int(os.path.split(imagePath)[-1].split(".")[0])

# extract the face from the training image sample

faces.append(imageNp)

Ids.append(ID)

return faces, Ids

def TrainImages():

check\_haarcascadefile()

assure\_path\_exists("TrainingImageLabel/")

recognizer = cv2.face\_LBPHFaceRecognizer.create()

harcascadePath = "haarcascade\_frontalface\_default.xml"

detector = cv2.CascadeClassifier(harcascadePath)

faces, ID = getImagesAndLabels("TrainingImage")

try:

recognizer.train(faces, np.array(ID))

print(ID)

except:

#mess.\_show(title='No New Images Found', message='')

adminpanel.message.configure(text='Please Take Images to proceed with profile saving!')

return

recognizer.save("TrainingImageLabel\Trainner.yml")

res = "Profile Saved Successfully"

adminpanel.message1.configure(text=res)

def TakeImages():

check\_haarcascadefile()

columns = ['Student ID']

assure\_path\_exists("StudentDetails/")

assure\_path\_exists("TrainingImage/")

serial = 0

exists = os.path.isfile("StudentDetails\StudentDetails.csv")

if (exists==False):

with open("StudentDetails\StudentDetails.csv", 'a+',newline="") as csvFile1:

writer = csv.writer(csvFile1)

writer.writerow(columns)

csvFile1.close()

Id = (adminpanel.studentid.get())

if (Id!=''):

cam = cv2.VideoCapture(0)

harcascadePath = "haarcascade\_frontalface\_default.xml"

detector = cv2.CascadeClassifier(harcascadePath)

sampleNum = 0

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)

sampleNum = sampleNum + 1

cv2.imwrite("TrainingImage\"+ Id + '.' +str(sampleNum) +".jpg",gray[y:y+h, x:x+w])

cv2.imshow('Taking Images', img)

if cv2.waitKey(100) & 0xFF == ord('q'):

break

elif sampleNum > 100:

break

cam.release()

cv2.destroyAllWindows()

res = "Images Taken for ID : " + Id

row = [Id]

with open('StudentDetails\StudentDetails.csv', 'a+',newline="") as csvFile:

writer = csv.writer(csvFile)

writer.writerow(row)

csvFile.close()

adminpanel.message.configure(text=res)

else:

res = "Enter Correct Student ID!"

adminpanel.message.configure(text=res)

def adminpanel():

if(tkpassword.get()=="" or tkusername.get()=="" ):

tklblerror.config(text="Enter Credentials!")

print("Admin Credentials not entered to login")

import mysql.connector

mydb = MySQLdb.connect(host='localhost',user='root',passwd='',db='capstoneatd')

mycursor = mydb.cursor()

mycursor.execute("SELECT \* FROM adminlogin where dbadminid='"+tkusername.get()+"' and dbadmpassword='"+tkpassword.get()+"'")

myresult = mycursor.fetchall()

y=0

for x in myresult:

y=y+1

if(y==1):

window.destroy()

admin=tk.Tk()

admin.geometry("950x620")

admin.resizable(False,False)

bg = PhotoImage(file = "C:/xampp/htdocs/attend - Copy/bg.png")

label1 = Label( admin, image = bg)

label1.place(x = 0, y = 0)

admin.title("ADMIN PANEL")

admin.configure(background='#262523')

tklblstuser = tk.Label(admin, text="Enter Student ID :", width=15, fg="black", height=1, font=('times', 15, ' bold '))

tklblstuser.place(x=50, y=215)

adminpanel.studentid = tk.Entry(admin,width=30 ,fg="black",font=('times', 15, ' bold '))

adminpanel.studentid.place(x=250, y=215)

tknewregistrationbtn = tk.Button(admin, text="Take Images",command=TakeImages,fg="black" ,bg="#ea2a2a" ,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))

tknewregistrationbtn.place(x=50, y=315)

tksavebtn = tk.Button(admin, text="Save Profile",command=TrainImages,fg="black" ,bg="#ea2a2a" ,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))

tksavebtn.place(x=250, y=315)

tkdeletebtn = tk.Button(admin, text="Delete Profile",command=deleteprofile,fg="black" ,bg="#ea2a2a" ,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))

tkdeletebtn.place(x=450, y=315)

tkdviewstdsbtn = tk.Button(admin, text="View Registrations",command=viewregistrations,fg="black" ,bg="#ea2a2a" ,width=20 ,activebackground = "white" ,font=('times', 11, ' bold '))

tkdviewstdsbtn.place(x=650, y=215)

tkviewimagesbtn = tk.Button(admin, text="View Images",command=viewimages,fg="black" ,bg="#ea2a2a" ,width=20 ,activebackground = "white" ,font=('times', 11, ' bold '))

tkviewimagesbtn.place(x=650, y=315)

adminpanel.message = tk.Label(admin, text="" ,bg="#00aeff" ,fg="black" ,width=80,height=1, activebackground = "yellow" ,font=('times', 16, ' bold '))

adminpanel.message.place(x=0, y=500)

adminpanel.message1 = tk.Label(admin, text="" ,bg="#00aeff" ,fg="black" ,width=80,height=1, activebackground = "yellow" ,font=('times', 16, ' bold '))

adminpanel.message1.place(x=0, y=550)

admin.mainloop()

else:

tklblerror.config(text="ID or Password not found for Admin")

print("Admin login failed")

def viewregistrations():

os.startfile("C:\\xampp\\htdocs\\attend - Copy\\StudentDetails\\StudentDetails.csv")

def viewimages():

os.startfile("C:\\xampp\\htdocs\\attend - Copy\\TrainingImage")

def deleteprofile():

if(adminpanel.studentid.get()==""):

adminpanel.message1.config(text="Enter Student ID to Delete Profile")

else:

for filename in glob.glob("C:/xampp/htdocs/attend - Copy/TrainingImage/ "+adminpanel.studentid.get()+"\*.jpg"):

os.remove(filename)

def facultypanel():

if(tkpassword.get()=="" or tkusername.get()=="" ):

tklblerror.config(text="Enter Credentials!")

print("Faculty Credentials not entered to login")

return

import mysql.connector

mydb = MySQLdb.connect(host='localhost',user='root',passwd='',db='capstoneatd')

mycursor = mydb.cursor()

mycursor.execute("SELECT \* FROM facultylogin where dbfacultyid='"+tkusername.get()+"' and dbfacpassword='"+tkpassword.get()+"'")

myresult = mycursor.fetchall()

y=0

for x in myresult:

y=y+1

if(y==1):

facultypanel.facultyid=tkusername.get()

window.destroy()

admin=tk.Tk()

admin.geometry("950x620")

admin.resizable(False,False)

bg = PhotoImage(file = "C:/xampp/htdocs/attend - Copy/bg.png")

label1 = Label( admin, image = bg)

label1.place(x = 0, y = 0)

admin.title("FACULTY PANEL")

admin.configure(background='#262523')

facultypanel.tv= ttk.Treeview(admin,height =13,columns = ('date','time'))

facultypanel.tv.column('#0',width=82)

#facultypanel.tv.place(x=300,y=100)

#facultypanel.tv.column('name',width=130)

facultypanel.tv.column('date',width=133)

facultypanel.tv.column('time',width=133)

facultypanel.tv.grid(row=2,column=0,padx=(20,20),pady=(150,0),columnspan=4)

facultypanel.tv.heading('#0',text ='STUDENT ID')

#facultypanel.tv.heading('name',text ='NAME')

facultypanel.tv.heading('date',text ='DATE')

facultypanel.tv.heading('time',text ='TIME')

facultypanel.tkslot = tk.Entry(admin,width=32 ,fg="black",font=('times', 15, ' bold ') )

facultypanel.tkslot.place(x=600, y=173)

facultypanel.slotlbl = tk.Label(admin, text="Enter Slot :",bg="white", width=10, fg="black", height=1, font=('times', 15, ' bold '))

facultypanel.slotlbl.place(x=450, y=173)

"""

facultypanel.daylbl = tk.Label(admin, text="Day :",bg="white", width=5, fg="black", height=1, font=('times', 15, ' bold '))

facultypanel.daylbl.place(x=450, y=223)

facultypanel.tkday = tk.Entry(admin,width=5 ,fg="black",font=('times', 15, ' bold ') )

facultypanel.tkday.place(x=520, y=223)

facultypanel.monthlbl = tk.Label(admin, text="Month :",bg="white", width=5, fg="black", height=1, font=('times', 15, ' bold '))

facultypanel.monthlbl.place(x=590, y=223)

facultypanel.tkmonth = tk.Entry(admin,width=5 ,fg="black",font=('times', 15, ' bold ') )

facultypanel.tkmonth.place(x=660, y=223)

facultypanel.tkyear = tk.Entry(admin,width=5,height=1 ,fg="black",font=('times', 15, ' bold ') )

facultypanel.tkyear.place(x=800, y=223)

facultypanel.yearlbl = tk.Label(admin, text="Year :",bg="white", width=5, fg="black", height=1, font=('times', 15, ' bold '))

facultypanel.yearlbl.place(x=730, y=223)

"""

tknewregistrationbtn = tk.Button(admin, text="Take Attendence",command=TrackImages,fg="black" ,bg="#ea2a2a" ,width=14 ,activebackground = "white" ,font=('times', 11, ' bold '))

tknewregistrationbtn.place(x=450, y=315)

tkeditbtn = tk.Button(admin, text="Edit Attendence",command=EditAttendance,fg="black" ,bg="#ea2a2a" ,width=14 ,activebackground = "white" ,font=('times', 11, ' bold '))

tkeditbtn.place(x=610, y=315)

tkpostbtn = tk.Button(admin, text="Post Attendence",command=postattd,fg="black" ,bg="#ea2a2a" ,width=14 ,activebackground = "white" ,font=('times', 11, ' bold '))

tkpostbtn.place(x=770, y=315)

facultypanel.fperror = tk.Label(admin, text="",bg="white", width=80, fg="red", height=1, font=('times', 15, ' bold '))

facultypanel.fperror.place(x=0, y=515)

admin.mainloop()

else:

tklblerror.config(text="ID or Password not found for Faculty")

print("Faculty login failed")

def EditAttendance():

if(facultypanel.tkslot.get()==""):

facultypanel.fperror.config(text="Enter Slot!")

else:

from datetime import date

today = date.today()

d1 = today.strftime("%d%m%Y")

attexists = os.path.isfile("C:\\xampp\\htdocs\\attend - Copy\\Attendance\\Attendance\_"+facultypanel.facultyid+"\_"+facultypanel.tkslot.get()+"\_"+d1+".csv")

if(attexists):

os.startfile("C:\\xampp\\htdocs\\attend - Copy\\Attendance\\Attendance\_"+facultypanel.facultyid+"\_"+facultypanel.tkslot.get()+"\_"+d1+".csv")

facultypanel.fperror.config(text="")

else:

facultypanel.fperror.config(text="Slot Not Found!")

def postattd():

import csv

import MySQLdb

mydb = MySQLdb.connect(host='localhost',user='root',passwd='',db='capstoneatd')

cursor = mydb.cursor()

from datetime import date

today = date.today()

d1 = today.strftime("%d%m%Y")

tbname="Attendance\_"+facultypanel.facultyid+"\_"+facultypanel.tkslot.get()+"\_"+d1

#cursor.execute("Create table "+tbname+" (id varchar(255),date varchar(255),time varchar(255));")

csv\_data = csv.reader(open("C:\\xampp\\htdocs\\attend - Copy\\Attendance\\"+tbname+".csv"))

for row in csv\_data:

print(row)

print(row[0]+row[1]+row[2])

print("INSERT INTO atd (facid,slot,stdid,date,time) VALUES('"+facultypanel.facultyid+"',"+facultypanel.tkslot.get()+",%s, %s, %s)")

if(row[0]!='Id'):

cursor.execute("INSERT INTO atd (facid,slot,stdid,date,time) VALUES('"+facultypanel.facultyid+"','"+facultypanel.tkslot.get()+"',%s, %s, %s)", row)

mydb.commit()

cursor.close()

print ("Done")

def TrackImages():

if(facultypanel.tkslot.get()==""):

facultypanel.fperror.config(text="Enter Slot!")

else:

print("into else")

check\_haarcascadefile()

assure\_path\_exists("Attendance/")

assure\_path\_exists("StudentDetails/")

for k in facultypanel.tv.get\_children():

facultypanel.tv.delete(k)

msg = ''

i = 0

j = 0

recognizer = cv2.face.LBPHFaceRecognizer\_create() # cv2.createLBPHFaceRecognizer()

exists3 = os.path.isfile("TrainingImageLabel\Trainner.yml")

if exists3:

recognizer.read("TrainingImageLabel\Trainner.yml")

else:

mess.\_show(title='.yml file not found ', message='Please contact admin.')

return

harcascadePath = "haarcascade\_frontalface\_default.xml"

faceCascade = cv2.CascadeClassifier(harcascadePath)

cam = cv2.VideoCapture(0)

font = cv2.FONT\_HERSHEY\_SIMPLEX

col\_names = ['Id','Date', 'Time']

exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")

if exists1:

df = pd.read\_csv("StudentDetails\StudentDetails.csv")

else:

mess.\_show(title='Student Details Missing in csv format', message='Students details are missing, please check!')

cam.release()

cv2.destroyAllWindows()

window.destroy()

nameList=[]

while True:

ret, im = cam.read()

gray = cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

faces = faceCascade.detectMultiScale(gray, 1.2, 5)

for (x, y, w, h) in faces:

cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)

serial, conf = recognizer.predict(gray[y:y + h, x:x + w])

#print(serial)

if (conf < 50):

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')

date1 = datetime.datetime.fromtimestamp(ts).strftime('%d%m%Y')

timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')

#aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values

ID = df.loc[df['Student ID'] == serial].values

ID = str(ID)

ID = ID[2:-2]

if str(ID) not in nameList:

nameList.append(str(ID))

attendance = [str(ID), str(date), str(timeStamp)]

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')

exists = os.path.isfile("Attendance\Attendance\_"+facultypanel.facultyid+"\_"+facultypanel.tkslot.get()+"\_"+ date1 + ".csv")

if exists:

with open("Attendance\Attendance\_"+facultypanel.facultyid+"\_"+facultypanel.tkslot.get()+"\_"+ date1 + ".csv", 'a+',newline='') as csvFile1:

writer = csv.writer(csvFile1)

writer.writerow(attendance)

csvFile1.close()

else:

with open("Attendance\Attendance\_"+facultypanel.facultyid+"\_"+facultypanel.tkslot.get()+"\_"+ date1 + ".csv", 'a+',newline='') as csvFile1:

writer = csv.writer(csvFile1)

writer.writerow(col\_names)

writer.writerow(attendance)

csvFile1.close()

else:

ID = 'Unknown'

cv2.putText(im, ID, (x, y + h), font, 1, (255, 255, 255), 2)

cv2.imshow('Taking Attendance', im)

if (cv2.waitKey(1) == ord('q')):

break

with open("Attendance\Attendance\_"+facultypanel.facultyid+"\_"+facultypanel.tkslot.get()+"\_"+ date1 + ".csv", 'r') as csvFile1:

reader1 = csv.reader(csvFile1)

for lines in reader1:

print(lines)

i = i + 1

if (i > 0):

if (i % 2 == 0):

iidd = str(lines[0]) + ' '

facultypanel.tv.insert('', 0, text=iidd, values=( str(lines[1]), str(lines[2])))

csvFile1.close()

cam.release()

cv2.destroyAllWindows()

def clearusername():

tkusername.delete(first=0,last=10)

def clearpassword():

tkpassword.delete(first=0,last=18)

def tick():

time\_string = time.strftime('%H:%M:%S')

clock.config(text=time\_string)

clock.after(200,tick)

window = tk.Tk()

window.geometry("950x620")

window.resizable(True,False)

window.title("Attendance System")

window.configure(background='#262523')

window.resizable(False,False)

bg = PhotoImage(file = "C:/xampp/htdocs/attend - Copy/bg.png")

label1 = Label( window, image = bg)

label1.place(x = 0, y = 0)

datef = tk.Label(window, text = day+"-"+mont[month]+"-"+year, fg="orange",bg="white" ,width=57 ,height=1,font=('times', 22, ' bold '))

datef.place(x=0,y=100)

clock = tk.Label(window,fg="orange",bg="white" ,width=57 ,height=1,font=('times', 22, ' bold '))

clock.place(x=0,y=140)

tick()

tklblusername = tk.Label(window, text="Enter Employee ID :", width=15, fg="black", height=1, font=('times', 15, ' bold '))

tklblusername.place(x=150, y=215)

tklblpassword = tk.Label(window, text="Enter Password :", width=15, fg="black", height=1, font=('times', 15, ' bold '))

tklblpassword.place(x=150, y=320)

tkusername = tk.Entry(window, width=20, bg="white", fg="red", font=('times', 25, ' bold '))

tkusername.place(x=400, y=210)

tkpassword = tk.Entry(window, width=20, bg="white", fg="red", font=('times', 25, ' bold '))

tkpassword.place(x=400, y=310)

tkclearusernamebtn = tk.Button(window, text="X", command=clearusername ,fg="red" ,bg="white" ,width=1 ,activebackground = "red" ,font=('times', 11, ' bold '))

tkclearusernamebtn.place(x=720, y=215)

tkclearusernamebtn = tk.Button(window, text="X", command=clearpassword ,fg="red" ,bg="white" ,width=1 ,activebackground = "red" ,font=('times', 11, ' bold '))

tkclearusernamebtn.place(x=720, y=315)

tkfacultyloginbtn = tk.Button(window, text="Faculty", command=facultypanel ,fg="black" ,bg="#ea2a2a" ,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))

tkfacultyloginbtn.place(x=585, y=415)

tkadminloginbtn = tk.Button(window, text="Admin", command=adminpanel ,fg="black" ,bg="#ea2a2a" ,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))

tkadminloginbtn.place(x=435, y=415)

tklblerror = tk.Label(window, text="", width=90, fg="red", height=1, font=('times', 15, ' bold '))

tklblerror.place(x=0, y=495)

window.mainloop()

**WEB Portal**

**signin.php**

<?php

$showAlert = false;

$showError = false;

$exists=false;

if($\_SERVER["REQUEST\_METHOD"] == "POST") {

include 'dbconnect.php';

$username = $\_POST["username"];

$password = $\_POST["password"];

$category = $\_POST["category"];

if($category=='student'){

$sql = "Select \* from studentlogin where dbstudentid='$username'and dbstdpassword='$password'";

$result = mysqli\_query($conn, $sql);

$num = mysqli\_num\_rows($result);

}

if($category=='admin'){

$sql = "Select \* from adminlogin where dbadminid='$username' and dbadmpassword='$password'";

$result = mysqli\_query($conn, $sql);

$num = mysqli\_num\_rows($result);

}

if($category=='faculty'){

$sql = "Select \* from facultylogin where dbfacultyid='$username'and dbfacpassword='$password'";

$result = mysqli\_query($conn, $sql);

$num = mysqli\_num\_rows($result);

}

if($num == 0) {echo"Credentials Not Found!";}

if($num>0)

{

session\_start();

$exists="Username not available";

$\_SESSION['userid']=$username;

$\_SESSION['category']=$category;

$\_SESSION['password']=$password;

if($category=='admin')

header("Location: http://localhost/attend%20-%20Copy/portal/admin.php", true, 301);

if($category=='faculty')

header("Location: http://localhost/attend%20-%20Copy/portal/faculty.php", true, 301);

if($category=='student')

header("Location: http://localhost/attend%20-%20Copy/portal/student.php", true, 301);

exit();

}

}

?>

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1,shrink-to-fit=no">

<link rel="stylesheet" href=

"https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"

integrity=

"sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk"

crossorigin="anonymous">

</head>

<body>

<?php

if($showAlert) {

echo ' <div class="alert alert-success

alert-dismissible fade show" role="alert">

<strong>Success!</strong> Your account is

now created and you can login.

<button type="button" class="close"

data-dismiss="alert" aria-label="Close">

<span aria-hidden="true">×</span>

</button>

</div> ';

}

if($showError) {

echo ' <div class="alert alert-danger

alert-dismissible fade show" role="alert">

<strong>Error!</strong> '. $showError.'

<button type="button" class="close"

data-dismiss="alert aria-label="Close">

<span aria-hidden="true">×</span>

</button>

</div> ';

}

if($exists) {

echo ' <div class="alert alert-danger

alert-dismissible fade show" role="alert">

<strong>Error!</strong> '. $exists.'

<button type="button" class="close"

data-dismiss="alert" aria-label="Close">

<span aria-hidden="true">×</span>

</button>

</div> ';

}

?>

<div class="container my-2 ">

<h1 class="text-center">Online Attendence Portal</h1>

<form action="signin.php" method="post">

<div class="form-group">

<label for="username">User ID :</label>

<input type="text" class="form-control" id="username"

name="username" aria-describedby="emailHelp" style="width:20%;">

</div>

<div class="form-group">

<label for="password">Password :</label>

<input type="password" class="form-control"

id="password" name="password" style="width:20%;">

</div>

<div class="form-group">

<label for="category">Login As :</label><br>

<!--<input type="radio" id="category1" name="category" value="admin">

<label for="category1">Admin&nbsp&nbsp</label>-->

<input type="radio" id="category2" name="category" value="faculty">

<label for="category2">Faculty&nbsp&nbsp</label>

<input type="radio" id="category3" name="category" value="student">

<label for="category3">Student</label><br>

</div>

<button type="submit" class="btn btn-primary">

Login In

</button>

</form>

</div>

<script src="

https://code.jquery.com/jquery-3.5.1.slim.min.js"

integrity="

sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"

crossorigin="anonymous">

</script>

<script src="

https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

integrity=

"sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"

crossorigin="anonymous">

</script>

<script src="

https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"

integrity=

"sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"

crossorigin="anonymous">

</script>

</body>

</html>

**dbconnect.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$database = "capstoneatd";

$conn = mysqli\_connect($servername,$username, $password, $database);

if($conn) {

#echo "success";

}

else {die("Error". mysqli\_connect\_error());}

?>

**faculty.php**

<?php

session\_start();

if(!isset($\_SESSION["userid"]))

header("Location: http://localhost/attend%20-%20Copy/portal/signin.php", true, 301);

#echo"home page ".$\_SESSION["userid"]." ".$\_SESSION['category'];

?>

<!DOCTYPE html>

<html>

<head>

<style>

table {

border-collapse: collapse;

width: 60%;

padding: 10%;

}

th, td {

text-align: left;

padding: 8px;

}

tr:nth-child(even){background-color: #f2f2f2}

th {

background-color: #157DEC;

color: white;

}

#head{

background-color: red;

}

</style>

<meta charset="utf-8">

<meta name="viewport" content=

"width=device-width, initial-scale=1,

shrink-to-fit=no">

<link rel="stylesheet" href=

"https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"

integrity=

"sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk"

crossorigin="anonymous">

<title>Faculty</title>

</head>

<body>

<div id="head">

<a style{align:left;} href="logout.php">Logout</a>

</div>

<div id="cont">

<center>

<h2>Your Courses</h2>

<table border="4">

<tr>

<th><b>Sr.No.</b></td>

<th><b>Slot</b></td>

<th><b>Course Name</b></td>

<th><b>Action</b></td>

</tr>

<?php

include "dbconnect.php"; // Using database connection file here

$id=$\_SESSION["userid"];

$records = mysqli\_query($conn,"select \* from faculty\_slot where facid='$id'"); // fetch data from database

$i=1;

while($data = mysqli\_fetch\_array($records))

{

?>

<tr>

<td><?php echo $i++; ?></td>

<td><?php echo $data['slot']; ?></td>

<td><?php echo $data['coursename']; ?></td>

<td><a href="fatdv.php?id=<?php echo $data['facid']."-".$data['slot']; ?>">Process</a></td>

</tr>

<?php

}

?>

</table>

</div>

</body>

</html>

**fatdv.php**

<?php

session\_start();

if(!isset($\_SESSION["userid"]))

header("Location: http://localhost/attend%20-%20Copy/portal/signin.php", true, 301);

if(isset($\_GET['id'])){

$cred=explode("-",$\_GET['id']);

$\_SESSION['facid']=$cred[0];

$\_SESSION['slot']=$cred[1];}

$date=0;

if($\_SERVER["REQUEST\_METHOD"] == "POST"){

if(isset($\_POST['viewbtn'])){

$date=$\_POST['date'];

$cred=explode("-",$date);

$date=$cred[2]."-".$cred[1]."-".$cred[0];}

if(isset($\_POST['addbtn']) && isset($\_POST['date']) ){

include "dbconnect.php";

$facid=$\_SESSION['facid'];$slot=$\_SESSION['slot'];

$date=$\_POST['date'];$stdid=$\_POST['stdid'];

$cred=explode("-",$date);

$date=$cred[2]."-".$cred[1]."-".$cred[0];

$sql = "INSERT INTO atd (facid, slot, stdid,date,time)

VALUES ('$facid', '$slot', '$stdid','$date','-')";

if ($conn->query($sql) === TRUE) {

#echo "New record created successfully";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

}

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Faculty</title>

<style>

table {

border-collapse: collapse;

width: 60%;

padding: 10%;

}

th, td {

text-align: left;

padding: 8px;

}

tr:nth-child(even){background-color: #f2f2f2}

th {

background-color: #157DEC;

color: white;

}

#head{

background-color: red;

}

#formid{

padding:20px;

}

</style>

<meta charset="utf-8">

<meta name="viewport" content=

"width=device-width, initial-scale=1,

shrink-to-fit=no">

<link rel="stylesheet" href=

"https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"

integrity=

"sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk"

crossorigin="anonymous">

</head>

<div id="head">

<a style{align:left;} href="logout.php">Logout</a>

</div>

<body>

<div id='formid'>

<form action="fatdv.php" method="post">

<h2><?php echo('Slot - '.$\_SESSION['slot']." ");?>Attendence Details</h2>

<label for="username">Choose Date :</label>

<input type="date" class="form-control" id="date"name="date" style="width:20%;"><br>

<input type="submit" name="viewbtn" value="View Attendence" class="btn btn-primary"><br><br>

<label for="username">Student ID &nbsp&nbsp :</label>

<input type="text" class="form-control" id="stdid"name="stdid" style="width:20%;"><br>

<input type="submit" name="addbtn" value="Add Record" class="btn btn-primary">

</form>

</div>

<center>

<br>

<table border="2">

<tr>

<th>Sr.No.</td>

<th>Student ID</td>

<th>Date</td>

<th>Time</td>

<th>Action</td>

</tr>

<?php

include "dbconnect.php";

$facid=$\_SESSION['facid'];$slot=$\_SESSION['slot'];

$records = mysqli\_query($conn,"select \* from atd where facid='$facid' and slot='$slot' and date='$date'");

$i=1;

while($data = mysqli\_fetch\_array($records))

{

?>

<tr>

<td><?php echo $i++; ?></td>

<td><?php echo $data['stdid']; ?></td>

<td><?php echo $data['date']; ?></td>

<td><?php echo $data['time']; ?></td>

<td><a href="fatdv.php?id=<?php echo $data['facid']."-".$data['slot']; ?>">Delete</a></td>

</tr>

<?php

}

?>

</table><br><br>

<a href="faculty.php">Back</a>

</body>

</html>

**student.php**

<?php

session\_start();

if(!isset($\_SESSION["userid"]))

header("Location: http://localhost/attend%20-%20Copy/portal/signin.php", true, 301);

echo"Logged in ".$\_SESSION["userid"];

?>

<!DOCTYPE html>

<html>

<style>

table {

border-collapse: collapse;

width: 60%;

padding: 10%;

}

th, td {

text-align: left;

padding: 8px;

}

tr:nth-child(even){background-color: #f2f2f2}

th {

background-color: #157DEC;

color: white;

}

#head{

background-color: red;

}

#formid{

padding:20px;

}

</style>

<meta charset="utf-8">

<meta name="viewport" content=

"width=device-width, initial-scale=1,

shrink-to-fit=no">

<link rel="stylesheet" href=

"https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"

integrity=

"sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk"

crossorigin="anonymous">

<head>

<title>Student</title>

</head>

<body>

<div id="head">

<a style{align:left;} href="logout.php">Logout</a>

</div>

<center>

<h2>Your Course Details</h2>

<table border="2">

<tr>

<td>Sr.No.</td>

<td>course title</td>

<td>Slot</td>

<td>Action</td>

</tr>

<?php

$id=$\_SESSION["userid"];

include "dbconnect.php"; // Using database connection file here

$records = mysqli\_query($conn,"select \* from std\_registration where stdid='$id'"); // fetch data from database

$i=1;

while($data = mysqli\_fetch\_array($records))

{

$clsid=$data['classid'];

$records1 = mysqli\_query($conn,"select \* from faculty\_slot where classid='$clsid'");

while($data1 = mysqli\_fetch\_array($records1)){

?>

<tr>

<td><?php echo $i++; ?></td>

<td><?php echo $data1['coursename'];?></td>

<td><?php echo $data1['slot']; ?></td>

<td><a href="satdv.php?id=<?php echo $data1['facid']."-".$data1['slot']; ?>">View Attendance</a></td>

</tr>

<?php

}

}

?>

</table>

<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.5.0/Chart.min.js"></script>

<canvas id="myChart" style="width:100%;max-width:600px"></canvas>

<script>

/\*var xValues = ["Italy", "France", "Spain", "USA", "Argentina"];

var yValues = [55, 49, 44, 24, 15];

var barColors = ["red", "green","blue","orange","brown"];

new Chart("myChart", {

type: "bar",

data: {

labels: xValues,

datasets: [{

// backgroundColor: barColors,

data: yValues

}]

},

options: {

legend: {display: false},

title: {

display: true,

text: "Attendence Percentage"

}

}

});\*/

</script>

</body>

</html>

**satdv.php**

<?php

session\_start();

if(isset($\_GET['id'])){

$cred=explode("-",$\_GET['id']);

$\_SESSION['facid']=$cred[0];

$\_SESSION['slot']=$cred[1];}

?>

<?php

if(!isset($\_SESSION["userid"]))

header("Location: http://localhost/attend%20-%20Copy/portal/signin.php", true, 301);

echo"Logged in ".$\_SESSION["userid"];

?>

<!DOCTYPE html>

<html>

<style>

table {

border-collapse: collapse;

width: 60%;

padding: 10%;

}

th, td {

text-align: left;

padding: 8px;

}

tr:nth-child(even){background-color: #f2f2f2}

th{

background-color: #157DEC;

color: white;

}

#head{

background-color: red;

}

#formid{

padding:20px;

}

</style>

<meta charset="utf-8">

<meta name="viewport" content=

"width=device-width, initial-scale=1,

shrink-to-fit=no">

<link rel="stylesheet" href=

"https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"

integrity=

"sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk"

crossorigin="anonymous">

<head>

<div id="head">

<a style{align:left;} href="logout.php">Logout</a>

</div>

<title>Attendance Details</title>

</head>

<body>

<center>

<h2><?php echo('Slot - '.$\_SESSION['slot']." ");?>Attendence Details</h2>

<br>

<table border="2">

<tr>

<td>Sr.No.</td>

<td>Date</td>

<td>Time</td>

</tr>

<?php

include "dbconnect.php";

$facid=$\_SESSION['facid'];$slot=$\_SESSION['slot'];

$stdid=$\_SESSION["userid"];

$records = mysqli\_query($conn,"select \* from atd where facid='$facid' and slot='$slot' and stdid='$stdid'");

$i=1;

while($data = mysqli\_fetch\_array($records))

{

?>

<tr>

<td><?php echo $i++; ?></td>

<td><?php echo $data['date']; ?></td>

<td><?php echo $data['time']; ?></td>

</tr>

<?php

}

?>

</table><br><br>

<a href="student.php">Back</a>

</body>

</html>

**logout.php**

<?php

session\_start();

session\_destroy();

header("Location: http://localhost/attend%20-%20Copy/portal/signin.php", true, 301);

?>

**REFERENCES**

[1] Jr-Jen Huang, Yi-Yu Chu, and Yen-Jen Chen, “The System Design and Implementation of Vehicle Management”, Journal of Advances in Computer Networks, Vol. 1, No. 1, March 2013

[2] Rohit Dhall, Vijender Solanki, “An IoT Based Predictive Connected Car Maintenance Approach”,

International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 4, Nº3

[3] MIT App Inventor http://ai2.appinventor.mit.edu/

[4] K.N. Manoj Kumar, Kailasa Akhi, Sai Kumar Gunti, M. Sai Prathap Reddy, “Implementing smart home using firebase”, International Journal of Research in Engineering and Applied Sciences (IJREAS), Vol. 6 Issue 10, October - 2016, pp. 193~198

[5] Navdeep Singh, “Study of Google Firebase API for Android”, International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 4, Issue 9, September 2016

[6] Sonam Khedkar, Swapnil Thube, “Real Time Databases for Applications”, International Research Journal of Engineering and Technology (IRJET) Real Time Databases for Applications, Volume: 04 Issue: 06 | June -2017

[7] Harshada Chaudhari, “Raspberry Pi Technology: A Review” International Journal of Innovative and Emerging Research in Engineering Volume 2, Issue 3, 2015

[8] Cheah Wai Zhao, Jayanand Jegatheesan, Son Chee Loon, “Exploring IOT Application Using Raspberry Pi”, International Journal of Computer Networks and Applications Volume 2, Issue 1, January - February (2015)

[9] How to set up Raspberrypi https://www.raspberrypi.org/help/videos/

[10] Connecting an Arduino to Raspberry Pi for the best of both worlds <https://conoroneill.net/2013/06/05/connecting-an-arduino-to-raspberry-pi-for-the-best-of-both-worlds/>

[11] Maria-Alexandra Paun, Jean-Michel Sallese and Maher Kayal, “Hall Effect Sensors Design, Integration and Behavior Analysis” Journal of Sensor and Actuator Networks ISSN 2224-2708, Published: 8 February 2013

[12] Paun, M.A.; Sallese, J.M.; Kayal, M. “Temperature Considerations on Hall Effect Sensors

Current-Related Sensitivity Behaviour”, In *Proceedings of the 19th IEEE International* Conference *on Electronics, Circuits, and Systems (ICECS)*, Seville, Spain, 9–12 December 2012

[13] Electronics Basics – How a Potentiometer Works https://randomnerdtutorials.com/electronics-basics-how-a-potentiometer-works/

[14] J. Parthasarathy, “POSITIONING AND NAVIGATION SYSTEM USING GPS”, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, Volume XXXVI, Part 6, Tokyo Japan 2006

[15] Phongsak Keeratiwintakorn, “Real-Time Tracking Management System Using GPS, GPRS and Google Earth”, Conference Paper, June 2008

[16] Anas Siddiqui, Usman Saleem, Abdul Ur Rehman, Sohaib, Shiraz Latif, “GPS and GSM Based Advanced Vehicle Monitoring and Information System”, First International Conference on Modern Communication & Computing Technologies (MCCT'14)

[17] Accelerometer Basics: https://learn.sparkfun.com/tutorials/accelerometer-basics/all

[18] Infrared Temperature Measurement Theory and Application Author and Presenter: John Merchant, Sales Manager, Mikron Instrument Company Inc.

[19] Paun, M.A.; Sallese, J.M.; Kayal, M., “Temperature Influence Investigation on Hall Effect Sensors Performance Using a Lumped Circuit Model”. In *Proceedings of the 11th IEEE Sensors Conference*, Taipei, Taiwan, 28–31 October 2012

[20] Working Principle of Temperature Sensor and Its Application

<https://www.efxkits.us/lm35-temperature-sensor-circuit-working/>

[21] What is Voltage Detector Used For https://www.wonkeedonkeetools.co.uk/voltage-

detectors-and-testers/what-is-a-voltage-detector-used-for/

**BIODATA**

Photo

Name : Sai Abhishek

Mobile Number : 9493911988

E-mail : abhishek.18bcn7116@vitap.ac.in

Permanaent Address : Sai Sri Homes, Flat- G1, Venkatraya Nagar,

Nizampet, Hyderabad (500090)

**NOTE:** Its **MANDATORY** for a student to attach all the PPT’s, Sample Materials, Specification Sheets, Programming Codes and a 5-10 minutes demo Video of the Project Digitally In CD . Stick the Compact Disk (CD) in the final page of the Thesis after binding it.