End Term Presentation for Capstone Project

Attendance Assistant using Deep Learning

Final Year B. Tech CSE (CSF)

Presented to:

Dr. Sharmishta Desai

Literature Survey and Requirements Gathering

Introduction and Problem Statement

- Manual attendance tracking at MITWPU Campus is a time-consuming process prone to inefficiencies and potential malpractices. The current system lacks automation, leading to increased per-class time spent on attendance management. To address these challenges, there is a need for the development of an Automated Attendance Tracking System.

- The solution should leverage computer vision for accurate data capture, utilize cloud infrastructure for efficient storage and processing, and incorporate advanced data science techniques for analytics.

The system should be user-friendly, providing both teachers and students with a seamless experience while ensuring security and tamper resistance to mitigate the risk of malpractices in attendance tracking. The goal is to enhance overall efficiency and significantly reduce the manual effort involved in attendance monitoring.

Motivation

1. Time taken to take attendance is about 5 minutes in each class. That amounts to around 5 times 6 which is 30 minutes wasted each day per class. For 2 CSF, 2 AI, and 1 CSBS, and 8 CSE Panels, 13 panels for CSE alone waste more than 13 x 30 = 390 Minutes, which is more than 6 hours daily, just marking attendance.
2. This amounts to 6 Hours * 5 * 4 * 10 = 1200 Hours per year just for CSE (50 Days) or 2 months
3. Even after all that time, several more hours are spent on arguments, discussions and manual entry and marking by teachers and students, just for attendance. We do not find this an ideal use of time, and would like to combat this.
4. Malpractices still happen despite of all efforts and time spent. Attendance is mismarked by human error as well. We aim to reduce this error.

Papers Referred

Sr.No	*Publication Title with Author*	*Year*	*Positive Points of the Publication*	*Gaps of the Publication*
1	Title: *"A Comparative Study of Facial Recognition Techniques: With focus on low computational power."* [1] Author: *Schenkel, T., Ringhage, O. and Branding, N.*	2019	1) The publication compares five performance metrics, including recall and F-score, providing a comprehensive evaluation of facial recognition techniques. 2) It addresses the importance of balancing low computational time and prediction ability for security systems, offering practical guidelines for implementation. 3) The research questions are clearly defined, focusing on significant differences in performance, training time, and prediction time among different facial recognition techniques and classifiers.	1) The document lacks detailed information on the specific facial recognition techniques and classifiers used in the experiments. 2) It does not provide a detailed breakdown of the dataset used for training and testing the facial recognition models. 3) While the document mentions the comparison of results, it does not delve into the specific findings or implications of these comparisons.

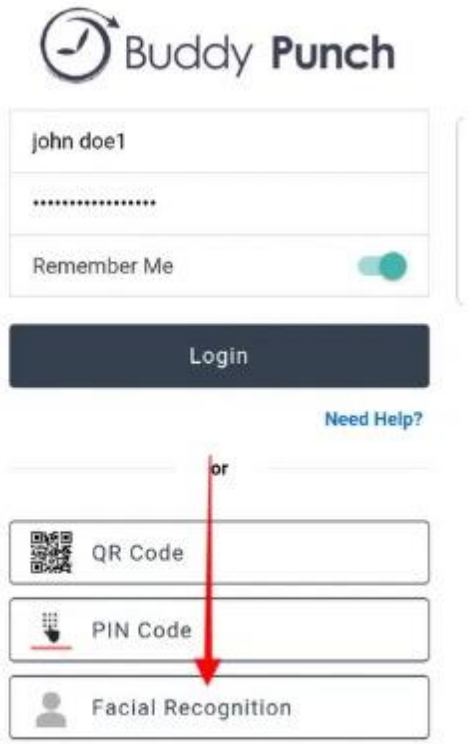
2	Title: *"A Comparative Study on Facial Recognition Algorithms"* [2] Author: *Sanmoy Paul and Sameer Acharya*	2018	1) Comparative Analysis: The study provides a comparative analysis of different facial recognition algorithms, allowing developers to make informed choices based on recognition accuracies. 2) Algorithm Selection: By studying the advantages and disadvantages of various algorithms, developers can select the best facial recognition algorithm for their specific implementation needs. 3) Future Improvements: The research suggests future efforts to test on a larger set of images to enhance the accuracy of CNN and explore combining multiple machine learning classification algorithms for increased recognition accuracy and handling large datasets.	1) The document lacks detailed discussion on the specific methodologies used for training and testing the algorithms, which could provide more clarity on the experimental setup. 2) There is no mention of the computational resources or hardware specifications used for running the experiments, which could impact the reproducibility and scalability of the results. 3) The publication does not delve into the potential biases or limitations in the dataset used for training and testing the facial recognition models, which could affect the generalizability of the findings.

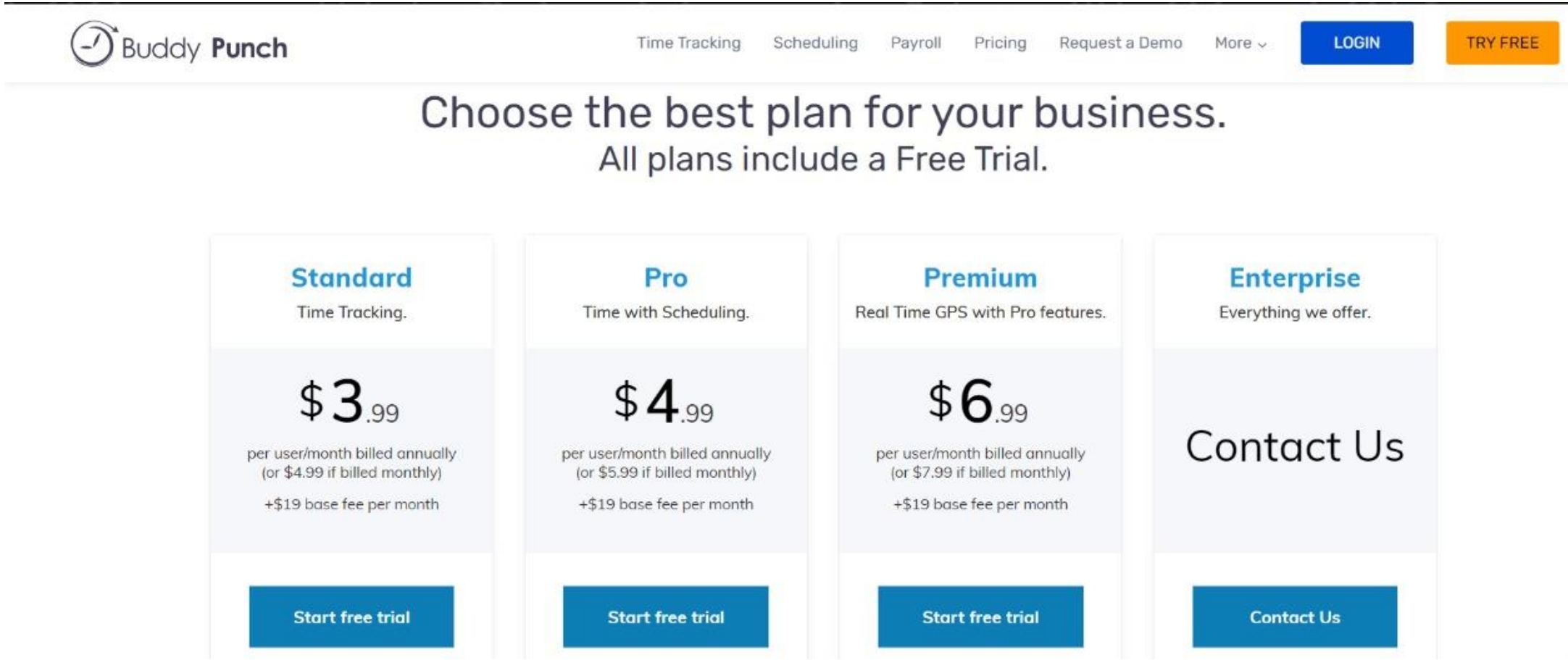
3	Title: *"A comparison of facial recognition algorithms."* [3] Author: *Delbiaggio, Nicolas.*	2017	1) Thesis covers a comprehensive comparison of facial recognition algorithms like Eigenfaces, Fisherfaces, LBPH, and OpenFace. 2) The study includes a detailed explanation of each algorithm, their strengths, weaknesses, and performance in a test case scenario. 3) The findings highlight OpenFace as the most accurate algorithm for facial recognition, providing valuable insights for further research in the field.	1) Lack of Exploration of Real-World Applications: The paper focuses on comparing facial recognition algorithms in a controlled setting. However, it does not delve into the practical applications of these algorithms in real-world scenarios. 2) Limited Discussion on Algorithm Limitations: While the strengths of the algorithms are discussed, there is a lack of emphasis on the limitations of each algorithm. 3) Absence of Future Research Directions: The paper concludes with the identification of the most accurate algorithm but fails to suggest potential future research directions in the field of facial recognition.

4	Title: *"Evaluating impact of race in facial recognition across machine learning and deep learning algorithms."* [4] Author: *Coe, James, and Mustafa Atay.*	2021	1) The paper provides a detailed comparison of various facial recognition algorithms, including Eigenfaces, Fisherfaces, Local Binary Pattern Histogram, deep convolutional neural network algorithm, and OpenFace. 2) It highlights the efficiency and accuracy of these algorithms in real-life settings, with OpenFace being identified as the algorithm with the highest accuracy in identifying faces. 3) The study's findings offer valuable insights for practitioners in selecting the most suitable algorithm for facial recognition applications and suggest ways for academicians to enhance the current algorithms' accuracy further.	1) The paper focuses on a few specific facial recognition algorithms like Eigenfaces, Fisherfaces, and Local Binary Pattern Histograms. It lacks exploration of a wider range of algorithms available in the field, potentially missing out on newer, more accurate models. 2) While the study evaluates the algorithms' accuracy, it does not delve into their performance in real-life settings or practical applications. This gap could impact the algorithms' effectiveness when deployed in scenarios beyond controlled test environments. 3) The paper mentions the use of a custom dataset for testing the algorithms but does not elaborate on the dataset's diversity or size.

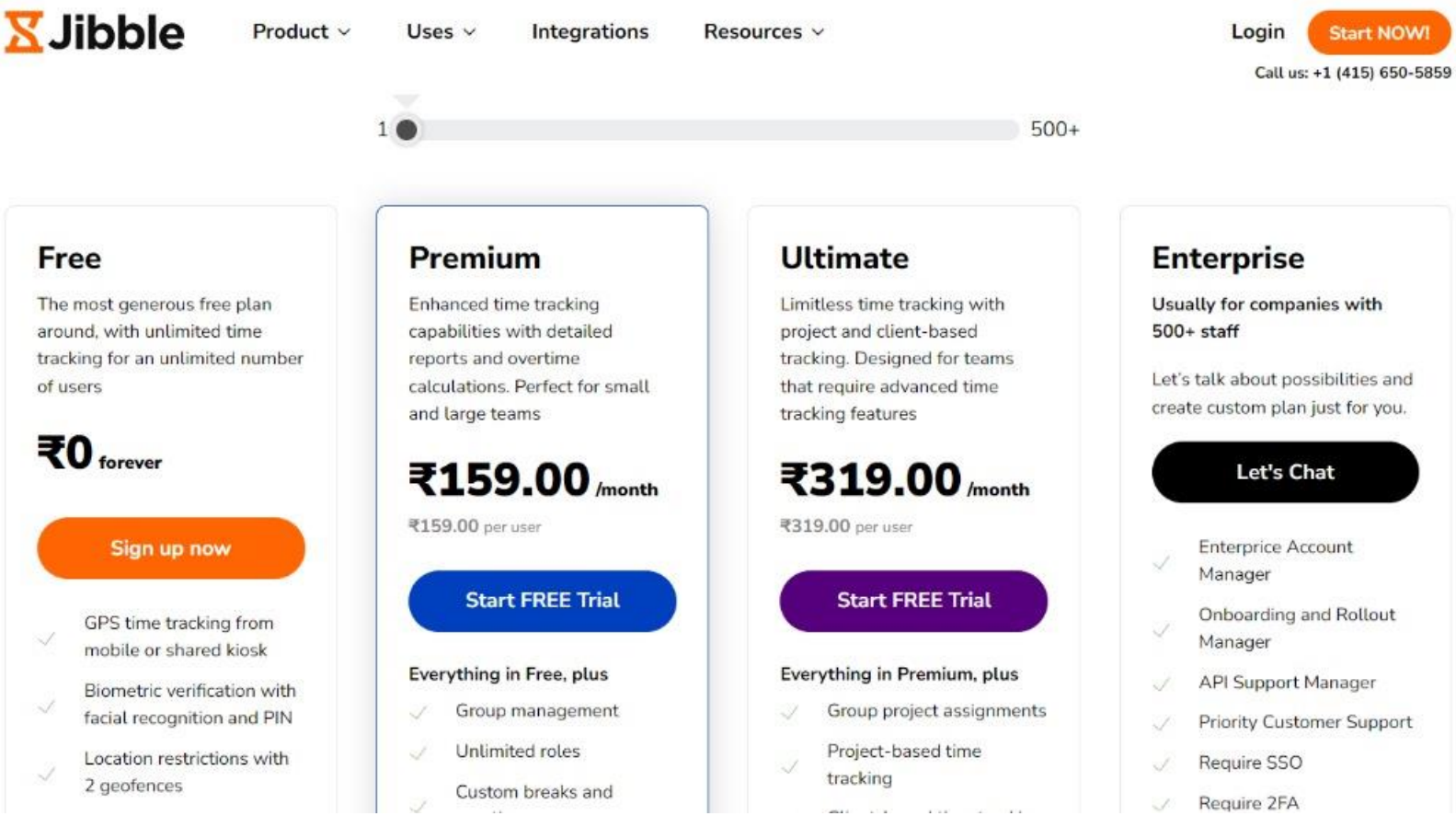
Market Research

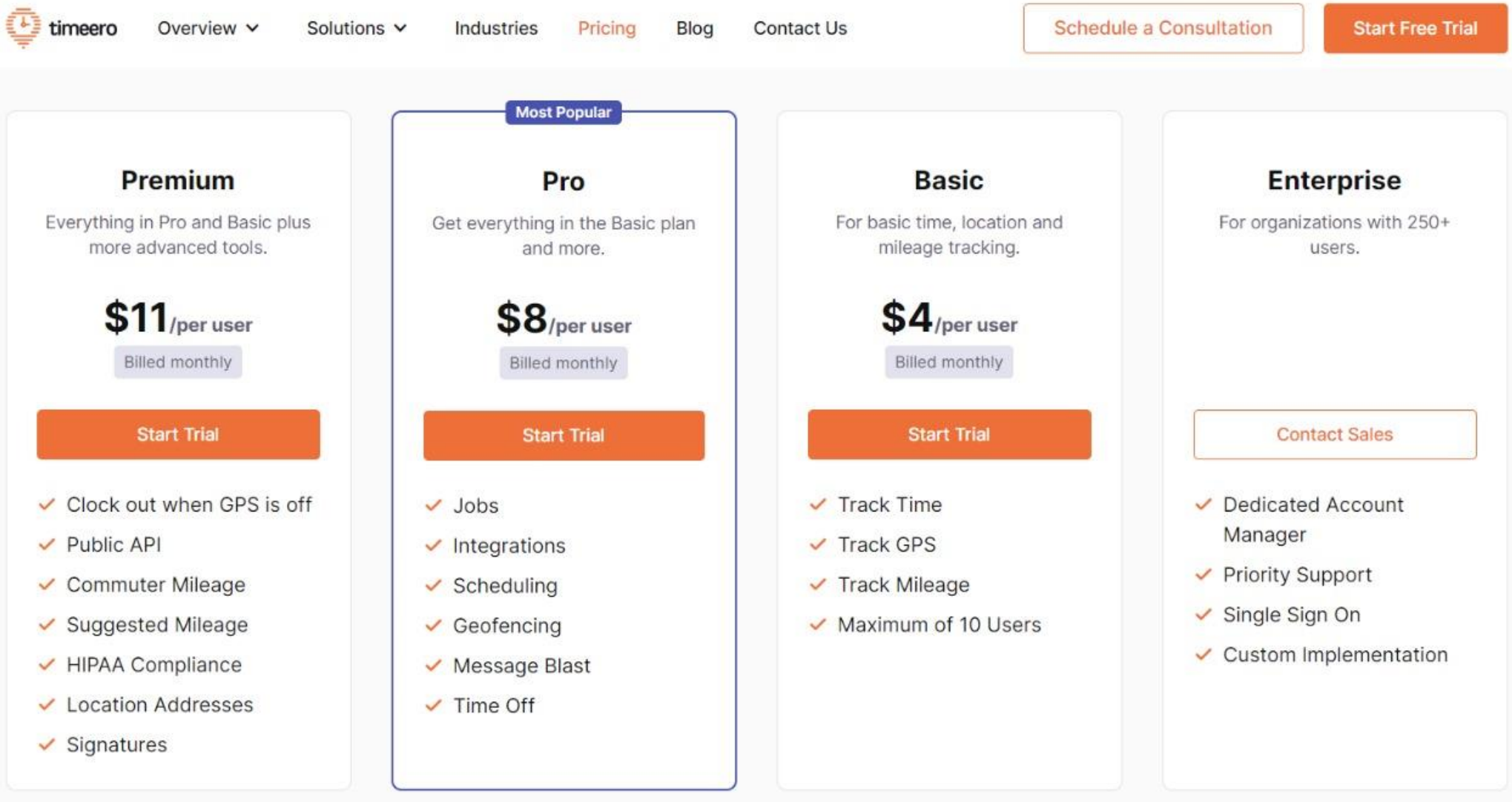
Existing Products in Market and Cost Comparison




Buddy Punch
Time Tracking
Scheduling
Payroll
Pricing
Request a Demo
More
LOGIN
TRY FREE
Choose the best plan for your business.
All plans include a Free Trial.
Standard
Time Tracking.
$3.99
per user/month billed annually
(or $4.99 if billed monthly)
+$19 base fee per month
Start free trial
Pro
Time with Scheduling.
$4.99
per user/month billed annually
(or $5.99 if billed monthly)
+$19 base fee per month
Start free trial
Premium
Real Time GPS with Pro features.
$6.99
per user/month billed annually
(or $7.99 if billed monthly)
+$19 base fee per month
Start free trial
Enterprise
Everything we offer.
Contact Us
Contact Us

Cost Comparison




timeero
Overview
Solutions
Industries
Pricing
Blog
Contact Us
Schedule a Consultation
Start Free Trial
Most Popular
Premium
Everything in Pro and Basic plus more advanced tools.
$11/per user
Billed monthly
Start Trial
Clock out when GPS is off
Public API
Commuter Mileage
Suggested Mileage
HIPAA Compliance
Location Addresses
Signatures
Pro
Get everything in the Basic plan and more.
$8/per user
Billed monthly
Start Trial
Jobs
Integrations
Scheduling
Geofencing
Message Blast
Time Off
Basic
For basic time, location and mileage tracking.
$4/per user
Billed monthly
Start Trial
Track Time
Track GPS
Track Mileage
Maximum of 10 Users
Enterprise
For organizations with 250+ users.
Contact Sales
Dedicated Account Manager
Priority Support
Single Sign On
Custom Implementation

Home Features Pricing Contact Login **Free Trial**

Get started in minutes, no credit card needed.

Monthly Yearly

# of Users (slide to adjust)

5 Users

Free

Free

Everything in Premium

- Up to 5 Users
- Free for 30 Days

Pro

$19.50 USD / MO

$ 12.00 base + $ 1.50 per user

- GPS time tracking
- Shift scheduling
- Payroll calculation

Premium

$26.00 USD / MO

$ 16.00 base + $ 2.00 per user

Everything in Pro, plus

- Alert notifications
- Attendance points

Enterprise

Custom pricing

Everything in Premium, plus

- Enterprise Account Manager
- Priority Support

truein

Industry ▾ Solutions ▾ Pricing Customer Stories About Us Login Schedule a Demo

Time and Attendance Solution for Contractual and Distributed workforce

Truein Pricing Plans

Premium

₹ 390 /user/year

+ ₹69,000 /year (base price) ⓘ

Get started

✓ Face recognition based Time Tracking from Mobile

Enterprise

More than 2000 staff?

Talk to Sales

Everything in Premium +

Existing Market Competitors

- Buddy Punch
- Timeero
- FareClock
- Jibble
- QuickBooks

Research Gaps Identified / Implementation Scope

Gap 1: Lack of Specific Facial Recognition Libraries and Models Documentation

Existing literature fails to provide a comprehensive overview of specific facial recognition libraries and models. The omission of detailed discussions on these crucial components hinders a holistic understanding of the technological landscape.

Gap 2: Absence of Hardware Elements Documentation

A significant gap is identified in the absence of detailed documentation regarding the hardware elements employed. The current body of work overlooks crucial insights into the hardware aspects, limiting the understanding of the complete facial recognition system.

Gap 3: Insufficient Information on Interlinking Hardware and Software

The interplay between hardware and software components is a critical aspect of facial recognition systems. However, the existing literature lacks substantial information on how these elements are intricately interlinked, hindering a comprehensive grasp of the system's architecture.

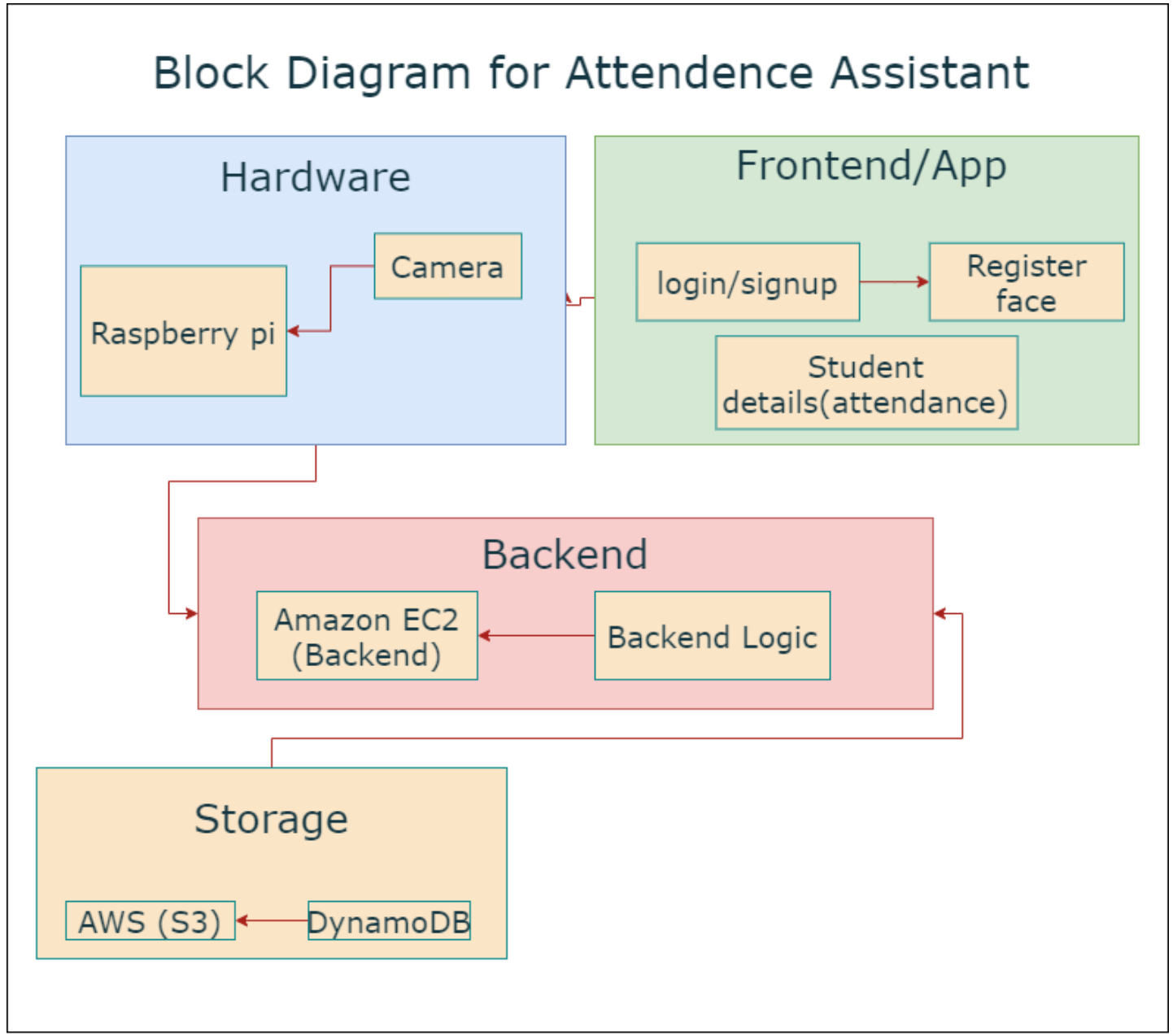
Gap 4: Inadequate Details on Storage and Integration of Information

The literature review reveals a gap in information pertaining to the storage and seamless integration of facial recognition data. Understanding the mechanisms for data storage and integration is essential for evaluating the system's overall efficiency, and this aspect requires further exploration.

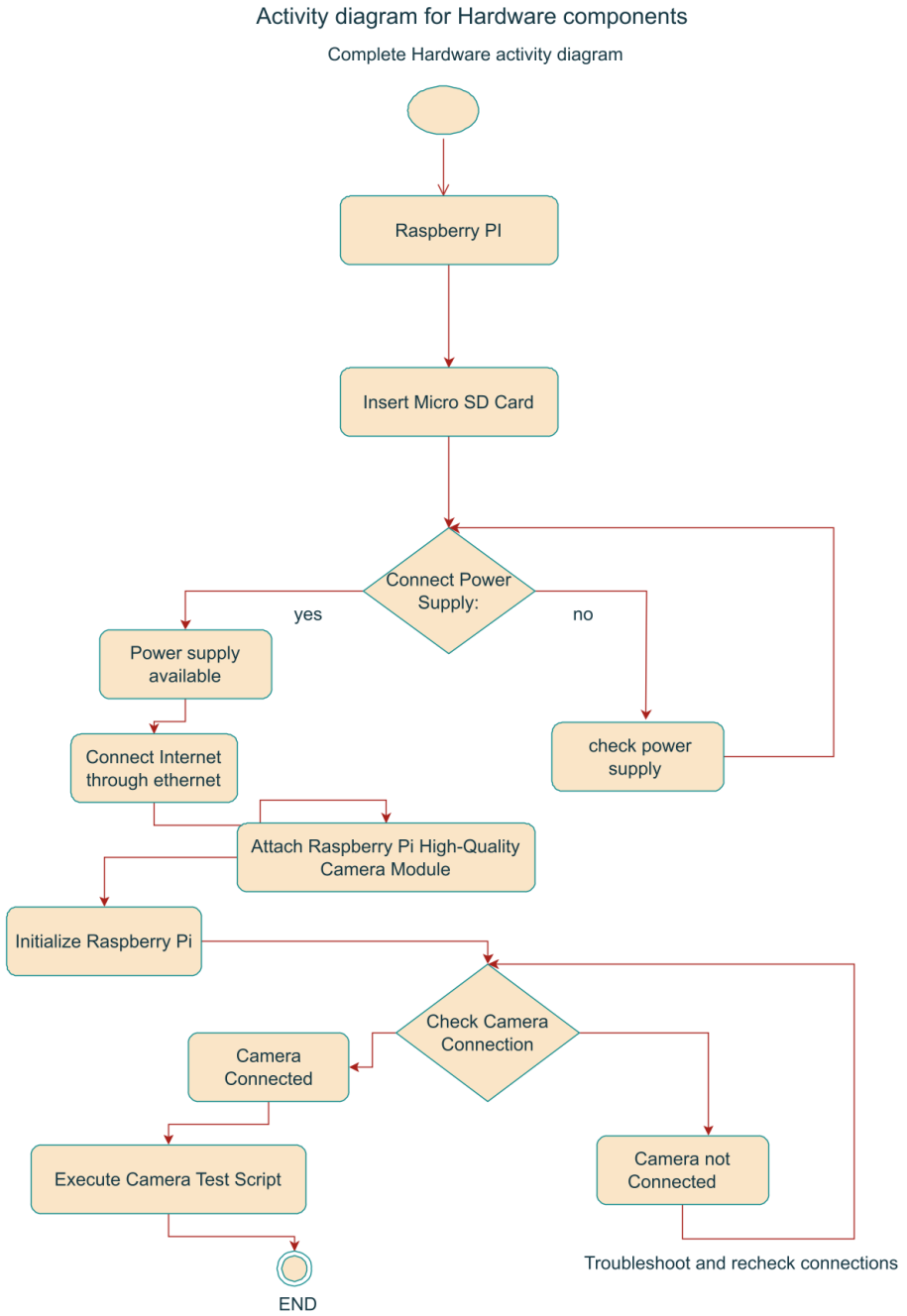
Gap 5: Lack of Seamless Integration

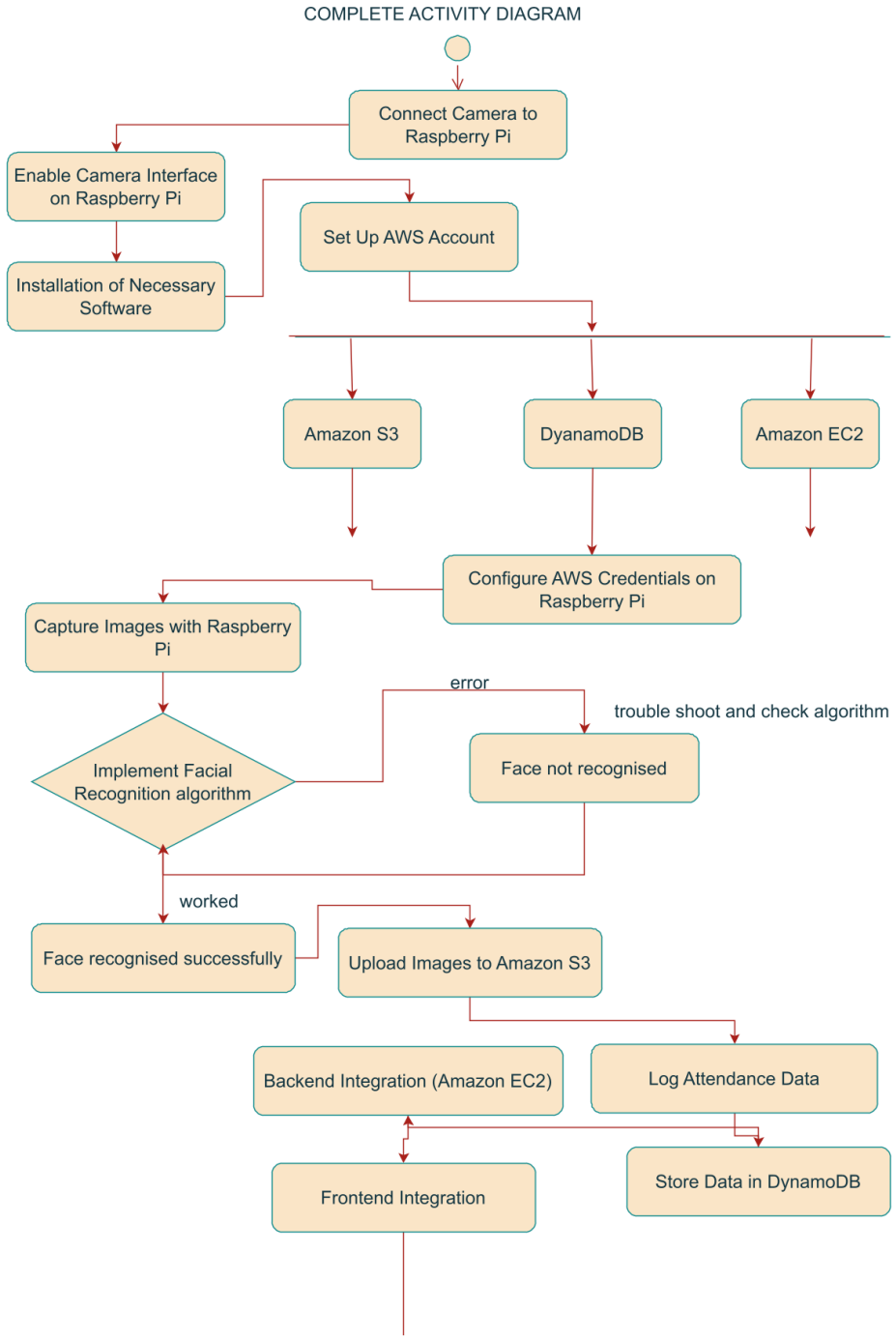
The current research landscape highlights a significant gap concerning the seamless integration of services. Notably, existing systems rely on manual user initiation, requiring individuals to independently open dedicated services on their personal devices. This operational hurdle indicates a crucial area for improvement in achieving a more streamlined and user-friendly experience. Addressing this gap is paramount for enhancing the overall efficiency and user adoption of the services, warranting further exploration and innovation in the integration protocols employed.

System Design


Block Diagram for Attendence Assistant
Hardware
Camera
Raspberry pi
Frontend/App
login/signup
Register face
Student details(attendance)
Backend
Amazon EC2 (Backend)
Backend Logic
Storage
AWS (S3)
DynamoDB

Activity Diagram





Implementation

FastAPI 0.1.0 OAS 3.1

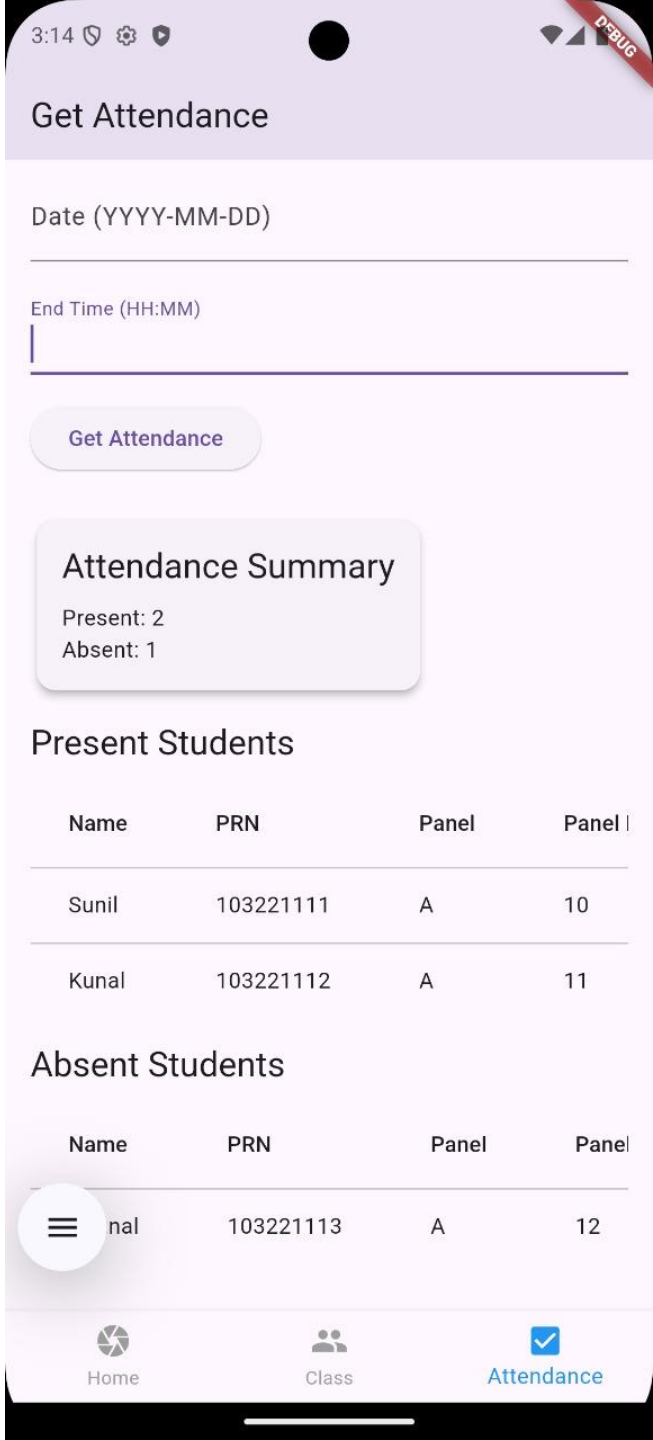
/openapi.json

Upload Images or Attendance Info from App/Website/Pi

- POST `/upload/add_student_face_from_url` Add Student Face From Url Route
- POST `/upload/add_student_face` Add Student Face Route
- POST `/upload/add_class_photo_from_url` Add Class Photo From Url Route
- POST `/upload/add_class_photo` Add Class Photo Route
- POST `/upload/add_attendance` Add Attendance Route
- POST `/upload/add_face_encoding` Add Face Encoding Route
- POST `/upload/update_face_encoding` Update Face Encoding Route

Students

- GET `/student/test` Test route
- POST `/student/add_student` Add a student



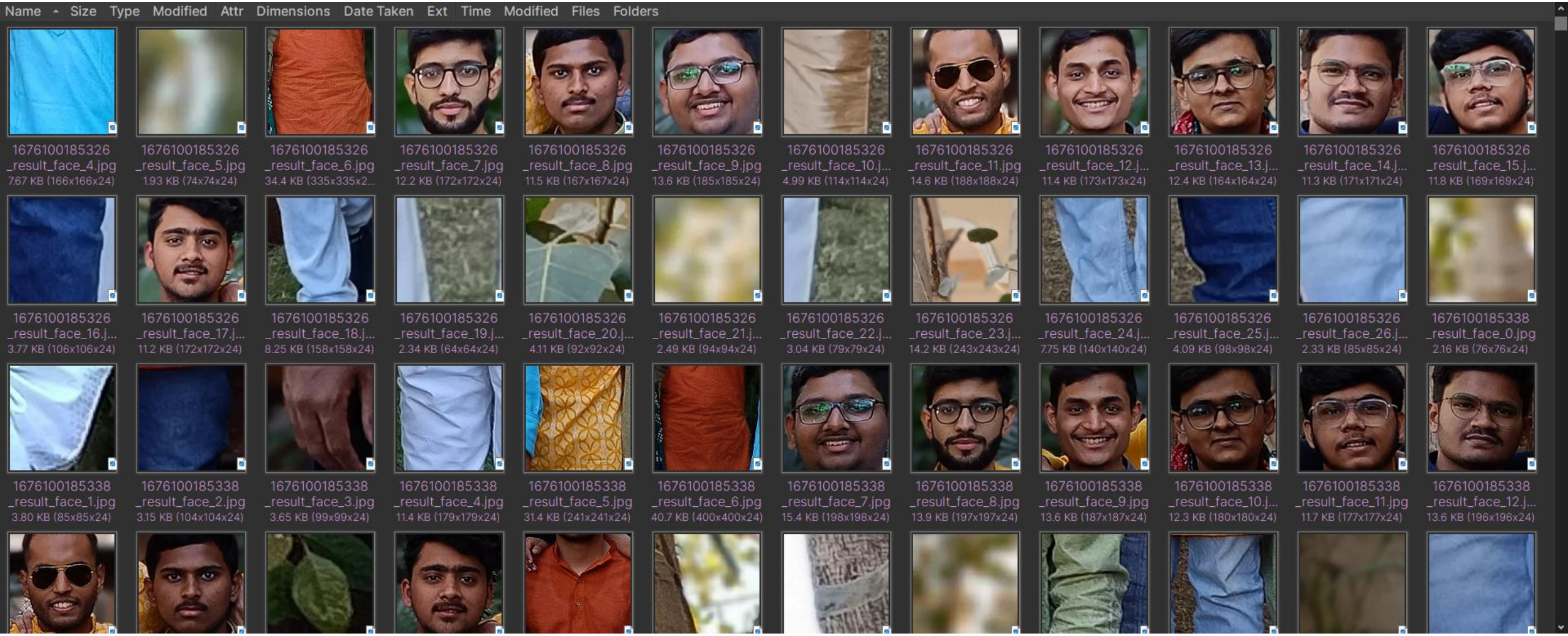
Source Code

Creation of Dataset for Training

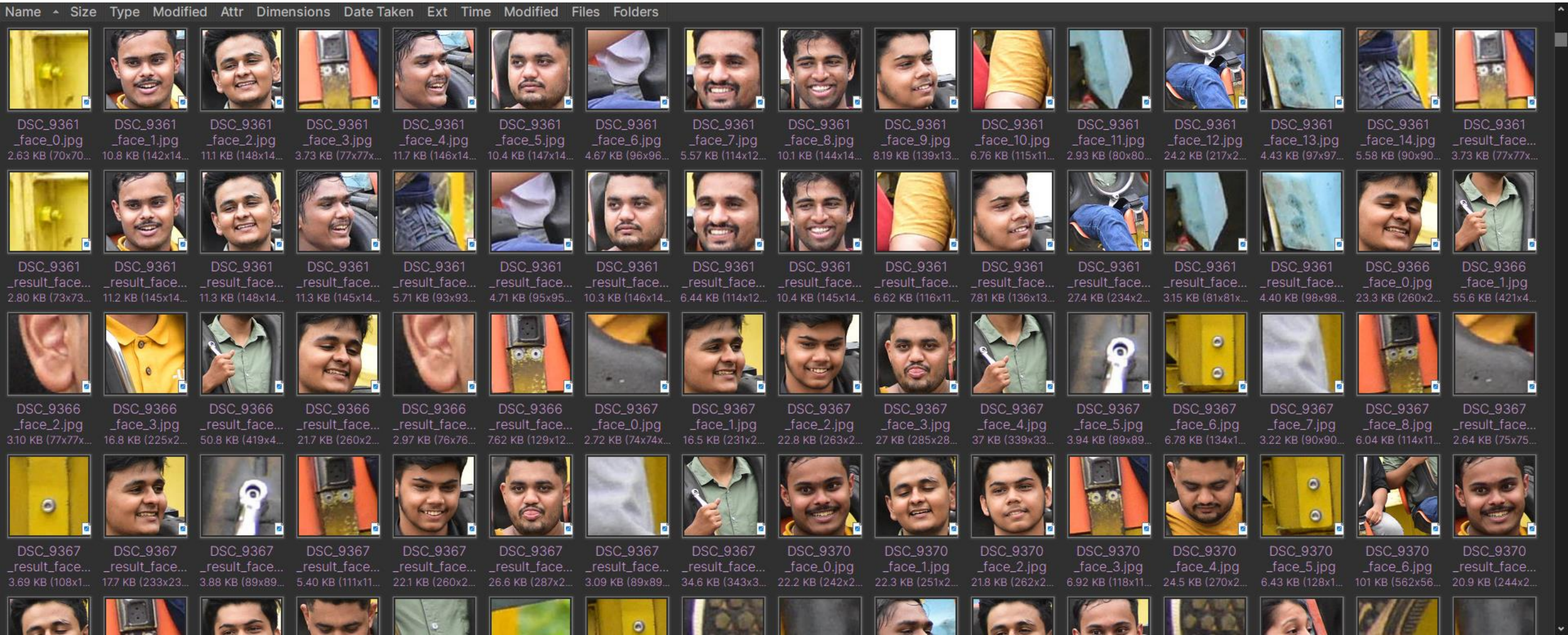
Multiple images from the past 4 years Involving a group of 15 people were Taken.

Cropped Faces using opencv-haar-cascades

```python
def detect_and_crop_faces(input_folder, output_folder, padding=10):
    if not os.path.exists(output_folder):
        os.makedirs(output_folder)

    face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

    for filename in os.listdir(input_folder):
        if filename.lower().endswith(('png', 'jpg', 'jpeg', 'webp')):
            image_path = os.path.join(input_folder, filename)
            image = cv2.imread(image_path)

            if image is None:
                continue

            gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

            for i, (x, y, w, h) in enumerate(faces):
                x1 = max(x - padding, 0)
                y1 = max(y - padding, 0)
                x2 = min(x + w + padding, image.shape[1])
                y2 = min(y + h + padding, image.shape[0])

                face_crop = image[y1:y2, x1:x2]
                output_path = os.path.join(output_folder, f"{os.path.splitext(filename)[0]}_face_{i}.jpg")
                cv2.imwrite(output_path, face_crop)
                print(f"Saved cropped face: {output_path}")

input_folder = os.path.join(os.getcwd(), "input_images")
output_folder = os.path.join(os.getcwd(), "output_images")

detect_and_crop_faces(input_folder, output_folder)
```


Name Size Type Modified Attr Dimensions Date Taken Ext Time Modified Files Folders
1676100185326_result_face_4.jpg 7.67 KB (166x166x24)
1676100185326_result_face_5.jpg 1.93 KB (74x74x24)
1676100185326_result_face_6.jpg 34.4 KB (335x335x2...
1676100185326_result_face_7.jpg 12.2 KB (172x172x24)
1676100185326_result_face_8.jpg 11.5 KB (167x167x24)
1676100185326_result_face_9.jpg 13.6 KB (185x185x24)
1676100185326_result_face_10.j... 4.99 KB (114x114x24)
1676100185326_result_face_11.jpg 14.6 KB (188x188x24)
1676100185326_result_face_12.j... 11.4 KB (173x173x24)
1676100185326_result_face_13.j... 12.4 KB (164x164x24)
1676100185326_result_face_14.j... 11.3 KB (171x171x24)
1676100185326_result_face_15.j... 11.8 KB (169x169x24)
1676100185326_result_face_16.j... 3.77 KB (106x106x24)
1676100185326_result_face_17.j... 11.2 KB (172x172x24)
1676100185326_result_face_18.j... 8.25 KB (158x158x24)
1676100185326_result_face_19.j... 2.34 KB (64x64x24)
1676100185326_result_face_20.j... 4.11 KB (92x92x24)
1676100185326_result_face_21.j... 2.49 KB (94x94x24)
1676100185326_result_face_22.j... 3.04 KB (79x79x24)
1676100185326_result_face_23.j... 14.2 KB (243x243x24)
1676100185326_result_face_24.j... 7.75 KB (140x140x24)
1676100185326_result_face_25.j... 4.09 KB (98x98x24)
1676100185326_result_face_26.j... 2.33 KB (85x85x24)
1676100185338_result_face_0.jpg 2.16 KB (76x76x24)
1676100185338_result_face_1.jpg 3.80 KB (85x85x24)
1676100185338_result_face_2.jpg 3.15 KB (104x104x24)
1676100185338_result_face_3.jpg 3.65 KB (99x99x24)
1676100185338_result_face_4.jpg 11.4 KB (179x179x24)
1676100185338_result_face_5.jpg 31.4 KB (241x241x24)
1676100185338_result_face_6.jpg 40.7 KB (400x400x24)
1676100185338_result_face_7.jpg 15.4 KB (198x198x24)
1676100185338_result_face_8.jpg 13.9 KB (197x197x24)
1676100185338_result_face_9.jpg 13.6 KB (187x187x24)
1676100185338_result_face_10.j... 12.3 KB (180x180x24)
1676100185338_result_face_11.jpg 11.7 KB (177x177x24)
1676100185338_result_face_12.j... 13.6 KB (196x196x24)

This generated 18, 832 possible faces (245x245px) each



Test Code

```
import os
import cv2
import numpy as np
import dlib
import face_recognition
from deepface import DeepFace
from facenet_pytorch import InceptionResnetV1
from mtcnn import MTCNN
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics import accuracy_score
```

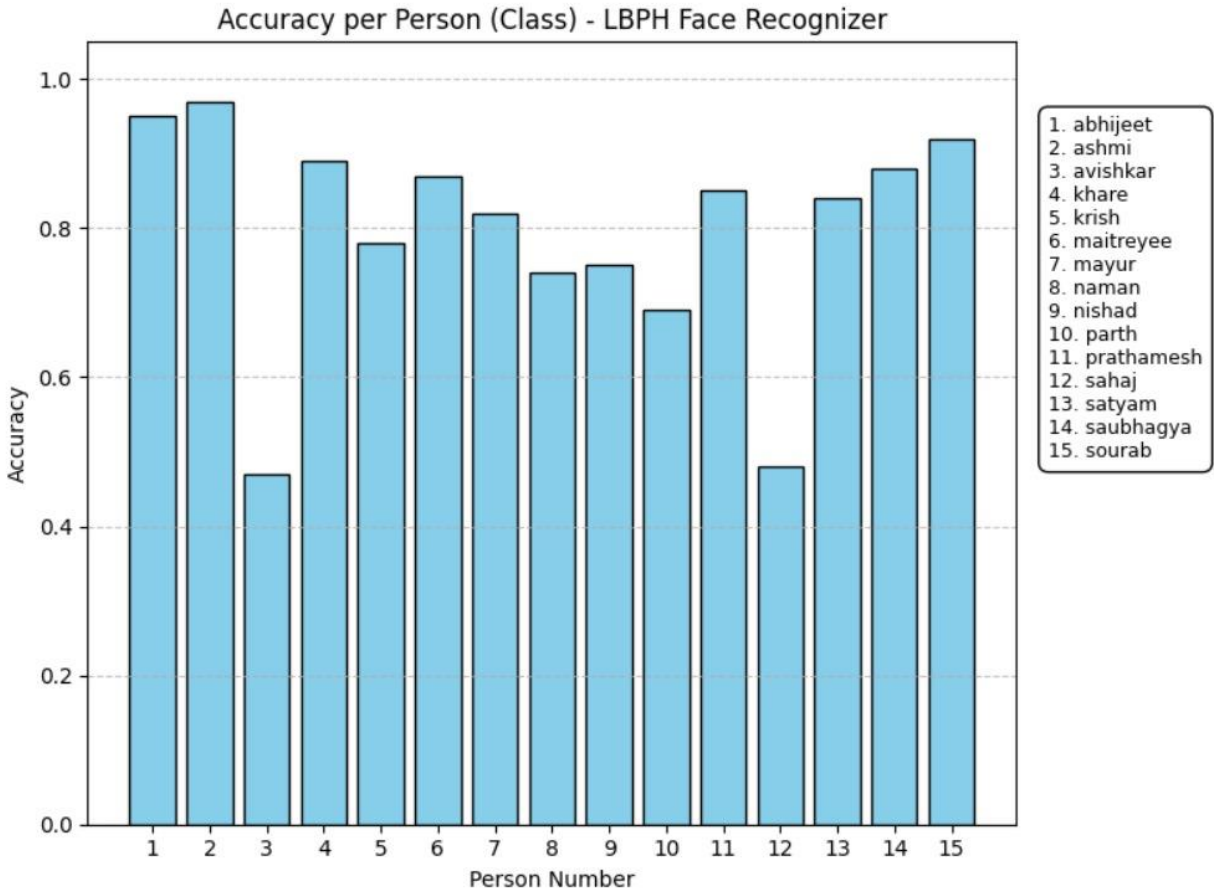
```
# Paths
TRAIN_DIR = "train_db"
TEST_DIR = "test_db"

# Store embeddings
dlib_encodings = {}
facenet_encodings = {}
lbph_recognizer = cv2.face.LBPHFaceRecognizer_create()
```

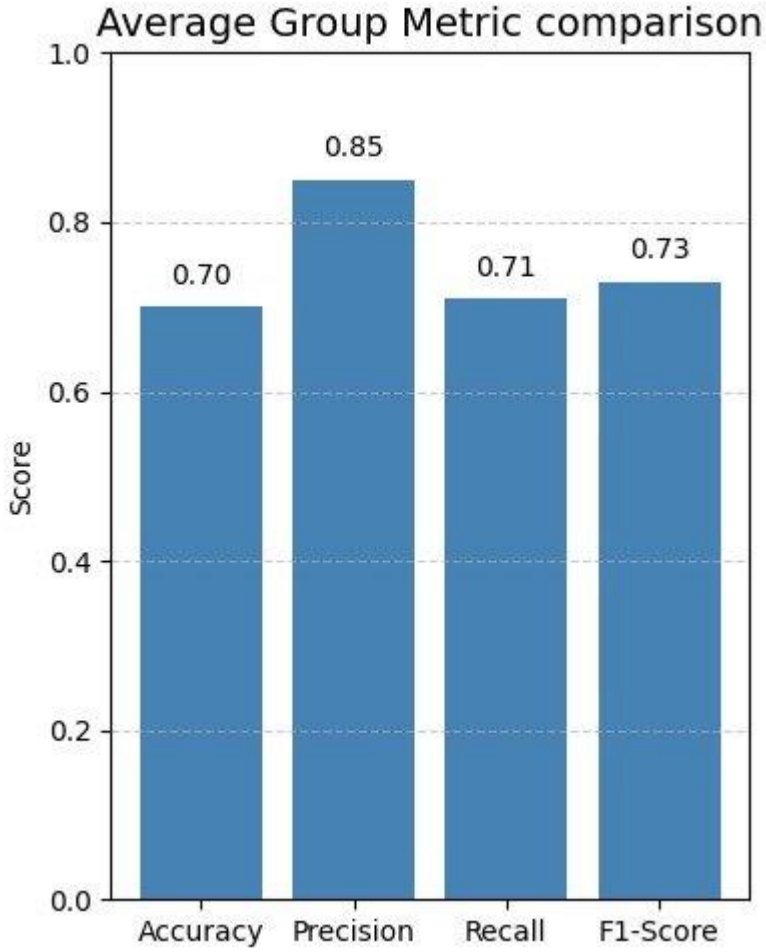
```
# ———————— 1. Load Training Data ————————

def load_images_from_folder(folder):···

train_images, train_labels, label_map = load_images_from_folde

# ———————— 2. Train Dlib & FaceNet Embeddings ————————

def get_dlib_embedding(image):···

def get_facenet_embedding(image):···

for img, label in zip(train_images, train_labels):···

# ———————— 3. Train LBPH ————————

gray_images = [cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) for img i
lbph_recognizer.train(gray_images, np.array(train_labels))

# ———————— 4. Test on Unlabeled Images ————————

def recognize_face_dlib(image):···

def recognize_face_facenet(image):···

def recognize_face_lbph(image):···
```

```
for test_img in test_images:
    dlib_results.append(recognize_face_dlib(test_img))
    facenet_results.append(recognize_face_facenet(test_img))
    lbph_results.append(recognize_face_lbph(test_img))

# ---------- 5. Compare Results ----------

ground_truth = [folder for folder in os.listdir(TEST_DIR) if os.path.isdir(os.path.join(TEST_DIR,
folder))]

dlib_accuracy = accuracy_score(ground_truth, dlib_results)
facenet_accuracy = accuracy_score(ground_truth, facenet_results)
lbph_accuracy = accuracy_score(ground_truth, lbph_results)

print(f"Dlib Accuracy: {dlib_accuracy:.2f}")
print(f"FaceNet Accuracy: {facenet_accuracy:.2f}")
print(f"LBPH Accuracy: {lbph_accuracy:.2f}")
```

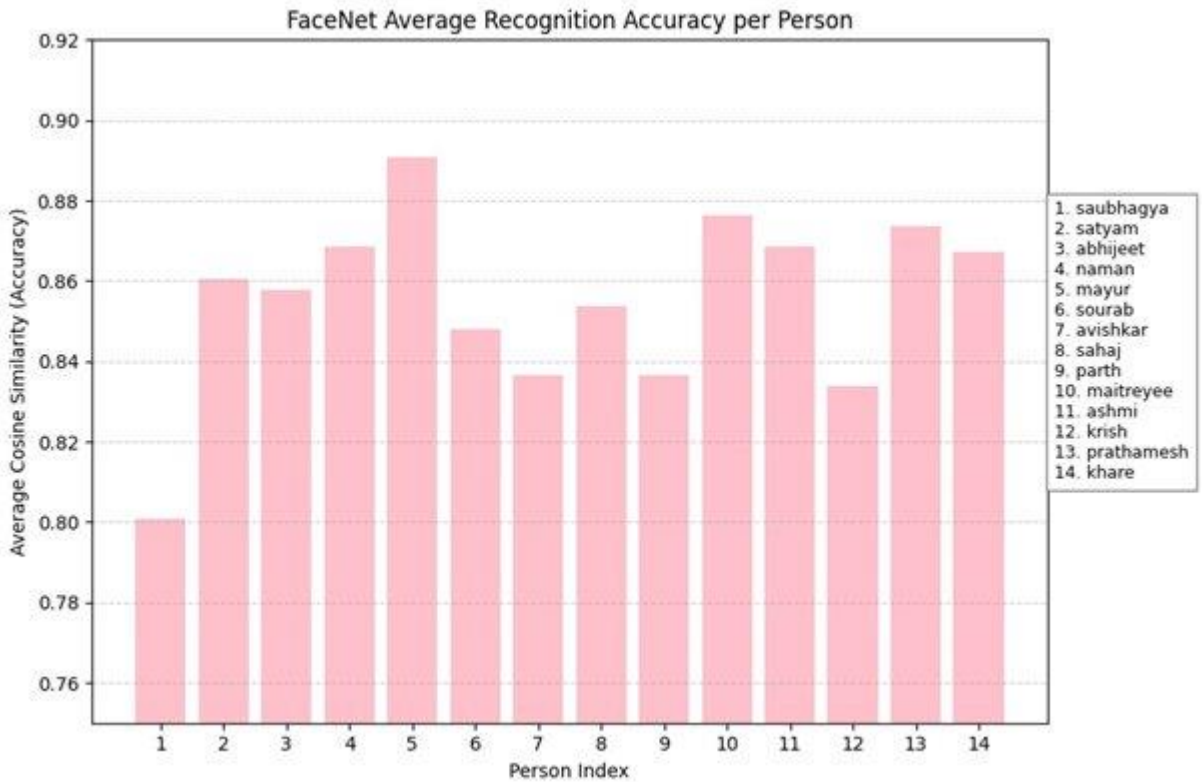
Results

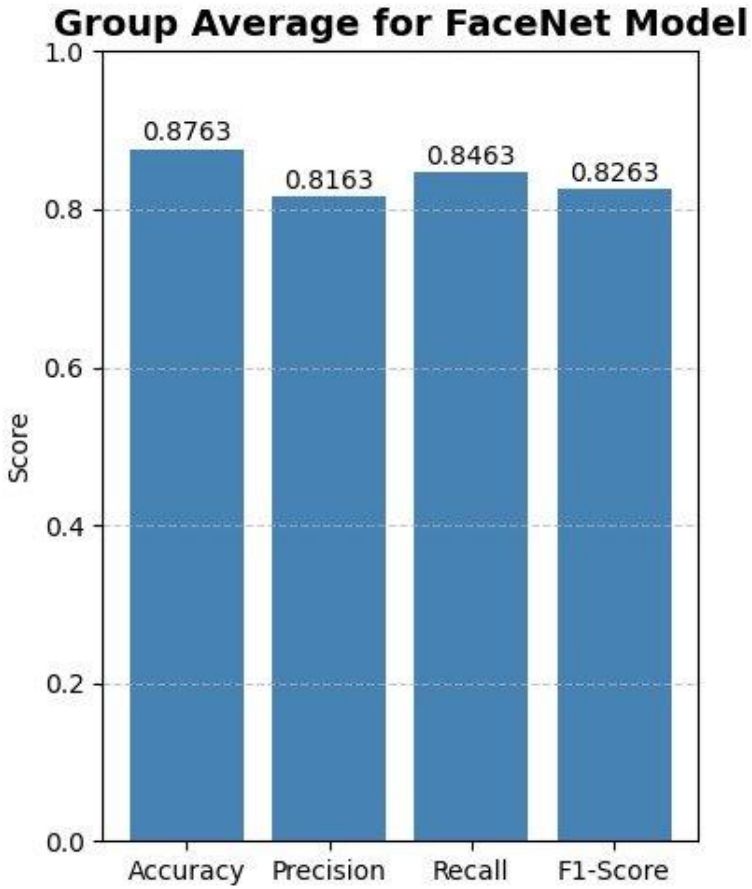
LBPH


Accuracy per Person (Class) - LBPH Face Recognizer
Accuracy
Person Number
1. abhijeet
2. ashmi
3. avishkar
4. khare
5. krish
6. maitreyee
7. mayur
8. naman
9. nishad
10. parth
11. prathamesh
12. sahaj
13. satyam
14. saubhagya
15. sourab

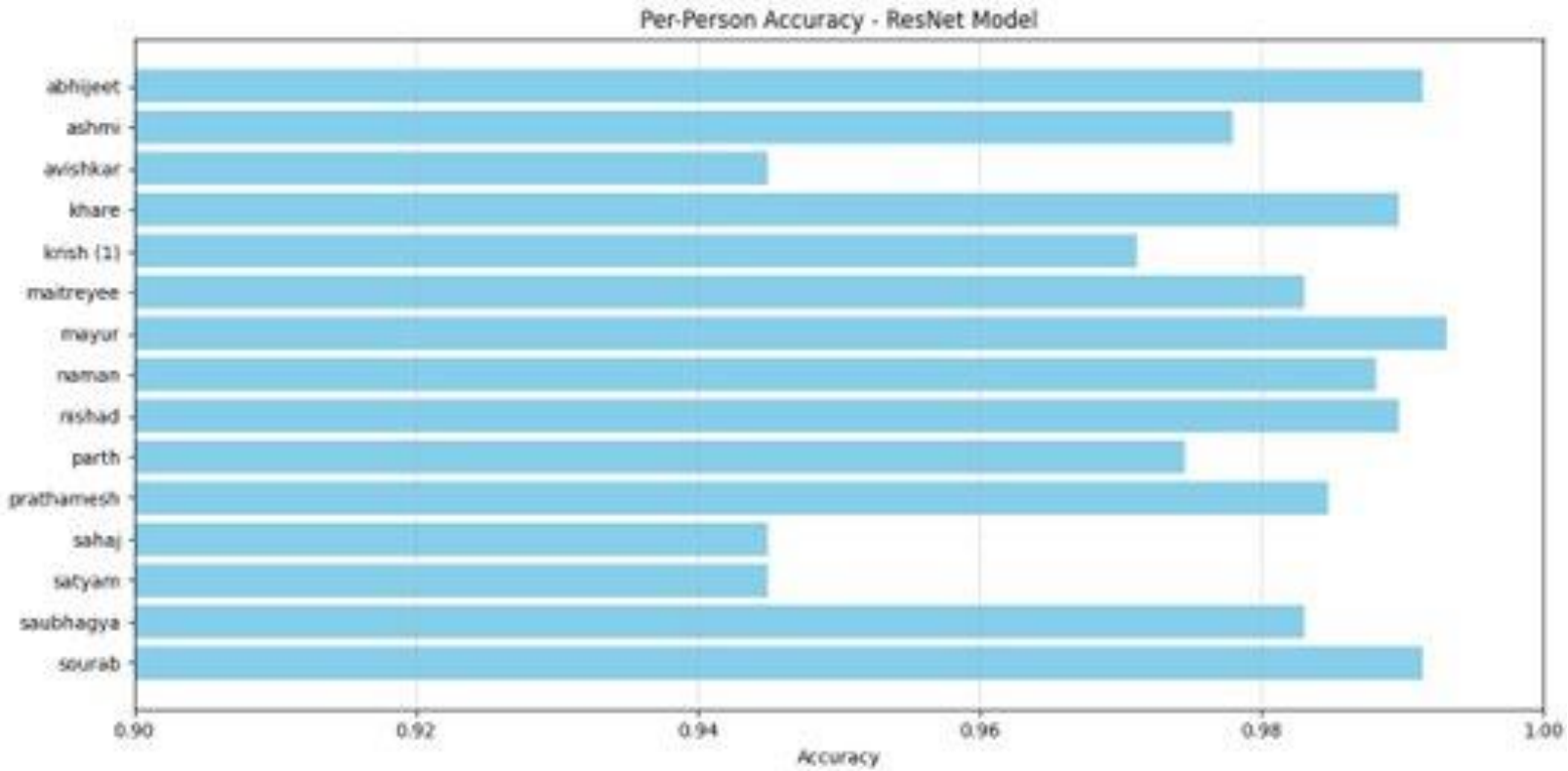

Average Group Metric comparison
Score
0.70
0.85
0.71
0.73
Accuracy
Precision
Recall
F1-Score

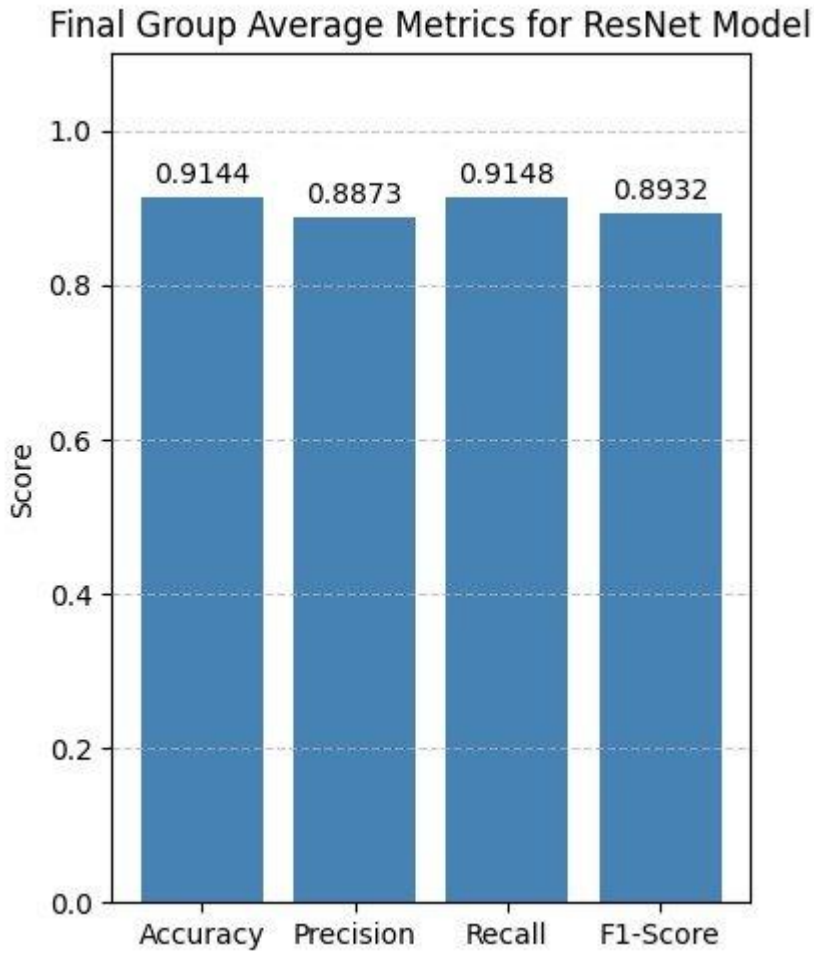
Face-net


FaceNet Average Recognition Accuracy per Person
Average Cosine Similarity (Accuracy)
0.92
0.90
0.88
0.86
0.84
0.82
0.80
0.78
0.76
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Person Index
1. saubhagya
2. satyam
3. abhijeet
4. naman
5. mayur
6. sourab
7. avishkar
8. sahaj
9. parth
10. maitreyee
11. ashmi
12. krish
13. prathamesh
14. khare


Group Average for FaceNet Model
Score
1.0
0.8
0.6
0.4
0.2
0.0
0.8763
0.8163
0.8463
0.8263
Accuracy
Precision
Recall
F1-Score

Res-net


Per-Person Accuracy - ResNet Model
abhijeet
ashmi
avishkar
khare
krish (1)
maitreyee
mayur
naman
rishad
parth
prathamesh
sahaj
satyam
saubhagya
sourab
0.90
0.92
0.94
0.96
0.98
1.00
Accuracy


Final Group Average Metrics for ResNet Model
Score
1.0
0.8
0.6
0.4
0.2
0.0
0.9144
0.8873
0.9148
0.8932
Accuracy
Precision
Recall
F1-Score

Deployment Strategies

- The application is designed to be deployed on a cloud-based server, ensuring scalability and accessibility.
- The backend is built using FastAPI, which allows for efficient handling of requests and responses.
- The database is managed using MongoDB, providing a flexible and scalable solution for data storage.
- The frontend is developed using Flutter, enabling cross-platform compatibility for both Android and iOS devices.
- The application can be deployed on platforms like AWS, Google Cloud, or Azure, ensuring high availability and reliability.

Security Aspects and Project Maintenance

1. The system will implement secure user authentication using JWT (JSON Web Tokens) to protect user data and prevent unauthorized access.

2. All sensitive data, including user credentials and face encodings, will be encrypted before being stored in the database.

3. The system will use HTTPS for secure communication between the frontend and backend components, ensuring that data transmitted over the network is encrypted.

4. The project will follow best practices for data privacy and compliance with relevant regulations, such as GDPR, to protect user information.

5. The system will implement access controls to restrict access to sensitive data and functionalities based on user roles (e.g., admin, teacher, student).

6. The project will include regular security audits and vulnerability assessments to identify and address potential security risks.

7. The system will implement logging and monitoring mechanisms to track user activities and detect any suspicious behavior.

Privacy Improvements

Federated Learning and on-Device Training: Implement a federated learning framework so endpoints (e.g., classroom tablets) collaboratively improve the recognition model without sharing raw images—protecting sensitive biometric data by sharing only encrypted weight updates.

Anti-Spoofing and liveness Detection: Integrate texture analysis and micro-motion cues to distinguish live faces from photographs or video replays, safeguarding against presentation attacks

Bias Auditing: Regularly evaluate performance metrics (accuracy, false positives/negatives) across gender, age, and skin-tone strata to identify and mitigate algorithmic bias

Future Aspects

Integration of advanced Deep-Learning Models

- Adopt Lightweight CNNs and Vision Transformers: Replace or augment traditional feature-based methods (e.g., Eigenfaces, Fisherfaces) with compact convolutional neural networks (MobileNet, EfficientNet) or vision-transformer variants to boost accuracy under varied lighting and poses—while still enabling on-device inference.

- Hybrid Pipeline Architecture: Combine fast, classical face detection (e.g., OpenCV Haar cascades) with a secondary deep-learning re-identification stage to balance speed and precision in live classroom or office settings.

Dataset Expansion and Synthetic Augmentation

- Larger, More Diverse Training Sets: Scale beyond our initial 5-person dataset (=18k crops) by collecting images across multiple sessions, cameras, and demographics to reduce bias and improve generalization.

- GAN-Based Augmentation: Leverage generative adversarial networks to synthesize varied facial expressions, occlusions (masks, scarves), and lighting conditions—ensuring robust attendance capture even when subjects wear accessories or move dynamically.

Broader Applications and Commercialization

• Enterprise Time-Tracking Systems: Extend the attendance assistant to corporate environments,

integrating with HR systems for automated timekeeping and employee verification at entrances.

• Smart-campus and IoT Integration: Link attendance data with campus access control, library entry

logs, and canteen payments to create a unified student experience.

• Analytics Dashboard: Offer administrators real-time dashboards showing attendance trends, tardiness patterns, and automated alerts for absenteeism spikes.

Publication Details

International Journal on Science and Technology

E-ISSN: 2229-7677 • Impact Factor: 9.88

A Widely Indexed Open Access Peer Reviewed Multidisciplinary Bi-monthly Scholarly International Journal

Call for Paper Volume 16 Issue 2 April-June 2025 ✱ Submit your research before last 3 days of June to publish your research paper in the issue of April

Home | Research Paper | Editors & Reviewers | Current Issue | Publication Archive | Conference | Contact Us

Submit Research Paper

Thank you.

Your research paper is submitted.

Your paper will be reviewed and you will get notification message by SMS or email about its selection and acceptance for publication in our journal.

We have sent an email, containing pass code, to the specified 1st author's email address; using the paper id and the pass code, you can track your paper's status.

Submitted paper details:

Email Address	kpt.krishnaraj@gmail.com
Research Paper Id	4719
Research Paper Title	Machine Learning-Powered Facial Recognition-Based Attendance System

"IJSAT" <editor@ijsat.org>
to Krish <kpt.krishnaraj@gmail.com>

04:3

⚠ Download pictures or always download pictures from this sender. To preserve privacy, external content was not downloaded.

Dear Krishnaraj Thadesar,

Thank you for submitting your research paper.

It will be reviewed by one of our corresponding reviewer and then we will inform you about the status of the review.

You can also track status of the submitted research paper using the following paper id and pass code:

Submitted paper details:

Research Paper Id	4719
Pass Code	MjgzODA0
Research Paper Title	Machine Learning-Powered Facial Recognition-Based Attendance System

References

[1] Paul, Sanmoy and Acharya, Sameer Kumar, A Comparative Study on Facial Recognition Algorithms (December 21, 2020). e-journal - First Pan IIT International Management Conference – 2018, Available at SSRN: https://ssrn.com/abstract=3753064 or http://dx.doi.org/10.2139/ssrn.3753064

[2] Kaur, P., Krishan, K., Sharma, S.K. and Kanchan, T., 2020. Facial-recognition algorithms: A literature review. Medicine, Science and the Law, 60(2), pp.131-139.

[3] Kukula EP, Elliott SJ. Evaluation of a facial recognition algorithm across three illumination conditions. IEEE Aerospace and Electronic Systems Magazine. 2004 Sep;19(9):19-23.

[4] Kukula EP, Elliott SJ. Evaluation of a facial recognition algorithm across three illumination conditions. IEEE Aerospace and Electronic Systems Magazine. 2004 Sep;19(9):19-23.

[5] Emami S, Suciu VP. Facial recognition using OpenCV. Journal of Mobile, Embedded and Distributed Systems. 2012 Mar 30;4(1):38-43

References

[6] Chen J, Jenkins WK. Facial recognition with PCA and machine learning methods. In2017 IEEE 60th international Midwest symposium on circuits and systems (MWSCAS) 2017 Aug 6 (pp. 973-976). IEEE.

[7] Schenkel T, Ringhage O, Branding N. A Comparative Study of Facial Recognition Techniques: With focus on low computational power.

[8] Paul, S. and Acharya, S.K., 2020, December. A comparative study on facial recognition algorithms. In e-journal-First Pan IIT International Management Conference–2018.

[9] Delbiaggio, N., 2017. A comparison of facial recognition's algorithms.

[10] Coe, J. and Atay, M., 2021. Evaluating impact of race in facial recognition across machine learning and deep learning algorithms. Computers, 10(9), p.113.

[11] Dirin, Amir, Nicolas Delbiaggio, and Janne Kauttonen. "Comparisons of facial recognition algorithms through a case study application." (2020): 121-133

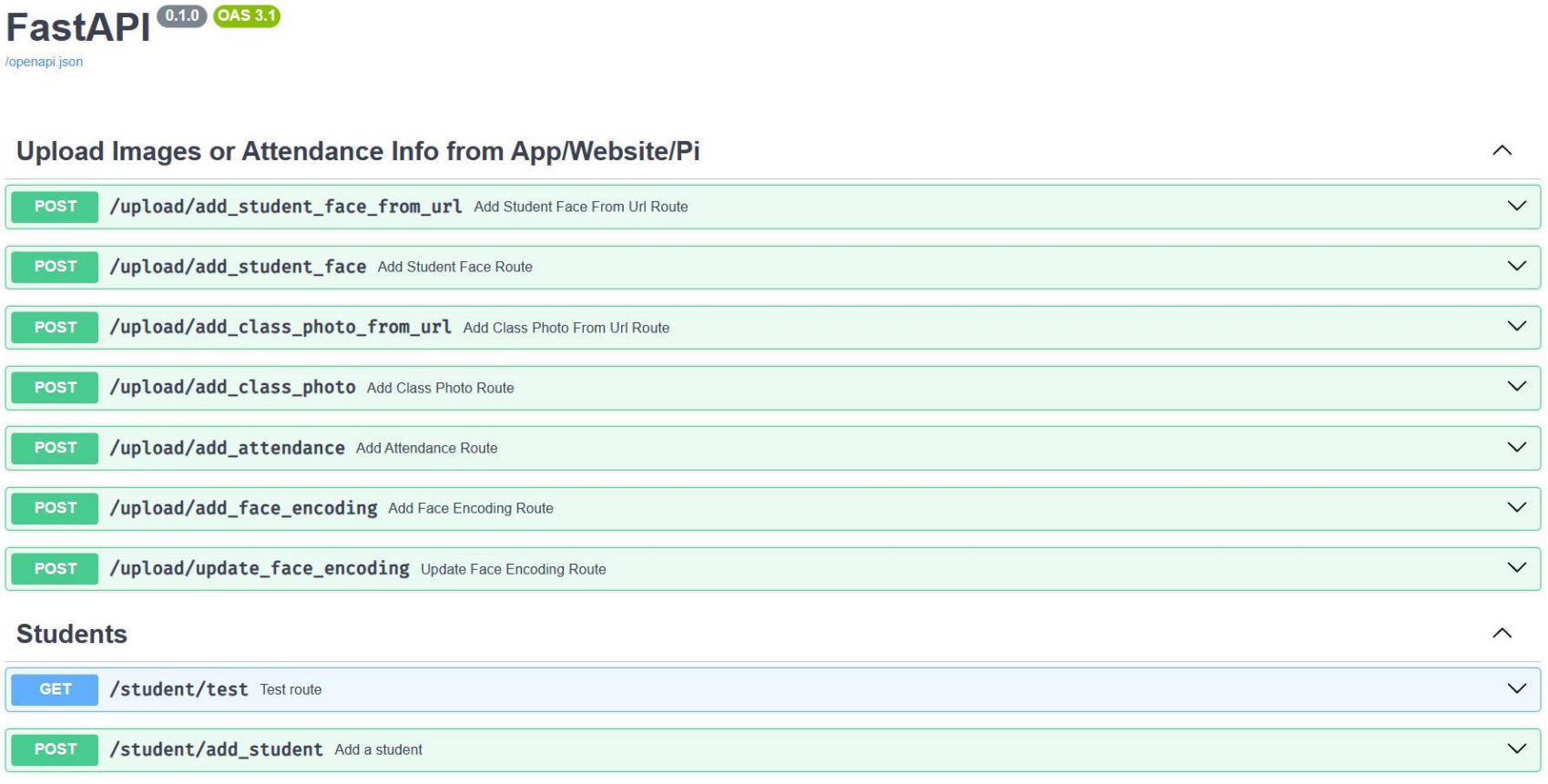
Individual Presentation 1

Krishnaraj Thadesar
PRN: 1032210888
Roll Number: 15
Panel: A

Aim

1. Design and implement the backend API and face-recognition engine for the Attendance-Assistant system.
2. Learn about various Machine learning models to recognize faces.

Contribution



Students

GET	/student/test	Test route
POST	/student/add_student	Add a student
POST	/student/get_student_from_panel_id	Get students from panel id
POST	/student/get_student_encoding	Get student encoding From student ID
GET	/student/get_all_students	Get all students

Face Recognition

GET `/face_rec/test` Test route

Panels, Schools and Specializations

GET `/panels/test` Test route

POST `/panels/add_panel` Add a panel

GET `/panels/get_all_panels` Get all panels

POST `/panels/add_school` Add a school

GET `/panels/get_all_schools` Get all schools

Individual Presentation 2

Parth Zarekar

PRN: 1032210846

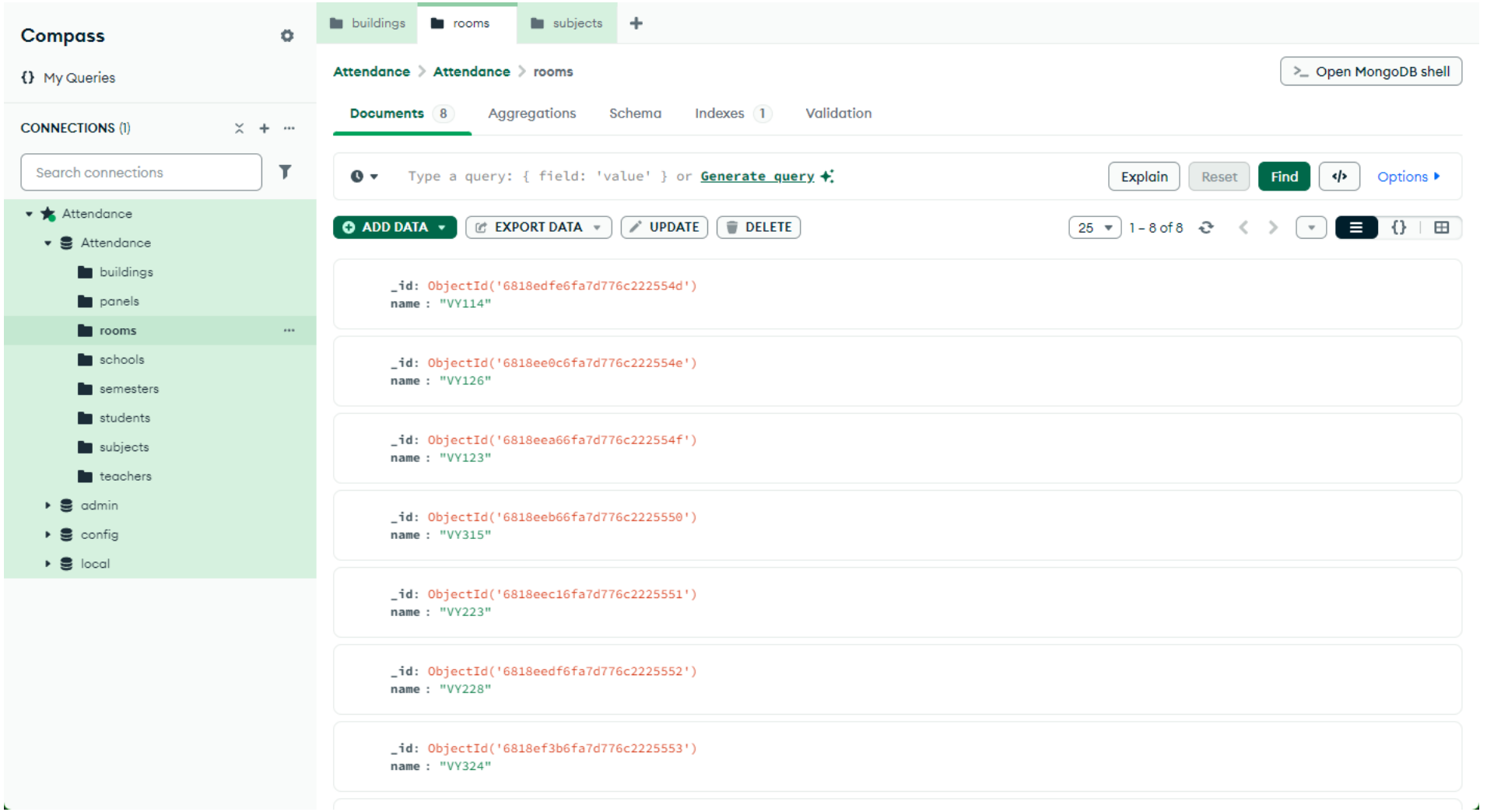
Roll Number: 09

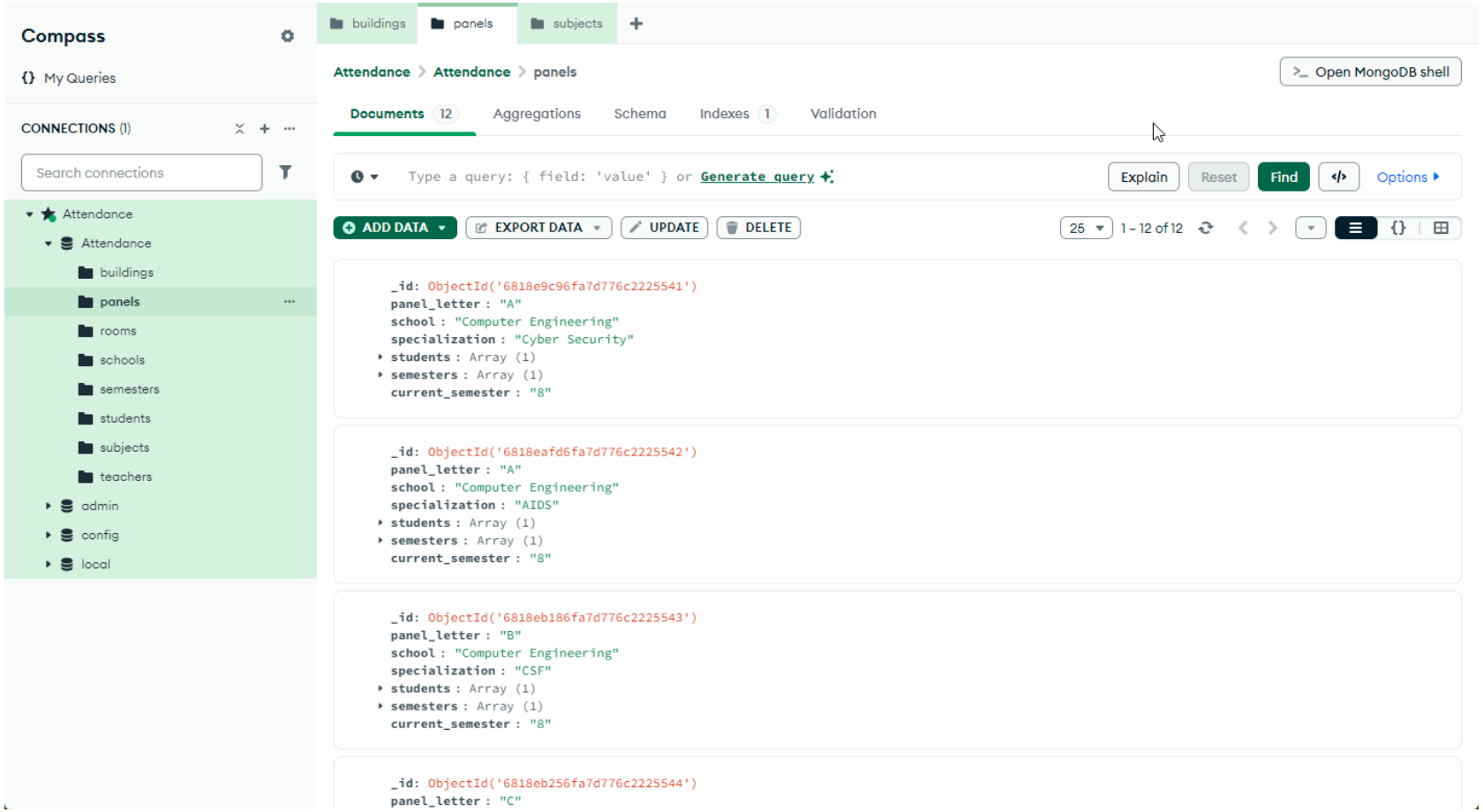
Panel: A

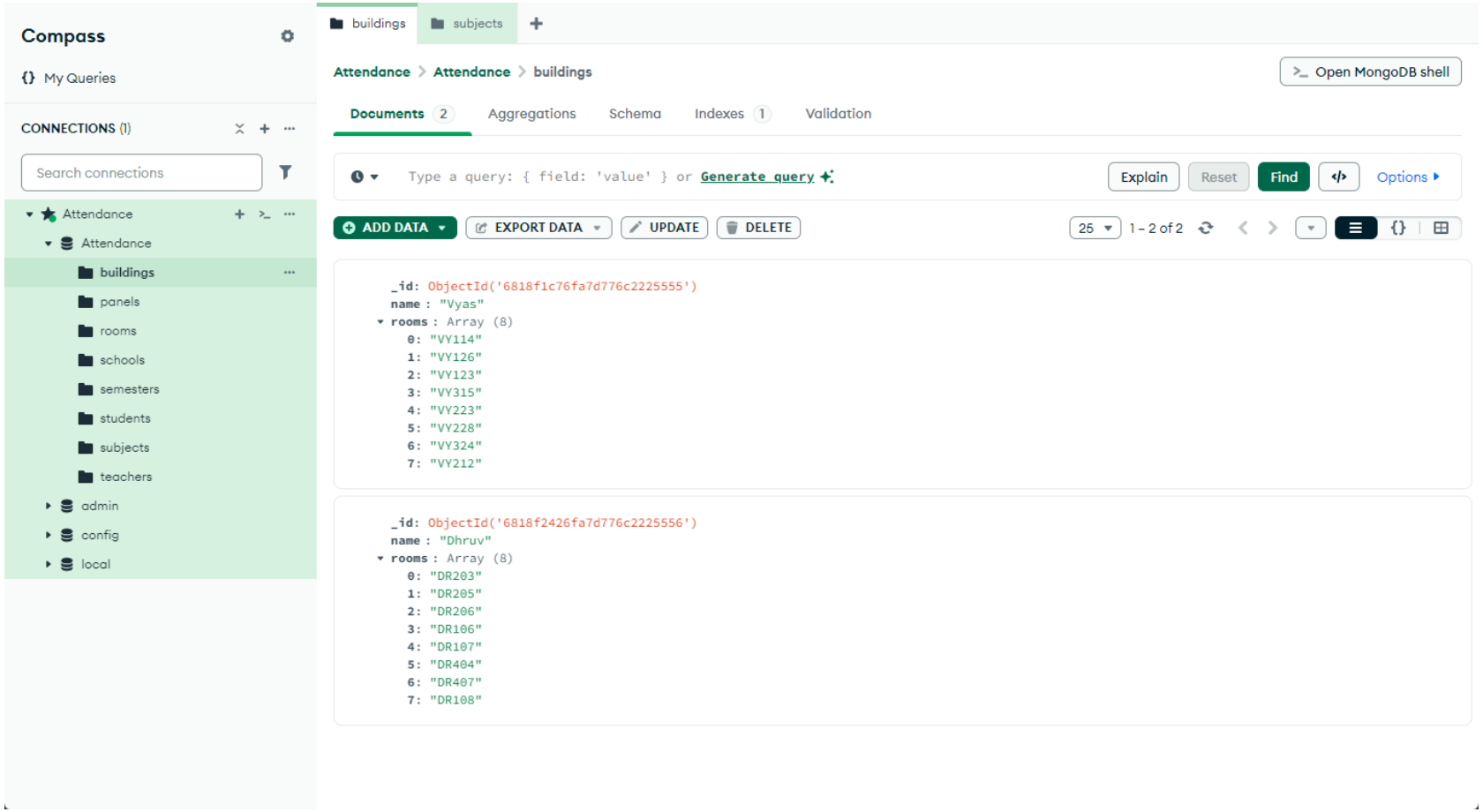
Aim

- Learn how to deploy large projects on MongoDB
- Integrate NoSQL databases in a live project.

Contribution




Compass
My Queries
CONNECTIONS (1)
Search connections
Attendance
Attendance
buildings
panels
rooms
schools
semesters
students
subjects
teachers
admin
config
local
buildings
panels
subjects
Attendance > Attendance > panels
Open MongoDB shell
Documents 12
Aggregations
Schema
Indexes 1
Validation
Type a query: { field: 'value' } or Generate query
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
25
1 - 12 of 12
_id: ObjectId('6818e9c96fa7d776c2225541')
panel_letter : "A"
school : "Computer Engineering"
specialization : "Cyber Security"
students : Array (1)
semesters : Array (1)
current_semester : "8"
_id: ObjectId('6818eafd6fa7d776c2225542')
panel_letter : "A"
school : "Computer Engineering"
specialization : "AIDS"
students : Array (1)
semesters : Array (1)
current_semester : "8"
_id: ObjectId('6818eb186fa7d776c2225543')
panel_letter : "B"
school : "Computer Engineering"
specialization : "CSF"
students : Array (1)
semesters : Array (1)
current_semester : "8"
_id: ObjectId('6818eb256fa7d776c2225544')
panel_letter : "C"


Compass
My Queries
CONNECTIONS (1)
Search connections
Attendance
Attendance
buildings
panels
rooms
schools
semesters
students
subjects
teachers
admin
config
local
buildings
subjects
Attendance > Attendance > buildings
Open MongoDB shell
Documents 2
Aggregations
Schema
Indexes 1
Validation
Type a query: { field: 'value' } or Generate query
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
25
1 - 2 of 2
_id: ObjectId('6818f1c76fa7d776c2225555')
name : "Vyas"
rooms : Array (8)
0: "VY114"
1: "VY126"
2: "VY123"
3: "VY315"
4: "VY223"
5: "VY228"
6: "VY324"
7: "VY212"
_id: ObjectId('6818f2426fa7d776c2225556')
name : "Dhruv"
rooms : Array (8)
0: "DR203"
1: "DR205"
2: "DR206"
3: "DR106"
4: "DR107"
5: "DR404"
6: "DR407"
7: "DR108"

Individual Presentation 3

Sourab Karad

PRN: 1032211150

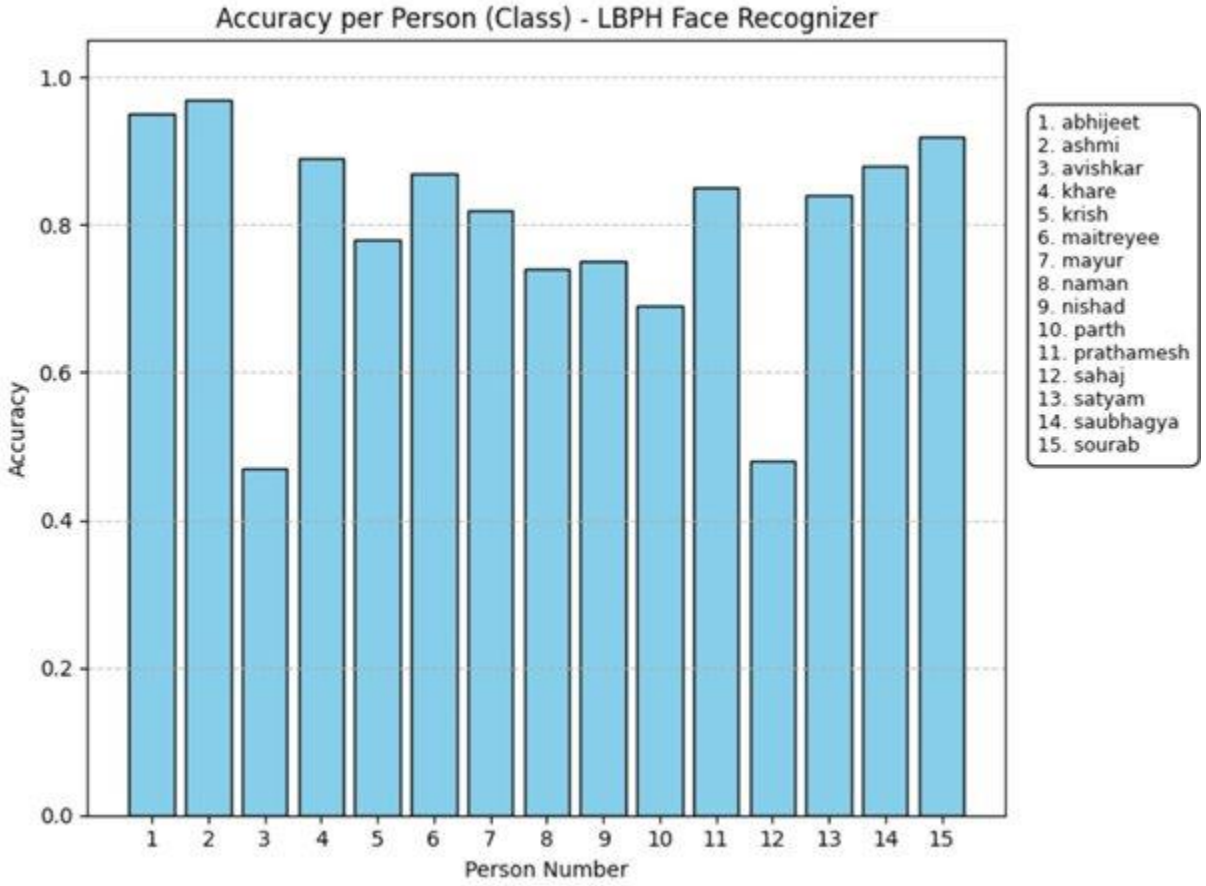
Roll Number: 40

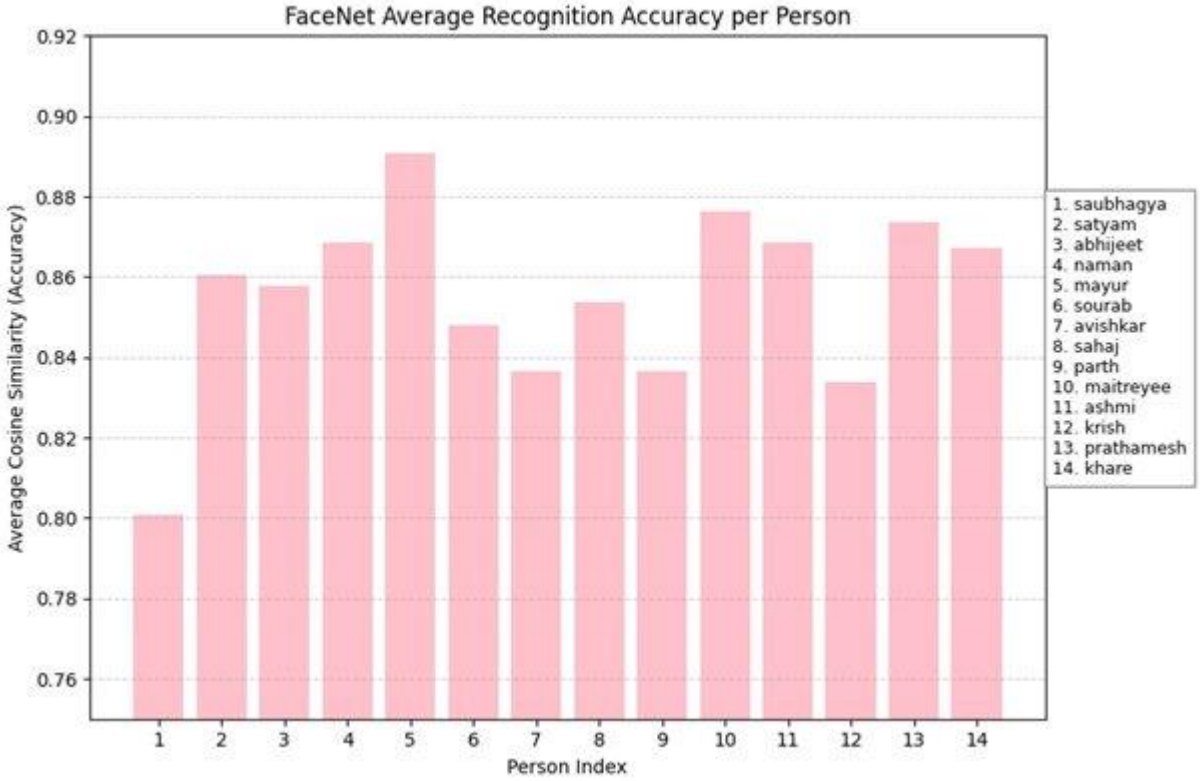
Panel: A

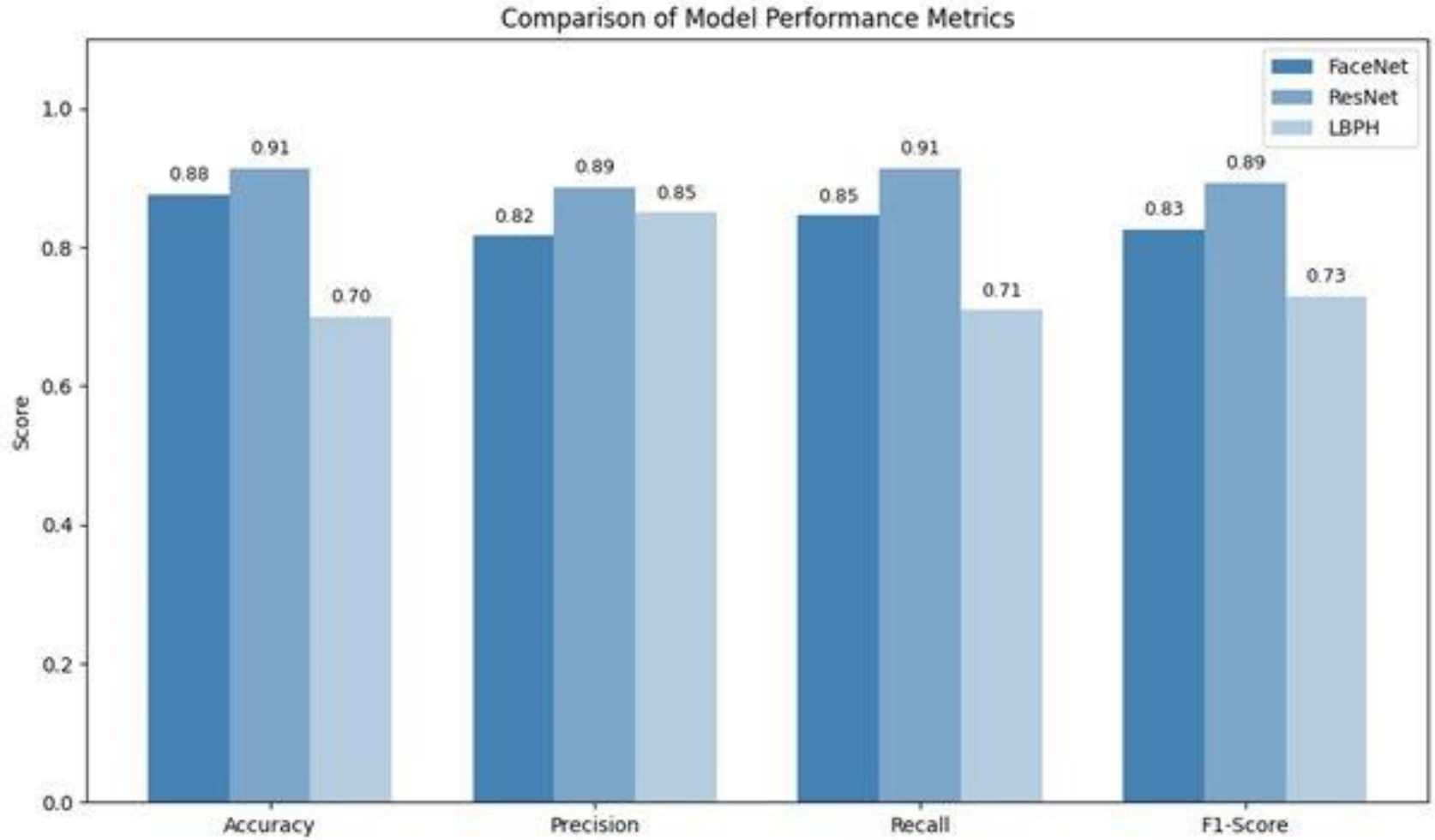
Aim

- Research the different types of algorithms used to detect and recognize faces.
- Compare each algorithm and train and test them on a database.

Contribution






Comparison of Model Performance Metrics
FaceNet
ResNet
LBPH
Score
1.0
0.8
0.6
0.4
0.2
0.0
0.88
0.91
0.70
0.82
0.89
0.85
0.85
0.91
0.71
0.83
0.89
0.73
Accuracy
Precision
Recall
F1-Score

Individual Presentation 4

Saubhagya Singh

PRN: 1032211144

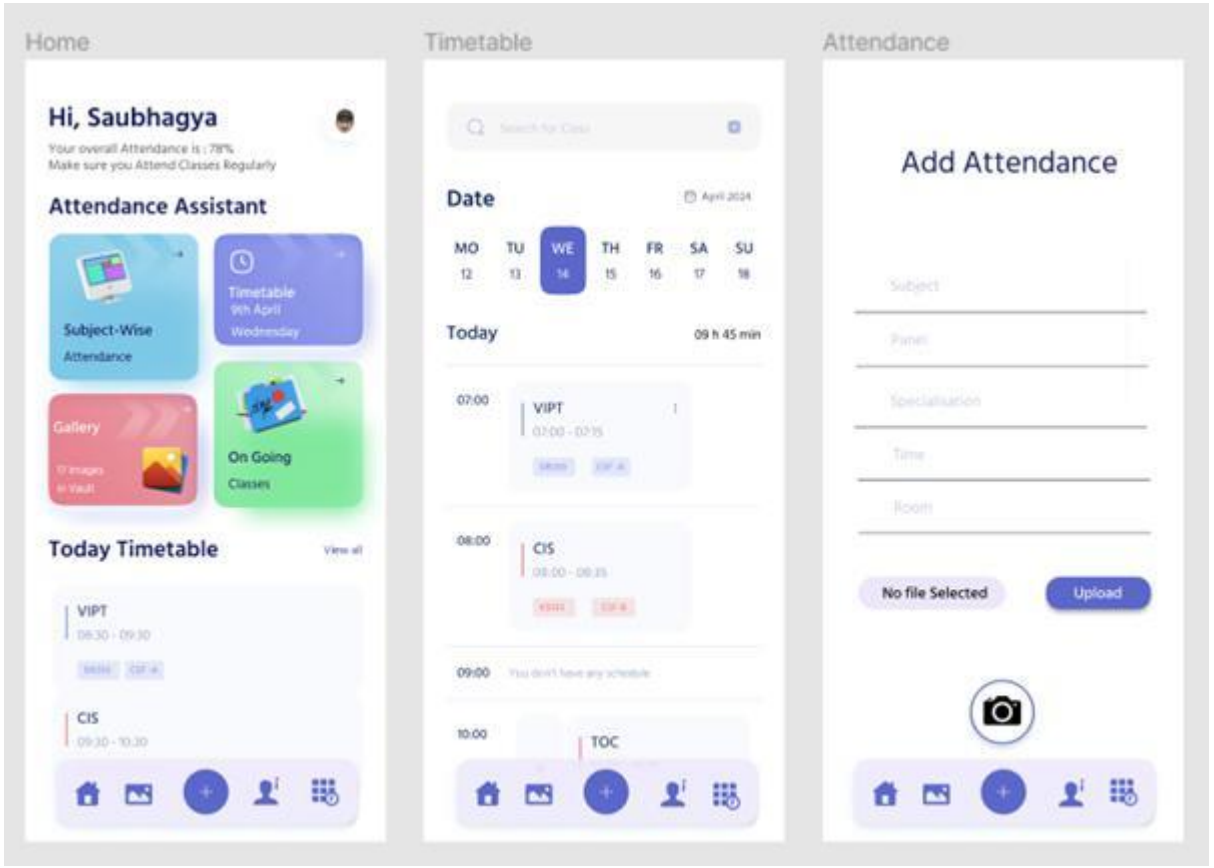
Roll Number: 38

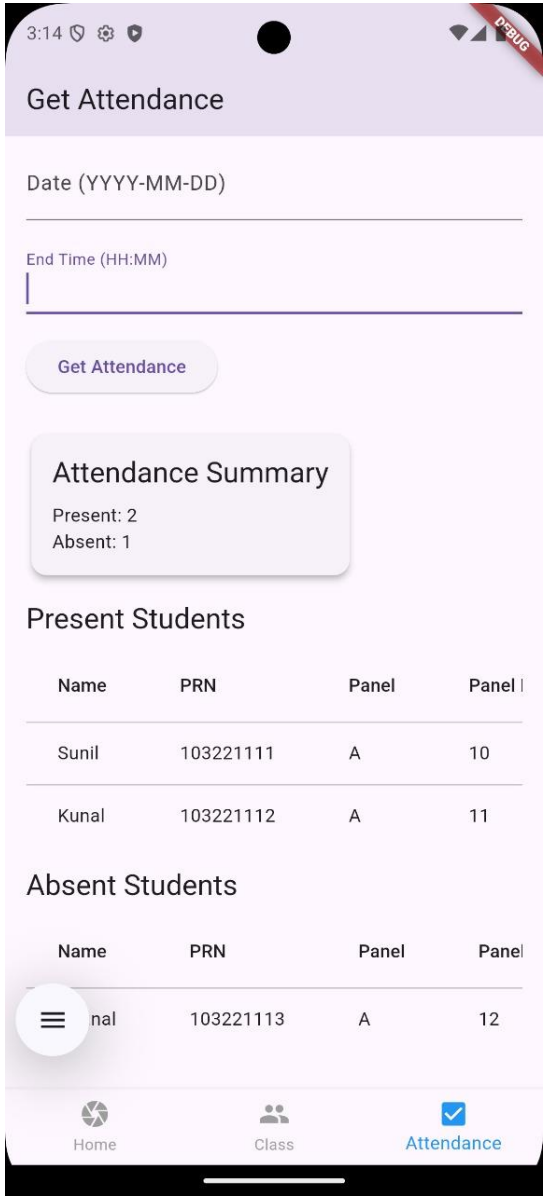
Panel: A

Aim

1. Develop a client side Application to access and perform Attendance management.
2. Integrate an app with backend and a database, with relevant API calls to understand the working of an entire full stack application.
3. Familiarity with Flutter

Contribution





Thank You!