**PROJECT TITLE**

**PROJECT SYNOPSIS**

OF 6 MONTHS INDUSTRIAL TRAINING

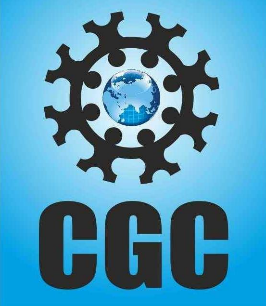
**BACHELOR OF TECHNOLOGY**

## Branch

SUBMITTED BY

Name:

Roll No.



**CGC COLLEGE OF ENGINEERING, MOHALI**

## **CGC College of Engineering**

Project Synopsis

### Title page:

1. Name of Student(s)
2. PTU Roll No(s).
3. Branch
4. Batch
5. Proposed Topic

* **Introduction**

**Introduction to the Project:**

The Face Recognition Based Attendance Management System (FAMS) is a sophisticated solution designed to streamline the attendance tracking process in various educational and organizational settings. Developed using Python's Tkinter library for the graphical user interface (GUI) and incorporating computer vision techniques, FAMS offers efficient and accurate attendance management.

The system allows users to register students or employees by capturing their images along with their unique identifiers, such as enrollment numbers or names. These images are then processed and stored in a database for future recognition. Utilizing the OpenCV library, FAMS employs the LBPH (Local Binary Patterns Histograms) Face Recognizer to train a model on the collected images, enabling it to recognize individuals during the attendance-taking process.

FAMS offers two primary modes of attendance tracking: automatic and manual. In the automatic mode, the system uses the trained model to detect and recognize faces in real-time, recording attendance with timestamps. The manual mode provides flexibility for users to input attendance data manually, offering an alternative for situations where face recognition may not be feasible or necessary.

Furthermore, FAMS includes features for administrators to manage student or employee records, view attendance reports, and perform administrative tasks through a secure login panel. The system's intuitive interface and robust functionality make it a valuable tool for organizations seeking efficient attendance management solutions while leveraging the power of facial recognition technology.

* **Technology Used:**
* **Python:** The core programming language used for developing the system.
* **OpenCV (Open Source Computer Vision Library):** OpenCV is used for image processing tasks such as face detection and recognition.
* **Tkinter:** Tkinter is used for creating the graphical user interface (GUI) of the application.
* **PIL (Python Imaging Library):** PIL is used for image processing tasks such as loading and saving images.
* **pandas:** Pandas is a data manipulation and analysis library used for handling data related tasks.
* **NumPy:** NumPy is used for numerical computing tasks.
* **datetime:** The datetime module is used for handling date and time related operations.
* **csv:** The csv module is used for reading and writing CSV files.
* **pymysql:** PyMySQL is a Python library used to connect with MySQL databases.
* **Key Features:**
* **Taking Images for Datasets:** Users can input their enrollment numbers and names, and the system captures their images to create a dataset for training the face recognition model.
* **Training the Model:** Once the dataset is created, the system trains a face recognition model using the captured images.
* **Automatic Attendance:** The system can automatically mark attendance by recognizing faces in real-time using the trained model.
* **Manually Fill Attendance:** Users can manually fill attendance for specific subjects by entering enrollment numbers and names.
* **Admin Panel:** There's an admin panel for user authentication and checking registered students.
* **Face Recognition:**The OpenCV library is used for face detection and recognition.

LBPH (Local Binary Patterns Histograms) algorithm is utilized for face recognition.

Haar cascades are used for detecting faces in images.

* **Error Handling:**The system includes error screens to handle scenarios where users forget to input enrollment numbers, names, or subject names.
* **Security:** There's a login feature for an admin panel to ensure security and restrict access to sensitive functionalities.
* **User Interaction:** Users can clear input fields, receive notifications, and check attendance sheets through the GUI.
* **File Management:** The system handles file operations such as saving images, training the model, and storing attendance data in CSV files.
* **Integration with External Tools:** The system integrates with external tools like MySQL databases for storing attendance data.
* **Platform Dependency:**The system's paths and file operations are platform-dependent, which means they might need adjustment when deployed on different systems.
* **Feasibility Study**:

A feasibility study is a comprehensive analysis conducted to evaluate the practicality and viability of a proposed project, system, or business venture. It serves as a crucial step in the decision-making process, helping stakeholders assess the potential benefits, risks, and challenges associated with pursuing a particular course of action.

1. **Technical Feasibility:**

* The system uses common technologies like Python, OpenCV, and Tkinter for GUI development, which are well-supported and have extensive documentation.
* It relies on facial recognition algorithms provided by OpenCV, which are widely used and well-tested.
* The system requires a webcam for capturing images, which is a standard feature available on most modern computers.

1. **Operational Feasibility:**

* The system is user-friendly and designed to be operated by non-technical users such as teachers or administrative staff.
* It provides clear instructions and feedback during different stages of operation, such as image capture, training, and attendance marking.

1. **Economic Feasibility:**

* The system uses open-source libraries and tools, which significantly reduces the cost of development and deployment.
* It requires standard hardware components like a webcam, which are readily available and affordable.
* There may be some initial investment required for training personnel and setting up the system, but the long-term operational costs are minimal.

1. **Schedule Feasibility:**

* The system development timeline depends on factors such as the complexity of customization required, the availability of resources, and the testing phase.
* Developing the basic functionality of the system should be achievable within a reasonable timeframe, but additional features or customization may extend the development schedule.

1. **Legal and Ethical Feasibility:**

* The system must comply with legal regulations regarding data privacy and security, especially when dealing with sensitive information such as student data.
* Ethical considerations include obtaining consent from individuals before capturing and processing their facial images, as well as ensuring that the system is not used for unauthorized surveillance or tracking.
* **Methodology/ Planning of work**

Face Recognition Based Attendance Management System implemented using Python with the help of the OpenCV library for computer vision tasks and the Tkinter library for creating the graphical user interface (GUI).

Here's an overview of the methodology and planning based on the code:

1. **User Interface (UI):**

The system provides a graphical user interface (GUI) for users to interact with.

It includes options for taking images of students, training the recognition model, filling attendance automatically, and manually filling attendance.

1. **Functionality:**

* **Taking Images:** Users can input student enrollment numbers and names, and the system captures images of the students' faces. These images are stored in a directory for training the recognition model.
* **Training Images:** The system trains a face recognition model using the captured images of the students. It uses the LBPH (Local Binary Patterns Histograms) algorithm for face recognition.
* **Automatic Attendance:** Once trained, the system can automatically recognize students' faces and mark their attendance based on pre-trained data. It detects faces using a webcam and matches them with the pre-trained model.
* **Manually Fill Attendance:** There's an option for users to manually fill attendance. They can input subject names and then enter student enrollment numbers and names to mark their attendance.

1. **Data Management:**

* The system uses CSV files to store student details and attendance records.
* It creates CSV files for storing manually filled attendance records.
* Attendance records are also stored in a database, with tables created dynamically based on the subject and date.

1. **Error Handling:**

The system includes error handling mechanisms to prompt users if they miss providing necessary information, such as subject names or student details.

1. **Administrative Panel:**

There's an administrative panel where users can log in using a username and password. Once logged in, they can check registered students' details.

1. **Integration and Deployment:**

* The system seems designed to be deployed locally on a machine, as it interacts with local directories and webcam resources.
* It integrates the functionality of capturing images, training models, and managing attendance seamlessly through the GUI.

1. **Backend Operations:**

* The program uses OpenCV for face detection and recognition tasks.
* It interacts with the file system to save captured images, train the model, and save attendance records.
* Database operations are performed using MySQL to store attendance data.

### Database Integration:

### The program integrates with a MySQL database to store student details and attendance records.

### Tables are created dynamically for storing attendance records.

### File Handling:

### The program performs file handling operations to manage captured images, training data, and attendance records.

### Modular Approach:

### The code is structured into functions for better modularity and readability.

### Each functionality, such as taking images, training, and filling attendance, is encapsulated within separate functions.

### Facilities required for proposed work

### The proposed work seems to be an Attendance Management System based on Face Recognition technology.

1. **Hardware Requirements:**

* **Computers/Laptops:** You need computers or laptops to run the software and perform tasks like image processing, face recognition, database management, etc.
* **Webcam/Camera:** To capture images for face recognition.
* **Internet Connectivity:** For database management, updates, and possibly remote access to the system.

1. **Software** **Requirements:**

* **Operating System:** The system should run on an operating system compatible with the software you are using (e.g., Windows, Linux, macOS).
* **Programming Languages and Libraries:** You are using Python with libraries like OpenCV, tkinter, numpy, and pandas for image processing, GUI development, data manipulation, and face recognition.
* **Database Management System (DBMS):** You are using a MySQL database to store student information and attendance records.
* **Text Editor/IDE:** You need a text editor or an Integrated Development Environment (IDE) to write and edit your code. Examples include PyCharm, VS Code, Sublime Text, etc.

1. **Training Data:**

* **Images Dataset:** You need a dataset of images of students' faces for training the face recognition model.
* **Labeled Data:** Each image should be labeled with the corresponding student's identity (e.g., Enrollment ID, Name).

1. **Physical Setup**:

* **Proper Lighting:** Ensure sufficient and consistent lighting conditions where the face recognition will take place.
* **Camera Placement:** Cameras should be positioned at appropriate angles and heights for optimal face detection and recognition.
* **Privacy Considerations:** Ensure that the placement of cameras respects privacy norms and regulations.

1. **Database Management:**

* **Database Server:** You need a server to host your MySQL database.
* **Backup and Recovery Plan:** Implement a backup and recovery plan to prevent data loss in case of system failure or corruption.

1. **User Interface:**

**Graphical User Interface (GUI):** You have developed a GUI using tkinter for user interaction, which allows users to perform tasks like taking images, training the model, filling attendance manually, etc.

1. **Security Measures:**

* **Access Control:** Implement access control mechanisms to ensure that only authorized users can access sensitive functionalities like admin panels, attendance records, etc.
* **Data Encryption:** Implement data encryption techniques to secure data transmission and storage.

1. **Documentation and Support:**

* **User Manual:** Provide documentation for users on how to use the system effectively.
* **Technical Support:** Offer technical support to troubleshoot issues and provide assistance to users.

### Bibliography

Face Recognition Based Attendance Management System using Tkinter for the graphical interface and OpenCV for face detection and recognition. The system offers several functionalities: capturing images of students for dataset creation, training a face recognition model, taking attendance automatically through facial recognition, and manually filling attendance for subjects.

Upon running the application, users can input student details such as enrollment number and name, and then take their images for dataset creation. The images are stored in a designated folder along with the student details in a CSV file. Subsequently, users can train the face recognition model using the captured images.

The system also allows for automatic attendance taking based on recognized faces. It recognizes faces in real-time using a trained model and records attendance by matching recognized faces with the stored dataset. The attendance records are saved in CSV files for each subject and date.

Additionally, there's a provision for manually filling attendance. Users can enter subject names and input student enrollment numbers and names to manually mark attendance. The system prompts users to enter subject names and student details, and then records the attendance in CSV files.

The application includes an admin panel for checking registered students, ensuring security with a login feature. Overall, this system provides a comprehensive solution for managing attendance using face recognition technology.