

```
=====
=====
=====

READ FOR INFORMATION
=====
=====
=====
```

Project Description:

.....

This project is made by Malhar Girgaonkar of CSE A 2nd year roll number 21J41A0537 from Malla Reddy College Of Engineering (MREC) in 2023 from Hyderabad,Telangana,India.

This application is aimed to display my skills and its application on topics

- AI
- OpenCV
- CustomTkinter
- Python
- Regular Expression
- DBMS application
- DBMS queries and server capabilities
- Python module mysql.connect and sqlite3
- Python GUI

-Special expertise in modules like os,shutle,subprocess,Cv2,CustomTkinter,etc.

This is my second year project to be considered as a proof of my skills for grading,internship,placement and as a proof of merit.

This projects main aim is to perform face recognition on a varied input by user and provide output with bounding box for one or more face detected in the input.As a plus we also provide databse access and server connection facility.

```
=====
=====
=====

Usage:
```

.....

1. Clone the repository.
2. Install the required dependencies using `pip install -r requirements.txt`.
3. Run the main script: `python Login.py`.

```
=====
=====
=====

Features:
```

.....

- Login and Registration facility
- Face Recognition in images.
- Face Recognition in videos.
- Face Recognition through Webcamara

=====

Files and Directories:

.....

- 2-2 project:

-app data:

- Database:

-(stuff related to database and mysql)

-databasecreatelite.py=(code to create sqlite3

userinfo.db)

-Dependencies:

-harcascade_frontalface_default.xml.=(it is the face
recognition algorithm i am using)

-databasedependencies.py=(It is a program that creates
required database and table if they do not exist)

-requirements.py=(It is a program that checks if
required modules or packages are installed and if not installs them)

-icons:

-(Icons that can or are used in application)

-images:

-(This directory contains images that you will load to
application to do action on)

-videos:

-(This directory contains videos that you will load to
application to do action on)

-Aboutapp.py=(Contains information about application)

-ImageFaceRecogniton.py=(Contains code for gui and image based face
recognition)

-Login.py=(It has code for gui and login functionality)

-mainpage.py=(It has code for gui and mainpage to acess whole
applications features)

-mysqlconn.py=(it has mysql connection features and code snippet to
access,create table,delete,truncate,select database)

-Registration.py=(It has code for gui and Registration
functionality)

-sqlite3conn.py=(It has code to interract with sqlite3 database
directly)

-userinfo.db=(It is the databse i am using in place of mysql and it
uses sqlite3)

-VideoFaceRecogniton.py=(Contains code for gui and Video based face recognition)

-WebcamFaceRecogniton.py=(Contains code for gui and Webcam based face recognition)

=====
=====
=====

Database Structure:

--Database Details:

- Name:userinfo
- If using mysql.connect
 - Host: localhost
 - Username: root
 - Password: root
- if using sqlite3 no need of user credentials
- Tables:"userlogin","userpersonal","usercontact"

--Table Details: userlogin

-Primary Key: username

-Attributes:

- username (VARCHAR, 20)
- password (VARCHAR, 20)

--Table Details: userpersonal

-Primary Key: username

-Foreign Key: username (references userlogin.username)

-Attributes:

- username (VARCHAR, 20)
- firstname (CHAR, 20)
- lastname (CHAR, 20)
- gender (ENUM: 'Male', 'Female', 'Transgender', 'Others')
- dateofbirth (VARCHAR, 10)
- country (CHAR, 20)
- state (CHAR, 20)
- city (CHAR, 20)

--Table Details: usercontact

-Primary Key: username

-Foreign Key: username (references userlogin.username)

-Attributes:

- username (VARCHAR, 20)
- email (VARCHAR, 40)
- icc (ENUM: '+91', '+1', '+44', '+81')
- mobilenno (INT, 10)
- tandc_status (ENUM: 'on', 'off')

=====
=====
=====

Configuring:

.....
My project has following dependencies

PYTHON MODULES:

- CTkMessagebox
- customtkinter (and its submodules)
- tkinter (and its submodules)
- mysql.connector
- subprocess
- PIL (Image, ImageTk)
- cv2 (OpenCV)
- os
- shutil
- importlib
- sqlite3

ALGORITHM:

- harcascade_frontalface_default.xml
- details:

-The Haar Cascade Classifier is a machine learning object detection algorithm used to detect objects in images or video frames.

-It is primarily used for detecting faces but can be trained to detect other objects as well.

-The algorithm uses a set of trained Haar-like features to detect patterns in the input data.

Advantages:

-Speed: Haar Cascade is relatively fast and can achieve real-time processing, making it suitable for applications like real-time face detection in videos.

-Accuracy: It can achieve decent accuracy, especially when trained and fine-tuned for specific use cases.

-Lightweight: The trained classifier is lightweight, requiring less computational resources compared to more complex deep learning models.

-Simple Implementation: Implementing Haar Cascade is straightforward, and libraries like OpenCV provide easy-to-use functions for integration.

Disadvantages:

-Limited Complexity: Haar Cascade might struggle with detecting complex patterns or objects that have varying orientations, lighting conditions, or occlusions.

-Training Effort: Training a custom Haar Cascade classifier requires a substantial amount of positive and negative images and is more involved compared to using pre-trained models.

-False Positives/Negatives: Achieving high accuracy requires careful parameter tuning and training. False positives (detecting an object that isn't there) and false negatives (failing to detect a present object) can occur.

-Not Suitable for All Objects: While Haar Cascade works well for faces and certain objects, it might not be suitable for detecting objects with intricate textures or irregular shapes.

=====
=====
=====

License:

.....
This project is open source License.

Feel free to use it at your will :)
A thanks in heart is all i need ;)

=====
=====
=====

Contact:

.....

For any questions or feedback, please contact:

- Email: malhargirgaonkar@gmail.com
- Website: ...