Documentation for this Project

1) Go to the Project File.

2) Arrange the folder in following order.

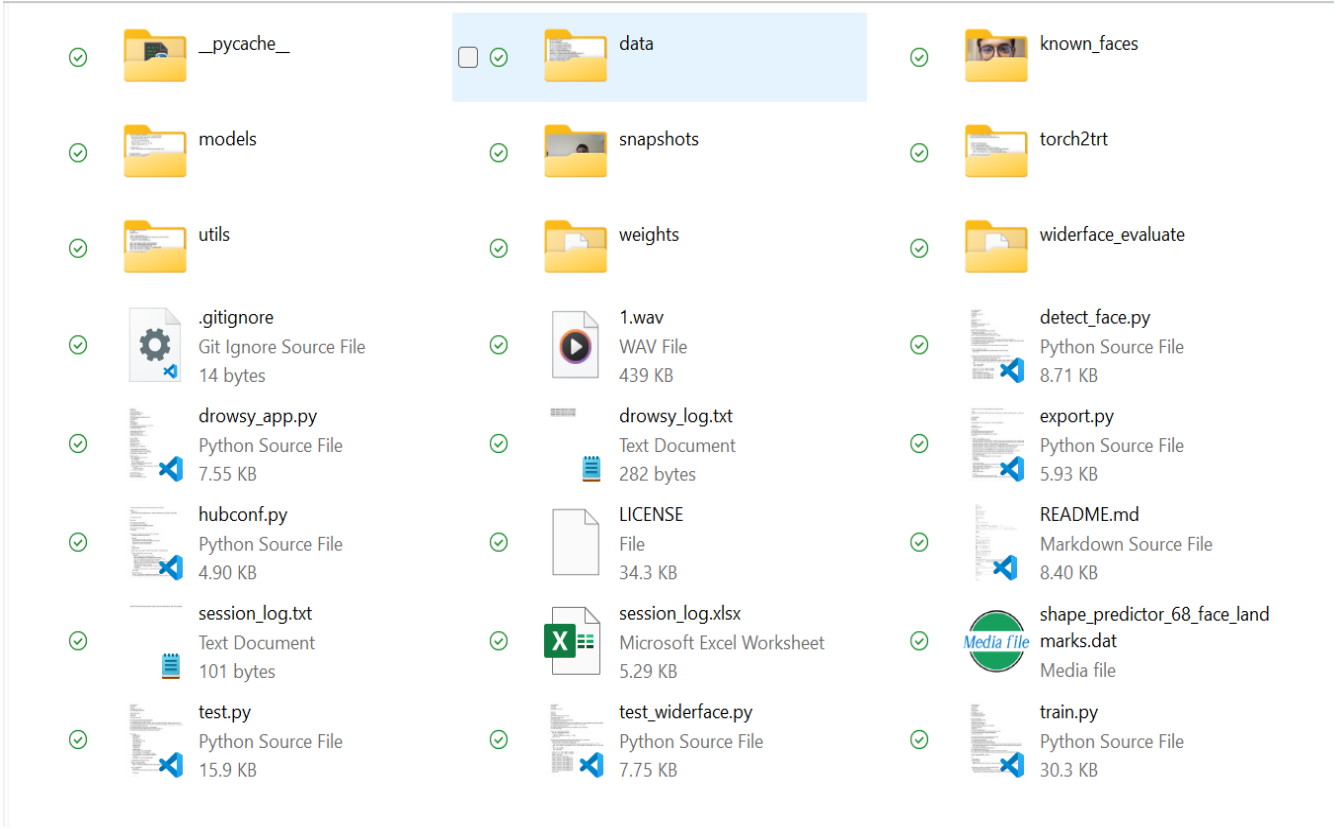
Folder Structure Overview

plaintext

Copy Edit

```
DriverDrowsinessDetection/
│
├── __pycache__/                    # Python cache files
├── data/                           # Dataset configuration files or annotations
├── known_faces/                    # Reference images of registered drivers
├── models/                         # YOLOv5-face models and architecture definitions
├── snapshots/                      # Captured drowsiness event images with timestamps
├── torch2trt/                      # Optional tools for TensorRT optimization (if applicable)
├── utils/                          # Utility scripts (EAR calculation, GUI helpers, etc.)
├── weights/                        # Pre-trained YOLOv5-face weight files (.pt)
├── widerface_evaluate/            # Wider Face dataset evaluation tools
├── .gitignore                      # Git tracking exclusions
├── 1.wav                           # Sound file played on drowsiness alert
├── detect_face.py                 # Script for running YOLOv5-face detection
├── drowsy_app.py                  # Main application file that integrates detection, recognitio
├── drowsy_log.txt                 # Text log of drowsiness detection events
├── export.py                      # Model export script (if converting to ONNX/TensorRT)
├── hubconf.py                     # Model hub configuration (for PyTorch loading)
├── LICENSE                         # License information
├── README.md                       # Project documentation
├── session_log.txt                # Text format event log
├── session_log.xlsx               # Excel file storing EAR, timestamp, and driver identity
├── shape_predictor_68_face_landmarks.dat  # Dlib landmark model for facial feature extraction
├── test.py                         # Model testing script
├── test_widerface.py              # YOLOv5-face testing on Wider Face dataset
├── train.py                        # Training script (if model retraining is required)
```

OR,



3) Open *drowsy_app.py* file

4) Open new terminal in vs code.

5) Type python *drowsy_app.py* and hit enter to run it.

6) It will take some time to open the webcam.

7) Then finally you can able to detect the person is drowsy or awake.

8) You can see the drowsy snapshots in snapshots folder and person details in session_log.xlsx file.

9) You can also add the number of known face to detect the name of an individual in known_faces folder.

10) You can check out the video named as Project execution.mp4