**ONLINE PROCTOR SERVICE**

***A Main Project Report submitted in partial fulfillment of the***

***Requirements for the award of degree of***

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted By**

**SIVANI DWARAMPUDI (17A91A0573)**

**SUNEEL VARMA (17A91A0581)**

***Under the esteemed guidance of***

**K.DEVI PRIYA, M.Tech, (Ph.D.)**

**Sr. Assistant Professor**

****

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**ADITYA ENGINEERING COLLEGE (A)**

**Approved by AICTE, permanently affiliated to JNTUK & Accredited by NBA, NAAC with ‘A’ Grade**

**Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956**

**Aditya Nagar, ADB Road - Surampalem – 533437, E.G.Dist., A.P.,**

**2016-2020**

**ADITYA ENGINEERING COLLEGE (A)**

**Approved by AICTE, permanently affiliated to JNTUK & Accredited by NBA, NAAC with ‘A’ Grade**

**Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956**

**Aditya Nagar, ADB Road - Surampalem – 533437, E.G.Dist., A.P.,**

**2017-2021**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

****

**CERTIFICATE**

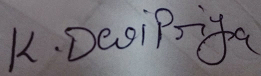
This is to certify that the Seminar work entitled *“***ONLINE PROCTOR SERVICE”** is being submitted by

**SIVANI DWARAMPUDI (17A91A0573)**

**SUNEEL VARMA (17A91A0581)**

In partial fulfillment of the requirements for award of the B. Tech degree in Computer Science and Engineering for the academic year 2020-2021.

**Project Guide Head of the Department**

****

Mrs.K.DeviPriya, M.Tech, (Ph.D) Mrs.A.Vanathi, M.E.,(Ph.D).,

Sr. Assistant Professor, Associate Professor,

Department of CSE Department of CSE

**External Examiner**

**DECLARATION**

We hereby declare that the project entitled **“ONLINE PROCTOR SERVICE”** is a genuine project. This work has been submitted to the **ADITYA ENGINEERING COLLEGE,** Surampalem affiliated to **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA** in partial fulfillment of the **B.Tech**degree**.** I/We further declare that this project work has not been submitted in full or part of the award of any degree of this or any other educational institutions.

**BY**

**SIVANI DWARAMPUDI (17A91A0573)**

**SUNEEL VARMA (17A91A0581)**

**ACKNOWLEDGEMENT**

It is with immense pleasure that we would like to express our indebted gratitude to our project supervisor **Mrs. K. DEVI PRIYA, Sr. Assistant Professor,** who has guided us a lot and encouraged us in every step of the project work, her valuable moral support and guidance throughout the project helped us to a greater extent.

We wish to thank our Project Coordinator **Mrs. K. DEVI PRIYA, Sr. Assistant Professor,** in the CSE Department for her support and valuable suggestions for the successful completion of this project.

Our deepest thanks to Our HOD **Mrs. A.Vanathi, Associate Professor, and Department of CSE** for inspiring us all the way and for arranging all the facilities and resources needed for our project.

We wish to thank our Dean academics, **Dr. S. Rama Sree**for her support and suggestions during our project work.

We owe our sincere gratitude to our Principal **Dr. M. Sreenivasa Reddy,** for providing a great support and for giving us the opportunity for doing the project.

We are thankful to our **College Management** for providing all the facilities in time to us for completion of our project.

Not to forget, **Lab Technicians, non-teaching staff and our friends** who have directly or indirectly helped and supported us in completing our project in time.

**ABSTRACT**

Online proctor service furnishes the path to identify student attentiveness towards class by monitoring them based upon the student's level of distractedness during the class and brings forth the advantages of conducting online exams without any malpractice. Detects a student's interest during the class based upon level of distraction, identifies the students who are indulging in committing malpractice during an assessment. Detects multiple individuals during an assessment, detects distraction by monitoring the frequency of user attempts to look away from the screen, captures when a user has crossed the average threshold of distractions and uploads the captured content to a cloud service

The proposed “**Online Proctor Service”** providesa way to monitor the student in various aspects few of them include malpractice detection and attentiveness detection, interestedness detection. This application is developed using Python in collaboration with Django and usage of few algorithms like 68 landmarks and Haarcascade frontal face for assessing user face and a deep learning model like yolo which will be trained to detect mobile devices. Scrutiny becomes easier and more accurate due to usage of advanced technologies and trained dataset. Several features like yawn count, blink count, gaze tracking, multiple individuals' detection, mobile phone detection was measured as a part of this project. Real time monitoring helps in easily identifying individuals who involve in forbidden activities. ­­Whenever, student cross minimum threshold of committing malpractice then that particular student data is captured using web camera and uploaded to the cloud service.

­­­­

**INDEX**

**Ch No Content Page**

**Abstract 5**

|  |
| --- |
| **1. Introduction** |
| 1.1 Existing System 08 |
| 1.2. Proposed System 09 |
| **2. Requirements Analysis** |
| 2.1. H/w & S/w Requirements 11 |
| 2.2. Software Requirements Specification  11 |
| **3. System Design** |
| 3.1. UML Diagrams 17 |
| **4. System Implementation** |
| 4.1. Selected Software 21 |
| 4.2. Training Model 30 |
| 4.3. Sample Code Snippets 33 |
| **5. Testing** |
| 5.1. Introduction of Testing 38 |
| 5.2. Test cases 39 |
| **6. Screenshots and Reports** |
| 6.1. Screenshots 45 |
| 6.2. Reports 56 |
| **7. Conclusion & Future Scope** |
| 7.1. Conclusion 59 |
| 7.2. Future Scope 59  **8. Bibliography**57 |
|  |
|  |

1. **INTRODUCTION**

This pandemic has created the demand for virtual communication all across the world every organization out there is shifting towards providing online services, and the education sector is one among them. The need for virtual communication in the educational sector has emerged the necessity for conducting exams and classes virtually. Online proctoring service helps in monitoring illicit activities performed by the student during the examination. It aids in finding out attentiveness of students during the online class. The main aim is to provide a comfortable, secure environment for students during the course of examination. Online proctor service makes the supervision easier due to the usage of advanced technologies in its design. It uses cutting-edge technologies like image processing, machine learning in order to bring forth both institution and student friendly service.

**1.1 Existing System:**

In the existing system, students used to go to the institution to attend classes and give examinations manually. It requires an external invigilator to monitor students. The existing system also requires an environment to organize classes. It is hard for external invigilator to monitor entire class room and it is not possible to manually monitor students during pandemic.

**1.1.1 Features:**

* Students can have interactive sessions
* Pen and paper based exam.

**1.1.2 Disadvantages:**

* Conducting classes or examinations in an offline mode is not possible in the current scenario.
* Students and management are at an equal level of risk of being attacked by the virus.
* As people come from different places there is a high possibility of virus spread even though the restrictions are strictly followed.
* As there is no proper mode of transportation as of now. It becomes tough for the student to come to the college to write exams and to attend classes.

## 1.2 Proposed System:

In our proposed model, Examinations are going to be conducted virtually(online based) and monitoring of student’s activity is done by using Image Processing and Machine Learning algorithms. This method overcomes the cost of conducting exams manually and it requires no external invigilator for surveillance. Scrutiny becomes easier and more accurate due to usage of advanced algorithms like 68 landmarks and Haarcascade frontal face for assessing user face. Apart from all of these a deep learning model was trained with dataset in order to detect mobile devices and multiple individuals during the spell of examination.

**1.2.1 Features:**

* Computer based examination where students are required to sit in front of the computer to give their exam.
* Students' facial movements will be recorded from time to time through the integrated camera present in the system.
* Any sort of malpractice will be closely monitored and the management can take the right decision as it has proofs like screenshots which are uploaded to the cloud.
* The adoption of the cloud server improves scalability and reliability of the application without worrying about the storage capability.
* The usage of a few algorithms will increase the performance and robustness of the system.
* Cost involved in designing this service is comparatively less than conducting exams manually.
* Not a paper based examination.

**1.2.2 Disadvantages:**

* Even though the model was trained with humongous data it takes a few seconds to detect a mobile phone.

# 

# 

# 2. REQUIREMENT ANALYSIS

**2.1 Hardware and Software requirements**

Hardware requirements are,

* + - Processor: Minimum 1GHz. Recommended 2GHz or more.
    - Ethernet connection (LAN) or a wireless adapter (Wi-Fi).
    - Hard Drive: Minimum 32GB. Recommended 64G
    - 2GB free disk space.
    - Memory (RAM): Minimum 4GB.
    - Operating system Windows 7 or newer. Software requirements are,

**Server:** Local host

**Presentation languages:** HTML, and CSS

**Scripting languages:** Python, Java Script

**Server side language:** Python

**Framework :**Django

**Cloud :**Google cloud

**2.2 Software Requirements Specification**

In this section we will discuss the Software Requirements Specification (SRS) for the project Online Proctor Service. And this SRS document defines how an application will interact with system hardware, other programs. This section is used to describe the non-functional requirements that define the quality of the system, such as reliability, serviceability, security, scalability, availability and maintainability. The SRS document provides a complete overview which assists in navigation and reader comprehension.

**2.2.1 Vision**

|  |  |
| --- | --- |
| **Project Abstract** | |
| **Name of the Project** | Online proctor Service |
| **Vision** | * For the corresponding educational institution Online Proctor Service furnishes the path to identify student attentiveness towards class by monitoring them based upon the student's level of distractedness during the class. * For the corresponding educational institution Online Proctor Servicebrings forth the advantages of conducting online exams without malpractice. |
| **Users / Actors of the System** | Student  Educational Institution |
| **System Features & Functional Capabilities** | 1.Algorithm should be able to detect a student's interest during the class based upon the level of distraction.  2.Algorithm should be able to identify the students who are indulging in committing malpractice during an assessment.  3.Algorithm should be able to detect multiple individuals during an assessment.  4.Algorithm should be able to detect distraction based upon monitoring the frequency of user attempts to look away from the screen.  5.Algorithm should be able to capture when a user has crossed the average threshold of distractions.  6.Algorithm should be able to upload the captured content to a cloud service. |
| **Technologies / Tools to be Used** | Python  Machine Learning  Image Processing  Deep Learning  Pattern Recognition  Google Cloud Platform  Google Drive API  HTML and CSS  PyCharm |
| **Third party libraries / APIs / Services to be used** | Tensor Flow, Python, Keras, Google Drive API |
| **Final Deliverable must Include** | A. Application archive  B. Complete Source code |
| **Documents** | Abstract   1. Vision Document 2. SRS document   Design Document  Test Case |

**2.2.2 Glossary**

|  |  |
| --- | --- |
| Student | Student is the person who attends classes/ gives examination virtually |
| Educational Institution | Educational institution will take care of observing students’ activities closely and make respective decisions. |
| Web interface | An application which can be accessed by the student across the web browser. |

**2.2.3 Scope**

This proposed system is to provide a solution to the management to track the student activities during the course of examination. Each and every students’ movement will be captured if they are indulged in malpractice. This system provides a secure and trustworthy environment to the student as well as the institution. Screenshots will be uploaded to the cloud if the particular student has crossed the minimum threshold. Management will be able to take decisions after reviewing the screenshots whether to continue or to terminate the

student. There exists multiple metrics like malpractice, interestedness to assess the student in a better way. Proposed system will be able to detect distraction based upon monitoring the frequency of user attempts to look away from the screen.

**2.2.4 Exclusions:**

* Students doesn’t have privileges to access the data in the cloud
* Students doesn’t have access to watch the metrics count

**2.2.5 Assumptions:**

* Users are supposed to have recognizable facial features.

**2.2.6 System Functions**

* Student
* Gives examination
* Attends Class
* Educational Institution
* View Interestedness count
* View Malpractice count
* View Screenshots uploaded to cloud
* Takes decision
* Give feedback

**2.2.7 Detailed Software Requirements**

**Table 2.1 Use case- Student**

|  |  |
| --- | --- |
| Actor Name | Student |
| Actor Id | TC01 |
| Description | Attends class  Gives Examination |
| Main Activities | Student attends class virtually  Students gives examination virtually |
| Frequency of Use | High |
| WorkEnvironment | Browser |
| Number of Users | Not limited |

**Table 2.2 Use case- Educational Institution**

|  |  |
| --- | --- |
| Actor Name | Educational Institution |
| Actor Id | TC02 |
| Description | Monitors student activity |
| Main Activities | Educational institution will be able to take decision based on the metrics calculated by the proposed system |
| Frequency of Use | High |
| Work Environment | Browser |
| Number of Users | Not limited |

**2.2.8 Functional Capabilities**

* The system can optimize images captured during the proctoring session.
* The system shows live proctoring updates in the front-end during the proctoring session.
* The system uses multithreading to use the processing resources efficiently.
* The system can be able to detect user behavior using facial features like eyes and lips.
* The system can be able to detect objects such as mobile devices.
* The system can upload the captured data to the cloud when a suspicious activity is detected.

### 2.2.9 Non-Functional Requirements

* As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states. (Usability)
* The system must be interactive and the delays involved must be less. So, in every action-response of the system, there are no immediate delays. (Performance)
* Flexible service-based architecture will be highly desirable for future extension (Supportability)
* The system should be available 24 X 7. (Reliability/ availability)
* Downtime should be less than 1%. (Reliability/availability)

**2.2.10 Exclusions:**

* Students cannot access the cloud directly.
* Students can’t see the values of interestedness count distractions count and malpractice count.

**2.2.11 Assumptions:**

* Database is considered here.
* Database that consists of information about the registered persons, no of projects, reviews given by users.
* Only the admin can access the whole data.
* The system has to authenticate a user, determine whether he is valid or not.

# 3. MODULES

There are different types of modules in this project,

* Machine Learning Models
* Deep Learning Models
* Image processing Modules
* Back-End Modules
* Front-End Modules

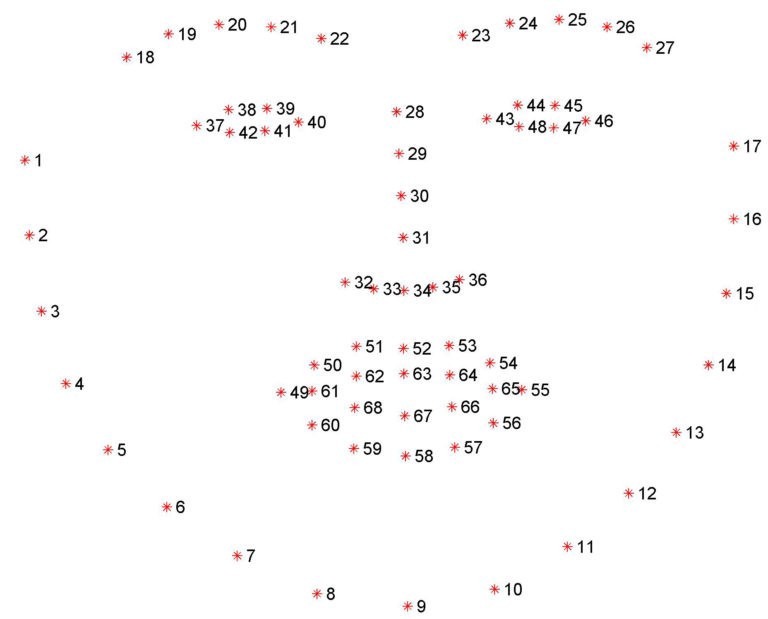
**3.1 Machine Learning Modules**

Machine learning is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

**3.1.2 Dlib:**

According to dlib’s GitHub page, dlib is a toolkit for making real world machine learning and data analysis applications in C++. While the library is originally written in C++, it has good, easy to use Python bindings.

* It ‘s a landmark’s facial detector with pre-trained models, the dlib is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person’s face.
* It detects faces even when they are not perfectly frontal to a good extend. Which is really good for a frontal face detector. But you can only expect so much from it.
* Detecting facial landmarks is a subset of the shape prediction problem. Given an input image, a shape predictor attempts to localize key points of interest along the shape.
* The pre-trained facial landmark detector inside the dlib library is used to estimate the location of 68 (x, y)-coordinates that map to facial structures on the face.

****

We used Dlib for analyzing facial features of the person who is sitting in front of the camera, Dlib algorithm takes an input frame which is an image from webcam in our application, after feeding the image to Dlib algorithm, it analyzes the image and returns a list of 68 co-ordinates in the image, where each co-ordinate denotes a specific point in the face, those co-ordinates are leveraged to calculate and estimate facial features of the person. We used this Dlib module for detecting activities like, Yawning, Blinking, Drowsiness, Looking Away from screen or not. We implemented all these using Dlib module. How they are implemented will be discussed in the implementation section.

**Steps to Install Dlib**:

Step 1: Install Visual Studio 2016

Step 2: Install CMake v3.8.2

Step 3: Install Anaconda 3

Step 4: Download Dlib from http://dlib.net/files/dlib-19.6.zip

Step 5: Build Dlib library

**3.2 Deep Learning Modules**

We used Deep learning to detect objects such as mobile phone or other electronic devices which are not allowed while taking an online class or an exam, why we used deep learning instead of machine learning because the scope of possibilities is many in case of object detection, which means an object to be detected can be anywhere in the frame and not with a defined shape. Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain albeit far from matching its ability allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

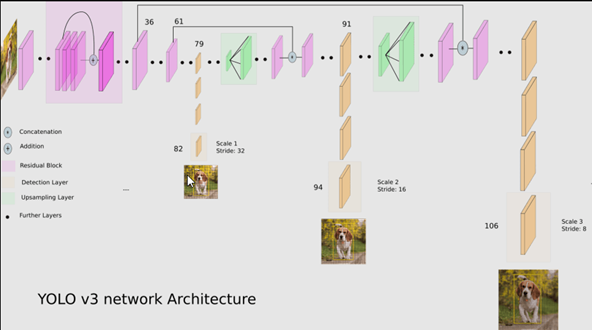
**3.2.1 What is Yolo?**

* "You Only Look Once" or YOLO is a family of deep learning models designed for fast object Detection.
* There are three main variations of YOLO, they are YOLOv1, YOLOv2, and YOLOv3.
* The first version proposed the general architecture, where the second version refined the design and made use of predefined anchor boxes to improve the bounding box proposal, and version three further refined the model architecture and training process.
* It is based on the idea that:

" *A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.* "

**3.2.1.1 Yolo Architecture**

* The inputs are a batch of images of shape (m, 416, 416, 3).
* YOLO v3 passes this image to a convolutional neural network (CNN).
* The last two dimensions of the above output are flattened to get an output volume of (19, 19, 425):
* Here, each cell of a 19 x 19 grid returns 425 numbers.
* 425 = 5 \* 85, where 5 is the number of anchor boxes per grid.
* 85 = 5 + 80, where 5 is (pc, bx, by, bh, bw) and 80 is the number of classes we want to detect.
* The output is a list of bounding boxes along with the recognized classes. Each bounding box is represented by 6 numbers (pc, bx, by, bh, bw, c). If we expand c into an 80-dimensional vector, each bounding box is represented by 85 numbers.
* Finally, we do the IoU (Intersection over Union) and Non-Max Suppression to avoid selecting overlapping boxes.



**3.2.1.2 Convolution layers in YoloV3:**

* It contains 53 convolutional layers which have been, each followed by batch normalization layer and Leaky ReLU activation.
* Convolution layer is used to convolve multiple filters on the images and produces multiple feature maps
* No form of pooling is used and a convolutional layer with stride 2 is used to downsample the feature maps.
* It helps in preventing loss of low-level features often attributed to pooling.

**3.2.1.3Input and Output of Yolov3:**

i. Input Preprocessing:

* The images need to be resized to 416 x 416 pixels before feeding it into our model or the dimensions can also be specified while running the python file

ii. Input Dimensions:

* The model expects inputs to be color images with the square shape of 416 x 416 pixels or it can also be specified by the user.

iii. The output of the model

* The output is a list of bounding boxes along with the recognized classes.
* Each bounding box is represented by 6 numbers (pc, bx, by, bh, bw, c). Here c is an 80-dimensional vector equivalent to the number of classes we want to predict, each bounding box is represented by 85 numbers.

iv. Output of Post-processing

* The Inference program produces a list of numpy arrays, the shape of which is displayed as the output. These arrays predict both the bounding boxes and class labels but are encoded.
* Further we will take each of the numpy arrays and decode the candidate bounding boxes and class predictions. If there are any bounding boxes that don't confidently describe an object (having class probabilities less than the threshold, 0.3) we'll ignore them.
* Here a maximum of 200 bounding boxes can be considered in an image.
* I have used the correct\_yolo\_boxes () function to perform translation of bounding box coordinates so that we can plot the original image and draw the bounding boxes.
* Again to remove those candidate bounding boxes that may be referring to the same object, we define the amount of overlap as non-max suppression threshold = 0.45.
* Also there is a need to rescale the coordinates of the bounding boxes to the original image along with displaying the label and scores on top of each of them.
* All these Post-processing steps need to be performed before we get the bounding boxes along with the recognized classes in our output image.

**3.2.1.4Speed and Accuracy of YoloV3:**

The speed and accuracy of YoloV3 is completely dependent upon the GPU and hardware of the system where it is running,

* Model Speed (in FPS) - On a Pascal Titan X it processes images at 30 FPS.
* Model Accuracy (On Testing dataset) - It has a mAP (Mean Average Precision) of 87.54% on VOC test set.

**3.2.1.5 Training and Testing YoloV3:**

In Order to train a Yolov3 we need a dataset as per the use case with all scenarios covered, here in our application the scenario is to detect a mobile device when a person uses it in front of the camera, so we need a dataset according to the use case. We created a relevant dataset of ~400 images which are close to the use case, and labelling. A label file has to be generated for each image which directs the yolov3 algorithm where to look for the object, for which we are training the model. A weights file of format ‘. weights’ will be generated based upon the training images, this ‘.weights’ file will be used for detecting the intended objects.

More in details information on training a yolov3 is covered in Implementation section.

**3.3 Image Processing**

Image processing is one vital part in this application, we used image processing to optimize images and converting them RGB color code to GRAY color code, without losing significant features of the images which helps detecting the features in the image.

OpenCV module helps doing image processing, such as doing dimensionality reduction from RGB to BW format, and also resizes the images to desired height and width. In case of YoloV3, we need to use an image of size 416 x 416, OpenCv helps converting that image to the specified resolution without loosing it’s features.

**3.4 Front-End**

**3.4.1 Django Template**

Templates are the third and most important part of [Django’s MVT Structure](https://www.geeksforgeeks.org/django-project-mvt-structure/). A template in Django is basically written in HTML, CSS and JavaScript in an .html file. Django framework efficiently handles and generates dynamically HTML web pages that are visible to end-user. Django mainly functions with a backend. Django provides a convenient way to generate dynamic HTML pages by using its template system.A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted. In HTML file, we can't write python code because the code is only interpreted by python interpreter not the browser. We know that HTML is a static markup language, while Python is a dynamic programming language. Django template engine is used to separate the design from the python code and allows us to build dynamic web pages. Django Templates can be configured in app\_name/settings.py

**3.4.2 HTML**

[HTML](https://www.w3.org/html/) is the language for describing the structure of Web pages

* Publish online documents with headings, text, tables, lists, photos, etc.
* Retrieve online information via hypertext links, at the click of a button.
* Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
* Include spread-sheets, video clips, sound clips, and other applications directly in their documents.

**3.4.3 CSS**

[CSS](https://www.w3.org/Style/CSS/) is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments.

* CSS stands for Cascading Style Sheets
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media
* CSS saves a lot of work. It can control the layout of multiple web pages all at once
* External style sheets are stored in CSS files

**3.4.3 JavaScript**

JavaScript is a dynamic programming language that's used for web development, in web applications, for game development, and lots more. It allows you to implement dynamic features on web pages that cannot be done with only HTML and CSS.

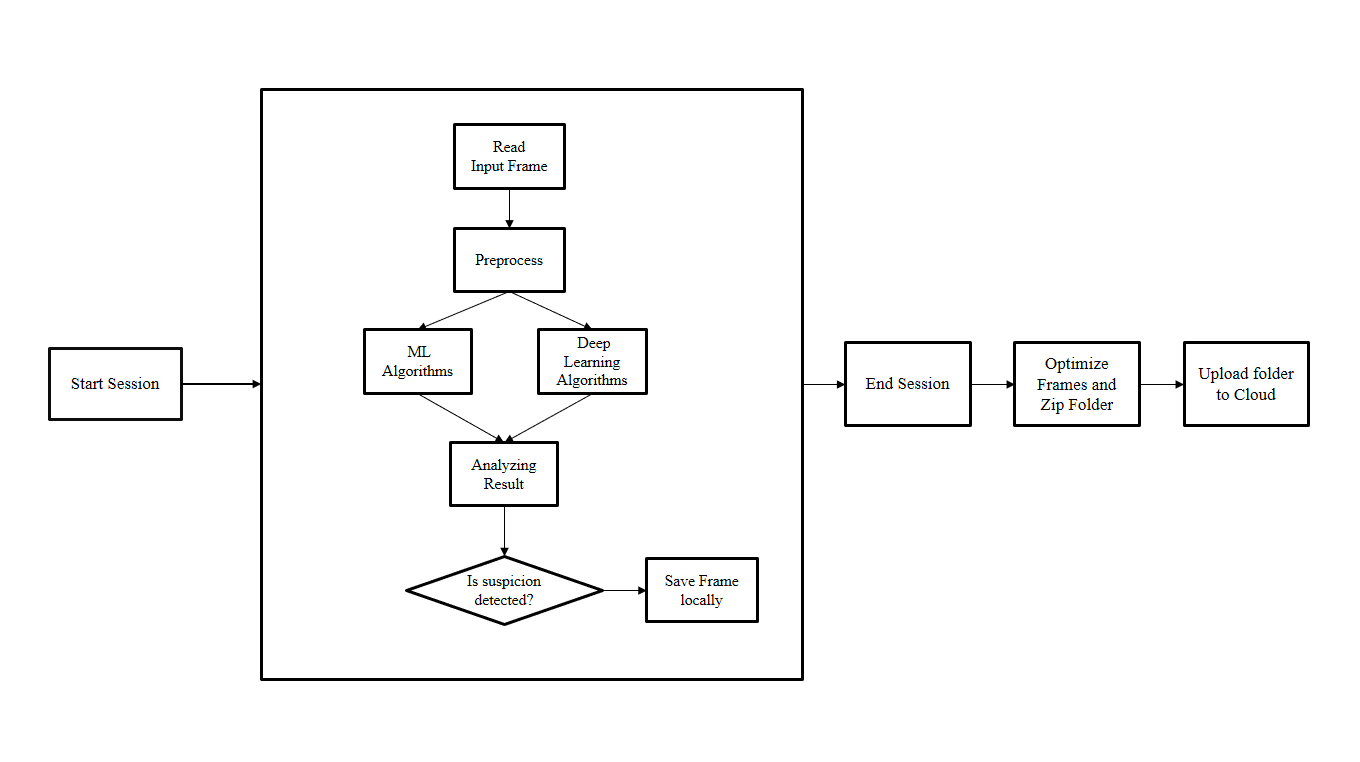
* JavaScript is a lightweight, interpreted programming language.
* Designed for creating network-centric applications.
* Complementary to and integrated with Java.
* Complementary to and integrated with HTML.
* Open and cross-platform.

**3.4.4 Ajax**

AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.

* Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.
* Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.
* With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.
* XML is commonly used as the format for receiving server data, although any format, including plain text, can be used.
* AJAX is a web browser technology independent of web server software.
* A user can continue to use the application while the client program requests information from the server in the background.
* Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger.
* Data-driven as opposed to page-driven

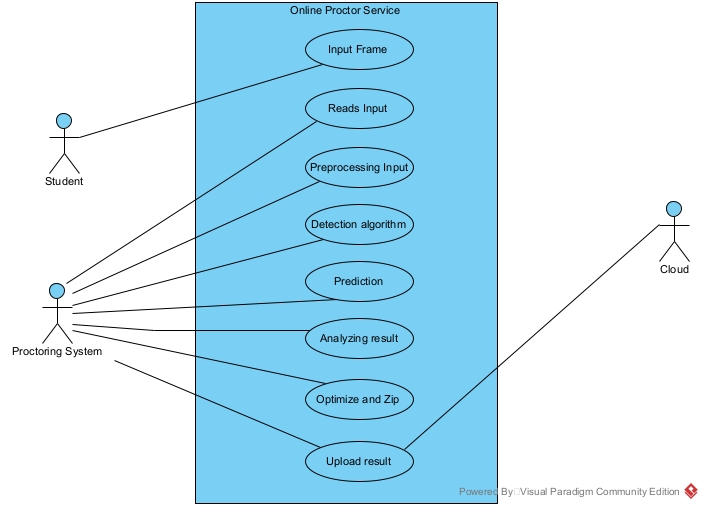
**3. SYSTEMDESIGN**

****

**3.1 UML Diagrams**

**3.1.1 Use case diagram for entire system**

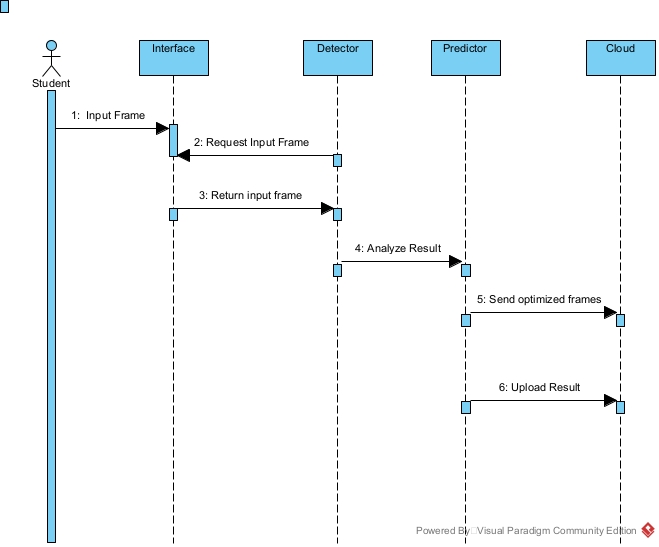
A use case diagram is a dynamic or behavior diagram in [UML.](https://www.smartdraw.com/uml-diagram/) Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. The below use case diagram represents the functionality of entire system. The system contains three actors, an actor in the Unified Modelling Language (UML) "specifies a role played by a user or any other system that interacts with the subject." "An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject.Student is someone who gives the examination, proctoring system takes care of monitoring student activities such as distractions, interestedness, malpractice. Cloud is where captured content of the student who is indulged in illicit activities are uploaded.



**Fig 3.1.1 Use Case Diagram**

**3.1.2 Sequence Diagram**

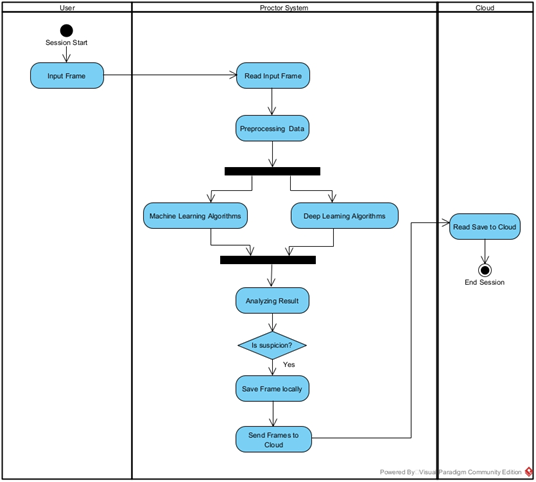
As the name suggests, sequence diagrams describe the sequence of messages and interactions that happen between actors and objects. Actors or objects can be active only when needed or when another object wants to communicate with them. All communication is represented in a chronological manner.A sequence diagram simply depicts interaction between objects in a sequential orderi.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.



### Fig 3.1.2 Sequence Diagram

**3.1.3 Activity Diagram**

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.



### Fig 3.1.3 Activity Diagram

**4. SYSTEM IMPLEMENTATION**

**4.1 Selected Software**

Considered software is PyCharm as development tool kit, Django as a primary framework, Python as a server-side scripting language, JavaScript for integrating AJAX, Google Drive as a cloud storage service and HTML, CSS for user-interface design.

**4.1.2 Technologies:**

**HTML:**

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. It is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet audience under the direction of the [W3C](http://www.w3.org/), the organization charged with designing and maintaining the language.The definition of HTML is Hypertext Markup Language.Hypertext is the method by which you move around on the web — by clicking on special text called hyperlinks which bring you to the next page. The fact that it is hyper just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking on links — there is no set order to do things in.

Mark-up is what HTML tags do to the text inside them. They mark it as a certain type of text (*italicized* text, for example).

HTML is a Language, as it has code-words and syntax like any other language.

**CSS:**

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media**.** CSS saves a lot of work. It can control the layout of multiple web pages all at once. External style sheets are stored in CSS files.

**Styles of CSS**

**Color:** Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values.

**Padding:**The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

**Border:** The CSS border properties allow you to specify the style, width, and color of an element's border.

**Margin:** The CSS margin properties are used to create space around elements, outside of any defined borders.

**Animations:** CSS allows animation of HTML elements without using JavaScript or Flash.

**JavaScript:**

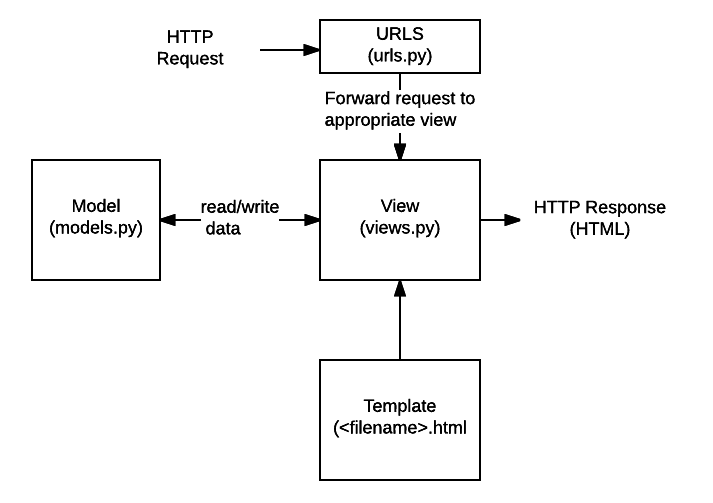
JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

**Validations in JavaScript:**

JavaScript provides a way to validate form's data on the client's computer before sending it to the web server. It would require just a loop through each field in the form and check for data. Data Format Validation: Secondly, the data that is entered must be checked for correct form and value.

**Django:**

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.



**URLs**: While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.

**View**: A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via models, and delegate the formatting of the response to templates.

**Models**: Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.

**Templates**: A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A view can dynamically create an HTML page using an HTML template, populating it with data from a model. A template can be used to define the structure of any type of file; it doesn't have to be HTML.

**Google Cloud Platform:**

Google Cloud Platform (GCP), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. Registration requires a credit card or bank account details. Google Cloud Platform provides infrastructure as a service, platform as a service, and server-less computing environments.

In April 2008, Google announced App Engine, a platform for developing and hosting web applications in Google-managed data centers, which was the first cloud computing service from the company. The service became generally available in November 2011. Since the announcement of App Engine, Google added multiple cloud services to the platform. Google Cloud Platform is a part of Google Cloud, which includes the Google Cloud Platform public cloud infrastructure, as well as Google Workspace (G Suite), enterprise versions of Android and Chrome OS, and application programming interfaces (APIs) for machine learning and enterprise mapping services.

**Service Account in Google Cloud:**

A service account is a special kind of account used by an application or a virtual machine (VM) instance, not a person. Applications use service accounts to make authorized API calls, authorized as either the service account itself, or as Google Workspace or Cloud Identity users through domain-wide delegation.

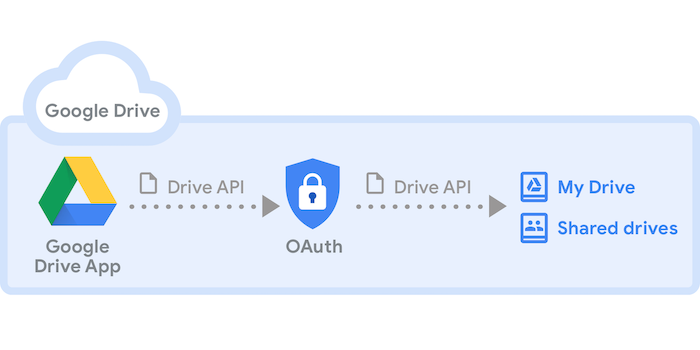
For example, a Compute Engine VM can run as a service account, and that account can be given permissions to access the resources it needs. This way the service account is the identity of the service, and the service account's permissions control which resources the service can access. A service account is identified by its email address, which is unique to the account.

**Differences between a service account and a user account:**

* Service accounts do not have passwords, and cannot log in via browsers or cookies.
* Service accounts are associated with private/public RSA key-pairs that are used for authentication to Google.
* You can let other users or service accounts impersonate a service account.
* Service accounts are not members of your Google Workspace domain, unlike user accounts. If you share Google Workspace assets, like docs or events, with all members in your Google Workspace domain, they are not shared with service accounts. Similarly, Google Workspace assets created by a service account are not created in your Google Workspace domain. As a result, your Google Workspace and Cloud Identity admins can't own or manage these assets.

**Google Drive API:**

The Google Drive API allows you to create apps that leverage Google Drive cloud storage. You can develop applications that integrate with Google Drive, and create robust functionality in your application using Google Drive API. This diagram shows the relationship between your Google Drive app, Google Drive, and Google Drive API:



**Python 3.6:**

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customization applications. This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

**Installation of Python 3.6:**

Step 1: Download the Python Installer binaries

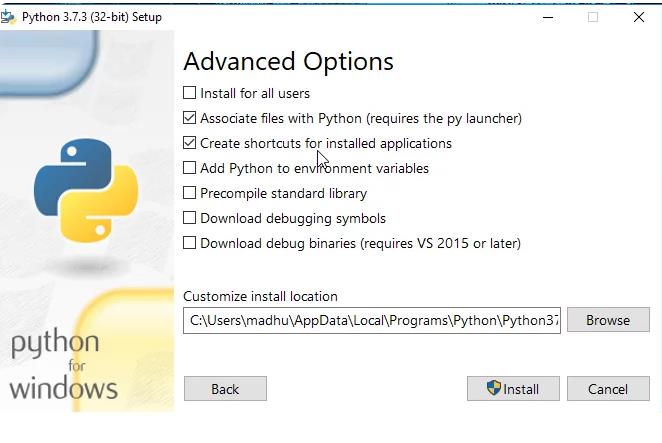
* 1. Open the [official Python website](https://www.python.org/downloads/windows/) in your web browser. Navigate to the Downloads tab for Windows.
  2. Choose the latest Python 3 release. In our example, we choose the latest Python 3.7.3 version.
  3. Click on the link to download Windows x86 executable installer if you are using a 32-bit installer. In case your Windows installation is a 64-bit system, then download Windows x86-64 executable installer.

## Step 2: Run the Executable Installer

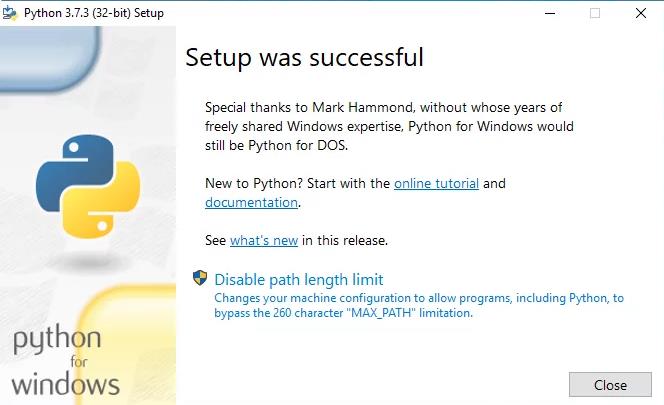
1. Once the installer is downloaded, run the Python installer.
2. Check the **Install launcher for all users** check box. Further, you may check the **Add Python 3.7 to path** check box to include the interpreter in the execution path.



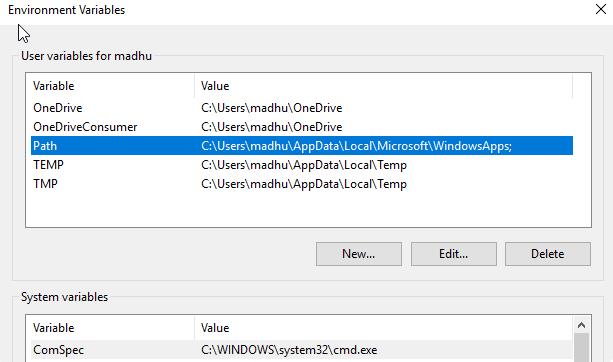
1. Select Customize installation.
2. Documentation
3. Install the global launcher for `.py` files. This makes it easier to start Python
4. Install for all users.
5. This takes you to Advanced Options available while installing Python. Here, select the Install for all users and Add Python to environmentvariables check boxes. Optionally, you can select the Associate files with Python, create shortcuts for installed applications and other advanced options. Make note of the python installation directory displayed in this step. You would need it for the next step. After selecting the Advanced options, click Install to start installation.



1. Once the installation is over, you will see a **Python Setup Successful** window.



1. Add environment variables



1. We have successfully installed python.

**Modules used:**

1. Dlib:
   1. shape\_predictor
   2. rectangle
2. cv2:
   1. VideoCapture
      1. Open video file or image file sequence or a capturing device or a IP video stream for video capturing
      2. Syntax: VideoCapture(index)
         1. Index: camera\_id + domain\_offset (CAP\_\*) id of the video capturing device to open, To open default camera using default backend just pass 0.
   2. CascadeClassifier
      1. function of OpenCV to point to the location where we have stored the XML file i.e., ‘haarcascade\_frontalface\_default.xml’
   3. cvtColor
      1. Is used to convert an image from one color space to another.
   4. COLOR\_BGR2GRAY
      1. OpenCV reads the RGB image, it usually stores the image in BGR (Blue, Green, Red) channel. For the purposes of image recognition, we need to convert this BGR channel to grey channel. The reason for this is grey channel is easy to process and is computationally less intensive as it contains only 1-channel of black-white
   5. ConvexHull
      1. The Convex Hull of a shape or a group of points is a tight-fitting convex boundary around the points or the shape or simply the shape of an object
   6. DrawContours
      1. To draw the contours, cv. DrawContours function is used. It can also be used to draw any shape provided you have its boundary points.
      2. Its first argument is source image, second argument is the contours which should be passed as a Python list, third argument is index of contours
   7. PutText
      1. cv2.putText() method is used to draw a text string on any image.
      2. It is the coordinates of the bottom-left corner of the text string in the image. The coordinates are represented as tuples of two values i.e. (X coordinate value, Y coordinate value).
   8. FONT\_HERSHEY\_PLAIN
      1. It gives small size sans-serif font
   9. Imencode
      1. imencode() function is to convert (encode) the image format into streaming data and assign it to memory cache. It is mainly used for compressing image data format to facilitate network transmission
   10. Waitkey
       1. waitKey() is a keyboard binding function. Its argument is the time in milliseconds. The function waits for specified milliseconds for any keyboard event. If you press any key in that time, the program continues. If 0 is passed, it waits indefinitely for a key stroke
3. Imutils
   1. resize
   2. face\_utils
      1. FACIAL\_LANDMARKS\_IDXS
      2. shape\_to\_np
4. numpy
   1. concatenate
   2. mean
5. scipy.spatial.distance.euclidean
6. Haarcase\_frontalface\_default.xml
7. Shape\_predictor\_68\_face\_landmarks.dat
8. django-googledrive-storage:
   1. This module helps storing files using google drive as a backend, the data is accessible only through API calls.

**4.2 Training Model:**

In order to detect objects such as mobile devices, we have to train the deep learning which is yolov3 in our case, with a customized dataset of multiple images of mobile devices comprising all possible scenarios of using a mobile device in front of a camera while an online class or an examination is taking place.

Steps for creating a customized model for object detection:

1. Collect relevant dataset
2. Clean and preprocessing the dataset
3. Labelling dataset and training model
4. Test the model with unlabeled dataset

**4.2.1**. **Collect relevant dataset**

Collecting data allows you to capture a record of past events so that we can use data analysis to find recurring patterns. From those patterns, you build predictive models using machine learning algorithms that look for trends and predict future changes. Predictive models are only as good as the data from which they are built, so good data collection practices are crucial to developing high-performing models. The data need to be error-free (garbage in, garbage out) and contain relevant information for the task at hand. For example, a loan default model would not benefit from tiger population sizes but could benefit from gas prices over time.

For this project we have created our own dataset by capturing random pictures which covers all the scenarios of using a mobile device during an online class/examination, we prepared this dataset as relevant as possible according to the use cases in the project, which yields in better performance.

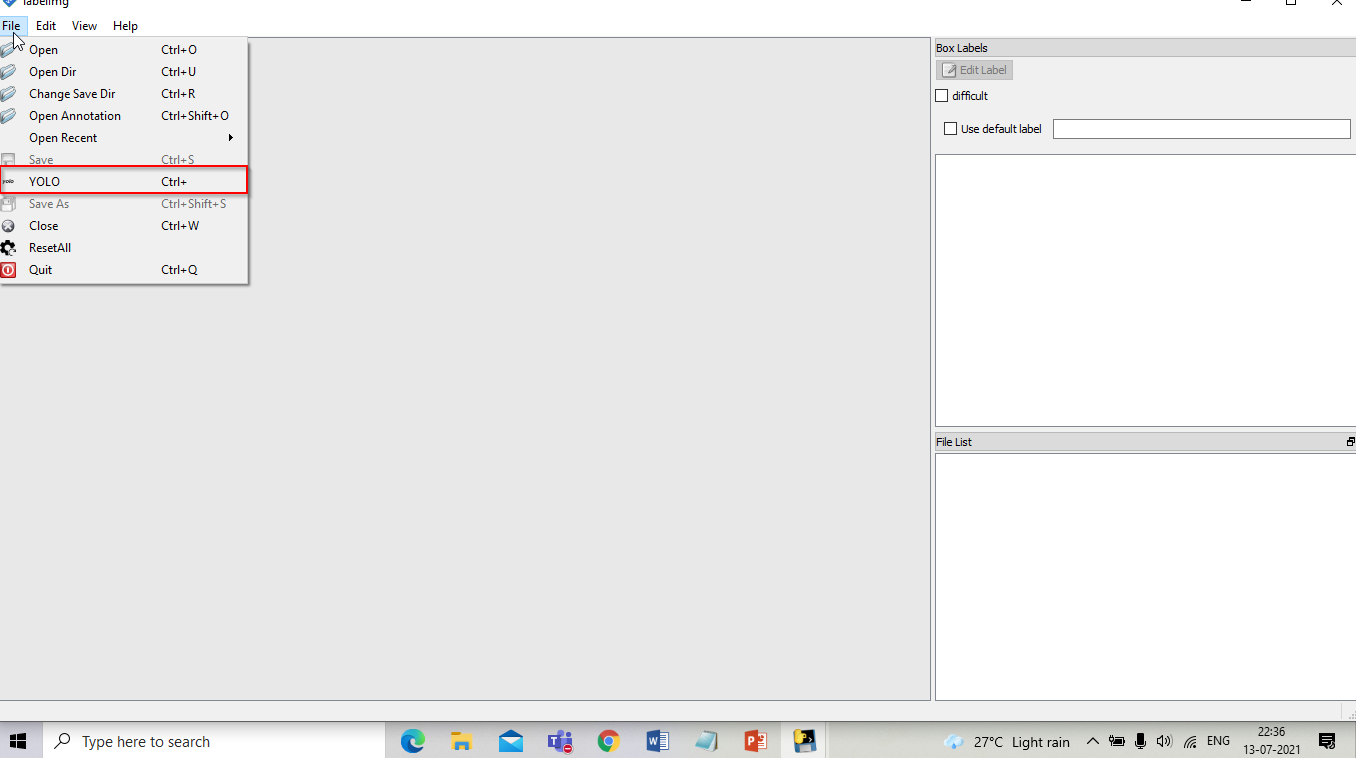
**4.2.2 Clean and preprocessing the dataset**

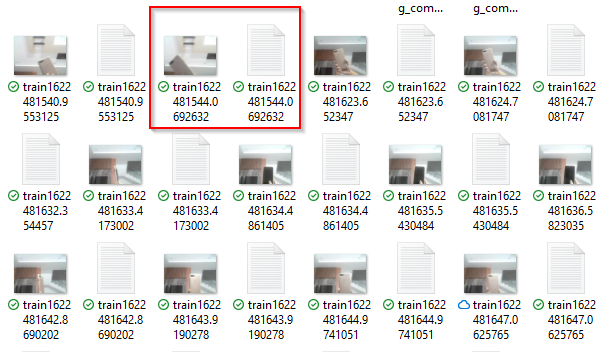
A simple script to sort out images based on the ratio of white and black colors helps cleaning the data. For text generation, we need to make sure that there are no random symbols which are not part of character dictionary, also, spellings should be correct and. if it is not checked, will make it difficult for the model to learn more robustly. To detect the discrepancy in data, use metadata, different checks or maybe uniqueness rule etc. Make sure we do data scrubbing and auditing before feeding to the model. Make sure to validate dataset against a standard data. Make sure we are also working on duplication elimination and data transformation.

The raw dataset is not preferable for using as training dataset because there will be so much noise in the image which is captured and the size of each images captured is very high, which is not training friendly. So in order to keep the image size optimized without losing any important features in the image,

**4.2.3 Labelling dataset and training model**

Labelling each image in the dataset is a very important part of the training, labelling an image involves selecting the target object which needs to be detected and giving a label name for it, this step has to be repeated for all the images in the dataset mostly 80% of it, rest 20% will be used for the testing purpose. For this project we trained a model with a dataset of size approximately 400 images which could detect mobile devices, we used ‘labelimg’ as a tool for labelling the images,





The tool creates a label file for each image, that file contains coordinates of the target object in the image. While training the yolov3 algorithm we use this labelled dataset.

**4.2.4 Test the model with unlabeled dataset**

In order to the trained model with labelled dataset, we have to use unlabeled data. Unlabeled sample of data should contain all the scenarios for which the model is trained for, any novel scenario in test data may affect the accuracy of the model because the model is not trained for that test scenario. Test data should be relevant to training data.Test results of trained model is having accuracy of 93%. Hence, the model can detect mobile devices which are placed within the camera range.

**4.3 Sample Code Snippets:**

‘Is\_yawning’ function is responsible for monitoring the user upper lips and lower lip, which constantly calculates the distance between them, and if the distance crossed the threshold then it will report to the driver function that a yawn has been detected.

def is\_yawning(shape, threshold):

"""functions returns true if yawn detected else false"""

top\_lip = shape[50:53]

top\_lip = np.concatenate((top\_lip, shape[61:64]))

low\_lip = shape[56:59]

low\_lip = np.concatenate((low\_lip, shape[65:68]))

top\_mean = np.mean(top\_lip, axis=0)

low\_mean = np.mean(low\_lip, axis=0)

distance = abs(top\_mean[1] - low\_mean[1])

return distance > threshold

‘Is\_drowsy’ function is completely dependent on the function ‘Is\_blinking’. It makes use of the global variables updated by blinks tracking function, if user eyes are in closed for more than a while then drowsiness will be detected.

def is\_drowsy(IsBlinking):

"""functions returns true if yawn detected else false"""

global COUNT

if IsBlinking:

COUNT += 1

if COUNT > 2:

return True

else:

COUNT = 0

return False

defis\_blinking(frame, shape):

"""functions returns true if yawn detected else false"""

EARThresh = 0.3

(lstart, lend) = face\_utils.FACIAL\_LANDMARKS\_IDXS["left\_eye"]

(rstart, rend) = face\_utils.FACIAL\_LANDMARKS\_IDXS["right\_eye"]

leftEye = shape[lstart:lend]

rightEye = shape[rstart: rend]

leftEAR = eyeAspectRatio(leftEye)

rightEAR = eyeAspectRatio(rightEye)

EAR = (leftEAR + rightEAR) / 2.0

leftEyeHull = cv2.convexHull(leftEye)

rightEyeHull = cv2.convexHull(rightEye)

cv2.drawContours(frame, [leftEyeHull], -1, (0, 0, 255), 1)

cv2.drawContours(frame, [rightEyeHull], -1, (0, 0, 255), 1)

return EAR <EARThresh

‘IsMobileDetected’ function makes use of 50% of processing power, it uses deep learning models to detect any mobile devices, which is trained with tailored dataset of size ~400 training images, which is specifically created for virtual class room scenarios.

defIsMobileDetected(frame, net, output\_layers):

blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), swapRB=True, crop=False)

net.setInput(blob)

results = net.forward(output\_layers)

for result in results:

for detection in result:

scores = detection[5:]

class\_id = np.argmax(scores)

confidence = scores[class\_id]

if confidence > 0.5:

return {"IsMobilePresent": True}

return {"IsMobilePresent": False}

Function “facialResult” is the driver for tracking blinking, drowsiness, yawning functionalities, because all of the functions are interdependent which means the output of one functionality will be fed as an input for some other function, so these functionalities cannot be calculated individually.

def facialResult(frame, shape, YAWN\_THRESH=20):

IsBlinking = is\_blinking(frame, shape)

IsDrowsy = is\_drowsy(IsBlinking)

IsYawning = is\_yawning(shape, YAWN\_THRESH) and IsBlinking

return { "IsBlinking": IsBlinking, "IsDrowsy": IsDrowsy, "IsYawning": IsYawning}

Python threading allows you to have different parts of your program run concurrently and can simplify your design. A thread is a separate flow of execution. This means that your program will have two things happening at once. But for most Python 3 implementations the different threads do not actually execute at the same time: they merely appear to.It’s tempting to think of threading as having two (or more) different processors running on your program, each one doing an independent task at the same time. That’s almost right. The threads may be running on different processors, but they will only be running one at a time.Getting multiple tasks running simultaneously requires a non-standard implementation of Python, writing some of your code in a different language, or using multiprocessing which comes with some extra overhead.

In this application, for detecting behaviour from user face and user background objects is completely independent activities, so we can leverage the concept of threading here, both the functionalities need common input which is a frame from user webcam, so we calculate these functions separate and simultaneously, which decreases the average execution time of the application.

""" threading """

activeThreads =

[threadPool.submit(facialResult, frame, shape),

threadPool.submit(IsMobileDetected, frame, net, output\_layers)]

for thread in concurrent.futures.as\_completed(activeThreads):

results.update(thread.result())

""" threading """

The purpose of optimizing the images before sending them to a cloud, because when we capture an image from webcam it is not storage friendly, the size will be huge, if we do not optimize those images to minimum size with best quality. The storage consumption will be huge, so in order to control storage consumption, proctor system will optimize them locally before uploading to cloud service.

At the end of session, the captured content will be zipped by proctor system and then moves too cloud service.

defoptimizeFolder(path):

"""Optimizes the images captured during the session"""

files = os.listdir(path)

if not files:

return False

else:

command = 'optimize-images ' + '"' + path + '"' + '\.'

os.system(command)

return True

defzipFolder(path):

"""zips the optimized folder to a file"""

sourceLocation = path

dropLocation = path[:path.rfind("\\")] + "\Images.zip"

zipf = zipfile.ZipFile(dropLocation, 'w', zipfile.ZIP\_DEFLATED)

zipdir(sourceLocation, zipf)

zipf.close()

Google Drive Storage is helpful for storing the data files which are consumed within the project, it gives local storage flexibility which is significant when there is no prior information on how much data system is going to consume.

from gdstorage.storage import GoogleDriveStorage

from django.db import models

# Define Google Drive Storage

gd\_storage = GoogleDriveStorage()

class Cloud(models.Model):

upload = models.FileField(upload\_to='./', storage=gd\_storage

**5. TESTING**

**5.1 Introduction to Testing**

Testing is a process of verifying and validating if the developed computer software is correct, complete and has the quality which is acceptable. That means, it is checking if a software system meets specifications and that it fulfills its intended purpose. Hence, it’s a universally accepted (and most debated topic) that testing computer software can never be completely established.

VERIFICATION means are we building the product right. Does our software actually achieve its goals without any bugs or gaps?

VALIDATION means are we building the right product. Is our product actually is what we should have built? Does it actually meet the high-level requirements?

**5.1.1 Testing Levels**

**Unit Testing**

Unit testing is done to check whether the individual modules of the source code are working properly. i.e., testing each and every unit of the application separately by the developer in the developer’s environment. It is also known as Module Testing or Component Testing.

**Integration Testing**

Integration Testing is the process of testing the connectivity or data transfer between couples of units tested modules. It is also known as Integration Testing or String Testing. It is subdivided into top-down approach, bottom-up approach and sandwich approach (combination of top down and bottom up).

**System Testing (end to end testing**)

It’s a black box testing. Testing the fully integrated application this is also called as end to end scenario testing. To ensure that the software works in all intended target systems. Verify thorough testing of every input in the application to check for desired outputs. Testing of the users’ experiences with the application.

**Acceptance Testing:**

User acceptance testing is used to determine whether the product is working for the user correctly. Specific requirements which are quite often used by the customers are primarily picked for the testing purpose. This is also termed as end-user testing.

**5.2 Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TID** | **Scenario** | **Test Case** | **Expected**  **Outcome** | **Actual**  **Outcome** |
| **T01** | When the distance between upper eyelid and lower eyelid is less than threshold then it should be classified as closed. | Classify eyelids position to closed, when user actually closes eyes, and increment closed count to one | Eye lids state should be closed.  Count should be incremented | Eye lids state is closed.  Count is increment |
| **T02** | When the distance between upper eyelid and lower eyelid is greater than threshold then eyes should be classified as open. | Classify eyelids position to open, when user actually opens eyes. | Eye lids state should be open. | Eye lids state is open. |

**5.2.1 Testing function ‘IsBlinking’**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TID** | **Scenario** | **Test Case** | **Expected**  **Outcome** | **Actual**  **Outcome** |
| **T03** | When closed count greater than threshold then drowsiness has to be detected | Classify drowsiness when user is actually drowsy. | Drowsiness should be detected | Drowsiness is detected |
| **T04** | When closed count less than threshold then drowsiness has not to be detected | Classify drowsiness when user is not actually drowsy. | Drowsiness should not be detected | Drowsiness is not detected |

**5.2.2 Testing Function ‘IsDrowsy’**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TID** | **Scenario** | **Test Case** | **Expected**  **Outcome** | **Actual**  **Outcome** |
| **T05** | When the distance between upper lip and lower lip is greater than threshold and eyes state is not in shrink then it should be classified as not yawning. | Detect if user is not yawning when user is not actually yawning | Yawn should not be detected | Yawn is not detected |
| **T06** | When the distance between upper lip and lower lip is greater than threshold and eyes state is in shrink then it should be classified as not yawning. | Classify drowsiness when user is actually drowsy. | Drowsiness should be detected | Drowsiness is detected |
| **T07** | When the distance between upper lip and lower lip is less than threshold and eyes state is in shrink then it should be classified as not yawning. | Detect if user is not yawning when user is not actually yawning | Yawn should not be detected | Yawn is not detected |
| **T08** | When the distance between upper lip and lower lip is less than threshold and eyes state is not in shrink then it should be classified as not yawning. | Detect if user is not yawning when user is not actually yawning | Yawn should not be detected | Yawn is not detected |

**5.2.3 Testing function ‘IsYawning’**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TID** | **Scenario** | **Test Case** | **Expected**  **Outcome** | **Actual**  **Outcome** |
| **T09** | When a mobile device is within the range of camera, then mobile has to be detected. | Classify user is using mobile, when user is actually using mobile. | User is not using mobile | User is not using mobile |
| **T10** | When a mobile device is not within the range of camera, then mobile has to be detected. | Classify user is not using mobile, when user is not actually using a mobile. | User is using mobile | User is using mobile |

**5.2.4 Testing Function ‘IsMobileDetected’**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TID** | **Scenario** | **Test Case** | **Expected**  **Outcome** | **Actual**  **Outcome** |
| **T11** | When user is looking at the screen. | Classify user as looking away from screen, when user actually not looking at the screen | User is looking away from screen | User is looking away from screen |
| **T12** | When user is not looking at the screen. | Classify user as not looking away from screen, when user not looking away from the screen | User is not looking away from screen | User is not looking away from screen |

**5.2.5 Testing Function ‘gazeTracking’**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TID** | **Scenario** | **Test Case** | **Expected**  **Outcome** | **Actual**  **Outcome** |
| **T13** | When more than one face has come within the range of camera | Classify multiple faces are detected when there are actually multiple faces | Multiple individuals are detected | Multiple individuals are detected |
| **T14** | When only one face is within the range of camera | Classify multiple faces are not detected when there are actually no multiple faces | Multiple individuals are not detected | Multiple individuals are not detected |

**5.2.6 Testing Function ‘multipleIndividual’**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TID** | **Scenario** | **Test Case** | **Expected**  **Outcome** | **Actual**  **Outcome** |
| **T15** | When a path exists to the folder | Verify the captured content is saved into path location | Captured content should be saved inside dedicated folder path | Captured content is present in the folder path |
| **T16** | When content present in side the folder, content needs to be optimized | Verify the captured content is optimized or not | Captured content must be optimized, the size of folder should be reduced | Captured content is optimized and folder size is reduced. |

**5.2.7 Testing Function ‘optimizeFolder’**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TID** | **Scenario** | **Test Case** | **Expected**  **Outcome** | **Actual**  **Outcome** |
| **T17** | When a path exists to the folder | Verify the folder is in optimized state or not | Folder should be in optimized state | Folder is in optimized state |
| **T18** | When optimized folder is exists it should be zipped and renamed | Verify the folder is zipped and renamed | Folder should be zipped and renamed | Folder is zipped and renamed |

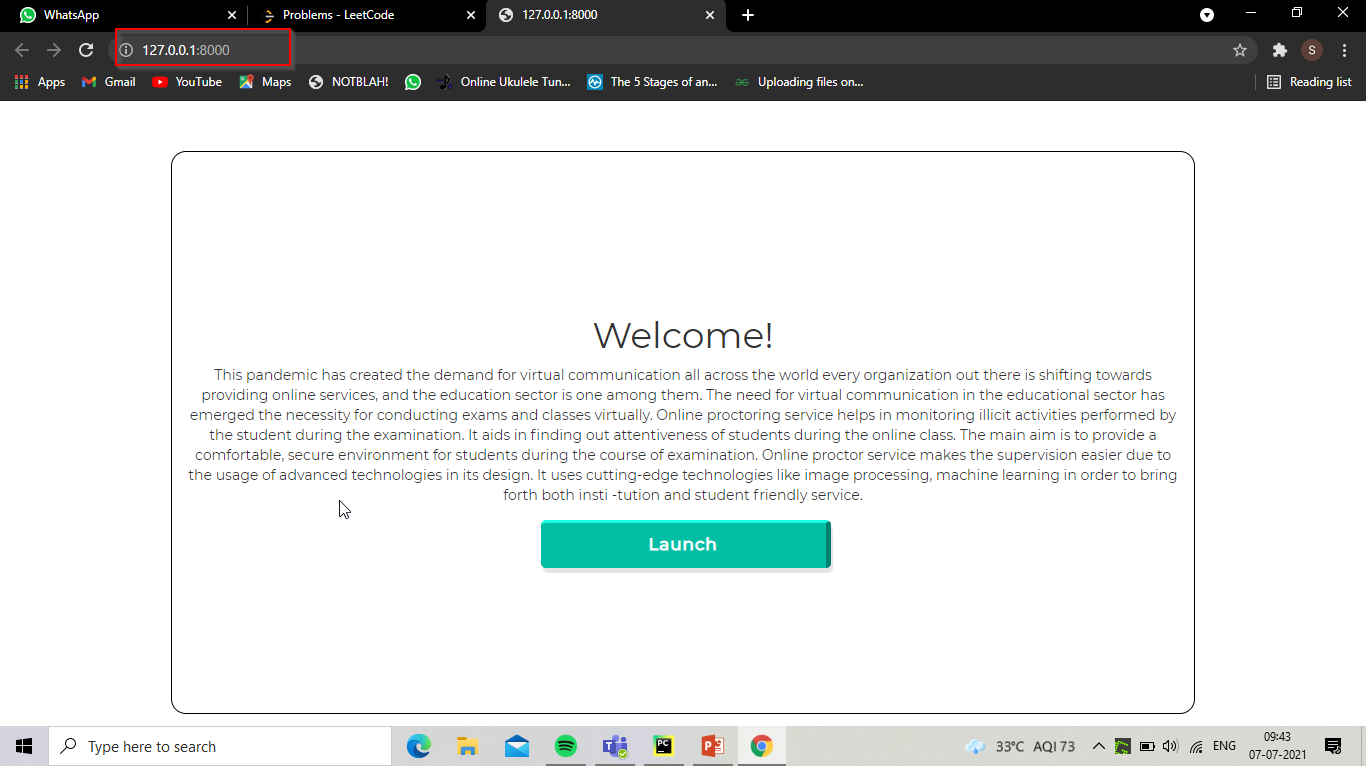
**5.2.8 Testing Function ‘ZipFolder’**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TID** | **Scenario** | **Test Case** | **Expected**  **Outcome** | **Actual**  **Outcome** |
| **T19** | When a folder exists in optimized and zipped state | Verify the folder is optimized, zipped and renamed | Folder should be in optimized, zipped and renamed state | Folder is optimized, zipped and renamed |
| **T20** | Folder has to be uploaded to drop location in cloud service | Verify the folder is uploaded to cloud service or not | Folder should be present in drop location in cloud service | Folder is present in drop location in cloud service |

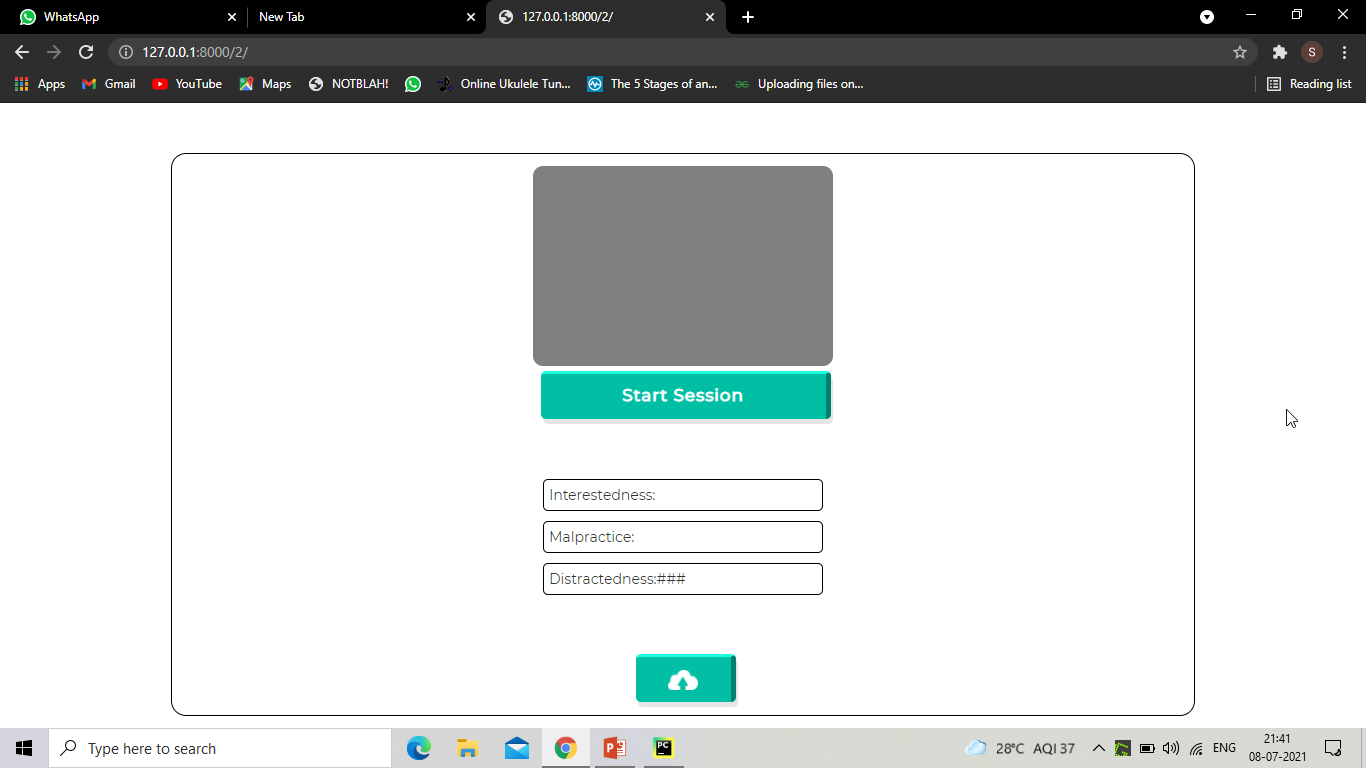
**5.2.9 Testing function ‘UploadToCloud’**

**6. SCREENSHOTS AND REPORTS**

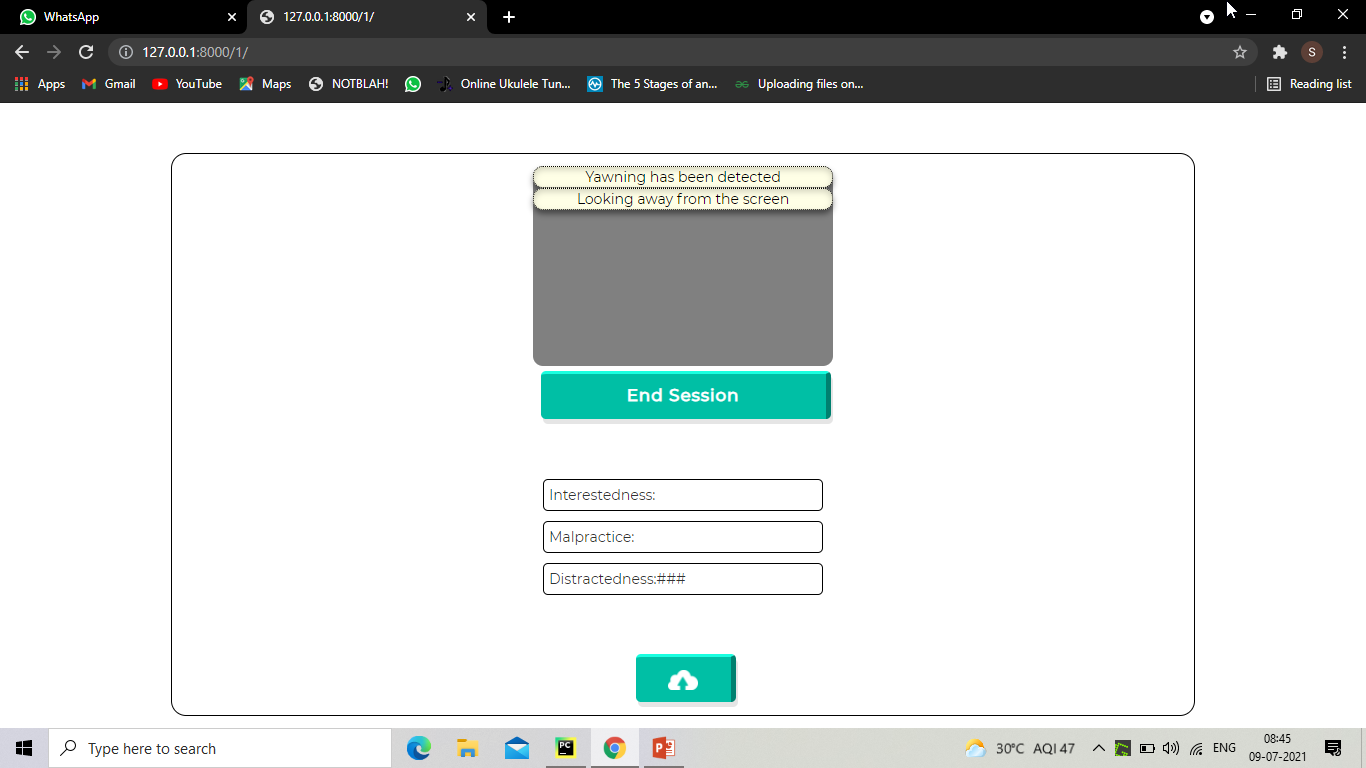
**6.1 Screenshots:**

****

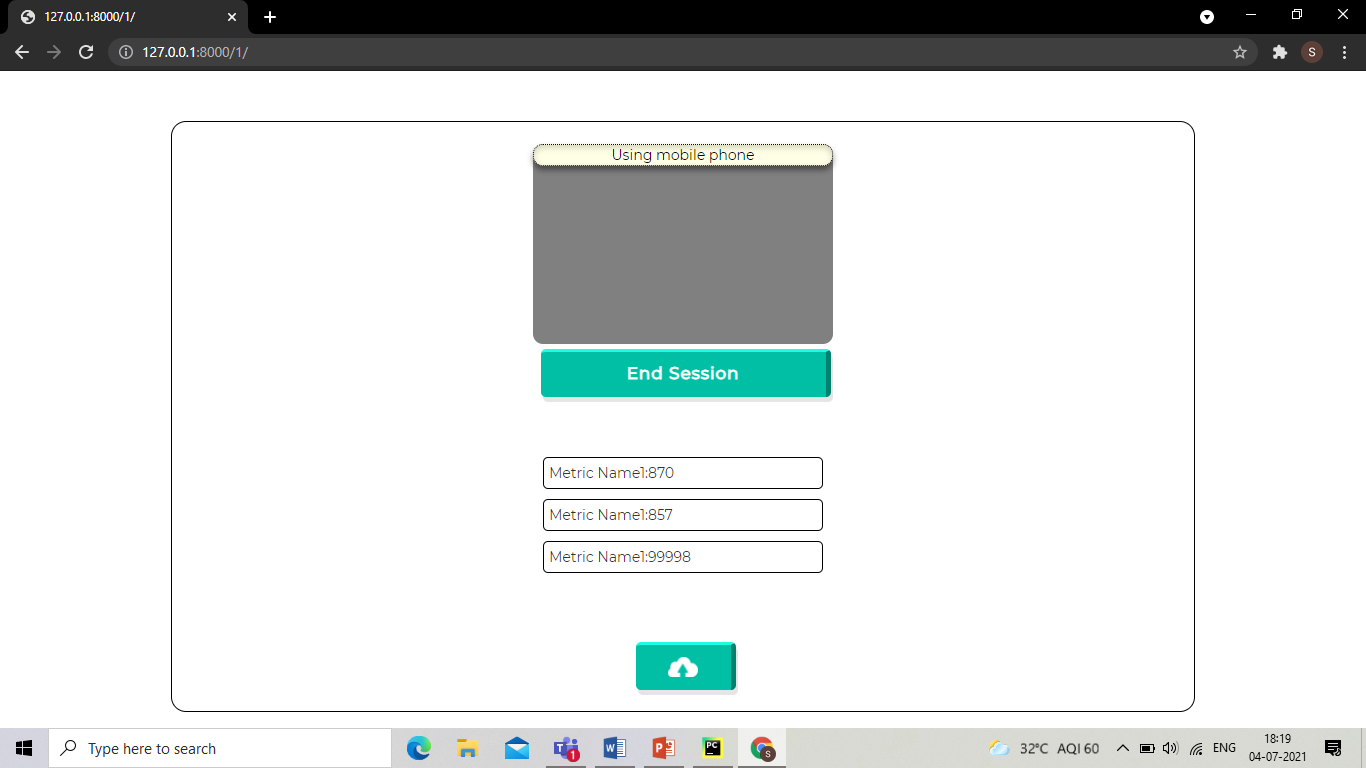
* On landing page, we could see a short description on what online proctor service is about.
* There is a launch button which takes to the main of the project.

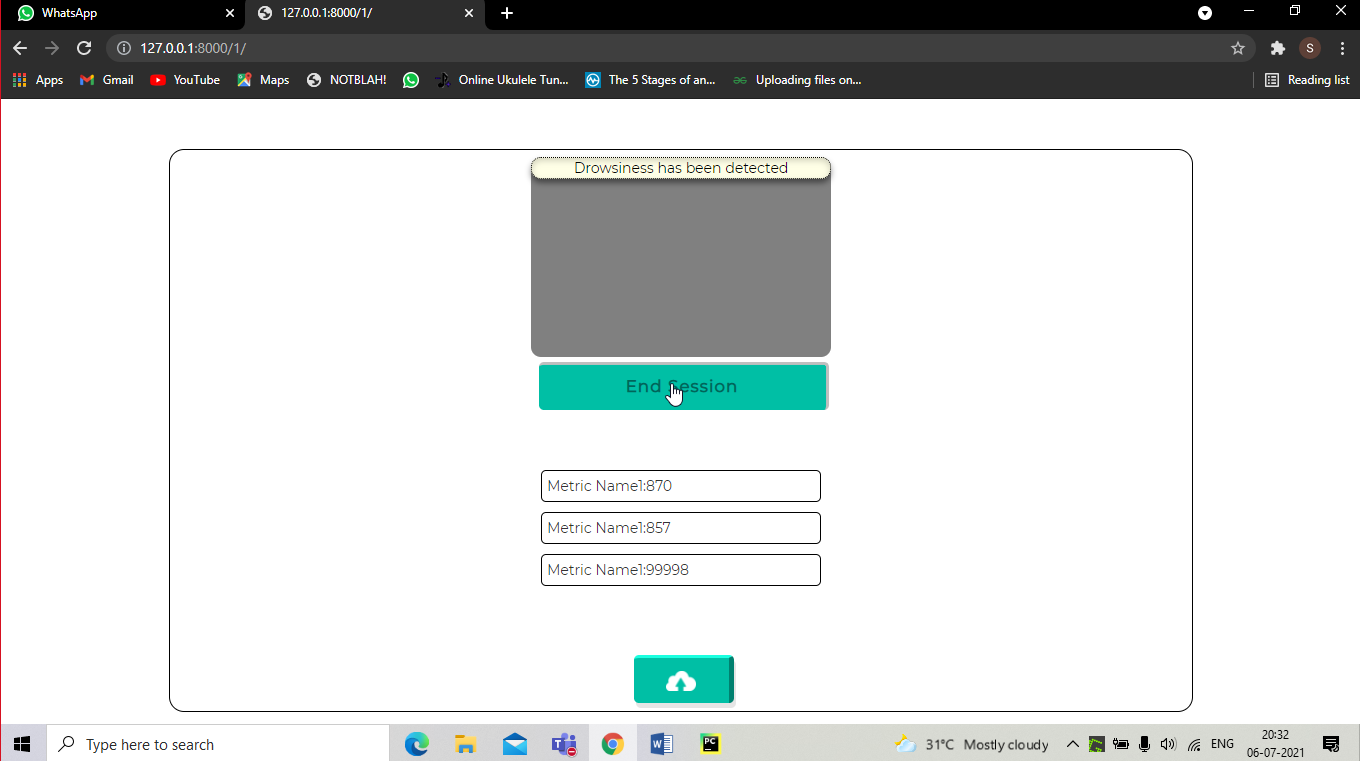


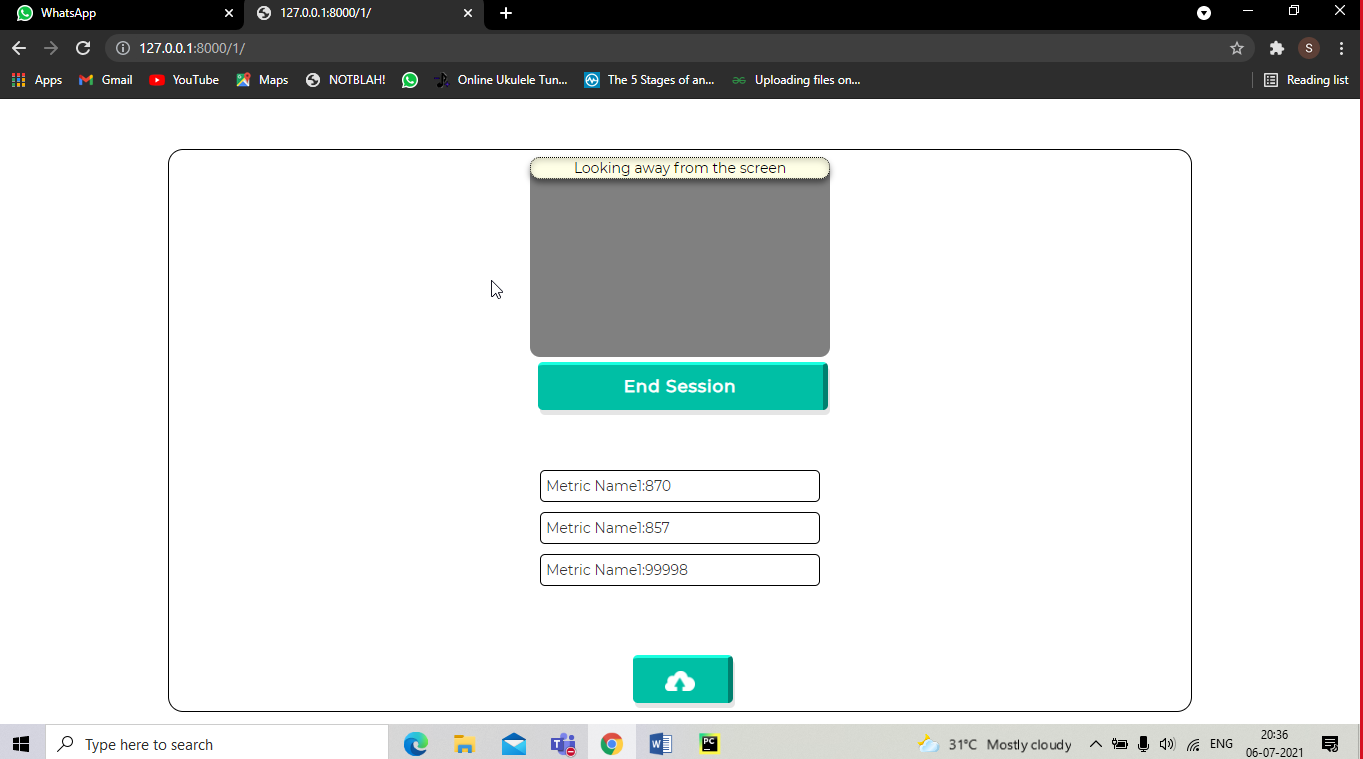
* On landing page, the gray window is where we can see live updates if user performs any activities. Which the proctor system is programmed to detect.
* Monitoring session starts after clicking on Start button.
* Metric Names are the different metrics we will calculate using information tracked throughout the session.
* After clicking on upload button the captured contents will be uploaded to cloud.

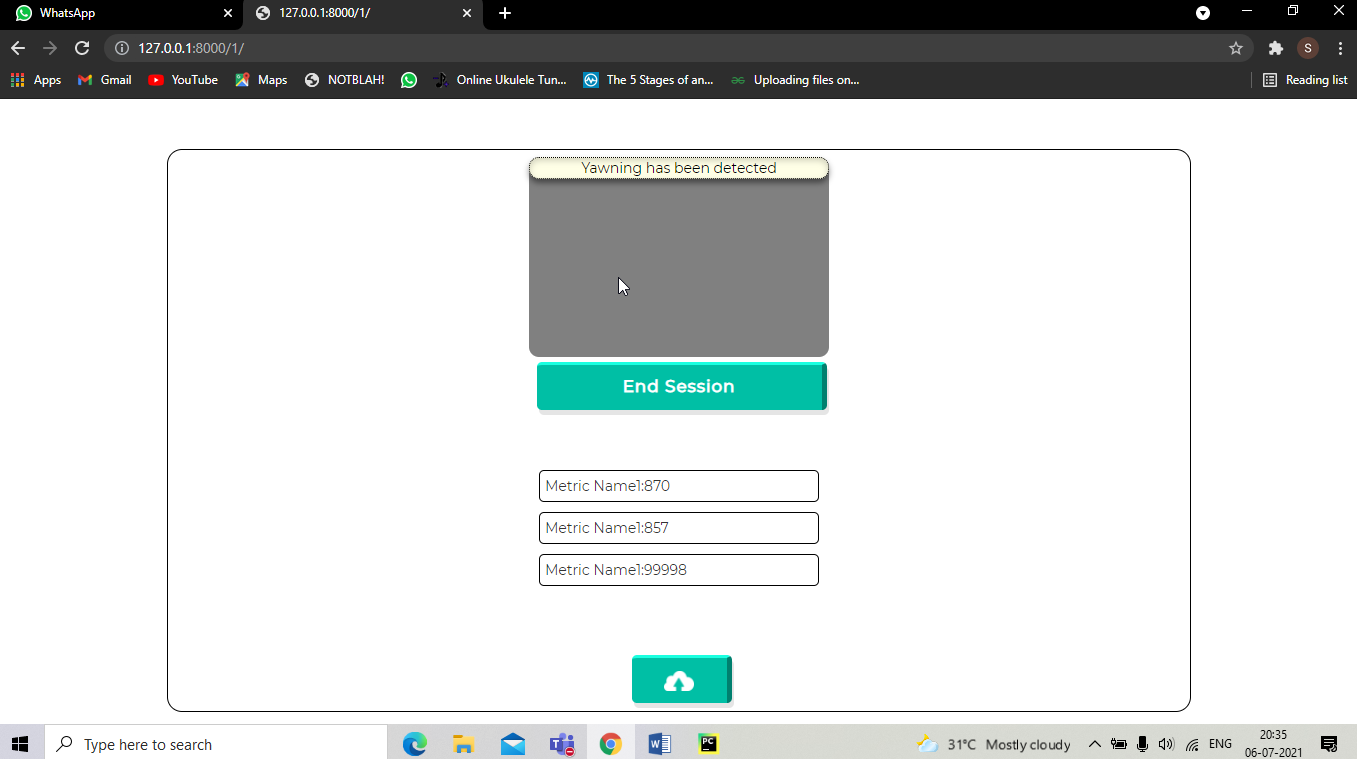


* This process continues till the end of the session, when we click on end session button, proctor system starts analyzes the metrics based upon the user behavior throughout the session.

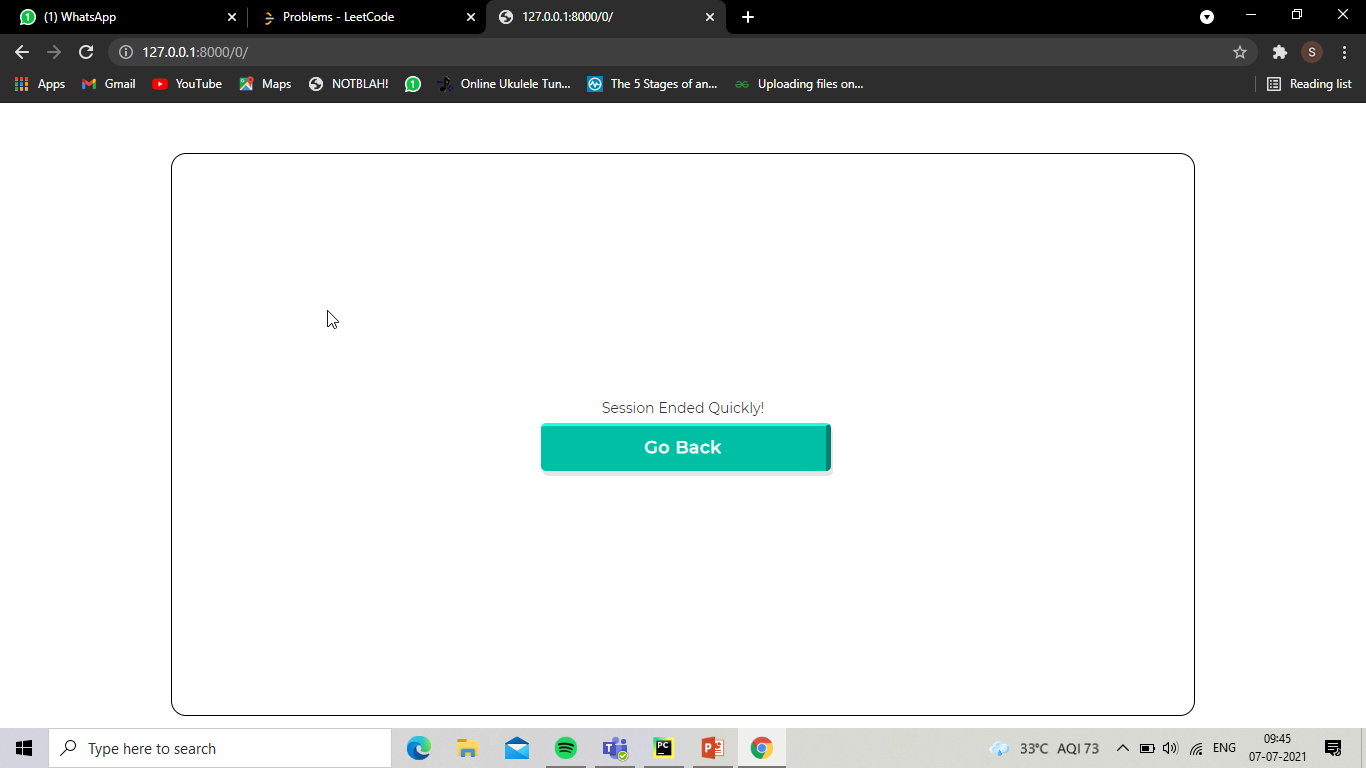
****



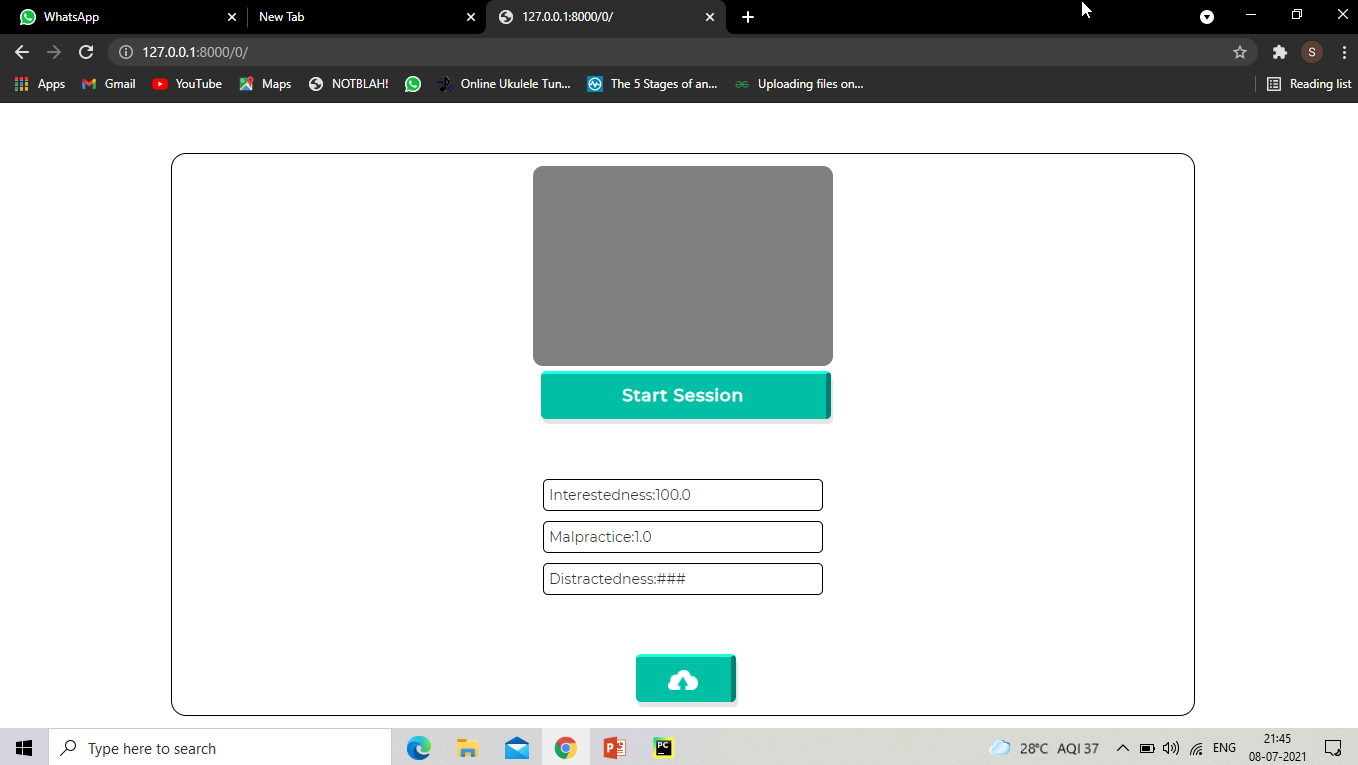
****

****

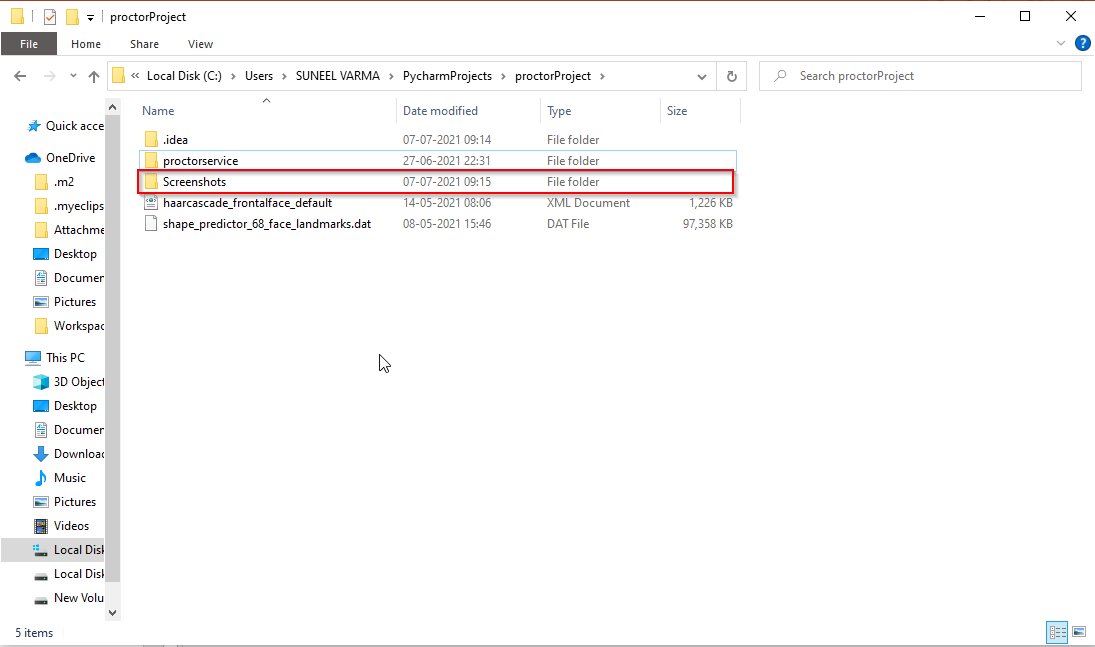
* This page keeps updating the gray window as long as monitoring is going on, we can find all proctoring updates in this window.
* Whenever something is detecting in the back-end system captures an image of user and saves it.
* We can end the session by clicking on end session button below the gray window.

****

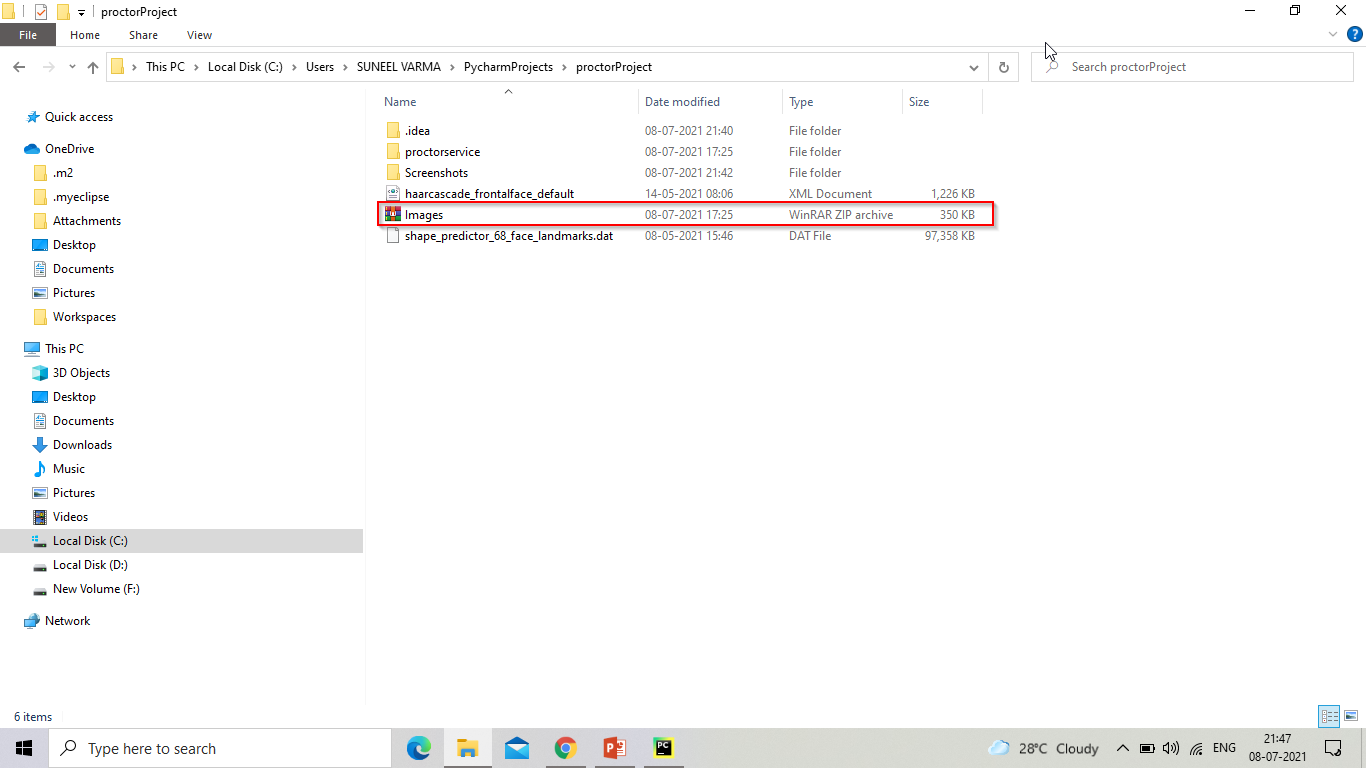
* If session ended within 1 minute, obviously we cannot judge user behavior with in this time frame, so we redirect to the main page again.
* An ideal session should go more than a minute.

****

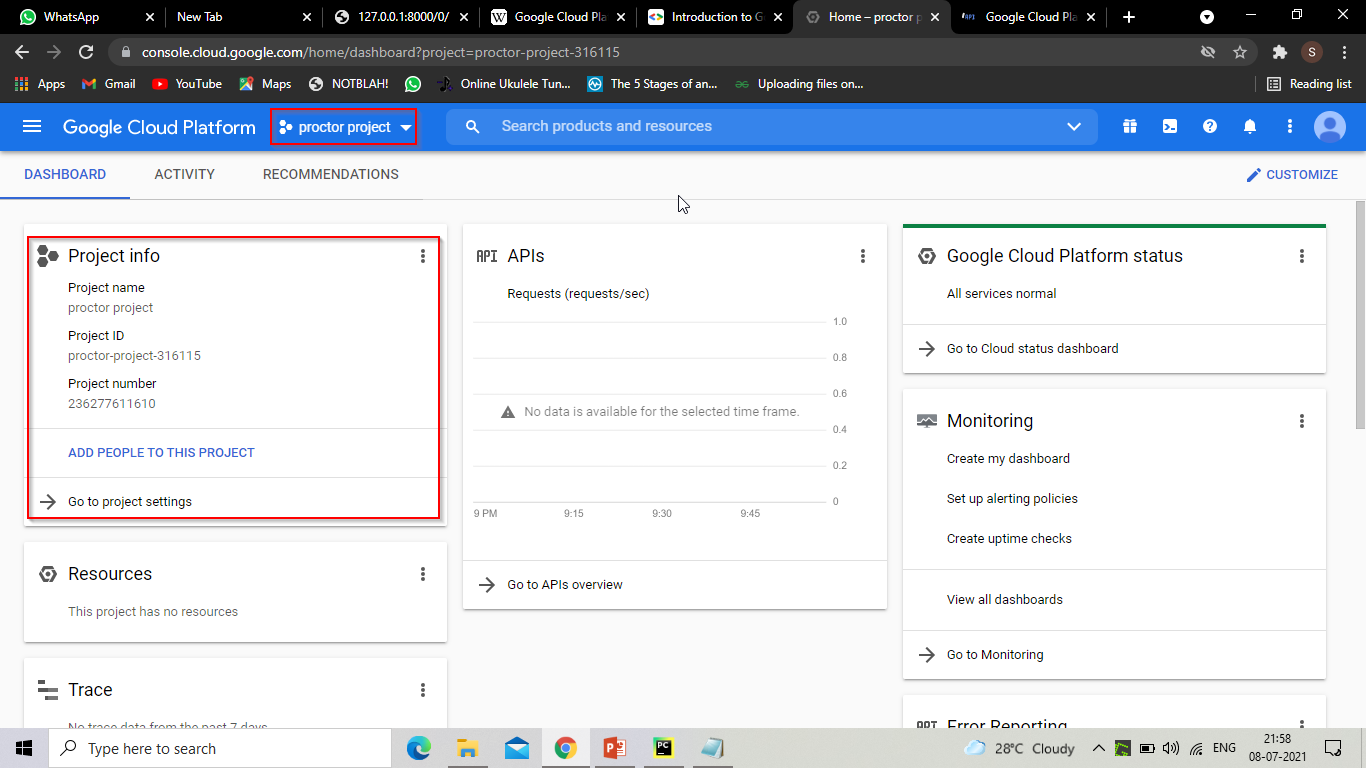
* After session has ended we will get the metric results.
* If metric results cross the threshold we can upload the images to cloud service by clicking on the upload button.



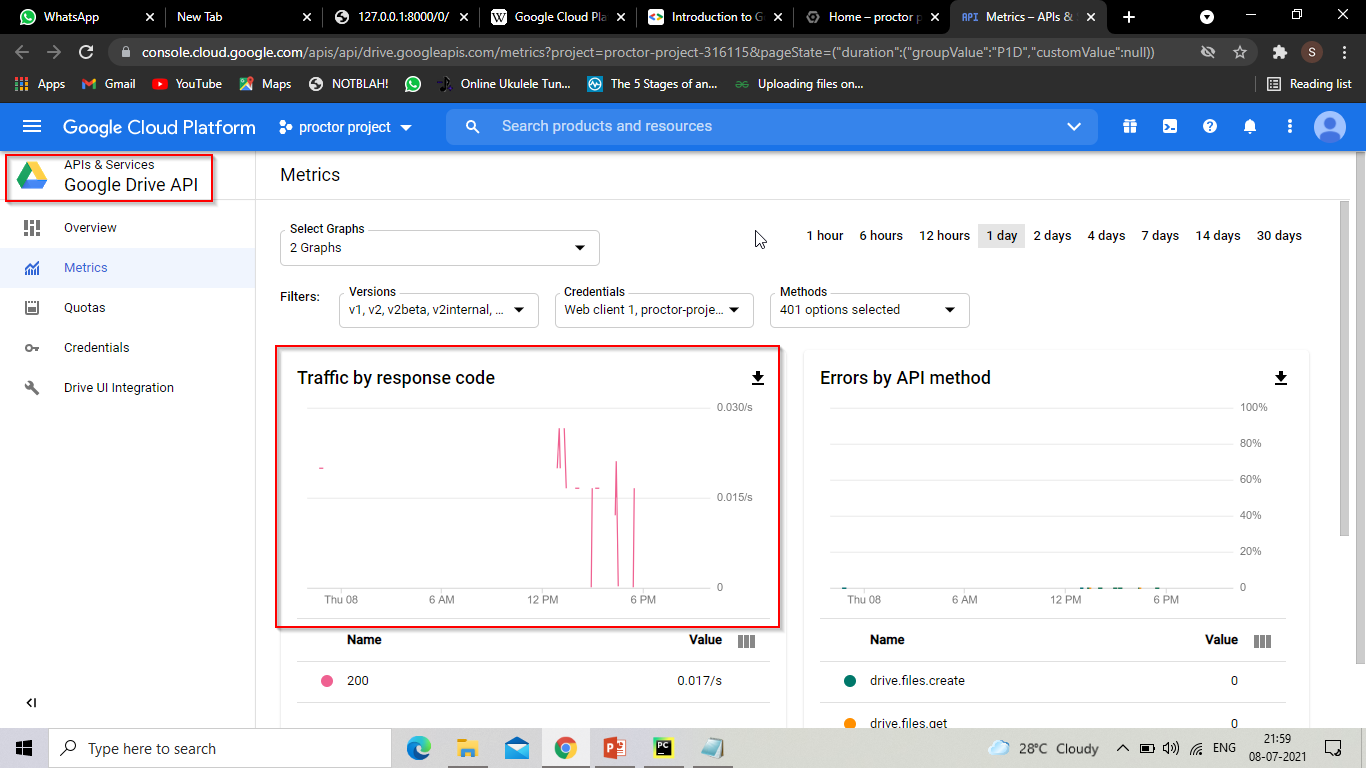
* A folder will be created at the start of the session
* In this folder all the captured content will be saved.

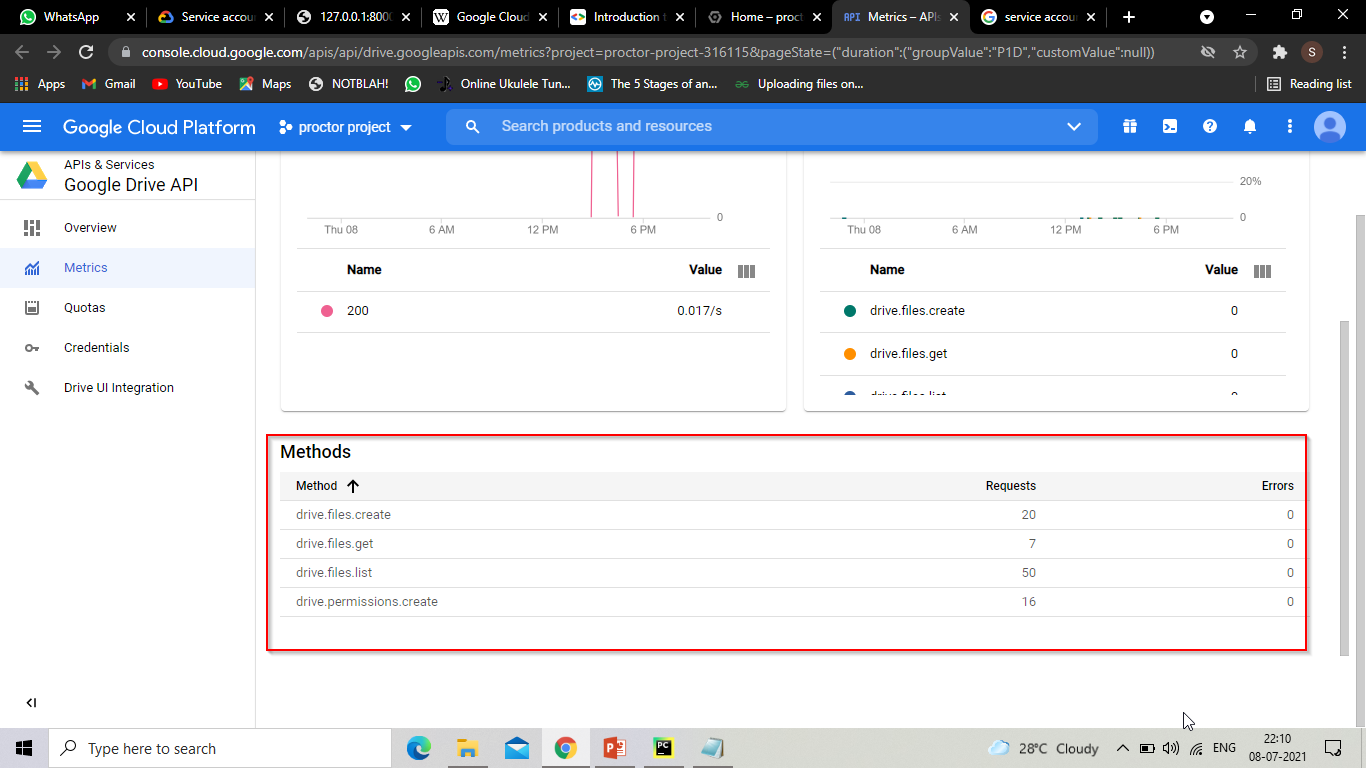


* The folder will be converted to a zip file, after optimizing all the images inside the folder, this zip file will be sent to cloud service in next immediate step.

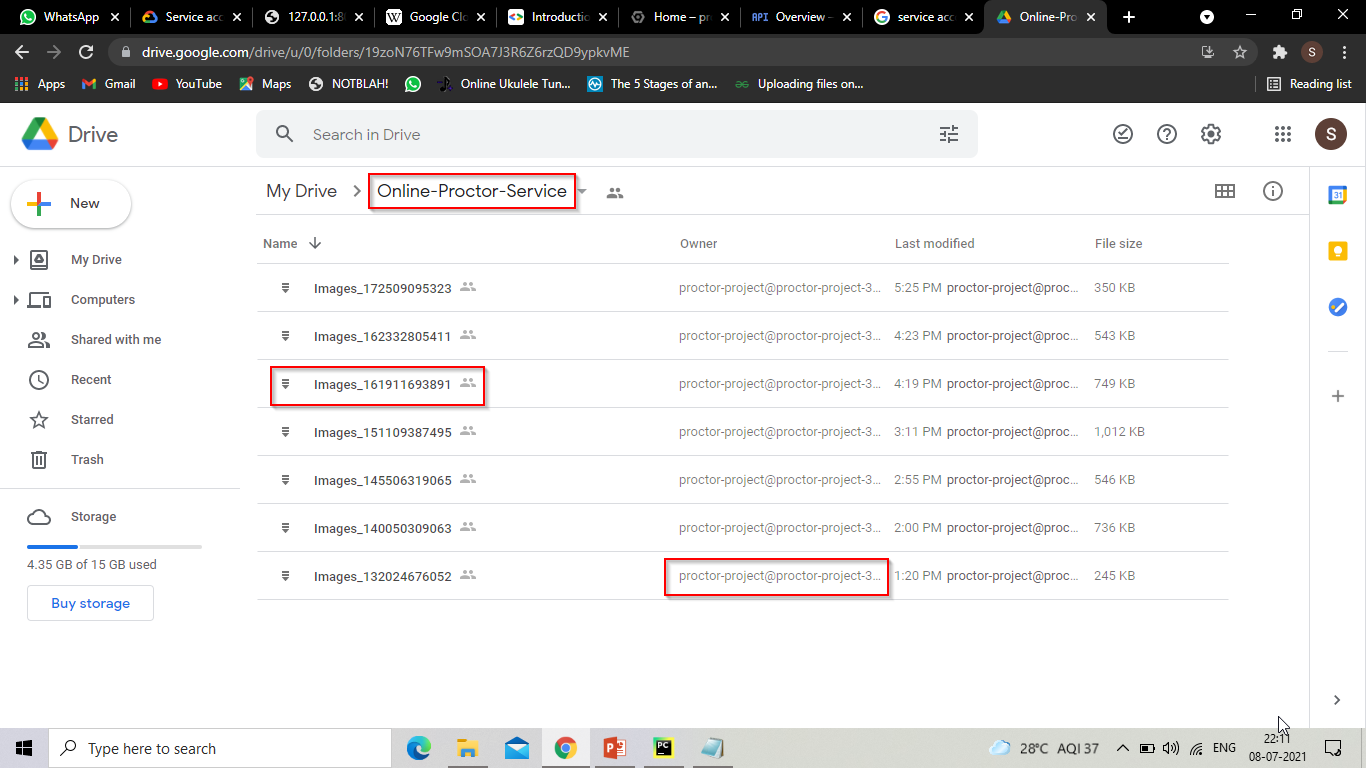


* Zipped folder will be sent to a cloud service which is Google Cloud Platform, through a google service account.
* A service account is operable only through API calls without an User interface.

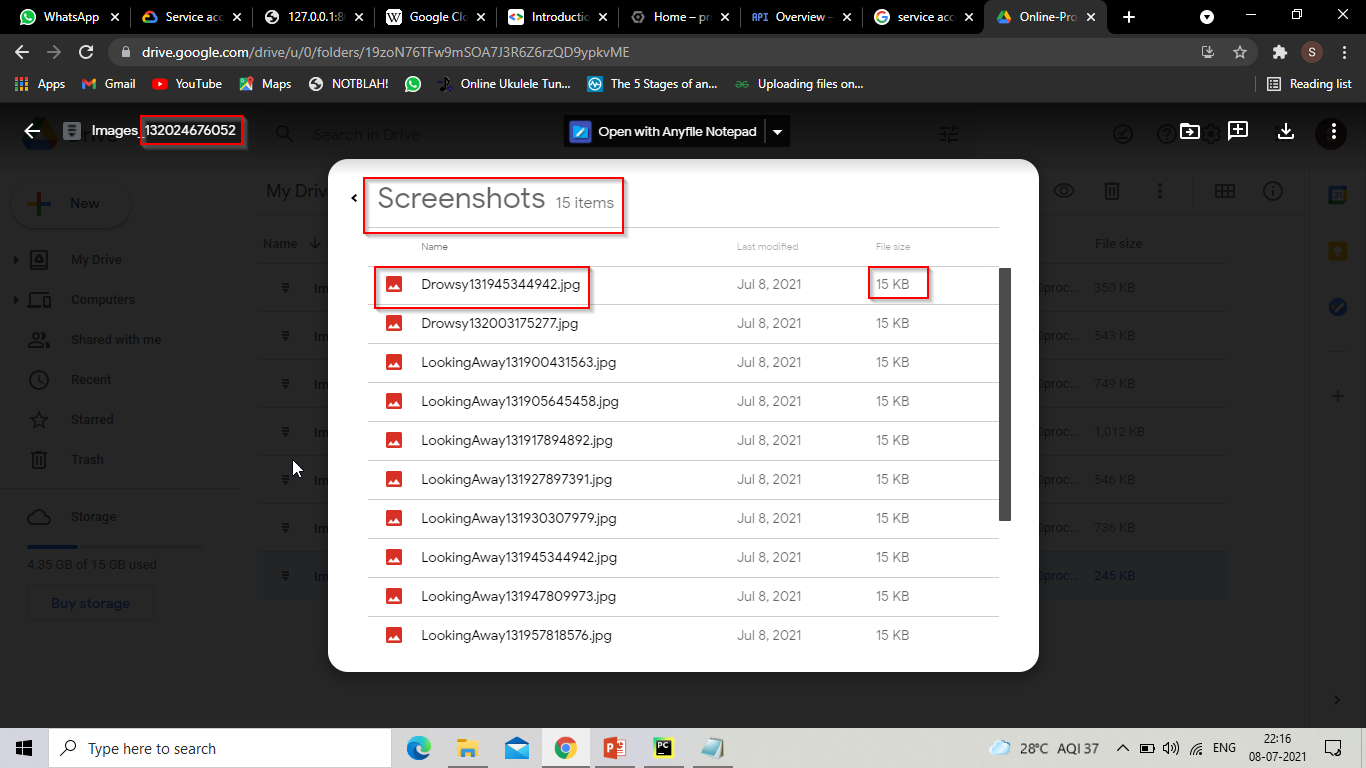




* The primary API which we are using for storing data is Google Drive API, which should be enable for the service account we are using.
* In this window we can track the number of API calls were made in history, in case of any errors occur in API, they can be tracked here in GCP.



* At the end, the content which we upload to cloud, would be accessible through Google Drive.
* In order to bring files to Admin Google drive, user has to give permissions to a file location where he/she wish to store files.
* Here, **service account** has access to admin folder ‘Online-Proctor-Service’, here admin can find all the captured content after end of session.



* Each image captured in session will be present here with respective label name.
* Here we can also find the consistent and lowest possible size of each image, which is the result of image optimization.

**6.2 REPORTS:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TID** | **Input** | **Description** | **Expected Result** | **Report** |
| **T01** | Frame captured from user web camera. | When the distance between upper eyelid and lower eyelid is less than threshold then it should be classified as closed. | Eye lids state should be closed.  Count should be incremented | PASS |
| **T02** | Frame captured from user web camera. | When the distance between upper eyelid and lower eyelid is greater than threshold then eyes should be classified as open. | Eye lids state should be open. | PASS |
| **T03** | Frame captured from user web camera. | When closed count greater than threshold then drowsiness has to be detected | Drowsiness should be detected | PASS |
| **T04** | Frame captured from user web camera. | When closed count less than threshold then drowsiness has not to be detected | Drowsiness should not be detected | PASS |
| **T05** | Frame captured from user web camera. | When the distance between upper lip and lower lip is greater than threshold and eyes state is not in shrink then it should be classified as not yawning. | Yawn should not be detected | PASS |
| **T06** | Frame captured from user web camera. | When the distance between upper lip and lower lip is greater than threshold and eyes state is in shrink then it should be classified as not yawning | Drowsiness should be detected | PASS |
| **T07** | Frame captured from user web camera. | When the distance between upper lip and lower lip is less than threshold and eyes state is in shrink then it should be classified as not yawning. | Yawn should not be detected | PASS |
| **T08** | Frame captured from user web camera. | When the distance between upper lip and lower lip is less than threshold and eyes state is not in shrink then it should be classified as not yawning | Yawn should not be detected | PASS |
| **T09** | Frame captured from user web camera. | When a mobile device is within the range of camera, then mobile has to be detected. | User is not using mobile | PASS |
| **T10** | Frame captured from user web camera. | Classify user is not using mobile, when user is not actually using a mobile. | User is using mobile | PASS |
| **T11** | Frame captured from user web camera. | When user is not looking at the screen. | User is looking away from screen | PASS |
| **T12** | Frame captured from user web camera. | When user is not looking at the screen. | User is not looking away from screen | PASS |
| **T13** | Frame captured from user web camera. | When more than one face has come within the range of camera | Multiple individuals are detected | PASS |
| **T14** | Frame captured from user web camera. | When only one face is within the range of camera | Multiple individuals are not detected | PASS |
| **T15** | When a path exists to the folder | Verify the captured content is saved into path location | Captured content should be saved inside dedicated folder path | PASS |
| **T16** | When content present inside the folder, content needs to be optimized | Verify the captured content is optimized or not | Captured content must be optimized, the size of folder should be reduced | PASS |
| **T17** | When a path exists to the folder | Verify the folder is in optimized state or not | Folder should be in optimized state | PASS |
| **T18** | When optimized folder is exists it should be zipped and renamed | Verify the folder is zipped and renamed | Folder should be zipped and renamed | PASS |
| **T19** | When a folder exists in optimized and zipped state | Verify the folder is optimized, zipped and renamed | Folder should be in optimized, zipped and renamed state | PASS |
| **T20** | Folder has to be uploaded to drop location in cloud service | Verify the folder is uploaded to cloud service or not | Folder should be present in drop location in cloud service | PASS |

**7. CONCLUSION AND FUTURE SCOPE**

**7.1 Conclusion:**

Interestedness of an individual is assessed by considering factors such as number of look away from screens, drowsiness and engaging with mobile devices. Malpractice can be judged based upon whether a user using other resources such as mobile devices or user seeking help from other individual during assessment this task is accomplished by using algorithmic approaches like 68 landmarks and Haarcascade frontal face for assessing user face and a deep learning model like yolo which will be trained with sample set of data to detect mobile device. Maintaining an average threshold to judge the interestedness and malpractice, if the user crosses the threshold limit, then the algorithm uploads the captured images of user to a cloud service which can be manually scrutinized to take decision in complex situations.

**7.2 Future Scope:**

The future scope describes the updates that will be released while releasing the next versions of the application,

* Providing a better user experience by enhancing the user interface.
* Providing better and interactive features for students where students can actually take up examination.
* Provide faster performance in terms of detecting objects.
* Online proctor service has high scope in the future as we see more institutions shift towards conducting exams virtually.

**8. BIBLIOGRAPHY**

**Books Referred:**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Book** | **Author** |
| 1. | Mastering Concurrency in Python: Create faster programs using concurrency. | Quan Nguyen |
| 2. | Python Crash Course: A Hands-On, Project-Based Introduction to Programming | Eric Matthes |
| 3. | Django for Beginners: Build websites with Python and Django | William S. Vincent |

**Websites referred:**

<https://stackoverflow.com>

https://[www.w3schools.com](http://www.w3schools.com/)

https://codepen.io

https://www.geeksforgeeks.org/

**Papers referred:**

<http://www.nmis.isti.cnr.it/falchi/Draft/2018-MMEDIA-Preprint.pdf>