Protracktor
Local Proctoring and
Activity Tracker

Quad-Core

Amit Kumar Bala Antu	203051003
Kunal Verma	203050121
Mallela Niteesh Kumar Reddy	203050065
Nimesh Aggarwal	203050049
Shubhranshu Maurya	203050096

Contents

1 Introduction

1.1 Motivation

Today due to the COVID-19 pandemic schools, colleges and students are facing an unprecedented situation which has led to online learning and conducting of proctored examinations through video conferencing a new normal.

This new pattern of learning and conducting exams has created several challenges for both students as well as teachers. Students are facing several issues related to prolonged screen time, network data related issues while attempting a proctored quiz, procrastination and sometimes inadvertently spending too much time on screen on certain unproductive activities.

We came up with an idea of **Protracktor** which will try to tackle the above issues to make this system more streamlined for students.

1.2 Objective

Through Protracktor our objective was to develop such an application which:

- Keeps track of network connectivity and also tries to maintain a report of data consumption by each process in use. This could help one to plan their daily data consumption by each application.

- Keeps track of your screen time - the current application in use - to give a better understanding of how much time on each process one is spending. This will help one to manage their screen time accordingly.

- During online proctored tests and quizzes students face network connectivity issues where if a student loses internet connectivity then they have to turn on the camera and screen recorder. While attempting the tests it creates too much hassle and distraction we also wish to automate this process and incorporate it in the application such that as soon the internet goes off it will start the recording.

2 Dependencies

- Python3
- OpenCV
- xdotools
- nethogs
- urllib
- Tkinter
- PyQt5

- PyAutoGUI

- time

- Threading

- subprocess

- matplotlib

- operator

- json

- notify2

- signal

- CSV

3 Using Protracktor

The initial build support only linux based platforms.

3.1 Getting dependencies ready

In order to use Protracktor you first need to make sure you have all the dependencies installed. If you haven't installed them don't worry we have your back. We have made an installer.sh file which will take care of all the dependencies and requirements for the project.
To run installer.sh, go to the project directory "Protracktor>source" where installer.sh is present and then run the following commands

```
$ chmod +x installer.sh
$ ./installer.sh
```

If some requirements are still missed please follow the steps below to manually install the dependencies.
Install python on your system by running the following command on your terminal:

```
$ sudo apt-get install python3.6
```

Screen activity module needs `xdotool` to get window's information.

```
$ sudo apt-get install xdotool
```

For Internet Tracking Module install `nethogs`

```
$ sudo apt-get install nethogs
$ sudo setcap "cap_net_admin,cap_net_raw=ep" /usr/sbin/nethogs
```

To install `PyAutoGUI` additionally you need to install the `scrot` application, as well as Tkinter and other python libraries requirement:

```
$ sudo apt-get install scrot
$ sudo apt-get install python3-tk
$ sudo apt-get install python3-dev
$ sudo pip3 install pyautogui
$ sudo pip3 install python3-xlib
$ sudo pip3 install numpy==1.18.5
$ sudo pip3 install notify2==0.3.1
$ sudo pip3 install matplotlib==3.3.1
$ sudo pip3 install opencv_python==4.1.2.30
$ sudo pip3 install PyQt5==5.15.1
```

All the functionalities and how to use them is depicted ahead in the next section.

4   How To-Guide

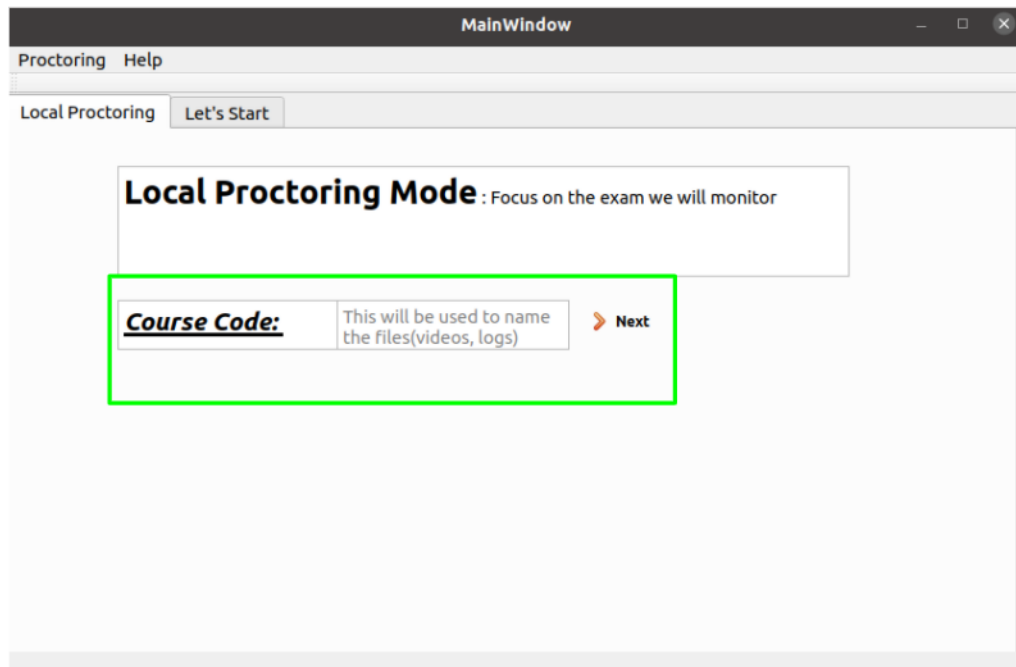After installing all the dependencies and the requirements only thing left is see how to operate our program.
Go to the main directory "Protracktor" and in that go to source directory. After that run the following command in terminal

```
$ python3 main.py
```

This will start the application!

4.1   Main window

The main window has two tabs namely "Local Proctoring" and "Let's Start"

1. First enter the course code you want to start proctoring for.

2. This field is used to name the log files and video files with additional timestamps added as a naming convention for files

3. Click on "Next" to move to the next tab "Let's Start"

On clicking next after entering the course code you will be directed to the next tab ("Let's Start" tab).
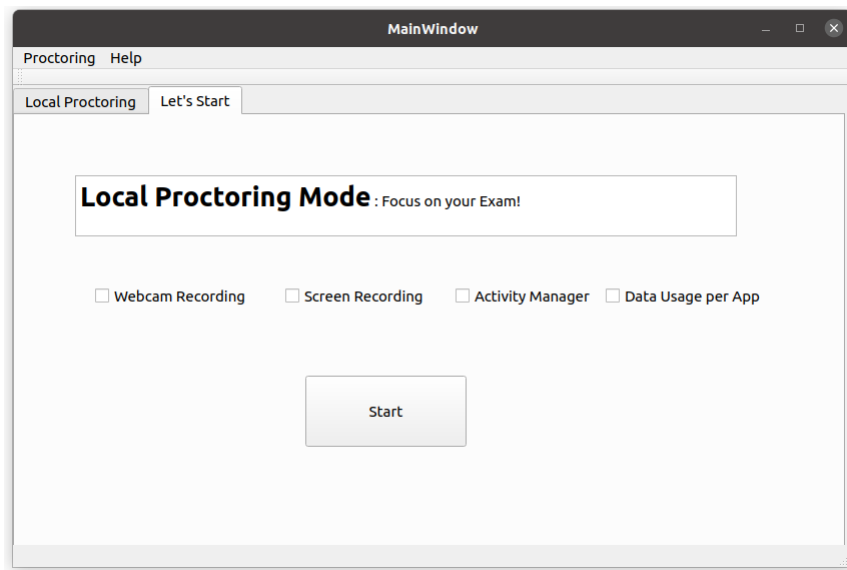
1. Here there are four check boxes named according to the functionalities they will provide, namely:

   (a) Webcam Recording
   (b) Screen Recording
   (c) Activity Manager
   (d) Data usage per app

2. Check on the required functionalities you want monitored during the proctoring.

### 4.1.1 functionality of each checkbox/function

- Webcam Recording: Accesses the webcam and records
  Note: Internet loss triggered - Webcam recording will only start if there is a loss in internet connectivity detected
  This assumes that you are already on some proctoring platform like ms teams initially and if there is a loss of internet connectivity then only webcam recording will start.

- Screen Recording: Records the entire computer screen
  Note: Internet loss triggered - Screen recording will only start if there is a loss in internet connectivity detected
  This assumes that you are already on some proctoring platform like ms teams initially and if there is a loss of internet connectivity then only webcam recording will start.

- Activity Manager: Monitors the activities related to which apps were used during the proctoring

- Data usage per app: monitors the data used by per app during the proctoring

After checking on the appropriate check boxes you can start the monitoring by clicking on "Start" button.
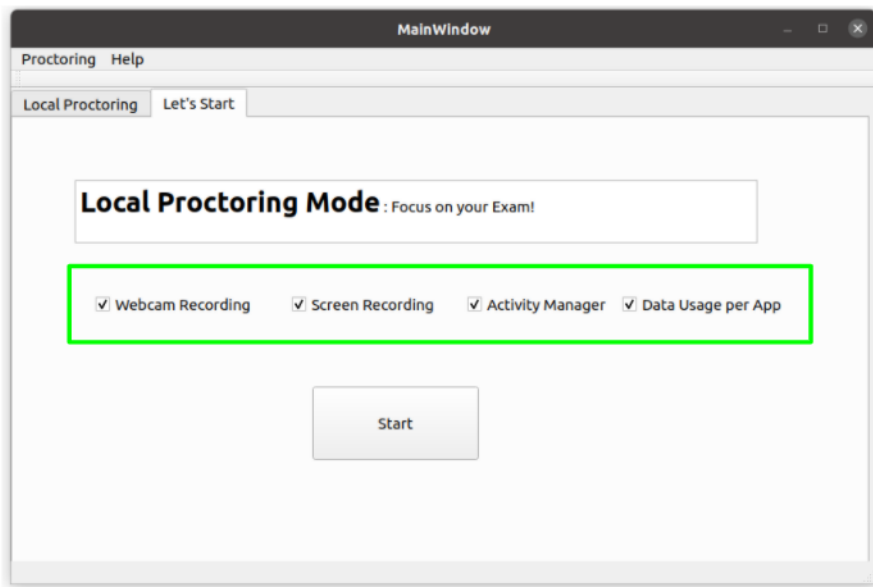
After clicking on start a new pop-up window will open
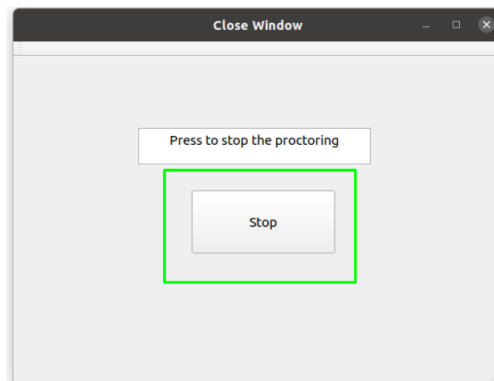
Whenever you want to stop the proctoring/monitoring just click on the stop button.

This will close the application also.
All the corresponding outputs will be saved in the "Results" directory.

4.2   Action bar - viewing the results

4.2.1   Proctoring

This option will direct you to various log files and graphs generated by the software. By clicking on this Menu Option you will get a drop-down with following options.

- Results
    - Data Usage
    - Activity Monitor
    - Screen Recordings
    - Video Recordings

- Graph
    - Latest activity tracker
    - Latest Data Usage

The Results section will take you the corresponding folder where log files and graphs are stored.
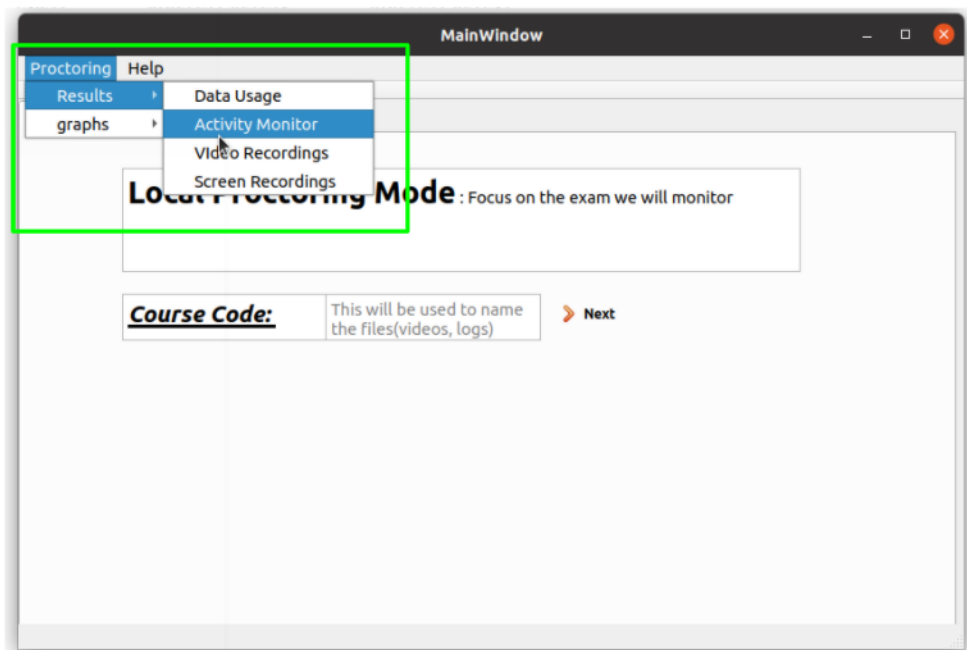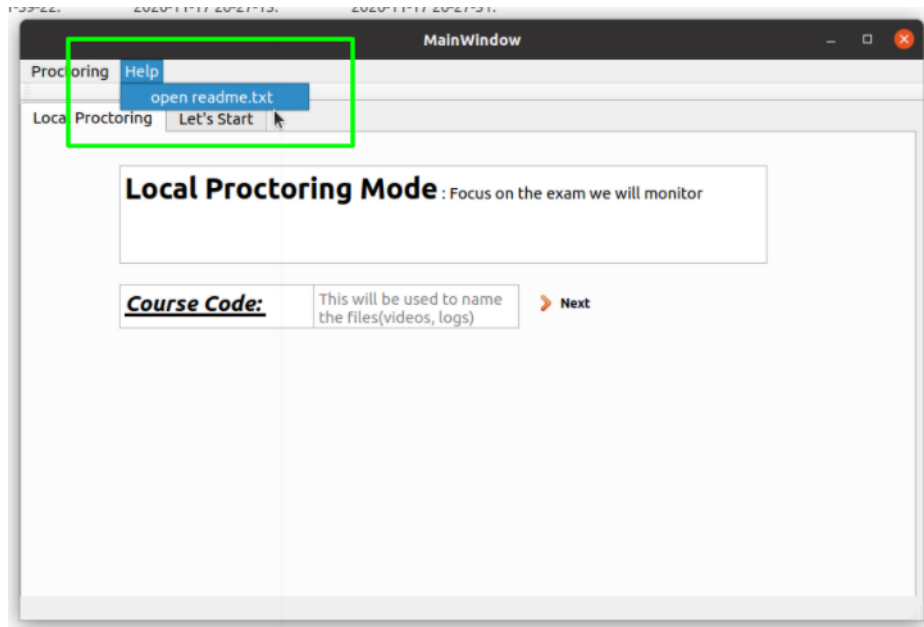
The Graph section will display the latest graph of the corresponding option you will select.
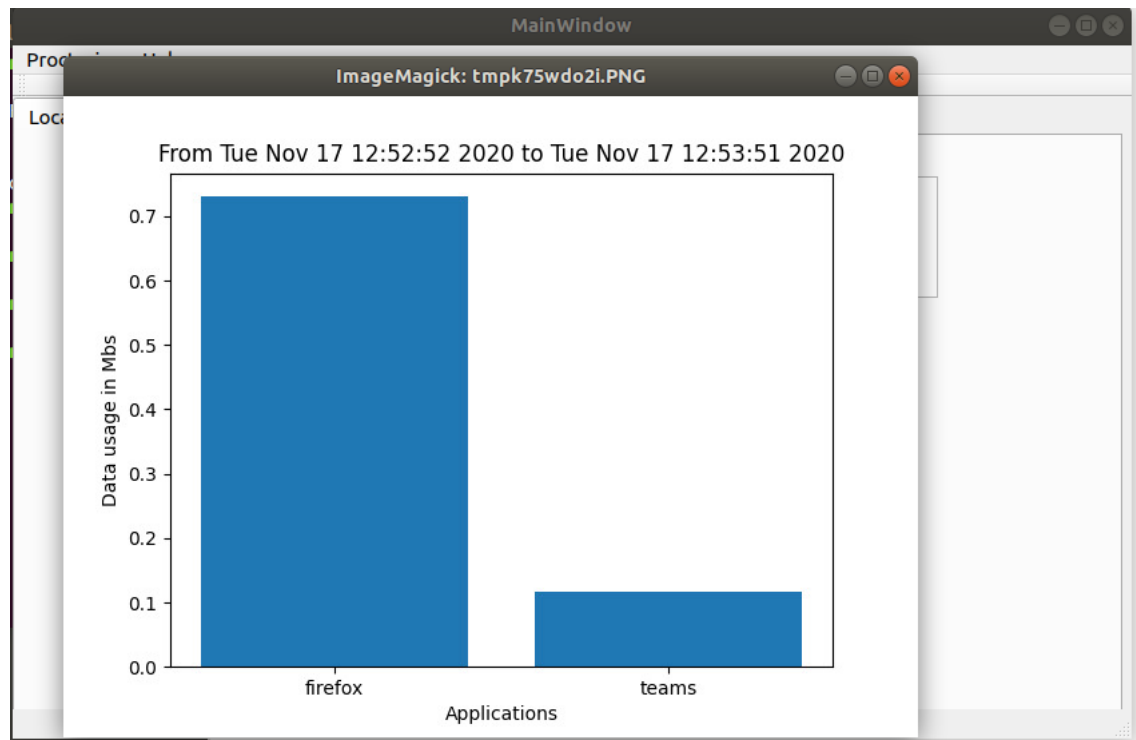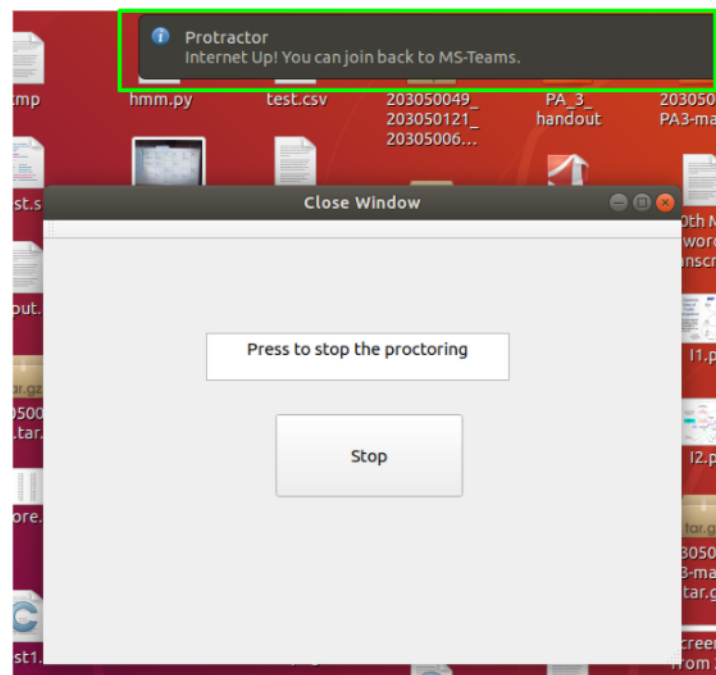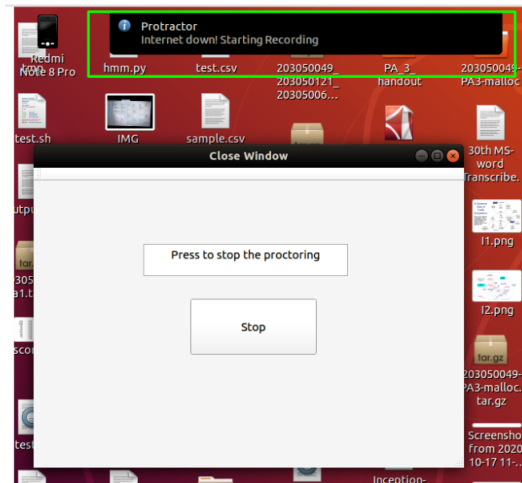
4.2.2   Help

This option will display the open readme.txt to you which will direct you to the readme file.

4.3   Notification

Two types of notification are issued to the user.

- When Internet goes down.
- When Internet comes back.

5   Usefulness

The current scenario of online semester and online proctoring will most likely to continue in future and this local proctoring program is the first step in easing both students and invigilators burden by hassle free management due to loss of internet connection. The data generated in the form of log files, videos, time

stamps, data usage, and activity usage can be used very efficiently for training of future AI based proctoring models. The initial runs of the program on the current video based proctoring through online channels were really useful. If used by the students or invigilators they will surely appreciate the usefulness of video recording and screen recording at the instant of internet loss and they can focus on the exam rather than focussing on the setup.

6 Future Enhancements

We would like to make the platform more robust with more functionalities. We would like to make the program support multiple platforms.
This Would be helpful in gathering data and making ai based proctoring systems

7 References

1. https://askubuntu.com/questions/779873/is-there-software-which-time-tracks-window-application-usage/780542780542

2. https://www.geeksforgeeks.org/bar-plot-in-matplotlib/