PayPal password: Sharanseptember19!

Apigee: Sharanseptember19!

Heroku password: sharanseptember19!

Microsoft teams password: Gokunarutoash19!

AWS: sharanseptember19!

Sharan19

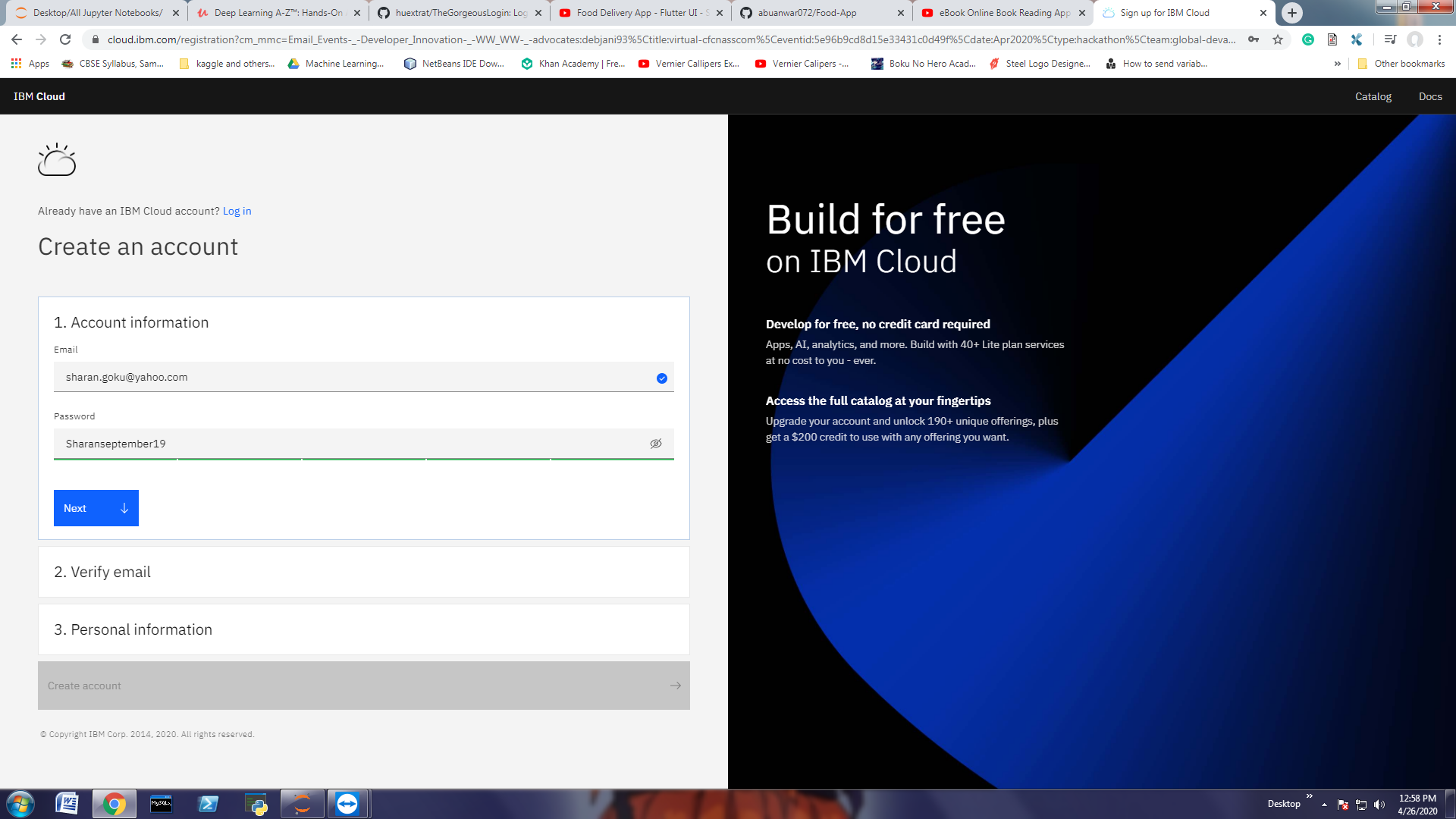
Our project is a system that utilizes neural networks and computer vision to allow users to control their desktop using their gaze. This could be very helpful for people with physical disabilities.

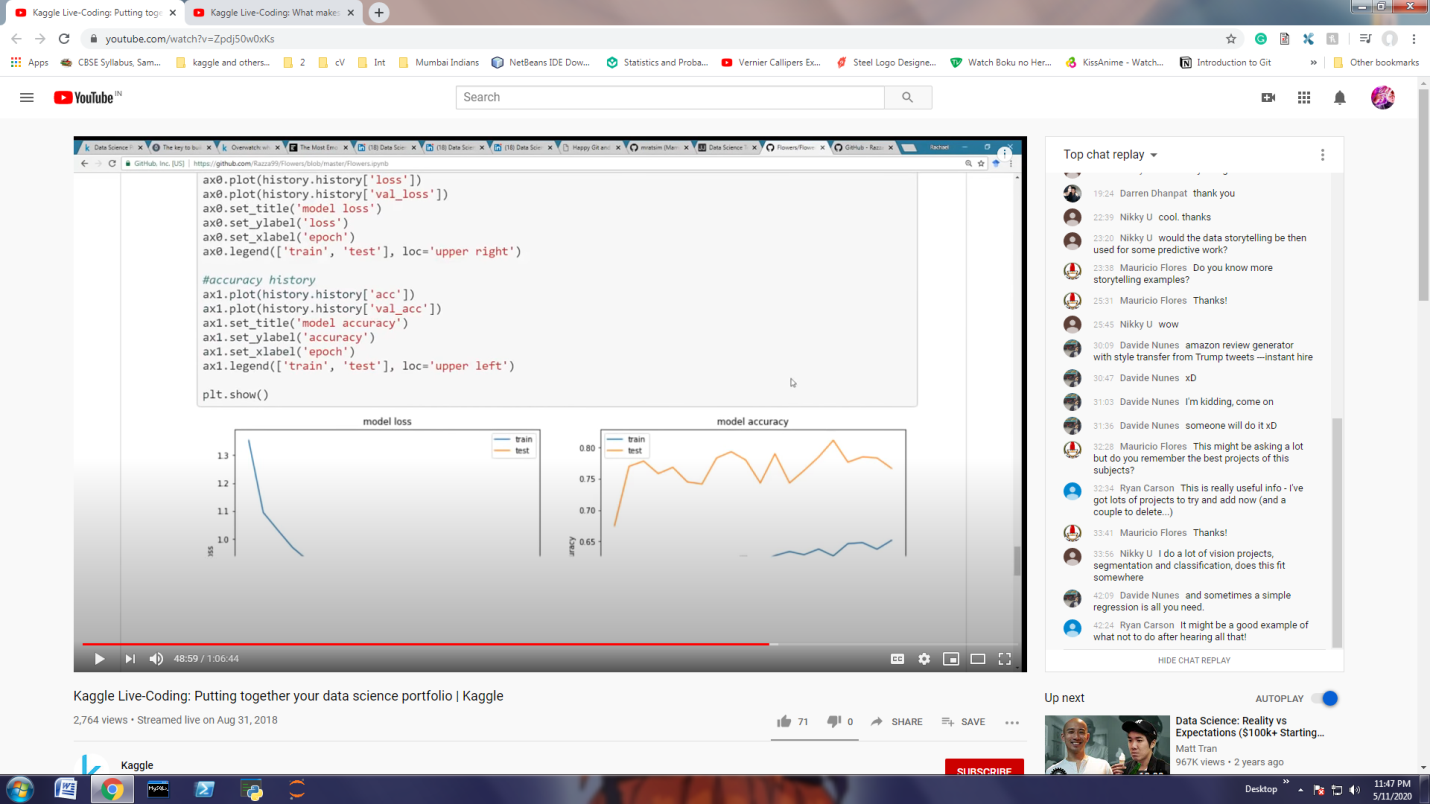
Run shell commands in jupyter notebooks using ! mark. To use python variable inside shell command wrap it inside {}

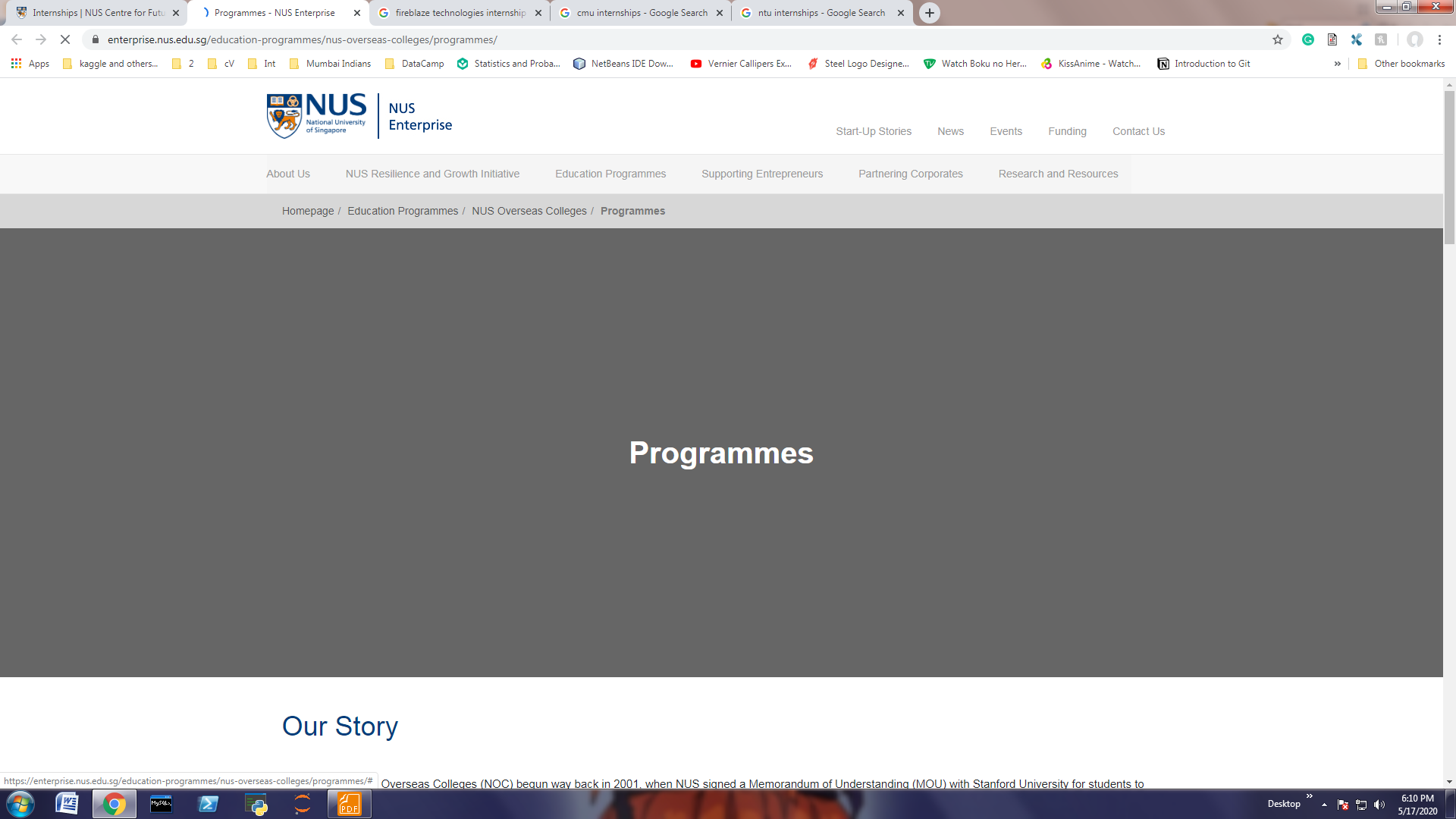
Look for documentation using ? symbol followed by function name.

Similarly ?? for source code of the function

Minute 65

1. Group by, Where, Having, distinct, union, intersect, count, limit, sum, max, min, avg, etc…the basics
2. Filtering on IS NULL, IS NOT NULL
3. Join multiple tables on primary / foreign keys
4. Case Whens to create categories
5. Datetime functions like “Interval”, Datediff()
6. Rank(), DenseRank(), row\_number() and how to use window (OVER) functions in the manner of “PARTITION BY….ORDER BY”. An interview question I’ve gotten is “how do you get the third date descending in a table”
7. Very helpful to know CTEs and being able to UNION ALL several different tables together
8. I think it’s nice to know recursive / iterative queries but this would be a pretty advanced level and not really required in most cases.
9. 





## Objective:

A Web and mobile application that helps people deal with skin cancer related problems. Our solution tries to be a bridge between doctors and patients suffering from skin cancer.

## Implementation:

We wish to implement this idea of ours using Python, Flask web framework , Flutter SDK for cross-platform app development and Firebase/MySql as the database to store patient details. We also plan to scale the application by deploying it on Heroku or Google Cloud Platform.

## Applications:

Patients can upload their skin pictures to doctors or directly onto the website from where they can infer whether they have skin cancer or not. They will also have an AI chatbot and forum to help with prevention steps and suggestions. Moreover, doctors will able to upload skin images received from patients and check if the patient's condition is improving or deteriorating. Doctors will also be able to de-haze(removes blur) the images and increase image pixel quality(pixel resolution upscaling) with our website which will enable them to make better decisions, thereby saving lives of people.

Our solution can be extended to help doctors of all hospitals to better manage their patient cases and make data driven decisions.

CREATE TABLE client (

client\_id INT PRIMARY KEY,

client\_name VARCHAR(40),

branch\_id INT,

FOREIGN KEY(branch\_id) REFERENCES branch(branch\_id) ON DELETE SET NULL

);

CREATE TABLE works\_with (

emp\_id INT,

client\_id INT,

total\_sales INT,

PRIMARY KEY(emp\_id, client\_id),

FOREIGN KEY(emp\_id) REFERENCES employee(emp\_id) ON DELETE CASCADE,

FOREIGN KEY(client\_id) REFERENCES client(client\_id) ON DELETE CASCADE

);

CREATE TABLE branch\_supplier (

branch\_id INT,

supplier\_name VARCHAR(40),

supply\_type VARCHAR(40),

PRIMARY KEY(branch\_id, supplier\_name),

FOREIGN KEY(branch\_id) REFERENCES branch(branch\_id) ON DELETE CASCADE

);

INSERT INTO employee VALUES(100, 'David', 'Wallace', '1967-11-17', 'M', 250000, NULL, NULL);

INSERT INTO branch VALUES(1, 'Corporate', 100, '2006-02-09');

UPDATE employee

SET branch\_id = 1

WHERE emp\_id = 100;

INSERT INTO employee VALUES(101, 'Jan', 'Levinson', '1961-05-11', 'F', 110000, 100, 1);

INSERT INTO employee VALUES(102, 'Michael', 'Scott', '1964-03-15', 'M', 75000, 100, NULL);

INSERT INTO branch VALUES(2, 'Scranton', 102, '1992-04-06');

UPDATE employee

SET branch\_id = 2

WHERE emp\_id = 102;

INSERT INTO employee VALUES(103, 'Angela', 'Martin', '1971-06-25', 'F', 63000, 102, 2);

INSERT INTO employee VALUES(104, 'Kelly', 'Kapoor', '1980-02-05', 'F', 55000, 102, 2);

INSERT INTO employee VALUES(105, 'Stanley', 'Hudson', '1958-02-19', 'M', 69000, 102, 2);

*-- Stamford*

INSERT INTO employee VALUES(106, 'Josh', 'Porter', '1969-09-05', 'M', 78000, 100, NULL);

INSERT INTO branch VALUES(3, 'Stamford', 106, '1998-02-13');

UPDATE employee

SET branch\_id = 3

WHERE emp\_id = 106;

INSERT INTO employee VALUES(107, 'Andy', 'Bernard', '1973-07-22', 'M', 65000, 106, 3);

INSERT INTO employee VALUES(108, 'Jim', 'Halpert', '1978-10-01', 'M', 71000, 106, 3);

*-- BRANCH SUPPLIER*

INSERT INTO branch\_supplier VALUES(2, 'Hammer Mill', 'Paper');

INSERT INTO branch\_supplier VALUES(2, 'Uni-ball', 'Writing Utensils');

INSERT INTO branch\_supplier VALUES(3, 'Patriot Paper', 'Paper');

INSERT INTO branch\_supplier VALUES(2, 'J.T. Forms & Labels', 'Custom Forms');

INSERT INTO branch\_supplier VALUES(3, 'Uni-ball', 'Writing Utensils');

INSERT INTO branch\_supplier VALUES(3, 'Hammer Mill', 'Paper');

INSERT INTO branch\_supplier VALUES(3, 'Stamford Lables', 'Custom Forms');

*-- CLIENT*

INSERT INTO client VALUES(400, 'Dunmore Highschool', 2);

INSERT INTO client VALUES(401, 'Lackawana Country', 2);

INSERT INTO client VALUES(402, 'FedEx', 3);

INSERT INTO client VALUES(403, 'John Daly Law, LLC', 3);

INSERT INTO client VALUES(404, 'Scranton Whitepages', 2);

INSERT INTO client VALUES(405, 'Times Newspaper', 3);

INSERT INTO client VALUES(406, 'FedEx', 2);

*-- WORKS\_WITH*

INSERT INTO works\_with VALUES(105, 400, 55000);

INSERT INTO works\_with VALUES(102, 401, 267000);

INSERT INTO works\_with VALUES(108, 402, 22500);

INSERT INTO works\_with VALUES(107, 403, 5000);

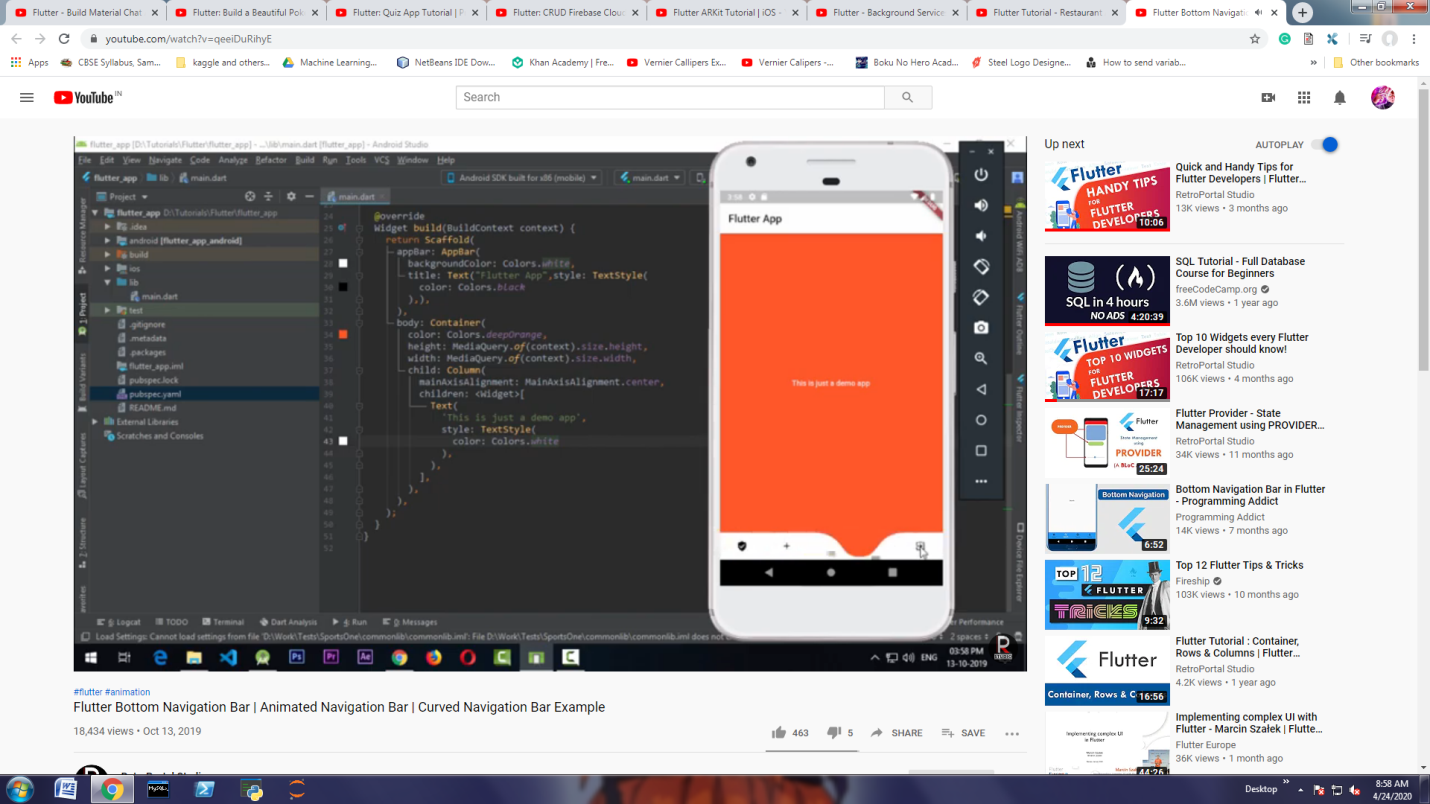
INSERT INTO works\_with VALUES(108, 403, 12000);

INSERT INTO works\_with VALUES(105, 404, 33000);

INSERT INTO works\_with VALUES(107, 405, 26000);

INSERT INTO works\_with VALUES(102, 406, 15000);

INSERT INTO works\_with VALUES(105, 406, 130000);



A video conference platform for webinar and lecture hosting enhanced by high level of security, neat User Interface and analytics.

We make use of state of the art technologies like face recognition and face spoofing to ensure that only people who received the invite to join the call are able to do so. The app prompts you to upload a photo of your face or send it to the host.

Later, when a person wishes to join a meeting/webinar, he would be asked to post a live photo. This photo will be compared against the original photo to ensure that the person attending the meeting is the one we had sent the link to.

Upon this, we also run Face Spoofing algorithms to ensure that a printed photo or the identity card of the attendee is not falsely used to bypass the security check. You will not be allowed to join others in the meeting if you fail the face comparison test or spoof test and would have to request the host for a One Time Password (OTP) for special access.

Upon entering the meeting, you can use all the video conferencing features that the host has enabled and these include common features like chat and muting video/voice.

The scope for an app with these features is immense and is the need of the hour as we have been hearing about security breaches which could prove to be fatal in case of high profile meeting. We have also been hearing about the rising cases of zoom bombing wherein random people enter the meeting without permission or by breaking through the security and shout inappropriate phrases which deprive the quality of meaningful meetings and induces fear and disgust in the community.

Our app does not use any information collected from the user whatsoever for any kind of purposes ensuring the highest level of security possible.

The app can further be extended to monitor facial landmarks to study the attention and interest displayed by attendees, which could be helpful in case of office meetings or lecture classes for students as this would enable teachers/professors to develop efficient teaching methodologies. Monitoring the grasping rate of the child can play a huge role in imparting quality education. The results so obtained can be visualized on a beautiful dashboard for effective interpretation.

Work from Home and online learning has become today's need and is growing rapidly for the good. It is our job to ensure the same quality that you otherwise get in an off-screen experience from the comfort of your home.

Our app is easy to use meaning teachers will not have to spend too much time up skilling themselves to teach students.

Another growing area of concern among people of all age groups is the rising anxiety and burnouts caused due to change in work methodology and excessive screen time. Our app tries to address this solution with calm, beautiful and eye-soothing User Interface.

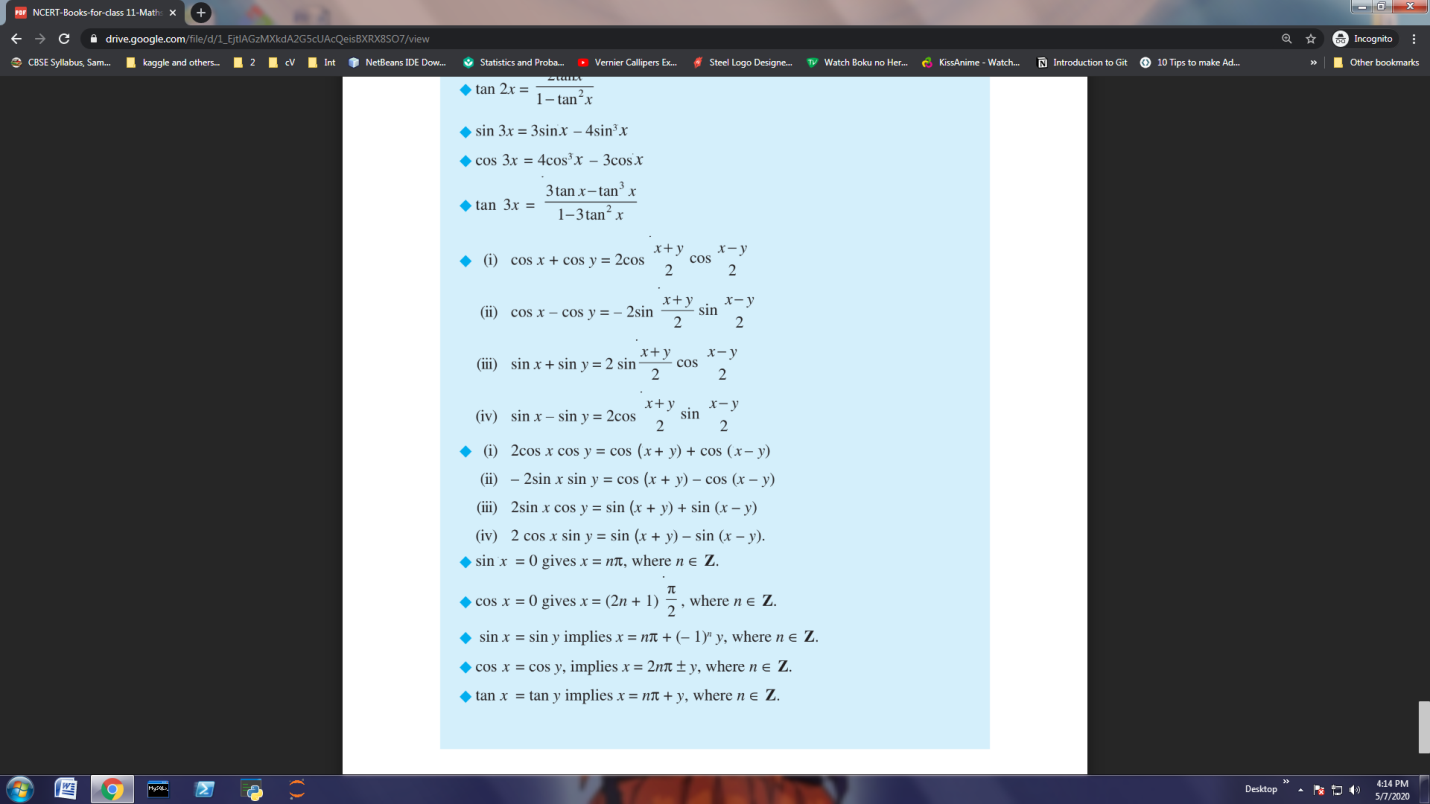
Finally, an application that ensures quality of life and security is the need of the hour and we are confident that our app gets those spot on!

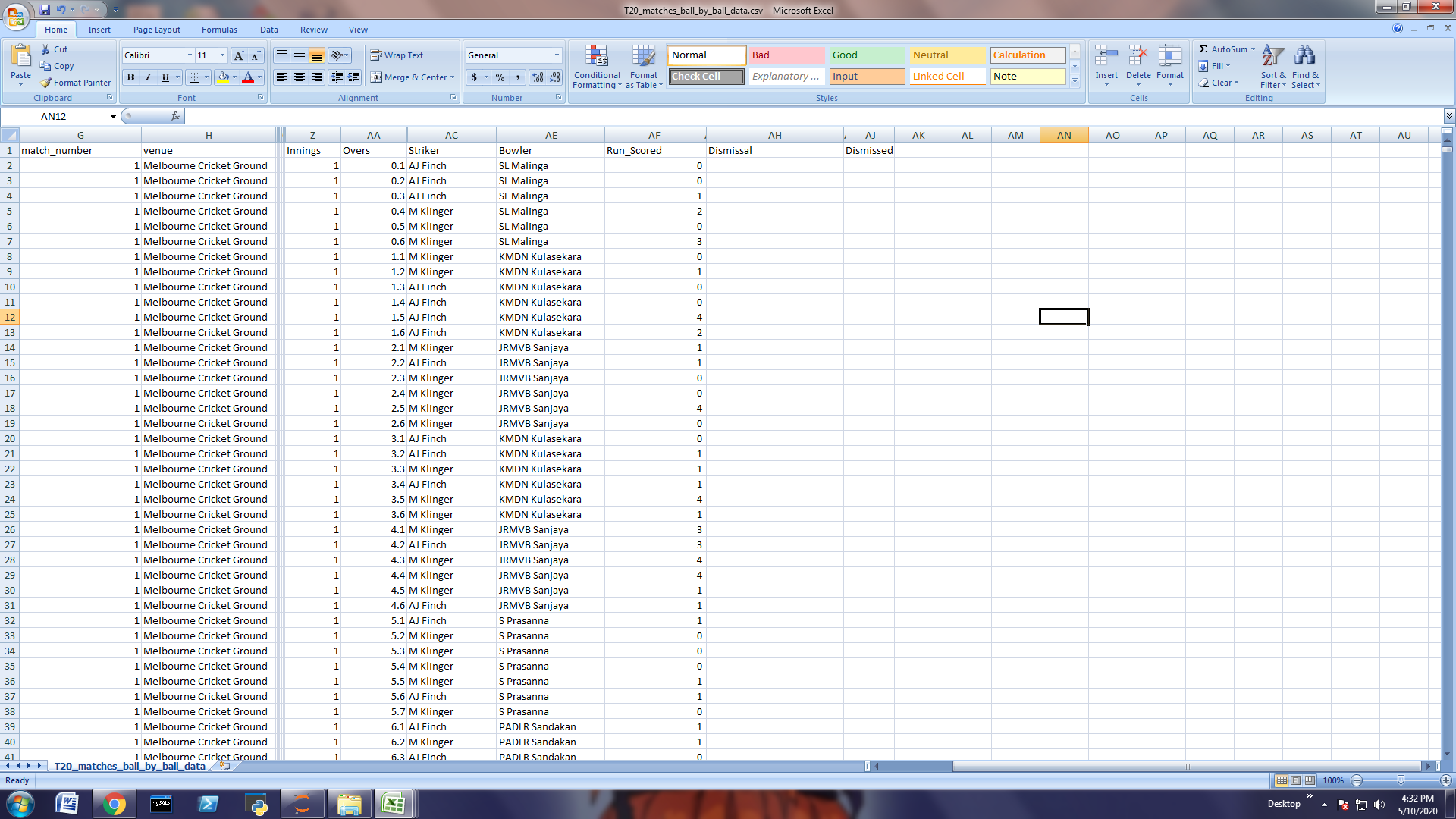
1) Anti-cheat computer vision software for conducting examinations in an online medium (mobile phone). One way to implement it could be using Deep Learning (Convolutional Neural Networks) to check the video stream for potential malpractices like open books, verbal discussion of answers, etc. There is no one in the market who does this right now. So, the first one to enter this space will have a high chance of winning.

2) “Classroom feature” to provide value to schools and coaching centers. We can safely assume that a student would not be enrolled in a coaching centre/ tuition and an E-learning app at the same time. Ability of their teachers to upload videos of their teachings along with the Byju’s videos would supplement the learning of the child.

(Business model: Ideally, since the teacher will be teaching the concepts, he/she could need all the chapter videos or only a select few. Hence, with this feature the teacher can rent videos of a particular topic from Byju’s for ‘X’ amount for a duration of 1 year).

3) If features like letting teachers create quizzes and sharing documents to their respective classrooms were added to the above concept, it would build an online eco-system thereby reducing student attrition and increase new enrollments as more students would explore the platform when they log in to attend their teacher’s online classroom.



****

## Inspiration

One of our team members has a disabled younger sister. She always struggles with using her computer, because her arms disability does not allow her to use the mouse or a keyboard. He told about it to the rest of our team and we understood that there are millions of great people, who are not able to use their electronic devices as efficiently as they potentially could, because of some uncontrollable circumstances. Because of that, many potentially brilliant writers, engineers, musicians, etc. could not conveniently follow their passions and create great things for humanity, and enjoy their lives a bit more. We decided that we are willing and we are capable of taking action towards helping those people. We decided to build software that solves this problem for physically disabled people and makes this world a little bit better.

## What it does

Our first prototype enables people to navigate around their desktop, using only their eyes. This includes moving the mouse and pressing the buttons.

## How we built it

We built the base in Python 3, we used PyQt5 to create a GUI. Then we used OpenVINO and dlib library's pre-trained models and applied some transfer learning on them, in order to estimate the real-time gaze direction. We used OpenCV for detecting blinks and pyautogui library for controlling the mouse movements, and we used Docker for containerizing our application.

## Challenges we ran into

First and the biggest challenge was the optimization of the inference of our deep learning models in order to make our app run faster.

Solving all the version compatibility issues(we ended up using Docker to solve this problem).

## Accomplishments that we're proud of

The biggest accomplishment that we are proud of is the fact that we were able to turn what we had imagined into life.

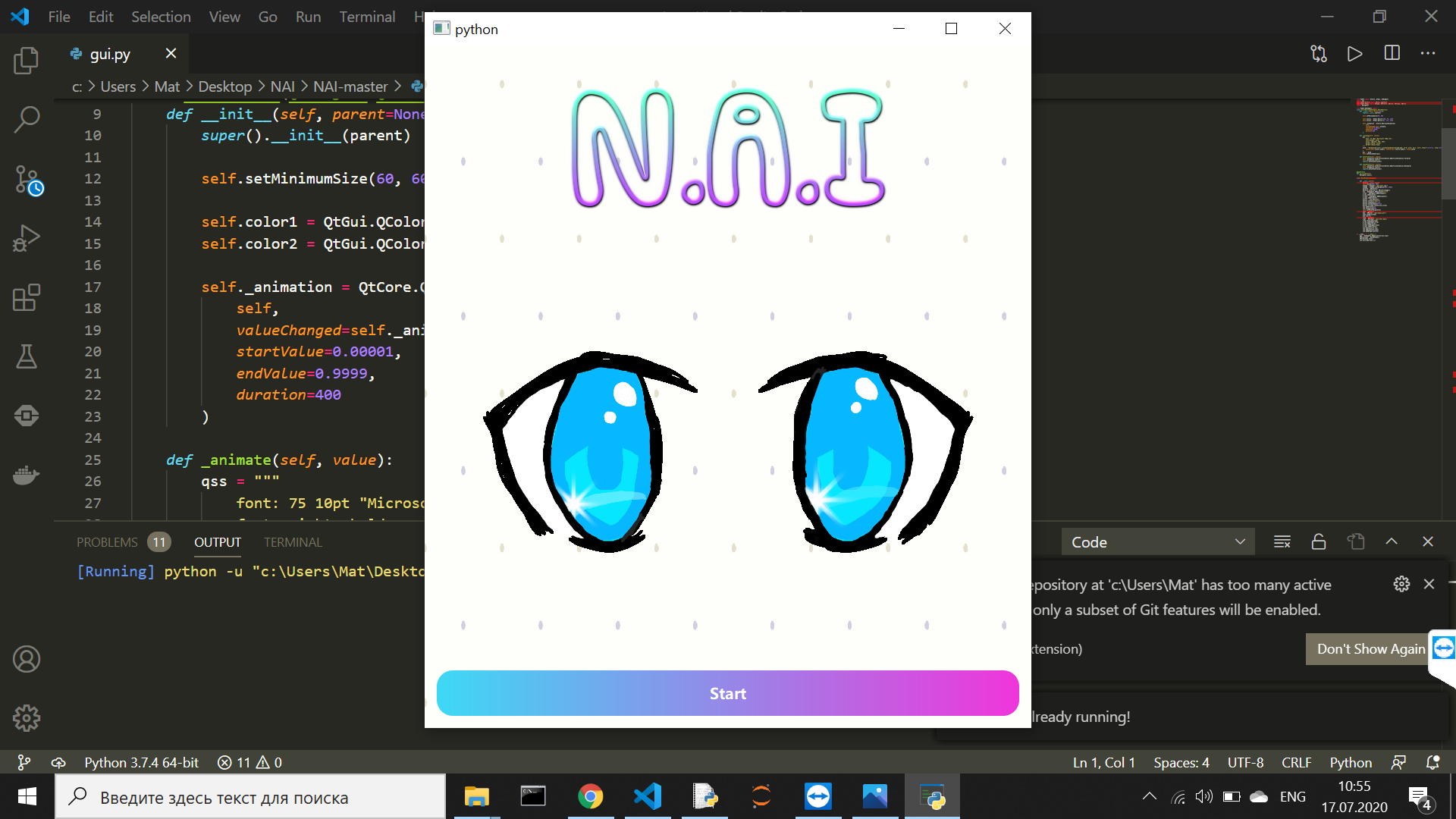
We are also very proud that even the first version of our software is readily available to help disabled people immediately.

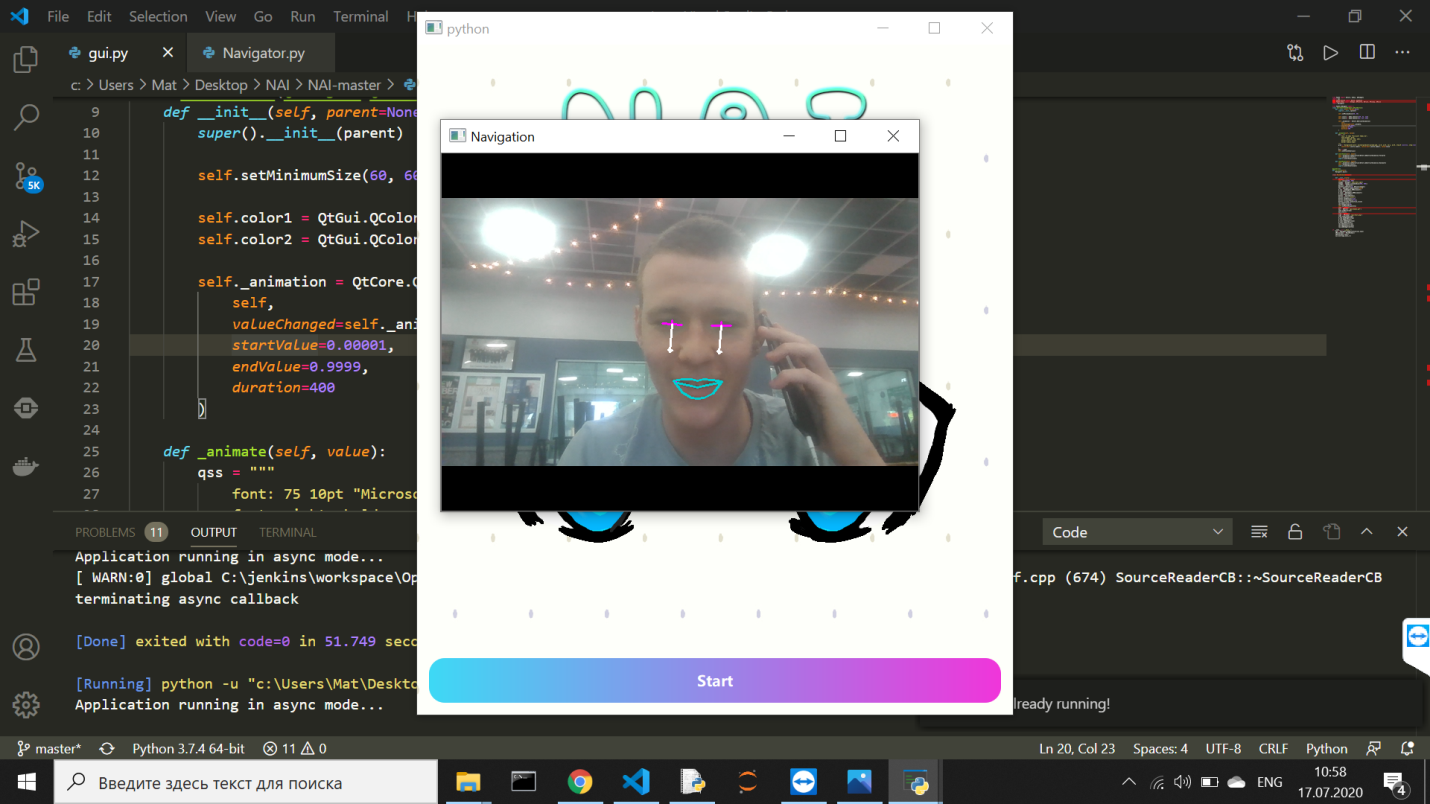
## What we learned

We are also very happy that we gained more experience in working with Deep Learning applications and learned new cutting edge technologies like Docker. We felt all the importance and power of good teamwork.

## What's next for NAI - Artificial Intelligence Desktop Navigation

Making the existing application more robust and adding voice capabilities to our application. We also plan to make the application compatible with all operating systems. We would also have a website enabling people to download our software with just the web soon.





Hi. In this demo I am going to show a DeepFake web app that I built using the streamlit framework.

Deepfakes are synthetic media in which a person in an existing image or video is replaced with someone else's face.

Now, let us have a look at how it works.

Here, I have uploaded 2 files:

One is 02.png which is the Russian president Putins picture and the other file is 00.mp4 which is former US president Obamas video.

Lets click on generate video.

Lets wait until the synthetic video is generated. Deepfakes rely on generative adversarial networks.

Generative Adversarial Networks are a class of Neural Networks in which two neural networks compete against each other to generate results.

Ok so here is the generated video and we can see that we now have a video as if the Russian president was the one speaking. You can also download the generated video. So this is the power of Deepfakes and this has huge potential in the media and entertainment industry.. So that was about it for this video. Hope you liked it.

**Hi. This is our project Vocai, a voice based code editor assistant. It helps you automate typing repetitive and mundane part of the code and lets you focus on the logic of the application your building. It is completely on the web, so there is nothing to be installed or setup.**

**For now, our platform only supports the Python language and hence you can get help related to python code.**

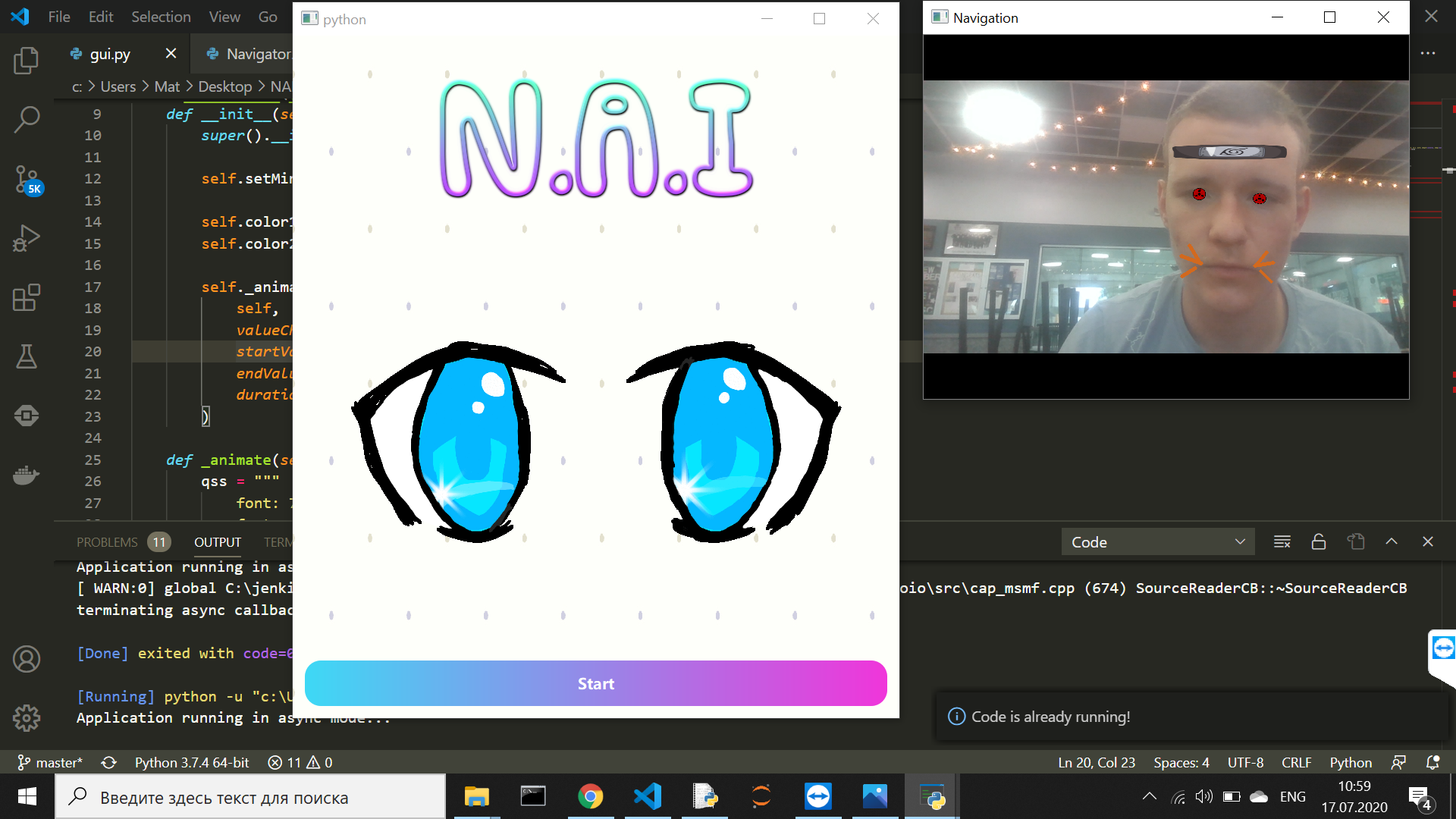
**This web editor is very customizable. You can adjust the font size, tab size and also change the editor theme and key binding mode among many other handy features.**

**The code area your are able to see right now is your work environment and you can save this after you are done editing.**

**To use the voice functionality, click the Start recording button and give your prompt. This prompt will be sent to wit.ai for intent and entity extraction which will again be processed for giving the final output to the user.**

**<recording>**

**You can also clear cache. The future plans for this project includes broadening the support languages and use cases they can handle. Thank you.**

****

Hi. This is Apollo19’s web platform which enables doctors to treat skin cancer patients effectively. It comes with multiple features ranging from AI based skin cancer detection, pixel up scaling to intelligent chat bots and digital prescriptions. Now, let us have a look at the website.

Firstly, we have the Patient Dashboard Page. This is where doctors can retrieve images of the patient’s skin for analysis and later prescribing medicines. After downloading the concerned patient’s image from here, the doctor is expected to move on to the skin cancer detector page.

Here, even if the doctor has minimal knowledge related to skin cancer, he or she can view images of skin cancer to compare against the patient’s image and decide whether it’s actually a symptom of skin cancer or not.

Now, you can use super resolution or in simple words increase image resolution by using our image up scaling feature. This is very helpful to regain pixels lost through image transmission across the internet and can make a huge difference in accurately predicting the outcome. Our model can upscale pixels by a factor of 2.

Then, the doctor can use our Deep learning based classification model to predict among 7 different types of skin cancers. Our model has an accuracy of 96% and has been trained on a lot of images. The prediction can take up to 1 minute since it is a powerful classification model and hence, I have already uploaded the image and retrieved the prediction.

The doctor can then come back to the patient dashboard page and prescribe medication. Important features here include E-signature for ensuring authenticity and verification of the doctor, and digital prescription which holds information related to medicine and dosage will be sent to the corresponding patient.

Let me show you how this works.

Lastly, doctors can interact with the chatbot to solve their doubts. The chatbot uses state of the art intent extraction and entity recognition to give the best answer.

Please note that we have temporarily hosted the website on a personal computer @ .ngrok.io and hence only 1 computer will be able to run it at a time.

So this was a short demo of the web platform and we hope this will help a lot of hospitals by increasing efficiency and decreasing the cost and time taken to treat patients.

Thank you.

Tech stack used:

First let us go with the web platform:

For building the web platform, we have used Python and a web framework called streamlit.

Technologies used by other platforms are:

Wit.ai for entity and intent extraction in chatbot, Tensorflow for training the deep learning model which was trained on publicly available skin Ham dataset. We have used ngrok for temporarilty hosting the website and firebase for connecting the mobile and web apps.

The mobile app was built using Flutter and Firebase and we used tensorflow lite to run inference of the skin cancer prediction model on the mobile.

Now, let us see a demo of the web platform that we built for doctors

**# Apollo19 - AGBI Digital HealthTech Challenge**

**<b><u>Demo Link</u></b>: https://youtu.be/E1D\_kdOuTtc <br>**

**A mobile and web application to help patients and doctors effectively fight against skin cancer.**

**# Steps to run the program:**

**<ol>**

**<li> Clone the repository </li>**

**~~~**

**git clone https://github.com/Sharan-Babu/agbihack.git**

**cd agbihack**

**~~~**

**<li> Install necessary libraries from requirements.txt by running the following command</li>**

**~~~**

**pip install requirements.txt**

**~~~**

**<li> Use the following command in terminal to run the app:</li>**

**~~~**

**streamlit run medical\_hack.py**

**~~~**

**</ol>**

Wit.ai

Hi. In this article, I am going to teach you how to use wit.ai through a fun mini project. By the end of this you will understand the basics of Natural Language Processing, intent recognition and entity extraction.

Ok, so let us consider a real world example to get ourselves excited. How do big companies like Amazon, Flipkart evaluate the customer reviews for a product or automate a considerable amount of their customer care services? This is done with the help of Natural Language Processing (abbreviated as NLP). It is a branch of Artificial Intelligence/ Machine Learning that deals with the interaction between computers and humans through the medium of natural language. This could be text or speech. We ultimately want to derive meaningful insights from the end user’s input.

Now, let us learn about 2 important concepts in NLP which will help us in implementing the project that follow this theory.

First, let us understand what intent recognition is. As the name suggests, we expect the computer to infer the intent from user’s input (text or audio) , understand the purpose of the message and thereby take meaningful actions.

Example of this in everyday life: When you ask Siri or google assistant to set a reminder to wake you up in the morning, it doesn’t call your most frequented contact. This is because it rightly understands what you want to do and hence does the same.

‘’’google definition:

**Intent** classification is the automated association of text to a specific purpose or goal. In essence, a classifier analyzes pieces of text and categorizes them into intents such as Purchase, Downgrade, Unsubscribe, and Demo Request.

‘’’

Now, let us move on to what entity extraction is…. Going with the setting of the above voice assistant example, the term morning in your request “Set a reminder to wake me up in the morning” is very vague.

A more meaningful input to the device would be, “Wake me up every day at 7 am”. Now, your assistant wakes you up daily exactly at 7 am because not only did it recognize the intent of the user but it also extracted the entity. In this case the time ‘7 am’ is the entity that helps your assistant to do the right thing.

Note that people who built this algorithm had to define numbers like this to be extracted as time before hand and trained the AI algorithm accordingly.

‘’’ google definition:

Named-entity recognition is a subtask of information extraction that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions

‘’’

What will we be building?

We will be building a voice enabled code editor assistant with the help of python and wit.ai ( No heavy code, we promise).

Wit.ai is a Facebook owned open source chatbot framework with advanced natural language processing capabilities. The best part being you won’t have to code anything or need knowledge needed to build chatbots. All you have to do is type and click and your chatbot is ready to be served in your app with the help of only a simple API call.

Now, let us build the chatbot from ground up using wit.ai and I will be