VIT - IOT

(INDUSTRY CERTIFICATE INTERNSHIP PROGRAM)

PROJECT

SMARTBRIDGE

Let's Bridge the Gap

WORKMATES

SHARON DAVIS J
(sharondavis.j2020@vitstudent.ac.in)

TANYA
(tanya.2019@vitstudent.ac.in)

GOPINATH K
(gopinath.k2019@vitbhopal.ac.in)

SUDARSHAN U
(u.sudarshan2019@vitbhopal.ac.in)

INTELLIGENT ACCESS CONTROL SYSTEMS FOR SAFETY CRITICAL AREAS IN INDUSTRIES

CONTENT

S.NO	CONTENT	PAGE NO.
1.	INTRODUCTION	5-6
2.	LITERATURE SURVEY	7-8
3.	THEORETICAL ANALYSIS	9-11
4.	FLOWCHART	12
5.	RESULT	13
6.	ADVANTAGES / DISADVANTAGES	14
7.	APPLICATIONS	15
8.	CONCLUSION	16
9.	FUTURE SCOPE	16
10	BIBLIOGRAPHY	17
11.	APPENDIX	18-26

INTRODUCTION

OVERVIEW:

The access control system is to provide quick, convenient access to those persons who are authorized, while at the same time, restricting access to unauthorized people. Attacks on civil and institutional objects are becoming a potential threat in several parts of the world.

The target might be a bank or a company, and the attack might be motivated by money or ideological reasons, but the essential pattern is the same. Due to the increase of these types of attacks, it is important that advanced scientific and technological solutions are applied to real-life applications. One of the important security tasks is to assure efficient access control.

PURPOSE:

The purpose of access control is to grant entrance to a building or office only to those who are authorized to be there. In some industries it is necessary for workers to wear safety helmets while working. So, to check weather workers are taking safety precautions or not we are proposing this system.

We can train our classifier to identify helmet with Clarify API. There will be video streaming near the entry of the industries where we can first detect the face of a person and if any person is present then we can capture the image of that moment and send it to Clarifai API to detect whether the person is wearing helmet or not.

If the person is wearing a helmet, we can give him access by opening the door. If he is not wearing then we can restrict his access by not opening the door. We can even warn him through voice commands to take safety precautions.

LITERATURE SURVEY

EXISTING PROBLEM:

The goal of access control is to minimize the security risk of unauthorized access to physical and logical systems. Access control is a fundamental component of security compliance programs that ensures security technology and access control policies are in place to protect confidential information, such as customer data.

Access control systems are complex and can be challenging to manage in dynamic IT environments that involve on-premises systems and cloud services. After some high-profile breaches, technology vendors have shifted away from single sign-on (SSO) systems to unified access management, which offers access controls for on-premises and cloud environments.

PROPOSED SOLUTION:

Access control is the act of restricting access to a selected group of people or systems. That group is authorized to access the system. To check if a person is authorized to access, the person typically has to be authenticated.

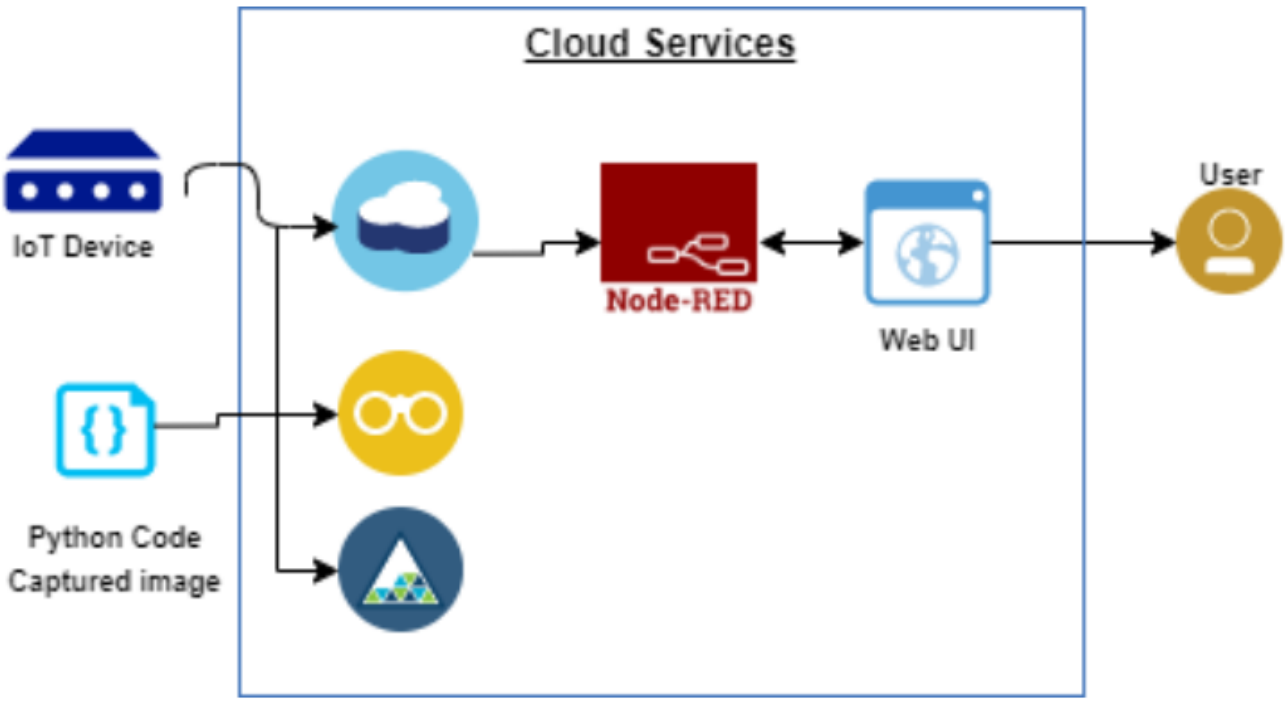
It is important because it reduces the risk of unauthorized access to computer systems and physical areas – thus, it is the foundation of data, network, and information security. Access control is a compliance requirement for some organizations.

There are still challenges associated with access control – these are mainly due to modern IT's highly-distributed nature. It can be tricky to keep track of moving and evolving assets when they are spread out. One example includes password fatigue – this is when a user struggles to remember many passwords that are part of their daily routine. This is why access systems that are password less are growing popular.

Access control systems are an essential commodity for virtually every industry. Here are some of the most commonly found applications of access control systems.

THEORETICAL ANALYSIS

BLOCK DIAGRAM:



SOFTWARE DESIGNING:

PYTHON IDLE:

- IDLE is integrated development environment (IDE) for editing and running Python 3.9.6
- The IDLE GUI (graphical user interface) is automatically installed with the Python interpreter. IDLE was designed specifically for use with Python.
- IDLE has a number of features to help you develop your Python programs including powerful syntax highlighting.

CLARIFAI API:

Clarifai Inc. is an artificial intelligence (AI) company that specializes in computer vision and uses machine learning and deep neural networks to identify and analyze images and videos. The company offers its solution via API, mobile SDK, and on-premise solutions. Clarifai is headquartered in New York City with two satellite offices in San Francisco and Washington D.C. Clarifai API produces an accuracy score for each image and classifies them under the appropriate class.

NODE RED:

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click. JavaScript functions can be created within the editor using a rich text editor. A built-in library allows you to save useful functions, templates or flows for re-use.

IBM CLOUD:

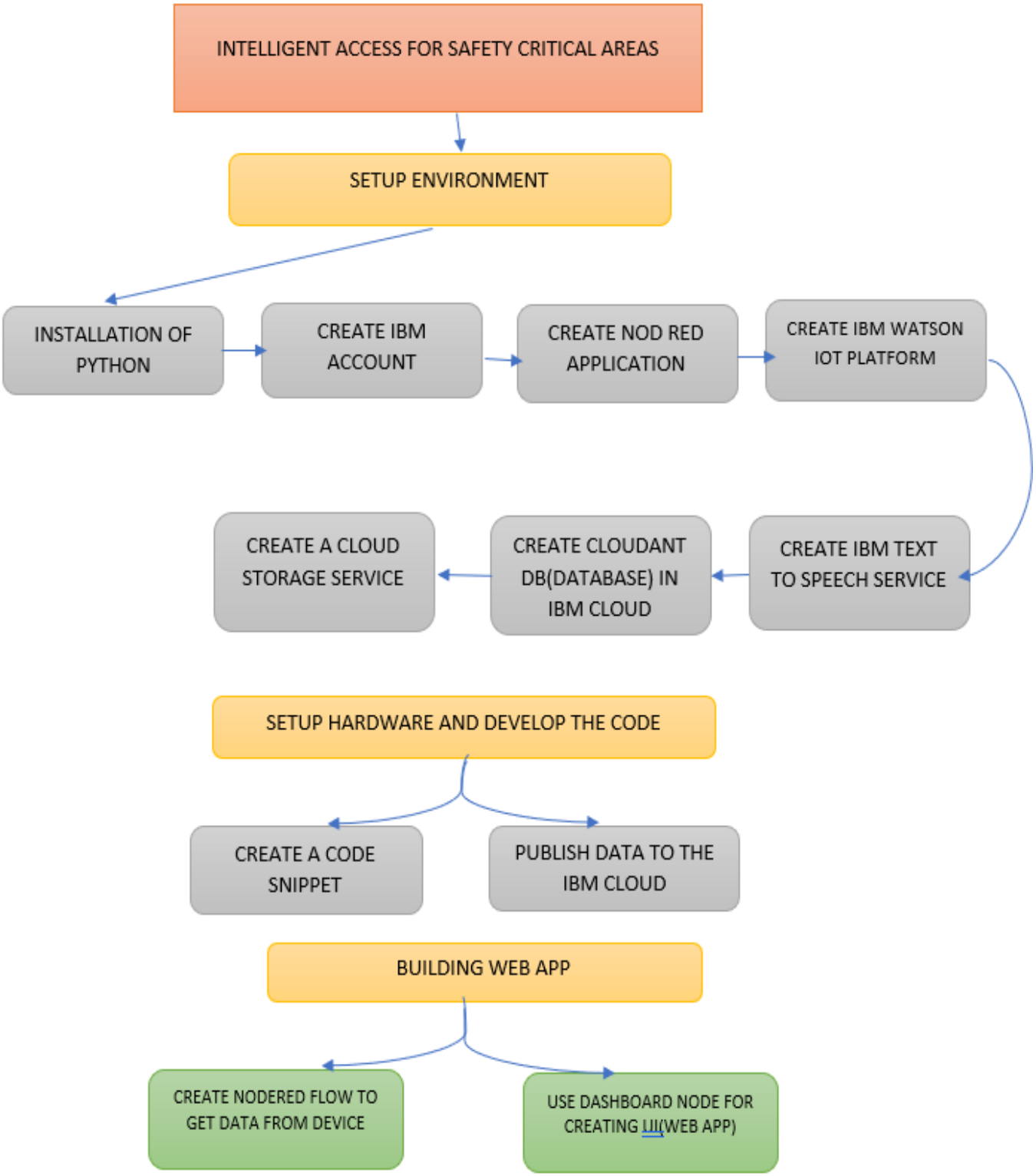
IBM Cloud is a suite of cloud computing services from IBM that offers both platform as a service (PaaS) and infrastructure as a service (IaaS). With IBM Cloud IaaS, organizations can deploy and access virtualized IT resources such as compute power, storage and networking over the internet. It can be used to build scalable infrastructure at a lower cost, deploy new applications instantly and scale up workloads based on demand all within a security rich platform.

IBM TEXT TO SPEECH:

Text-to-speech (TTS) is a type of assistive technology that reads digital text aloud. It's sometimes called "read aloud" technology.With a click of a button or the touch of a finger, TTS can take words on a computer or other digital device and convert them into audio. TTS is very helpful for kids and adults who struggle with reading. But it can also help with writing and editing, and even with focusing.

TTS works with nearly every personal digital device, including computers, smartphones, and tablets. All kinds of text files can be read aloud, including Word and Pages documents. Even online web pages can be read aloud.

FLOWCHART:


INTELLIGENT ACCESS FOR SAFETY CRITICAL AREAS
SETUP ENVIRONMENT
INSTALLATION OF PYTHON
CREATE IBM ACCOUNT
CREATE NOD RED APPLICATION
CREATE IBM WATSON IOT PLATFORM
CREATE IBM TEXT TO SPEECH SERVICE
CREATE CLOUDANT DB(DATABASE) IN IBM CLOUD
CREATE A CLOUD STORAGE SERVICE
SETUP HARDWARE AND DEVELOP THE CODE
CREATE A CODE SNIPPET
PUBLISH DATA TO THE IBM CLOUD
BUILDING WEB APP
CREATE NODERED FLOW TO GET DATA FROM DEVICE
USE DASHBOARD NODE FOR CREATING UI(WEB APP)

RESULT:

There are many types of access control software and technology, and often, multiple components are used together to maintain access control. The software tools may be on premises, in the cloud or a hybrid of both. They may focus primarily on a company's internal access management or may focus outwardly on access management for customers. Some of the types of access management software tools include the following:

- Reporting and monitoring applications
- Password management tools
- Provisioning tools
- Identity repositories
- Security policy enforcement tools

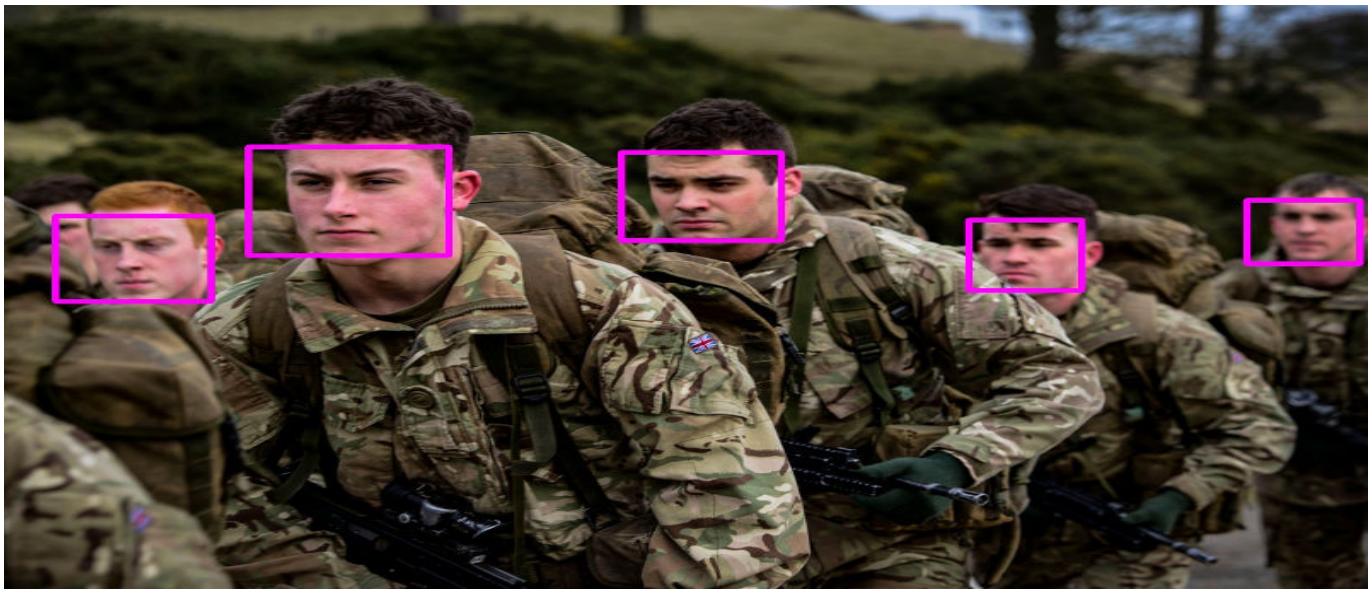
Access control systems are complex and can be challenging to manage in dynamic IT environments that involve on-premises systems and cloud services.

ADVANTAGES:

1. Increase ease of access for employees
2. Get rid of traditional keys
3. Save money and energy
4. Keep track of who comes and goes
5. Protect against unwanted visitors
6. Give employees the freedom to work when they need to
7. Prevent against data breaches
8. Create a safe work environment
9. Reduce theft and accidents
10. Provide access to multiple buildings and locations

DISADVANTAGES:

1. Difficult to know the access right of a given subject.
2. Difficult to revoke a user's right on all objects.
3. Difficult to know who can access a given object.
4. Difficult to revoke all access right to an object.

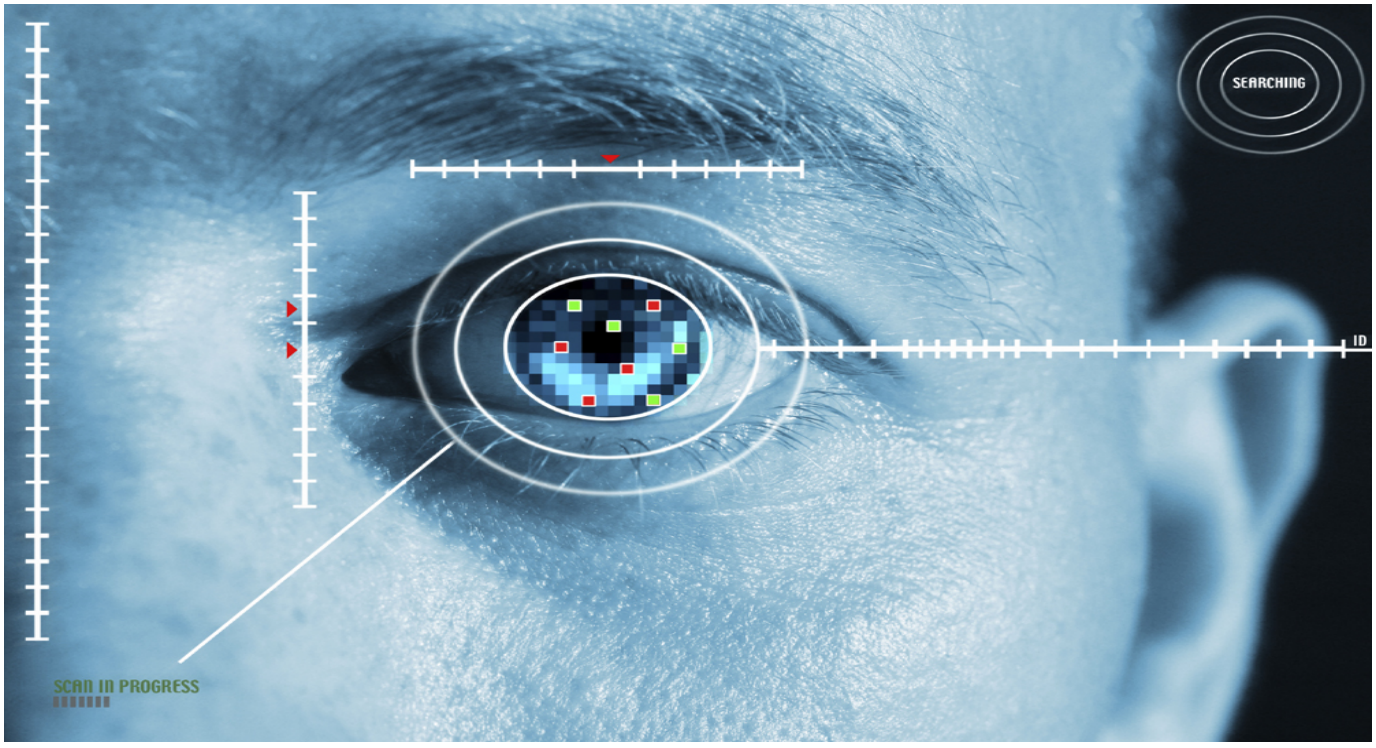


APPLICATIONS:

An access control system is a set of specialized equipment and software designed to organize, on a given territory, a system for restricting, registering and controlling the access of people and vehicles through “access points” (doors, gates, checkpoints, etc.).

- Protection of material values
- Information
- Property
- Equipment

Security of employees and visitors control, accounting and access control to the object (identification of the person, identification of zones and access times, an opening of doors, turnstiles, barriers, etc.)



CONCLUSION:

The need for excellent security has never been greater. Physical and virtual threats are ever-evolving, thus demanding advanced technology, in-depth analytics, and stringent safety measures. Keys and simple passwords no longer cut it. The right access control system can help you secure physical and informational assets, cut personnel costs, and keep your staff and employees safe.

Whether you have a small company or a global enterprise, a reliable and reputable access control system can help you meet security challenges head-on.

FUTURE SCOPE:

A continued trend towards system integration and increasing levels of interoperability will make it easier to monitor and control access always.

Video analytics cameras can be used to create a system that allows only one person at a time to access a secured area. This virtually eliminates the risk of unauthorized personnel sneaking into an area by tailing someone else. Biometric scanners will become more affordable and widely adopted.

These improvements, as well as other technologies that will undoubtedly be developed, will ensure that access control continues to provide strong security while offering value for its investment.

BIBLIOGRAPHY

- A Guide to the Internet of Things Infographic (from INTEL):

http://www.intel.com/content/www/us/en/internetof-things/infographics/guide-to-iot.html

- Benner, Thorsten and Mirko Hohmann, “The Encryption Debate we Need,” Global Public Policy Institute (May 19, 2016), at

http://www.gppi.net/publications/global-internet-politics/article/the-encryption-debate-weneed/

- Perlroth, Nicole, “Defense Secretary Takes Position against a Data ‘Back Door’,” The New York Times (March 2, 2016), at

http://www.nytimes.com/2016/03/03/technology/defense-secretary-takes-position-against-a-databack-door.html?_r=1

APPENDIX

SOURCE CODE:

```
#import pyttsx3
import cv2
import datetime
import time
#import playsound
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
import pygame
stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())
from clarifai_grpc.grpc.api import service_pb2, resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2
import ibm_boto3
from ibm_botocore.client import Config, ClientError
from ibmcloudant.cloudant_v1 import CloudantV1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator
# Constants for IBM COS values
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud" #
Current list avaiable at
https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints
COS_API_KEY_ID =
"QK-6lL2JX1TK7yjNZJCi0GEcrUbVlKbbZJIah_OyVx-A" # eg
"W00YixxxxxxxxxxMB-odB-2ySfTrFBIQQWanc--P3byk"
COS_INSTANCE_CRN =
"crn:v1:bluemix:public:iam-identity::a/e296b12a0dfa4ddfbe54f3d5eca9badb::s
```

```
erviceid:ServiceId-aa9e5973-a4e1-405f-94e3-7c976e56a6d1" # eg
"crn:v1:bluemix:public:cloud-object-storage:global:a/3bf0d9003xxxxxxxxxx1c
3e97696b71c:d6f04d83-6c4f-4a62-a165-696756d63903::"
# Create resource
cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)
authenticator =
BasicAuthenticator('apikey-v2-qy5qcpihygwozrugn8t18q4j8owzr269lxi4rkjmb
c1', '220fd8d9797d68fd912dd4d7bd1ecdd0')
service = CloudantV1(authenticator=authenticator)
service.set_service_url('https://apikey-v2-qy5qcpihygwozrugn8t18q4j8owzr269
lxi4rkjmbc1:220fd8d9797d68fd912dd4d7bd1ecdd0@341e9043-5fc4-47e8-8d6
5-ae1a1c302533-bluemix.cloudantnosqldb.appdomain.cloud')
bucket = "sharonhelmet"
def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name,
bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5
        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15
        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
```

```
        )
        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )
        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))
# This is how you authenticate.
metadata = (('authorization', 'Key 62ab2f4fa9bf46f487fa50bdf63deaf2'),)
#face_classifier=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
#eye_classifier=cv2.CascadeClassifier("haarcascade_eye.xml")
#It will read the first frame/image of the video
video=cv2.VideoCapture(0)
while True:
    check,frame=video.read()
    gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow('Face detection', frame)
    picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
    cv2.imwrite("helmet777.jpg",frame)
    with
open(r'C:\Users\Sharon\Desktop\SmartInternz-IoT\VIT-cv2\Haar_cascade\helm
et777.jpg', "rb") as f:
        file_bytes = f.read()
    request = service_pb2.PostModelOutputsRequest(
```

```
        # This is the model ID of a publicly available General model. You may
use any other public or custom model ID.
        model_id='aaa03c23b3724a16a56b629203edc62c',
        inputs=[
resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Image(ba
se64=file_bytes)))
        ])
    response = stub.PostModelOutputs(request, metadata=metadata)
    if response.status.code != status_code_pb2.SUCCESS:
        raise Exception("Request failed, status code: " + str(response.status.code))
    a= []
    for concept in response.outputs[0].data.concepts:
        if(concept.value > 0.8):
            a.append(concept.name)
    #print(a)
    t=1
    for i in a:
        if(i == "person" or i == "people"):
            #print("Person is detected")
            for j in a:
                if j == "helmet":
                    print("Person is wearing the helmet and you are allowed into the
industry")
                    #engine.say("Person is wearing the helmet and you are allowed
into the industry")
                    #engine.runAndWait()
                    t=1
                    break
                else:
                    t=0
```

```
            if(t==0):
                print("Person is not wearing the helmet")
                print('Playing..')
                pygame.mixer.init()
                pygame.mixer.music.load('new1.mp3')
                pygame.mixer.music.play()
                while not(pygame.mixer.music.get_busy()):
                    pygame.mixer.quit()
                picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
                cv2.imwrite(picname+".jpg",frame)
                multi_part_upload('sharonhelmet', picname+'.jpg', picname+'.jpg')

json_document={"link":COS_ENDPOINT+'/'+bucket+'/'+picname+'.jpg'}
                response = service.post_document(db="helmet1",
document=json_document).get_result()
            elif(t==1):
                break
        #drawing rectangle boundries for the detected eyes
        #for(ex,ey,ew,eh) in eyes:
            #cv2.rectangle(frame, (ex,ey), (ex+ew,ey+eh), (127,0,255), 2)
            #cv2.imshow('Face detection', frame)
        #waitKey(1)- for every 1 millisecond new frame will be captured
        Key=cv2.waitKey(1000)
        if Key==ord('q'):
            #release the camera
            video.release()
            #destroy all windows
            cv2.destroyAllWindows()
            break
```

UI OUTPUT SCREENSHOT:

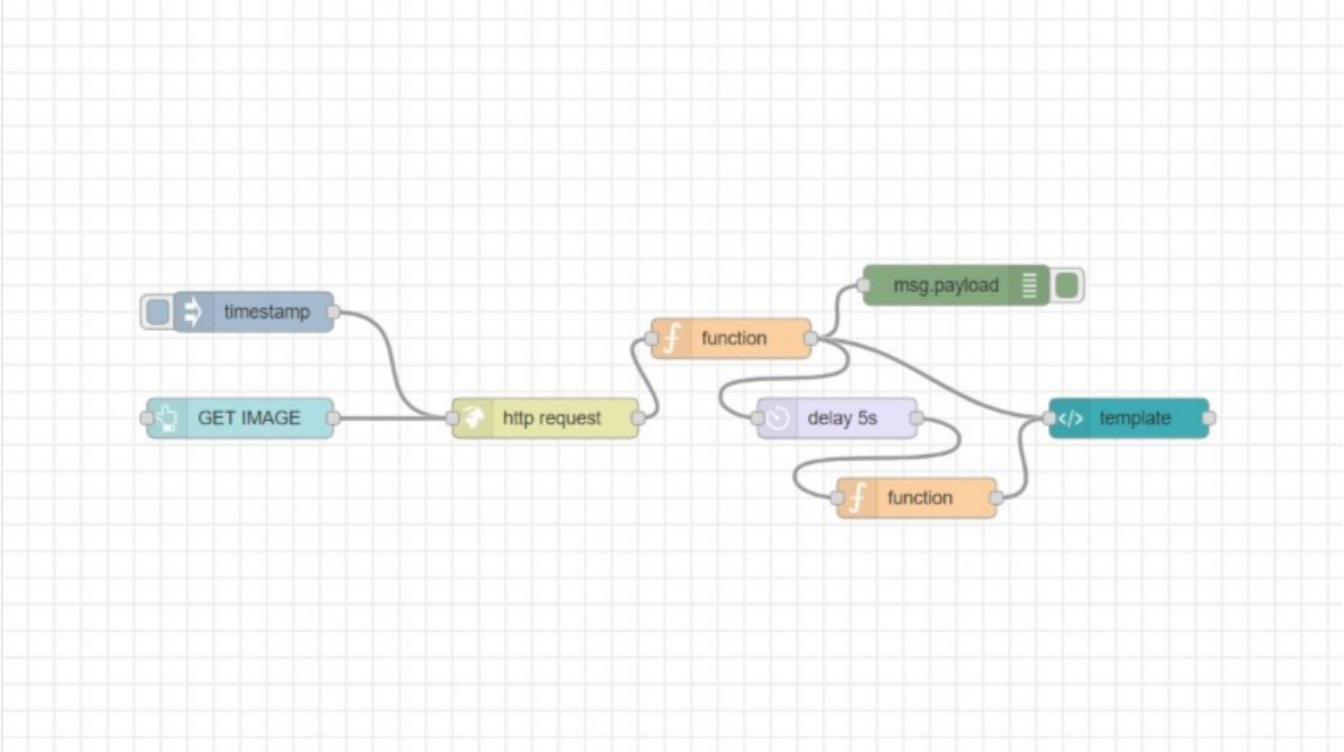


FIG 1: NODE RED DIAGRAM

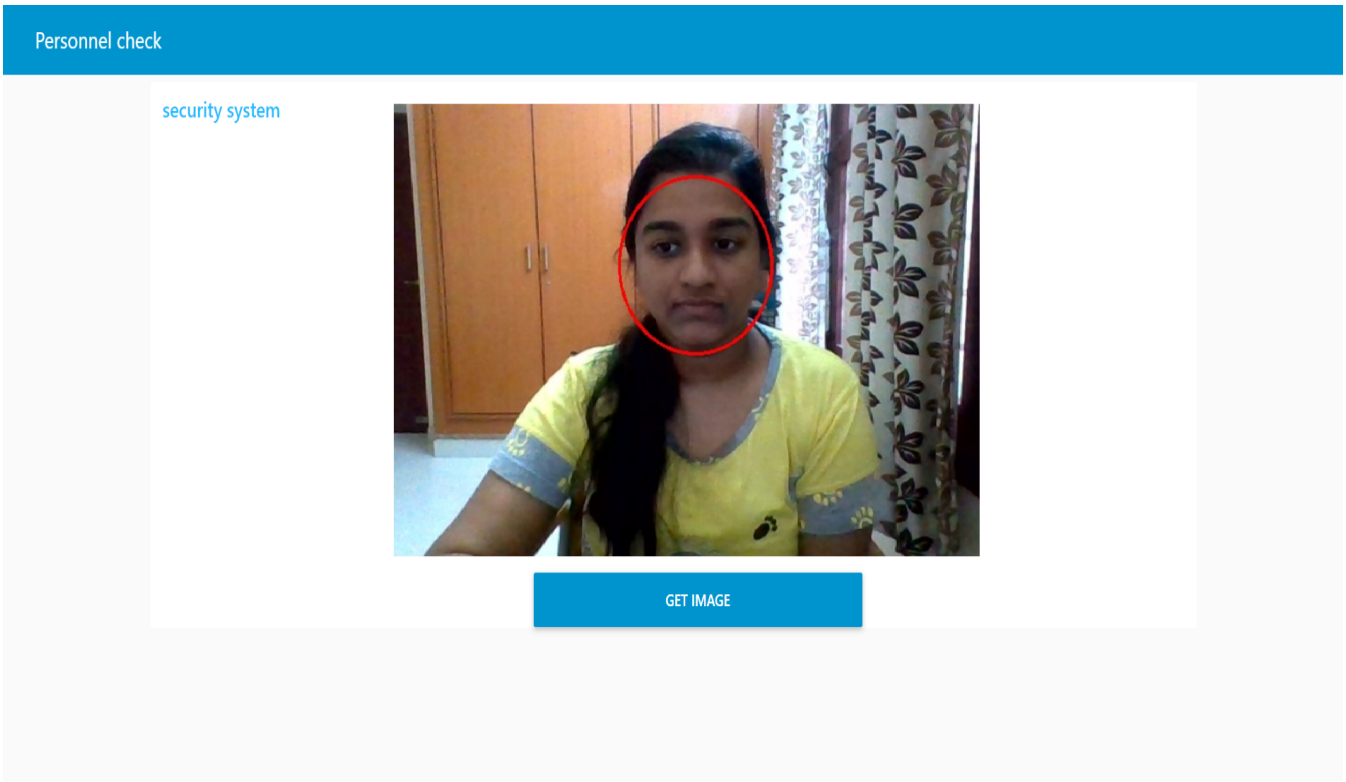


FIG 2: HELMET DETECTION IMAGE

FIG 3: WEB UI IMAGE

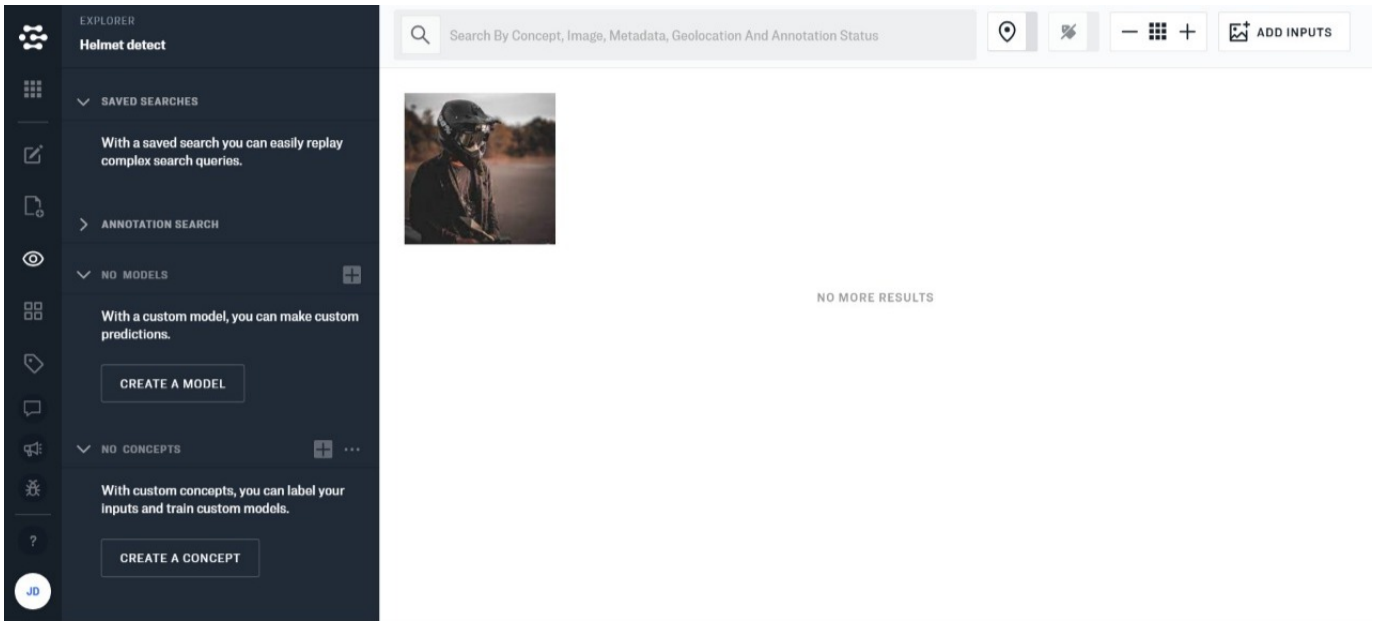


FIG 4: CLARIFAI API

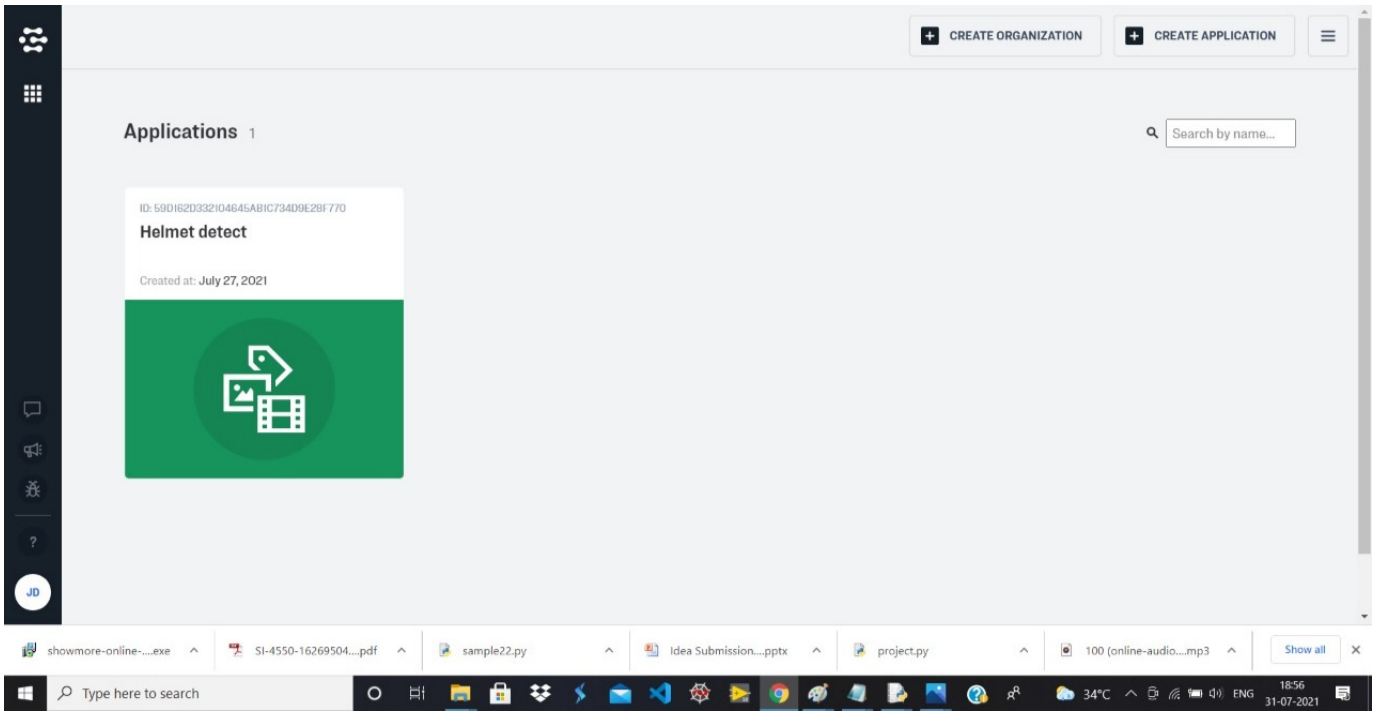


FIG 5: CLARIFAI API DASHBOARD

------X--------------X-------