**PROJECT REPORT**

Emerging Methods for Early Detection of

Forest Fire

Team id:PNT2022TMID27896

|  |  |
| --- | --- |
| Team leader | Jaisri.S |
| Team Member | Anukeerthana.G |
| Team Member | Gowtham.V |
| Team Member | Gopinathan.M |

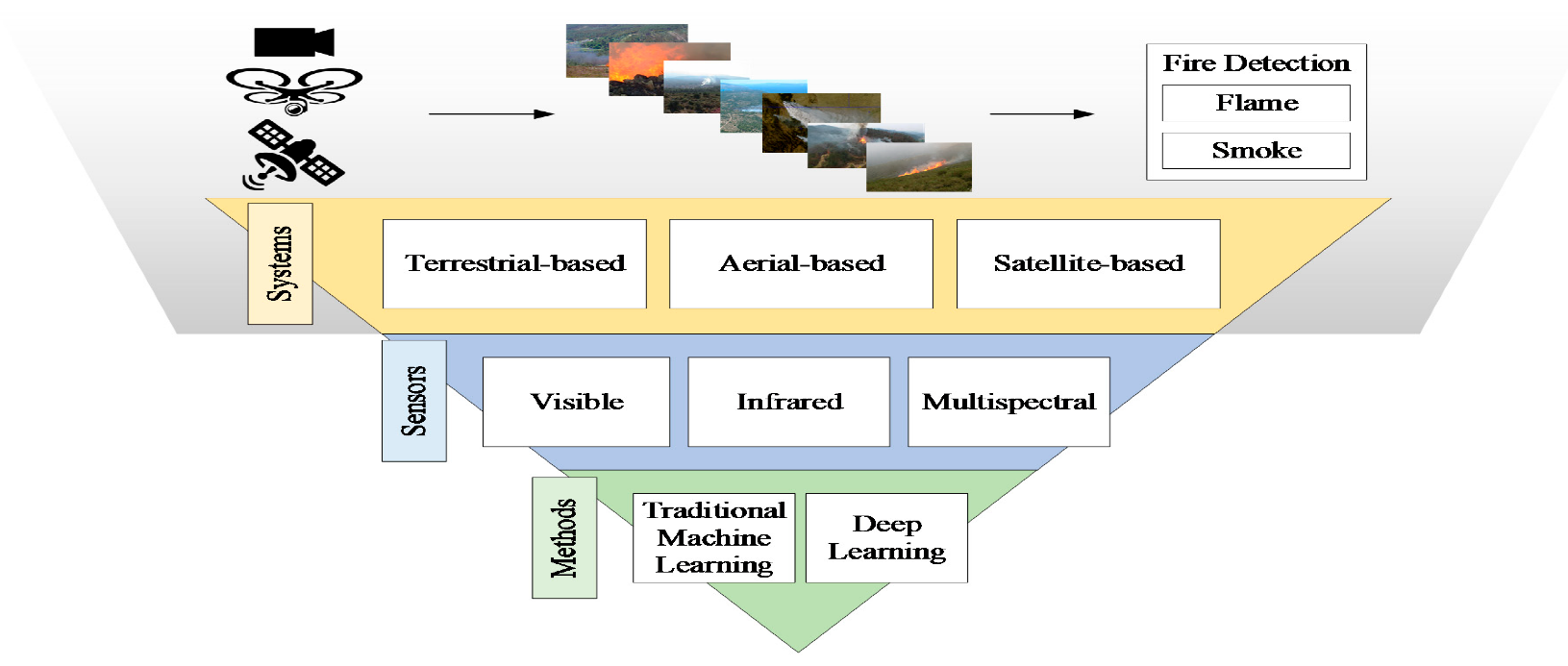
**CHAPTER** 1

**INTRODUCTION**

**1.1PROJECT OVERVIEW**

 Forests are the largest [terrestrial ecosystem](https://www.sciencedirect.com/topics/earth-and-planetary-sciences/terrestrial-ecosystem) in the world and are an important part of the global ecosystem.  Forest fires are a key factor in breaking the ecological balance of forests .One of the major challenges is providing reliable information for making decisions to manage wildfire events.

An artificial neural network framework is used to generate an accurate flame propagation map. For this purpose, live data can be collected like video image processing, sensor networks, image processing, Convolutional Neural Network,  OpenCV, convolutional Neural Networks for Computer vision AI. After repeated training and testing, the forest fire prediction results based on the artificial neural network is successfully done.



**1.2PURPOSE**

The Algorithm depends on previous weather conditions in order to predict the fire hazard level of a day. The implementation of the algorithm using data from Lebanon demonstrated its ability to accurately predict the hazard of fire occurrence.

It is difficult to predict and detect Forest Fire in a sparsely populated forest area and it is more difficult if the prediction is done using ground-based methods like Camera or Video-Based approach. Satellites can be an important source of data prior to and also during the Fire due to its reliability and efficiency. The various real-time forest fire detection and prediction approaches, with the goal of informing the local fire authorities

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 EXISTING PROBLEM**

Forest ﬁres have been and still are serious problem for theEuropean Union and for all other countries in Europe. In the year 2000, the EU has established the European Forest Fire Information system (EFFIS) , which will soon become part of the European Emergency Management Service, maintainedby the Copernicus Earth Observation Programme . This system provides valuable near real-time and also historical data on the forest ﬁres in Europe, the Middle East and North Africa. According to the annual report of EFFIS for 2016 , more than 54 000 forest ﬁres have occurred all around Europe and they have led to nearly 376 thousand hectares of burnt areas.If we compare these values to the average values from the EFFIS reports for the period 2006-2015, the number of forest ﬁres have decreased by 13327 or by nearly 20%. This decrease can be explained with the more severe actions and sanctions towards the arsonists and with the introduction of more advanced technical solutions for early detection of the ﬁre.Conﬁrmation for this are the devastating forest ﬁres form 2018, which took place in the Attica region of Greece and led to more than 90 fatalities and to more than 200 injured people,as well as to the destruction to thousands of buildings .

Forest Fires can be divided into 4 categories in the forests of Hungary based on tree and other vegetation species:

• underground burning, peat ﬁre

•ﬁre in undergrowth or dead fallen leaves;

• ﬁre in seedlings and saplings;

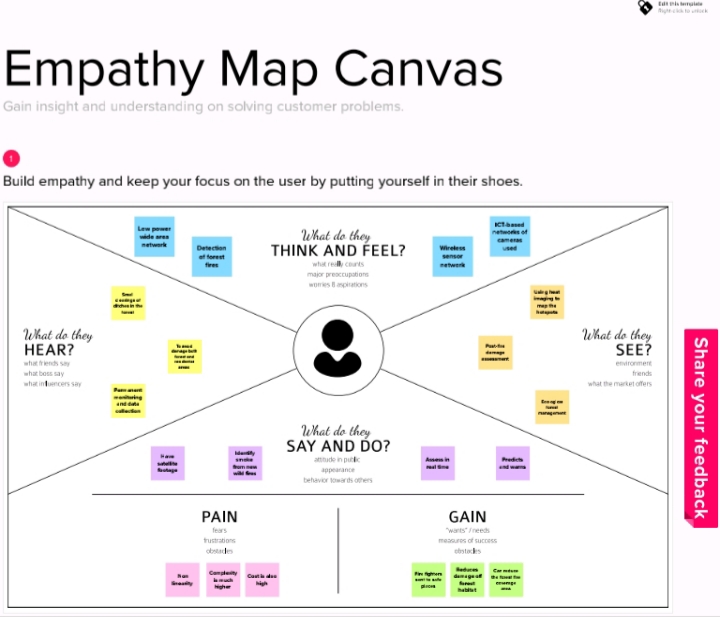
• ﬁre in trunks and shrouds

* 1. **REFERENCES**

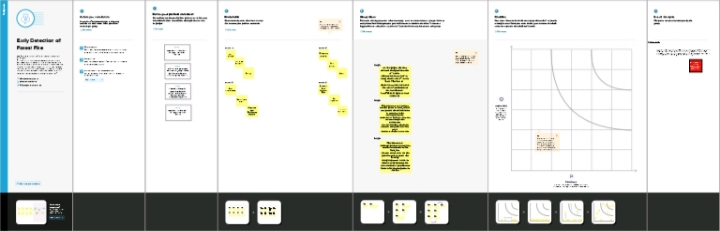
1. Georgi Hristov; Jordan Raychev; Diyana Kinaneva; Plamen Zahariev , Published in: 2018 28th EAEEIE Annual Conference (EAEEIE).
2. Chi Yuan, Youmin Zhang, and Zhixiang Liu , Published in: 2015, Published by NRC Research Press.
3. Mohamed Hefeeda and Majid Bagheri , Published in: June 26, 2008.
4. PRIYADARSHINI M HANAMARADDI , Published in: January 2016.
5. Dr. Panagiotis Barmpoutis, Periklis Papaioannou, Dr. Kosmas Dimitropoulos, Dr. Nikos GRAMMALIDIS , Published in: 11 November 2020.
6. Vinay Chowdary , Mukul Kumar Gupta , Rajesh Singh, Published in:2018
7. European Forest Fire Information System <http://effis.jrc.ec.europa.eu>
8. Official webpage of Movidius at:https://www.movidius.com

**CHAPTER 3**

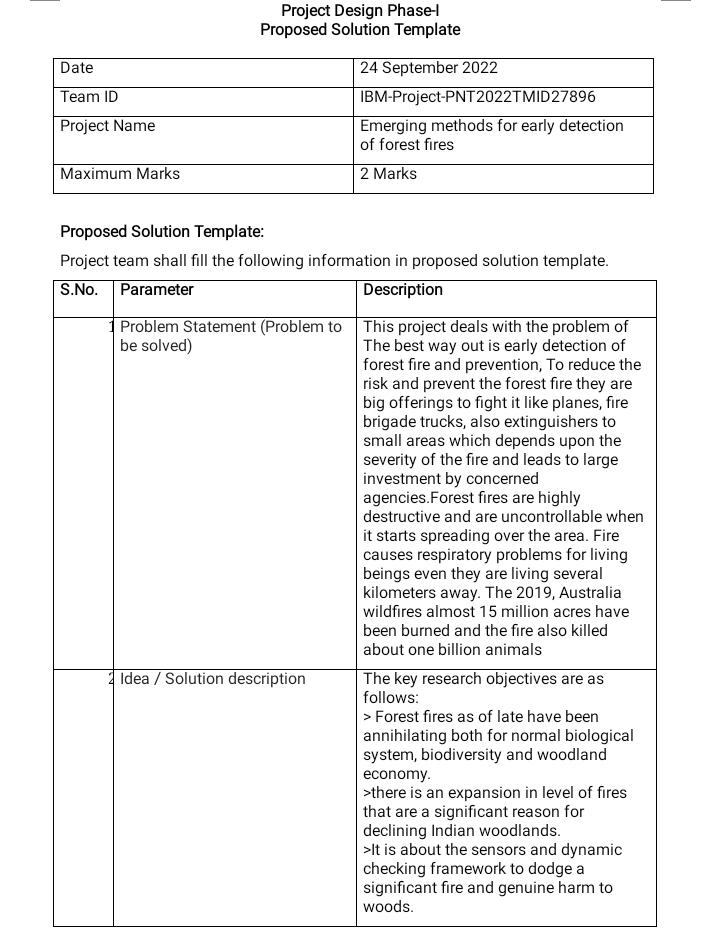
**IDEATION AND PROPOSED SOLUTION**

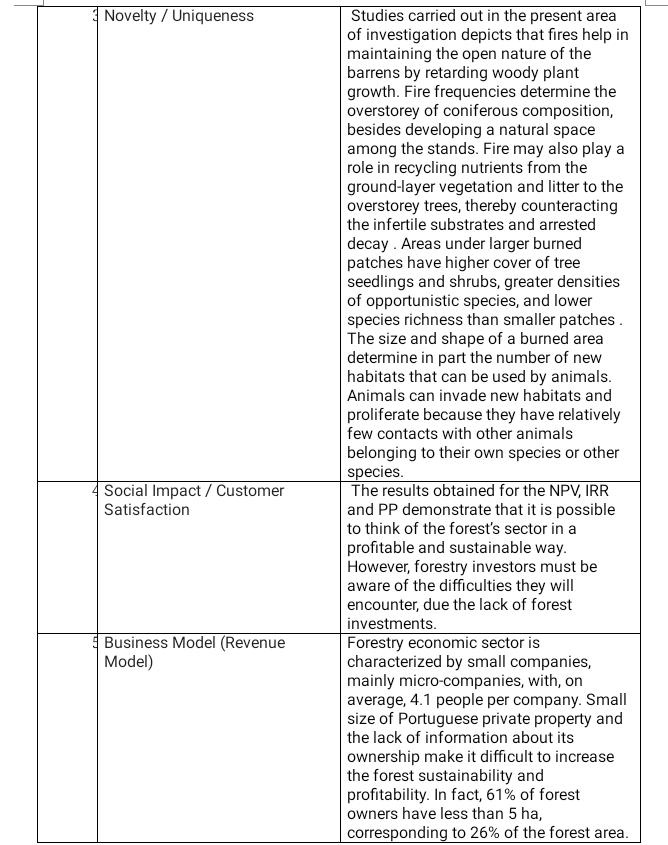
**3.1 EMPATHY MAP CANVAS**

**3.2 BRAINSTORMING**

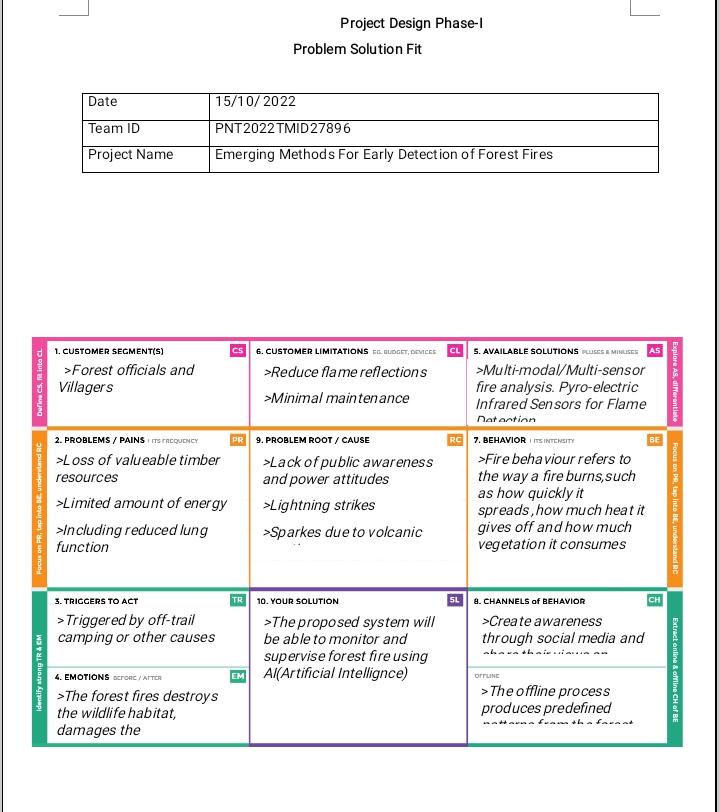


**PROJECT DESIGN PHASE-1**

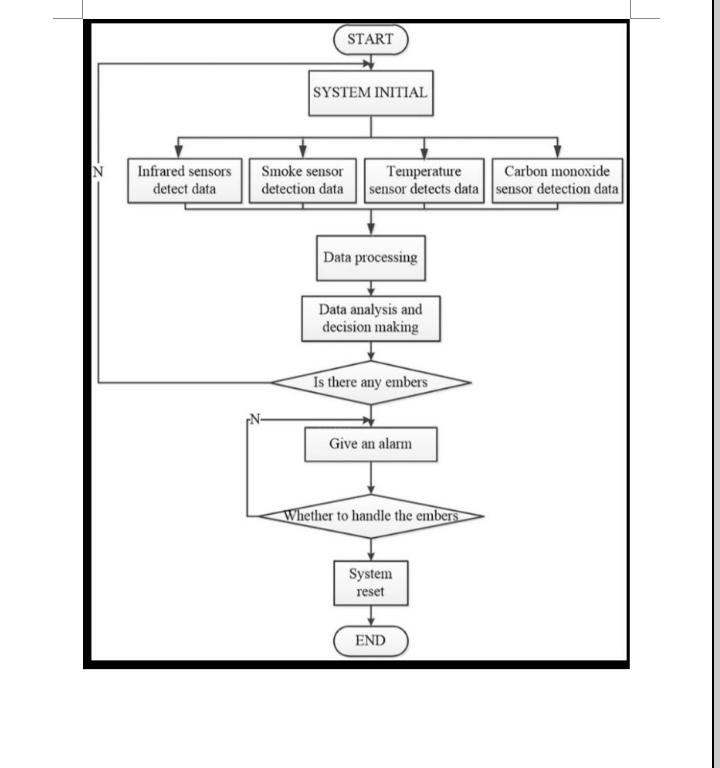
**3.2 PROPOSED SOLUTION**



**3.3 PROBLEM SOLUTION FIT**



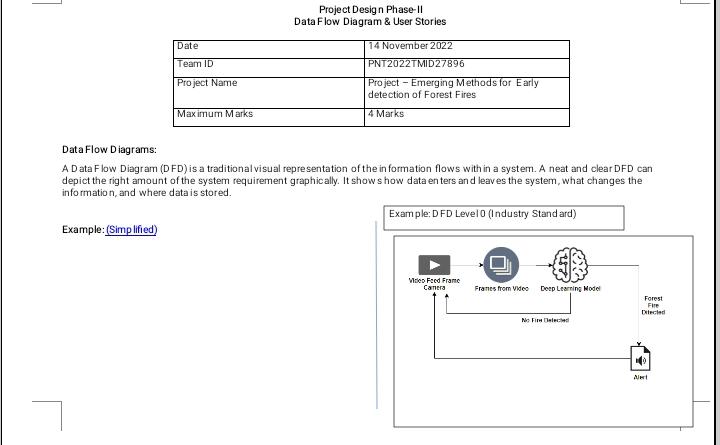
**3.4 SOLUTION ARCHITECTURE**

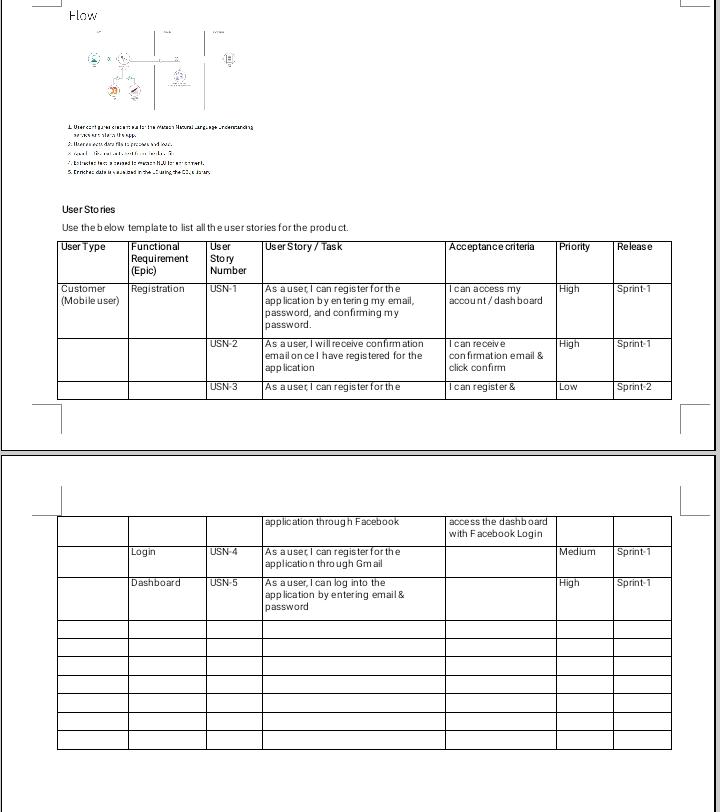
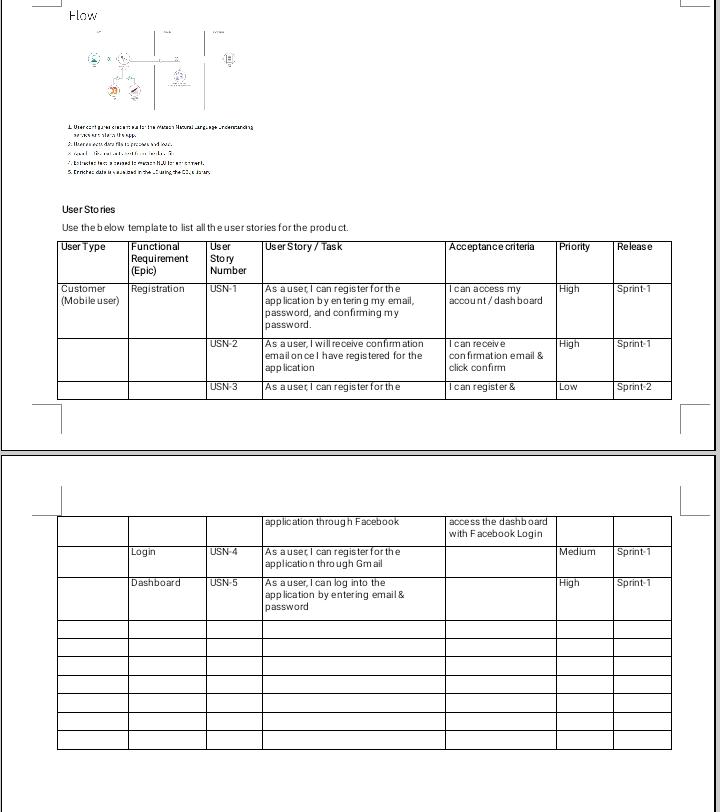


**CHAPTER-4**

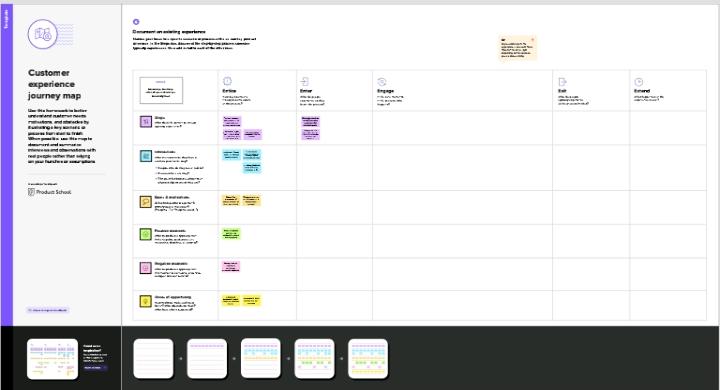
**PROJECT DESIGN PHASE-2**

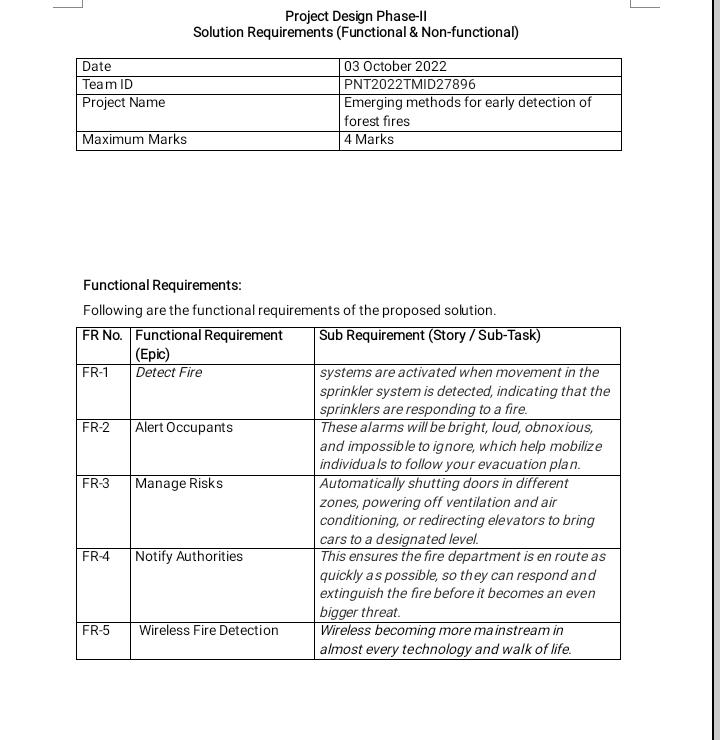
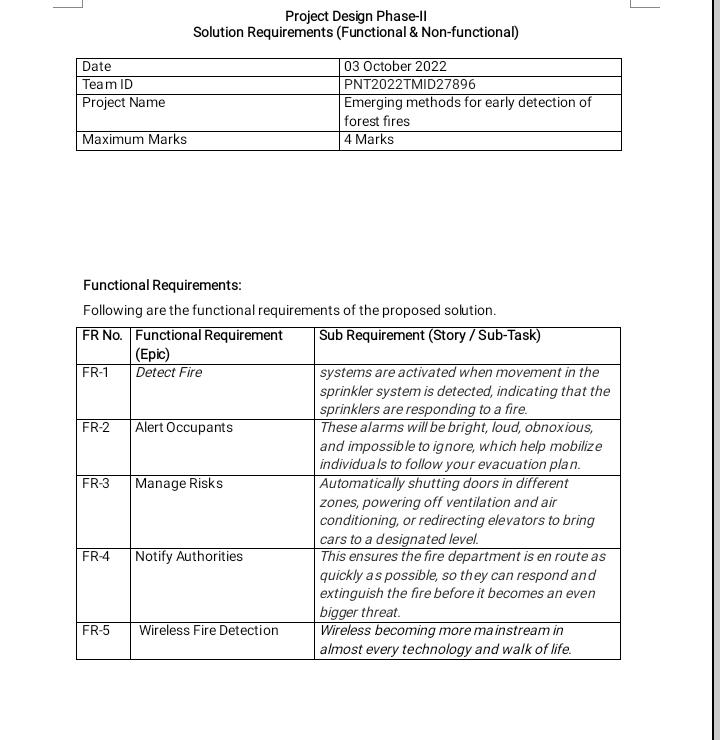
**4.1 DATA FLOW**

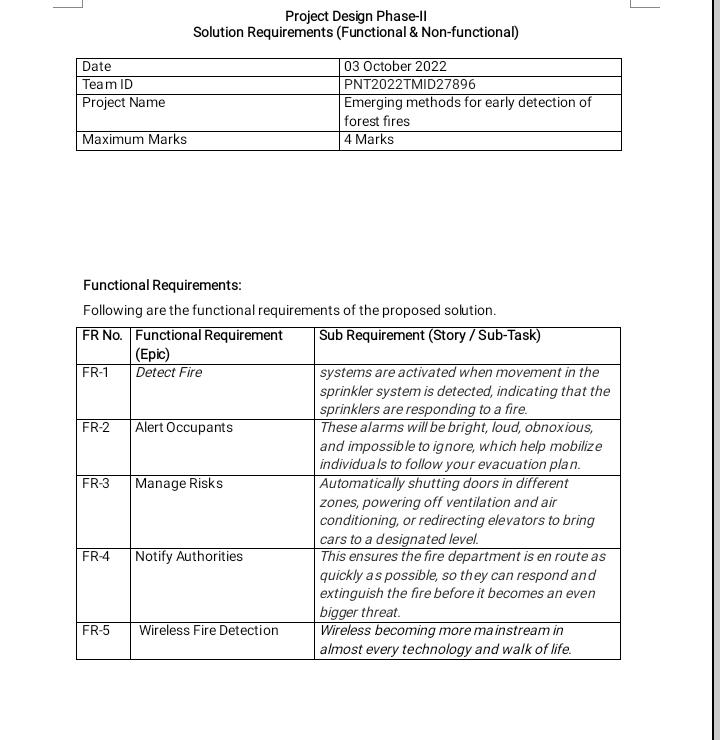


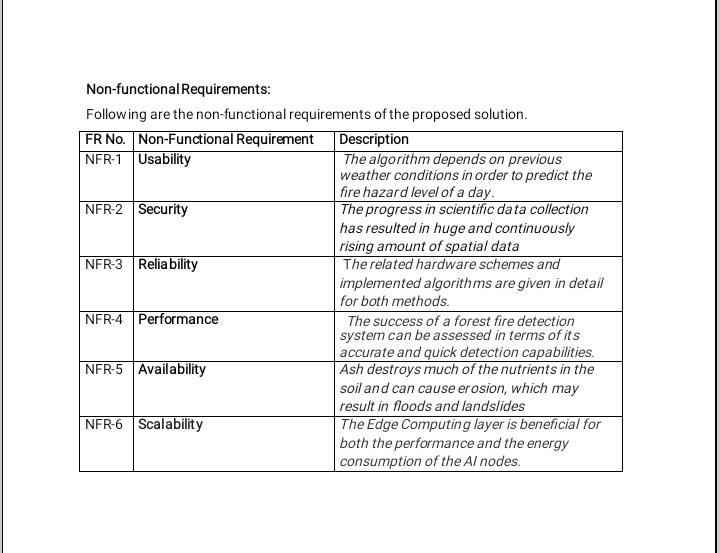
 

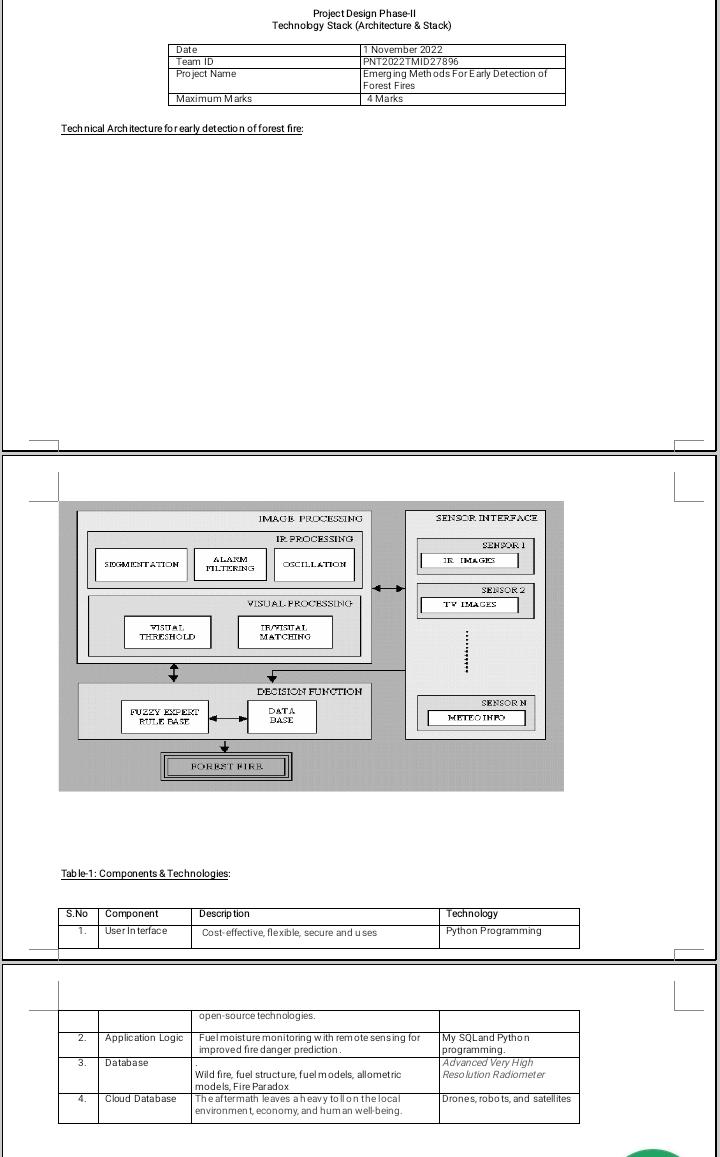
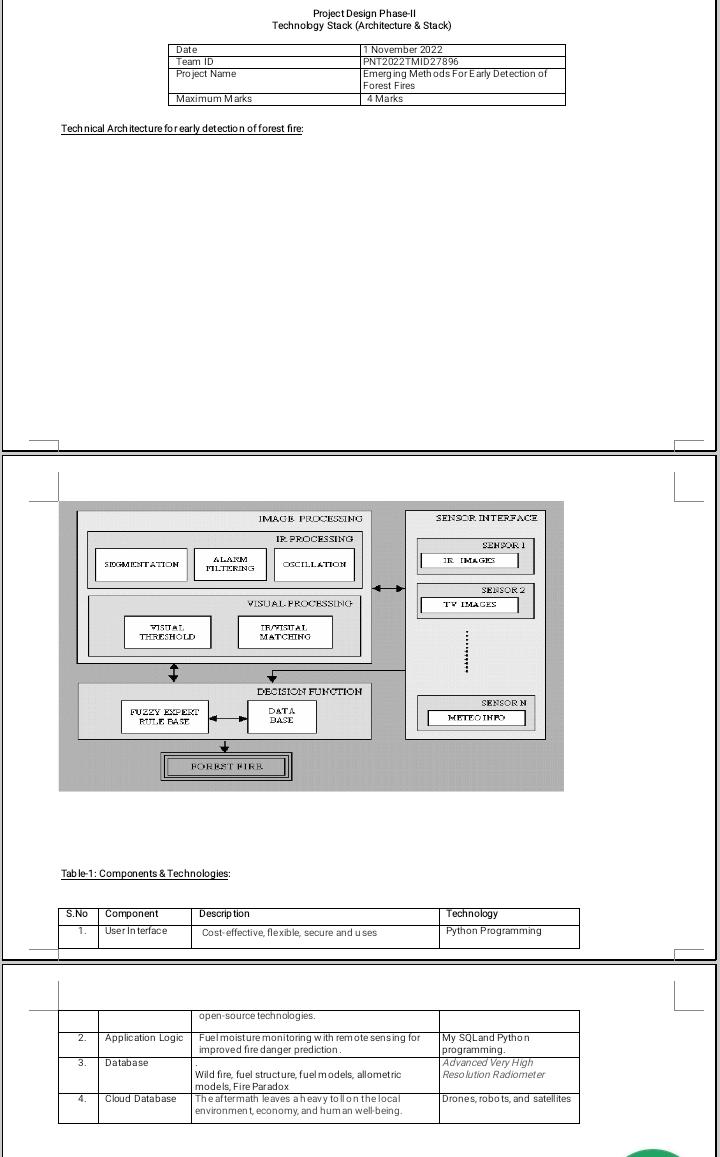
**4.2 CUSTOMER JOURNEY**

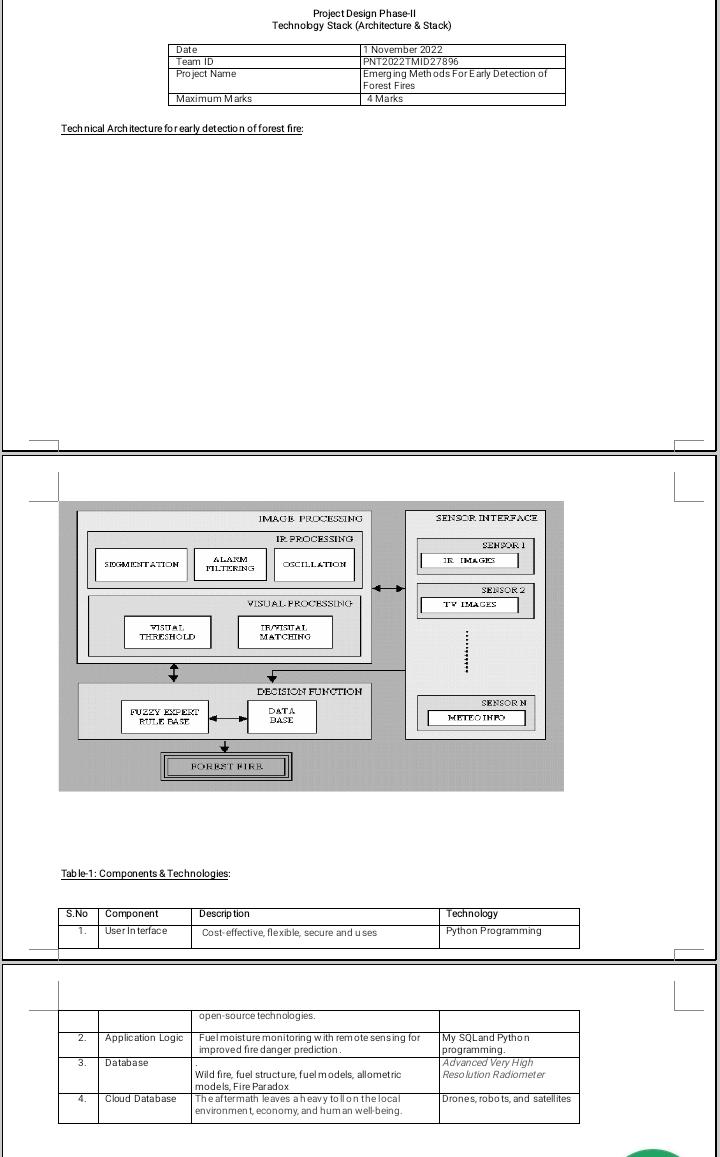
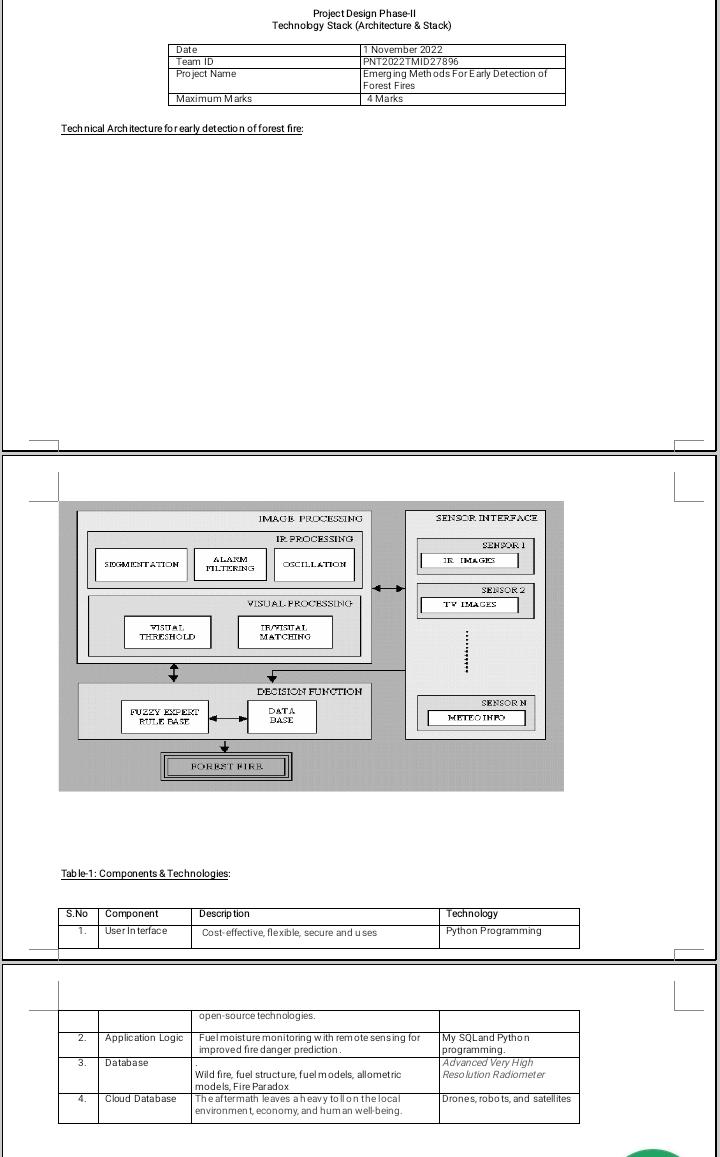


**4.3 FUNCTIONAL REQUIREMENT**





**4.4 TECHNOLOGY ARCHITECTURE**



# 5.CODING SOLUTION

**5.1Feature 1**

In Feature 1 module we have made data collection and Image preprocessing for and Model training.

# importing Required Libraries:

import keras from keras.preprocessing.image import ImageDataGenerator

import matplotlib.pyplot as plt import numpy as np batch\_size = 32

# image resizing and preprocessing :

train\_datagen = ImageDataGenerator( shear\_range=0.2, rotation\_range=180, zoom\_range=0.2, horizontal\_ﬂip=True, ) val\_datagen = ImageDataGenerator( rescale=1./255 )

train\_generator = train\_datagen.ﬂow\_from\_directory( 'train\_set/', target\_size=(150, 150), batch\_size=batch\_size, class\_mode='binary' )

val\_generator = val\_datagen.ﬂow\_from\_directory( 'test\_set/', target\_size=(150, 150), batch\_size=batch\_size, class\_mode='binary' )

**Creating the sequential model**: from keras.models

import Sequential from keras.layers import Convolution2D from keras.layers import MaxPooling2D from keras.layers import Activation from keras.layers import Dropout from keras.layers import Flatten from keras.layers

import Dense model=Sequential()

model.add(Convolution2D(32,(3,3),input\_shape=(150,150,3))) #Convolutional 2D Layer model.add(Activation('relu'))

model.add(MaxPooling2D(pool\_size=(2,2))) # MaxPooling Layer model.add(Flatten()) #Flatten Layer to make a array model.add(Dense(150))

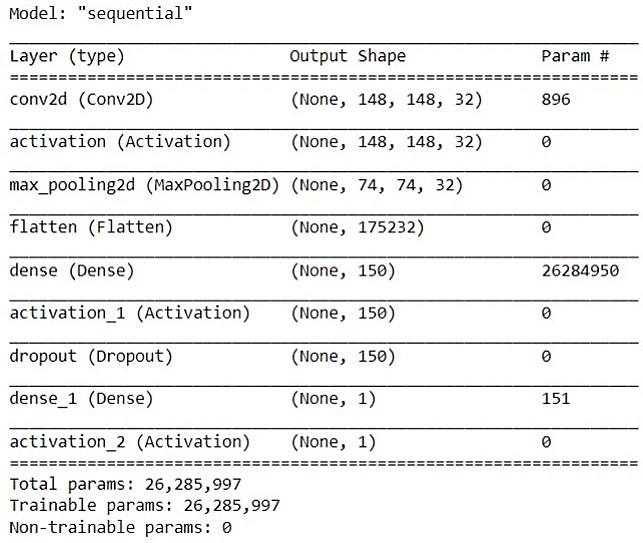
model.add(Activation('relu')) model.add(Dropout(0.5)) model.add(Dense(1))

model.add(Activation('sigmoid'))

model.compile( loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'] )

# Model summary :

model.summary()



# Feature 2

import cv2 import numpy as np

from keras.preprocessing import image from keras.models import load\_model from twilio.rest import Client from playsound import playsound model=load\_model('forest1.h5') video=cv2.VideoCapture(0) name=['forest','with ﬁre'] while(True):

ret,frame=video.read() cv2.imshow('frame',frame) cv2.imwrite('image.jpg',frame) img=image.load\_img('train\_set/forest/NoFire

(1).bmp',target\_size=(64,64)) x=image.img\_to\_array(img) x=np.expand\_dims(x,axis=0) pred=model.predict(x)

index=np.argmax(pred) if index==0:

account\_sid='AC50d663c8a7c2d8b35b1fc09dfda93bda' auth\_token='86f345babfa094d1015a0e1137dbb679' client =Client(account\_sid,auth\_token) message=client.messages \

.create(body='-------Fire is detected,Stay Alert !!! ',

from\_='+19457581434',to='+916369 659 356')

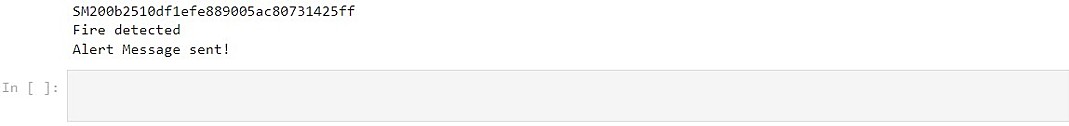
print(message.sid) print('Fire detected') print("Alert Message sent!")

playsound('tornado-siren.mp3')

else:

print('No Danger') cv2.imshow("image.jpg",frame) if cv2.waitkey(2)&0xff == ord('q'):

break video.release() cv2.destroyAllWindows()



# 6.TESTING

**6.1Test Cases & User Acceptance Testing Testing with input video recording from user end:**

import cv2

import numpy as np

from keras.preprocessing import image from keras.models

import load\_model from twilio.rest import Client from playsound import playsound model**=**load\_model('forest1.h5') video**=**cv2**.**VideoCapture(0) name**=**['forest','with ﬁre'] while(True):

ret,frame**=**video**.**read() cv2**.**imshow('frame',frame) cv2**.**imwrite('image.jpg',frame)

img**=**image**.**load\_img('train\_set/forest/NoFire (1).bmp',target\_size**=**(64,64)) x**=**image**.**img\_to\_array(img)

x**=**np**.**expand\_dims(x,axis**=**0)

pred**=**model**.**predict(x) index**=**np**.**argmax(pred) if index**==**0:

account\_sid**=**'AC50d663c8a7c2d8b35b1fc09dfda93bda' auth\_token**=**'86f345babfa094d1015a0e1137dbb679' client **=**Client(account\_sid,auth\_token) message**=**client**.**messages \

**.**create(body**=**'-------Fire is detected,Stay Alert !!! ',

from\_**=**'+19457581434',to**=**'+916369 659 356')

print(message**.**sid) print('Fire detected') print("Alert Message sent!")

playsound('tornado-siren.mp3')

else:

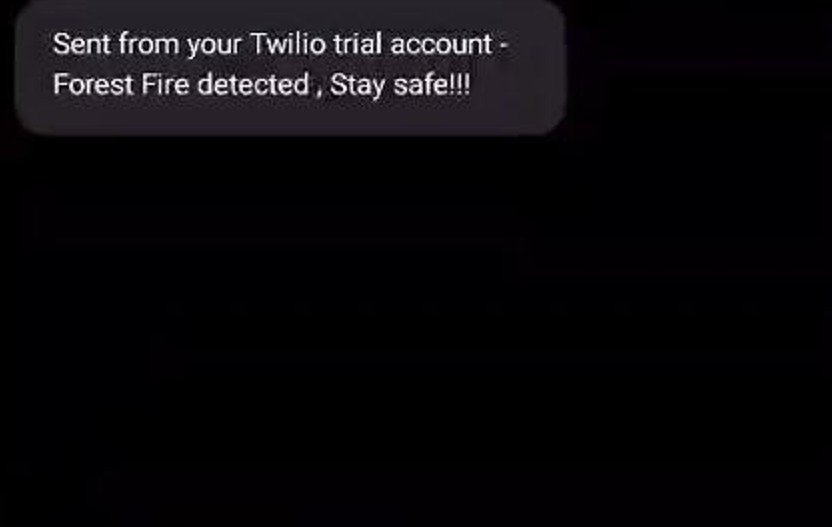
print('No Danger') cv2**.**imshow("image.jpg",frame) if cv2.waitkey(2)&0xff == ord('q'):

break

video**.**release() cv2**.**destroyAllWindows()

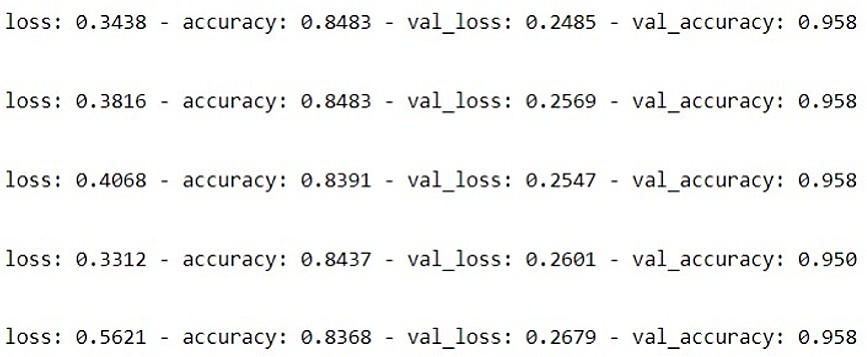
**OUTPUT:**





**7.Result**

# 7.1 Performance Metrics



* 1. **ADVANTAGES & DISADVANTAGES**

# Advantages :

* Easily detect and Estimate the Forest Fire.
* Most Accurate
* Flexible Model which can give maximized outcome
* No Speciﬁc Requirements needed to implement the model

# Disadvanatges:

* Training model is time consuming process.
* Error in CV can cause damage to camera
* Access of camera are prohibited due to personal issues

# 8.CONCLUSION :

Thus we have constructed a model that can identify the effects of the forest ﬁre and it can analyse the forest ﬁre by advanced AI techniques and CNN Algorithm then the Prediction model is Checked and then the model is connected with Twilio account credentials of the Developer consisting of phone numbers of the persons in the surroundings of the people in the area of easy forest ﬁre zone then an security sound alert system is developed to make a alert sound which is downloaded from internet then the entire model is deployed to the IBM Cloud account that we have created was made with the studies we have done.

# 9.FUTURE SCOPES :

1. It can be developed as a Web or Android Application.
2. In future Alternate Advanced technologies can be Implemented.
3. The Identiﬁcation and tracking system can be implemented if possible.

# 10.APPENDIX :

**Source Code :** [**https://github.com/IBM-EPBL/IBM-Project-31025-**](https://github.com/IBM-EPBL/IBM-Project-31025-1660194638/blob/main/Final%20Deliverables/EntireModel.ipynb)[**1660194638/blob/main/Final%20Deliverables/EntireModel.ipynb**](https://github.com/IBM-EPBL/IBM-Project-31025-1660194638/blob/main/Final%20Deliverables/EntireModel.ipynb)

# GitHub & Project Demo Link

**Github:** [**https://github.com/IBM-EPBL/IBM-Project-21389-1659779191**](https://github.com/IBM-EPBL/IBM-Project-21389-1659779191)

**DEMO LINK**

<https://drive.google.com/file/d/1b22f4cDmxQ9abyJnIz6FD80NT2GXLj1D/view?usp=share_link>