A

Major Project

On

MISSING CHILD IDENTIFICATION USING DEEP LEARNING AND LBPH ALGORITHM

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

G.SRAVANI(187R1A05K4)

B.MADHAV(187R1A05J7)

B.SUNIL(187R1A05J1)

Under the Guidance of

Dr.G. SOMASEKHAR

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**MISSING CHILD IDENTIFICATION USING DEEP LEARNING AND LBPH ALGORITHM**" being submitted by **G.SRAVANI(187R1A05K4)** , **B.MADHAV(187R1A05J7)** , **B.SUNIL(187R1A05J1)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

The results embodied in this have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr.G. SOMASEKHAR
Associate Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K.Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on__________________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Dr.G.SOMASEKHAR,** Associate Professor, for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing,help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to the Project Review Committee(PRC) **Mr. A. Uday Kiran , Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao, Mr. A. Kiran Kumar, Mrs. G. Latha** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,**Head,Department of Computer Science and Engineering for providing excellent infrastructure and a nice atmosphere for completing the project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

G.SRAVANI(187R1A05K4)

B.MADHAV(187R1A05J7)

B.SUNIL(187R1A05J1)

ABSTRACT

Crimes are at rise and becoming difficult for police to identify and rescue the Missing Persons. Our Proposed System will use Face Recognition Algorithms and will have the capability for IRIS recognition as well to detect Missing Persons. Face Recognition begins with extracting the coordinates of features such as width of mouth, width of eyes, pupil, and comparing the result with the measurements stored in the database and returning the closest record (facial metrics). Nowadays, there are a lot of face recognition techniques and algorithms found and developed around the world. It is proven by numerous published papers related to facial recognition including facial feature extraction, facial algorithm improvements, and facial recognition implementations. We will be using advanced algorithms like LBPH for our system and also compare to other older algorithms to prove higher accuracy of our system. We will be using Gabor filter algorithm to extract the features of IRIS of individual missing child which can be used for IRIS recognition.Apart from the technicalities, we need to build a robust and easy to use User Interface which will allow the User to use the system and take the benefit of it.

We will be building a Web based system integrated with Backend Machine Learning server. It will allow users to login, upload details of missing children, browse for a missing child, Search a missing child, Report a Missing child and more. The Backend ML system will handle all the search, detection and recognition using our Face Recognition and Iris Recognition Model and all the data stored in the database.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

Building a Web-based system integrated with a Backend Machine Learning server with Features to allow users to login, upload details of missing children, browse for a missing child, Search a missing child, Report a Missing child and more.

1.2 PROJECT PURPOSE

Building an easy to use Web-based system integrated with a Backend Machine Learning server which can be used as an Ecosystem by a Community and Department for Search and Rescue Operation of missing Children.

1.3 PROJECT FEATURES

Usage of Both Face Recognition and Iris Recognition.Web based system integrated with Backend Machine Learning server.Website with Features to allow users to login, upload details of missing child, browse for a missing child, Search a missing child, Report a Missing child and more Easy to Use User Interface.A Ecosystem which a Community and Department can use for Search and Rescue Operation.

2.SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

System Analysis is an important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role as an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

Building an easy to use Web-based system integrated with a Backend Machine Learning server which can be used as an Ecosystem by a Community and Department for Search and Rescue Operation of missing Children.

2.2 EXISTING SYSTEM

- In the existing system , it is time consuming and it is less flexible.
- Mostly missing child cases are reported to police. So even if a child is found, it is difficult to identify him / her from the reported missing cases.
- The chance of loss of records is high and also record searching is difficult.
- Maintenance of the system is also very difficult.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- No Proper way of using and interacting with the System.
- Image filtering is difficult.
- No Ecosystem for a Community or Department to Interact.
- Face of a Child can change quickly in a few years.

2.3 PROPOSED SYSTEM

We will be building a Web based system integrated with a Backend Machine Learning server. It will allow users to login, upload details of a missing child, browse for a missing child, Search a missing child, Report a Missing child and more. The Backend ML system will handle all the search, detection and recognition using our Face Recognition and Iris Recognition Model and all the data stored in the database.We will be using advanced algorithms like LBPH for our system and also compare to other older algorithms to prove higher accuracy of our system. We will be using Gabor filter algorithm to extract the features of IRIS of individual missing child which can be used for IRIS recognition.Apart from the technicalities, we need to build a robust and easy to use User Interface which will allow the User to use the system and take the benefit of it.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features :We will be building a Web based system integrated with a Backend Machine Learning server. Website with Features to allow users to login, upload details of missing child, browse for a missing child, Search a missing child, Report a Missing child and more.A Ecosystem which a Community and Department can use for Search and Rescue Operation.Both Face Recognition and Iris Recognition Available.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business is put forth with a very general plan for the project and some cost estimates.

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give the best return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. The following are some of the important financial questions asked during preliminary investigation:

- They conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, which gives an indication that the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

● Processor	:	i3 or above
● Storage	:	120 GB or Above
● RAM	:	8GB or Above

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

● Operating System	:	Windows 10
● Programming Languages	:	Python 3.10.4 version
● IDE	:	VS Code

3.ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for missing child identification using deep learning

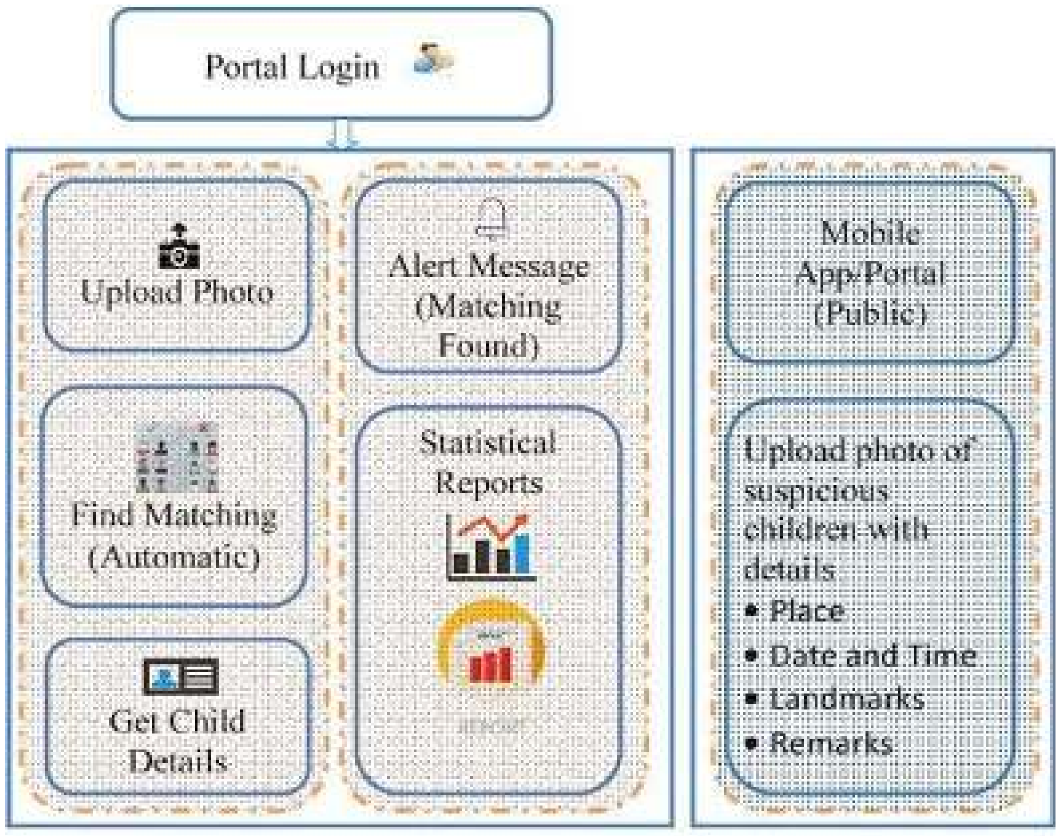

Portal Login
Upload Photo
Find Matching (Automatic)
Get Child Details
Alert Message (Matching Found)
Statistical Reports
Mobile App/Portal (Public)
Upload photo of suspicious children with details
• Place
• Date and Time
• Landmarks
• Remarks

Figure 3.1: Project Architecture of missing child identification using deep learning and lbph algorithm

3.2 DESCRIPTION

Upload Photo : It consists of a national portal for storing details of missing children along with the photo.Whenever a child missing is reported, along with the FIR, the concerned officer uploads the photo of the missing child into the portal. Public can search for any matching child in the database for the images with them.

Face Detection : Firstly,face patterns are generated using a linear binary pattern histogram algorithm.The images are made black and white.Here,the part of the images that looks more like the original lbph face pattern is found .

Extraction : Sixty eight specific points that exist on every face are figured out by using the face landmark estimation algorithm. From the landmarks found,image transformations like scaling, shearing and rotation are used by the open cvs affine transformation.

Iris Detection : In this a gabor filter with discrete wavelet is used in a myriad of image processing applications for edge detection,texture analysis,feature extraction etc..

Result Matching : The classifier has been trained in such a way that it can take the measurements from a test image and gives the closest match as output.

3.3 USE CASE DIAGRAM

In the use case diagram, we have basically two actors who are the police and the user.

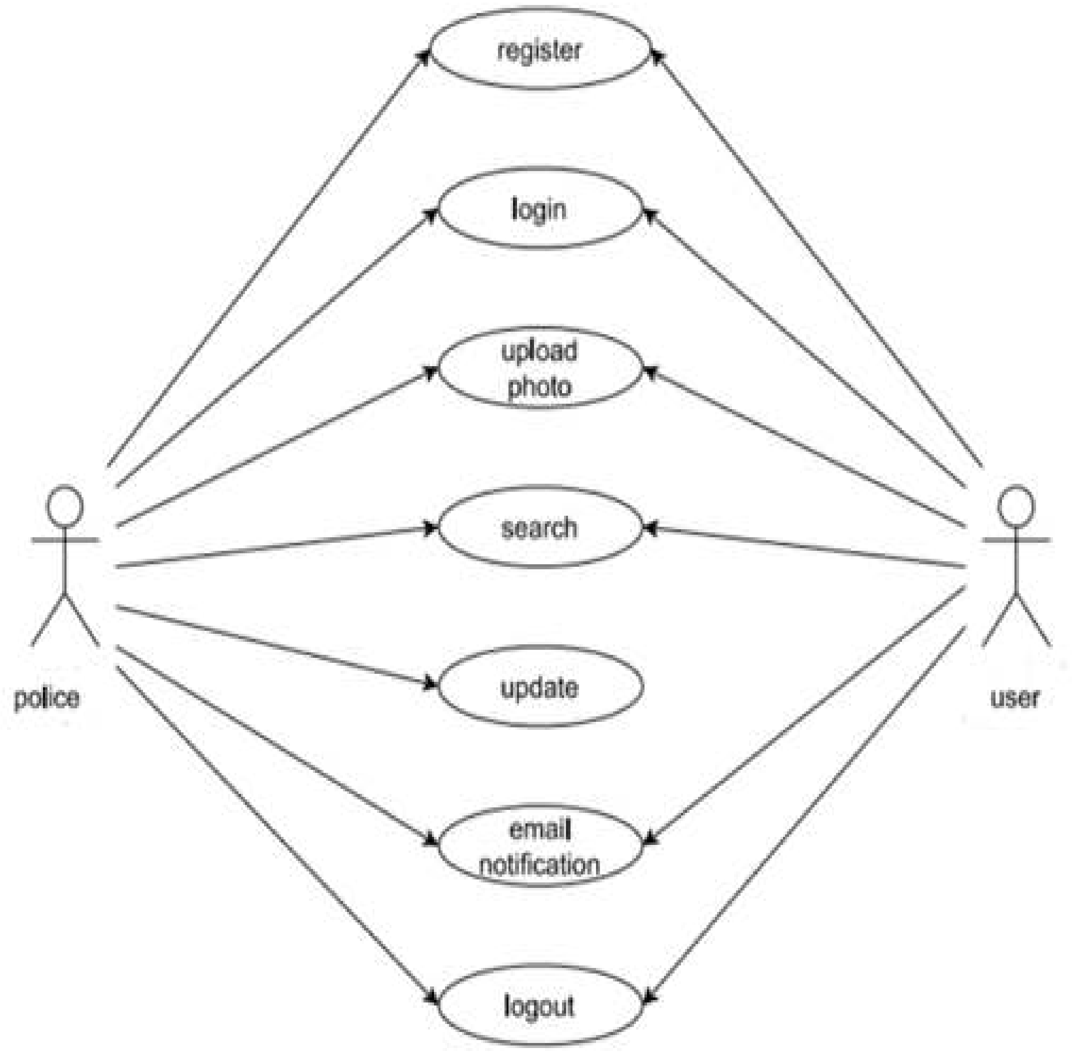


Figure 3.3: Use Case Diagram for missing child identification using deep learning and lbph algorithm

3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects.

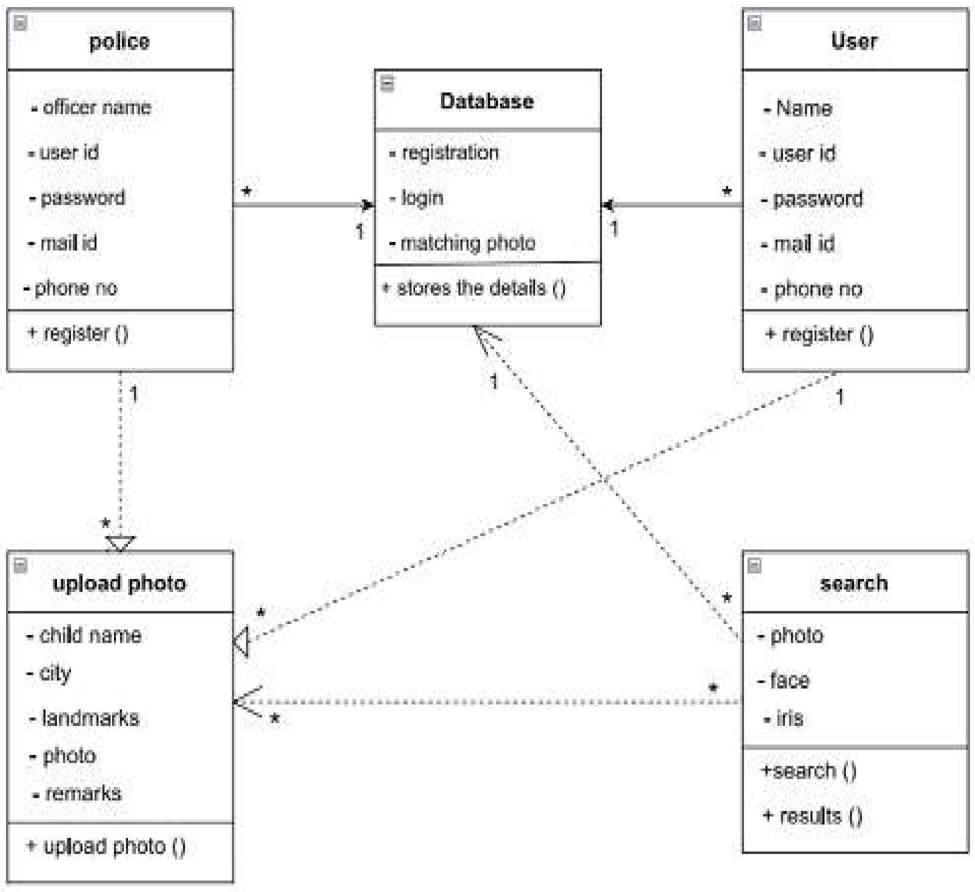


Figure 3.4: Class Diagram for missing child identification using deep learning and lbph algorithm

3.5 SEQUENCE DIAGRAM

It describes the object interactions arranged in a time sequence .

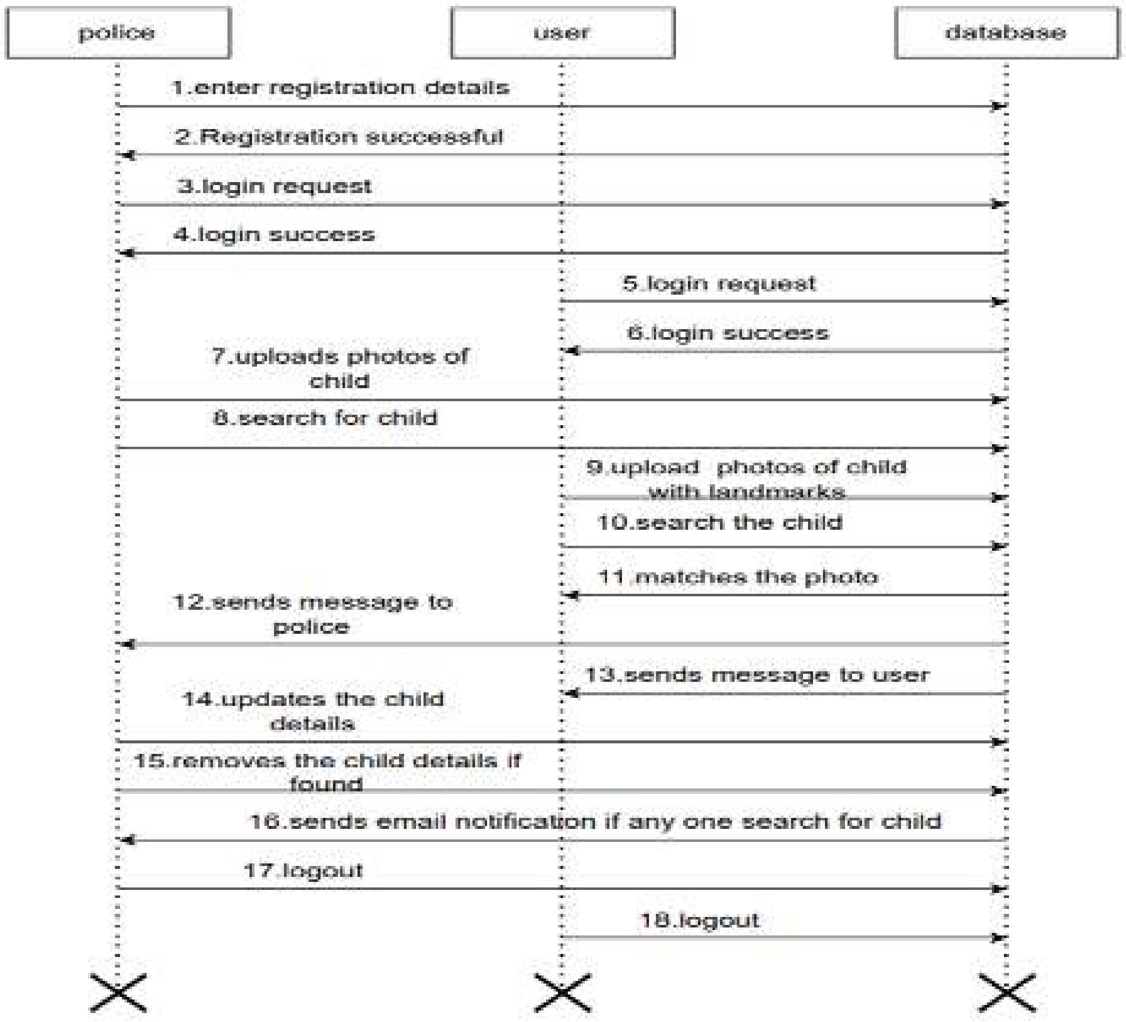

police
user
database
1.enter registration details
2.Registration successful
3.login request
4.login success
5.login request
6.login success
7.uploads photos of child
8.search for child
9.upload photos of child with landmarks
10.search the child
11.matches the photo
12.sends message to police
13.sends message to user
14.updates the child details
15.removes the child details if found
16.sends email notification if any one search for child
17.logout
18.logout

Figure 3.5 Sequence Diagram for missing child identification using deep learning and lbph algorithm

3.6 ACTIVITY DIAGRAM

It describes the flow of activity states.

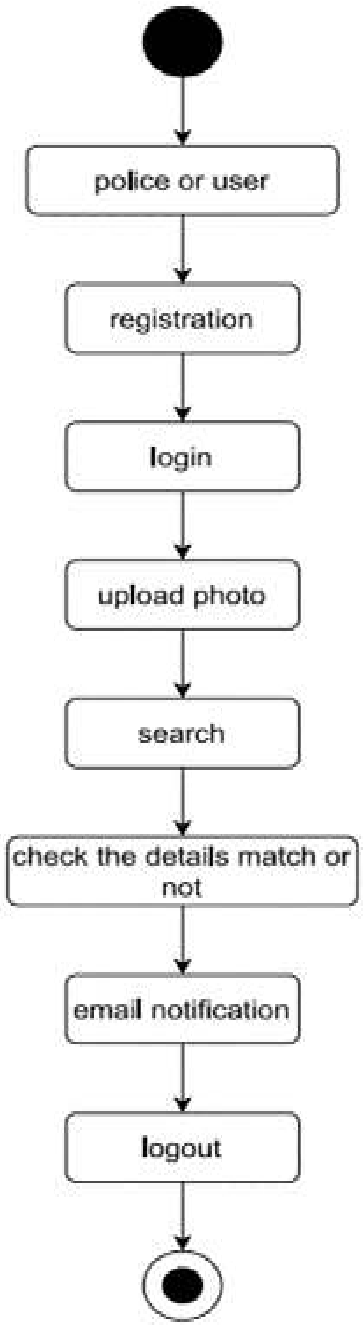


Figure 3.6: Activity Diagram for missing child identification using deep learning and lbph algorithm

4.IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
import base64
import os
import random
import datetime
from weakref import ref
import FaceRecognition
import IrisRecognition
import numpy as np
import pandas as pd
import systemcheck
from multiprocessing import shared_memory
from flask import Flask, render_template, url_for, redirect, jsonify, Response,
abort, session, request, send_file
from werkzeug.utils import secure_filename
import sqlite3
import shutil
from mailSend import send_mail
conn = sqlite3.connect('data.db')
print ("Opened database successfully")
conn.execute('''CREATE TABLE IF NOT EXISTS USERS
    (ID INTEGER PRIMARY KEY AUTOINCREMENT,
    USERNAME          TEXT UNIQUE,
    PASSWORD          TEXT,
    NAME              TEXT,
    EMAIL             TEXT,
    CONTACT           TEXT,
    TYPE              TEXT);''')
    conn.execute('''CREATE TABLE IF NOT EXISTS DETAILS
    (ID INTEGER PRIMARY KEY AUTOINCREMENT,
    NAME      TEXT,
     AGE       TEXT,
    GENDER     TEXT,
    AADHAR     TEXT,
    M_DATE     TEXT,
    PARENT     TEXT,
    P_CONTACT  TEXT,
    P_ADDRESS  TEXT,
    P_REMARK   TEXT,
    F_DATE     TEXT,
    FOUNDER    TEXT,
    F_CONTACT  TEXT,
    F_ADDRESS  TEXT,
    F_REMARK   TEXT,
    STATUS     TEXT);''')
conn.close()
shape = (480, 640, 3)
app = Flask(__name__)
UPLOAD_FOLDER = os.path.join(os.path.dirname(os.path.realpath(__file__)), 'faces/')
IRIS_FOLDER = os.path.join(os.path.dirname(os.path.realpath(__file__)), 'iris/')
TEMP_FOLDER = os.path.join(os.path.dirname(os.path.realpath(__file__)), 'temp/')
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
app.config['IRIS_FOLDER'] = IRIS_FOLDER
app.config['TEMP_FOLDER'] = TEMP_FOLDER
outputFrame = None
number = random.randint(1000000, 9999999)
def html_return(msg, redirect_to = "/", delay = 5):
    return f"""
                <html>
                    <head>
                        <title>Missing Child </title>
                        <meta http-equiv="refresh"
```

```
content="{delay};URL='{redirect_to}'" />
 </head>
 <body>
<h2> {msg}</h2>
 <p>This page will refresh automatically.</p>
</body>
</html>
"""
@app.route('/', methods=['get', 'post'])
def login_page():
   if request.method == 'POST':
      username, password = request.form['username'], request.form['password']
      if username == "niltech" and password == "Niltech@12345":
         session['user'] = username+"_Admin"
         print("Admin Login 1")
         return render_template('index.html', user=(session['user']))
 Else:
         Try:
            conn = sqlite3.connect('data.db')
            print ("Opened database successfully 1")
            cursor = conn.execute(f"SELECT USERNAME, PASSWORD, TYPE, NAME from
USERS")
            for row in cursor:
               if  row[0] == username  and row[1] == password:
                  if len(row[2]) > 1:
                     session['user'] = row[3]+"_Admin"
                     print("Police Login")
                  Else:
                     session['user'] = row[3]
                     print("User Login")
                  conn.close()
                  return render_template('index.html',
user=(session['user']))
 return render_template('login-page.html')
        except Exception as e:
               print("DB Error 1: ", e)
    elif 'user' in session.keys():
        if "_Admin" in session['user']:
            print("Police Login 1")
        Else:
            print("User Login 1")
        return render_template('index.html', user=(session['user']))
    Else:
        return render_template('login-page.html')
@app.route('/logout/')
def logout():
    session.clear()
    return redirect(url_for('login_page'))
@app.route('/add_missing/', methods=['get', 'post'])
def add_missing():
    if 'user' in session.keys():
        if request.method == 'POST':
if 'file' not in request.files:
                return redirect(request.url)
            files = request.files.getlist('file')
            print("Files:",files)
            if files[0].filename == '':
                return redirect(request.url)
if 'irisfiles' in request.files:
                print("Got Iris Images")
            iris_files = request.files.getlist('irisfiles')
            print("Iris Files:",iris_files)
            name = request.form['name']
 #Save Face Files
            if files:
 for i in range(10):
try:
     os.mkdir(os.path.join(app.config['UPLOAD_FOLDER'], name))
                    print("Folder Created (FACE):", name)
                    Break
                except Exception as e:
                        print("Folder Already Exists:", e)
                        name = request.form['name']+str(i)

            for file in files:
                filename = secure_filename(file.filename)
                file.save(os.path.join(app.config['UPLOAD_FOLDER'], name,
filename))
 #Save Iris Files
            if iris_files:
                for i in range(10):
                    Try:
                        os.mkdir(os.path.join(app.config['IRIS_FOLDER'], name))
                        print("Folder Created (IRIS):", name)
                        Break
                    except Exception as e:
                            print("Folder Already Exists:", e)
                            name = request.form['name']+str(i)

                for file in iris_files:
                    Try:
                        filename = secure_filename(file.filename)
                        file.save(os.path.join(app.config['IRIS_FOLDER'], name,filename))
```

```
        Except:
                print("IRIS file..")
  age = request.form['age']
          gender  = request.form['gender']  aadhar  = request.form['aadhar']
  mdate  = request.form['mdate']
          parent  = request.form['parent']
          pcontact  = request.form['pcontact']
          paddress  = request.form['paddress']
          premark  = request.form['premark']
          fdate  = request.form['fdate']
          fname  = request.form['fname']
          fcontact  = request.form['fcontact']
          faddress  = request.form['faddress']
          fremark  = request.form['fremark']
          if files:
             FaceRecognition.add_face(os.path.join(app.config['UPLOAD_FOLDER'],
name), name=name)
             print("Face Training Completed")
          if iris_files:
                         IrisRecognition.add_iris(os.path.join(app.config['IRIS_FOLDER'],  name),
name=name)
             print("Iris Training Completed")
          print("Updating Database")
          conn = sqlite3.connect('data.db')
                 conn.execute(f"INSERT  INTO  DETAILS  (NAME,  AGE,  GENDER,  AADHAR,
M_DATE,   PARENT,   P_CONTACT,   P_ADDRESS,   P_REMARK,   F_DATE,   FOUNDER,
F_CONTACT, F_ADDRESS, F_REMARK, STATUS ) VALUES \  ('{name}','{age}', '{gender}',
'{aadhar}',  '{mdate}',  '{parent}',  '{pcontact}',  '{paddress}',  '{premark}',  '{fdate}',  '{fname}',
'{fcontact}', '{faddress}', '{fremark}', 'Missing' )")
 conn.commit()
          conn.close()
          return redirect(request.url)
return render_template('add_missing.html', user=(session['user']))
   Else:
return redirect(url_for('login_page'))
@app.route('/update_info/', methods=['get', 'post'])
def update_info(): if 'user' in session.keys():
  if "_Admin" not in session['user']:
         return html_return("Only Admin / Police can Add or Update Users")
 if request.method == 'POST':
         name = request.form['name']
         print("Updating Database")
         conn = sqlite3.connect('data.db')
           cursor = conn.execute(f"SELECT ID, AGE, GENDER, AADHAR, STATUS, M_DATE,
PARENT, P_CONTACT, P_ADDRESS, P_REMARK, F_DATE, FOUNDER, F_CONTACT,
F_ADDRESS, F_REMARK FROM DETAILS WHERE NAME='{name}' ")
 data_temp = []
         for row in cursor:
            data_temp = row
         if len(data_temp) > 0:
            print("User Found in Database")
            conn.execute(f"DELETE FROM DETAILS WHERE ID='{data_temp[0]}'")
         Else:
            print("User Not Found in Database")
               return html_return(f"{name} User Details not found in Database. Check for Correct
Name.")
         age = request.form['age']
         if len(age) < 1:
            age = data_temp[1]
         Try:
            gender  = request.form['gender']
            if len(gender) < 1:
               gender = data_temp[2]
         Except:
```

```
gender = data_temp[2]
            aadhar  = request.form['aadhar']
 if len(aadhar) < 1:
                aadhar = data_temp[3]    status = request.form['status’]
if len(status) < 1:
 status = data_temp[4]
            mdate  = request.form['mdate']
            if len(mdate) < 1:
                mdate = data_temp[5]
            parent  = request.form['parent']
            if len(parent) < 1:
                parent = data_temp[6]
            pcontact  = request.form['pcontact']
            if len(pcontact) < 1:
                pcontact = data_temp[7]
            paddress  = request.form['paddress']
            if len(paddress) < 1:
                paddress = data_temp[8]
            premark  = request.form['premark']
            if len(premark) < 1:
                premark = data_temp[9]
            fdate  = request.form['fdate']
            if len(fdate) < 1:
                fdate = data_temp[10]
            fname  = request.form['fname']
            if len(fname) < 1:
                fname = data_temp[11]
            fcontact  = request.form['fcontact']
            if len(fcontact) < 1:
                fcontact = data_temp[12]
            faddress  = request.form['faddress']
            if len(faddress) < 1:
```

```
 faddress = data_temp[13]
fremark  = request.form['fremark']
            if len(fremark) < 1:
                fremark = data_temp[14]
conn.execute(f"INSERT INTO DETAILS (NAME, AGE, GENDER, AADHAR, M_DATE, PARENT, P_CONTACT, P_ADDRESS, P_REMARK, F_DATE, FOUNDER, F_CONTACT, F_ADDRESS, F_REMARK, STATUS ) VALUES \
                ('{name}','{age}', '{gender}', '{aadhar}', '{mdate}', '{parent}', '{pcontact}', '{paddress}', '{premark}', '{fdate}', '{fname}', '{fcontact}', '{faddress}', '{fremark}', '{status}' )")
            conn.commit()
            conn.close()
            print("Updated User Details in Database")
            message = f"Following Missing(s) profile Updated:\n {name}"
            send_mail(message)
            return html_return(f"Updated {name} User Details in Database")
        return render_template('update_info.html', user=(session['user']))
    Else:
        return redirect(url_for('login_page'))
@app.route('/add_user/', methods=['get', 'post'])
def add_admin():
     if "_Admin" not in session['user']:           return html_return("Only Admin / Police can Add or Update Users"
    if 'user' in session.keys():
        if "_Admin" not in session['user']:
            return html_return("Only Admin / Police can Add or Update Users")
 if request.method == 'POST':
            print("Got User Enroll details")
            username = request.form['username']
            password = request.form['password']
            name = request.form['name']
            mail = request.form['mail']
            contact = request.form['phone']
 Type = request.form['Type']
 print("Updating Database", end = " ")
            Try:
               conn = sqlite3.connect('data.db')
       conn.execute(f"INSERT INTO USERS (USERNAME, PASSWORD, NAME, EMAIL,
CONTACT, TYPE) VALUES ('{username}', '{password}', '{name}', '{mail}', '{contact}',
'{Type}' )")
               conn.commit()
               conn.close()
               print("| User Added Successfully")
               message = f"Following Missing(s) profile Added:\n {name}"
               send_mail(message)
            except Exception as e:
               print("Failed. ERROR:", e)
            return redirect(url_for('login_page'))
         return render_template('add_user.html', user=(session['user']))
      Else:
         return redirect(url_for('login_page'))
@app.route('/update_admin/' , methods=['GET', 'POST'])
def update_admin():
   if 'user' in session.keys():
      if "_Admin" not in session['user']:
         return html_return("Only Admin / Police can Add or Update Users")
      print("RM", request.method)
      if request.method == 'POST':
         print("Got Admin Update details")
         userid = request.form['username1']
         print("Got userid")
         password = request.form['password1']
         print("Got userid")
         if password == "DEL":
            if userid != "niltech":
```

```
 Try:  conn = sqlite3.connect('data.db')
                    conn.execute(f"DELETE from USERS where USERNAME = '{userid}';")
                    conn.commit()  conn.close()
return html_return("Successfully Deleted Admin User: "+str(userid), delay = 3)
                except Exception as e:
                    return html_return("Deletion failed for Admin User:
 "+str(userid)+". Reason: "+str(e))
              Else:
                return html_return("Cannot Delete Master Default Admin User: "+str(userid))
            Else:
              Try:conn = sqlite3.connect('data.db')
                        conn.execute(f"UPDATE USERS set PASSWORD = '{password}' where
 USERNAME = '{userid}';")
                conn.commit()
                conn.close()
                return html_return("Password Updated for Admin: "+str(userid) +" if exists.")
              except Exception as e:
                      return html_return("Password Update failed for Admin User: "+str(userid)+".
 Reason: "+str(e))
   Else: return redirect(url_for('login_page'))
 @app.route('/all_missing/')
 def all_missing():
   if 'user' in session.keys():
       conn = sqlite3.connect('data.db')
            cursor = conn.execute("SELECT NAME, AGE, GENDER, M_DATE, STATUS from
 DETAILS")
       users_list = []
       for row in cursor:
          users_list.append(row)
       conn.close()
       return render_template('all_missing.html', user=(session['user']),
 users_list=users_list)
 Else: return redirect(url_for('login_page'))
@app.route('/profile/<name>')
def profile(name):
   if 'user' in session.keys():
      conn = sqlite3.connect('data.db')
      cursor = conn.execute(f"SELECT NAME, AGE, GENDER, M_DATE, STATUS from
DETAILS where NAME = '{name}'")
      cursor = conn.execute(f"SELECT ID, AGE, GENDER, AADHAR, STATUS, M_DATE,
PARENT, P_CONTACT, P_ADDRESS, P_REMARK, F_DATE, FOUNDER, F_CONTACT,
F_ADDRESS,
F_REMARK FROM DETAILS WHERE NAME='{name}' ")
      user_details = []
      for row in cursor:
         user_details = row
      conn.close()
      img_list = os.listdir(UPLOAD_FOLDER + '/' + name)
      Try:
         with open(UPLOAD_FOLDER + '/' + name + '/' + img_list[0], 'rb') as (image):
            image = base64.b64encode(image.read()).decode('utf-8')
      Except:
         with open("dummy-profile-pic.png", 'rb') as (image):
            image = base64.b64encode(image.read()).decode('utf-8')
            return render_template('profile.html', user=(session['user']), user_details=user_details,
image=image, name=name)
   Else:
      return redirect(url_for('login_page'))
@app.route('/delete_user/<name>')
def delete_user(name):
   if 'user' in session.keys():
      suc = 1
      Try:
         conn = sqlite3.connect('data.db')
```

```
conn.execute(f"DELETE from DETAILS where NAME = '{name}';")
conn.commit()
          conn.close()
          message = f"Following Missing(s) profile Deleted:\n {name}"
          send_mail(message)
      except Exception as e:
          print("Unable to delete User from Database. Reason:", e)
          suc = 0
      Try:
          FaceRecognition.remove_face(name)
      except Exception as e:
          print("Unable to delete User from Face Model. Reason:", e)
          suc = 0
      Try:
          IrisRecognition.remove_iris(name)
      except Exception as e:
          print("Unable to delete User from Iris Model. Reason:", e)
          suc = 0
      Try:
          shutil.rmtree(os.path.join(app.config['UPLOAD_FOLDER'], name))
      except Exception as e:
          print("Unable to delete Face Images. Reason:", e)
          suc = 0
      Try:
          shutil.rmtree(os.path.join(app.config['IRIS_FOLDER'], name))
      except Exception as e:
          print("Unable to delete Iris Images. Reason:", e)
          suc = 0
      if suc == 0:
          return html_return("Deletion Completed with some Issues")
      return html_return("User Deletion Completed")
  Else: return redirect(url_for('login_page'))
```

```
@app.route('/searchname/', methods=['get', 'post'])
 def searchname():
     if 'user' in session.keys():
         if request.method == 'POST':
             name = request.form['name']
             name = name.strip()
             if len(name) < 1:
                 return html_return("Kindly Enter Some Name to Search")
             users_list = []
             conn = sqlite3.connect('data.db')
                 cursor = conn.execute(f"SELECT NAME, AGE, GENDER, M_DATE, STATUS from
 DETAILS WHERE NAME LIKE '%{name}%'")
             for row in cursor:
                 users_list.append(row)
                 cursor = conn.execute(f"SELECT NAME, AGE, GENDER, M_DATE, STATUS from
 DETAILS WHERE PARENT LIKE '%{name}%'")
             for row in cursor:
                 users_list.append(row)
                 cursor = conn.execute(f"SELECT NAME, AGE, GENDER, M_DATE, STATUS from
 DETAILS WHERE FOUNDER LIKE '%{name}%'")
             for row in cursor:
                 users_list.append(row)
             conn.close()
             if len(users_list) > 0:
                 message = "Following Missing(s) were found while Name Search:\n"
                 for nm in users_list:
                     message += f"{nm} \n"
                 send_mail(message)
                 return render_template('all_missing.html', user=(session['user']), users_list=users_list)
             Else:
                 return html_return("Sorry... No Matching Name Found.", redirect_to="/add_missing/")
         return render_template('index.html', user=(session['user']))
```

```
  Else:return redirect(url_for('login_page'))
@app.route('/searchaddress/', methods=['get', 'post'])
def searchaddress():
    if 'user' in session.keys():
        if request.method == 'POST':
            address = request.form['address']
            address = address.strip()
            if len(address) < 1:
                return html_return("Kindly Enter Some Address to Search")
            users_list = []
            conn = sqlite3.connect('data.db')
                cursor = conn.execute(f"SELECT NAME, AGE, GENDER, M_DATE, STATUS from
DETAILS WHERE P_ADDRESS LIKE '%{address}%'")
            for row in cursor:
                users_list.append(row)
                cursor = conn.execute(f"SELECT NAME, AGE, GENDER, M_DATE, STATUS from
DETAILS WHERE F_ADDRESS LIKE '%{address}%'")
            for row in cursor:
                users_list.append(row)
            conn.close()
            if len(users_list) > 0:
                message = "Following Missing(s) were found while Address Search:\n"
                for nm in users_list:
                    message += f"{nm} \n"
                send_mail(message)
                return render_template('all_missing.html', user=(session['user']), users_list=users_list)
            Else:
                                        return html_return("Sorry... No Matching Address Found.",
redirect_to="/add_missing/")
        return render_template('index.html', user=(session['user']))
    Else:
        return redirect(url_for('login_page'))
```

```
@app.route('/searchface/', methods=['get', 'post'])
def searchface():
  if 'user' in session.keys():
        if request.method == 'POST':
            if 'file' not in request.files:
                return redirect(request.url)
            shutil.rmtree(os.path.join(app.config['TEMP_FOLDER']))
            Try:
                os.mkdir("temp")
            Except:
                Pass
            print("Searching Faces")
            files = request.files.getlist('file')
            print("Files:",files)
            if files[0].filename == '':
                return redirect(request.url)
            #Save Face Files
                                    foldername = str(datetime.datetime.now())[:-7].replace(" ",
"").replace("-","").replace(":","")
            if files:
                os.mkdir(os.path.join(app.config['TEMP_FOLDER'], foldername))
                print("Folder Created:", foldername)
                for file in files:
                    filename = secure_filename(file.filename)
                            file.save(os.path.join(app.config['TEMP_FOLDER'], foldername, filename))
recognised = FaceRecognition.checkface_folder(os.path.join(app.config['TEMP_FOLDER'],
foldername))
            print("Face Search Completed")
            print(recognised)
            users_list = []
            conn = sqlite3.connect('data.db')
            for name in set(recognised):
```

```
  cursor = conn.execute(f"SELECT NAME, AGE, GENDER, M_DATE, STATUS from
DETAILS WHERE NAME LIKE '%{name}%' ")
 for row in cursor:
users_list.append(row)
            conn.close()
            if len(users_list) > 0:
                message = "Following Missing(s) were found while Face Search:\n"
                for nm in users_list:
                    message += f"{nm} \n"
                send_mail(message)
                return render_template('all_missing.html', user=(session['user']), users_list=users_list)
            Else:

return html_return("Sorry... No Matching Face Found.",
redirect_to="/add_missing/")
        return render_template('index.html', user=(session['user']))
    Else:
        return redirect(url_for('login_page'))
@app.route('/searchiris/', methods=['get', 'post'])
def searchiris():
    if 'user' in session.keys():
        if request.method == 'POST':
            if 'file' not in request.files:
                return redirect(request.url)
            shutil.rmtree(os.path.join(app.config['TEMP_FOLDER']))
            Try:
                os.mkdir("temp")
            Except:
                Pass
            print("Searching Iris")
            files = request.files.getlist('file')
            print("Files:",files) if files[0].filename == '':
```

```
return redirect(request.url)
 #Save Face Files
folder name = str(datetime.datetime.now())[:-7].replace(" ", "").replace("-","").replace(":","")
 if files:
        os.mkdir(os.path.join(app.config['TEMP_FOLDER'], foldername))
        print("Folder Created:", foldername)
        for file in files:
          filename = secure_filename(file.filename)
                file.save(os.path.join(app.config['TEMP_FOLDER'], foldername, filename))
recognised      =      IrisRecognition.checkiris_folder(os.path.join(app.config['TEMP_FOLDER'],
foldername))
      print("Iris Search Completed")
      print(recognised)
      users_list = []
      conn = sqlite3.connect('data.db')
      for name in set(recognised):
          cursor = conn.execute(f"SELECT NAME, AGE, GENDER, M_DATE, STATUS from
DETAILS WHERE NAME LIKE '%{name}%' ")
        for row in cursor:
          users_list.append(row)
      conn.close()
      if len(users_list) > 0:
        message = "Following Missing(s) were found while Iris Search:\n"
        for nm in users_list:
          message += f"{nm} \n"
        send_mail(message)
        return render_template('all_missing.html', user=(session['user']), users_list=users_list)
      Else:
        return html_return("Sorry... No Matching Iris Found.", redirect_to="/add_missing/")
    return render_template('index.html', user=(session['user']))
  Else:
    return redirect(url_for('login_page'))
```

```
@app.errorhandler(404)
def nice(_):
return render_template('error_404.html')
app.secret_key = 'q12q3q4e5g5htrh@werwer15454'
if __name__ == '__main__':
    app.run(host='0.0.0.0', port= 5000, debug = True)#80)
# global outputFrame ## Warning: Unused global
```

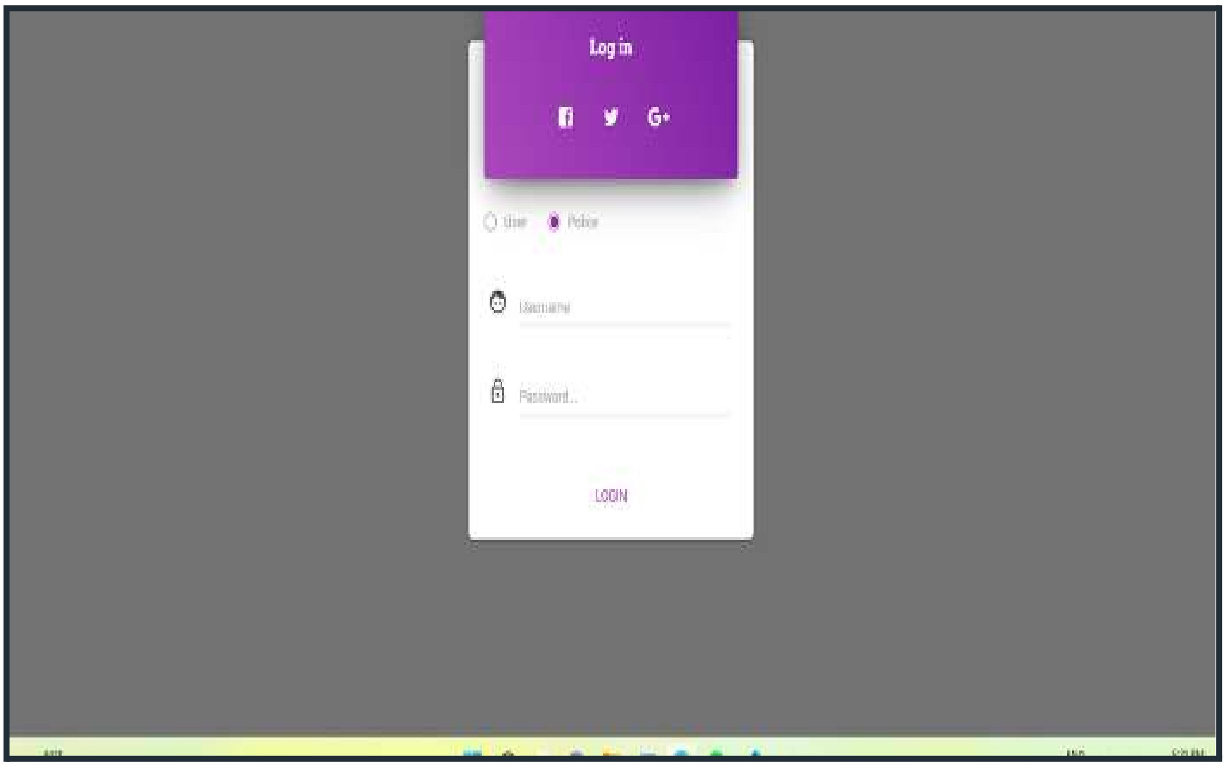
DRIVE LINK :

https://docs.google.com/document/d/18v7jTRZ5aZieFoRkFNQlFPfwL-KzqFU7yrI-wcdaONM/edit?usp=sharing

5. RESULTS

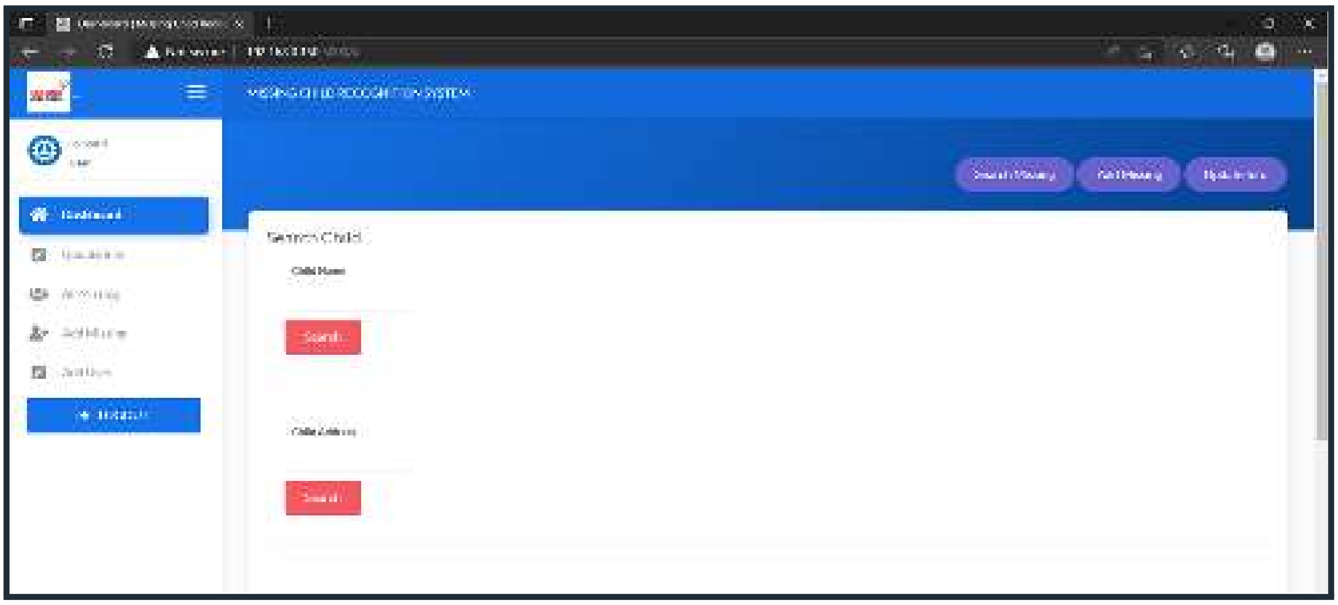
5.RESULTS

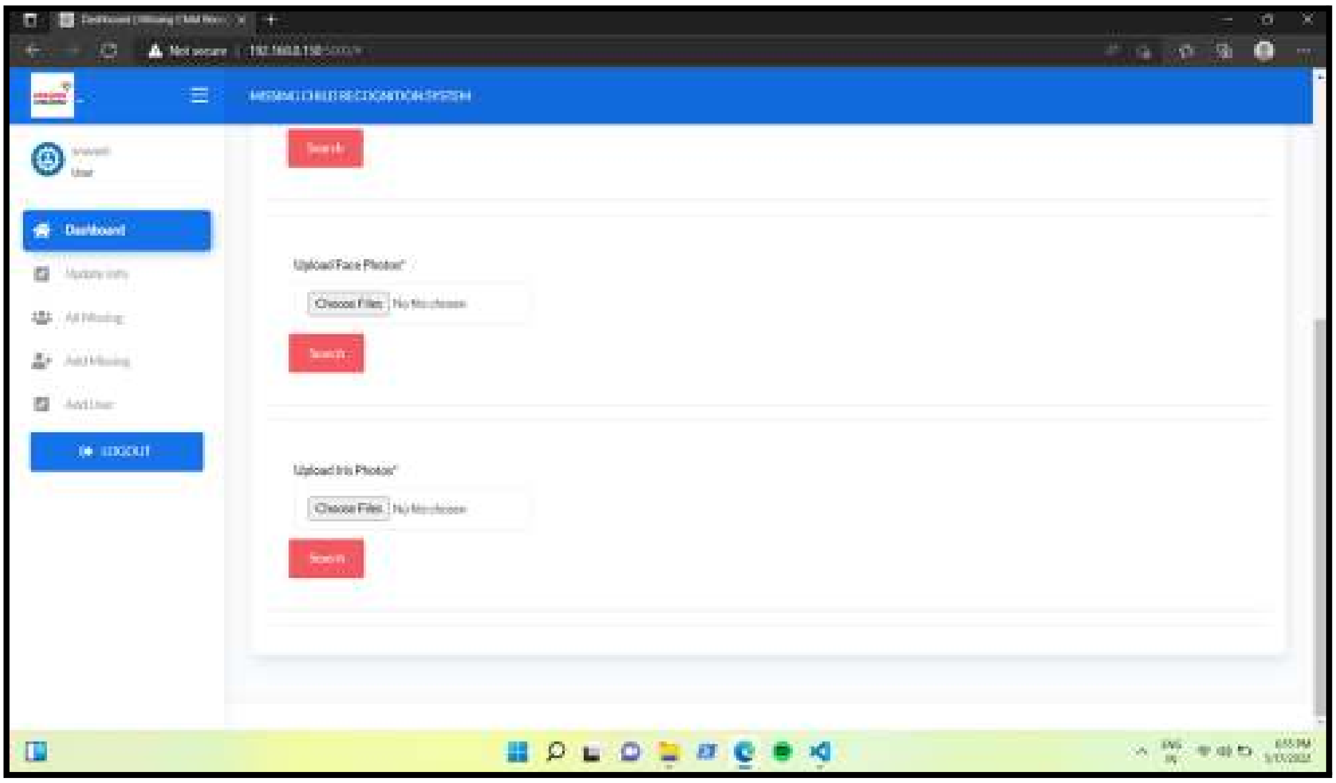
5.1 SCREENSHOT : LOGIN PAGE



Screenshot 5.1: login page for user and police

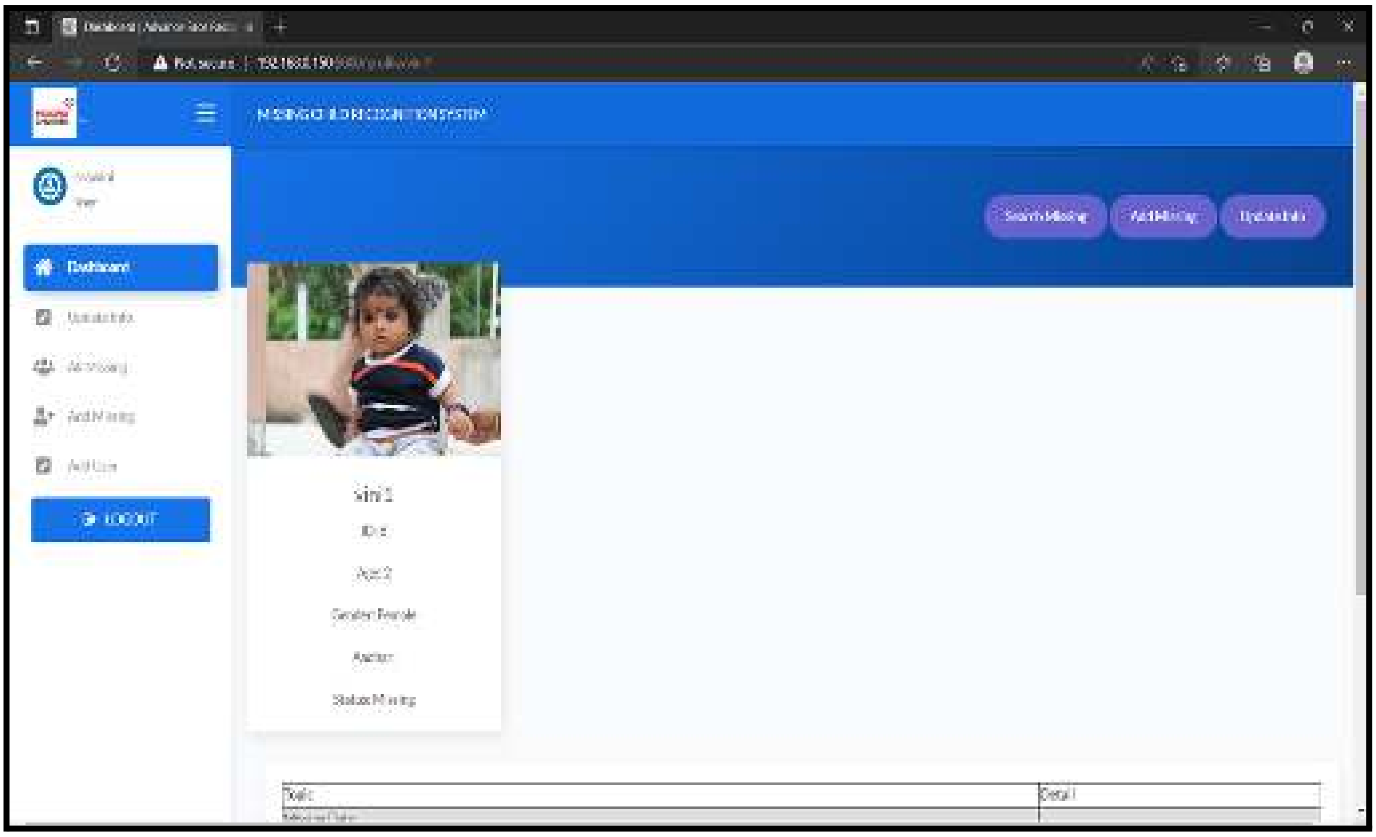
5.2 SCREENSHOT : SEARCHING MISSING CHILD





Screenshot 5.2 : Dashboard for searching missing child

5.3 SCREENSHOT : MISSING CHILD DETAILS



vini1

ID: 6

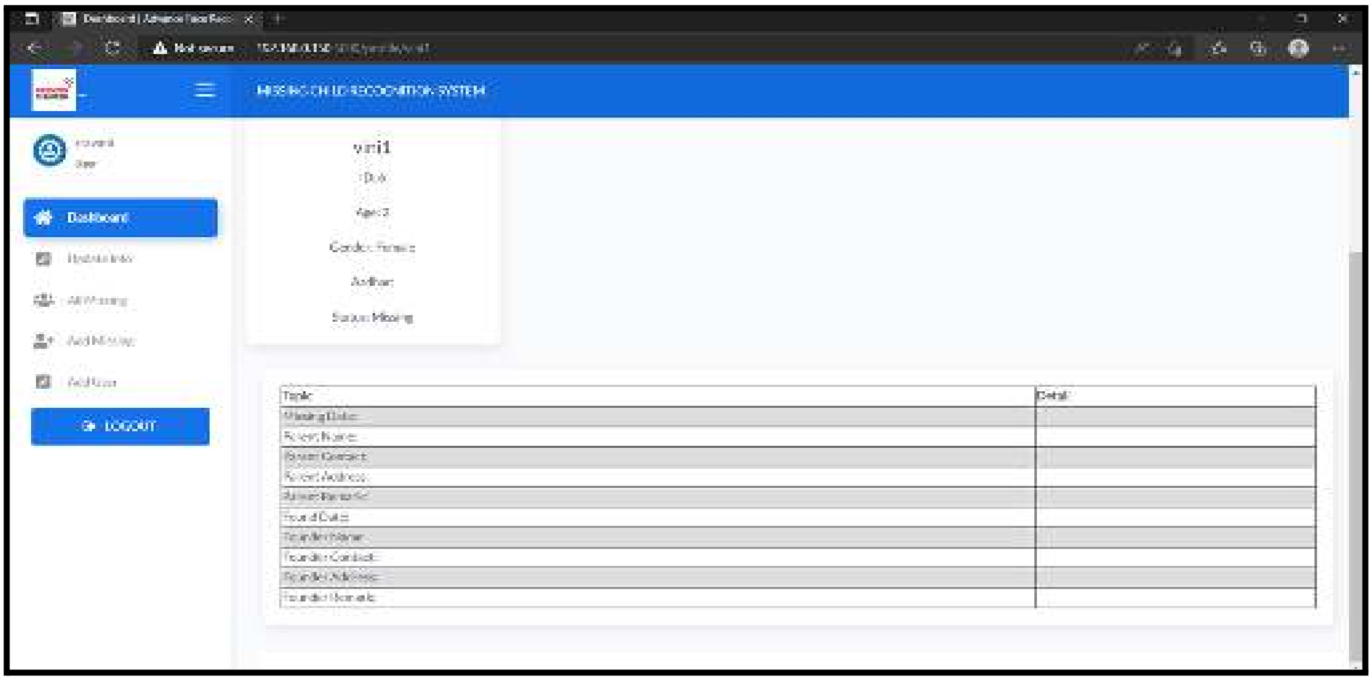
Age: 2

Gender: Female

Aadhar:

Status: Missing

Topic	Detail
Missing Date:	



vini1

ID: 6

Age: 2

Gender: Female

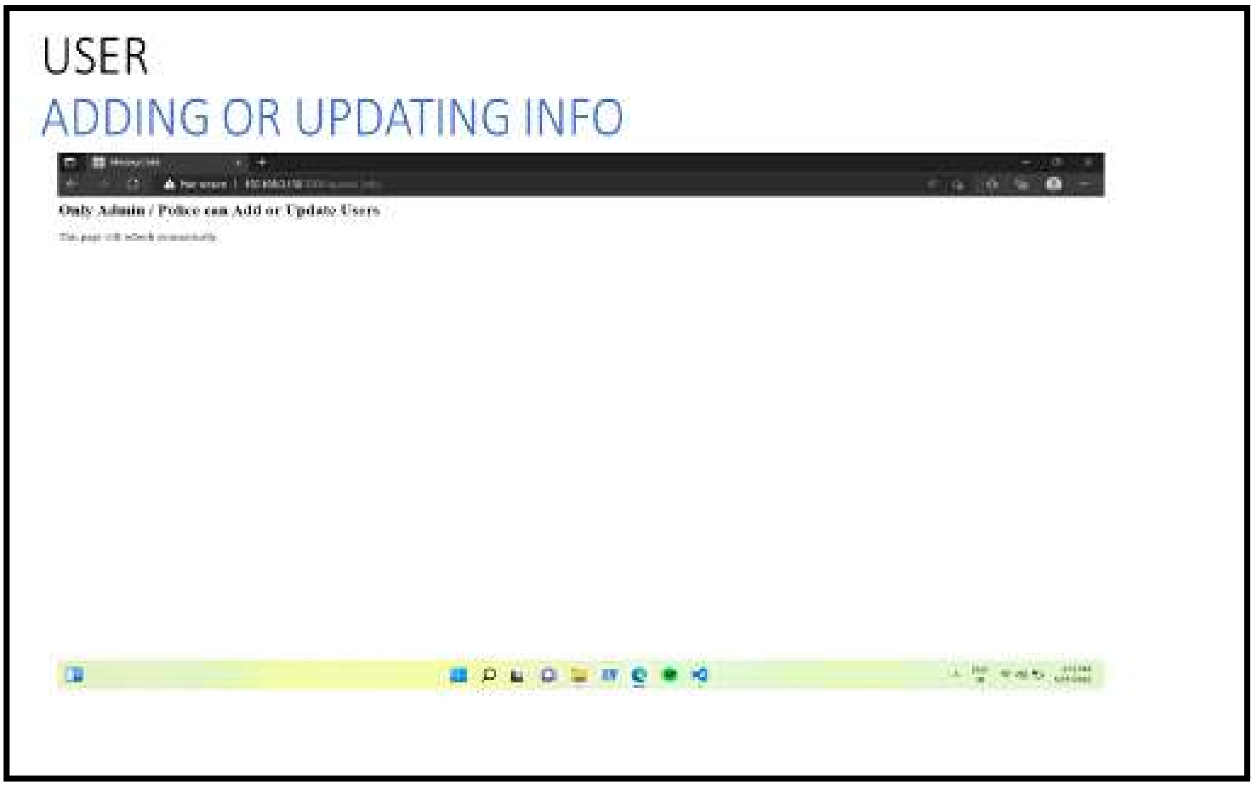
Aadhar:

Status: Missing

Topic	Detail
Missing Date:	
Parent Name:	
Parent Contact:	
Parent Address:	
Parent Remarks:	
Found Date:	
Founder Name:	
Founder Contact:	
Founder Address:	
Founder Remarks:	

Screenshot 5.3 : Missing child detail

5.4 SCREENSHOT : USER (ADDING OR UPDATING INFO)


USER

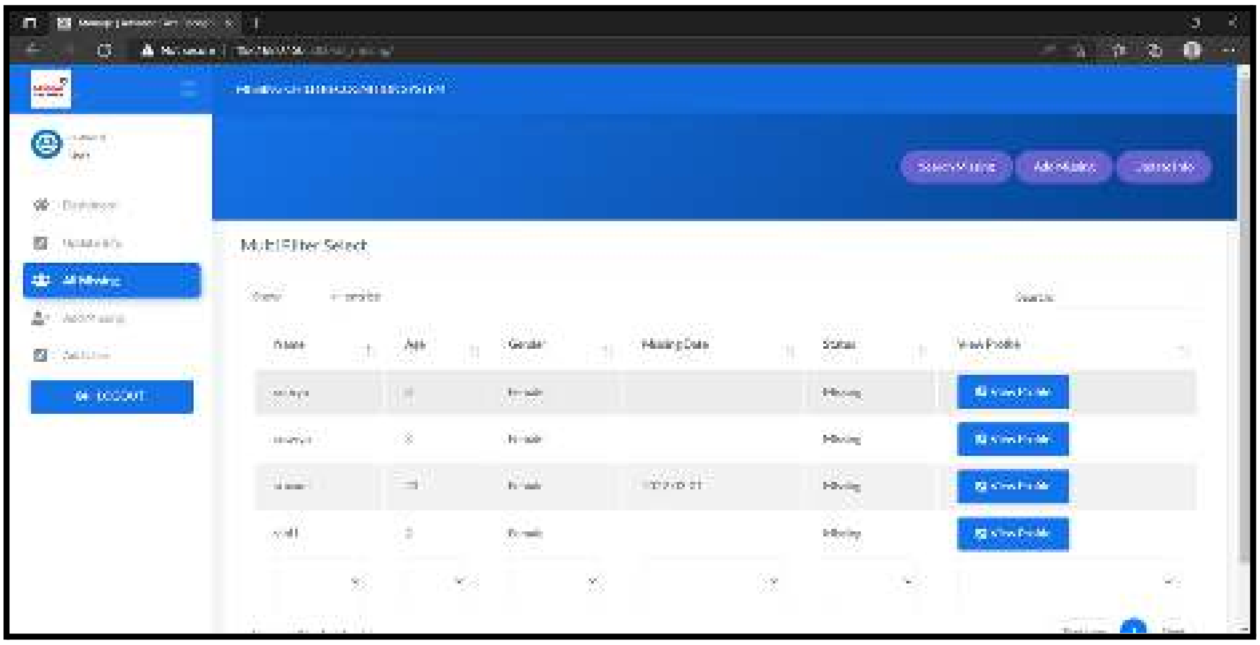
ADDING OR UPDATING INFO

Only Admin / Police can Add or Update Users

This page will refresh automatically

Screenshot 5.4 : User(Adding or Updating Info)

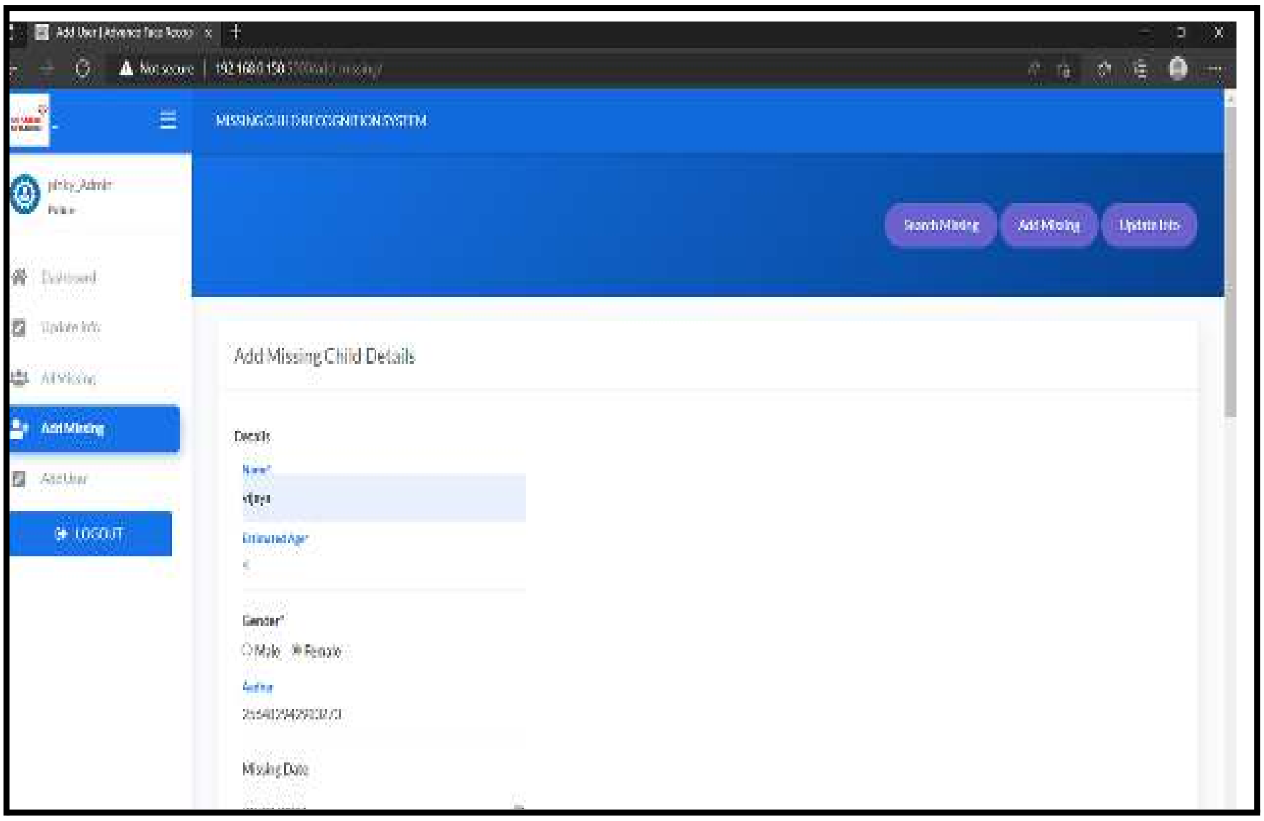
5.5 SCREENSHOT: ALL MISSING CHILD DETAILS


Multi Filter Select

Name	Age	Gender	Missing Date	Status	View Profile
[illegible]	[illegible]	Female		Missing	View Profile
[illegible]	[illegible]	Female		Missing	View Profile
[illegible]	[illegible]	Female	[illegible]	Missing	View Profile
[illegible]	[illegible]	Female		Missing	View Profile

Screenshot 5.5 : All Missing Child details

5.6 SCREENSHOT : ADDING MISSING CHILD DETAILS



Add Missing Child Details

Details

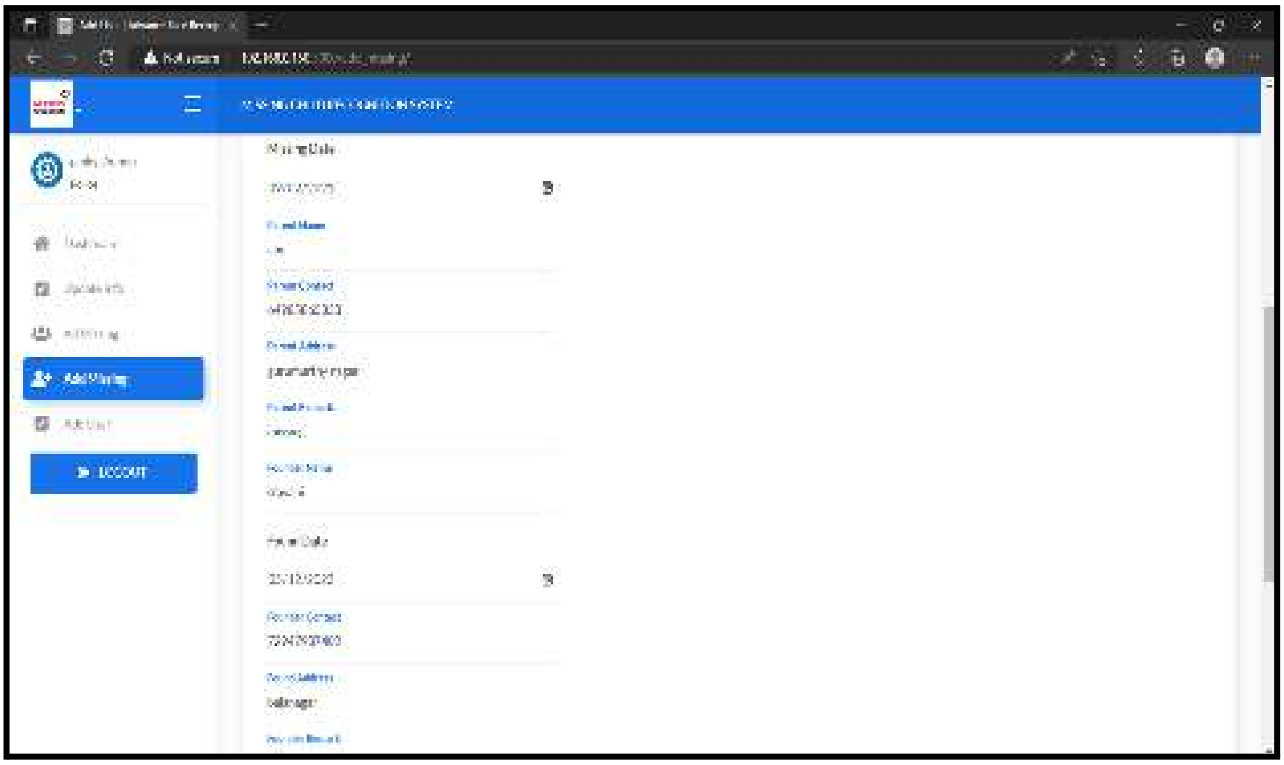
Name*
vijaya

Estimated Age*
[illegible]

Gender*
Male [✓ Female]

Aadhar
[illegible]

Missing Date



Missing Date
[illegible]

Parent Name
[illegible]

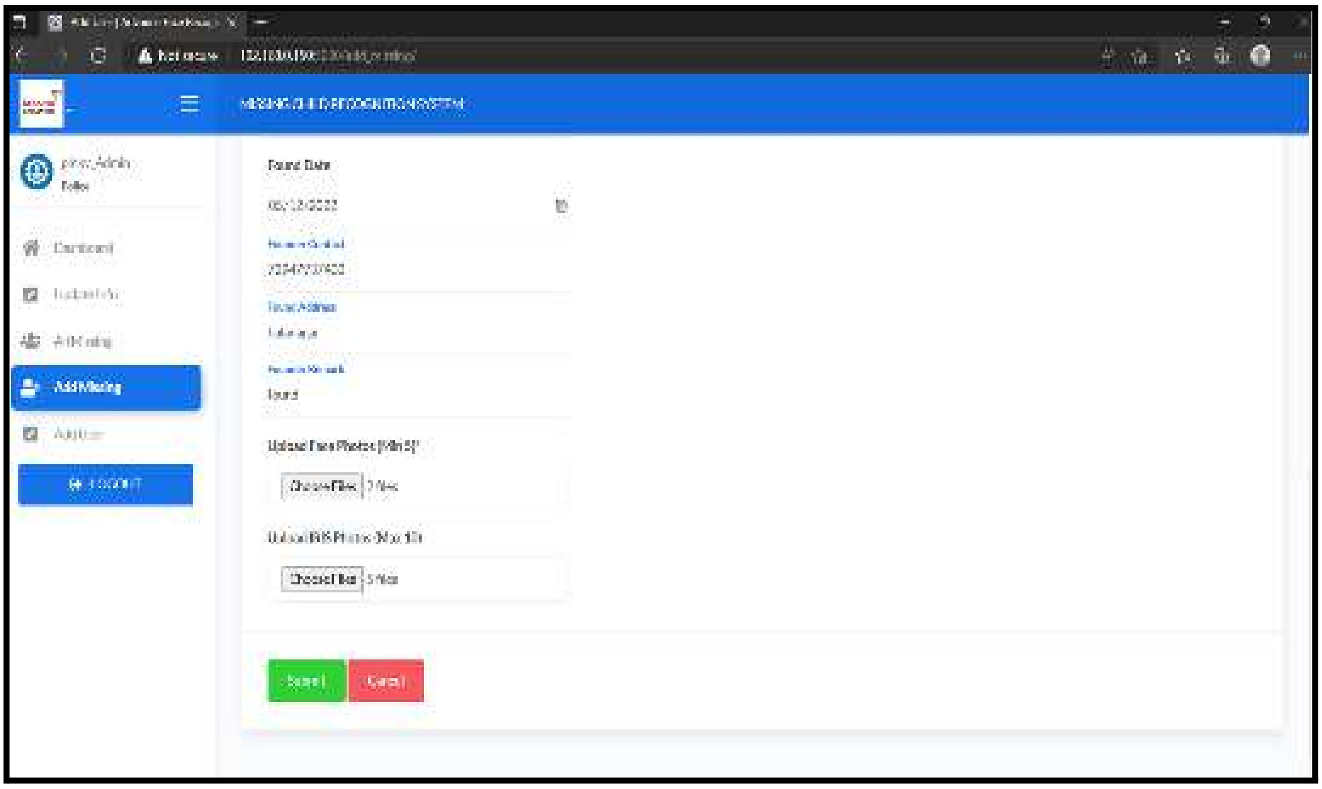
Parent Contact
[illegible]

Parent Address
[illegible]

Found Date
[illegible]

Founder Contact
[illegible]

Found Address
[illegible]



Found Date

05/12/2022

[illegible]

[illegible]

Found Address

[illegible]

[illegible]

Found

Upload Face Photos (Min 5)*

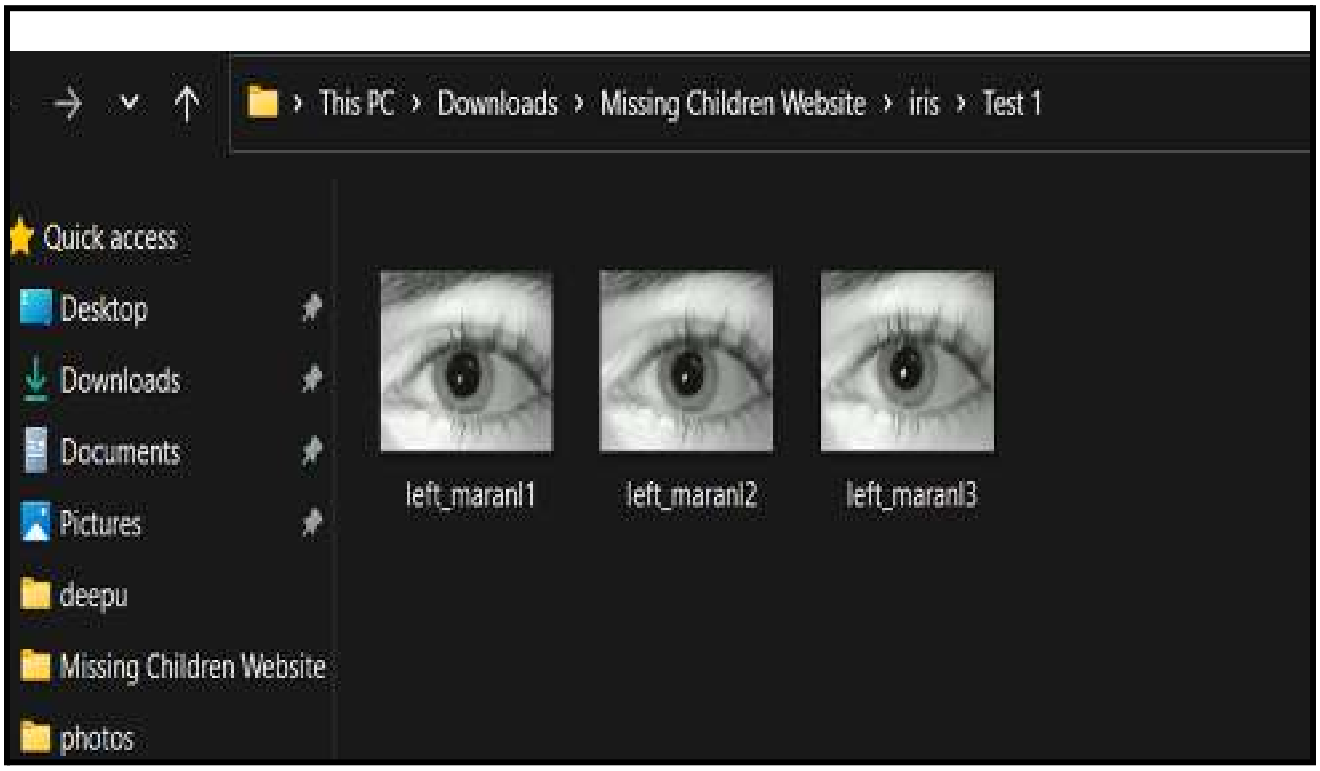
Choose Files | 7 files

Upload IRIS Photos (Max 10)

Choose Files | 3 files

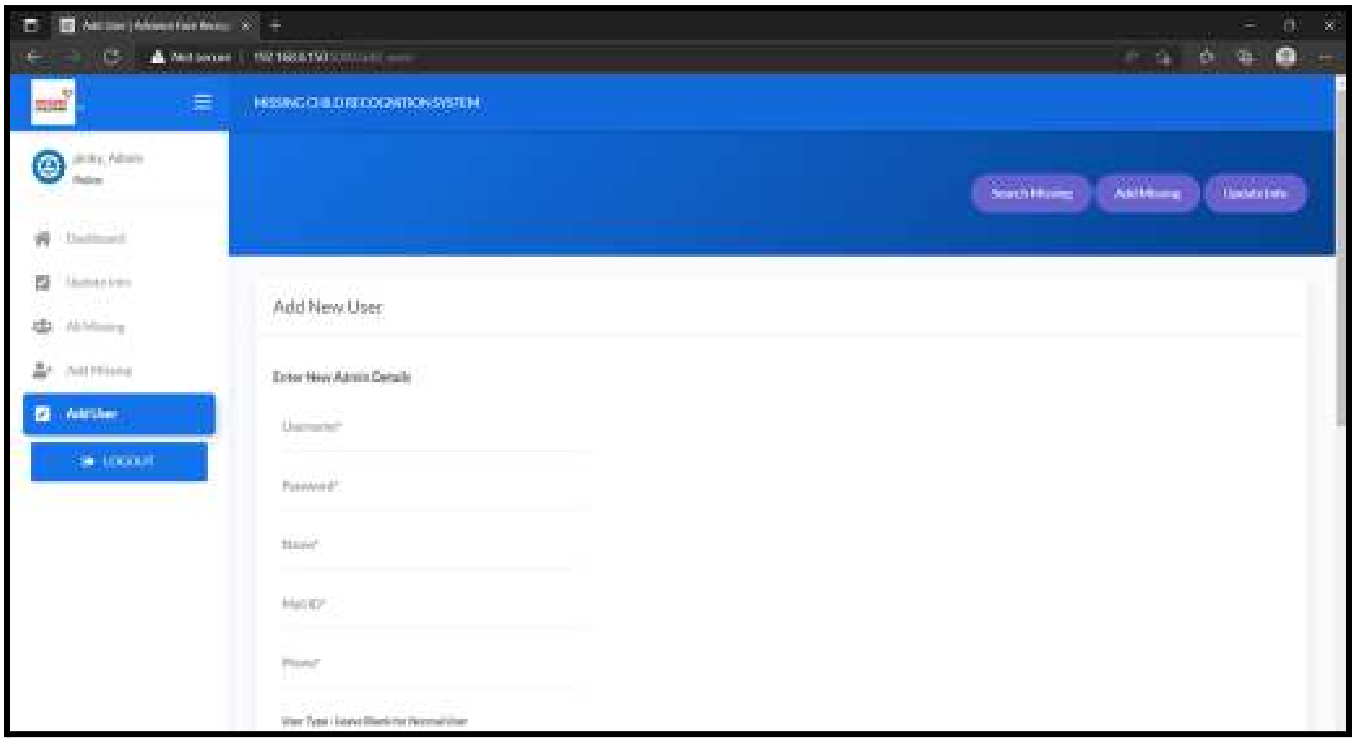
Screenshot 5.6 : Adding missing child details

5.7 SCREENSHOT : IRIS IMAGES



Screenshot 5.7 : Iris images in folder

5.8 SCREENSHOT : POLICE (ADDING NEW USER)



Add New User

Enter New Admin Details

Username*

Password*

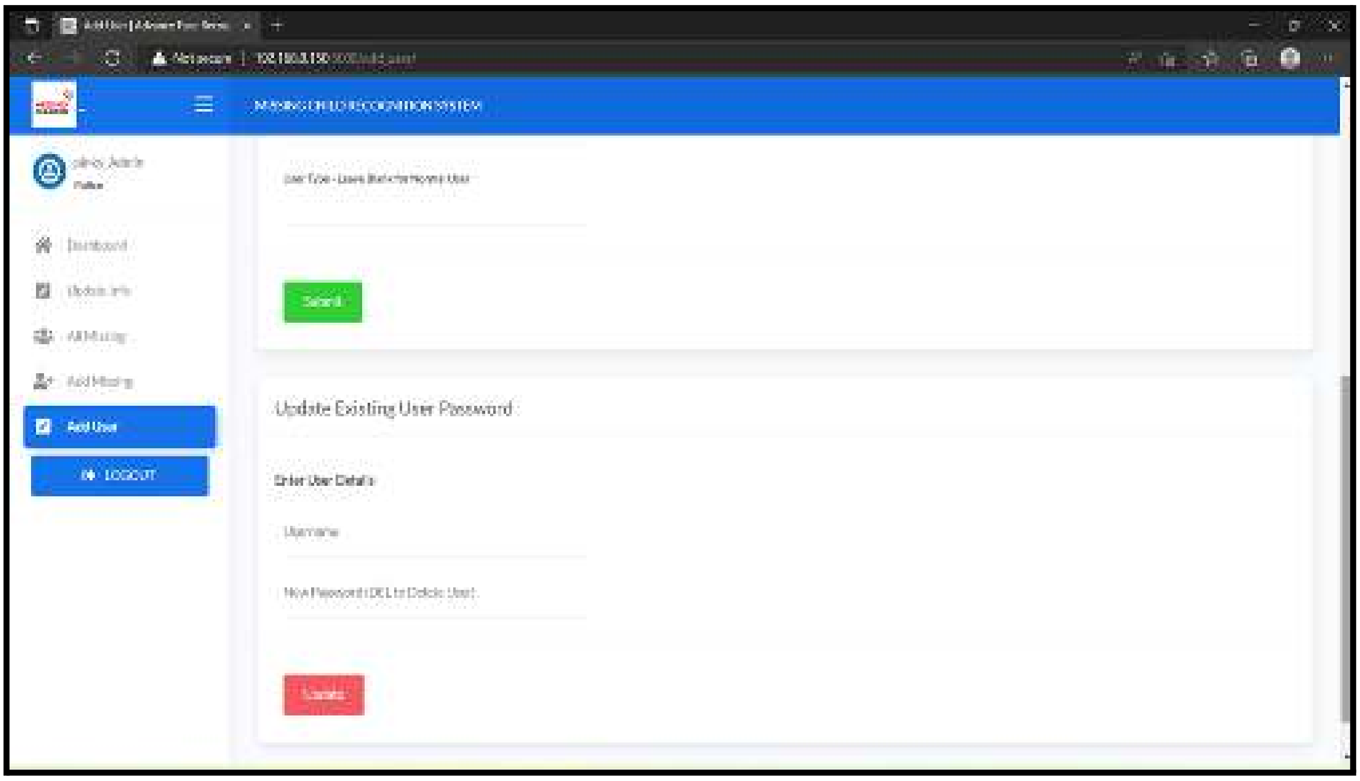
Name*

Mail ID*

Phone*

User Type : Leave Blank for Normal User

Screenshot 5.8.1 : Police(Adding new user)


Add User | Advance Face Recog
Not secure | 192.168.0.150
MISSING CHILD RECOGNITION SYSTEM
pinky, Admin
Police
Dashboard
Update Info
All Missing
Add Missing
Add User
LOGOUT

User Type : Leave Blank for Normal User

Submit

Update Existing User Password

Enter User Details

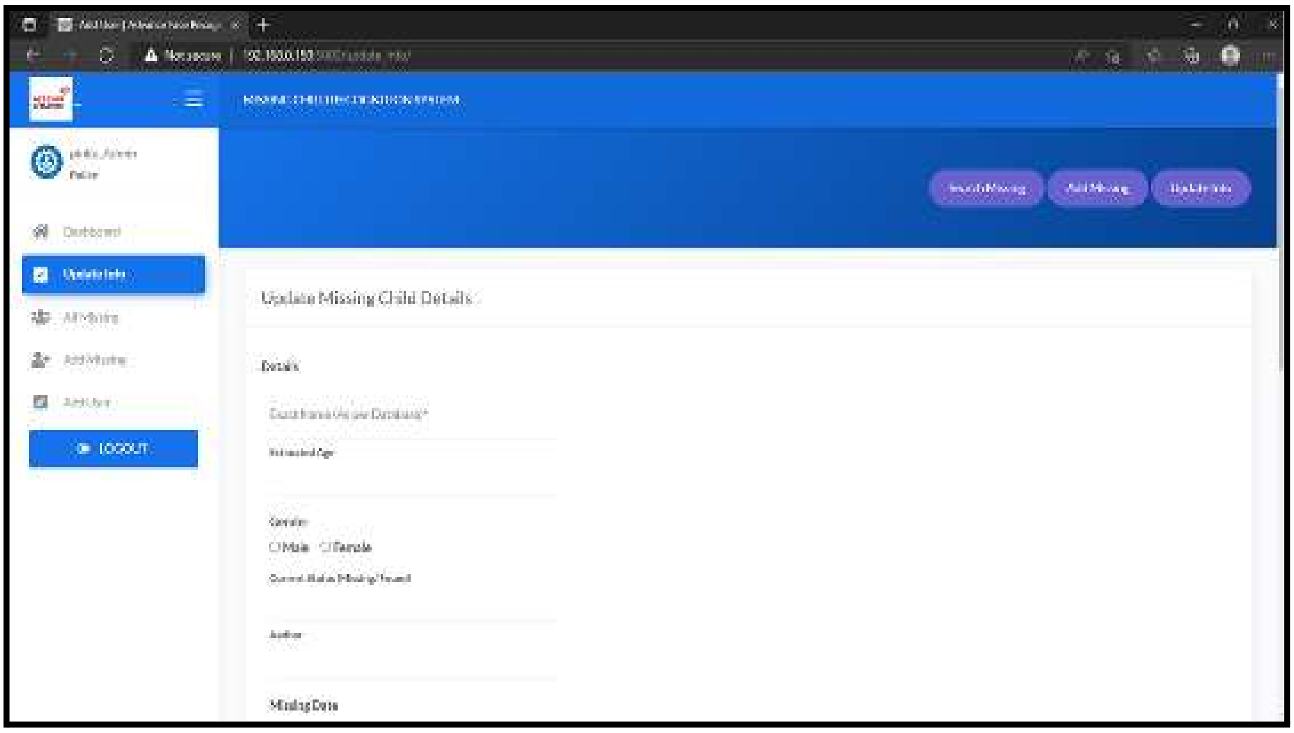
Username

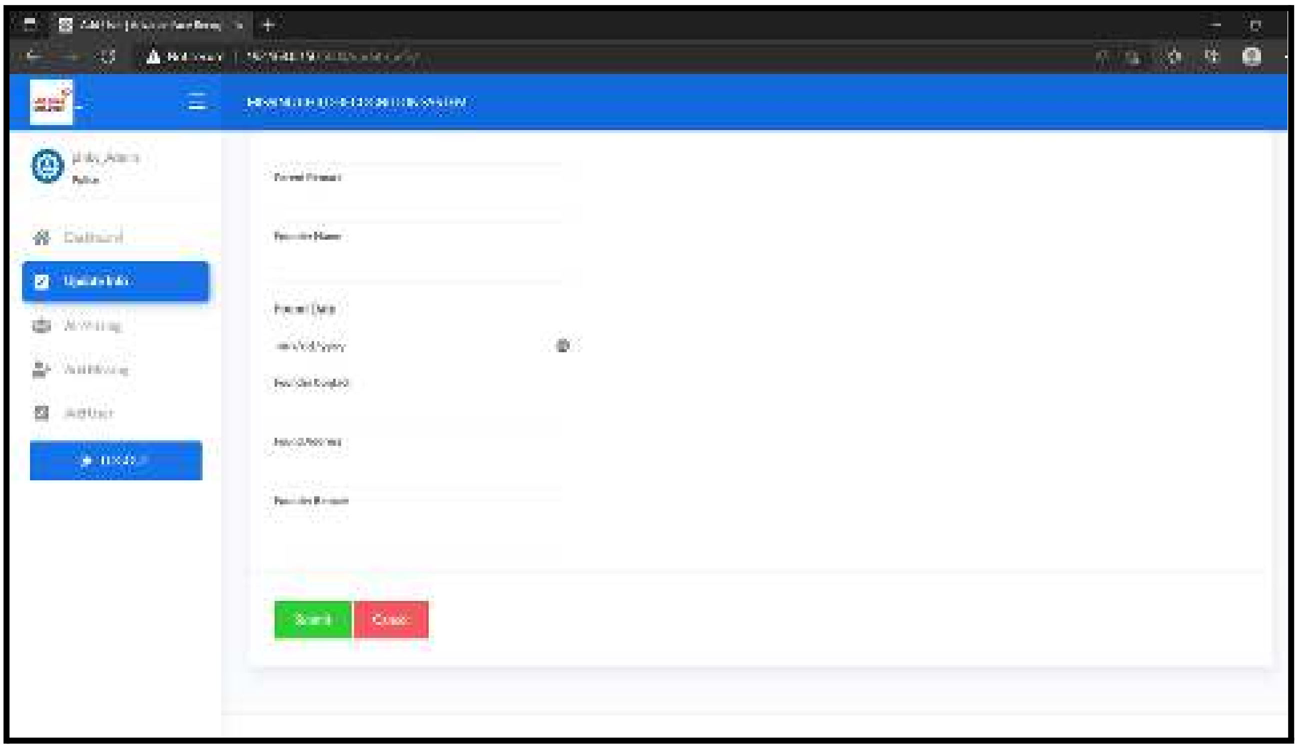
New Password (DEL to Delete User)

Update

Screenshot 5.8.2 : Police(Updating existing user password)

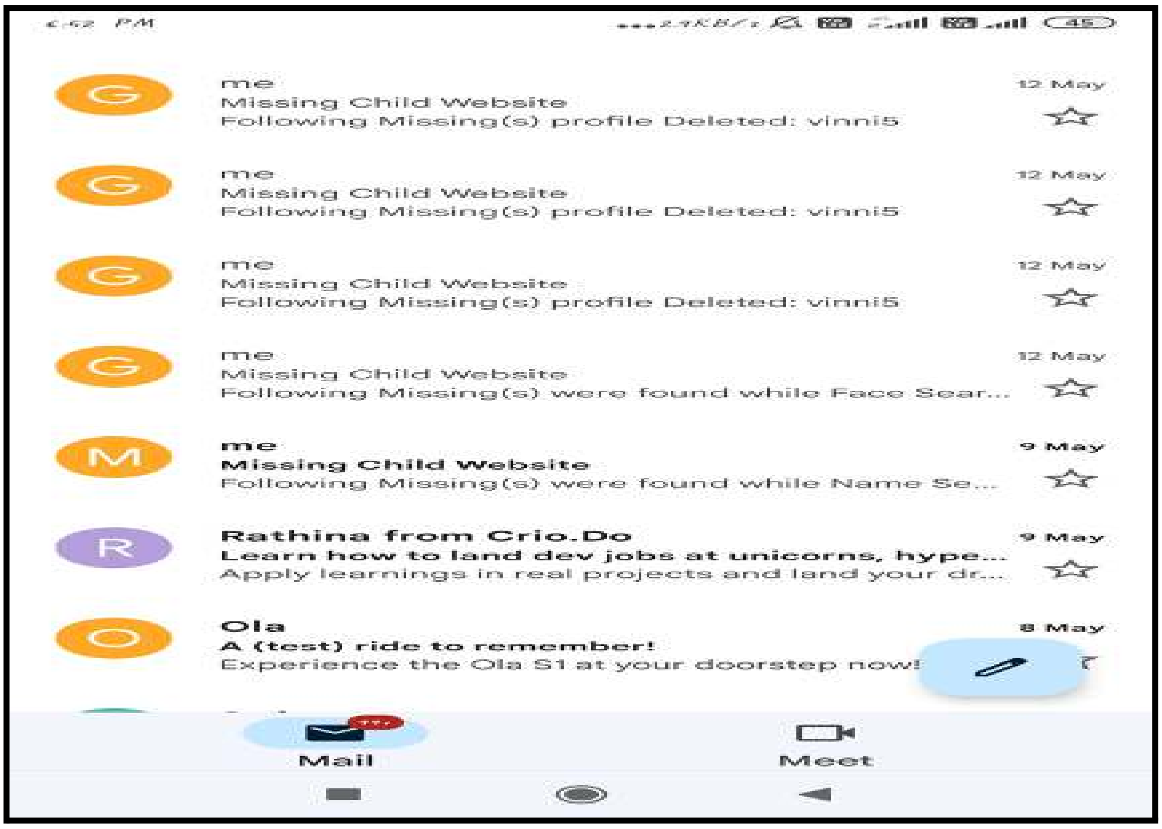
5.9 SCREENSHOT : POLICE (UPDATING MISSING CHILD DETAILS)





Screenshot 5.9 : Police (updating missing child details)

5.10 SCREENSHOT : EMAIL NOTIFICATION



6:52 PM

me — Missing Child Website — Following Missing(s) profile Deleted: vinni5 — 12 May

me — Missing Child Website — Following Missing(s) profile Deleted: vinni5 — 12 May

me — Missing Child Website — Following Missing(s) profile Deleted: vinni5 — 12 May

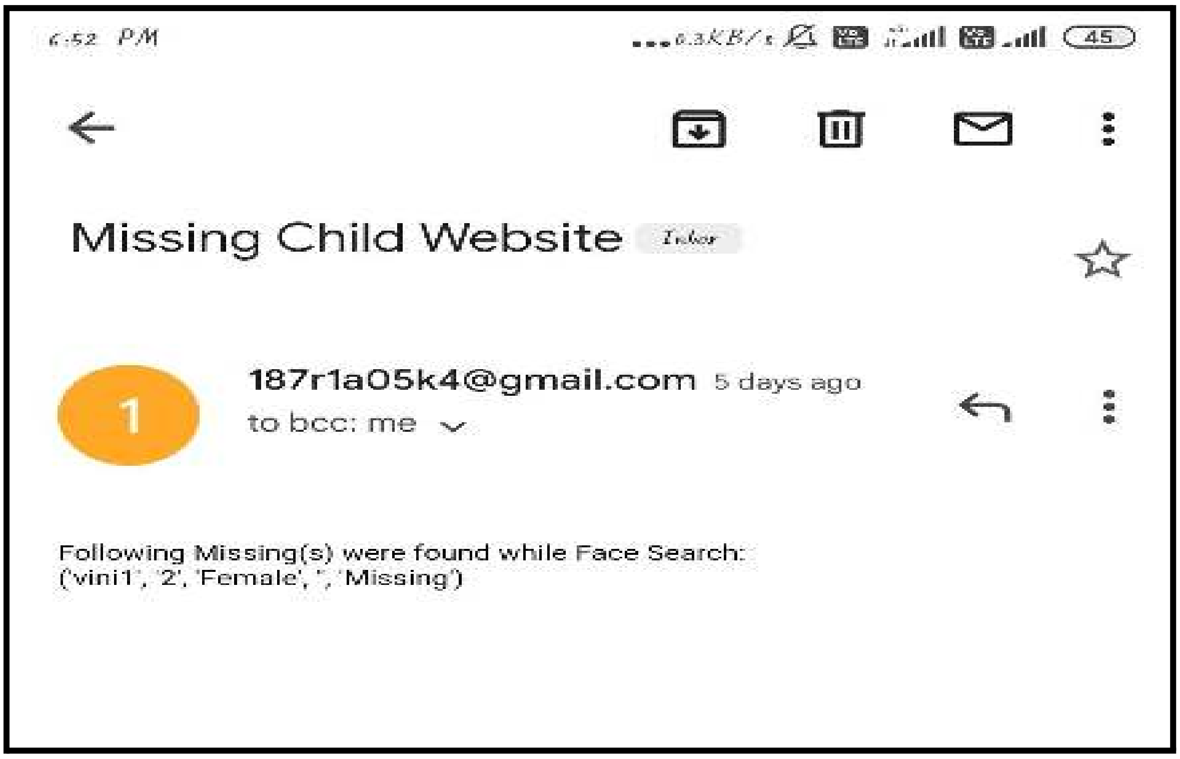
me — Missing Child Website — Following Missing(s) were found while Face Sear... — 12 May

me — Missing Child Website — Following Missing(s) were found while Name Se... — 9 May

Rathina from Crio.Do — Learn how to land dev jobs at unicorns, hype... — Apply learnings in real projects and land your dr... — 9 May

Ola — A (test) ride to remember! — Experience the Ola S1 at your doorstep now! — 8 May

Screenshot 5.10.1 :email notification



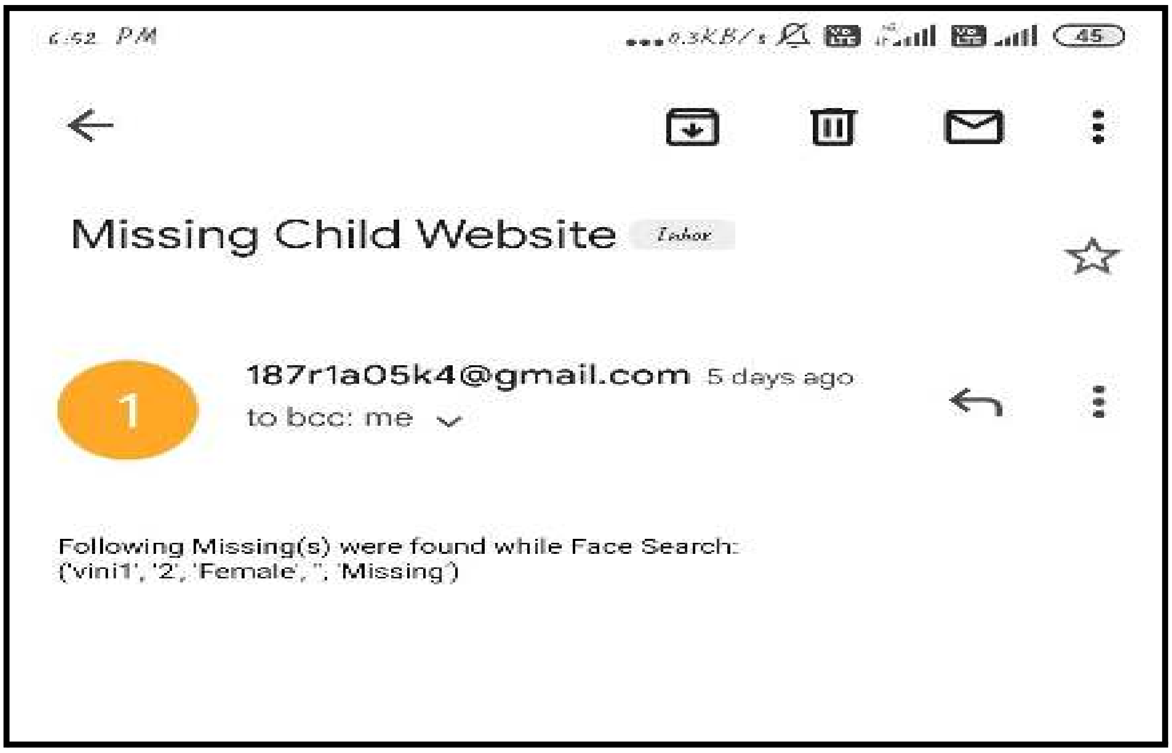
6:52 PM

Missing Child Website Inbox

187r1a05k4@gmail.com 5 days ago
to bcc: me

Following Missing(s) were found while Face Search:
('vini1', '2', 'Female', '', 'Missing')

Screenshot 5.10.2 : email notification for face search

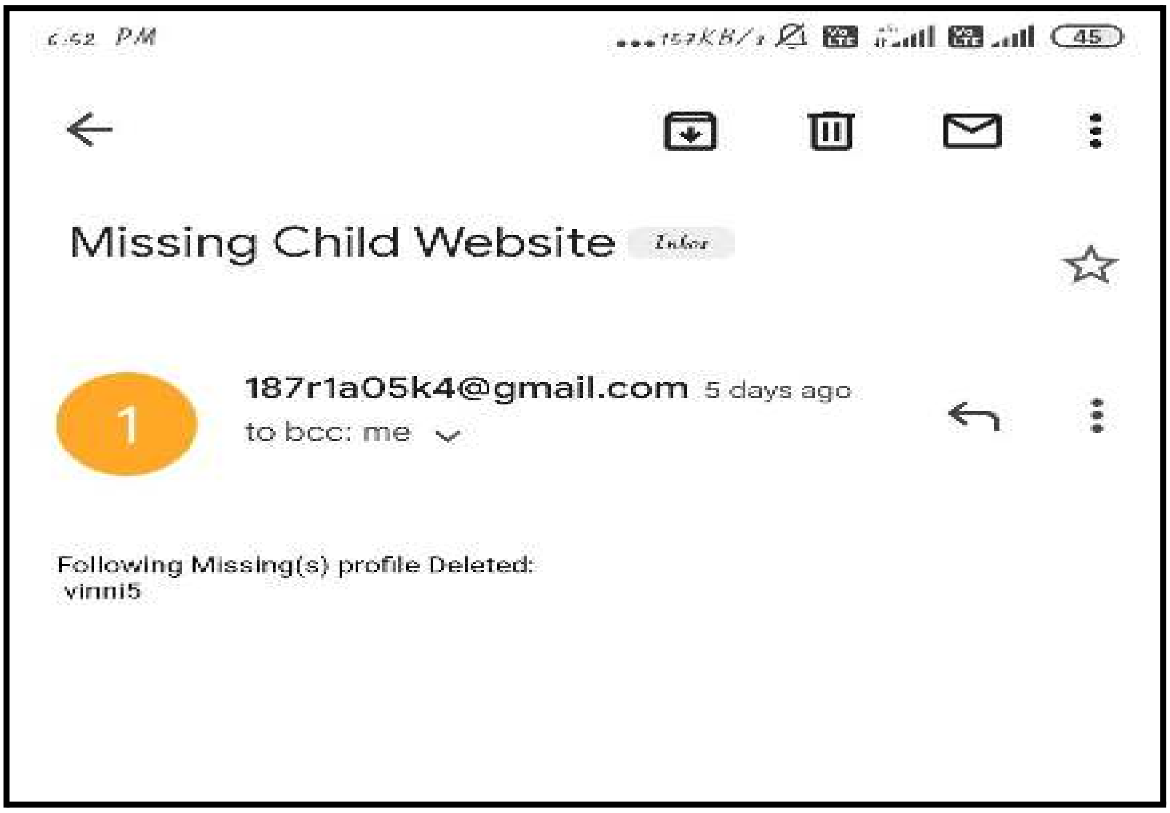

6:52 PM

Missing Child Website Inbox

187r1a05k4@gmail.com 5 days ago
to bcc: me

Following Missing(s) were found while Face Search:
('vini1', '2', 'Female', '', 'Missing')

Screenshot 5.10.3 : Email notification for name search


6:52 PM

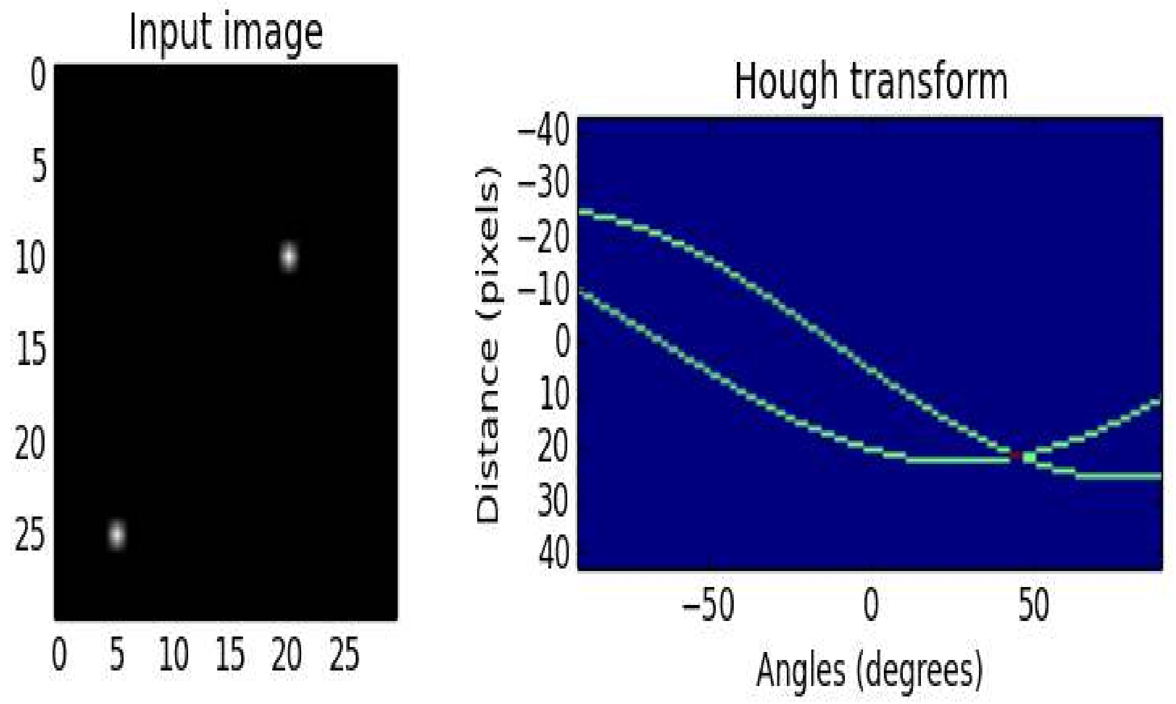
Missing Child Website Inbox

187r1a05k4@gmail.com 5 days ago
to bcc: me

Following Missing(s) profile Deleted:
vinni5

Screenshot 5.10.4 : Email notification for profile deletion

5.11 SCREENSHOT : GRAPH FOR IMAGE ANALYSIS



Screenshot 5.11 : Graph for image analysis

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover conceivable faults or weaknesses in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly ,and that program inputs produce valid outputs.All decision branches and internal code flow should be validated.It is the testing of individual software units of the application.It is done after the completion of an individual unit before integration.This is a structural testing,that relies on knowledge of its construction and is invasive.Unit tests perform basic tests at component level and test a specific business process,application,and/or system configuration.Unit tests ensure that each unique path of a business path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes.

6.3 TEST CASES

Test case ID	Test case name	Purpose	Actual Output	Validation
1	Adding a missing child	Used for adding a Missing child	The system ask for details and adds the missing child	Yes
2	Searching a missing child	Used for searching a missing child	System is able to take any details of missing child like name,address,face image or iris image	Yes

7.CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

A missing child identification system is proposed , which combines facial feature extraction based on deep learning and matching based on LBPH. We use the Gabor filter for iris detection.The classification achieved a higher accuracy of 90% which shows the proposed methodology of face recognition could be used for reliable missing children identification.

7.2 FUTURE SCOPE

In the future, when we upload an image of a missing child the iris and face recognition will be compared simultaneously.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", Nature, 521(7553):436–444, 2015.

[2] O. Deniz, G. Bueno, J. Salido, and F. D. la Torre, "Face recognition using histograms of oriented gradients", Pattern Recognition Letters, 32(12):1598–1603, 2011.

[3] C. Geng and X. Jiang, "Face recognition using sift features", IEEE International Conference on Image Processing(ICIP), 2009.

[4] Rohit Satle, Vishnuprasad Poojary, John Abraham, Shilpa Wakode, "Missing child identification using face recognition system", International Journal of Advanced Engineering and Innovative Technology (IJAEIT), Volume 3 Issue 1 July - August 2016.

[5] https://en.wikipedia.org/wiki/FindFace

[6]https://www.reuters.com/article/us-china-trafficking-apps/mobileapp-helps-china-recover-hundreds-of-missing-childrenidUSKBN15J0GU

[7] Simonyan, Karen and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition", International Conference on Learning Representations (ICLR), April 2015.

[8] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in British Machine Vision Conference, vol. 1, no. 3, pp. 1-12, 2015.

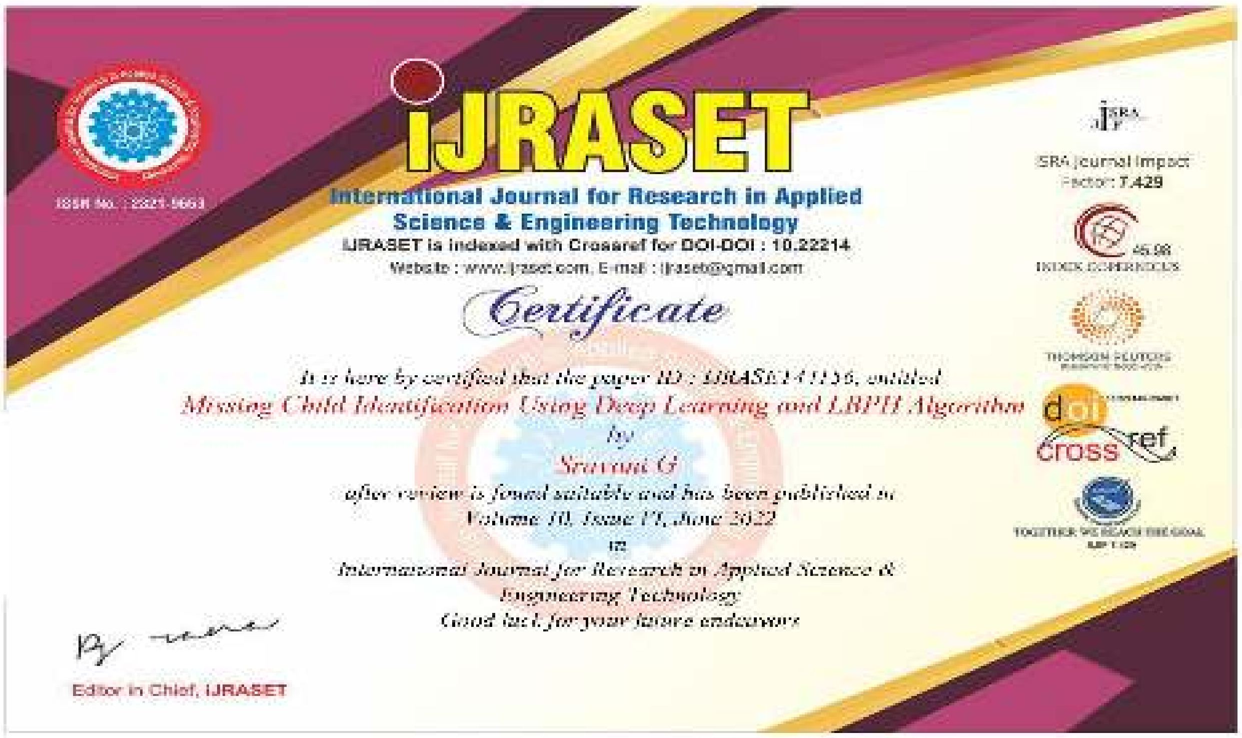
[9] A. Vedaldi, and K. Lenc, "MatConvNet: Convolutional Neural Networks for MATLAB", ACM International Conference on Multimedia, Brisbane, October 2015.

8.2 WEBSITES

https://www.google.com/search?sa=X&rlz=1C1CHBF_enIN981IN981&biw=1536&bih=746&q=missing+child+identification+system+%22website%22&ved=2ahUKEwjop7_Qj4n4AhUhTmwGHe3JBCUQ5t4CegQILBAB

8.3 GITHUB LINK

https://github.com/GouraboinaSravani/Missing-child-identification-using-deep-learning-and-lbph-algorithm.git



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

TOGETHER WE REACH THE GOAL

Certificate

It is here by certified that the paper ID : IJRASET44156, entitled

Missing Child Identification Using Deep Learning and LBPH Algorithm

by

Sravani G

after review is found suitable and has been published in

Volume 10, Issue VI, June 2022

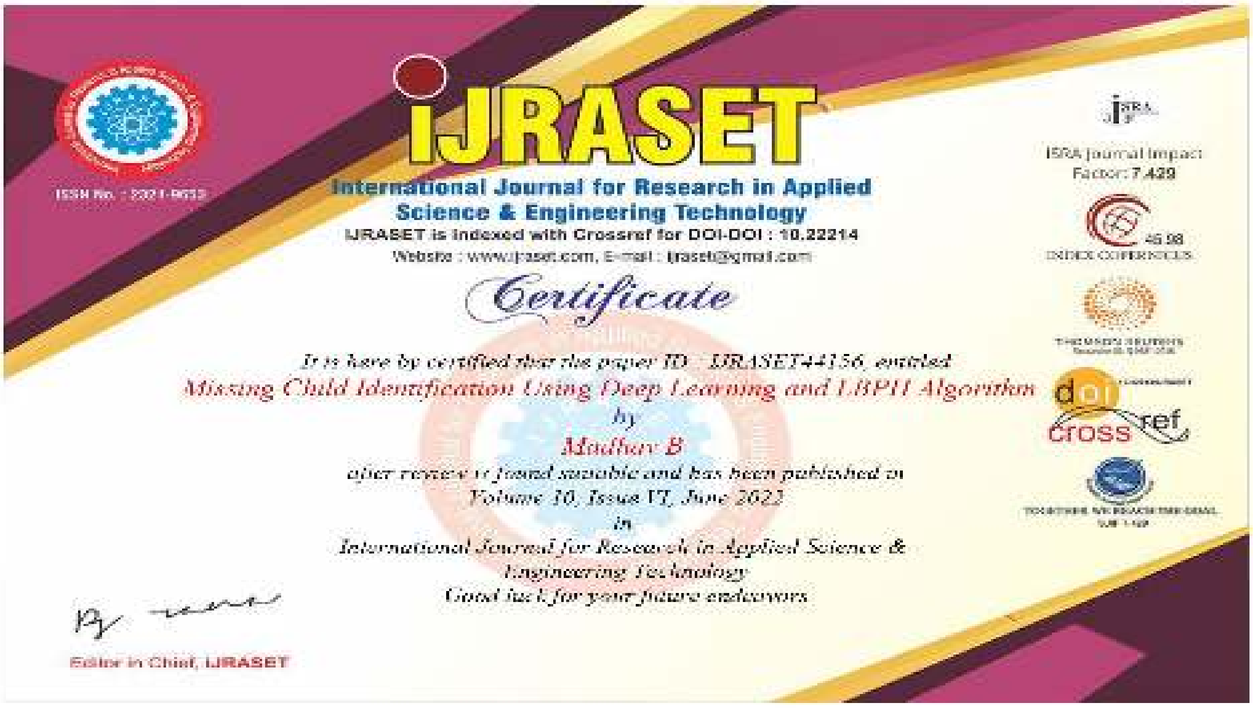
in

International Journal for Research in Applied Science & Engineering Technology

Good luck for your future endeavors

[illegible]

Editor in Chief, IJRASET



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

TOGETHER WE REACH THE GOAL

Certificate

It is here by certified that the paper ID : IJRASET44156, entitled

Missing Child Identification Using Deep Learning and LBPH Algorithm

by

Madhav B

after review is found suitable and has been published in

Volume 10, Issue VI, June 2022

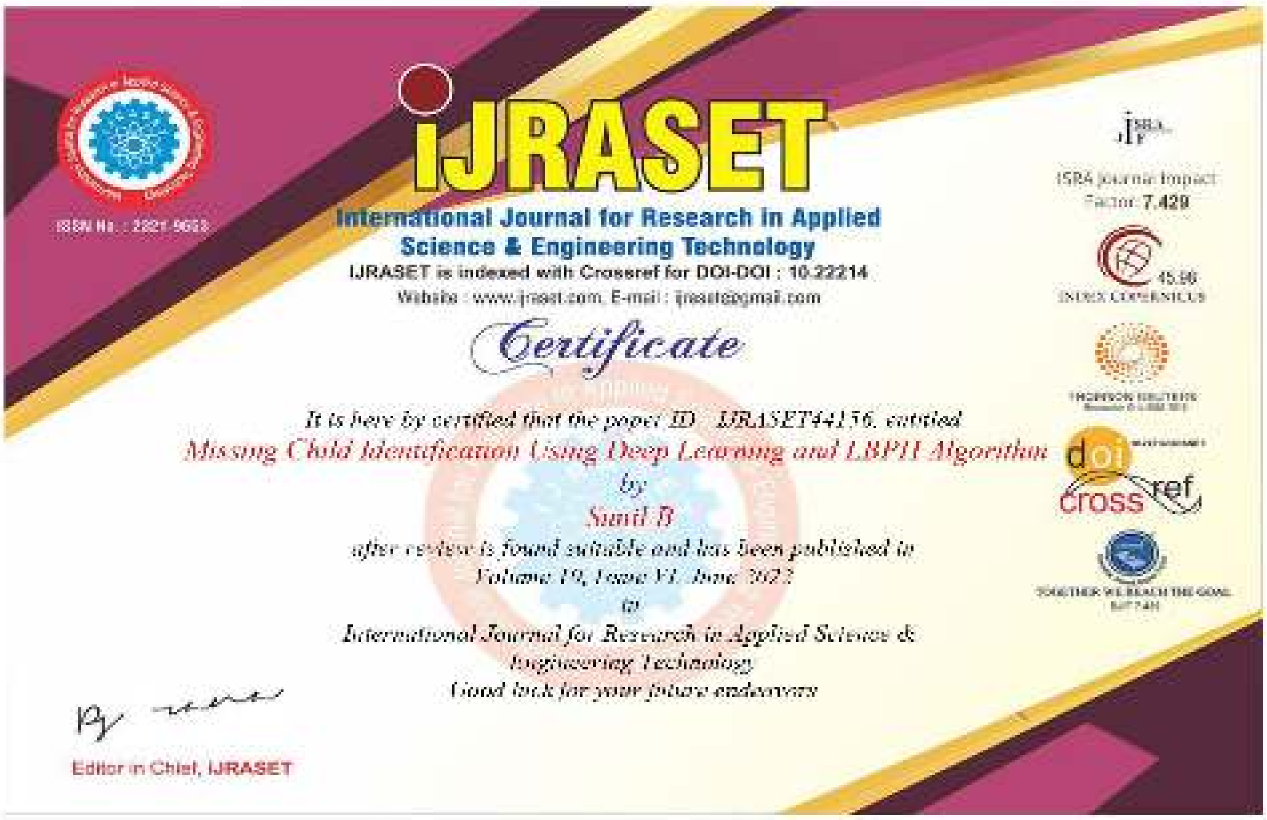
in

International Journal for Research in Applied Science & Engineering Technology

Good luck for your future endeavors

[illegible]

Editor in Chief, IJRASET



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

doi crossref

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET44156, entitled

Missing Child Identification Using Deep Learning and LBPH Algorithm

by

Sunil B

after review is found suitable and has been published in

Volume 10, Issue VI, June 2022

in

International Journal for Research in Applied Science & Engineering Technology

Good luck for your future endeavors

[illegible]

Editor in Chief, IJRASET