VIDEO SURVEILLANCE

A Main Project Report

submitted by

SAYUJ A P (MES20MCA-2046)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

July 2022

DECLARATION

I undersigned hereby declare that the project report **VIDEO SURVEILLANCE**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bona fide work done by me under supervision of Prof. Hyderali K, Associate Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: KUTTIPPURAM

Date: 06-07-2022

SAYUJ A P (MES20MCA-2046)

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **VIDEO SURVEILLANCE** is a bona fide record of the Main Project work carried out by **SAYUJ A P (MES20MCA-2046)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

At the very outset I would like to thank the Almighty's mercy towards me over the years...

I wish to express my sincere thanks to my project guide Ms. Priya J D Assistant professor, Dept. of Master of Computer Applications who guided me for the successful completeness of this project. I also thank her for valuable suggestions, guidance, constant encouragement, boundless corporation, constructive comments and motivation extended to me for completion of this project work.

I would express my sincere thanks to my internal guide Prof. Hyderali K Associate professor for his immense guidance to complete the project successfully.

I would like to express my sincere thanks to all the faculty members of Master of Computer Applications department for their support and valuable suggestion for doing the project work.

Last but not least my graceful thanks to my parents, friends and also the persons who supported me directly and indirectly during the project.

SAYUJ A P (MES20MCA-2046)

Abstract

This is an innovative approach to video surveillance software project. We normally find video cameras in museum and other organization that continuously record and save the recorded video footage for days or months. This utilizes a lot of battery life and storage capacity to store these large video footage. Well this video surveillance software is an enhanced version of organization security that continuously monitors but only records unusual changes in the organizations. This project is implemented in a Museum, so we can improve the security of objects collected in it. As soon as the system catches any unusual activity it takes steps and informs the admin by: Sending an notification to the admin about the unusual activity and Sending an image of the activity to the admin so that he may check the problem seriousness and react accordingly. The main advantage of the system is that it instantly alerts the user about any suspicious activity at the place, and requires much less or no storage space as compared to the traditional surveillance system.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

When we talk about security of any place, then Video Surveillance cameras are the most popular equipment for this purpose. It has gained immense popularity as an effective security measure. In the near future, most institutions and organizations will have surveillance cameras installed. Consequently, this will reduce burglary incidents and enhance social stability. Video Surveillance system targets to detect suspicious behaviour in a specific area using different motion detection techniques. The system is based on the open source image-processing library (OpenCV). And this project this system is implemented in a Museum.

1.1.1 Motivation

There have been many security related issues are reporting nowadays. However, a small percentage of the population is currently using smart surveillance cameras in order to secure their homes, offices, museums In this project, we develop a low cost smart security system based on wireless surveillance cameras.

1.2 Objective

Video Surveillance system is developed to provide the solution to secure our organizations.Through, this application admin who is one of the users of this application will manage all the activities

of users such as Add Items, Add Showcase, Manage Security, and View Alerts.

The main objective of this project is to investigate and develop an integrated video surveillance system which can convey filtered surveillance information, including alarms and images, to the admin. Once suspicious event is detected, the video surveillance system will instantaneously send user a notification alert indicates motion detection. Recording only if suspicious events will be an option in order to reduce the storage capacity.

This project provides a simple and beautiful interface for the admin as well as to the users. Just admin needs to add item details that can be view by the users. All the information will be stored in the database and that will be help to maintain all the information of users.

1.3 Report Organization

The project report is divided into five sections.

Section 2 describes literature survey.

Section 3 describes the methodology used for implementing the project.

Section 4 gives the results and discussions.

Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

Intelligent visual surveillance systems deal with the real-time monitoring of persistent and transient objects within a specific environment. The primary aims of these systems are to provide an automatic interpretation of scenes and to understand and predict the actions and interactions of the observed objects based on the information acquired by sensors.The main stages of processing in an intelligent visual surveillance system are moving object detection and recognition, tracking, behavioural analysis and retrieval. These stages involve the topics of machine vision, pattern analysis,artificial intelligence and data management.The recent interest for surveillance in public, military and commercial scenarios is increasing the need to create and deploy intelligent or automated visual surveillance systems. In scenarios such as public transport, these systems can help monitor and store situations of interest involving the public, viewed both as individuals and as crowds. Current research in these automated visual surveillance systems tends to combine multiple disciplines such as those mentioned earlier with signal processing,telecommunications, management and socio-ethical studies. Never the less there tends to be a lack of contributions from the field of system engineering to the research.The growing research interest in this field is exemplified by the IEEE and IEEE workshops and conferences on visual surveillance and special issues that focus solely on visual surveillance in journals like [7–9] or in human motion analysis like in. This chapter surveys the work on automated surveillance systems from the aspects of

* image processing/computer vision algorithms which are currently used for visual surveillance;

* surveillance systems:different approaches to the integration of the different vision algorithms to build a completed surveillance system;

* distribution, communication and system design: discussion of how such methods need to be integrated into large systems to mirror the needs of practical CCTV installations in the future.

Chapter 3

Methodology

3.1 Introduction

This is an innovative approach to video surveillance software project. We normally find video cameras in museum and other organization that continuously record and save the recorded video footage for days or months. This utilizes a lot of battery life and storage capacity to store these large video footage. Well this video surveillance software is an enhanced version of organization security that continuously monitors but only records unusual changes in the organization. This project is implemented in a Museum, so we can improve the security of objects collected in it. As soon as the system catches any unusual activity it takes steps and informs the admin by: Sending an notification to the admin about an unusual activity and Sending an image of the activity to the admin so that he may check the problem seriousness and react accordingly.The main advantage of the system is that it instantly alert about any suspicious activity at the place, and requires much less or no storage space as compared to the traditional surveillance system.

The entire project is divided into two parts :

1. Web part
2. Technical part

In the web part it consist of a web application.

In the technical part its an Desktop application works on Open CV library.

Video Surveillance system targets to detect suspicious behavior in a specific area using different motion detection techniques. The system is based on the open source image-processing library (Open CV). Once suspicious event is detected, the Video surveillance system will instantaneously send user a notification alert indicates motion detection. Recording only of suspicious events will be an option in order to reduce the required storage capacity.

Open CV:- is a open source library of programming functions for image processing and performing computer vision tasks.

3.2 Modules

1.ADMIN

* Login
* Showcase Add
* Items Add
* Manage Security
* View Alerts
* View complaints
* Post Reply

2.USER

* Register
* Login
* View Items
* Post Complaint
* View Reply

3.SECURITY

* Start Camera
* Capture Frame

* Process Image

* Detect Intruder

* Generate Alerts

3.3 Background Subtraction Algorithm

Background Subtraction Algorithm:-

The usual assumption is that the images of the scene without the intruding objects exhibit some regular behavior that could be well described by a statistical model. If we have a statistical model of the scene, an intruding object can be detected by spotting the parts of the image that don't fit the model. This process is usually known as "background subtraction". The idea behind motion detection techniques is to first build a background model from a sequence of images in order to find the objects of interest from the difference between that background estimation and the current frame. Therefore, the accuracy of the segmentation process depends on how well the background is modeled. This simple approach will compare between the current frame and the background (static first frame as assumed), then we will compute the absolute difference; which is defined as Delta. The difference between the current frame and background frame(first frame); is computed as the following equation:

Delta = Current frame - Background frame

Finally, Applying threshold of difference (Delta) will identify whether there is motion detected or not.

3.4 Steps in Background Substraction Algorithm

Steps involving Background Subtraction Algorithm :

1. Collect the images.
2. Pre-process the image (converted into binary code).
3. Overlays the images.
4. Subtract the pixel values.
5. Obtain Delta (Difference between Current frame and Background frame).
6. If Delta greaterthan Threshold.
7. Generate alerts.

3.5 Flowchart

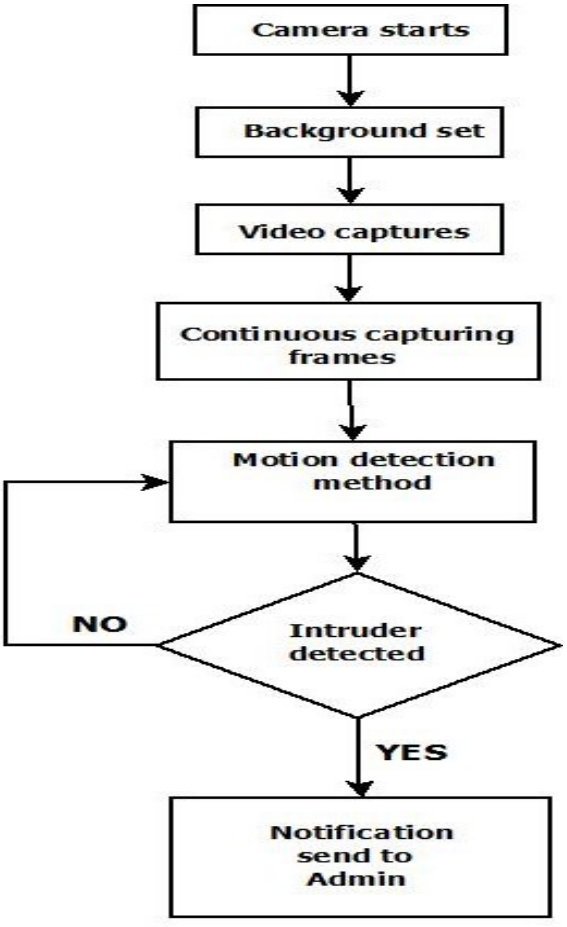

Figure 3.1: functions of this project

3.6 Working

In this project I am developing a web and desktop application for the surveillance system. And this project is implemented in a Museum. There are 3 sections in this project as follows

1) Admin

2)User

3)Security

The first module is Admin. Admin has the control of the overall system and it done by web.Admin can can directly login to the system, and he can add items to each showcase. Admin can give a description about the items stored in the Museum. The Admin who can track all the activities of this system. If any intruder is detected in the museum the system will generate a alert and send the detected image of the intruder to the admin.

The second module is User, it is also a web application and users can login by username and password. They can see the items stored in the museum and also users get an awareness of how valuable these things. Users can also give the complaints to the admin and admin can give reply to the user.

The third module is Security, it is developed as a desktop application. And it is a fully automated system. It uses The Background subtraction algorithm. In this module the first, camera will be start and set a background image then it capture the foreground activities. This two will be converted into frames. If any motion detected the system will produce a alarm and send a notification to admin. This module is completed with the open CV library.

3.7 Developing Environment

* Language: Python

* Front End: Html, CSS, Java script

* Back end: Python-django, Anaconda

* Database : MySQL

* IDE: PyCharm

* OS: Windows/Linux

3.8 Agile Methodology

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins,teams cycle through a process of planning,executing and evaluating. Continuous collaboration is vital.

3.8.1 User Story

A user story is a tool used in agile software development to capture a description of a software feature from an end-user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work drives collaboration, creativity, and a better product overall.There are 16 user stories.The user story is given below :

User StoryID	**As a <type of user>**	**I want to**	**So that I can**
1	Admin	Login	Login successful with correct username and password
2	Admin	Adding showcase details	Admin can list the showcase details
3	Admin	Add items	Admin can add items to the showcase
4	User	Registration	User can register to the system and create a profile
5	User	Login	Login and access the system
6	User	View showcase	View the showcase details
7	User	View items	View the items listed in the showcase
8	User	Post complaints	User post complaints
9	Admin	View complaints	View complaints and post response
10	User	View response	View response post by the Admin
11	Admin	Set background	Set background image
12	Security	Capture the videos	Capture the videos
13	Security	Frames	Videos converted into frames
14	Security	Processing images	Processing images
15	Security	Detect intruder	Detect intruder
16	Security	Alert	Generate alerts

Table 3.1: User Story

3.8.2 Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.The priority of each user stories are given in the product backlog.

User Story ID	**Priority <High/ Medium/ Low>**	**Size (Hours)**	**Sprint <#>**	**Status <Planned/ In progress/ Completed**	**Release Date**	**Release Goal**
1	Medium	3	1	Completed	21/04/2022	Login
2	High	4		Completed	26/04/2022	Add showcase details
3	Medium	3	2	Completed	29/04/2022	Add items to showcase
4	High	4		Completed	30/04/2022	User registration
5	Medium	2		Completed	03/05/2022	Login
6	Medium	3		Completed	11/05/2022	View showcase
7	Medium	2		Completed	23/05/2022	View items
8	Medium	3	3	Completed	27/05/2022	Post complaints
9	Medium	2		Completed	31/05/2022	post response
10	Medium	4		Completed	03/06/2022	View response
11	Medium	3	4	Completed	17/06/2022	Set background
12	Medium	4		Completed	21/06/2022	Capture the videos
13	High	2		Completed	22/06/2022	Converted into frames
14	High	3		Completed	27/06/2022	Processing images
15	High	3		Completed	06/07/2022	Detect intruder
16	High	3		Completed	06/07/2022	Generate alert

Table 3.2: Product Backlog

3.8.3 Project Plan

A project plan that has a series of tasks laid out for the entire project, listing task durations, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it.The project plan shows when each sprint start

User StoryID	**Task Name**	**Start Date**	**End Date**	**Days**	**Status**
1	Sprint 1	20/04/2022	21/04/2022	2	Completed
2		25/04/2022	26/04/2022	2	Completed
3	Sprint 2	29/04/2022	29/04/2022	1	Completed
4		30/04/2022	30/04/2022	1	Completed
5		02/05/2022	03/05/2022	2	Completed
6		09/05/2022	11/05/2022	3	Completed
7		20/05/2022	23/05/2022	4	Completed
8	Sprint 3	26/05/2022	27/05/2022	2	Completed
9		30/05/2022	31/05/2022	2	Completed
10		02/06/2022	03/06/2022	2	Completed
11	Sprint 4	17/06/2022	17/06/2022	1	Completed
12		20/06/2022	21/06/2022	2	Completed
13		22/06/2022	22/06/2022	1	Completed
14		27/06/2022	27/06/2022	1	Completed
15		06/07/2022	06/07/2022	1	Completed
16		06/07/2022	06/07/2022	1	Completed

Table 3.3: Project Plan

3.8.4 Sprint Plan

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story.Here the sprint backlog this project is given below :

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
UserStory #1,#2																
UI Designing	21/04/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Database Connectivity	21/04/2022	3	0	0	0	0	2	0	0	1	0	0	0	0	0	0
Coding	26/04/2022	2	0	0	0	0	1	0	0	0	1	0	0	0	0	0
Testing	26/04/2022	3	0	0	1	1	1	0	0	0	0	0	0	0	0	0
UserStory #3,#4,#5 #6,#7																
UI Designing	29/04/2022	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
Database Connectivity	03/05/2022	3	0	0	0	1	1	0	1	0	0	0	0	0	0	0
Coding	11/05/2022	3	0	0	0	1	0	0	0	1	0	0	0	1	0	0
Testing	23/05/2022	3	0	0	2	1	1	0	0	0	0	0	0	0	0	1
UserStory #8,#9#10																
UI Designing	27/05/2022	4	1	1	1	0	1	0	0	0	0	0	0	0	0	0
Database Connectivity	27/05/2022	4	0	0	0	0	0	1	1	0	0	0	0	0	1	1
Coding	03/06/2022	5	1	0	0	1	0	1	0	0	0	1	0	0	0	1
Testing	23/05/2022	3	0	0	1	0	0	0	2	2	0	0	0	0	0	0
UserStory #11,#12#13 #14,#15,#16																
UI Designing	17/06/2022	2	1	0	0	0	1	0	0	0	0	0	0	0	0	0
Database Connectivity	22/06/2022	3	1	0	0	1	0	0	1	0	0	0	0	0	0	0
Coding	06/07/2022	2	0	1	0	1	1	0	0	0	0	0	0	0	0	0
Testing	06/07/2022	3	0	0	1	0	0	1	1	0	0	0	0	0	0	0
Total		50	7	4	6	6	8	3	3	6	4	1	0	2	1	2

Table 3.4: Sprint Plan

3.8.5 Sprint Actuals

The sprint actual is done based on the sprint plan. The sprint actual is divided into 4 sprints in which each of the tasks specified in the sprint backlog are done based on each sprint.Each sprint needs to be completed based on the dates in the sprint backlog.

Backlog Item	Status & completion date	Original estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Completed Y or N
UserStory #1,#2																	
UI Designing	21/04/2022	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	Y
Database Connectivity	21/04/2022	3	0	0	0	0	2	0	0	1	0	0	0	0	0	0	Y
Coding	26/04/2022	2	0	0	0	0	1	0	0	0	1	0	0	0	0	0	Y
Testing	26/04/2022	3	0	0	1	1	1	0	0	0	0	0	0	0	0	0	Y
UserStory #3,#4,#5 #6,#7																	
UI Designing	29/04/2022	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	Y
Database Connectivity	03/05/2022	3	0	0	0	1	1	0	1	0	0	0	0	0	0	0	Y
Coding	11/05/2022	3	0	0	0	1	0	0	0	1	0	0	0	1	0	0	Y
Testing	23/05/2022	3	0	0	2	1	1	0	0	0	0	0	0	0	0	1	Y
UserStory #8,#9#10																	
UI Designing	27/05/2022	4	1	1	1	0	1	0	0	0	0	0	0	0	0	0	Y
Database Connectivity	27/05/2022	4	0	0	0	0	0	1	1	0	0	0	0	0	1	1	Y
Coding	03/06/2022	5	1	0	0	1	0	1	0	0	0	1	0	0	0	1	Y
Testing	23/05/2022	3	0	0	1	0	0	0	2	2	0	0	0	0	0	0	Y
UserStory #11,#12#13 #14,#15,#16																	
UI Designing	17/06/2022	2	1	0	0	0	1	0	0	0	0	0	0	0	0	0	Y
Database Connectivity	22/06/2022	3	1	0	0	1	0	0	1	0	0	0	0	0	0	0	Y
Coding	06/07/2022	2	0	1	0	1	1	0	0	0	0	0	0	0	0	0	Y
Testing	06/07/2022	3	0	0	1	0	0	1	1	0	0	0	0	0	0	0	Y
Total		50	7	4	6	6	8	3	3	6	4	1	0	2	1	2	

Table 3.5: Sprint Actual

Chapter 4

Results and Discussions

4.1 Results

This is an innovative approach to video surveillance software project. Well this video surveillance software is an enhanced version of organization security that continuously monitors but only records unusual changes in the organization. So this is usefull for any organizations so we can improve our security.

1. Home page of this web application

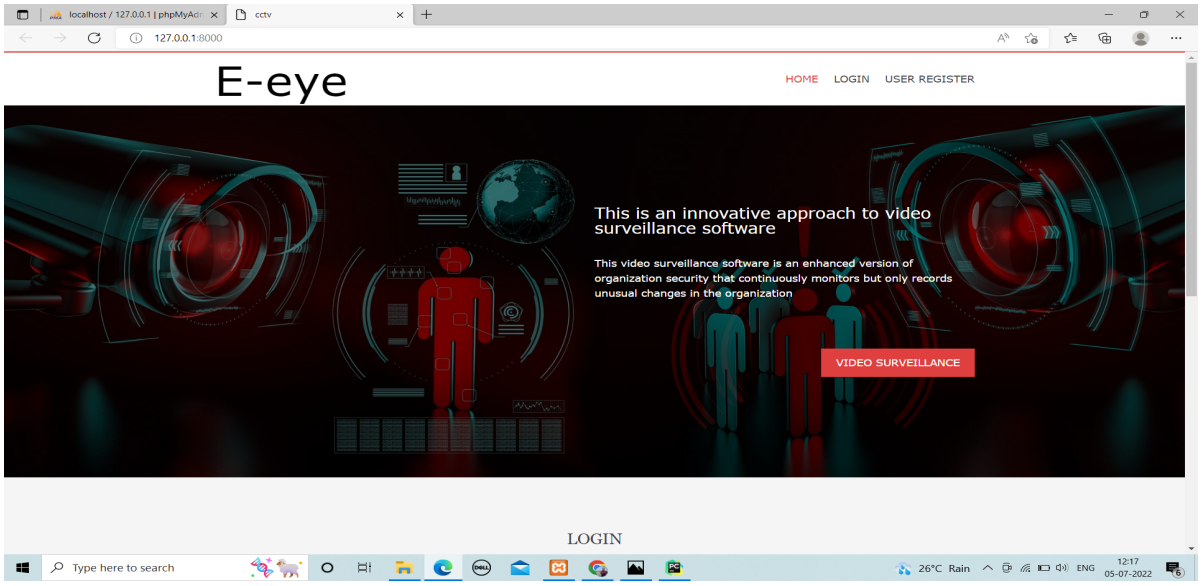

Figure 4.1: Home page

2. The Admin can first login in the web after adding their details in the database.

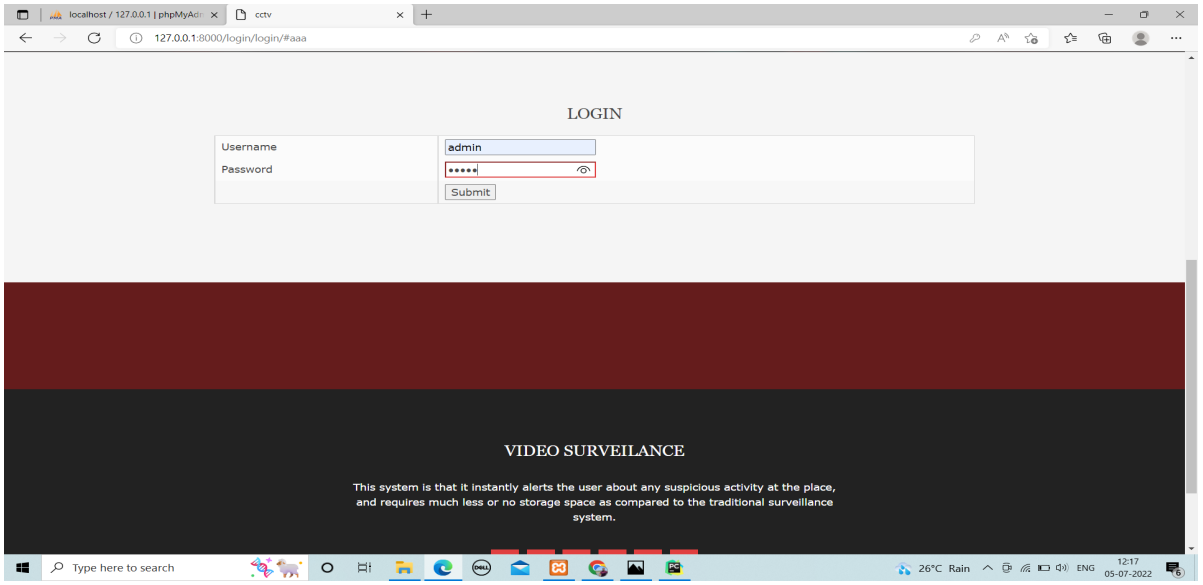

Figure 4.2: Admin login

3. After the Admin logins in they go to the home page.

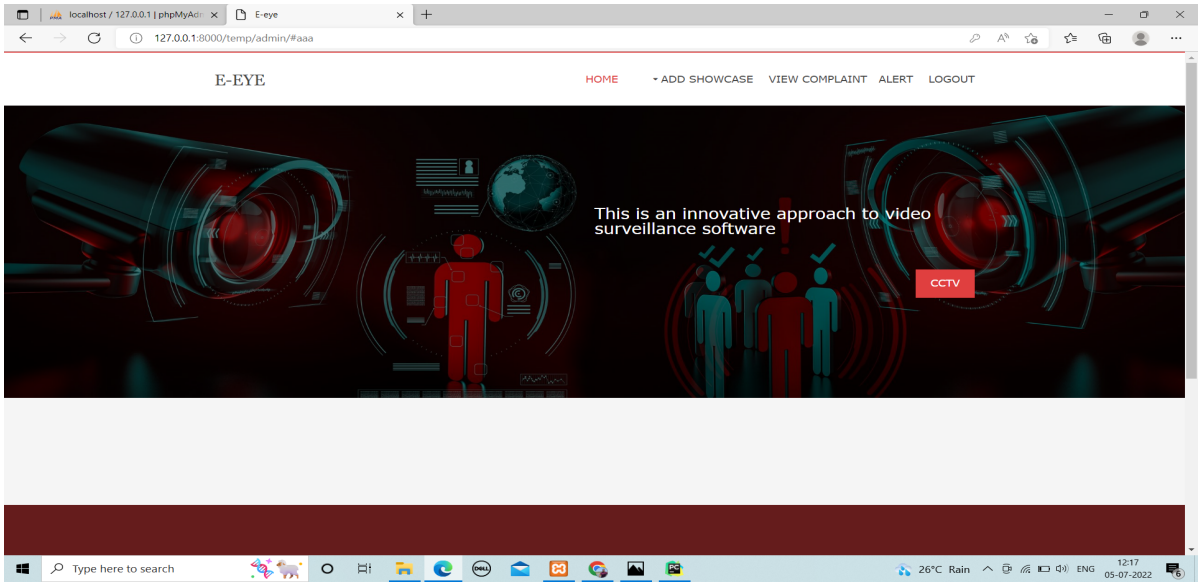

Figure 4.3: Admin home page

4. Admin can add items.

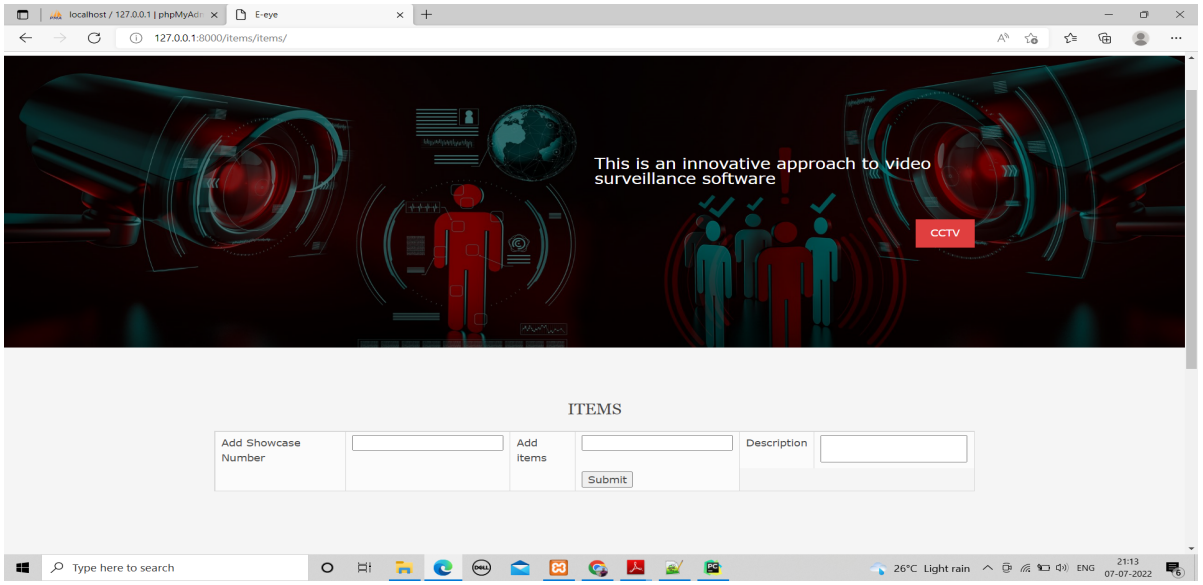

Figure 4.4: Adding items detils

5. The User can register here if it is the first time they are using.

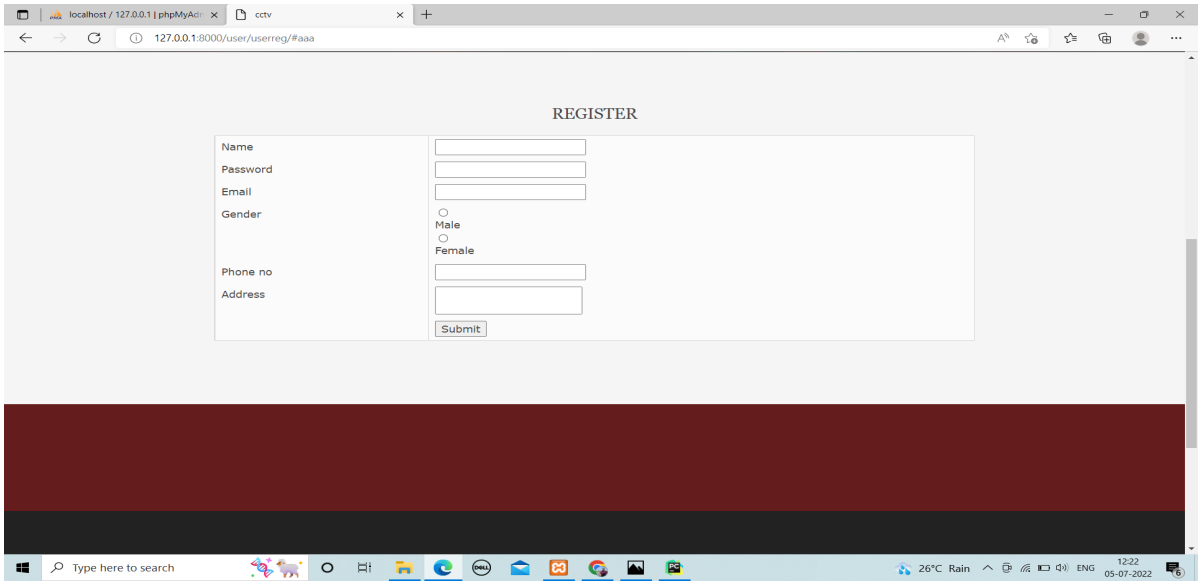

Figure 4.5: User registration

6. The User can login here.

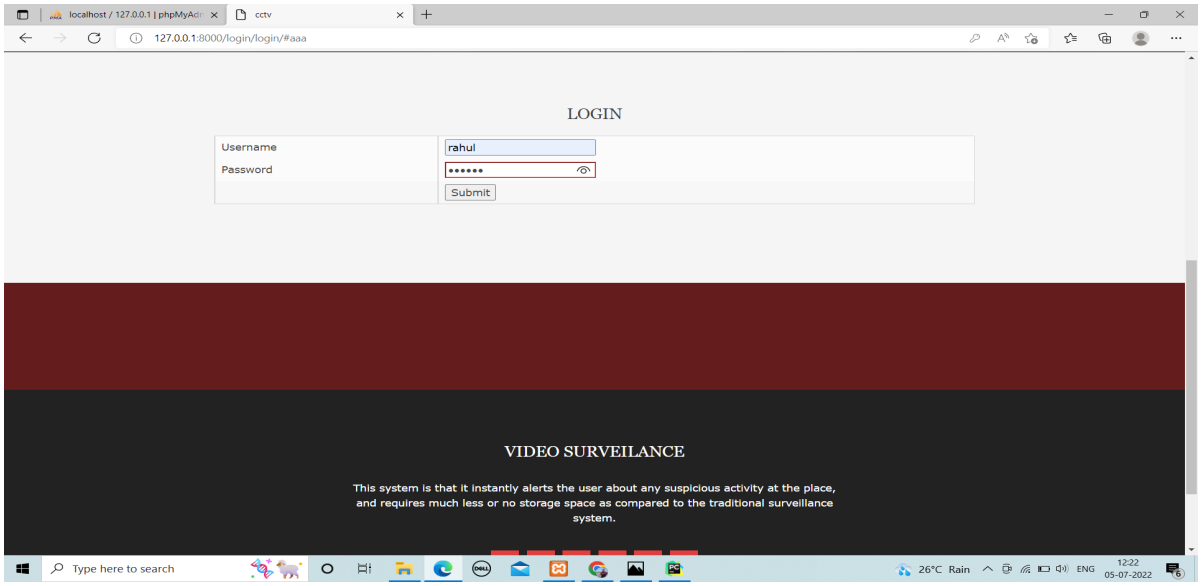

Figure 4.6: User login

7. The home page of user.

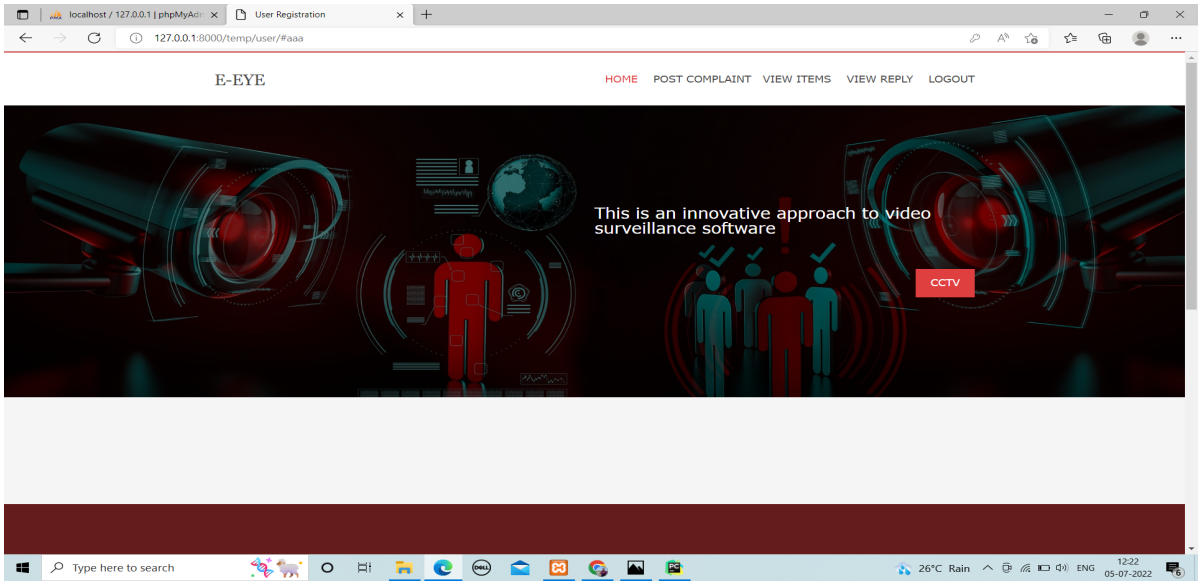

Figure 4.7: Home page of user

8. The user can view the items stored in the museum.

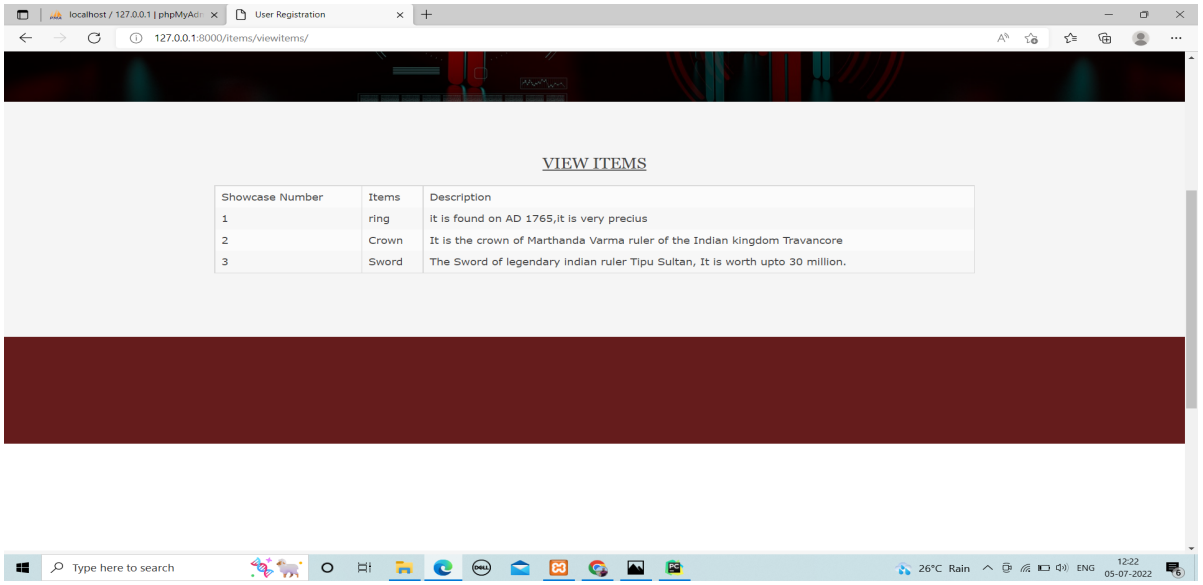

Figure 4.8: View items

9. Image detection.

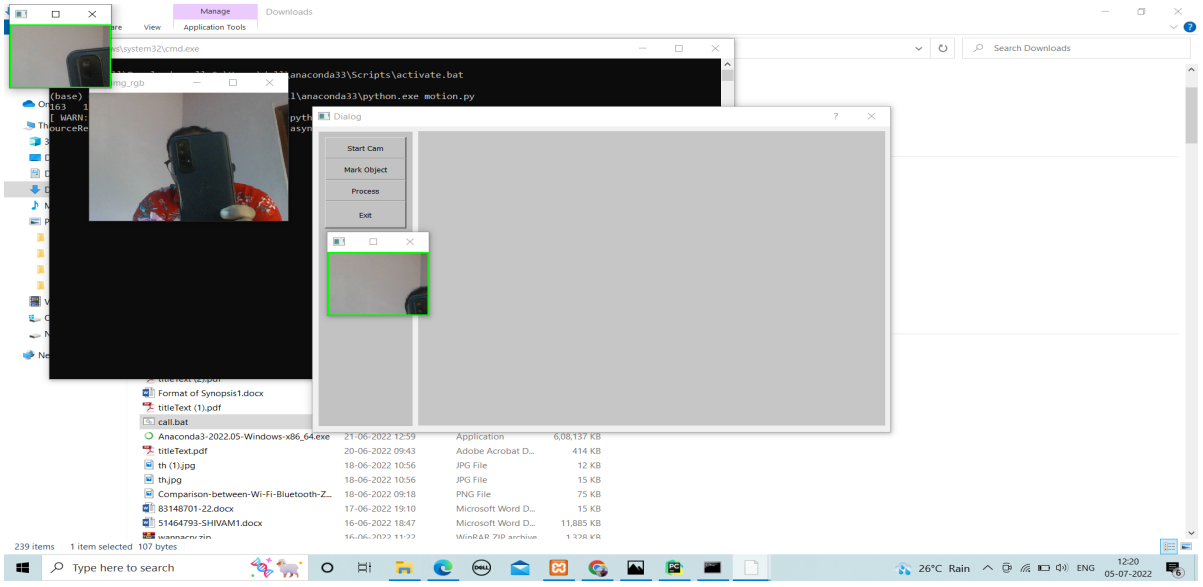

Figure 4.9: Image detection

10. Alert generated.

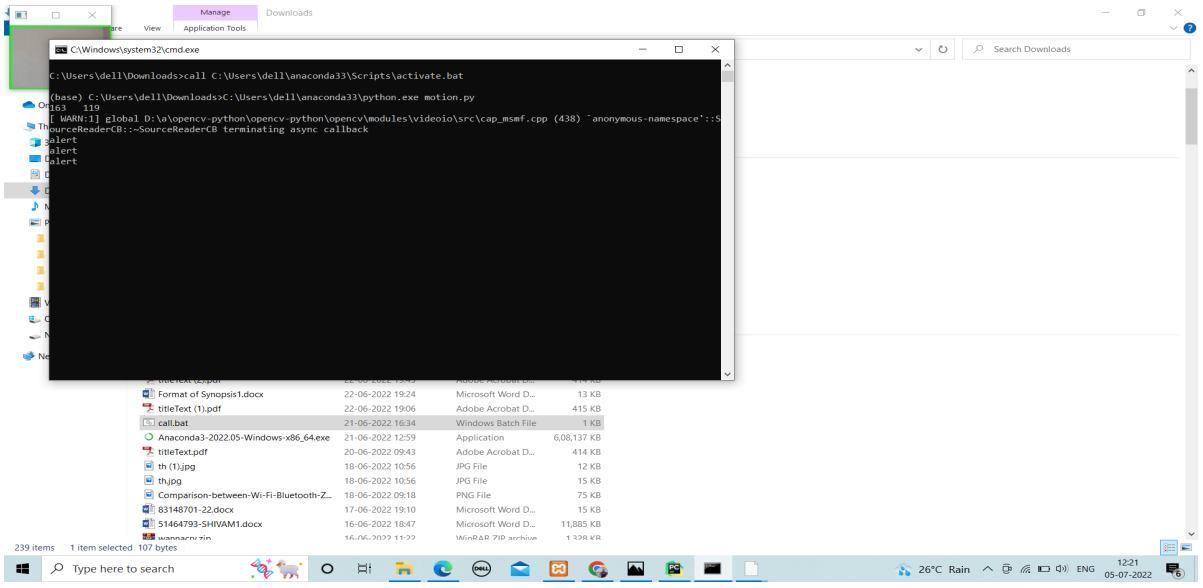

Figure 4.10: Alert generated.

11. The detected image is send to the admin.

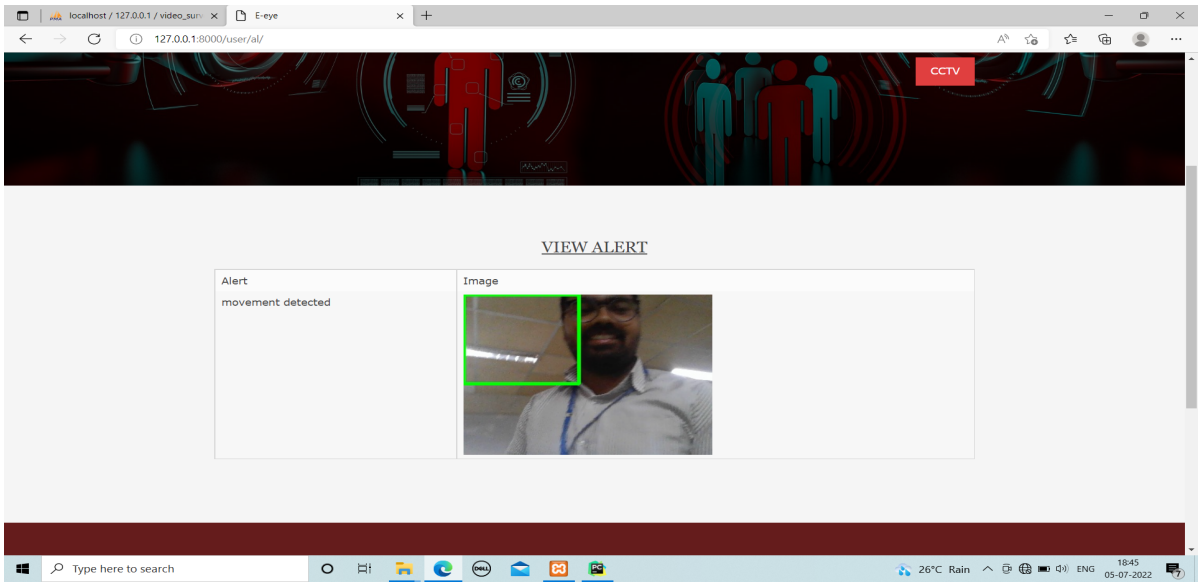

Figure 4.11: Detected image

4.2 Discussions

In future We can use different techniques for motion detection, this will improve our security. And also add feature of Face recognition; this feature will learn the surveillance system the faces of authorised people who always exist in the place to exclude them from motion detection function. In this project we use the OpenCV computer vision techniques, we can implement many different applications such as:

* Object recognition (shape detection).

* Measuring size of objects in meters relative to reference object.

* Object Tracking.

* Reality interactive games depends on cameras

Chapter 5

Conclusions

At the end, we conclude that Video Surveillance systems are very important in our life. Using the surveillance cameras can solve many security related issues in life. In this project we used Background subtraction techniques are the core idea of motion detection process. As we can perfectly compute/estimate the background model, we will accurately separate the foreground from background and find out the foreground mask. We used python programming language and OpenCV library to implement the video surveillance system. The system is easy to use, makes communication easier, reduces the time consumption and paper work. All necessary validations are carried out in this project where ever required and hence is a reliable system. The project is developed using PyCharm as front end and MySQL, as back end.The project has been developed, tested, documented and implemented successfully. This has been developed as versatile and user friendly as possible.At the final stage of this project with a proud feeling that some thing new had developed.

References

[1] https://PyCharm.com/tutorials/flask.

[2] http://stackoverflow.com/questions/23806738/opencv-background-subtraction-how-to-precompute-background-model?rq=1

[3] https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html

[4] https://docs.python.org/2/library/socket.html

[5] https://stackoverflow.com/questions/3044580/multiprocessing-vs-threading-python

Appendix

Source Code

```
"""videosurveilance URL Configuration

The 'urlpatterns' list routes URLs to views. For more information please see:
   https://docs.djangoproject.com/en/3.2/topics/http/urls/
Examples:
Function views
   1. Add an import: from my_app import views
   2. Add a URL to urlpatterns: path('', views.home, name='home')
Class-based views
   1. Add an import: from other_app.views import Home
   2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
Including another URLconf
   1. Import the include() function: from django.urls import include, path
   2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from django.conf.urls import url,include
from login import views
urlpatterns = [
   path('admin/', admin.site.urls),
   url('^login/',include('login.url')),
   url('^user/',include('user.url')),
   url('^items/',include('items.url')),
   url('^main/',include('main.url')),
   url('complaint/',include('complaint.url')),
   url('temp/',include('temp.url')),
   url('$',views.login)
]
from django.http import HttpResponseRedirect
from django.shortcuts import render
from login.models import Login



# coding: utf-8

# In[1]:


# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'facecnt.ui'
#
```

Appendix

```
# Created by: PyQt5 UI code generator 5.9.2
#
# WARNING! All changes made in this file will be lost!

from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtGui import QPixmap
import os
import math
import time
import cv2
import pymysql

from skimage.metrics import structural_similarity as ssim

from datetime import datetime


class Ui_Dialog(object):
   # as smaller size significantly speeds up processing almost without affecting quality.
   width = 480
   height = 340
   height_orig=0
   width_orig=0
   camstatus=0
   count=0

   sx=0
   sy=0
   ex=0
   ey=0

   tm1=None
   tm2=None



   cap=None
   img=None


   def process(self):

      self.tm1=datetime.now()
      self.camstatus=1

      f=open('cord.txt','r')
      fd=f.read()
      f.close()
      cord=fd.split("#")



      self.cap = cv2.VideoCapture(0)
      width = 320
      height = 240
      self.cap.set(cv2.CAP_PROP_FRAME_HEIGHT, height)
      self.cap.set(cv2.CAP_PROP_FRAME_WIDTH, width)
      crop2 = cv2.imread('crop.bmp', 1)
      while True:
         ret, img_bgr = self.cap.read()
         if ret is False:
            print ("Error grabbing frame from camera")
            break
```

```
        cv2.imshow('img_rgb', img_bgr)


        cv2.rectangle(img_bgr, (int(cord[2]), int(cord[0])), (int(cord[3]), int(cord[1])), color=(0, 255, 0), thickness=3,
              lineType=8)
        crop1= img_bgr[int(cord[0]):int(cord[1]), int(cord[2]):int(cord[3])]


        cv2.imshow('c1', crop1)
        cv2.imshow('c2', crop2)

        # convert the images to grayscale
        grayA = cv2.cvtColor(crop1, cv2.COLOR_BGR2GRAY)
        grayB = cv2.cvtColor(crop2, cv2.COLOR_BGR2GRAY)

        (score, diff) = ssim(grayA, grayB, full=True)
        diff = (diff * 255).astype("uint8")
#         print("SSIM: {}".format(score))

        if score<.5:
#             print('change')
            self.tm2=self.tm1
            self.tm1=datetime.now()
            print('alert')
            import winsound
            frequency=2500
            duration=2000
            winsound.Beep(frequency,duration)

            cv2.imwrite('C:\\Users\\dell\\Desktop\\CCTV\\videosurveilance\\static\\img.jpg',img_bgr)
            imgg=cv2.imwrite('C:\\Users\\dell\\Desktop\\CCTV\\videosurveilance\\static\\img.jpg',img_bgr)
            db = pymysql.connect(host='localhost',user='root',password='',database='video_surveilance')
            c = db.cursor()

            sql="INSERT INTO `alert`( `alert`) VALUES ('movement detected')"
            c.execute(sql)
            db.commit()
            c.close()
            db.close()
#             try:

#                 tdiff=self.tm1-self.tm2
# #                 print(tdiff.total_seconds())
#                 print(tdiff.total_seconds())
#                 if tdiff.total_seconds()>7:
#                     print('alert')
#                     import winsound
#                     frequency=2500
#                     duration=2000
#                     winsound.Beep(frequency,duration)







#         difference = cv2.subtract(grayA, grayB)
#
#         ret, mask = cv2.threshold(difference, 0, 255,cv2.THRESH_BINARY_INV)
#
#         cv2.imshow('thresh', mask)
```

```
            key = cv2.waitKey(1)
            if key & 0xFF == 27:
                self.camstatus=0
                break
            elif self.camstatus==0:
#               self.camstatus=0
                break

        self.cap.release()
        cv2.destroyAllWindows()

    def close(self):
        self.camstatus=0



    def startcam(self):
        self.camstatus=1
        self.cap = cv2.VideoCapture(0)
        width = 320
        height = 240
        self.cap.set(cv2.CAP_PROP_FRAME_HEIGHT, height)
        self.cap.set(cv2.CAP_PROP_FRAME_WIDTH, width)
        while True:
            ret, img_bgr = self.cap.read()
            if ret is False:
                print ("Error grabbing frame from camera")
                break
            cv2.imshow('img_rgb', img_bgr)
            key = cv2.waitKey(1)
            if key & 0xFF == 27:
                self.camstatus=0
                break
            elif self.camstatus==0:
#               self.camstatus=0
                break

        #    if camstatus==0:
        #       break;

        self.cap.release()
        cv2.destroyAllWindows()

    def capbase(self):
        ret, img_bgr = self.cap.read()
        if ret is False:
            print ("Error grabbing frame from camera")
        else:
            cv2.imwrite("back.bmp", img_bgr)
            self.camstatus=0
            self.img = cv2.imread('back.bmp', 1)
            # displaying the image
            cv2.imshow('image', self.img)
            # setting mouse hadler for the image
            # and calling the click_event() function
            cv2.setMouseCallback('image', self.click_event)
            # wait for a key to be pressed to exit
            cv2.waitKey(0)
            # close the window
            cv2.destroyAllWindows()
```

Appendix

```
   def click_event(self,event, x, y, flags, params):
      self.count+=1
      # checking for left mouse clicks
      if event == cv2.EVENT_LBUTTONDOWN:

         # displaying the coordinates
         # on the Shell
         print(x, ' ', y)

         self.ex=x
         self.ey=y

         cv2.rectangle(self.img, (self.sx, self.sy), (self.ex, self.ey), color=(0, 255, 0), thickness=3, lineType=8)

         w=self.ex-self.sx
         h=self.ey-self.sy

         f=open('cord.txt','w+')
         f.write(str(self.sy)+"#"+str(self.ey)+"#"+str(self.sx)+"#"+str(self.ex))
         f.close()
#        crop_img = self.img[y:y+h, x:x+w]
         crop_img = self.img[self.sy:self.ey, self.sx:self.ex]

         cv2.imwrite("crop.bmp", crop_img)

         cv2.destroyAllWindows()



         # displaying the coordinates
         # on the image window
#         font = cv2.FONT_HERSHEY_SIMPLEX
#         cv2.putText(self.img, str(x) + ',' +
#                  str(y), (x,y), font,
#                  1, (255, 0, 0), 2)
         cv2.imshow('image', self.img)

      # checking for right mouse clicks
      if event==cv2.EVENT_RBUTTONDOWN:

         # displaying the coordinates
         # on the Shell
#         print('right')
         print(x, ' ', y)


         self.sx=x
         self.sy=y

         # displaying the coordinates
         # on the image window
#         font = cv2.FONT_HERSHEY_SIMPLEX
#         b = self.img[y, x, 0]
#         g = self.img[y, x, 1]
#         r = self.img[y, x, 2]
#         cv2.putText(self.img, str(b) + ',' +
#                  str(g) + ',' + str(r),
#                  (x,y), font, 1,
```

Appendix

```
#                   (255, 255, 0), 2)
         cv2.imshow('image', self.img)



   def setupUi(self, Dialog):
      Dialog.setObjectName("Dialog")
      Dialog.resize(929, 572)
      self.frame = QtWidgets.QFrame(Dialog)
      self.frame.setGeometry(QtCore.QRect(10, 10, 151, 551))
      self.frame.setStyleSheet("background-color: rgb(194, 194, 194);")
      self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
      self.frame.setFrameShadow(QtWidgets.QFrame.Raised)
      self.frame.setObjectName("frame")
      self.pushButton = QtWidgets.QPushButton(self.frame)
      self.pushButton.setGeometry(QtCore.QRect(10, 10, 131, 41))
      self.pushButton.setObjectName("pushButton")
      self.pushButton_2 = QtWidgets.QPushButton(self.frame)
      self.pushButton_2.setGeometry(QtCore.QRect(10, 50, 131, 41))
      self.pushButton_2.setObjectName("pushButton_2")
      self.pushButton_3 = QtWidgets.QPushButton(self.frame)
      self.pushButton_3.setGeometry(QtCore.QRect(10, 90, 131, 41))
      self.pushButton_3.setObjectName("pushButton_3")
      self.pushButton_4 = QtWidgets.QPushButton(self.frame)
      self.pushButton_4.setGeometry(QtCore.QRect(10, 130, 131, 51))
      self.pushButton_4.setObjectName("pushButton_4")
#      self.pushButton_5 = QtWidgets.QPushButton(self.frame)
#      self.pushButton_5.setGeometry(QtCore.QRect(10, 180, 131, 41))
      self.pushButton.clicked.connect(self.startcam)
      self.pushButton_2.clicked.connect(self.capbase)
      self.pushButton_3.clicked.connect(self.process)
      self.pushButton_4.clicked.connect(self.close)



      self.retranslateUi(Dialog)
      QtCore.QMetaObject.connectSlotsByName(Dialog)

   def retranslateUi(self, Dialog):
      _translate = QtCore.QCoreApplication.translate
      Dialog.setWindowTitle(_translate("Dialog", "Dialog"))
      self.pushButton.setText(_translate("Dialog", "Start Cam"))
      self.pushButton_2.setText(_translate("Dialog", "Mark Object"))
      self.pushButton_3.setText(_translate("Dialog", "Process"))
      self.pushButton_4.setText(_translate("Dialog", "Exit"))
#      self.pushButton_5.setText(_translate("Dialog", "Exit"))
      #self.label_4.setText(_translate("Dialog", "Total Face Detected"))
      #self.label_6.setText(_translate("Dialog", "Total Male Faces"))
      #self.label_7.setText(_translate("Dialog", "Total Female Faces"))


if __name__ == "__main__":
   import sys
   app = QtWidgets.QApplication(sys.argv)
   Dialog = QtWidgets.QDialog()
   ui = Ui_Dialog()
   ui.setupUi(Dialog)
   Dialog.show()
   sys.exit(app.exec_())
```

Database Design

Attribute Name	Datatype	Width	Description
Login id	int	50	Primary Key
Username	varchar	100	
Password	varchar	100	
Type	varchar	100	
U id	int	50	

Table A.1: Login

Attribute Name	Datatype	Width	Description
C id	int	50	Primary Key
U id	int	50	
Complaint	Varchar	100	
Reply	varchar	100	
Date	date		
Time	time	6	

Table A.2: Complaint

Attribute Name	Datatype	Width	Description
Items id	int	50	Primary Key
Items	varchar	100	
Description	Varchar	100	
Showcase num	int	10	

Table A.3: Items

Attribute Name	Datatype	Width	Description
User id	int	50	Primary Key
Username	varchar	100	
Password	Varchar	100	
Gender	varchar	20	
E-mail	varchar	100	
Phone no	varchar	11	
Address	varchar	100	

Table A.4: User

Attribute Name	Datatype	Width	Description
A id	int	11	Primary Key
Alert	varchar	50	
image	varchar	700	

Table A.5: Alert

Dataflow Diagram

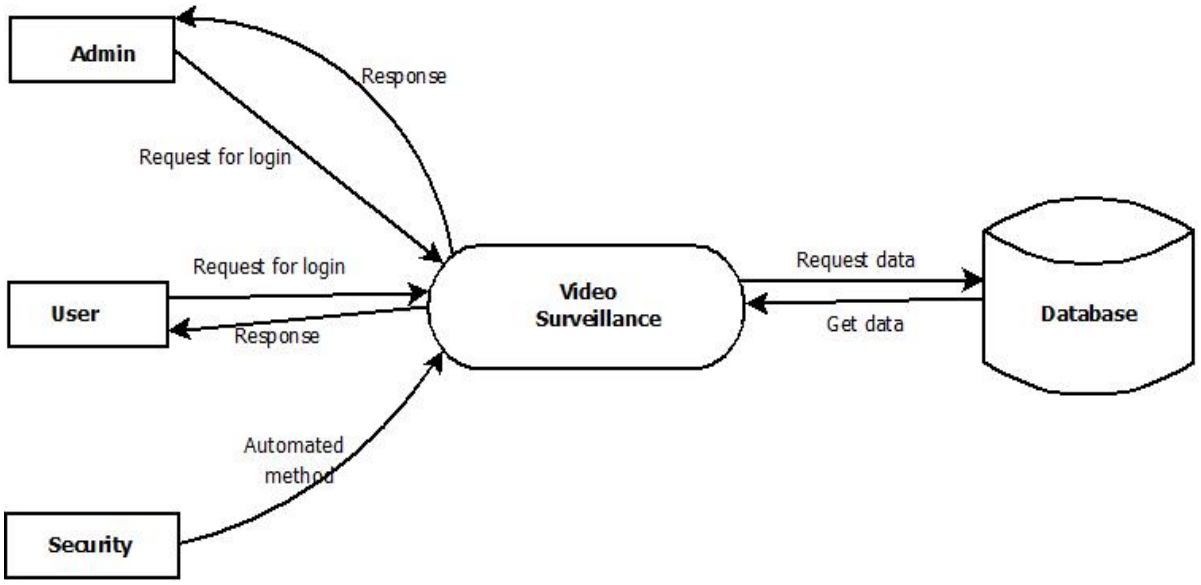

Figure A.1: DFD - Level 0

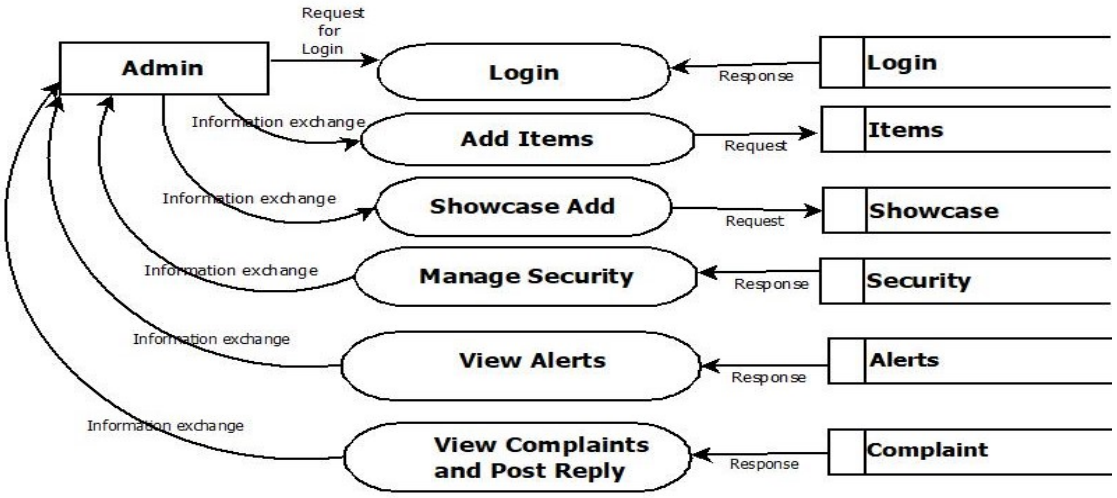

Figure A.2: DFD - Level 1

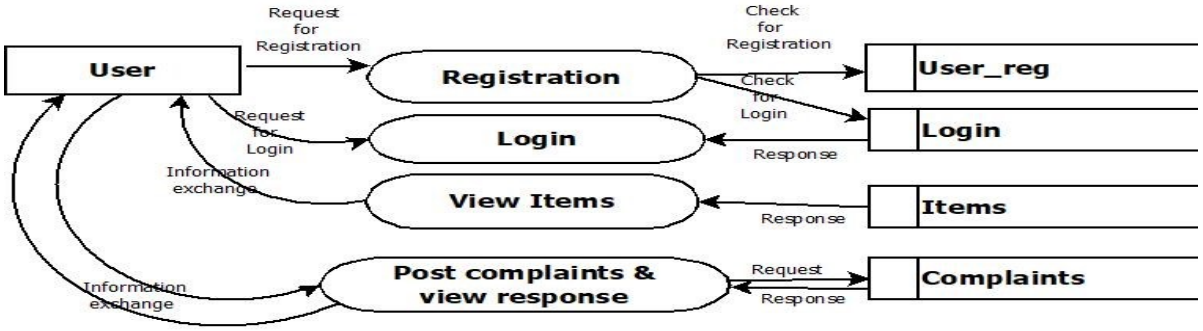

Figure A.3: DFD - Level 2

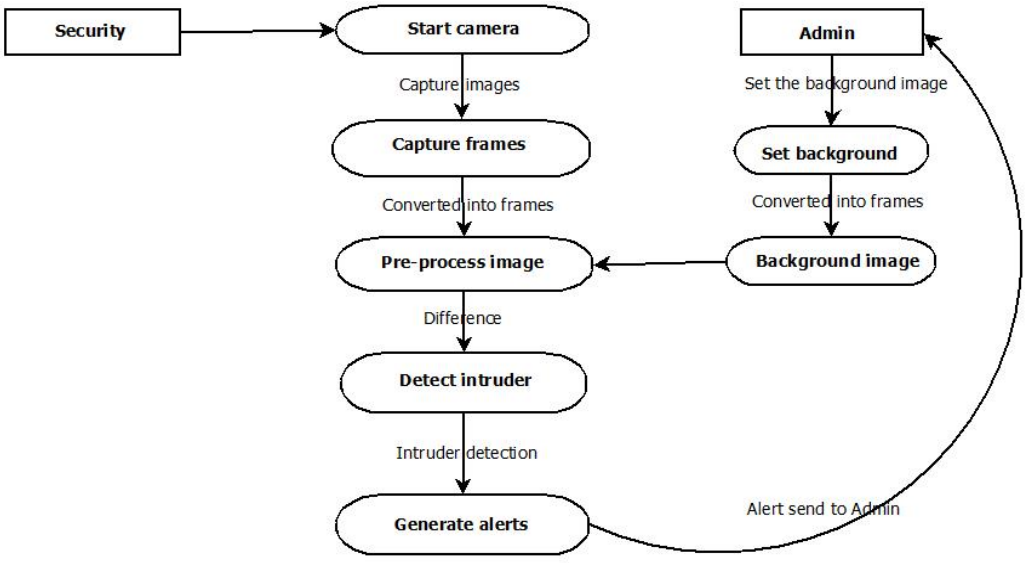

Figure A.4: DFD - Level 3