# VIRTUAL E-COMMERCE (AUGMENTED REALITY)

A Major Project Report submitted to

**Jawaharlal Nehru Technological University, Hyderabad**

In partial fulfillment for the requirement for the award of B.Tech Degree in Computer Science and Engineering

By

| | |
|---|---|
| **CHIRUMALLA KAMAL** | **18641A05E2** |
| **EDELLI SRIKANTH** | **18641A05D4** |
| **GOURISHETTY CHANDAN** | **18641A05D8** |
| **KUDIKALA ROHITH** | **18641A05E7** |
| **KETHAM SAI SANTHOSH** | **18641A05E4** |

**Under the Guidance of**

Mrs. K. Rekha Devi

Assistant Professor

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VAAGDEVI COLLEGE OF ENGINEERING**

(UGC Autonomous, Accredited by NBA, Accredited by NAAC with “A”)

Bollikunta, Warangal Urban-506 005 (2018-2022)

**VAAGDEVI COLLEGE OF ENGINEERING**

(UGC Autonomous, Accredited by NBA, Accredited by NAAC with “A”)

Bollikunta, Khila Warangal (Mandal), Warangal Urban-506 005 (T.S)

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

# CERTIFICATE

This is to certify that the mini project entitled **VIRTUAL E-COMMERCE (AUGMENTED REALITY)** is submitted by ***CH.KAMAL, E.SRIKANTH, G.CHANDAN, K.ROHITH, K.SAI SANTHOSH*** bearing Roll No(s) **18641A05E2, 18641A05D4, 18641A05D8, 18641A05E7, 18641A05E4** in partial fulfillment of the requirements for the award of the Degree in Bachelor of Technology in Computer Science and Engineering during the academic year 2021-2022.

**Guide** **Head of the Department** **External Examiner**

# Acknowledgement

First and foremost, we sincerely thank our esteemed institution **Vaagdevi College of Engineering, Warangal** for giving us this great opportunity to fulfill our dreams of becoming Software Engineers.

We would like to thank the Head of the Institution and Principal **Dr. K. Prakash** for the extensive support and for helping us in times we need it. Finally, we thank eachand everyone who extended their help directly or indirectly and made this project successful.

We are highly obliged to the Head of the Department of Computer Science and Engineering, **Dr. N. Satyavathi** for her constant support and extensive help in each and every step.

We sincerely thank our Minor project guide **Mrs. K. Rekha Devi**, Assistant Professor of ComputerScience and Engineering for encouraging and supporting us throughout the project from start to end.

**With Sincere Regards,**

| | |
|---|---|
| **CH.KAMAL** | **(18641A05E2)** |
| **E.SRIKANTH** | **(18641A05D4)** |
| **G.CHANDAN** | **(18641A05D8)** |
| **K.ROHITH** | **(18641A05E7)** |
| **K.SAI SANTHOSH** | **(18641A05E4)** |

# DECLARATION

We hereby declare that the project work entitled “**VIRTUAL E-COMMERCE (AUGMENTED REALITY**)” is submitted to the Department of Computer Science and Engineering in fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering in record of the original work done by our team under the guidance of **Mrs.K. Rekha Devi** and this project had not been submitted to the any other university for the award for any Degree or Diploma.

**With Sincere Regards,**

| | |
|---|---|
| **CH.KAMAL** | **(18641A05E2)** |
| **E.SRIKANTH** | **(18641A05D4)** |
| **G.CHANDAN** | **(18641A05D8)** |
| **K.ROHITH** | **(18641A05E7)** |
| **K.SAI SANTHOSH** | **(18641A05E4)** |

# ABSTRACT

Online shopping has certainly replaced the traditional way of shopping for daily goods and clothing. When we choose online shopping, we get the benefit of credibility. Today, almost every online store offers cash on delivery, free shipping, and discounted prices. These online shopping stores eliminate the hassles of parking, getting stuck in traffic jams and standing in long queues for billing. They have also benefited those people who always complain of a shortage of time. This is the reason, the majority of people have turned to online shopping. Here, they enjoy easy access to an attractive price range, prompt customer support, and free home delivery. There is no doubt that these are some of the attractive features that catch the attention of consumers. Although there is one small issue that could make people lose interest in online shopping, it might not be possible to try-on clothes in such cases. Our motive here is to increase the time efficiency and improve the accessibility of clothes virtually try on in an E-commerce website. Our proposed approach is mainly based on the extraction of the user image from the video stream, and alignment of models. Extraction of the user allows us to create an augmented reality environment by isolating the user area from the video stream and superimposing it onto a virtual environment in the user interface. We use the Haarcasecade pretrained XML model in order to align the 2D cloth models with the user. Finally, the model is superimposed on the user in real-time.

# INDEX

**CONTENTS** **Page No.**

# 1.INTRODUCTION

Augmented Reality allows e-commerce customers to preview products or experience services in their own environment and on their own time, before electing to make a purchase. Using AR, your customers can preview products and be more likely to pick the right product the first time.

Augmented reality (AR) is an enhanced version of the real physical world that is achieved through the use of digital visual elements, sound, or other sensory stimuli delivered via technology. It is a growing trend among companies involved in mobile computing and business applications in particular.

Augmented reality continues to develop and become more pervasive among a wide range of applications. Since its conception, marketers and technology firms have had to battle the perception that augmented reality is little more than a marketing tool. However, there is evidence that consumers are beginning to derive tangible benefits from this functionality and expect it as part of their purchasing process.

For example, some early adopters in the retail sector have developed technologies that are designed to enhance the consumer shopping experience. By incorporating augmented reality into catalog apps, stores let consumers visualize how different products would look like in different environments. For furniture, shoppers point the camera at the appropriate room and the product appears in the foreground.

Augmented reality (AR) involves overlaying visual, auditory, or other sensory information onto the world in order to enhance one's experience.

Retailers and other companies can use augmented reality to promote products or services, launch novel marketing campaigns, and collect unique user data.

Unlike virtual reality, which creates its own cyber environment, augmented reality adds to the existing world as it is.

## 1.1. EXISTING SYSTEM

In other words, online shopping or e-shopping is a form of electronic commerce that allows consumers to directly buy goods or services from a seller over the Internet using a web browser.

With online shopping we are no longer required to visit malls to make a purchase.

More and more people have turned to the Internet to buy everything from food, clothes, houses, and groceries.

The idea of not having to wait in long queues or in traffic or search through racks of clothing looking for the items or being able to shop at any time—preferably in pajamas—has caused more people to turn to shop online.

### Advantages:

- Saves time and efforts.
- The convenience of shopping at home.
- Wide variety/range of products are available.
- Good discounts / lower prices.
- Get detailed information about the product.
- We can compare various models/brands.

### Disadvantages

- Can't able to try products.
- Delay in the delivery.
- Lack of interaction.
- Returning the product.
- You can't touch the product.

## 1.2. PROPOSED SYSTEM

Buying apparel through e-commerce has certain limitations of trying apparel before buying it. Here we are providing a virtual room to try apparel through an e-commerce website before buying it. Providing a virtual room to try apparel through an e-commerce website before buying it.

So that users are motivated to buy online apparel through an e-commerce website and have lesser chances of returning and canceling of apparel because of having a facility to try apparel through a virtual room. we were included many things like goggles, clothes ear rings, Hats, jewelry etc.…

A sample application with the user interface is developed to test practically the performance. The user interface allows the user to choose a dress by means of cursor movements.

### Advantages:

- Saves time and efforts.
- Able to try products virtually.
- It reduces the number of product returns.
- Have total freedom regarding decision making, trying and choosing products at their own pace, without feeling the pressure to make a purchase.
- Encourages cross sales.
- Increases the connectivity with consumers and gain a competitive advantage in the market.
- Improving personalization.
- No-Contact Trial.

## 1.3. SOFTWARE REQUIREMENTS

The Software Requirements in this project include:

a. Python

b. OpenCV framework

c. Visual Studios(IDE)

d. Flask

## LIBRARIES USED

The Libraries used in this project are:

- **OpenCV**

Open CV (Open Source Computer Vision Library) is an open-source computer vision software library for machine learning. Open CV was developed to serve the purpose ofcomputer vision applications and to stimulate the usage of machine perception in commercially viable products. OpenCV's role in our project is to invoke the camera capture live video and preprocess the video frames.

- **Numpy**

NumPy is a Python library used for working with arrays. It also has functions for workingin the domain of linear algebra, Fourier transform, and matrices. It is an open-source project and you can use it freely. NumPy stands for Numerical Python. To perform operations on the face encodings array we made use of the Numpy library.

- **Dlib**

It‘s a landmark's facial detector with pre-trained models, the dlib is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person's face like image below.

- **Cmake**

CMake is an open-source, cross-platform tool that uses compiler and platform independent configuration files to generate native build tool files specific to your compiler and platform.

The CMake Tools extension integrates Visual Studio Code and CMake to make it easy to configure, build, and debug your C++ project.

- **Scipy**

SciPy is a scientific computation library that uses NumPy underneath.

SciPy stands for Scientific Python.

It provides more utility functions for optimization, stats and signal processing.

Like NumPy, SciPy is open source so we can use it freely.

SciPy was created by NumPy's creator Travis Olliphant.

- **Gunicorn**

Gunicorn is a pure-Python HTTP server for WSGI applications. It allows you to run any Python application concurrently by running multiple Python processes within a single dyno. It provides a perfect balance of performance, flexibility, and configuration simplicity.

**OpenCV modules**

OpenCV has a modular structure, which means that the package includes several shared or static libraries. The following modules are available:

- **Core** - a compact module defining basic data structures, including the dense multi-dimensional array Mat and basic functions used by all other modules.
- **imgproc** - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
- **Video** - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
- **calib3d** - basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
- **features2d -** salient feature detectors, descriptors, and descriptor matchers.

- **Objdetect** - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
- **highgui** - an easy-to-use interface to video capturing, image and video codecs, as well as simple UI capabilities.
- **Gpu** - GPU-accelerated algorithms from different OpenCV modules.

OpenCV is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Android and Mac OS. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi core processing. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 5 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

## 1.4. HARDWARE REQUIREMENTS

| **Components** | **Minimum** | **Recommended** |
|---|---|---|
| **Processor** | Intel Pentium , Dual Core | Intel Core i5 10th GEN |
| **RAM** | 256MB | 8GB |
| **Camera** | HD 720p Webcam | Full HD 1080p Webcam |
| **Disk** | 200MB | 80Gb |

## 1.5. LITERATURE SURVEY

The current method of online shopping does not guarantee the perfect size of the clothing. This results in a number of products being returned and the time taking to replace it with the correct sized one is long. This is a major setback for the online shopping industry.

The various approaches to obtain the desired results are as follows

Kjaerside et al. proposes a tag-based approach which requires manual labeling of body parts in order to create an augmented reality of the customer wearing a cloth simulation. Use of shape descriptors such as HistogramsofOriented Gradients (HOG) can also be an option for detection of the user and body posture estimation.

Onishi et al.propose a HOG-based approach to estimate human postures as discrete states. However, use of a limited number of skeletal posture states may not be convenient for a continuous body tracking system.

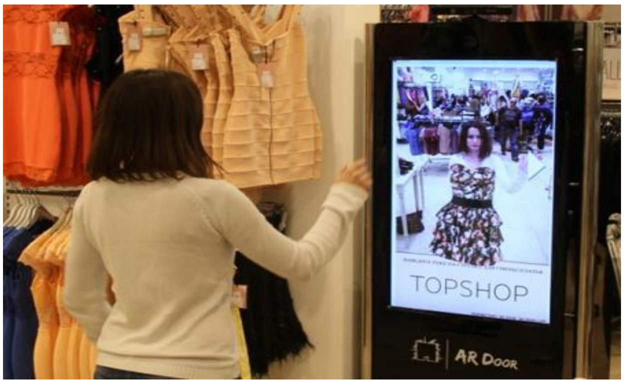
Shotton et al. have developed a real-time human pose recognition system – which is also available in the Microsoft Research Kinect SDK – that predicts the 3D positions of body joints from a single depth image.

Some of the approaches available in the market are

### Topshop

This solution was launched by a mega-fashion brand TOPSHOP. User has to pose with their arms above their heads and allow for the Kinect to take a photo. User can using gestures select the clothes he/she wants to try. Once selected, the clothes selected by the user are pasted onto the picture taken before. Major drawback with this solution is that it works on just a single image and that too in specific pose.

Fig 1.5.1. Top Shop's implementations demo at one of its stores.



## VIRTUAL MIRRORS

Virtual mirrors also known as virtual dressing rooms as mentioned above however not much work has been done on attempting the problem of color change and texture projection for shirts that we are after. One successful project is Virtual Mirror; this however was extremely heavy on the constraints it posed on the user who was using the system. The system required that a green shirt be worn and that there be texture on the shirt in the region where the texture had to be changed. These constraints meant that this would not be applicable on a plain any colored shirt.

## SWIVEL

An avatar is used here instead of a real person. A large collection is available on digital library where the user can choose for their preference.Since an avatar is used its not very impressive.

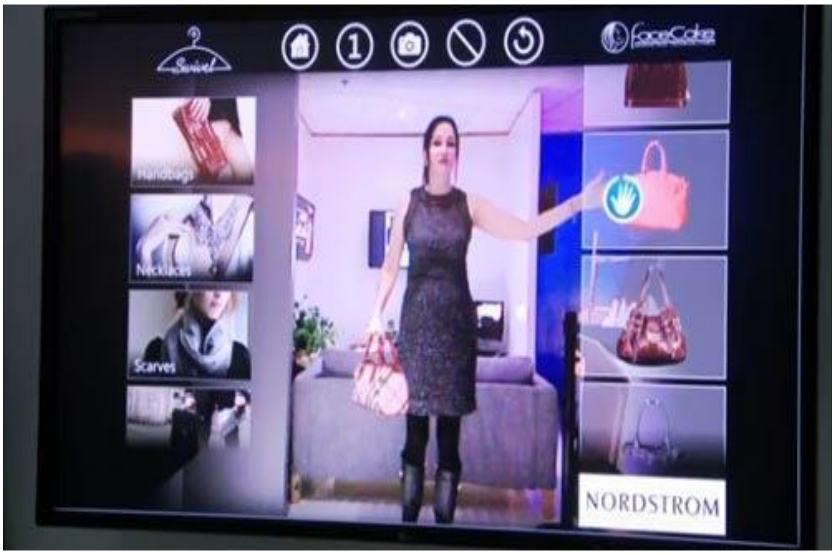


Fig 1.5.2. SWIVEL

E-commerce is growing rapidly on a global scale. Among many products purchased via the Internet, clothing is the first in terms of purchase frequency. However, there are growth barriers for this product category, which include, first of all, the client's fear of matching clothing to their own figure or complexion. This results in a high percentage of returns reaching up to 60% of transactions, which is more than that in other e-commerce sectors. One of the possible solutions to the abovementioned problem is the use of a virtual trial room (VTR), which allows you to try on clothes in terms of size, fit, style, or color on a computer or smartphone screen. The main purpose of the article is to determine the propensity to use a VTR in the age group of generation Y. The second goal is to compare the propensity to use by type of VTR: 2D vs. 3D. The methodology is based on the qualitative exploratory approach. To conduct research, content analysis and sentiment analysis were used. The results of the study indicate that the participants of the research have an ambivalent attitude towards VTR – on the one hand, they perceive VFRs as an interesting solution for Internet users (not only generation Y). On the other hand, however, they themselves show a distance to use the VTR. The analysis also showed that a two-dimensional type of VTR based on augmented reality technology has greater market opportunities.

## LenseCart Try Now:

With the 3D trial-on feature at Lenskart, you can actually see how the frames will look on you at 180 degrees' angle. You can try on various eyewear from brands including Hugo Boss, Carrera, Vincent Chase, John Jacobs, Tommy Hilfiger, Ray-Ban and many more and choose the one you think suits you the best.

Fig 1.5.3 Lensecart Try in 3D

## Tanishq Try on Jewellery

Tanishq Offers Virtual try on options on selective range of jewellery. Try and purchase according to the perfect size, fit and designs.

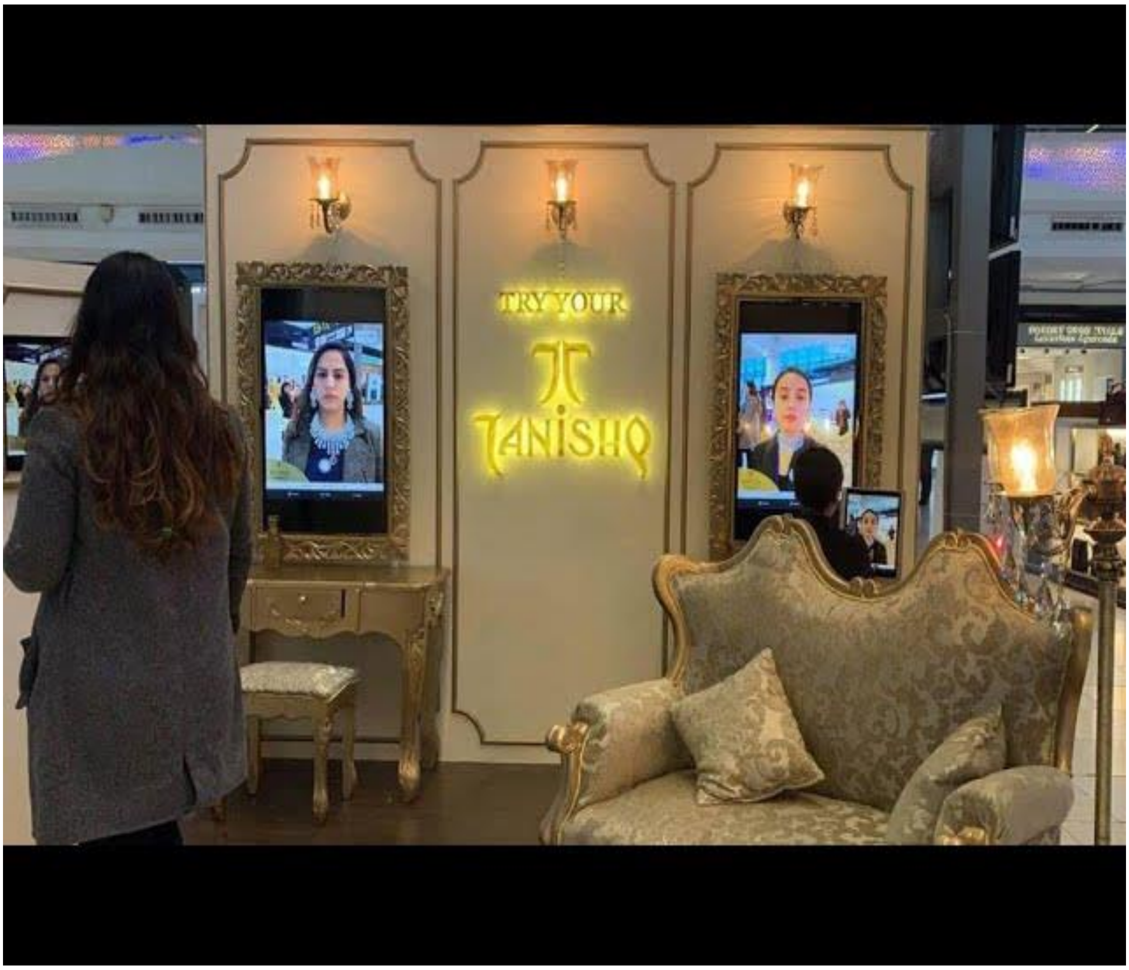


Fig 1.5.4: Tanishq Try on

## IKEA AR

IKEA has launched a new augmented reality (AR) application that allows users to test IKEA's products in real time through Apple iOS 11’s ARKit technology. Dubbed IKEA Place, the iPhone- and iPad-compatible free application features realistically-rendered, true-to-scale 3D products. "The app automatically scales products, based on room dimensions, with 98 percent accuracy," noted a press release by the company.

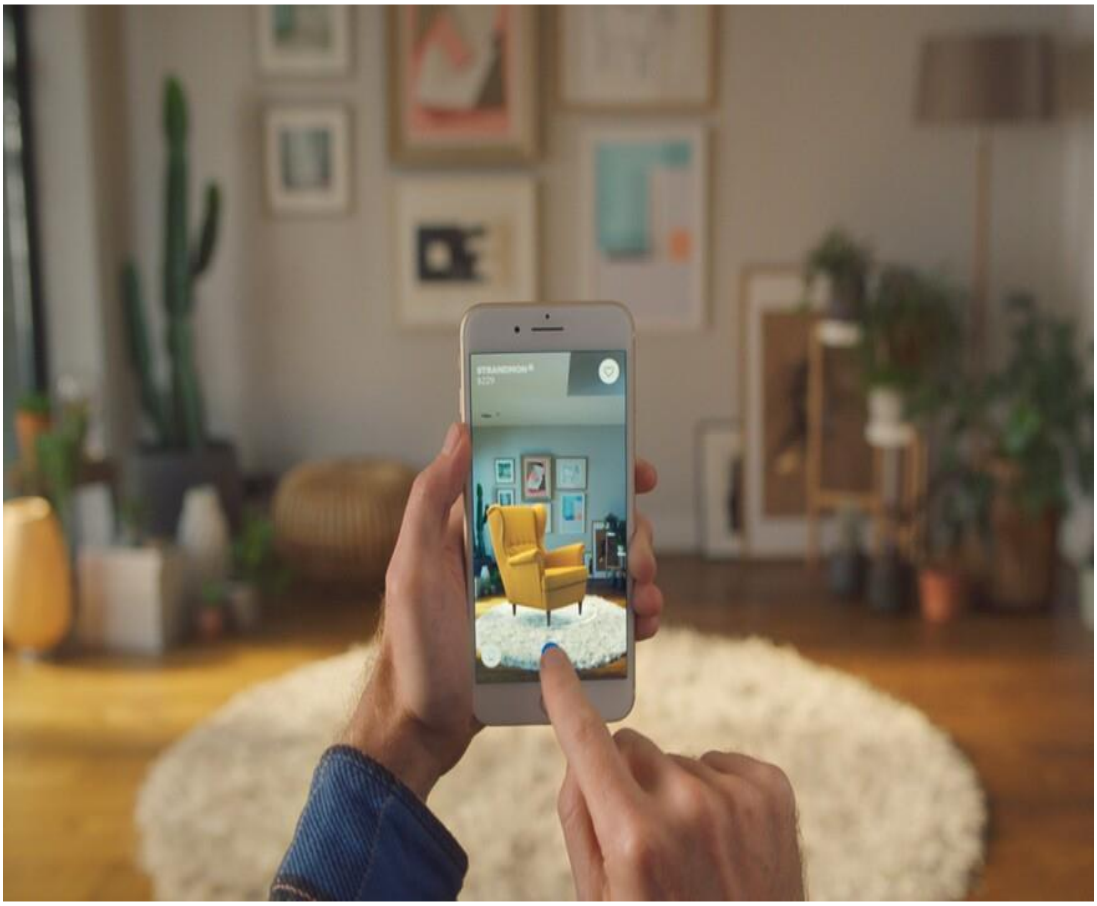


Fig 1.5.5: IKEA AR

In order to visualize a product within a space, the application scans the expanse of a room through an iPhone or an iPad camera. Users can browse through over 2,000 IKEA products on an online database, to make their selections. Once chosen, users must point the device to the desired spot in a room, then drag and drop the selected product onto the space. IKEA Place can also save each user's favorite products, share their selections on social media, and facilitate direct purchases through the IKEA website.

# 2. DESIGN OF THE PROJECT

## 2.1. Flow Chart

Flow chart is also process flowchart, process flow diagram. A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan. Elements that may be included in a flowchart are a sequence of actions, materials or services entering or leaving the process (inputs and outputs), decisions that must be made, people who become involved, time involved at each step, and/or process measurements.

- A **flowchart** is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

- The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

  Flowcharts are used to design and document simple processes or programs. Like other types of diagrams, they help visualize the process. Two of the many benefits are flaws and bottlenecks may become apparent. Flowcharts typically use the following main symbols:

- A process step, usually called an activity, is denoted as a rectangular box.
- A decision is usually denoted as a diamond.

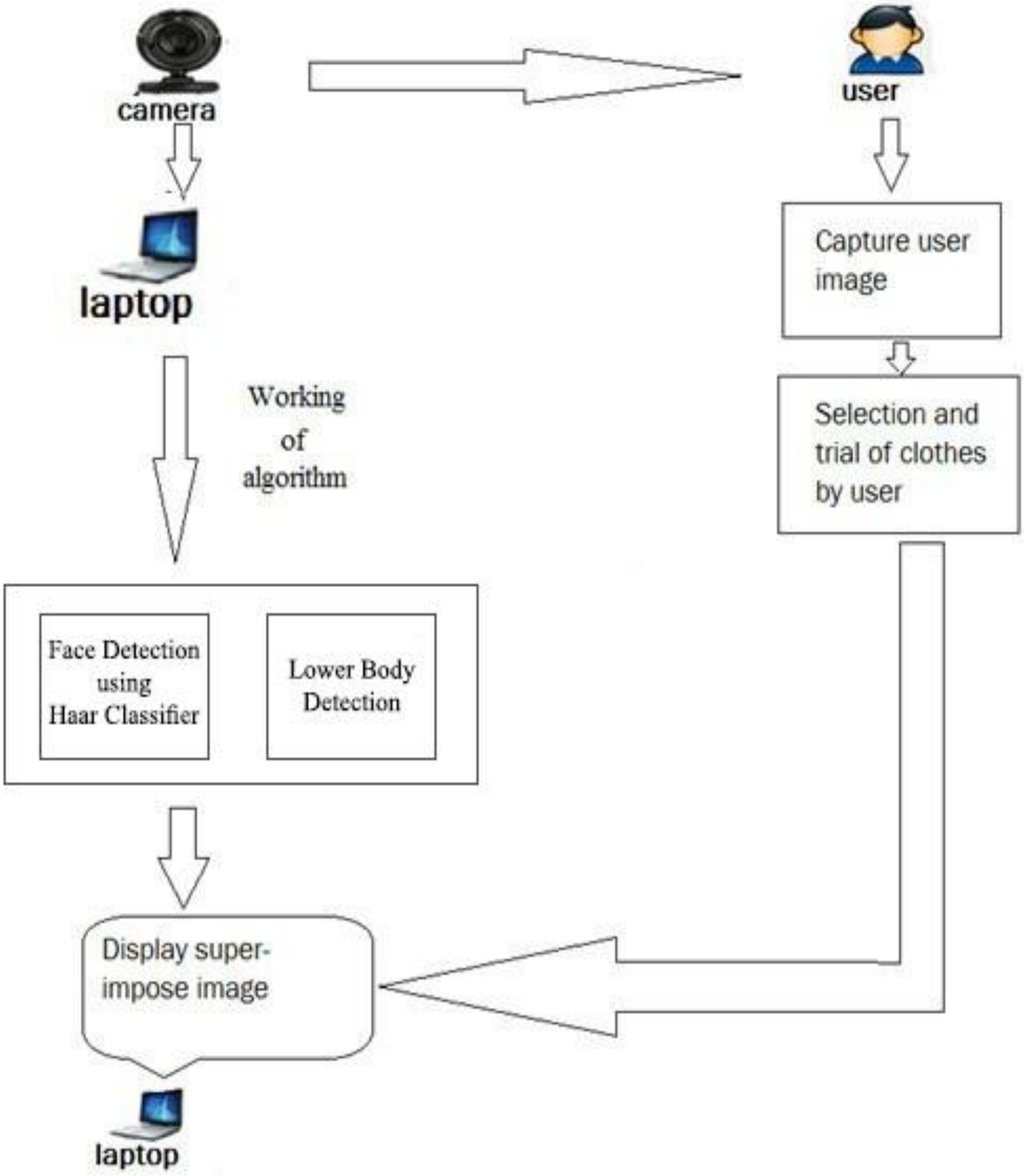

camera
user
laptop
Capture user image
Selection and trial of clothes by user
Working of algorithm
Face Detection using Haar Classifier
Lower Body Detection
Display super-impose image
laptop

**Fig 2.1: Flow chart**

## 2.2. Algorithms

### 2.2.1.Face Detection Algorithm

Face detection algorithm is used for checking whether the given input image contains any human face, and if face is present, returning the location of face in image. Face is an important thing for identifying presence of people. Here we are using face detection algorithm using haar classifier. Object detection for haar classifier is done by haar like feature. These features use the change in value of contrast between the adjacent rectangles. Here rectangle is a group of pixels. The haar like feature is formed using using two or more rectangles. The haar features can be easily scaled by maximizing and minimizing the size of the pixels. This haar feature is used for face detection. These haar feature is shown in figure 2.2.1.1., figure 2.2.1.2. and figure 2.2.1.3.

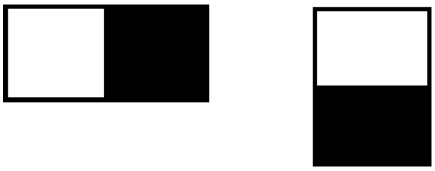


Fig. 2.2.1.1. Edge Features

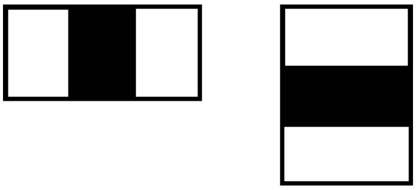


Fig. 2.2.1.2. Line Features

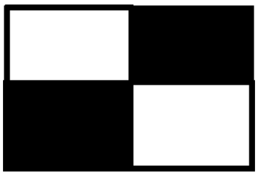


Fig. 2.2.1.3. Four Square Feature

OpenCV has a trainer as well as detector. This OpenCV can be trained using classifier for any object like car, rackets, etc. here we have to deal with face detection. OpenCV already contains the pre-trained classifier like eye, face. These trained classifiers are stored in an XML file and these XML files are stored in a desired path.

Initially we need to load XML classifier in our system. The image of the user captured by the camera is loaded. Then we find the face. If face is detected, then it returns position of detected face as rect(x,y,w,h). After that we can create the region of the face as shown in figure 2.2.1.4. [7]

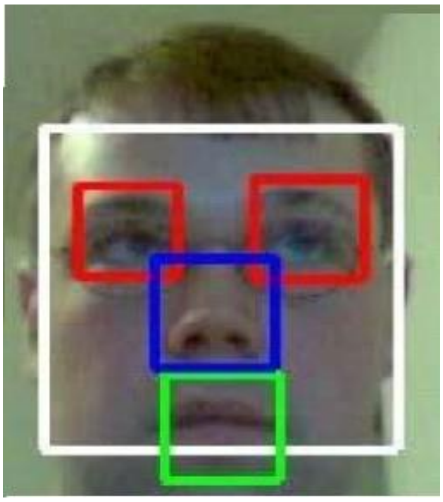


Fig. 2.2.1.4. Face Detection (ROI)

## 2.2.2.Lower Body Detection Algorithm

Lower body detection is done by using Haar classifiers. Haar Classifier is used to detect the humans in a moving video; features like eye detection, face detection, upper body, lower body and full body detection. Haar classifier are instructed with the help of negative and positive samples of images and saved as .xml files. Haar cascades is Haar classifiers one of the function. Some cascades used in OpenCV to detect human are haar cascade_upperbody, haar cascade_lowerbody, haarcascade_fullbody, haarcascade_frontalface etc. Haar classifier uses Region of Interest (ROI) for detecting lower body. Figure 2.2.2.1 shows the flowchart for lower body detection using haar classifier.

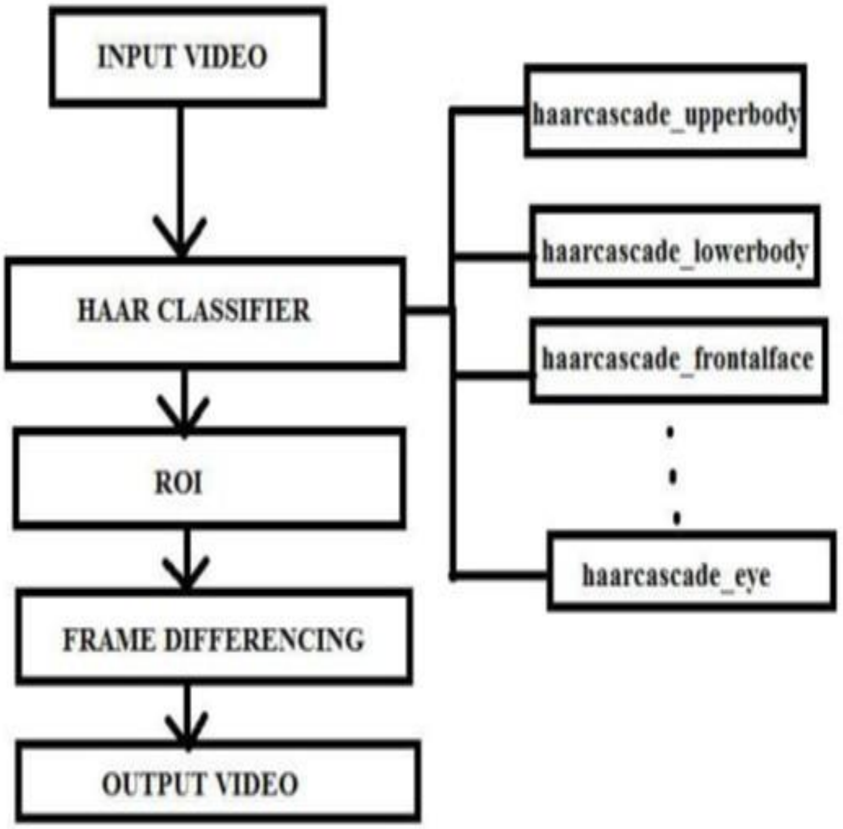


Fig. 2.2.2.1. Flowchart of lower body detection

Frame differencing identifies the moving body from reference and current frame. Region of interest draws bounding box on moving object such as lower body. After drawing the bounding box we get the ROI image which we show in our separate window. After getting the separate window we apply the dresses on that window. So lower body detection algorithm using haar classifier is very efficient algorithm to use.

### 2.2.3: What is Dlib?

It‘s a landmark’s facial detector with pre-trained models, the dlib is used to estimate the location of 68 coordinates (x, y) that map the facial points on a person’s face like image below.

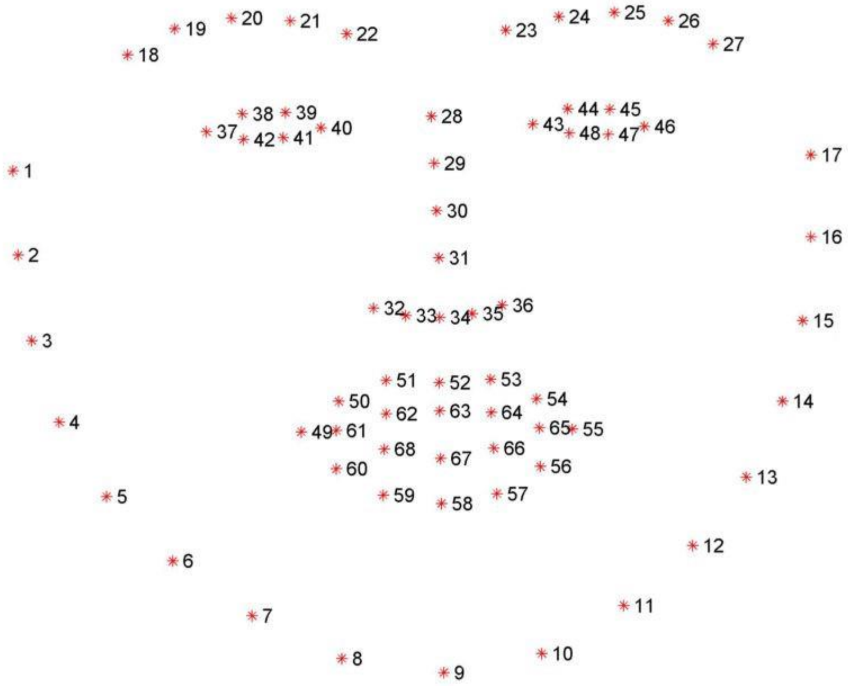


Fig 2.2.3.1: 63 Coordinates

These points are identified from the pre-trained model where the iBUG300-W dataset was used.

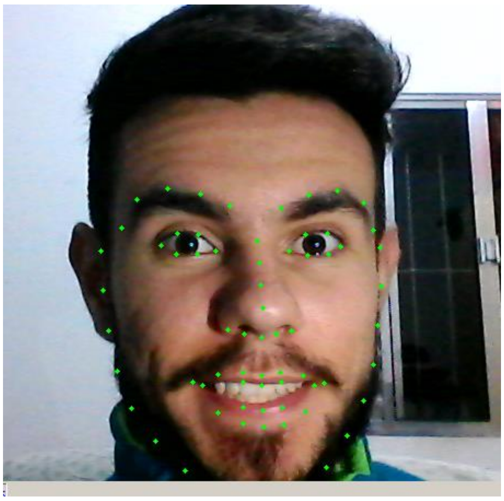


Fig 2.2.3.2: 63 coordinates Example

## 2.2.4: Cascade of Classifiers

"Instead of applying all the 6000 features on a window, group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region." - Face Detection using Haar Cascades.

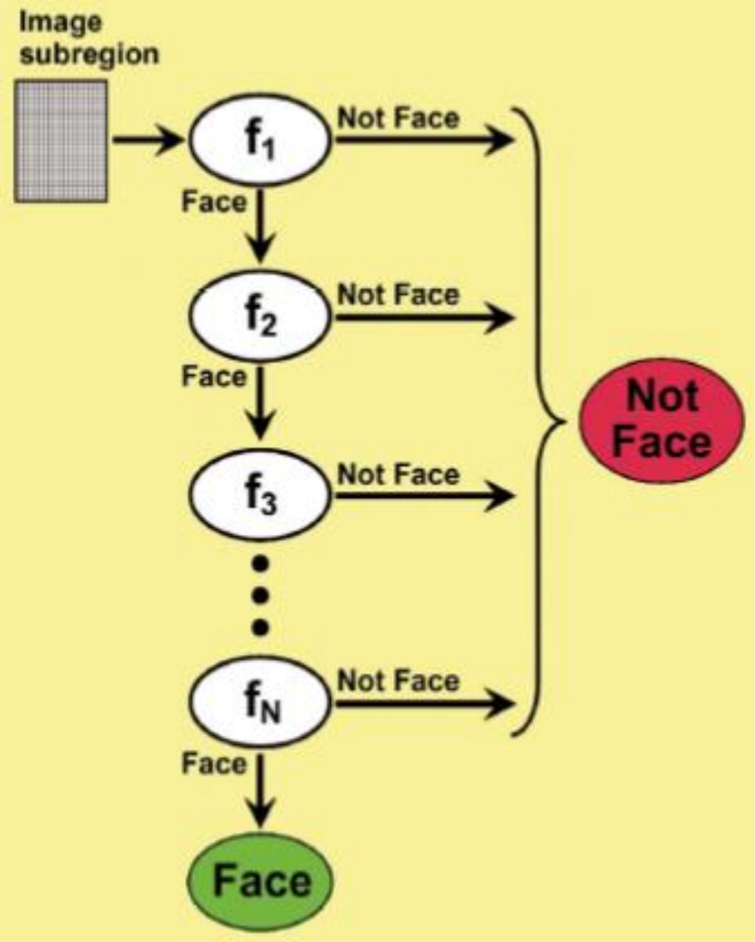


**Fig 2.2.4.1: Haarcasecades**

## OpenCV's pre-trained classifiers

OpenCV already contains many pre-trained classifiers for face, eyes, smile etc. Those XML files are stored in **opencv/data/haarcascades/** folder:

## 2.3.UML DIAGRAMS

UML stands for Unified Modeling Language and is a way of visually representing the architecture, design, and implementation of complex software systems. When you write code, there are thousands of lines in an application, and it's difficult to keep track of them. The relationships and hierarchies within a software system. UML diagrams divide this software system into components and subcomponents.

**Things in the UML:**

There are four kinds of things in the UML:

- Structural things
- Behavioral things
- Grouping things
- Annotational things

**Building Blocks of the UML:**

The vocabulary of the UML encompasses three kinds of building blocks:

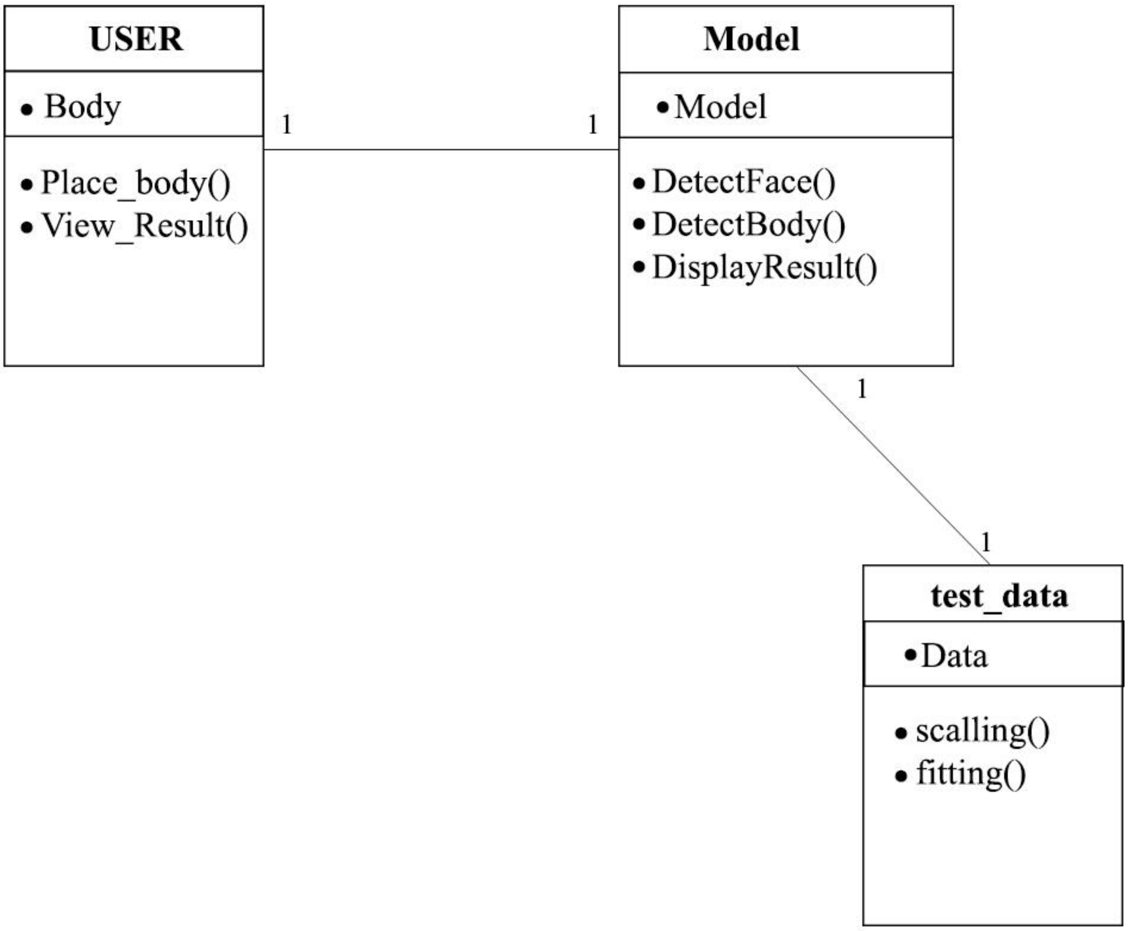
- Things
- Relationships
- Diagrams

Things are the abstractions that are first-class citizens in a model; relationships tie these things together; diagrams group interesting collections of things.

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

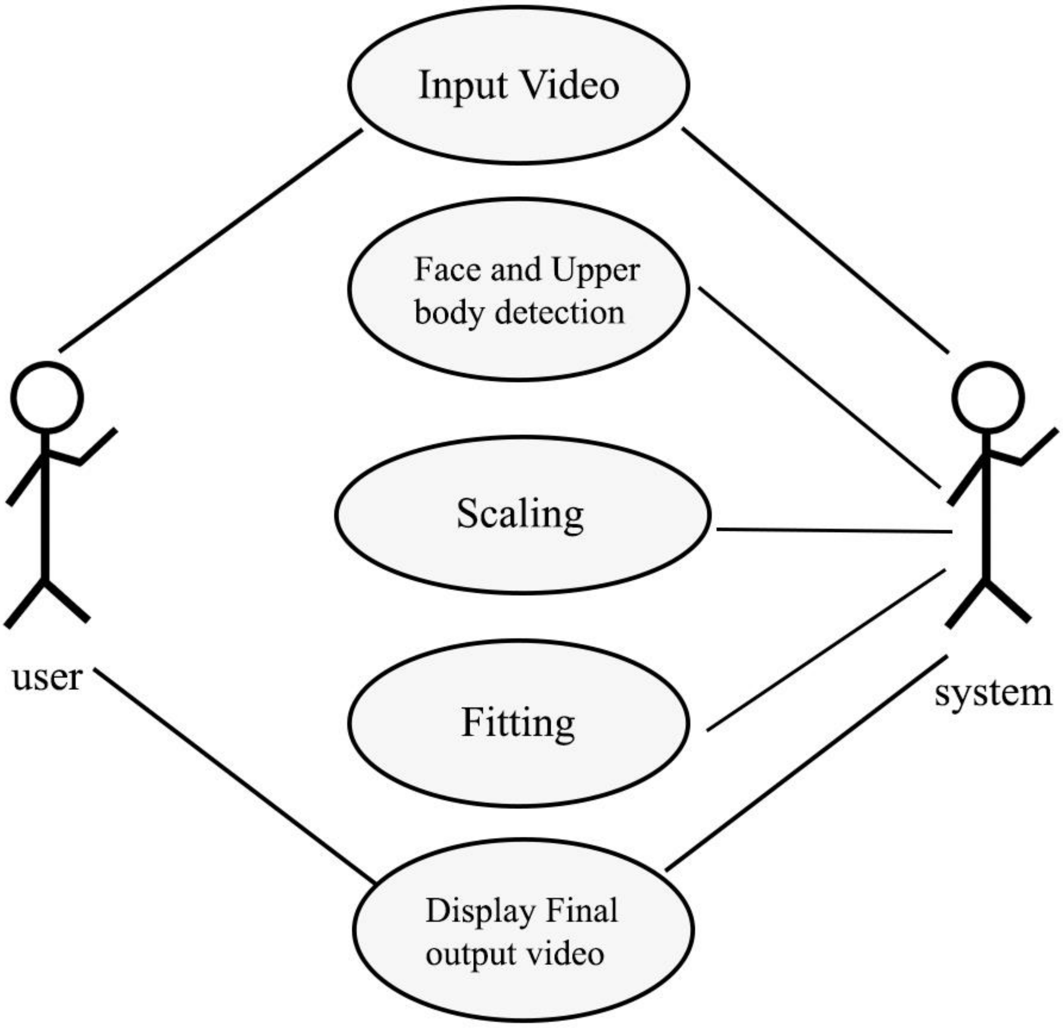
## CLASS DIAGRAM

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling.



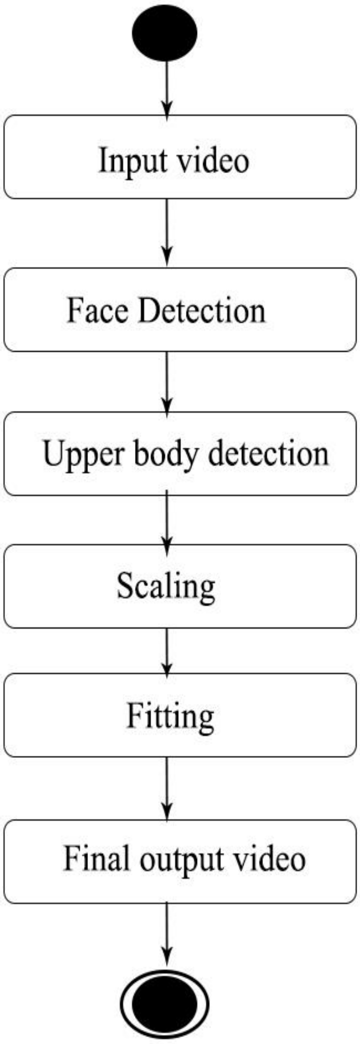
# USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



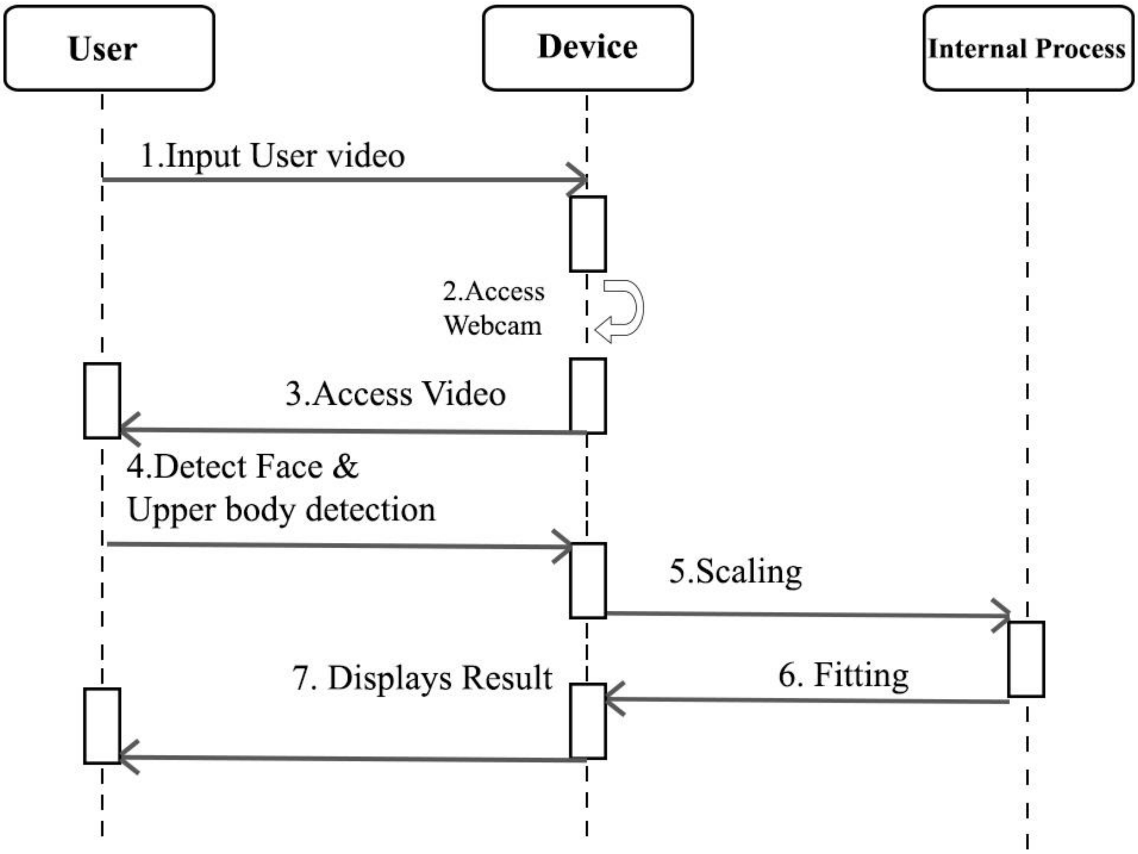
## ACTIVITY DIAGRAM

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.



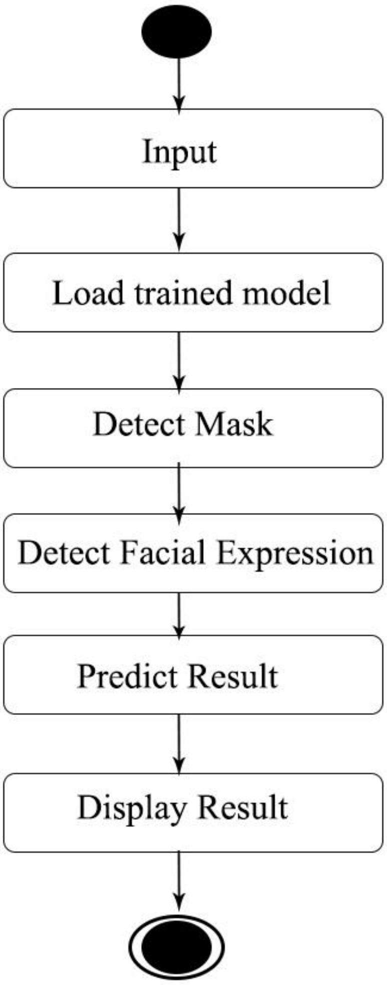
## SEQUENCE DIAGRAM

The sequence diagram represents the flow of messages in the system and is also termed an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time.



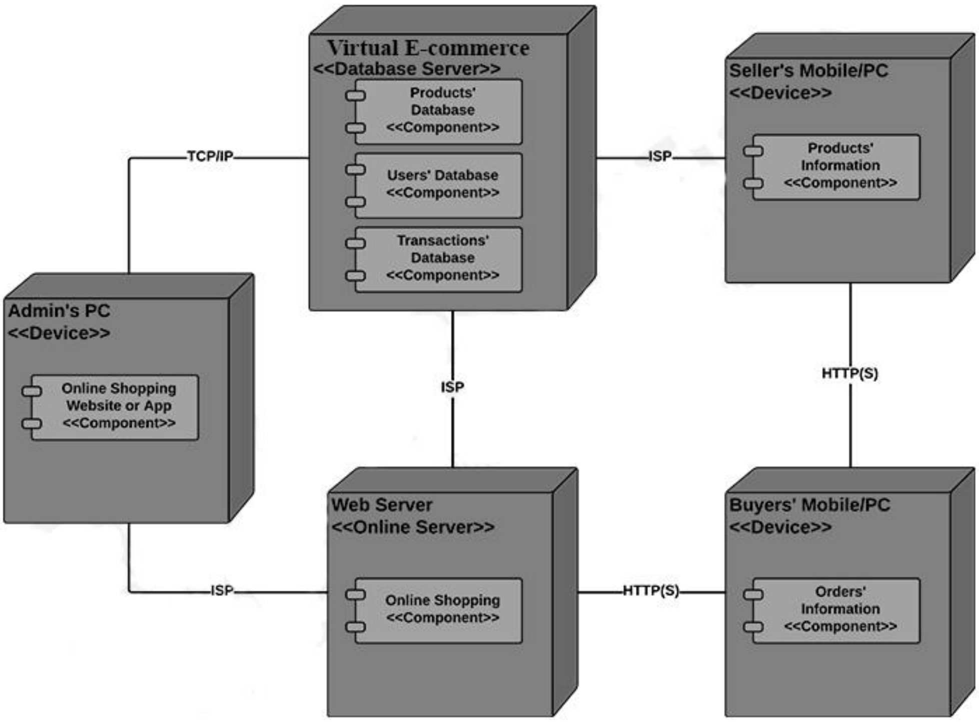
## STATE CHART DIAGRAM

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination.



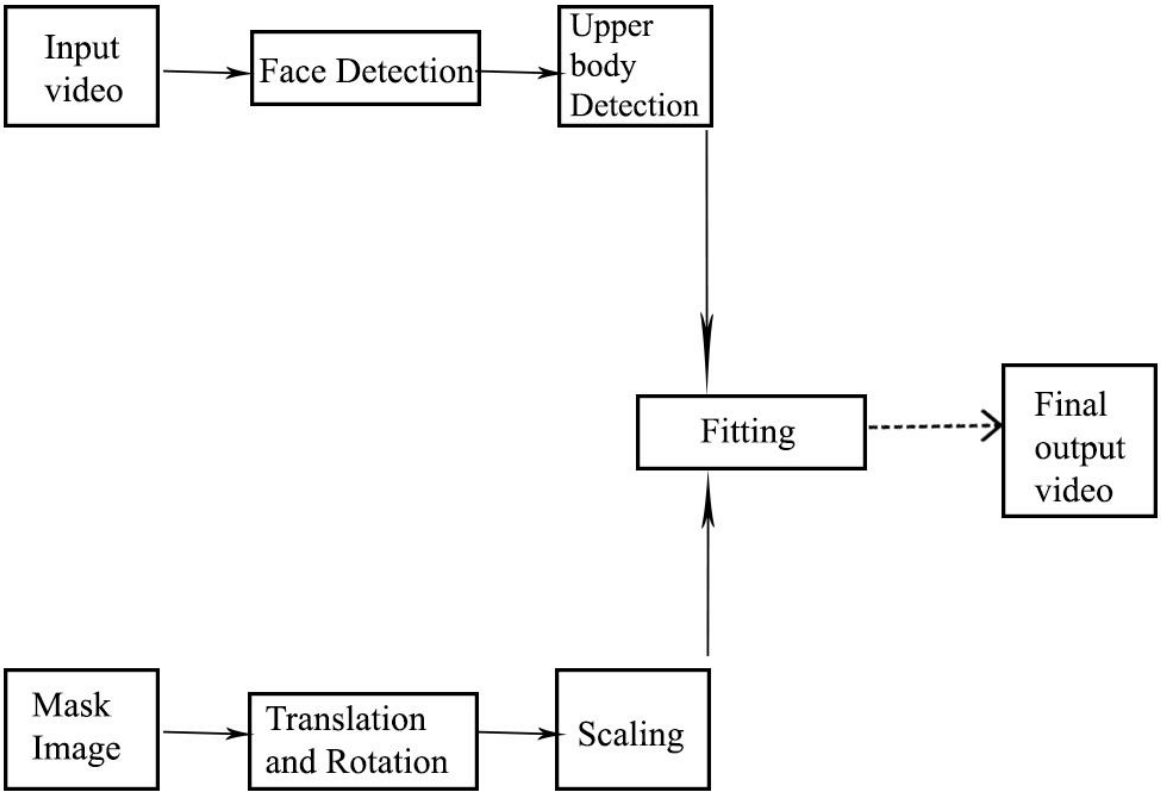
# DEPLOYMENT DIAGRAM

A UML deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modeling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system (topology of the hardware).


Virtual E-commerce
<<Database Server>>
Products' Database <<Component>>
Users' Database <<Component>>
Transactions' Database <<Component>>
Seller's Mobile/PC
<<Device>>
Products' Information <<Component>>
Admin's PC
<<Device>>
Online Shopping Website or App <<Component>>
Web Server
<<Online Server>>
Online Shopping <<Component>>
Buyers' Mobile/PC
<<Device>>
Orders' Information <<Component>>
TCP/IP
ISP
ISP
HTTP(S)
ISP
HTTP(S)

# DATA FLOW DIAGRAM

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops.



# 3. IMPLEMENTATIONS

## INSTALLATION OF PACKAGES IN PYTHON:

- Pip install opencv
- Pip install dlib
- Pip install cmake
- Pip install numpy
- Pip install gunicorn
- Pip install scipy
- Pip install imutils

## Haarcasecade xml files

Haarcasecade_eye.xml, haarcasecade_fullbody.xml, haarcasecade_frontalface.xml

## How to train haar cascade dataset ?

- Opencv provides some pre trained haar cascade and for creating your own haar cascade there are three methods are available.
- File provided by Auckland university to create haar cascade using command prompt but it is very tedious process and not that much of accurate(training time is more).
- Method provided by youtuber sendex but actually it does not uses windows to create haar cascade it uses aws ($5/month).
- Amin ahemadi gui creater for creating haar cascade (cascade trainer gui).
- Haar feature(white black boxes moving on to the image and extract the feature.)
- Integrall images approach(to make fasten the process of computation of data)
- Adaboost classifier (extract best feature and discarding non-relevant feature)

## Test.py

```
from tkinter_scroll import *
from tkinter import *
from PIL import Image
from PIL import ImageTk
import cv2, threading, os, time
from threading import Thread
from os import listdir
from os.path import isfile, join
import dlib
from imutils import face_utils, rotate_bound
import math
import sys

ACTIVE_IMAGES=[0 for i in range(10)]
def put_sprite(num, k):
    global SPRITES, BTNS
    SPRITES[num] = (1 - SPRITES[num])
    if SPRITES[num]:
        ACTIVE_IMAGES[num] = k
        BTNS[num].config(relief=SUNKEN)
    else:
        BTNS[num].config(relief=RAISED)

def draw_sprite(frame, sprite, x_offset, y_offset):
    (h,w) = (sprite.shape[0], sprite.shape[1])
    (imgH,imgW) = (frame.shape[0], frame.shape[1])
    if y_offset+h >= imgH:
        sprite = sprite[0:imgH-y_offset,:,:]
    if x_offset+w >= imgW:
        sprite = sprite[:,0:imgW-x_offset,:]
    if x_offset < 0:
        sprite = sprite[:,abs(x_offset)::,:]
        w = sprite.shape[1]
        x_offset = 0
    for c in range(3):
            frame[y_offset:y_offset+h, x_offset:x_offset+w, c] =  \
            sprite[:,:,c] * (sprite[:,:,3]/255.0) +  frame[y_offset:y_offset+h, x_offset:x_offset+w, c] *
(1.0 - sprite[:,:,3]/255.0)
    return frame

def adjust_sprite2head(sprite, head_width, head_ypos, ontop = True):
    (h_sprite,w_sprite) = (sprite.shape[0], sprite.shape[1])
    factor = 1.0*head_width/w_sprite
    sprite = cv2.resize(sprite, (0,0), fx=factor, fy=factor)
    (h_sprite,w_sprite) = (sprite.shape[0], sprite.shape[1])
    y_orig =  head_ypos-h_sprite if ontop else head_ypos
    if (y_orig < 0):
            sprite = sprite[abs(y_orig)::,:,:]
            y_orig = 0
    return (sprite, y_orig)

def apply_sprite2feature(image, sprite_path, haar_filter, x_offset, y_offset, y_offset_image,
adjust2feature, desired_width, x, y, w, h):
    sprite = cv2.imread(sprite_path,-1)
    (h_sprite,w_sprite) = (sprite.shape[0], sprite.shape[1])

    xpos = x + x_offset
    ypos = y + y_offset
    factor = 1.0*desired_width/w_sprite

    sub_img = image[y + y_offset_image:y+h,x:x+w,:]

    feature = apply_Haar_filter(sub_img, haar_filter, 1.3 , 10, 10)
    if len(feature)!=0:
        xpos, ypos = x, y + feature[0,1] #adjust only to feature in y axis (eyes)
        if adjust2feature:
            size_mustache = 1.2 #how many times bigger than mouth
            factor = 1.0*(feature[0,2]*size_mustache)/w_sprite
            xpos =  x + feature[0,0] - int(feature[0,2]*(size_mustache-1)//2) #centered respect to width
            ypos = y + y_offset_image + feature[0,1] - int(h_sprite*factor) #right on top

    sprite = cv2.resize(sprite, (0,0), fx=factor, fy=factor)
    image = draw_sprite(image,sprite,xpos,ypos)

def apply_sprite(image, path2sprite,w,x,y, angle, ontop = True):
    sprite = cv2.imread(path2sprite,-1)
    sprite = rotate_bound(sprite, angle)
    (sprite, y_final) = adjust_sprite2head(sprite, w, y, ontop)
    image = draw_sprite(image,sprite,x, y_final)

def calculate_inclination(point1, point2):
    x1,x2,y1,y2 = point1[0], point2[0], point1[1], point2[1]
    incl = 180/math.pi*math.atan((float(y2-y1))/(x2-x1))
    return incl

def calculate_boundbox(list_coordinates):
    x = min(list_coordinates[:,0])
    y = min(list_coordinates[:,1])
    w = max(list_coordinates[:,0]) - x
    h = max(list_coordinates[:,1]) - y
    return (x,y,w,h)

def apply_Haar_filter(img, haar_cascade,scaleFact = 1.05, minNeigh = 3, minSizeW = 30):
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    features = haar_cascade.detectMultiScale(
        gray,
        scaleFactor=scaleFact,
        minNeighbors=minNeigh,
        minSize=(minSizeW, minSizeW),
        flags=cv2.CASCADE_SCALE_IMAGE
    )
    return features

def get_face_boundbox(points, face_part):
    if face_part == 1:
        (x,y,w,h) = calculate_boundbox(points[17:22])
    elif face_part == 2:
        (x,y,w,h) = calculate_boundbox(points[22:27])
    elif face_part == 3:
        (x,y,w,h) = calculate_boundbox(points[36:42])
    elif face_part == 4:
        (x,y,w,h) = calculate_boundbox(points[42:48])
    elif face_part == 5:
        (x,y,w,h) = calculate_boundbox(points[29:36])
    elif face_part == 6:
        (x,y,w,h) = calculate_boundbox(points[1:17])
    elif face_part == 7:
         (x,y,w,h) = calculate_boundbox(points[0:6])
    elif face_part == 8:
        (x,y,w,h) = calculate_boundbox(points[11:17])
    return (x,y,w,h)

def cvloop(run_event):
    global ctr_mid
    global SPRITES
    i = 0
    video_capture = cv2.VideoCapture(0)
    video_capture.set(3,2048)
    video_capture.set(4,2048)
    (x,y,w,h) = (0,0,10,10)
    detector = dlib.get_frontal_face_detector()
    fullbody = cv2.CascadeClassifier('data/haarcascade_fullbody.xml')
    model = "data/shape_predictor_68_face_landmarks.dat"
    predictor = dlib.shape_predictor(model) # link to model:
http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2
    while run_event.is_set():
        ret, image = video_capture.read()
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        faces = detector(gray, 0)
        for face in faces:
            (x,y,w,h) = (face.left(), face.top(), face.width(), face.height())
            shape = predictor(gray, face)
            shape = face_utils.shape_to_np(shape)
            incl = calculate_inclination(shape[17], shape[26])
            is_mouth_open = (shape[66][1] -shape[62][1]) >= 10

            if SPRITES[3]:#Tiara
                apply_sprite(image, IMAGES[3][ACTIVE_IMAGES[3]],w+45,x-20,y+20, incl, ontop =
True)

            #Necklaces
            if SPRITES[1]:
                (x1,y1,w1,h1) = get_face_boundbox(shape, 6)
                apply_sprite(image, IMAGES[1][ACTIVE_IMAGES[1]],w1,x1,y1+125, incl,  ontop =
False)

            #Goggles
            if SPRITES[6]:
                (x3,y3,_,h3) = get_face_boundbox(shape, 1)
                apply_sprite(image, IMAGES[6][ACTIVE_IMAGES[6]],w,x,y3-5, incl, ontop = False)

            #Earrings
            (x0,y0,w0,h0) = get_face_boundbox(shape, 6) #bound box of mouth
            if SPRITES[2]:
                (x3,y3,w3,h3) = get_face_boundbox(shape, 7) #nose
                apply_sprite(image, IMAGES[2][ACTIVE_IMAGES[2]],w3,x3-40,y3+30,
incl,ontop=False)
                (x3,y3,w3,h3) = get_face_boundbox(shape, 8) #nose
                apply_sprite(image, IMAGES[2][ACTIVE_IMAGES[2]],w3,x3+40,y3+75, incl)

#             if SPRITES[5]:
#                 apply_sprite(image, IMAGES[5][ACTIVE_IMAGES[5]],w,x,y, incl, ontop = True)



            #Tops
            if SPRITES[4]:
                # (x,y,w,h) = (0,0,10,10)
                # apply_sprite2feature(image, IMAGES[7][ACTIVE_IMAGES[7]], fullbody, w//4,
2*h//3, h//2, True, w//2, x, y, w, h)
                (x1,y1,w1,h1) = get_face_boundbox(shape, 8)
                apply_sprite(image, IMAGES[4][ACTIVE_IMAGES[4]],w1+350,x1-230,y1+100,
incl,  ontop = False)

        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        image = Image.fromarray(image)
        image = ImageTk.PhotoImage(image)
        ctr_mid.configure(image=image)
        ctr_mid.image = image

    video_capture.release()

root = Tk()
root.title('TryOn')
app=FullScreenApp(root)

top_frame = Frame(root, bg='#077bd4', width=50, height=50, pady=3)
center = Frame(root, bg='white', width=50, height=40, padx=3, pady=3)
btm_frame = Frame(root, bg='#077bd4', width=50, height=50, pady=3)
```

```
root.grid_rowconfigure(1, weight=1)
root.grid_columnconfigure(0, weight=1)

top_frame.grid(row=0, sticky="ew")
center.grid(row=1, sticky="nsew")
btm_frame.grid(row=4, sticky="ew")

center.grid_rowconfigure(0, weight=1)
center.grid_columnconfigure(1, weight=1)

ctr_left = Label(center,bg='white', width=50, height=190)
ctr_mid = Label(center,bg='white',width=100, height=160, padx=0, pady=0)

ctr_left.grid(row=0, column=0, sticky="ns")
ctr_mid.grid(row=0, column=1, sticky="nsew")

scrollable_body = Scrollable(ctr_left, width=15)
SPRITES=[0 for i in range(10)]
BTNS=[]
IMAGES ={i:[] for i in range(10)}
PHOTOS ={i:[] for i in range(10)}
for img in sys.argv[1:]:
    IMAGES[int(img.rsplit('/',1)[0][-1])].append(img)
    image=ImageTk.PhotoImage(Image.open(img).resize((150,100)))
    PHOTOS[int(img.rsplit('/',1)[0][-1])].append(image)
for index in range(9):
    if len(PHOTOS[index]) > 0:
        for k,photo in enumerate(PHOTOS[index]):
            btn= Button(scrollable_body, highlightbackground='white',
text=IMAGES[index][k].rsplit('/',1)[1].replace('.png',")[:-1],bg='white', image=photo, command =
lambda index=index, k=k: put_sprite(index,k), compound=LEFT, width='300', height='200')
            btn.pack(side="top", fill="both", expand="no", padx="5", pady="5")
            BTNS.append(btn)
```

```
scrollable_body.update()

run_event = threading.Event()
run_event.set()
action = Thread(target=cvloop, args=(run_event,))
action.setDaemon(True)
action.start()

def terminate():
        global root, run_event, action
        run_event.clear()
        time.sleep(1)
        root.destroy()
root.protocol("WM_DELETE_WINDOW", terminate)
root.mainloop()
```

## Camera.py

This camera.py is used to capture the user video so, after capturing the user video the internal process will the fitting the clothes on to the user

```
import cv2

class VideoCamera(object):
    def __init__(self):
        # Using OpenCV to capture from device 0. If you have trouble capturing
        # from a webcam, comment the line below out and use a video file
        # instead.
        self.video = cv2.VideoCapture(0)
        # If you decide to use video.mp4, you must have this file in the folder
        # as the main.py.
        # self.video = cv2.VideoCapture('video.mp4')

    def __del__(self):
        self.video.release()

    def get_frame(self):
        success, image = self.video.read()
        # We are using Motion JPEG, but OpenCV defaults to capture raw images,
        # so we must encode it into JPEG in order to correctly display the
        # video stream.
        ret, jpeg = cv2.imencode('.jpg', image)
        return jpeg.tobytes()
```

## main.py

- We were used flask micro framework as a interface between backend(python) and front end(html).And for gui we used tkinter library which was written in python for design user interface.
- @app.route("/") it means whenever the home page is found involk the function below it under that function in the end of that function you are returning a function render_template("html file") and whatever html file is given in the argument that html page will be shown in the browser for that this html file should be available in template folder
- @app.route("/success",methods=["POST"]
- it means whenever the route is scuccess then involk the function below it post shows that we are sending data to server

```
#!/usr/bin/env python3
# from tryOn import TryOn as tryOn

from flask import Flask, render_template, Response,redirect,request
from camera import VideoCamera
import os
app = Flask(__name__)



@app.route('/<file_path>',methods = ['POST', 'GET'])
def tryon(file_path):
    file_path = file_path.replace(',','/')
    os.system('python test.py ' + file_path)
    return redirect('http://127.0.0.1:5000/',code=302, Response=None)



@app.route('/')
def index():
    return render_template('index.html')
```

```
def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route("/cart/<file_path>",methods = ['POST', 'GET'])
def cart(file_path):
    global CART
    file_path = file_path.replace(',','/')
    print("ADDED", file_path)
    CART.append(file_path)
    return render_template("checkout.html")

@app.route('/video_feed')
def video_feed():
    return Response(gen(VideoCamera()),
                    mimetype='multipart/x-mixed-replace; boundary=frame')

if __name__ == '__main__':
    app.run()
    app.run()
```

For User interface we have used HTML, CSS, and Javascript.

# 4. TESTING

Testing is the debugging program is one of the most critical aspects of the computer programming triggers, without programming that works, the system would never produce an output of which it was designed. Testing is best performed when user development is asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quantity but quality of the data used the matters of testing. Testing is aimed at ensuring that the system was accurately an efficiently before live operation commands.

## Testing objectives:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, testing is a process of executing a program with intent of finding an error.

- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has probability of finding an error, if it exists.
- The test is inadequate to detect possibly present errors.
- The software more or less confirms to the quality and reliable standards.

## Levels of Testing:

In order to uncover present in different phases we have the concept of levels of testing.

## The basic levels of Testing:

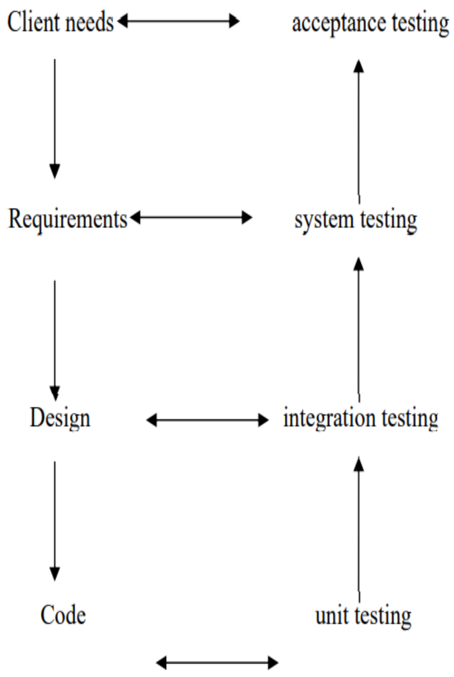


Fig 4.1: Levels of Testing.

## Code testing:

This examines the logic of the program. For example, the logic for updating various sample data and with the sample files and directories were tested and verified. Specification Testing: Executing this specification starting what the program should do and how it should performed under various conditions. Test cases for various situation and combination of conditions in all the modules are tested.

## Unit testing:

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during programming stage itself. In the testing step each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks for fields also. For example the validation check is done for varying the user input given by the user which validity of the data entered. It is very easy to find error debut the system.

Each Module can be tested using the following two Strategies:

1. Black Box Testing
2. White Box Testing

## BLACK BOX TESTING

## What is Black Box Testing?

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

## Black box testing - Steps

Here are the generic steps followed to carry out any type of Black Box Testing.

- Initially requirements and specifications of the system are examined.
- Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.

- Tester determines expected outputs for all those inputs.
- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested.

## Types of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones –

• **Functional testing** – This black box testing type is related to functional requirements of a system; it is done by software testers.

• **Non-functional testing** – This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability.

• **Regression testing** – Regression testing is done after code fixes , upgrades or any other system maintenance to check the new code has not affected the existing code.

## WHITE BOX TESTING

White Box Testing is the testing of a software solution's internal coding and infrastructure.It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability.White box testing is also known as clear, open, structural, and glass box testing.

It is one of two parts of the "box testing" approach of software testing. Its counter-part, blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing. The term "whitebox" was used because of the see-through box concept. The clear box or whitebox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings.

## How do you perform White Box Testing?

To give you a simplified explanation of white box testing, we have divided it into two basic steps. This is what testers do when testing an application using the white box testing technique:

### STEP 1) UNDERSTAND THE SOURCE CODE

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

### Step 2) CREATE TEST CASES AND EXECUTE

The second basic step to white box testing involves testing the application's source code for proper flow and structure. One way is by writing more code to test the application's source code. The tester will develop little tests for each process or series of processes in the application. This method requires that the tester must have intimate knowledge of the code and is often done by the developer. Other methods include manual testing, trial and error testing and the use of testing tools as we will explain further on in this article.

There are three main kinds of System testing:

Alpha Testing

BetaTesting

Acceptance Testing

### Alpha Testing:

This refers to the system testing that is carried out by the test team with the Organization.

### Beta Testing:

This refers to the system testing that is performed by a selected group of friendly customers.

### Acceptance Testing:

This refers to the system testing that is performed by the customer to determine whether or not to accept the delivery of the system.

## Integration Testing:

Data can be lost across an interface, one module can have an adverse effort on the other sub functions, when combined, may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. The testing was done with sample data. The developed system has run successfully for this sample data. The need for integrated test is to find the overall system performance.

## Output testing:

After performance of the validation testing, the next step is output testing. The output displayed or generated by the system under consideration is tested by asking the user about the format required by system. The output format on the screen is found to be correct as format was designed in the system phase according to the user needs. Hence the output testing does not result in any correction in the system.

## Test plan:

The test-plan is basically a list of testcases that need to be run on the system. Some of the testcases can be run independently for some components and some of the testcases require the whole system to be ready for their execution. It is better to test each component as and when it is ready before integrating the components. It is important to note that the testcases cover all the aspects of the system (ie, all the requirements stated in the RS document).

## What do you verify in White Box Testing ?

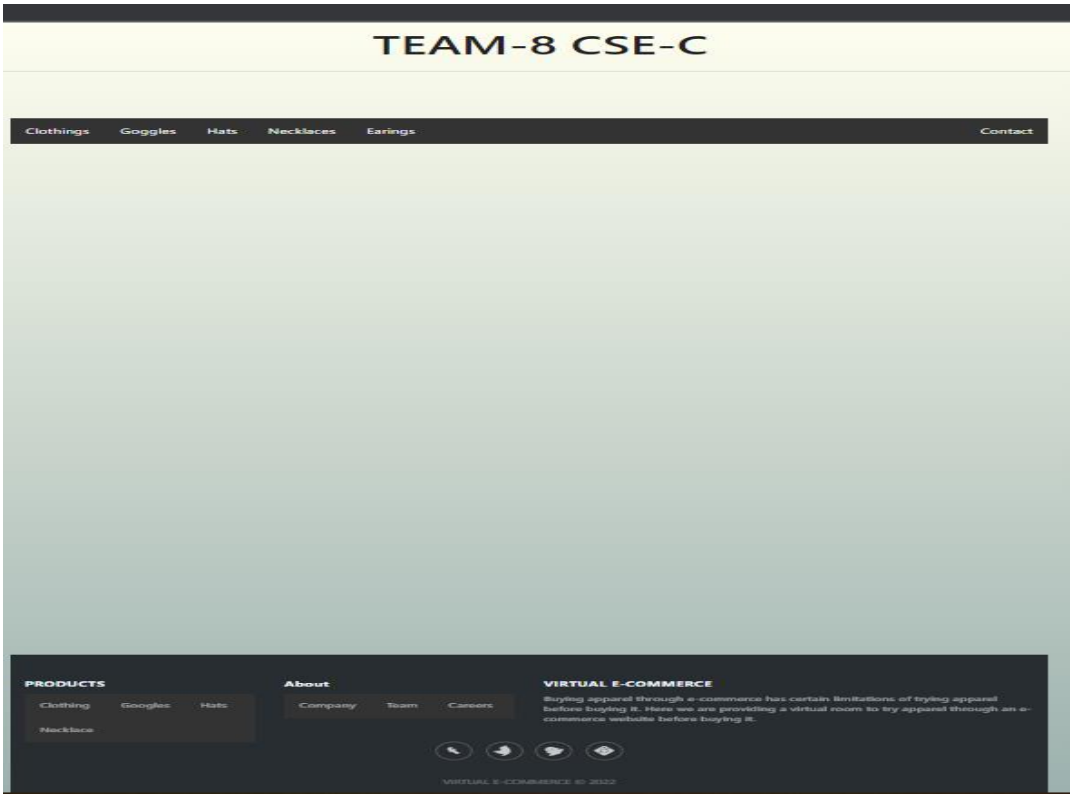
White box testing involves the testing of the software code for the following:

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object and function on an individual basis

| Test Scenario ID | Test Case ID | Test Case | Expected Output | Actual Output | Test Result |
|---|---|---|---|---|---|
| TS001 | TC001 | Verify the T-shirt is on the user. | Should display T-shirt on user | T-shirt on User | Pass |
| TS002 | TC002 | Verify the goggles are on the user's eyes. | Should display Goggles are on the user's eyes. | Goggles on the user's eyes. | Pass |
| TS003 | TC003 | Verify the Hat is on the user's head | Should display The hat is on the user's head. | Hat on user's head. | Pass |
| TS004 | TC004 | Verify the Necklace is on the user's neck. | Should display Necklace on the user's neck. | The necklace is on the user's neck. | Pass |
| TS005 | TC005 | Verify the Earrings is on the user's ear | Should display Earrings on the user's ear. | Earrings is on the user's ear. | Pass |
| TS006 | TC006 | Verify the T-shirt is exact on the user. | Should display T-shirt on user with exact fitting. | T-shirt on User Without exact fitting. | Fail |
| TS007 | TC007 | Verify the goggles are exact on the user's eyes. | Should display Goggles are on the user's eyes with exact fitting. | Goggles on the user's eyes with exact fitting. | Pass |
| TS008 | TC008 | Verify the Hat is on the user's head with exact fitting. | Should display The hat is on the user's head with exact fitting. | Hat on user's head without exact fitting. | Fail |
| TS009 | TC009 | Verify the Necklace is on the user's neck with exact fitting. | Should display Necklace on the user's neck with exact fitting. | The necklace is on the user's neck without exact fitting. | Fail |
| TS010 | TC010 | Verify the Earrings is on the user's ear with exact fitting. | Should display Earrings on the user's ear with exact fitting. | Earrings is on the user's ear without exact fitting. | Fail |

Table 4.1 Testcases and Results

# 5. RESULTS



Buying apparel through e-commerce has certain limitations of trying apparel before buying it. Here we are providing a virtual room to try apparel through an e-commerce website before buying it.

Fig 5.1: User Interface

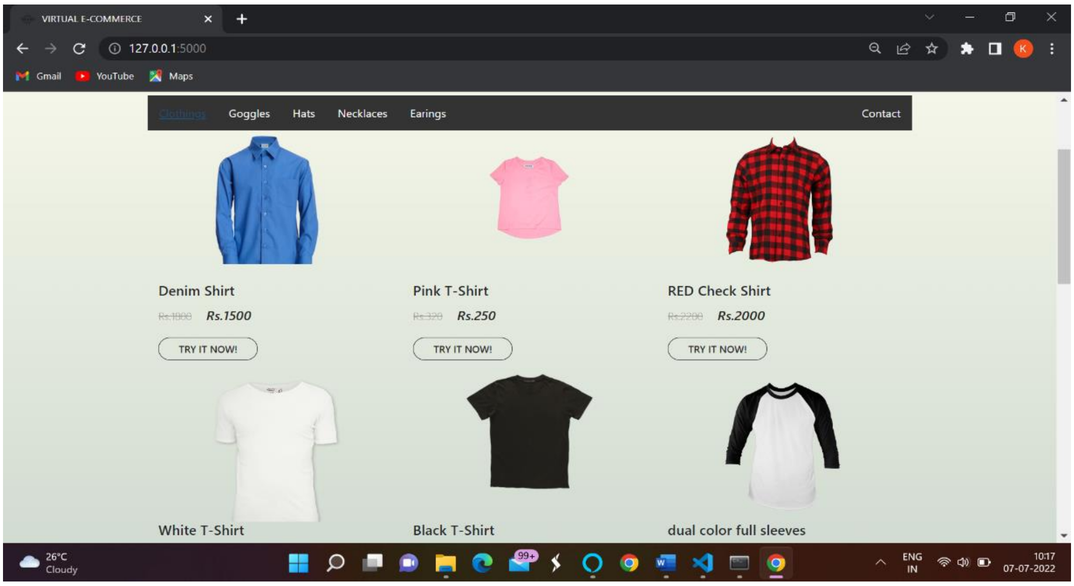


Fig 5.2: User Interface.

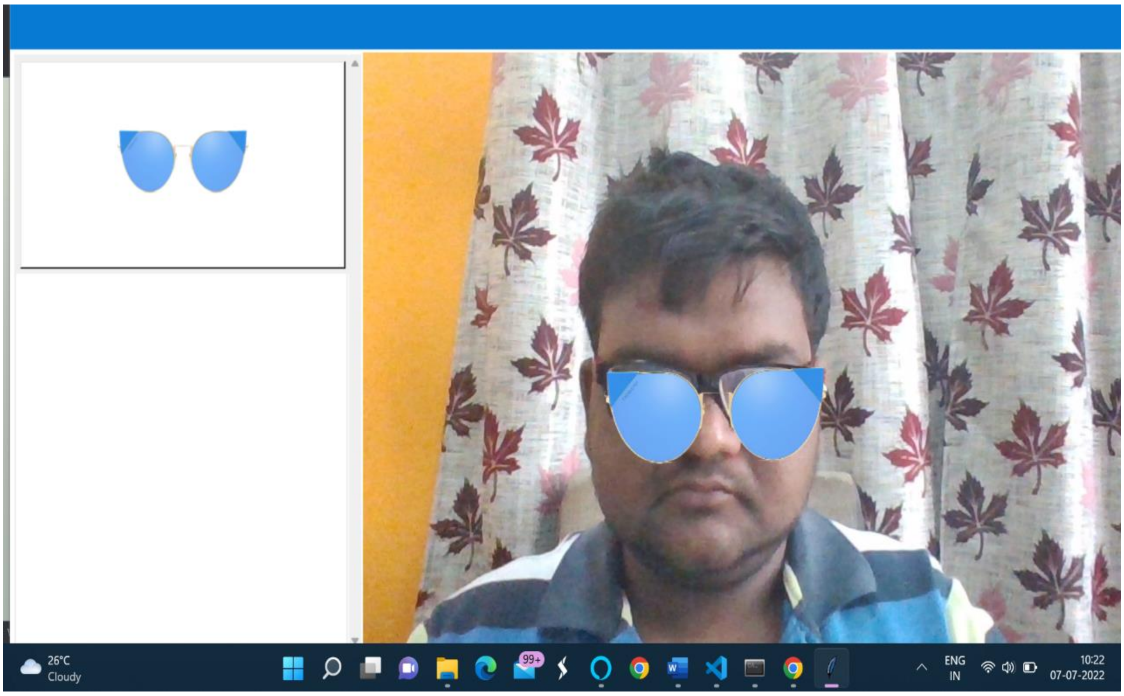


Fig 5.3: Blue Goggle on user’s face.

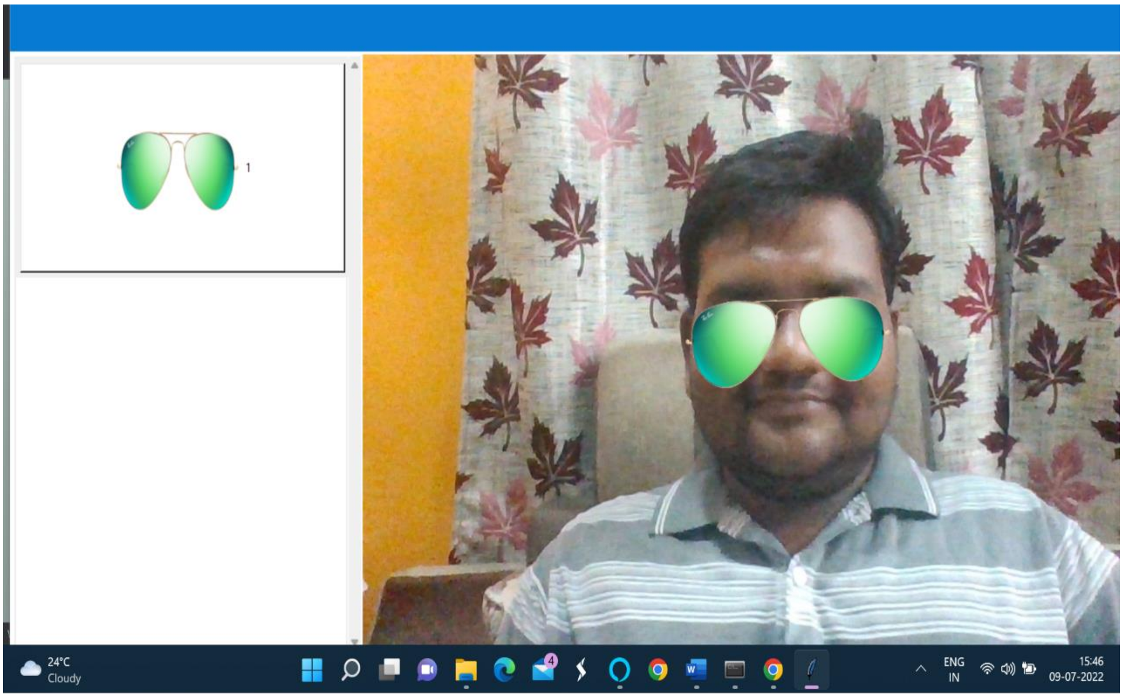


Fig 5.4: Green Shades on user’s face.

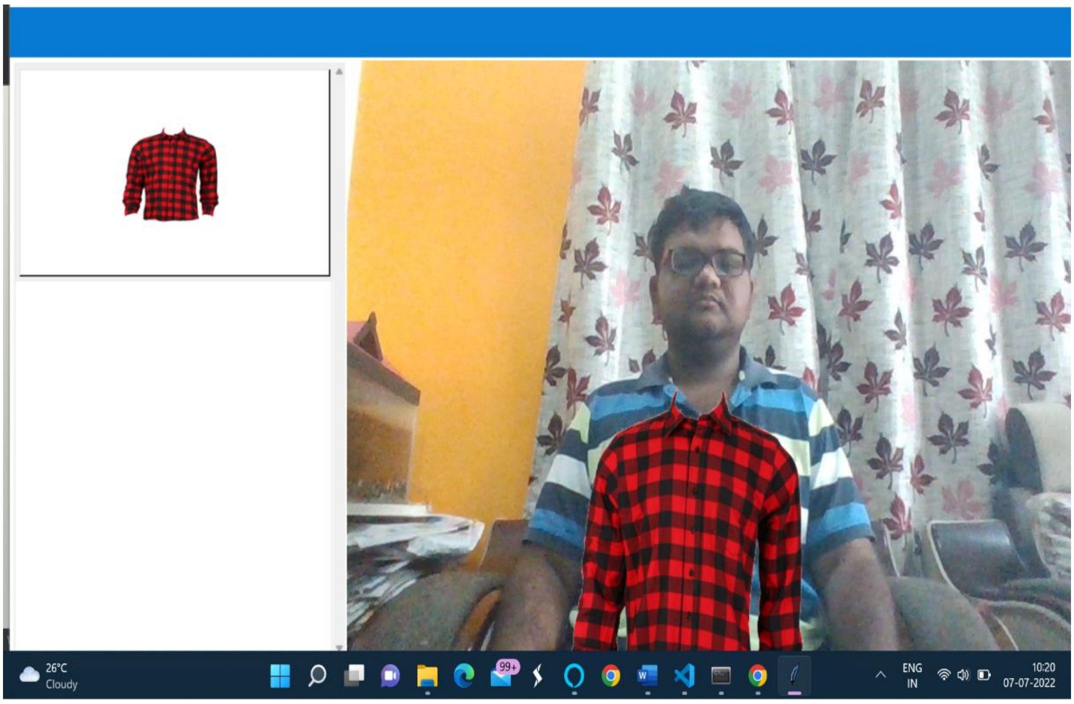


Fig 5.5: Red Shirt on user's body.

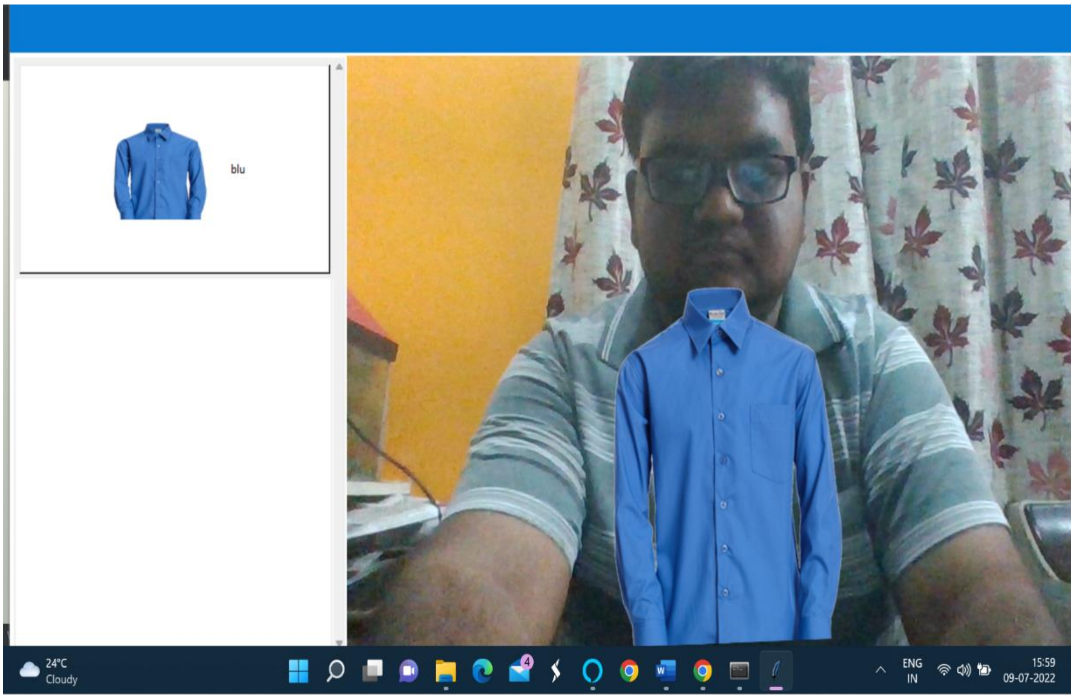


Fig 5.6: Denim shirt on user's body.

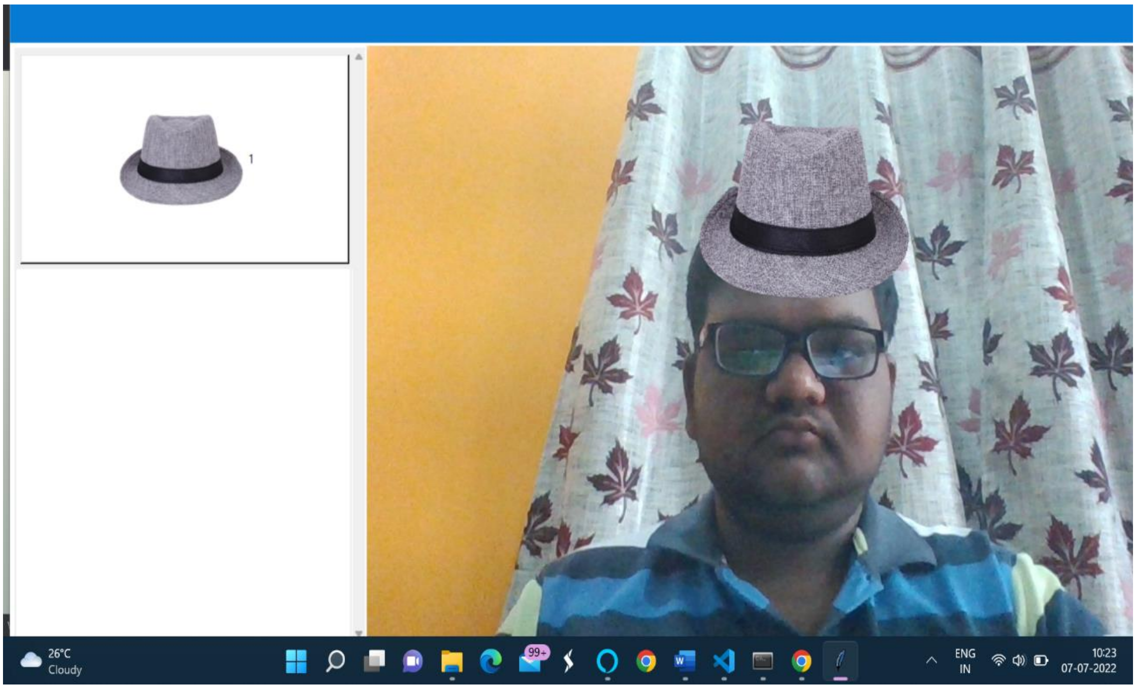


Fig 5.7: Grey Hat on user's head.

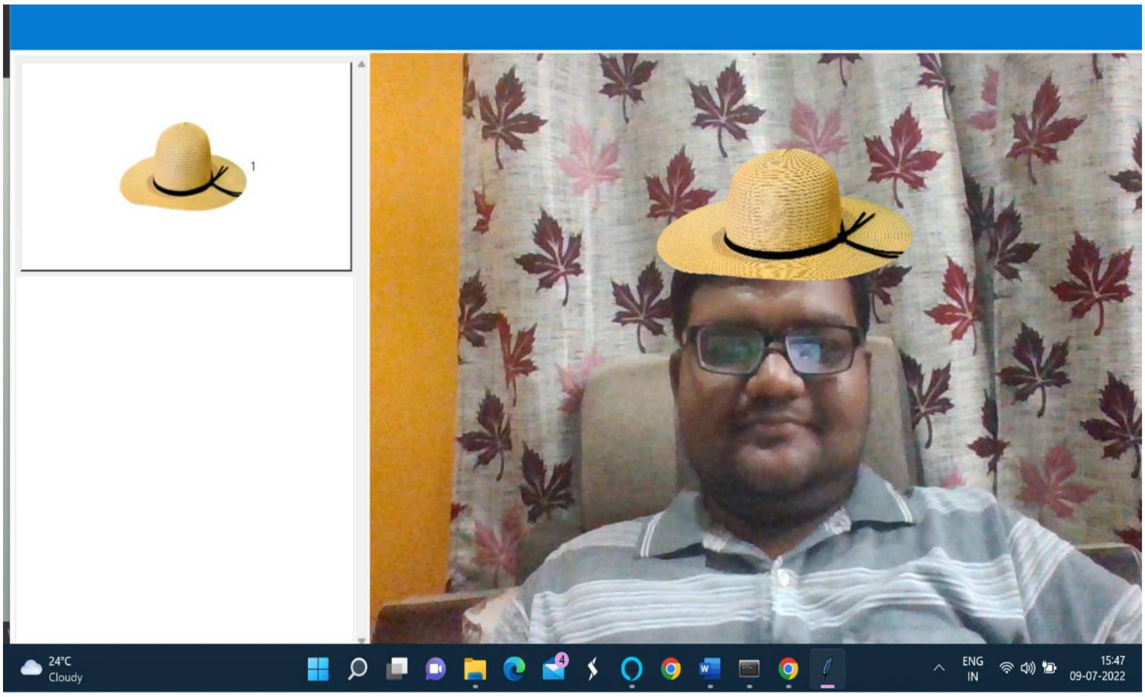


Fig 5.8: C-thread Hat on user's head.

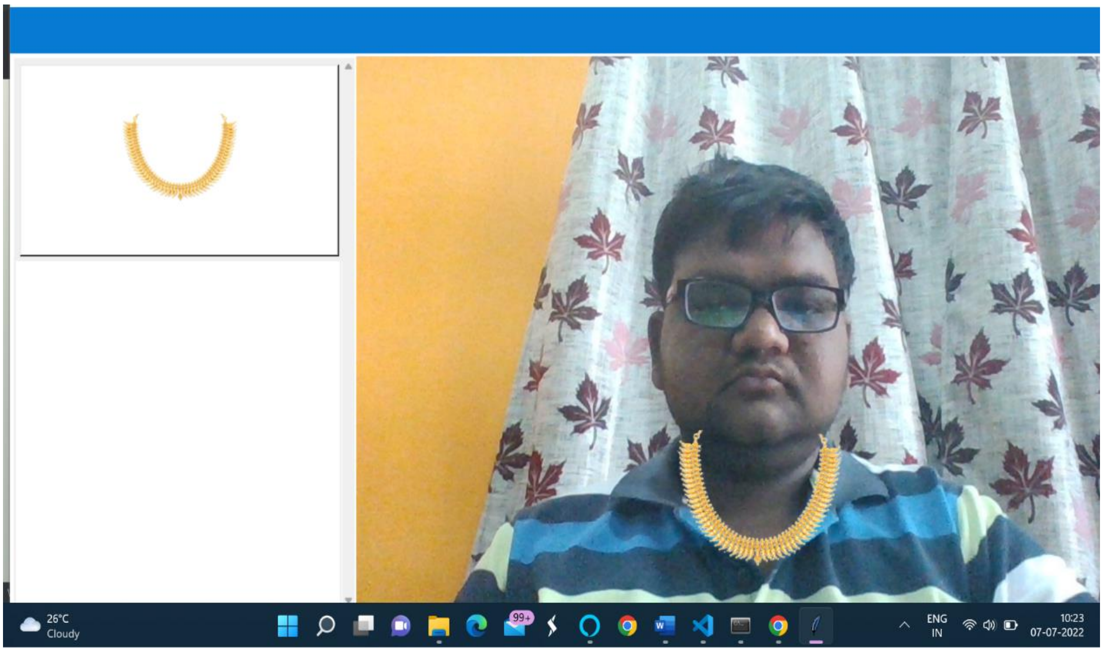


Fig 5.9: Necklace on user’s neck.

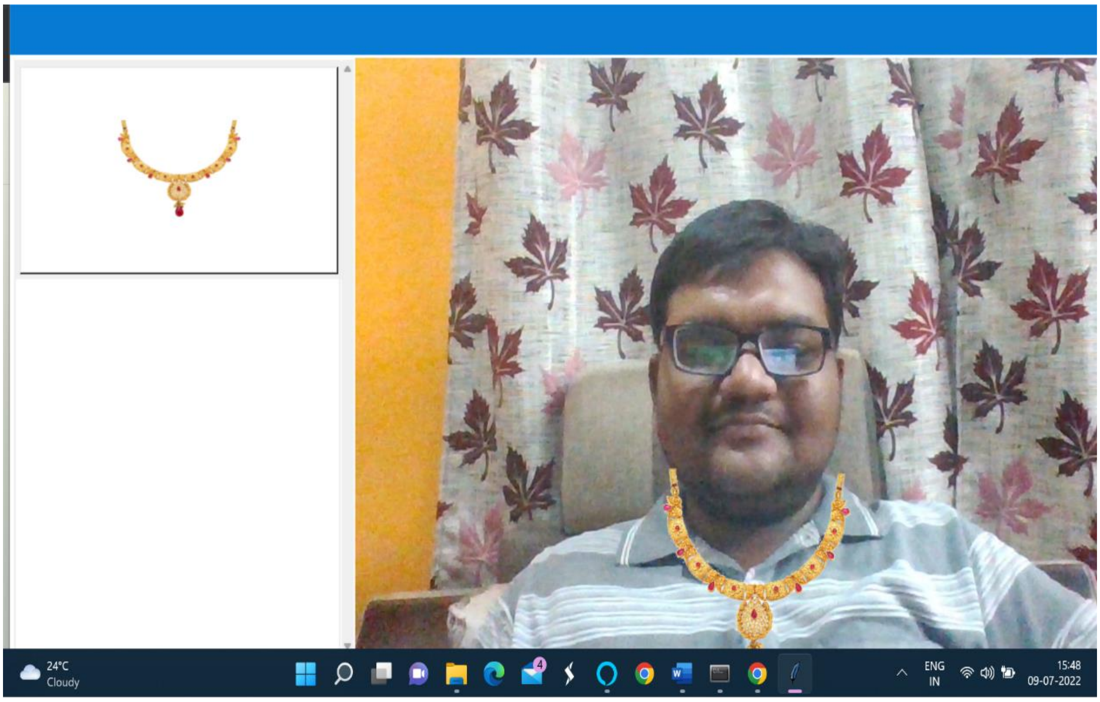


Fig 5.10: Necklace on user’s neck.

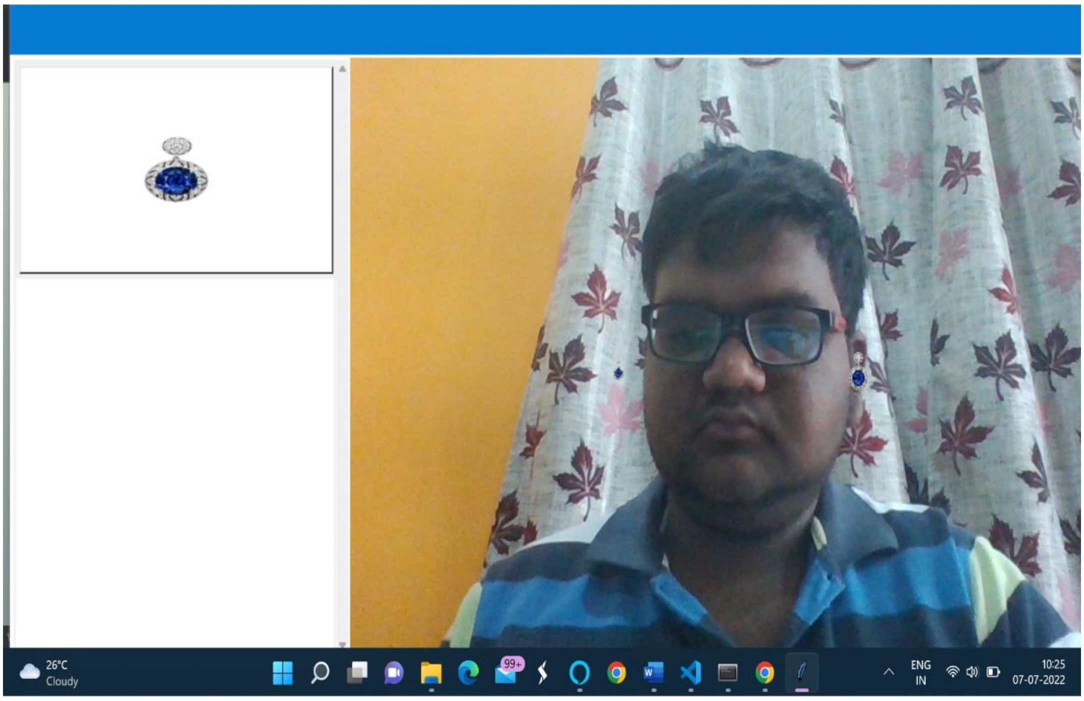


Fig 5.11: Earring on user's ear.

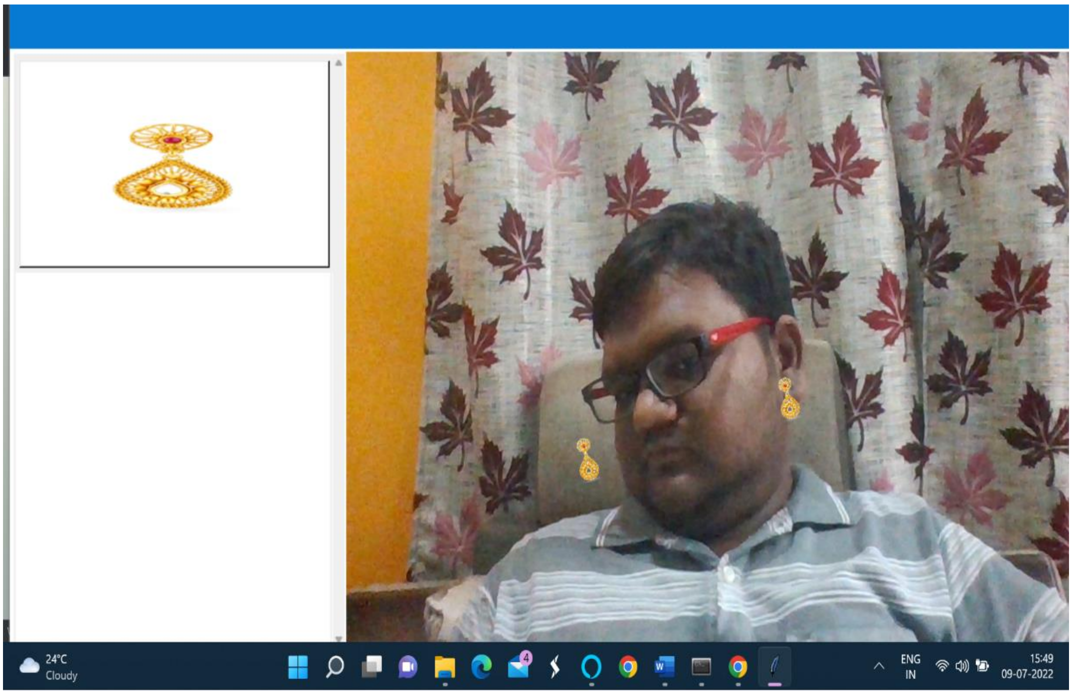


Fig 5.12: Earring on user's ear.

# 5. CONCLUSIONS AND FUTURE SCOPE

## Conclusion

In this paper, we have presented the design and execution of a usability and user experience test of our prototype of a virtual E-commerce. As the final output, system will display the available dresses from store. The user can access the GUI by cursor movements and select the apparel or select various category. Thus it will be very convenient and hassle-free for the customer to choose the perfect apparel through virtual means without muh of inconvenience. In this work we have been focusing on the customer interface of the virtually trying products. Another part of designing and developing a virtual E-commerce is the company interface, where a major concern is how to design and produce digital clothes for the virtually trying products.

## Future Scope

- After detecting body parts of target image there are certain operation we need to perform for make this accurate.
- So for this we need to train three type of network through which we can set clothes on target image.
    - PAN (pose alignment network)
    - TRN (texture refinement network)
    - FTN (fitting network)

# 7.BIBLIOGRAPHY

[1] Yashwant Kanduri, Sanugula Durga Prasad, “A modular Approach for Cloth Modelling in Virtual Fitting Room”, 2016 11th International Conference on Industrial and Information systems(ICIIS)

[2] Anagaha Ramesh, Ankit Raj Kushal, Brinda D, Vaishnavi S, Purohit Shrinivasacharya, “3D Virtual Trial Room”, Volume 6/Issue 13, International Journal of Engineering Research &Technology(IJERT)

[3] Vinaya Kulkarni, Sailee Morde, Bhairavi Pawar, Supriya Mahadik, Rutuja Dahore, “2D Virtual Trial Room Using Augmented Reality”, Volume 4/ Issue 1, International Journal on Future Revolution in Computer Science Engineering.

[4] Shreya Kamani, Neel vasa, Kriti Shrivastava “Virtual Trial Room Using Augmented Reality”, Volume 3, Number 6, International Journal of Advanced Computer Technology (IJACT)

[5] [online] https://en.wikipedia.org/wiki/OpenCV

[6] Srinivasan K.,Vivek S., "Implementation of virtual fitting room using image processing" , 2017 IEEE International Conference on Computer, Communication and Signal Processing(ICCSP2017)