# SUPERFIT APP AND WEBSITE

**A REPORT ON PROJECT BASED LEARNING**

**(SEMESTER -II)**

*Submitted by*


## NAME OF THE CANDIDATE(S)

SAGAR ABHYANKAR-10703

OM GUND-10705

SAMIR HENDRE–10708

MRUNAL GOLIVADEKAR-

10710

MUGDHA MALPE - 10713

**FIRST YEAR ENGINEERING**




**Society for Computer Technology and Research's**

## PUNE INSTITUTE OF COMPUTER TECHNOLOGY

**DHANKAWADI, PUNE – 43**


**A.Y. 2020-21**

# - CERTIFICATE-

This is to certify that the work incorporated in the report entitled **"SUPERFIT APP AND WEBSITE"** is carried out by **Mr. Abhyankar Sagar Ravindra (10703),** who is part a group of students with Project Id: **3A72021** under the subject ***Project Based Learning*** during A.Y. 2020-2021. Such material has not been submitted to any other University/Institute for any financial support. The literature related to the problem investigated has been appropriately cited and duly acknowledged wherever facilities and suggestions have been availed of.

**Date:** **Name & Sign of Project Guide**

**Place:** PUNE

**Name & Sign of PBL Coordinator** **Name & Sign of Head of Department**

**Prof. N. P. Sapkal** **Prof. E. M. Reddy**

# Abstract

In India, the recent Covid-19 pandemic has led to the downfall of the clothesline shopping industry due to restrictions being imposed on the physical transaction of clothing goods. This has created a rift between the supply and demand of these goods and has in turn led to the rise in popularity of online shopping giants such as Amazon, as people turn to them for their clothing needs. The problem now arises from the fact that these online clothing brands have their own size charts for clothes and don't follow a universal size chart. Hence, people get confused as to which sizes will fit them and end up getting dissatisfied if the clothes don't fit. The other problem is the negative environmental impact caused by the constant shipping and returning of these goods. Our SuperFit App and Website system aims to solve this problem by making the user use the camera on their device to automatically calculate their measurements and redirect them to the shopping website by only displaying results of clothes that we know will be a SuperFit for them. This will help encourage more people to shop online and stay home thereby not only preventing the spread of Covid-19 but also helping lessen the negative environmental impact by ensuring unnecessary returns don't take place. This System that we have developed can be used for the good of our citizens and can also help the environment in the long run.

# ACKNOWLEDGEMENT

I would like to express my gratitude towards Prof. Makrand Dinanath Bandkar Sir who gave me the chance to work on our project “SUPERFIT APP AND WEBSITE”. Timely guidance of the Sir inspired us to work on this project. Sir gave many different ideas which proved to be very helpful for this project. Sir also inspected our work and made continuous evaluation of the project. I would like to thank Sir for his remarkable role in completion of our project. Also, I would like to thank my brilliant group members who have contributed to this project in such crucial ways that the whole project could be completed in very short time period.


SAGAR RAVINDRA ABHYANKAR

Place: PUNE Name of Student (in Capital) &Sign

# TABLE OF CONTENT

## LIST OF TABLES

# LIST OF FIGURES

# Nomenclature

| Notation | Meaning |
|---|---|
| **URI** | Uniform Resource Locater |
| **HTML** | Hypertext Markup Language |
| **JSON** | JavaScript Object Notion |
| **CSS** | Cascading Style Sheets |
| **DICT** | Dictionary |
| **XML** | Extensible Markup Language |
| **AJAX** | Asynchronous JavaScript and XML |
| **TF** | TensorFlow |
| **CSRF** | Cross Site Request Forgery |

## Chapter 1
# INTRODUCTION

The Internet has become a crucial part of all of daily lives and this statement is duly highlighted by the intense and unknown situation we find ourselves in today. The Covid-19 pandemic has brought into effect the importance of technology at a scale more profound than ever before. We have seen a vast number of businesses shift their entire operations online in the last couple of years and an even greater number of new online businesses emerge. The pandemic has only contributed to the rapid shift towards mobile technology and its applications.

As we all know, buying of goods has become inherently more difficult due to societal social distancing norms and physical shops being closed down for most of the day in our country. One of the most affected domains in our country has been the shopping industry. Malls and shopping complexes were forcibly closed down due to them being high population density and Covid-19 risk regions and people were unable to buy goods such as clothes and garments, electronics, hardware supplies, and other such materials.

This created a supply and demand gap in the industry which continued to grow exponentially in the early days of the pandemic especially when a total lockdown was in place. Thus, people turned to online shopping company giants such as Amazon, Flipkart, Myntra, AJIO, etc. to satisfy their demand for material goods.

India's e-commerce order volume has increased by 36% in the last quarter of 2020 with over 330 million online buyers across the entire country. Around 71% of the total online Indian online audience has purchased goods online and the size of the market has not stopped increasing.

The Apparel and clothing industry is a special section of the online shopping model which has been booming due to the explosive growth of users across Amazon, Myntra, etc.

Since people do not have access to physical apparel retail stores coupled with the ever-increasing availability in varieties across thousands of clothing and apparels brands such as Nike, Adidas, Puma, etc. offered by these websites along with good discounts and huge sales, has created a lucrative incentive for the audience to purchase clothing goods online.

The problem in the online shopping industry arises now. The thousands of different online brands offer multiple different size charts for the same clothes and apparels, which makes for very inconsistent sizing across the board for the entire website.

For example, on AJIO, the clothing brand Puma offers a size Small (S) shirt with 38cm shoulder width whereas another clothing brand Superdry offers a similar size Small (S) shirt with a 40.6 cm shoulder width.

This discrepancy in sizes on the same website combined with the untold variety in brands that the website offers creates a real issue for the consumers who are buying the product.

Most of the time, the consumers don't even know what their sizes are and have to check every time they make a purchase online. In physical stores, an employee usually does this task by taking the customers measurements and telling them which size clothes to buy from which brand. Pune Central Mall is a great example of this practice. Due to the shopping having move online, the responsibility of finding the right size and knowing your own size is shifted to the consumer and this creates confusion in the minds of the user who is buying the clothing goods. This is especially true for those users who are not tech savvy and don't know how to operate electronics well, such as the elderly.

Often this leads to the shirts the consumer ordered not being a fit for them and them having to return them and try again. This not only dissuades them from purchasing online again but also creates a hassle for the website company who ends up taking a loss by giving a refund, not to mention the negative impact created on the environment caused by the unnecessary shipping of the product back and forth between the consumer and the company leading to the release of huge amounts of CO2 by the shipping vehicles. The scale of this issue is not small and thousands, if not tens of thousands of orders are returned every day. Almost 46% of returns to non-amazon merchants and 34% of returns to amazon merchants are due to the size of the clothing or apparel not being a match.

This is the problem created in the online shopping industry. To solve this problem, we have introduced our SuperFit app and website, which will help us not only create greater customer satisfaction but also help us offset some of the carbon footprint generated by the shipping and shopping industry.

The SuperFit app and website work in conjunction to help the user automatically know the size of their body measurements and redirect them to a website such as Amazon with all of their exact measurements being stored with us on our servers. That way neither do they have to calculate their own measurements and nor do they have to sit and search which brand size will fit them perfectly.

All of these calculations are performed on the website with the help of the app automatically, using a proprietary, in-house developed algorithm which can not only take the calculations of measurements of almost all your body parts but can also sort them through our database of brand sizes and display the results accordingly. That way our app helps you find the clothes that will fit your size perfectly, hence ensuring a SuperFit.

The Project work has been divided into mainly three parts:

**Part 1: Machine Learning Model**

This is the part of the project that deals with the actual computations that need to be performed. It takes in an image input consisting of a person and using Machine learning algorithms extracts the important parts of the body that need to be measured.

This model was developed by Om Gund (Roll No 10705) and works solely on the python programming language.

Once the measurements of the parts are available the computations need to be performed to get the dimensions in real world metrics.

This part of the project was developed upon by Sagar Abhyankar (Roll No 10703).

The main purpose of this model is to make sure the computation relating to the part of getting the users body size measurements from the image of that person is done here so it can return the accurate dimensions in real world metrics such as cm or inches.

This part of the project reduces the workload of the user as it automatically calculates their body measurements in real time based on a photo of themselves. This solves the problem of having the user do all the work of research to find out what size their shoulder width is, making it very easy for anyone to work with.

The main goal we had in our mind while designing this model was to ensure that anyone of any age should find it incredibly easy to use our website or app, which is why this model is designed such that it only takes an image as an input.

We thought that everyone would find it easy to just take an image of themselves and upload it to our website. That way the user does not have to do much work and can easily get their measurements just by supplying us with an image of themselves holding the checkerboard pattern on their phone from the app.

## Part 2: Website Frontend

The website frontend consists of 3 smaller sub-parts, mainly the User Interface, the Profile System and the Image Capture System.

The User Interface: The User interface was designed in such a way that the color scheme and modern design aspects would appeal to the younger generation, towards whom the app and website is marketed. However, it is also implemented in such a way that the website is incredibly user friendly and can be used by anyone.

The Profile System: The Profile System is designed to have a particular users data stored such that they only need to take measurements once. The size of the user's shoulder width is stored along with the persons profile and they can access their data by using the Login and Sign-Up page functions available under The Profile System. The profile system is also intuitively designed to be user friendly such that people with little to no internet exposure will be able to use it without any issues.

The Image Capture: The Image Capture System is designed to capture the image from the webcam of the user and send it to the Machine Learning Model for further processing using backend functionalities such as Django. This page is designed to be intuitively used by anyone to take a picture of themselves in the appropriate ways and all the instructions are available on the page for the user's reference prior to taking the image.

The User Interface and Profile System part of the project was mainly worked upon by Samir Hendre (Roll No 10708), Mrunal Golivadekar (10710) and Mugdha Malpe (Roll No 10713), with Sagar Abhyankar (Roll No 10703) working mainly on the Image Capture System.

## Part 3: Website Backend

The website backend part of the project deals with all of the user information being processed and stored on our Django server.

It consists of all the files which host both the Website Frontend and the Machine Learning Model and brings them together. Every file works together to produce the result you see on the website, and this is made possible by the integration done by the website backend.

This part of the project was mainly worked upon by Sagar Abhyankar (Roll No 10703) and Samir Hendre (Roll No 10708)

**Chapter 2**

# INDIVIDUAL ROLE

My Contribution (Work time ~60+ hours): -
In the want to complete the project in the stipulated time, I first started working on the part of project allotted to me. Later as we approached the date of project submission and my part of the project was completed, I helped to fill the gaps in other parts of the projects which were facing issues.

1)Made the Superfit App:
I made the super fit app from scratch. Added image taking and pattern displaying capabilities to the app. Without this app there is currently no way for the user to have the pattern required for scanning. This App uses faces calibration for measuring thr shoulder width (fig10)

2)Researched on non-client-side reference object:
In this part I did a thorough research on how to get the shoulder width size without using a pattern image(checkerboard) or any sort of client-side reference object for measuring the shoulder width. This part was extremely difficult since I needed to understand camera theory, trigonometry equations to get the result in the form of shoulder width. Although due to lack of time this research could not get completed, I have got interesting results to work with and this will be surely implemented in the upcoming version of Superfit app. For time being I shifted to client-side reference object-based measuring since it was feasible and easy to implement.

3)Pattern-Scanning Program:
The main function of this program is to return the shoulder width of the user in centimeters from an image. This pattern scanning program was split into two halves. One part of this function was done by me. In that I had to write a python program to find the pattern in the image and calculate the side length of the square shown in the pattern (fig 3). Once I have the length of the side of square in pixels, I do some basic mathematics to output the actual shoulder width in centimeters of the person. This part was successfully completed by me with about 98% + accuracy level.

4)Website-Backend, Backend and Frontend integration:

After I was done with my part of Picture scanning, I started helping with creating a Django backend system capable of delivering image taking and processing capabilities. For this, I spent significant time learning and experimenting various ways in which this could be achieved. I successfully fulfilled the following requirements of the project in the backend and a bit in the frontend for the webcam.

- Made the OpenCV code execute on the images uploaded to the server and return shoulder size values as output. Implemented a system to read, process and store images in the media folder of the backend.
- Designed the functional part of the frontend which captures the image on pressing “take picture” button.
- Solved the issue of getting this image from frontend to the backend in the form of DATA URI’s using ajax post.
- Designed an encoding system for images written in backend using sha256 Hashlib so that no image is left untracked in the backend and to make sure that the image processed is of the current user and not of some other user. (fig.9)
- Made the backend for the thankyou page designing a multi-input system so that its more efficient and allows the thankyou page to receive the final image output and size from either image upload option or through webcam without having to make two different pathways for serving the same purpose.
- Solved other minor issues in the backend.

## Chapter 3
# CONSTRUCTION

## Construction of the Superfit Website and App.

**Website Backend**: The SuperFit website backend is written using Django Framework which is a robust and modern web framework to design web applications. Django is a python-based Web-Framework. The backend is used to handle POST and GET requests demanded by the client/ user using the website and delivers the required data to and fro the webpage.

**Website Frontend**: The website frontend is written using Combination of JavaScript, HTML-5, CSS-3 and jQuery/Ajax, Bootstrap-4, FontAwesome and Cloudflare(loader). All of the frontend was hosted in the form of templates from the backend. The frontend of the website includes selecting styling of webpages to make them look better, responsive and alive. They display the data received from the backend(server) to the user and allow him to manage his information.

**Pattern-Scanning:** The brain of the project is the pattern scanning code which is written using python purely. This code uses the following main libraries which are available in the form of open-source modules: -

- OpenCV
- NumPy
- Matplotlib
- TensorFlow
- tf_bodypix

The code accepts an image of the user in the specified format (with -pattern) and returns the shoulder width of the user.

**App-:** The App is written in Java,Xml and provides functionalities such as taking an image and storing it for the user and providing the pattern required for taking the image. The app was constructed using Android Studios which is a friendly software for making android apps quickly.

## Chapter 4
# WORKING

## WEBSITE BACKEND (Django):

The purpose of backend is to handle user data, serve frontend templates as and when required. The backend also plays a crucial role in running python files on the user data such as images which cannot be achieved without a working backend.

In our project the backend works in the following way –

First the user sign's up to the website, during which he creates a unique username and password for himself. This data is then stored in the SQL database which is provided by Django. Now the user can login using his signup credentials and he will be allotted a profile by the backend. The user can now proceed to the scanning page (after going through the profile page) where he will be prompted to either upload his image with him holding the pattern scan image or take a picture using his webcam directly. Once the user takes the image it gets stored in the backend by deconstructing and reconstructing the DATA URI of the image taken. The image is stored in the images folder in the user app created in Django using hash function(sha256) so that the image cannot be hacked or accessed by anyone, and every image is observed by having a highly unique and secure image address. This storing of image is achieved by first reading the POST sent by the ajax which is in the form of a query dict. This query dictionary is then accessed for the DATA URI of the image sent by the frontend. Once the DATA URI is retrieved from the post dictionary it is ran through a function to convert the data uri into an actual image (achieved using OpenCV). This newly formed image is hashed, stored, and processed simultaneously to generate the output. The pattern scanning code is now run on this received image and the outputs are kept ready to be served as and when the user is redirected to the last page of the website (thankyou page). Here (on the thankyou page) the backend serves the image uploaded/taken by the user and the shows the shoulder width which was calculated by the pattern-scanning code. This is achieved by sending the image path of the image in the form of DATA URI to the frontend since the frontend JavaScript cannot directly serve files/images from the backend due to the updated security policy of modern-day web browsers.

If the user is happy with the size, he can select the proceed option and he gets redirected user by the backend using an up-to-date dictionary of links to amazon.in (e-commerce platform) with the size set to the size of the calculated size of the user for best results. The user can retake the image if he is unhappy with the results and the server accepts a new image for the user with which he can then proceed to shopping websites. The size for the user is stored in his profile so that he does not need to take a new image every time he wants to shop online. The backend also accepts page redirect requests within the server all the time allowing flexibility in accessing pages as wanted. All the POST/GET requests are secured by a CSRF (Cross Site Request Forgery protection) token which is a middleware which prevents any kind of session hacks and data breaches from happening which adds up to the security level of the website.

## WEBSITE FRONTEND:

The main purpose of the frontend is to touch up the appearance of the website and make it user friendly. It is also useful to make quick redirects by just requesting the new page for the user and not by making a post request first and then the backend redirecting the user to the designated web address which is inefficient. The website frontend comprises mainly 6 webpages(fig.11). All the webpages are responsive, meaning they can be accessed properly from phone as well as from a pc. Whether the user has a PC or a phone does not limit him from getting the maximum out of this website. Detailed information about them is as follows: -

- **Home Page:** It acts as the starting page/landing page of our website. The users will be directed to this page when they first visit the website. It has all the information about SuperFit and its components. The user will find the options to login/sign-up and information about creators of the website. The homepage also exhibits the achievements of SuperFit which includes customers served, carbon footprint reduction and the time saved in measuring processes globally (estimate).
- **Login Page:** On this page the user can login into the website if he has previously signed up and created an account. If the user does not have an existing account, he will be redirected to the Signup page.
- **Signup Page:** This page is used to provide user registration. The user can simply provide a username and password to create an account.

- **Profile Page:** Here the user will be able to see his profile including his profile picture, username, and email address. He can change his profile information any time he wants to with the text fields provided. On the lower half of this page the user can see his shoulder width size in his current profile (if any) from the previous scanning session. Below that are two options to
- **Instructions Page:** This page provides the instructions for following good practices while taking an image to maximize the accuracy of the calculated shoulder width. It also provides two pathways namely- Image Upload OR Webcam image using which the user can upload the image. Depending on the mode of image upload selected by the user he is redirected to the next page of the website.
- **Scanning Page:** The user accesses this page if he chooses not to upload the image from his device but rather wants to take an image with the webcam(live). This page asks for webcam access, and once granted the user will be able to see himself on the website. The user should be holding the pattern scan image in the SuperFit app in front of himself and take an image following the instructions given on the instructions page. When the user clicks the take picture button, the user will be able to see the image taken by him. Once the user is happy with the taken image, he can click on the "Done!" button and then he will be redirected to the Thankyou page. If he is not satisfied with the taken image, he can click on the take picture option again and a new image will be created.
- **Thankyou Page:** This page displays the image taken by the user while scanning and the size of the user's shoulder width. If the user feels that the calculated size is incorrect, he may use the "Retake" options and he will be sent back to the scanning page to take a new image. If the user is convinced with the displayed size, he may click the proceed button and continue to shop online with the size auto selected for him. The page also conveys a vote of thanks to the user for using the website and displays developer team information.

**SUPERFIT APP:**

This is the mobile supported version of the website and provides the same capabilities provided by the website. It plays a crucial role of providing the scanning pattern to the user. It can be also used to scan the user for retrieving the shoulder width. The app is in its early stages of development but provides an option for taking image of user and providing the scan pattern to the user. To use the website version of superfit, the user must still have this scanning pattern while taking the image in order to get valid results. (Fig 6,7)

## PATTERN SCANNING:

This is the program which processes the image given by the user. It is stored in utils.py in the backend of the website This program uses tf_bodypix library to isolate the torso of the user from the whole image. The amazing thing about this library is that it isolates the torso with 99% and above accuracy with any levels of noise in the image (fig 4) This property helps to boost the overall accuracy of the shoulder size. The "main" function returns the shoulder width of the person in pixels. It is called inside the "get_shoulder_width" function which uses the shoulder width in pixels and converts it into real world dimensions in centimeters. For this, first it detects the points on the pattern image which the user is holding and then detects and measures the sides of the squares in that pattern using OpenCV's good features to track function. The program uses 4 methods to evaluate the side length of the square in the pattern in the image. The results of all these 4 methods are then averaged out to obtain the final pixel length of the square to minimize error due to lack of accuracy in any one methodology. Now since the program knows the real-world dimensions of the pre-made pattern scan image it can then convert the shoulder width in pixels to shoulder width in centimeters. Once this is done the program then checks in which range does the measured shoulder width reading lie, for example – small, medium, large etc. Once the choice is made the program then returns the size.

## MEASUREMENT WITHOUT CLIENT-SIDE REFERENCE OBJECT: -

This part of the app is still under development and should be out soon. This feature of superfit can be only enjoyed by the superfit app users. They do not need to hold any kind of pattern while taking the image. Instead, they first need to calibrate their face by holding the camera approximately 30cms away from their face. Once this process is done, the scanning program basically uses the user's face for calibrating the pixels/centimeter ratio. This is possible when the camera lens information can be retrieved from the user's phone. The necessary information includes active sensor size (width, height) and camera lens information. This is internally retrieved, and the user does not need to enter them manually. The rest part of the code is similar to the website version of the code. (fig10)

## Chapter 5
# CALCULATIONS

**The pattern scanning programs use a few basic calculations and rules to evaluate the shoulder width**

1)Side length of the square used in the pattern = 1.98cm

2)Pixel width for shoulder (for example) = 287pixels

3) Pixel width of square (for example) = 12 pixels.

4)Shoulder Width = Pixel width (shoulder)/pixel width(square)*side length

Shoulder Width = (287/12) *1.98cm

Shoulder Width=47.83cm

5)Use of line slope to find out side of the square in the pattern: - Since the "good_features_to_track" function just tracks the points on the pattern, we need a system which connects the points which represent the sides of a square. This is achieved by taking into consideration co-ordinates of individual tracked points in pairs. If two points form a line whose slope is in the range of 87 to 90 degrees or -2 to 1 degrees, the two points basically represent the horizontal or vertical lines which are nothing but the sides of the square. The distance between these two points can then be easily evaluated.

Formula:

Slope (in degrees) = arctan(Y2-Y1/X2-X1)

Where Y1, Y2, X1, X2 represent the co-ordinates of pair of points in consideration.

## Chapter 6

# SPECIAL PROGRAMS AND CODE SNIPPETS

Throughout the project some of the codes which were developed were original and served the purpose of writing them very accurately and efficiently, as a result they stood out from the other code.

OpenCV special functions:

1)The torso detection code which was designed by Om Gund (10705) is almost 100% accurate and served the requirements of it being in purely based in python language and runs on single threading which reduces the load when it is run in the server.

```
def main(path):
    bodypix_model = load_model(download_model(BodyPixModelPaths.MOBILENET_FLOAT_50_S
TRIDE_16))
    print(path)
    f = cv2.imread(path)
    result = bodypix_model.predict_single(f)
    mask = result.get_mask(threshold=0.75).numpy().astype(np.uint8)
    part_mask = result.get_part_mask(mask,part_names=['torso_front','torso_back'])
    #part_mask_not = cv2.bitwise_not(part_mask)
    dimensions = f.shape
    image = np.zeros(dimensions, np.uint8)
    image[:] = (255, 255, 255)
    masked_image = cv2.bitwise_and(image, image, mask=part_mask)
    fr = masked_image


    dimensions = fr.shape
    height = dimensions[0]
    width = dimensions[1]
    frame = cv2.cvtColor(fr, cv2.COLOR_BGR2GRAY)
    font = cv2.FONT_HERSHEY_COMPLEX
    _, threshold = cv2.threshold(frame, 110, 255, cv2.THRESH_BINARY)

    # Detecting contours in image.
    contours, _= cv2.findContours(threshold, cv2.RETR_TREE,
                                  cv2.CHAIN_APPROX_SIMPLE)
    cord=[[],[]]
```

```
# Going through every contours found in the image.
   for cnt in contours :

        approx = cv2.approxPolyDP(cnt, 0.009 * cv2.arcLength(cnt, True), True)

        # draws boundary of contours.
        cv2.drawContours(fr, [approx], 0, (0, 0, 255), 5)

        # Used to flatted the array containing
        # the co-ordinates of the vertices.
        n = approx.ravel()
        i = 0

        for j in n :
            if(i % 2 == 0):
                x = n[i]
                y = n[i + 1]

                if(y<(height/2)):
                    cord[0].append(x)
                    cord[1].append(y)

                # String containing the co-ordinates.
                string = str(x) + " " + str(y)

                if(i == 0):
                    # text on topmost co-ordinate.
                    cv2.putText(fr, "Arrow tip", (x, y),
                                font, 0.5, (255, 0, 0))
                else:
                    # text on remaining co-ordinates.
                    cv2.putText(fr, string, (x, y),
                            font, 0.5, (0, 255, 0))
            i = i + 1

   xmin = min(cord[0])
   xmax = max(cord[0])

   x1,y1 = xmin,cord[1][cord[0].index(xmin)]
   x2,y2 = xmax,cord[1][cord[0].index(xmax)]
   sw = ((x2-x1)**2 + abs(y2-y1)**2)**0.5


   return sw
```

2)The code for pattern detection: The specialty about this code is that its very light weight and integrates very well with the torso width mapping code. It also works on the math model developed us rather than using a pre-built library. It evaluates 4 methodologies together and returns output as the average of them.

```
def get_shoulder_width(image, shoulder_pixels:int, actual_dimensions=1.98, points=8) ->
float:
    """Returns the shoulder width in centimeters in real life"""
    def most_common(lst) -> int:
        """Returns most common occurrence"""
        return max(set(lst), key=lst.count)
    img = cv2.imread(image)
    gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
    corners = cv.goodFeaturesToTrack(gray, points, 0.01, 8)
    corners = np.int0(corners)
    list_x = []
    list_y = []
    for i in corners:
        x, y = i.ravel()
        cv.circle(image, (x, y), 2, 255, -1)

        list_x.append(x)
        list_y.append(y)

        list_x.sort()
        list_y.sort()
    diff_x = [t - s for s, t in zip(list_x, list_x[1:])]
    diff_y = [t - s for s, t in zip(list_y, list_y[1:])]
    diff_x_limited = [i for i in diff_x if 8 < i < 25]
    diff_y_limited = [i for i in diff_y if 8 < i < 25]
    final_set = set(diff_x_limited + diff_y_limited)
    pixel_candidates_2 = diff_x_limited + diff_y_limited
    pixel_candidates = list(final_set)
    pixel_candidates.sort()
    pixels = pixel_candidates[-1]
    pixels_2 = sum(pixel_candidates_2) / len(pixel_candidates_2)
    pixel_avg = sum(pixel_candidates) / len(pixel_candidates)

    pixels_new = most_common(pixel_candidates_2)

    shoulder_width = (shoulder_pixels * actual_dimensions) / pixels

    shoulder_width_avg = (shoulder_pixels * actual_dimensions) / pixel_avg

    shoulder_width_2 = (shoulder_pixels * actual_dimensions) / pixels_2
    diff_x_indices = [diff_x.index(i) for i in diff_x if 0 < i < 4]
    diff_y_indices = [diff_y.index(i) for i in diff_y if 0 < i < 4]
    final_y_candidates = [diff_y[i] for i in diff_x_indices]
    final_x_candidates = [diff_x[i] for i in diff_y_indices]

    final_final = [i for i in final_y_candidates + final_x_candidates if 8 < i < 25]
    final_final.sort()
    try:
        omega_pixels = most_common(final_final)
    except:
        omega_pixels = pixel_avg
    omega_shoulder_width = (shoulder_pixels * actual_dimensions) / omega_pixels
    shoulder_width_new = (shoulder_pixels * actual_dimensions) / pixels_new
    plt.imshow(image), plt.show()
    return (omega_shoulder_width + (2 * shoulder_width_new) + shoulder_width +
shoulder_width_2 + shoulder_width_avg) / 6
```

Website Backend Special Functions (Django):

1)The following codes are special because they are completely original and highly logical for implementing routes in Django

```
def url_to_image(url):
    """Function to convert the recieved url string from frontend into an actual imag
e and store it in the media files"""
    x = hashlib.sha256(("".join(map(str,[random.randint(0,9) for i in range(10)])))).
encode('utf-8')).hexdigest()[0:52]

    x_string = f'{x}.jpeg'
    resp = urllib.request.urlopen(url)
    image = np.asarray(bytearray(resp.read()), dtype="uint8")

    image = cv2.imdecode(image, cv2.IMREAD_COLOR)
    global store
    store = image
    global img_path
    img_path = os.path.join(settings.MEDIA_ROOT,x_string)

    cv2.imwrite(img_path,image)
    global size

    size = get_shoulder_width(img_path)
```

2)The above shown function executes and satisfies 3 requirements at one go which are decoding image data uri, storing the image formed in encoded format and processing it to generate the output.

```
def file_path_encoder(instance,filename):
        """This function is used to provide a hashed,unique and secure imagepath
        to the media folder.It also tunnels intermediate values for exporting using
tunnel function.The output format is {unique-52-char-code}.{basefile-extension} """
        y= hashlib.sha256(("".join(map(str,[random.randint(0,9) for i in range(10)])
)).encode('utf-8')).hexdigest()[0:52]
        instance.original_file_name = filename
        base,ext = filename.split(".")
        new_name = f"{str(y)}.{ext}"
        tunnel(new_name)
        Timer(0.1, show_shoulder_width).start()
        return os.path.join(settings.MEDIA_ROOT,new_name)
def tunnel(path):
        """Function to export values which cannot be exported otherwise from inside
another function """
        global x
        x = path
```

3)The above given function implements threading in-order to run two simultaneous processes in two different files at once. Implementing this was tough to think of in the first go, but discovering this method was beneficial for the project in terms of handling image storing and processing queues.

Website Frontend Special Functions:

1)Following is the code for taking the image in the webcam and sending it to the backend in the form of data url . The specialty about this code is that it's the best implementation for this scenario and sends data urls instead of images which is very light for the server to digest. It also works on the principle of keeping minimum contact between the server and the frontend and generates just one POST request per image.

```
function takepicture() {
                var context = canvas.getContext('2d');
                if (width && height) {
                  canvas.width = width;
                  canvas.height = height;
                  context.drawImage(video, 0, 0, width, height);

                  var data = canvas.toDataURL('image/jpeg');
                  console.log(data)
                  var URL="{% url 'images_test:save_image' %}"
                  $.ajax({
                type: "POST",
                url: URL,
                data: data,

                      });
              }else {
                  clearphoto();
                }
              }
```

2)The following code for the inspirational quotes section was designed by Mugdha Malpe (10703). I have used a jumbotron section to display content like quotes and images of renowned personalities into an aesthetic design.

These code snippets are responsive for all devices like mobile phones, tablets, laptops, etc. which is achieved with the help of media queries.

The media queries are a special syntax for CSS that allows us to define some styles that will only be applied only when the specified conditions are satisfied.

```
<!--Start of Inspiration Section-->
<div id="inspiration" class="offset">

    <!--Start of Jumbotron -->
    <div class="jumbotron">
        <div class="col-12 text-center">
            <h1 clas="heading">The real inspiration that keeps us going!</h1>
            <div class="heading-underline"></div>
        </div>

        <div class="row">
            <div class="col-md-4 feedback">
                <div class="row">
                    <div class="col-md-4">
                        <img src="img/gucci.jpg">
                    </div>
                    <div class="col-md-8">
                        <blockquote>
                            <div class="text">
                                <i class="fas fa-quote-right" style="font-
size: 30px;"></i>
                                <h4><strong>Quality</strong> is remembered long afte
r price is forgotten.</h4>
                                <hr class="feedback-hr">
                                <h5>– Guccio Gucci <br>(Founder of Gucci)</h5>
                            </div>

                        </blockquote>
                    </div>
                </div>
            </div>
```

```
<div class="col-md-4 feedback">
    <div class="row">
        <div class="col-md-4">
            <img src="img/armani.jpeg">
        </div>
        <div class="col-md-8">
            <blockquote>
                <i class="fas fa-quote-right" style="font-size: 30px;"></i>
                <div class="text">
                    <h4><strong>Elegance</strong> is not catching somebody's eyes,
                        <br>
                        <strong>it's staying in somebody's memory.</strong></h4>
                    <hr class="feedback-hr">
                    <h5>– Giorgio Armani <br>(Founder of Armani)</h5>
                </div>

            </blockquote>
        </div>
    </div>
</div>
<div class="col-md-4 feedback">
    <div class="row">
        <div class="col-md-4">
            <img src="img/chanel.jpg">
        </div>
        <div class="col-md-8">
            <blockquote>
                <div class="text">
                    <i class="fas fa-quote-right" style="font-size: 30px;"></i>
                    <h4><strong>Beauty</strong> begins the moment you decide to <str
ong>be yourself</strong>.</h4>
                    <hr class="feedback-hr">
                    <h5>– Coco Chanel <br>(Founder of Chanel)</h5>
                </div>
            </blockquote>
        </div>
    </div>
</div>
</div><!--End of row-->
</div><!--End of jumbotron-->
</div>
<!--End of Inspiration Section-->
```

Following is the CSS supplementary code of the above code.

```
/* inspiration --*/
.feedback img {
  width: 100%;
  border-radius: 50%;
  size: 200px;
}
blockquote .svg-inline--fa {
  color: #9905fc;
  margin: 1 rem;
}
.feedback-hr{
  border-top: 0.05rem solid #9905fc;
}
.jumbotron{
  background-image: linear-gradient( #fc68f0, #b500fc)!important;
  font-family: 'Lato', sans-serif;
}
@media only screen and (max-width: 480px) {
  .feedback img {
    width: 100px!important;
    text-align: center;
    display:block;margin:0 auto;
  }
  .text h4{
    font-size: 26px!important;
  }
  .text h5{
    font-size: 16px!important;
  }
  }

@media only screen and (max-width: 780px) {
  .feedback img {
    width: 70px;
  text-align: center;
  }
  .text h4{
    font-size: 16px!important;
  }
  .text h5{
    font-size: 12px!important;
  }
  }
```

**Chapter 7**

# Dependencies

The project utilizes some preexisting libraries and modules which helped us to write and implement the logic of our project at a decent pace. The dependencies used in the complete project are as follows: -

- Python 3.9.1
- CSS 3
- Html 5
- JavaScript
- Pillow
- Django
- Django Crispy Forms
- Bootstrap
- OpenCV
- NumPy, Matplotlib
- TensorFlow
- TensorFlow-gpu
- tfjs_bodypix, tfjs_graph_convertor
- Hashlib,random,urllib,csrf,threading.

## Chapter 8
# FIGURES AND GRAPHS

**Fig 1: Sample Image taken by following the Instructions.**

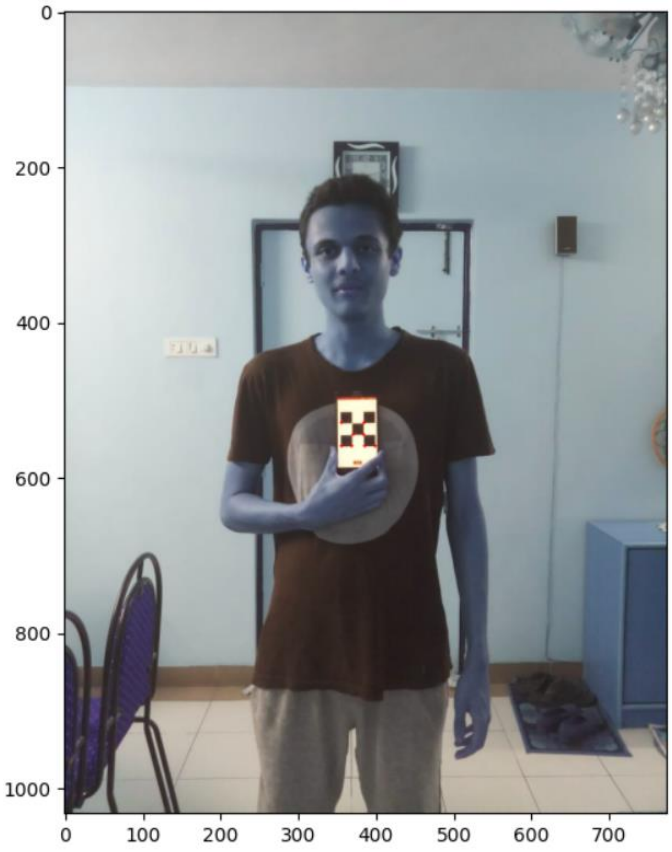



**Fig 2: Scanning Pattern**

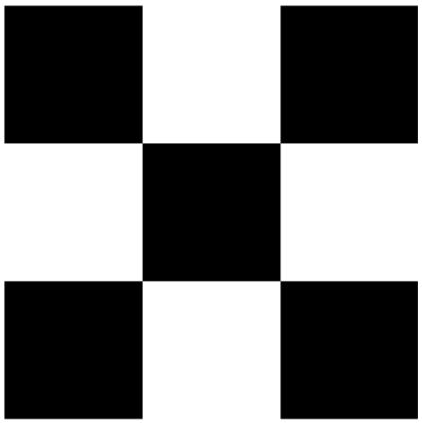


**Fig 3: Tracking points on the Pattern (red dots)**

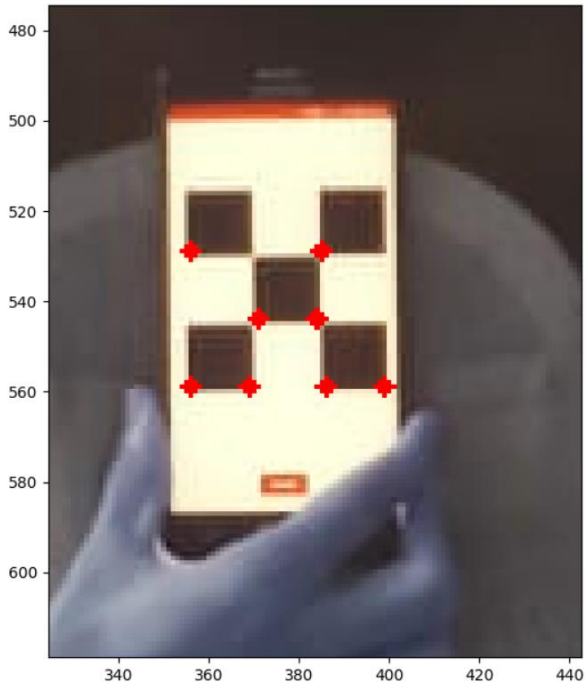



**Fig 4: Program tracking the torso perfectly to get shoulder width**

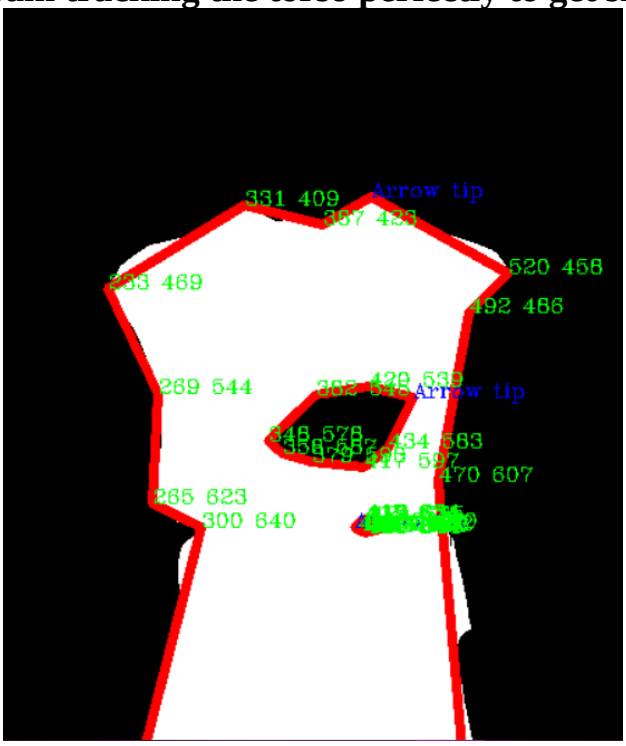


**Figure 5: The website flowchart**

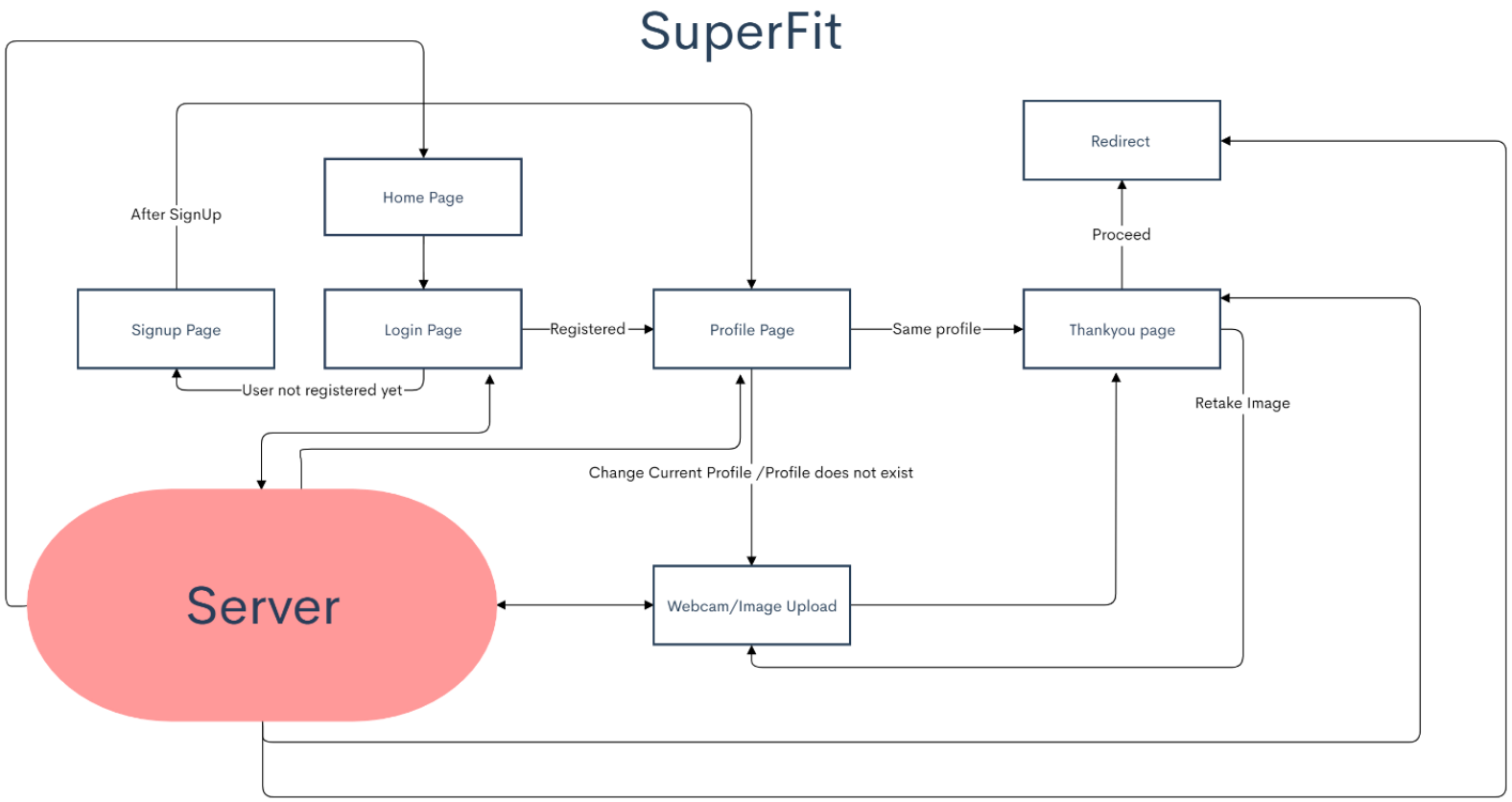


**Fig 6: The Superfit App**

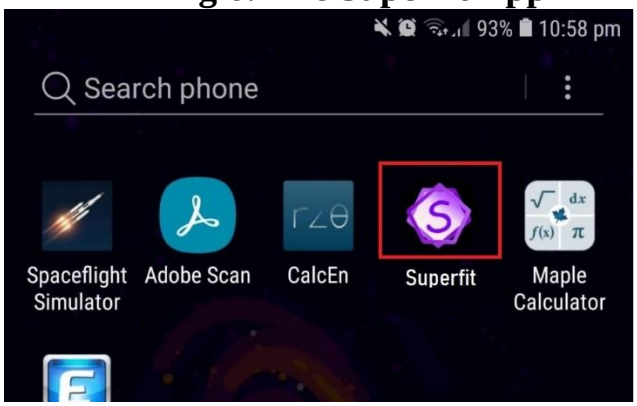



**Fig 7: The pattern displayed inside the app**

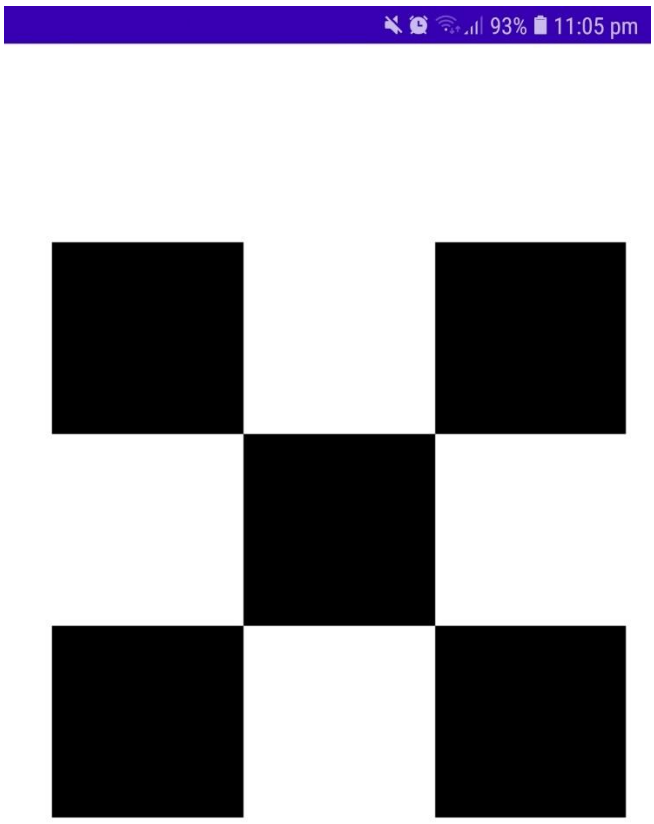


**Fig8: The Sensor Info Retrieving function and Retrieval in the App.**

```java
@RequiresApi(api = Build.VERSION_CODES.M)
private void calculateVFOV(CameraManager cManager) {
    try {
        for (final String cameraId : cManager.getCameraIdList()) {
            CameraCharacteristics characteristics = cManager.getCameraCharacteristics(cameraId);
            int cOrientation = characteristics.get(CameraCharacteristics.LENS_FACING);
            if (cOrientation == CameraCharacteristics.LENS_FACING_BACK) {
                float[] maxFocus = characteristics.get(CameraCharacteristics.LENS_INFO_AVAILABLE_FOCAL_LENGTHS);
                SizeF size = characteristics.get(CameraCharacteristics.SENSOR_INFO_PHYSICAL_SIZE);
                Rect activesize = characteristics.get(CameraCharacteristics.SENSOR_INFO_PRE_CORRECTION_ACTIVE_ARRAY_SIZE);
                float h = activesize.height();
                float w = activesize.width();

                float horizontalAngle = (float) (2 * atan(w / (maxFocus[0] * 2)) * 57.29);
                float verticalAngle = (float) (2 * atan(h / (maxFocus[0] * 2)) * 57.29);
                Log.v(TAG, String.valueOf(activesize.height()));
                Log.v(TAG, String.valueOf(h));
                Log.v(TAG, String.valueOf(w));
                Log.v(TAG, String.valueOf(verticalAngle));
                Log.v(TAG, String.valueOf(horizontalAngle));
                Log.v(TAG, String.valueOf(h / w));
            }
```

```
V/SENSOR INFO: 3106
V/SENSOR INFO: 3.468
V/SENSOR INFO: 4.624
V/SENSOR INFO: 51.431946
V/SENSOR INFO: 65.41236
V/SENSOR INFO: 0.75
```

**Figure 9: Hashed images stored in the backend media /images folder**

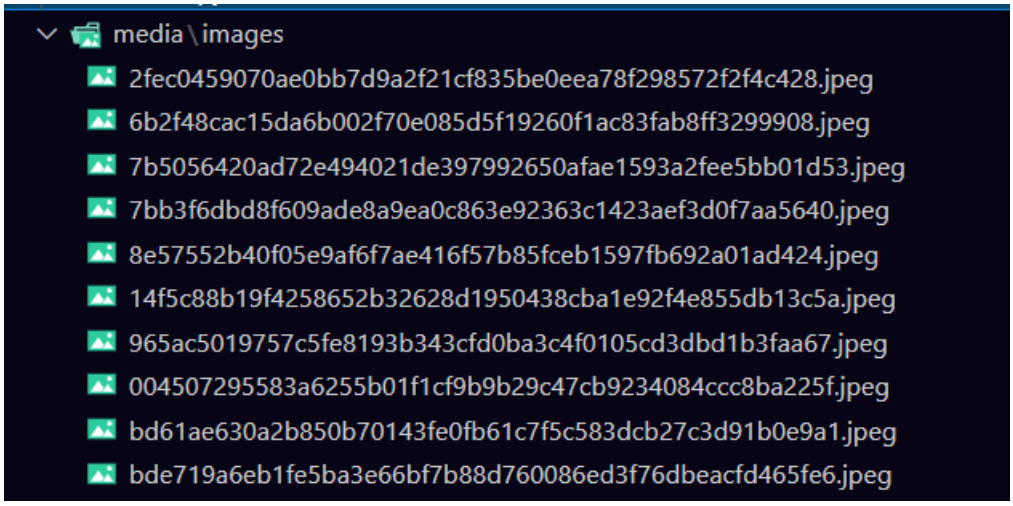
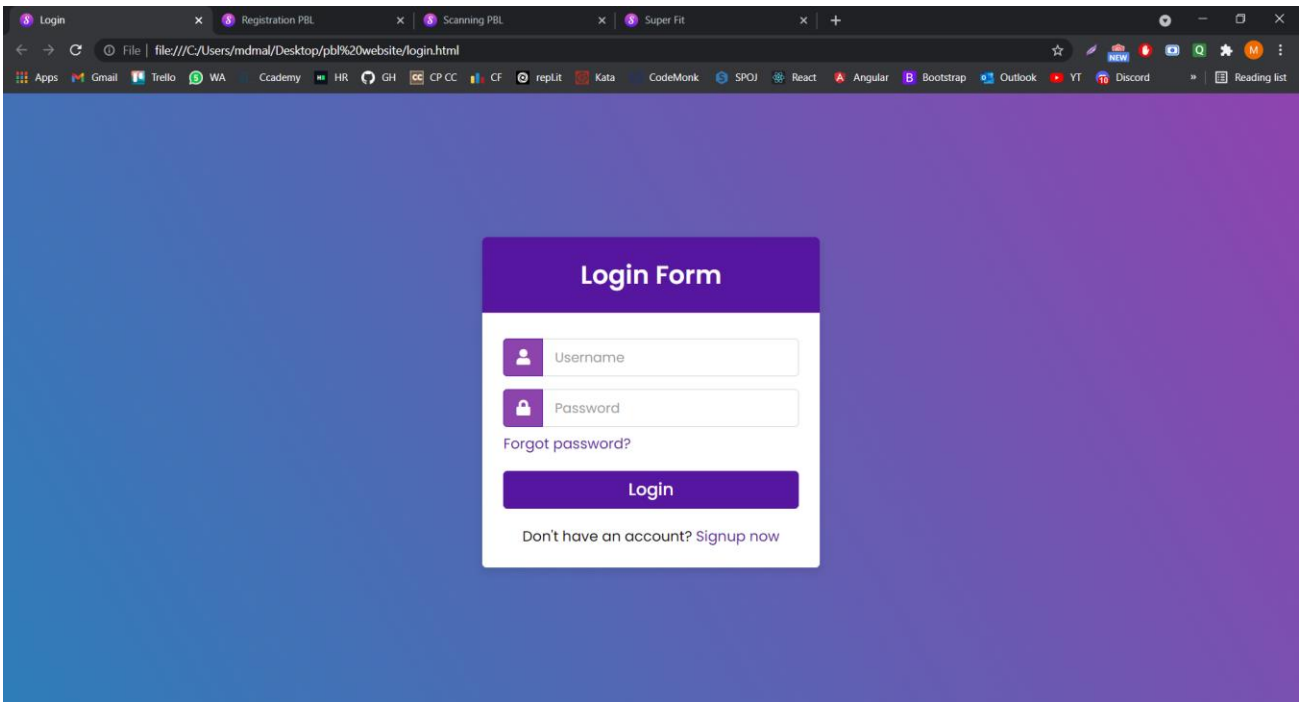


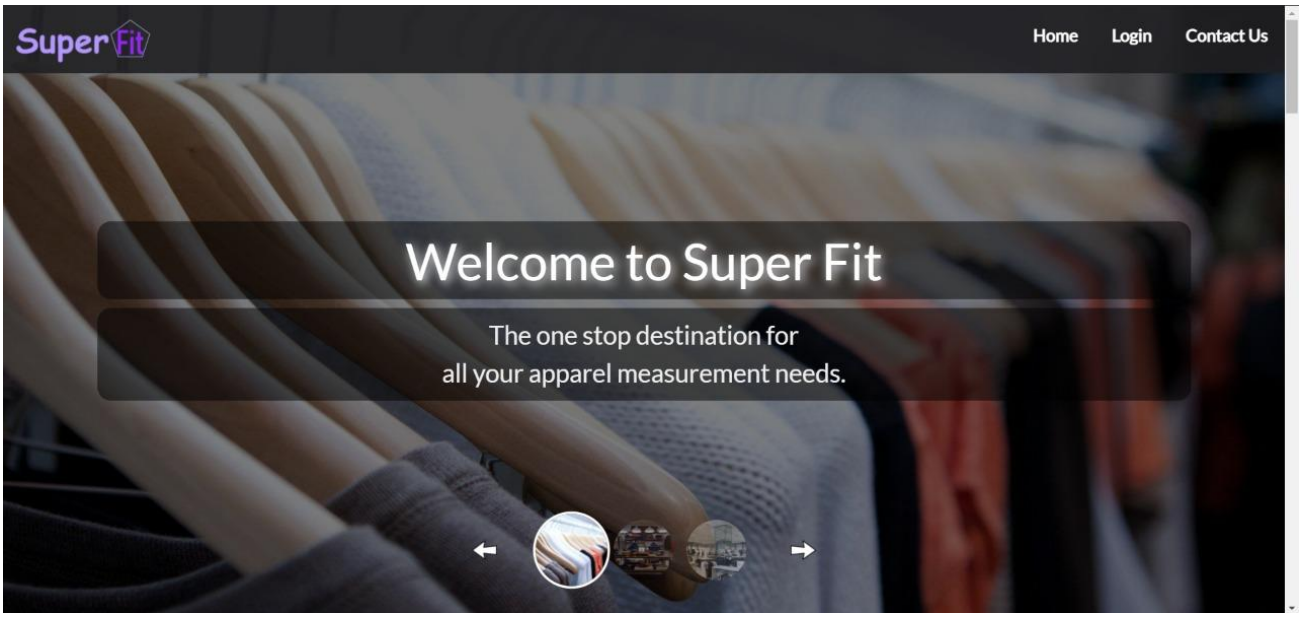

**Fig10: App version of Superfit using Face Calibration**

## Fig11: Screenshots of Frontend Website

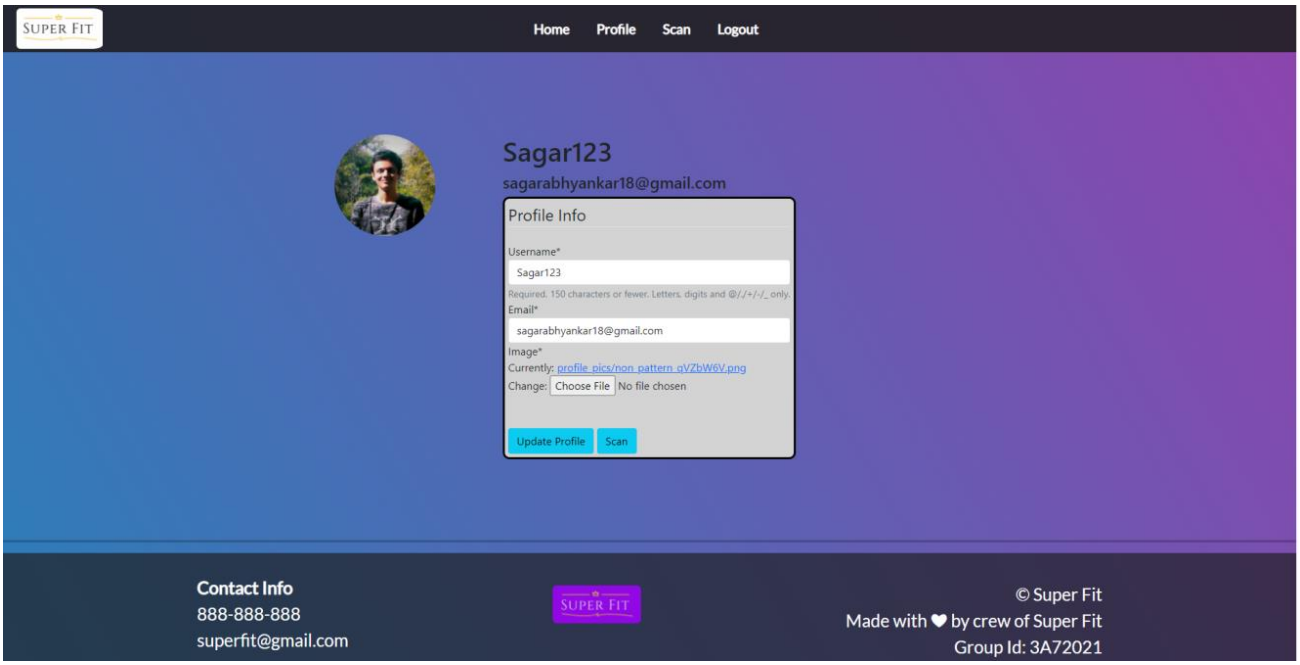
Login Page



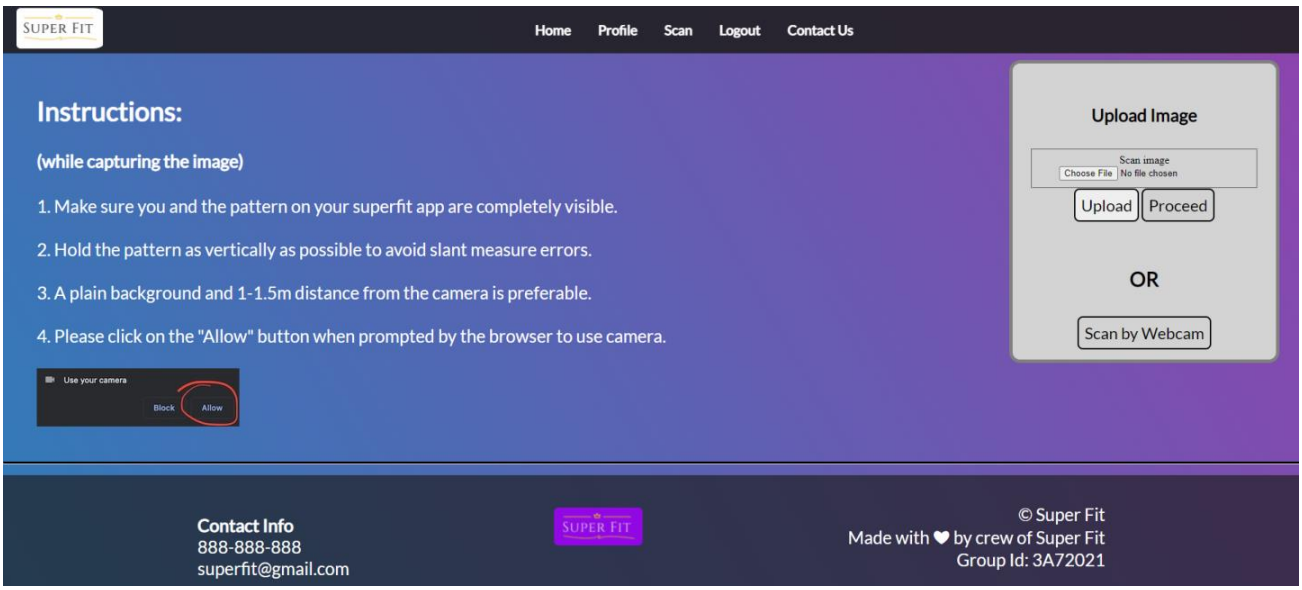

Homepage



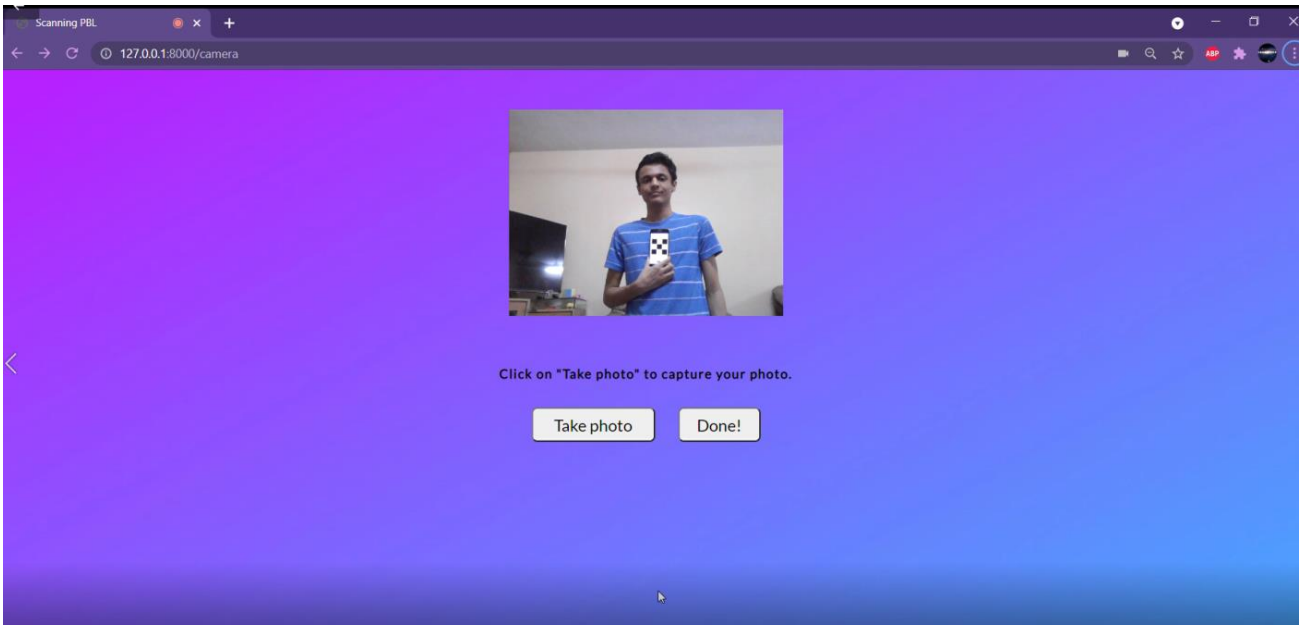
Profile Page



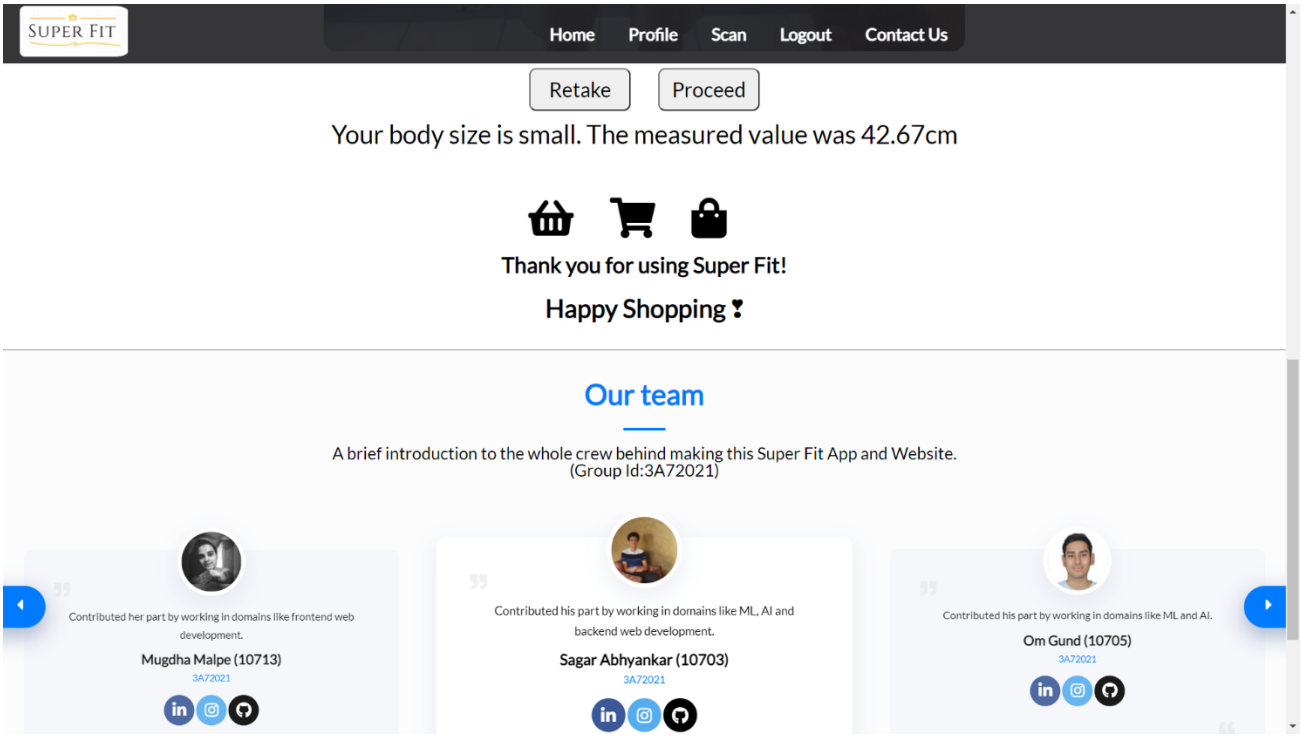

Instructions Page



Webcam Page




End Page/Thankyou Page



## LIST OF TABLES

**Size Chart**

| Shoulder Width in Centimeters | Allotted Size |
|---|---|
| 40-44 | small |
| 44-48 | medium |
| 44-52 | Large |
| 52-56 | XL |
| 56+ | 2XL |

Above shown is the table which the app and website use for categorizing the received shoulder width into small, medium, large etc.

## Chapter 9
# APPLICATIONS

Applications of this project include:

1) **E-Shopping for clothes**: This technology minimizes the confusion due to non-uniform size charts and enables the user to get accurate measurements for clothes.
2) **Shoe sizing**: Foot sizes can be obtained for convenient online shopping.
3) **Health apps**: The features of this app can be used to monitor inch loss.
4) **Contact less measuring:** In the future, rather than measuring yourself with standard measuring tape to measure sizes, users will be scanned to get body dimensions which is faster and provides decent accuracy.

The main aim of this website is to make online shopping easy so that people can shop from the comfort of their homes.

## Chapter 10
# ADVANTAGES AND DISADVANTAGES

## Advantages of Super fit
## Why shop Online?

- 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.

### 1) Saves time -

In contrast to physical stores where we have to spend time finding the perfect size, super fit measures the size and redirects you to online shopping platforms with clothes of the calculated measurements, hence saving time.

### 2) Lower possibilities of discrepancies-

Due to different size charts all over the world, it gets very confusing for consumers to choose the right size. Super fit offers the advantage of pre calculating the size and therefore saving us the effort of making the required conversions.

### 3) Convenience-

The main concern in online shopping is the fact that it is a big hassle returning clothes if they turn out to be the wrong size. This website enables size calculation which makes sure that the right size is ordered.

### 4) Saves money-

While returning apparels bought online, not all online platforms offer a refund due to which a lot of money is wasted. Ordering the right size reduces the chances of returning and money is saved.

## Disadvantages of Super fit

### 1) May not be accurate-

Since this method of size measurement is online, the measurements may not be as accurate as physical fittings.

### 2) Lesser options-

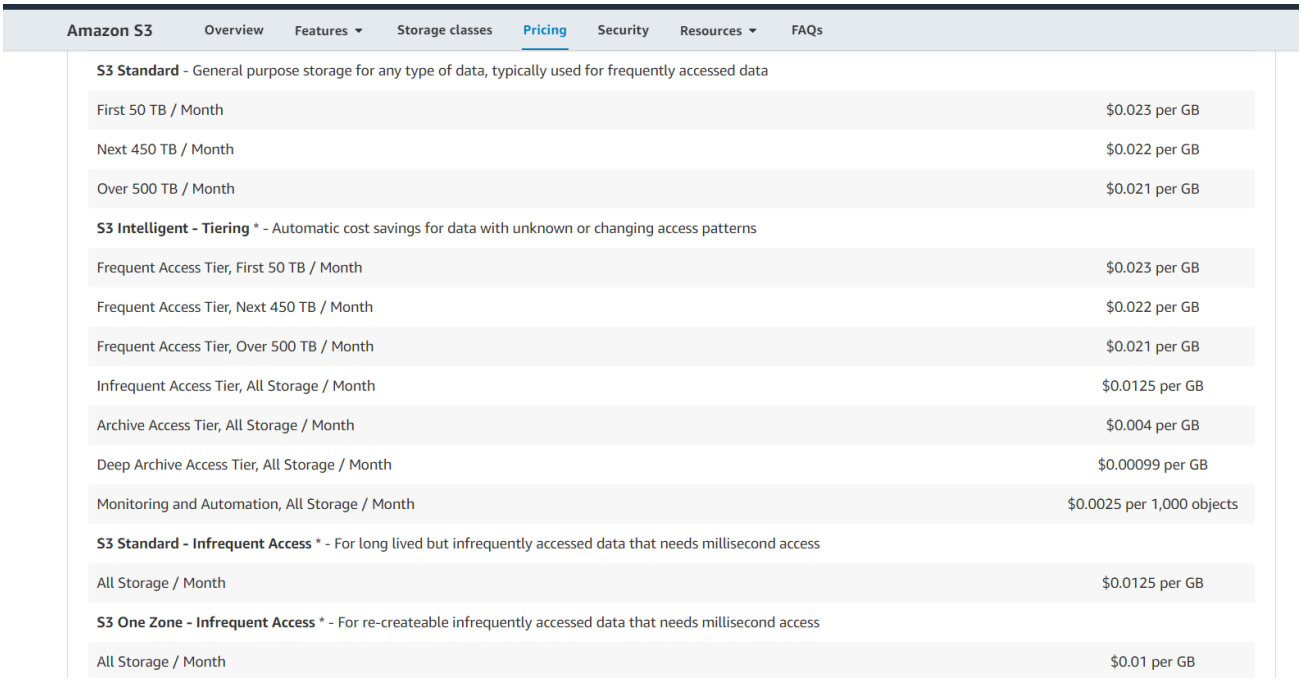
Since only one size is selected after measurement, lesser variety of clothes is available.

### 3) Necessity of a good webcam-

Though it is very slight, there is a possibility that measurements may not be accurate due to bad camera quality. Therefore, there is a need of a good webcam for precise results.

# Chapter 11
# COST ANALYSIS

The main cost of the running the website, originates from the cost to have a external database for storing images, since storing images in static files of the server or the database of the server is a bad practice. The cloud database used for this will be amazon web services (AWS). The pricing for the server is given in the following image



Amazon S3 | Overview | Features | Storage classes | Pricing | Security | Resources | FAQs

| | |
|---|---|
| **S3 Standard** - General purpose storage for any type of data, typically used for frequently accessed data | |
| First 50 TB / Month | $0.023 per GB |
| Next 450 TB / Month | $0.022 per GB |
| Over 500 TB / Month | $0.021 per GB |
| **S3 Intelligent - Tiering** * - Automatic cost savings for data with unknown or changing access patterns | |
| Frequent Access Tier, First 50 TB / Month | $0.023 per GB |
| Frequent Access Tier, Next 450 TB / Month | $0.022 per GB |
| Frequent Access Tier, Over 500 TB / Month | $0.021 per GB |
| Infrequent Access Tier, All Storage / Month | $0.0125 per GB |
| Archive Access Tier, All Storage / Month | $0.004 per GB |
| Deep Archive Access Tier, All Storage / Month | $0.00099 per GB |
| Monitoring and Automation, All Storage / Month | $0.0025 per 1,000 objects |
| **S3 Standard - Infrequent Access** * - For long lived but infrequently accessed data that needs millisecond access | |
| All Storage / Month | $0.0125 per GB |
| **S3 One Zone - Infrequent Access** * - For re-createable infrequently accessed data that needs millisecond access | |
| All Storage / Month | $0.01 per GB |

Depending upon the popularity of the website, we can select the right deal. Assuming the server to have 50,000 sessions per month the cost for image storage and handling is roughly estimated to be around 400 rupees, which is minimal. The maintenance cost for such websites is around 10000-15000 per 3 months. So, this sums up the cost analysis for the functioning of the website.

Which is roughly evaluated to be 3400 rupees per month at the maximum.

## Chapter 12
# CONCLUSION

The ever-increasing demand for shopping of clothes on online websites such as Amazon, Myntra, AJIO, Flipkart, etc. coupled with the huge variety of sizes and the ambiguity of the conventional sizing such as S, M, L, etc. has led to a huge problem in the industry which has not only caused multiple people to lose trust in online shopping of clothes but also to have a huge negative impact on the environment. Statistical surveys prove that people have a lot of difficulty finding the right size for themselves while shopping for clothes online. This is partly from the fact that some people are unaware of their sizes or just don't know how to navigate through the endless list of sizes offered by big brand platforms such as Amazon.

Our app and website are a step towards solving this problem and mitigating the effects caused by them. Our Machine Learning models are highly accurate and equipped with the tools to help everyone find their SuperFit. Our super user-friendly implementation of the websites interface allows anyone to find out what their measurements are, and our search finder helps them find shirts that they'll love which will be tailored towards their exact body features.

This project aims not only to make online shopping of clothes easy for everyone but also to make it an enjoyable experience where people do not have to worry about whether what they order will be a right fit for them or not. We aim to help the environment by reducing pollution caused by unnecessary returns by making sure everyone has the ability to find the right fit for their friends, their family or themselves!

Many more improvements are possible and our work to make SuperFit even more convenient for everyone to use will never cease.

## REFERENCES

The following websites were very useful for the reference work:


[1] https://stackoverflow.com/

[2] https://docs.djangoproject.com/en/3.2/

[3] https://www.geeksforgeeks.org/

[4] https://getbootstrap.com/docs/4.1/getting-started/introduction/

[5] https://www.youtube.com/

[6] https://developer.android.com/reference/android/hardware/camera2/package-summary

[7] https://github.com/

[8] https://docs.opencv.org/4.5.2/

[9] https://www.w3schools.com/

[10] https://docs.python.org/3/

[11] https://api.jquery.com/category/ajax/

[12] https://getbootstrap.com/docs/5.0/getting-started/introduction/

[13] https://fontawesome.com/

[14] https://www.cloudflare.com/en-in/cdn/

[15] https://fonts.google.com/