

المدرسة العليا للإعلام الآلي - 08 ماي 1945 - بسيدي بلعباس
Ecole Supérieure en Informatique
- 08 Mai 1945- Sidi Bel Abbes



MEMOIRE

En Vue de l'obtention du diplôme d'**ingénieur d'état**

Filière : **Informatique**

Spécialité : **Ingénierie des Systèmes Informatiques (ISI)**

Thème

An Artificial Intelligence based Personalized Style Recommendation System For Fashion Design

Présenté par :

- Mr Benzeghli Nor El Islam
- Mr Bourahla Karim

Soutenu le : **00/00/2021**

Devant le jury composé de :

- | | |
|----------------------------|-------------|
| - M/Mme/Mlle XXX | Président |
| - M Khaldi Belkacem | Encadreur |
| - M/Mme/Mlle XXX | Examinateur |
| - M/Mme/Mlle XXX | Examinateur |

Acknowledgement:

We would like to express our special thanks and gratitude to the following people for helping with this master thesis:

First of all to **our families** who supported us through the bad and through the good on this noble path and in life overall we cannot be grateful enough for them and words cannot describe the amount of support we had and still having.

To all the staff of our great school ESI SBA which was like family to us we cannot ask for better ones.

To our thesis framer **Dr. Khaldi Belkacem** who is one of the best professors in our university, who guided us to the correct path of realizing this thesis and gave us valuable advices and directions to perfect this domain.

To our Director of Studies in our school **Dr. Amar Bensaber Djamel** and **Dr. Benslimane Sidi Mohammed** for their great effort they've put in this school along our 5 years of rich study. Finally, we would like to give our thanks to our friends and colleagues for their support through this journey.

Bourahla Karim and Benzeghli Nor El Islam.

Contents

Introduction

1. Classbox	1
2. Problems and solutions	1
3. Overview of our solutions.....	2
4. Objective.....	3
5. Limitations.....	4

Chapter I Background

I.1. Artificial Intelligence.....	5
1.1.1 Artificial Intelligence Applications	6
1.1.2 Machine learning	7
1.1.3 Deep learning	9
1.1.3.1 Convolutional Neural Networks	10
I.2 Technical notions	11
I.2.1 Website	11
I.2.2 Responsivity.....	11
I.2.3 API (Application Programming Interface)	11
I.2.4 MVC & MVT	12
I.2.5 ORM	14
I.2.6 Front End.....	15
I.2.7 Back End	16

Chapter II Solution Design & Technologies

II.1. Solution Design	17
II.1.1 Use case diagrams	17
II.1.2 Component diagram	19
II.1.3 Deployment diagram	20
II.1.4 Class diagram.....	21

II.2. Technologies	22
III.2.1. Artificial Intelligence technologies	22
III.2.2. Web platform technologies.....	25

Chapter III Artifical Intelligence Algorithms

III. Artificial Intelligence algorithms	31
III.1 Virtual Try On	31
III.1.1 Data.....	32
III.1.2 Training	32
III.1.3 Testing	33
III.1.3 Optimizing the inference	35
III.2 Color palettes generator.....	39
III.2.1 Data.....	39
III.2.2 Training and Testing	41
III.2.3 AI Palettes	44
III.3 T-shirt Size Recommender.....	45
III.3.1 Data.....	45
III.3.2 Training and Testing	46

Chapter IV Web Platform

IV. Web platform	48
IV.1 Project creation	48
IV.2 Project structure	48
IV.3 Project files.....	48
IV.4 APIs.....	49
IV.5 Web Design.....	50
IV.5.1 User Front	50
IV.5.2 Admin Front	54
IV.6 Artificial Intelligence Linking	56
IV.6.1 T-shirt size recommendation	56
IV.6.2 Color palettes generator.....	57
IV.6.2 Virtual Try On	58

Chapter V Project management

V.1. Collaborative work	61
V.1.1 Introduction	61
V.2.2 Tools.....	61
V.3.3 Project monitoring	62

V.2. Project management methodology.....	63
V.2.1 Introduction	63
V.2.2 Method used	64

Conclusion

1. Conclusion.....	65
2. Future work.....	65
2.1 Virtual try on via Augmented Reality.....	65
2.2 Size estimation via Image.....	65
2.3 Outfit Matching	66
2.4 Popularity Prediction	66

References.....	67
------------------------	----

Table of Figures

Fig.1 Artificial Intelligence and its main categories	5
Fig.2 Supervised Learning vs Unsupervised Learning	7
Fig.3 Reinforcement Learning	8
Fig.4 ANN (Artificial Intelligence Neural Network)	9
Fig.5 CNN (Convolutional Neural Network)	10
Fig.6 Control Flow of MVT in django framework	14
Fig.7 Django Model based database using ORM	15
Fig.8 Client Use Case Diagram	17
Fig.9 Administrator Use Case Diagram	18
Fig.10 Component Diagram	19
Fig.11 Deployment Diagram	20
Fig.12 Class Diagram	21
Fig.13 Django modelisation	25
Fig.14 Django Development Methodology	26
Fig.15 Comparison of different virtual try-on models	31
Fig.16 Virtual Try-On Structure	32
Fig.17 Mask generation	32
Fig.18 U-2-Net Architecture	34
Fig.19 Human mask generation	34
Fig.20 Object mask generation	35
Fig.21 Virtual Try-On results before the optimization.....	35
Fig.22 U-2-Net Background Removal.....	36
Fig.23 Margin Problem	37
Fig.24 Margin removal algorithm	38
Fig.25 Virtual Try-on Final results (PF-AFN)	37
Fig.26 extract_urls Function content	40
Fig.27 extract_palette Function content	40

Fig.28 Data filtering algorithm	41
Fig.29 Data collected and filtered results	41
Fig.30 Similar data points typically exist close to each other	42
Fig.31 Tweaked KNN algorithm for palette prediction (n=1).....	43
Fig.32 Colors Palette Generator Structure	43
Fig.33 Aipalettes.com	44
Fig.34 T-shirt Size Recommendation Structure	45
Fig.35 T-shirt size guide based on height and weight	45
Fig.36 T-shirt size recommendation dataset sample	45
Fig.37 Lazypredict Classifier	46
Fig.38 Lazypredict Classifier Results	46
Fig.39 T-shirt Size Recommendation Results	47
Fig.40 Project Structure	48
Fig.41 Home Page	50
Fig.42 Product Page	51
Fig.43 Login Page	51
Fig.44 Shopping Cart	52
Fig.45 Checkout Page	52
Fig.46 Customize T Shirt Page	53
Fig.47 Virtual Try-on Page	53
Fig.48 Statistics Page	54
Fig.49 Orders Page	54
Fig.50 Products Management Page	55
Fig.51 Stock Management Page	55
Fig.52 Project management methodology	63
Fig.53 Incremental Methodology	64

Introduction

Fashion nowadays is really important in our life and also projects our taste in clothes, Fashion comes from the Latin word 'Facere' which means to make. Fashion is what we visualize. It is the inner feeling of every human being to present himself/herself according to one's own thinking. Fashion is a term commonly used concerning clothes, garments, textiles, etc. Fashion changes after a particular period. Around the world styles of dress and fashion are changing very fast and are becoming more and more diversifying curiosity, desire to be different, intimation, and combination are some of the reasons for the fashion change. Fashion changes from season to season because of the temperature variations by rain, humidity, heat, etc.

After this brief definition, let's talk about the customization of clothes, customized clothes are in demand for several reasons such as it may be suitable for different occasions in this part mostly they are presented as a gift with personalized text and images on the cloth; Add your special touch to the cloth, transfer your imagination to reality so you will be satisfied with what you are wearing also it can be stunning and you will be the designer of your own; Let people be creative. It will be easier if you can design and customize the product via your phone or pc in your home without visiting the store and what about even trying that product on you virtually and see how it looks using your image only or also measures your best-fit size based on your height and weight; it will be world-changing in the fashion industry and reduce a lot of time-consuming and errors like returning the product after the delivery because it doesn't fit you or even the design it's not how you want it. Many fashion designers waste too much time choosing what colors to pick for this cloth so why not exist a tool that helps us and recommends us the best color palette to use based on the first color we pick so our colors become more consistent. Imagine every option we talked about will be real and existing in our real world, isn't that a huge thing that will change our life in fashion design, fashion shopping and help the stores understand their customers more and more.

1. Classbox :

Classbox is a consulting firm where they give professional training in several fields such as pharmacy, tailoring, designing, ..etc. They had a project in mind that needed computer engineers to release it and here we are started working on it.

2. Problems and solutions :

After introducing what is the fashion industry is looking for we can easily identify what are the problems that it's facing and what are the obstacles to find a better solution for each problem, from those problems we can extract the main ones which are how to make fashion industries work online and without finding difficulties any in shipping the right products to their customers in terms of color, size, design and how to know if their picked product will really fit well on their body and the customer will like it.

Following a lot of researches, meetings, and discussions we did come up with several solutions for several problems; To make the fashion industry online is to use the notion of E-Commerce where it is an idea to sell the products online using a website. The best thing to see if the customer even like the product on his body is to try it online which we call Virtual Try-On, technically speaking the customer will upload a picture of himself after selecting the product and press the action button that will run the artificial intelligence algorithm in behind to the work and return us the same picture of the customer but with the product selected on his body next to the original picture so the customer can compare between the two pictures and see how it will be looking on him. Let's talk now about the design and colors, Sometimes the customer doesn't like the predefined design of a product so why not the customer be his own designer as we said in the introduction basically we give him the ability to design and customize the product with his own taste, he can upload images, add different colors, distinct texts with several fonts and for that, we will also provide artificial intelligence algorithm that generates best color palettes that fit the customer color input so the design will be compatible in colors. Finally, how can we predict the customer's best-fit body size? we have collected different data for different people about their size based on their age, height, and weight where will use it on our artificial intelligence algorithm and recommend the prediction to the customer, he will need only to enter those properties and see the results, he will not be forced to take the recommended size since it is just a recommendation. Now we have covered all our problems using as we expect optimal solutions.

3. Overview of our solutions :

1. Artificial Intelligence Algorithms

- a. Artificial intelligence (AI) is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence. AI is an interdisciplinary science with multiple approaches, but advancements in machine learning and deep learning are creating a paradigm shift in virtually every sector of the tech industry
- b. What are the technologies?

In AI there are many different algorithms and many distinct fields to use these algorithms as they are needed to be used. We will explain in each option that uses AI what technology it is using later on.

2. How to create a website?

- a. Creating a website will need development skills as known as coding. A website is separated into two parts Front End and Back End; Front End is what the customer or the user will see such as the interface, buttons, pictures...etc, In another part Back End is the opposite where everything is invisible to the user, where the server works and requests are sent to save data for example or request data.
- b. What are the technologies?
 - i. Front End
 1. HTML5 (HyperText Markup Language Version 5)
 2. CSS3 (Cascading Style Sheets Version 3)

3. Javascript
 4. jQuery
 5. Bootstrap
 6. AJAX (Asynchronous JavaScript And XML)
 7. Fabric.js (<http://fabricjs.com/>)
- ii. Back End
 1. Django Framework v3.2.5
 2. Python v3.8.5
 3. PostgreSQL

4. Objective :

Our objective is to create everything described in our introduction and solutions paragraph, We will list the main ones here :

1. E-Commerce Website for fashion products
 - a. Customer can :
 - i. Register
 - ii. Login
 - iii. Restore password
 - iv. Profile
 1. Edit shipping address
 2. Change password
 - v. Browse products
 - vi. Purchase products
 - vii. Customize his own product
 - viii. Virtually Try product
 - ix. Check his best-fit size
 - x. Browse products by category
 - xi. See related products
 - xii. Manipulate the shopping cart state
 - xiii. Save shopping cart state
 - xiv. Checkout
 1. as logged in
 2. as guest
 - b. Administrator Dashboard :
 - i. Manage products categories
 - ii. Manage products
 - iii. Manage orders
 - iv. See statistics
 1. Total users
 2. Total revenues
 3. Total orders
 - v. Send email notifications
 - c. Should be :

- i. Responsive
 - ii. User friendly
 - iii. Fashion style and theme
 - iv. Simple & Attractive
 - v. Real-time requests
 - vi. Eye relaxing
2. Artificial intelligence
 - a. Virtual Try-on
 - b. Size recommendation
 - c. Colors palette generator

5. Limitations :

1. Since we have three different AI options in our project it will be time-consuming to collect data and train it and see several accuracies to pick the more accurate ones, to train models we need high-end resources in compute-engines, especially in the virtual try-on model.
2. Data collection is a big part of AI however it's a big problem too, sometimes the data is unavailable so we need to construct it by scraping websites where we need permissions from those websites to allow us to scrap them. Forwardly data scrubbing, a very essential step to clean, parse, and sort the data with the output that we want. As we can see AI cannot work without Data science.
3. Fast response time while scaling from high-end resources to low-end in server deployment in inference after the training period.
4. Loading 3 models and all the data needed to be shown on the website, execution time has to be small and unnoticeable by the customer therefore we will try to optimize our requests and algorithms by tweaking the hyperparameters if there is.

Chapter I

Background

In this chapter, we will detail how algorithms of Artificial Intelligence will work and the steps of the website creation. Since there are multiple models & algorithms in AI we will compare each algorithm and choose the optimal one for our problem based on rapidity and accuracy, for the website we will base generally on the backend since the whole process and work is there rather than just the design itself.

I.1 Artificial Intelligence

In this chapter, we will explain artificial intelligence in general.

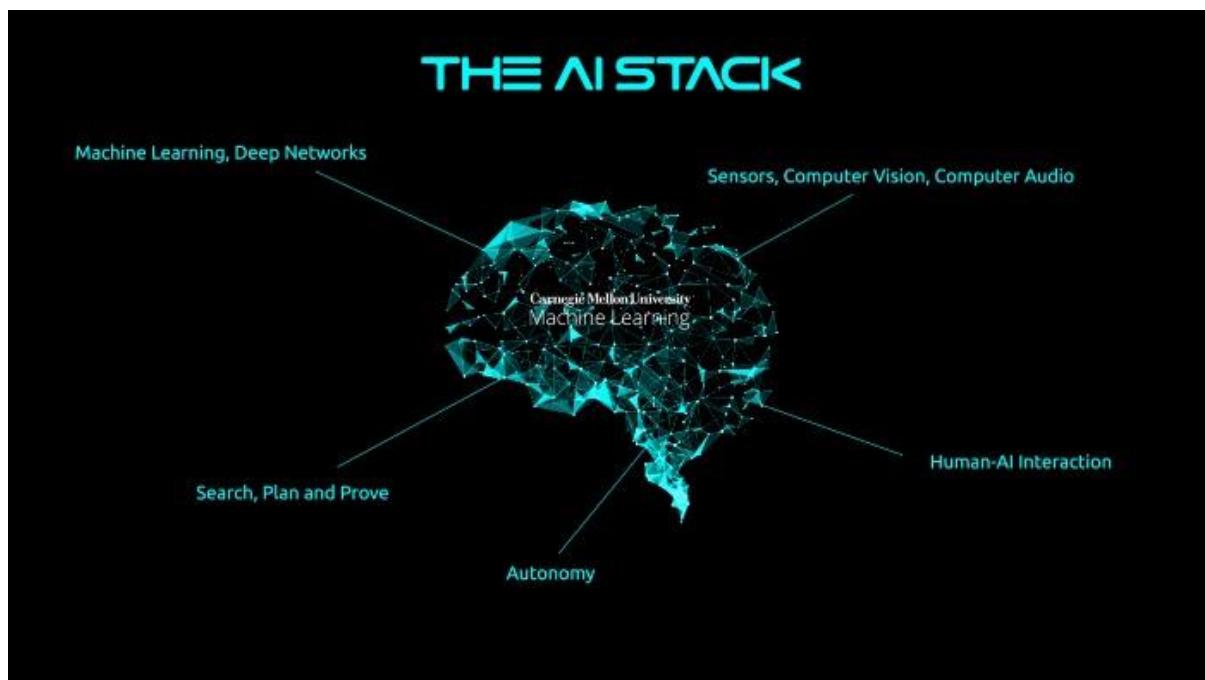


Figure 1. Artificial Intelligence and its main categories

The modeling of human intelligence processes by machines that are configured to think like humans and simulate their actions is referred to as artificial intelligence (AI).

The term can also refer to any machine that exhibits human-like characteristics such as learning and problem-solving.

The capacity of artificial intelligence to analyze and execute actions that do have the highest probability of attaining a given objective is its ideal feature. Machine learning is a subset of artificial intelligence that refers to the idea that computer systems can automatically learn from and be adaptable to changes without the assistance of humans. Deep learning

approaches facilitate automated learning by absorbing massive quantities of unstructured data such as text, pictures, or video.

When most people hear the words artificial intelligence, they immediately think of robots. This is because big-budget movies and literature depict human-like robots wreaking havoc on the planet. Nothing, however, could be farther from the truth.

Artificial intelligence is founded on the idea that human intellect may be characterized in such a manner that a computer can simply imitate it and carry out tasks ranging from simple to complicated. Simulating cognitive human processes includes artificial intelligence goals. To the degree that they can be concretely described, researchers and developers in the field are making remarkably quick progress in simulating processes such as learning, reasoning, and perception. Some think that in the near future, inventors will be able to create systems that can learn and reason about any subject faster than humans can. Others, on the other hand, remain skeptical, claiming that all cognitive activity is loaded with value judgments based on the human experience.

As technology improves, old artificial intelligence criteria become obsolete. Machines that calculate fundamental calculations or detect text using optical character recognition, for example, are no longer called artificial intelligence because these operations are now regarded as standard computer functions.

AI is constantly improving to help a wide range of businesses. A multidisciplinary method based on mathematics, computer science, linguistics, psychology, and other disciplines is used to wire machines.

I.1.1 Artificial Intelligence Applications :

Artificial intelligence applications are infinite. The technique can be used in several businesses and domains. AI is utilized for medication administration and various treatments in patients and operating room operations in the healthcare business [1].

Other examples include computers that play chess and self-driving cars. Machines of artificial intelligence. Each of these devices must measure the influence of any measures taken because the eventual result is affected by each action. The end result of chess is his victory. For self-driving automobiles, all exterior data must be taken into consideration and calculated in order to prevent a collision.

Artificial intelligence is also utilized in the financial sector to detect and flag bank and finance activities such as extraordinary use of debit cards or significant account deposits, all of which support the fraud department of a bank. Applications for Artificial Intelligence are also utilized to simplify and facilitate trade. This is done by facilitating the estimation of the supply, demand, and price of securities.

I.1.2 Machine Learning :

"Machine learning is the study of computer algorithms to enable computer programs to be automated through experience," says Tom M. Mitchell. Machine learning is a branch of artificial intelligence. ML is one way we expect AI to be accomplished. In order to find similar patterns and explore nuances machine learning relies upon work with datasets to examine and compare data.[3].

If you, for example, offer a machine study model with a lot of songs you like and their audio statistics (dance-ability, instrumentality, tempo, or genre). It is difficult to automate (depending on the supervised machine learning model used) and create a recommendation to suggest that you play music in the future, with a high likelihood rate, as well as Netflix, Spotify, and other companies.

In a simple example, loading an automated learning program that contains a large dataset of x-ray images along with the description (symptoms, things to take into account, and others) will make it possible that the data analysis on x-ray pictures can later be assisted (or perhaps automated). The learning model examines each image from the different datasets and searches for common patterns with comparable indications in images with labels. In addition (assuming we use an acceptable image ML algorithm) it compares its parameters with the examples previously collected when loading the models with new images to show how likely the images are to contain any indications it has previously analyzed.

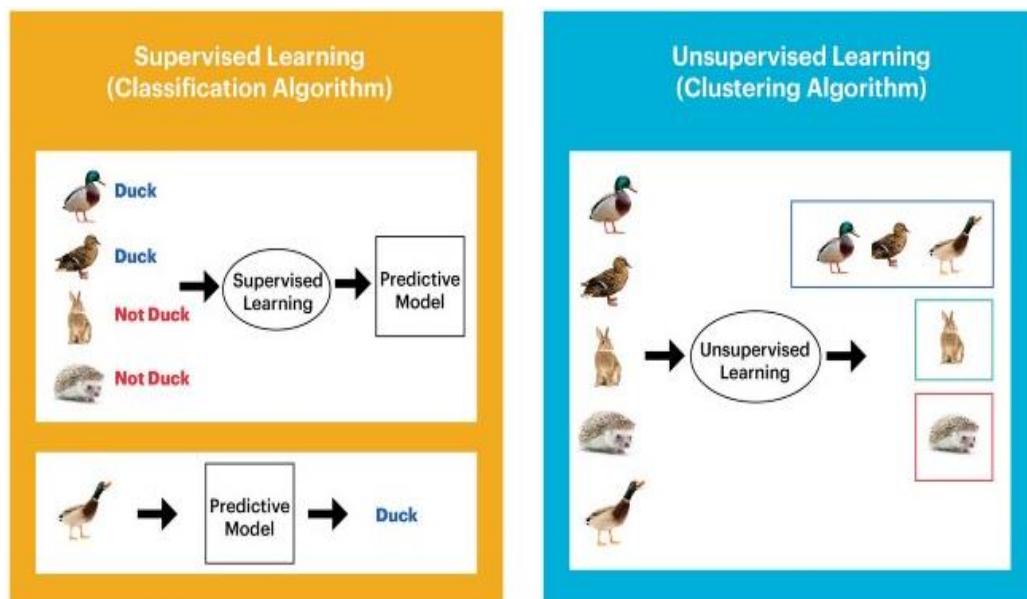


Figure 2. Supervised Learning (Classification/Regression) | Unsupervised Learning (Clustering)

The type of machine learning from our previous example, called "**supervised learning**," where supervised learning algorithms try to model relationships and dependencies between the target prediction output and the input features, such that we can predict the output values for new data based on those relationships, which it has learned from previous datasets fed

and it's called classification where there are several algorithms such as Super Vector Machine, k Nearest Neighbors, Neural Networks ...etc.

Unsupervised learning, another type of machine learning, is the family of machine learning algorithms, which have main uses in pattern detection and descriptive modeling. These algorithms have no output categories or data label(s) and are called clusters, just like supervised education, we have several algorithms extracting k-means, DB scan, Apriori, etc.

Semi-supervised training does the same trick as supervised learning, but it can use both labeled and unmarked training data. Often you look at many unlabelled data and a little labeled data in semi-supervised learning. Some researchers have found this process to be more accurate than non-monitored learning, but without the time and cost of labeled data.(Something like transcribing audio files or analysis of 3D pictures to make the creation of labels, which make it quite impossible to create a fully labeled data set, especially if you are using those massive data sets that are loved by profound learning tasks.)

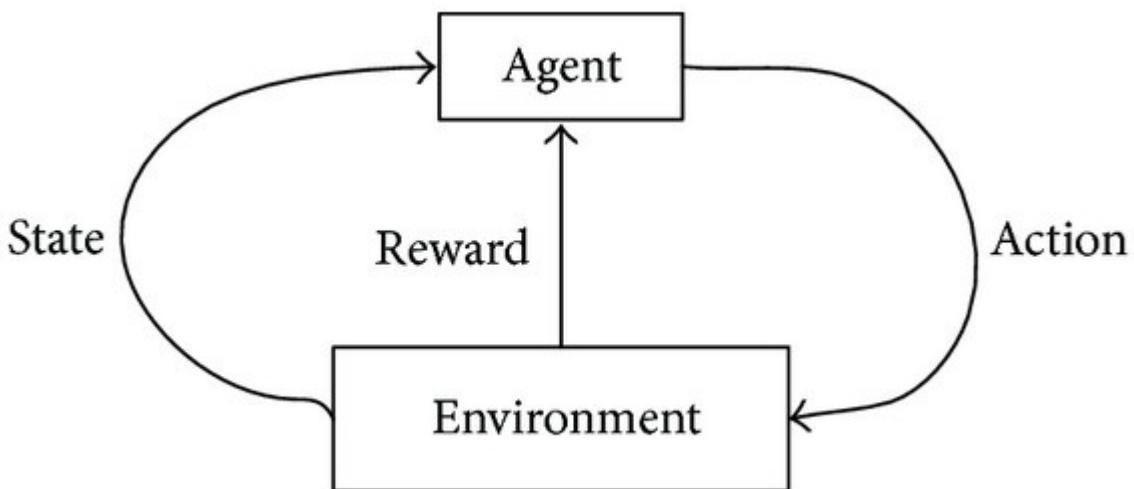


Figure 3. Reinforcement Learning

Reinforcement learning, the third popular type of machine learning, aims to take actions that maximize the reward and minimize the risk through observations collected through interaction with its environment. In this case, the enhancement learning algorithm (called the agent) constantly learns iteration from his environment. Computers reaching a superhuman state and beating people in computer games are a great example of reinforcement learning.

Machine learning, especially its advanced sub branches, i.e. deep learning and different types of neural networks, can be dazzling. In any case, it is "magic," irrespective of whether the public has sometimes problems observing their own inner functions. It is also "magic." While some tend to compare the functioning of the human brain with deep learning and neural networks.

I.1.3 Deep Learning :

Deep learning is a form of machine learning, which imitates the way people acquire certain kinds of knowledge. Deep learning, which includes statistics and predictive modeling, is an important data science element. This is extremely useful for data scientists whose task is to collect, analyze and interpret large quantities of information; deep learning facilitates and speeds up this process [2].

The easiest way to automate predictive analysis is to learn in-depth. Although classic machine learning algorithms are linear, deep learning algorithms stack themselves in an increasingly complex and abstract hierarchy.

Imagine a kid whose first word is a dog to understand profound learning. In pointing at objects and saying the word dog, the baby knows what a dog is — and isn't. "Yes, this is the dog," says the parent, "No, this is not the dog." As the baby continues to point to objects, the characteristics of all dogs become more aware. Without knowing this, the child clarifies a complex abstraction — a dog concept — by building a hierarchy in which the knowledge gained from the previous hierarchical levels of each level of abstraction is established.

A number of new deep learning architectures use multiple layers of multidimensional processing devices for feature removing and transforming, based on the Artificial Neural Networks (ANNs). Each layer uses the output for its input from the previous layer. What they learn forms a Hierarchy of Concepts, in which every level learns a little more abstractly and composites to transform their input data.

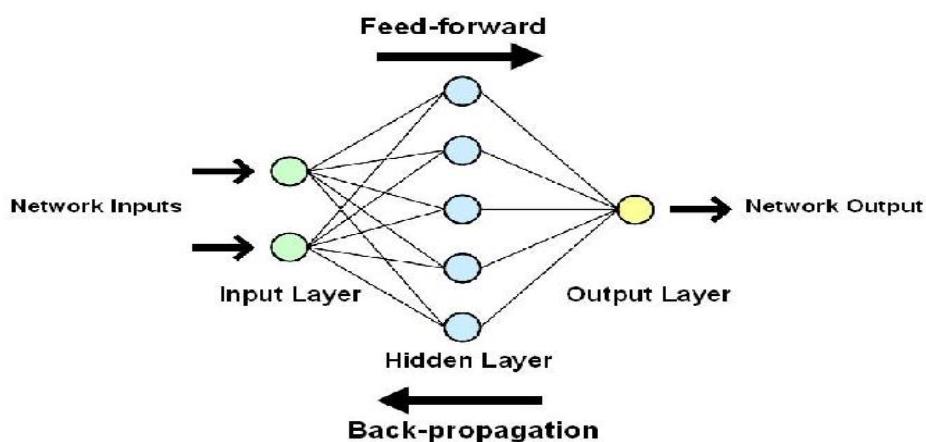


Figure 4. ANN (Artificial Intelligence Neural Network)

That means for an image, for instance, the pixel input could be a matrix, the first layer can encode the edges and compose the pixels, the next layer can make an edge arrangement, the next layer could encode the nose and the eyes, and the following layer can recognize that the frame has a face, etc.

While you may need to do a little fine-tuning, the deep learning process learns which features to place in which level on its own!

I.1.3.1 Convolutional Neural Networks :

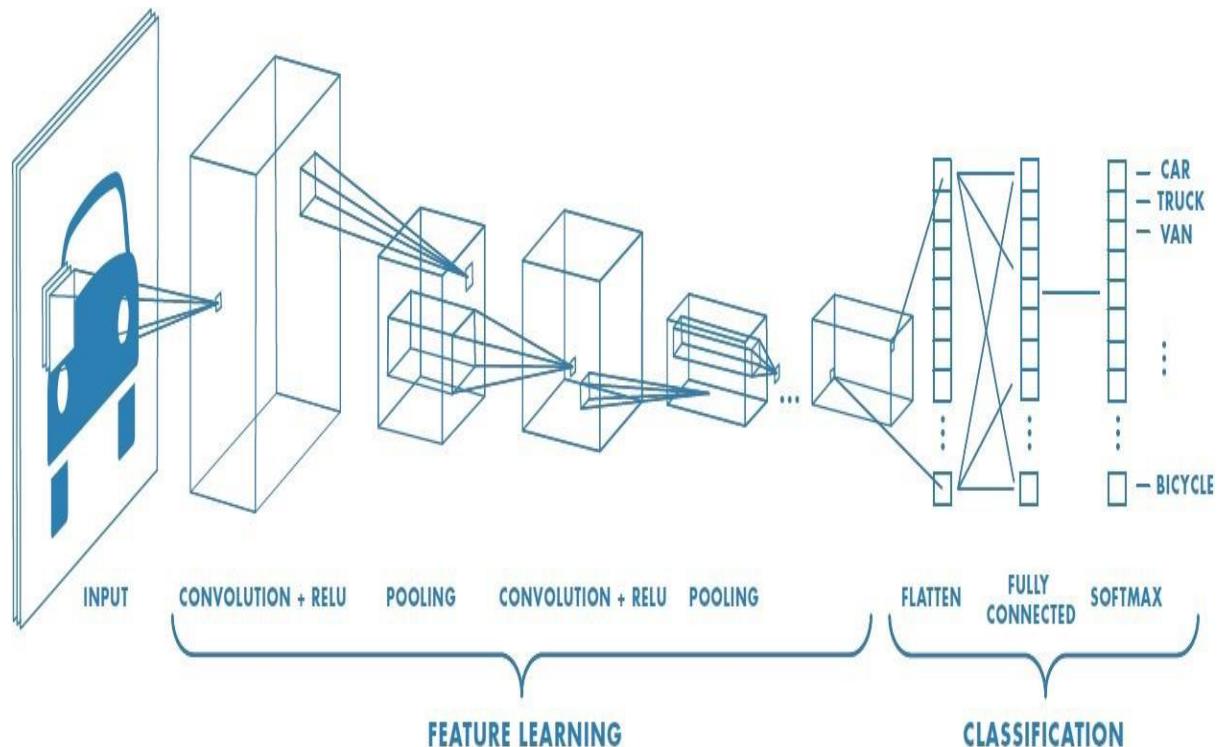


Figure 5. CNN (Convolutional Neural Network)

Convolutional Network (CNN) is a deep learning algorithm that takes an image input, assigns importance to several aspects/objects in the image (learnable biases and weights), and allows one to distinguish from the other.

In contrast to these other different classifiers, the pre-processing required in a Convolutional Network is much lower. While hand-made filters are made with sufficient training in primitive processes, ConvNets can learn these characteristics /filters[4].

ConvNet's architecture is similar to the connectivity pattern of neurons in the Human Brain and was inspired by the visual cortex' organization. Individual neurons only react to stimuli in a restricted visual field area called the receptive field. The entire visual area is covered by a collection of these fields.

Different architectures of CNNs are available that were key to developing algorithms that will have the power and capacity of the AI in the future as a whole such as ResNet, VGGNet, etc.

I.2 Technical notions

In this chapter, we discuss most of the technical terms and definitions.

I.2.1 Website

In 1989, when Tim Berners-Lee was working on CERN, a British scientist invented the World Wide Web (WWW). The Web was initially designed and developed to meet the demands of scientists in universities and institutions worldwide for automated sharing of information.

Web sites can be used in different ways: a personal website, a business website for a company, a government website, a website for the organization, etc. Websites may be the work of an individual, a corporation, or another organization and normally focus on a specific subject or purpose. Every website can contain a clickable link to any other website, which allows users to blur the distinction between individual sites.

In order to access the content, some websites require registration by users or subscriptions. Such web pages may include a number of business websites, news sites, academic newspapers, game websites, websites for file sharing, message boards, web-based emails, social networking sites, Websites that provide real-time inventory data, as well as websites that offer different services.

I.2.2 Responsivity

Responsive Web Design (RWD) is a web development technique that dynamically modifies the appearance of the website, depending on the display size and device orientation used to view it. RWD addresses the problem of designing multiple devices for customers, from small telephones to huge desktop monitoring.

I.2.3 API (Application Programming Interface)

The API is an acronym for an API, an intermediary software for communicating two applications. You will always use an API to send an instant notice or to check the weather on your phone, each time a Facebook app is used.

When you use an application on your mobile phone, the app connects to the Internet and sends data to a server. This data is then collected, played, and sent back to your phone by the server. This data is then interpreted and readable by the application. This is an API - everything is done via an API.

I.2.4 MVC (Model View Controller) & MVT (Model View Template)

The model view control (MVC) is an architecture pattern divided into three main logical parts for a single application: the model, the view, and the controlling system. Each component is designed to address specific aspects of application development. For the scalable and extensible industries standard projects, MVC is the most used web development framework.

Model

The model's component is consistent with all the user's data logic. This can represent the data transferred between the components of the view and the controller or any other business logic data. In a database, for example, customer information is retrieved, handled, and updated from and from the database by the object of the customer.

View

The view component is utilized for all the application's UI logic. The customer view contains, for example, the user interfaces with user components such as text fields, drop-downs, etc.

Controller

The controls provide as an interface between the Models and view components in order to process the business logic and receive requests, to manipulate data using the model component and to interact to achieve the final result. For instance, the customer controller handles all interactions and inputs in the customer view and changes the customer model database. The same controller is used for seeing client data.

Another design pattern comparable to MVC is Model View Template, sometimes called MVT. The design pattern of MVT, like MVC, similarly divides code into three pieces.

Template

The distinguishing part between MVT and MVC. Templates function as the display layer and are mainly the HTML code returning the data. The file contents might be static dynamically.

Differences between MVT and MVC

Model View Controller (MVC)	Model View Template (MVT)
MVC has a controller that drives both Model and View.	MVT has Viewed for receiving HTTP requests and returning HTTP responses.
The view tells how the user data will be presented.	Templates are used in MVT for that purpose.
In MVC, we have to write all the controller-specific code.	The controller part is managed by the framework itself.
Highly coupled	Loosely coupled
Modifications are difficult	Modifications are easy
Suitable for the development of large applications but not for small applications.	Suitable for both small and large applications.
Flow is clearly defined thus easy to understand.	Flow is sometimes harder to understand as compared to MVC.
It doesn't involve the mapping of URLs.	URL pattern mapping takes place.
Examples are ASP.NET MVC, Spring MVC, etc.	Django uses the MVT pattern.

Control Flow Of MVT

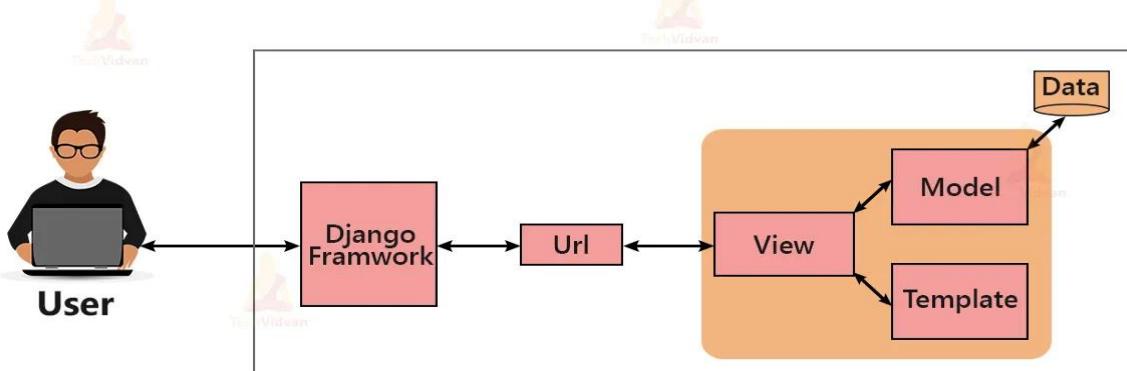


Figure 6. Control Flow of MVT in Django framework

I.2.5 ORM (Object-relational mapping)

Object-relation-mapping in computer science is a programming approach for converting data between the inconsistent types of systems by target-oriented programs. This produces a "virtual object database," which can be used as a programming language. Both free and commercial products map objects, while some programmers decide to construct ORM tools themselves.

Data management activities operate on objects which in object-driven programming are almost typically not in scale. An individual book of addresses, for example, should be taken into account with zero or more numbers and zero or more addresses. This can be modeled in an object-oriented execution of a personal object with an attribute/field to keep any element in data that is included in the entry: name of a person, phone number, and address list. The phone number list would include "PhoneNumber Objects" and so forth. In the programming language, each book entry shall be considered as a single item (it can be referenced by a single variable containing a pointer to the object, for instance). The object may be linked to additional methods such as returning the phone number, address, etc. you have asked.

Many typical databases such as DBMS, by comparison, are not object-oriented and can only store and process scalar information such as numeric tables and texts. The programmer either transforms (and then recovers) or only uses the scalar value in the program, an object value into groups of easier storage values. The first technique is to carry out mapping of objects.

The main challenge lists the objects in an atomized form and may be saved in the database while maintaining the properties and connections of the objects, so that they can be reloaded as an object if necessary. If this method of storage and recovery is adopted, the objects will be durable.

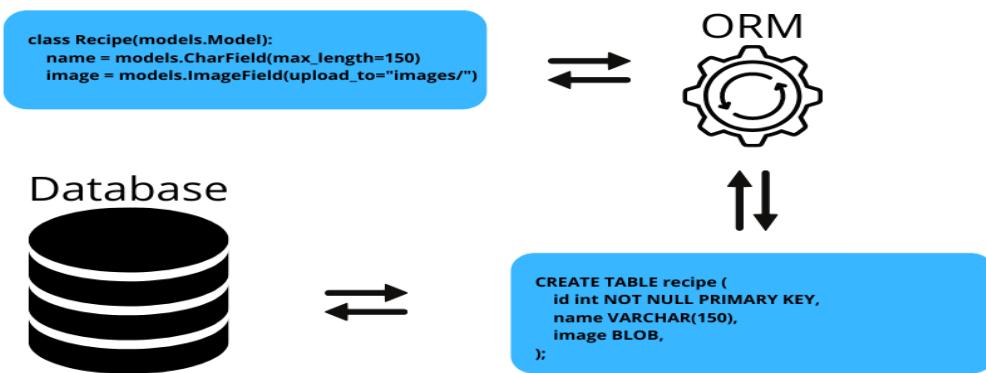


Figure 7. Django Model-based database using ORM

I.2.6 Front End :

The front end uses technologies such as the language of hypertext markup, JavaScript, and case sheets A combination of the front end (CSS). Developers are leading the way in creating and building websites and app users with experience, including buttons, menus, pages, links, graphs, etc.

HTML

The design and functioning of the full site are supplied using Hypertext Markup Language. In late 2017, the newest version was published, known as HTML5.2.

The new version provides more tools for the development of web apps to promote interoperability.

CSS

Cascading sheet styling enables developers the freedom and precision to construct attractive and dynamic website designs.

JavaScript

Dynamic components can be created on static web pages. The language of this event.

It enables developers to access different elements and reply on the main HTML page to server events.

Front-end frameworks including such Angular, Ember, Backbone, and React are also popular.

These frame models help developers to cope without losing the quality of the increasing demand for company software.

One of the main issues in the development process, which is also known as 'customer side development,' is the rapid pace of change in the tools, methods, and technology necessary for user experience for applications and websites.

A transparent user interface that is easily monitored due to the different sizes and resolutions of the mobile device or computer screen is apparently a primary purpose.

When it comes to the Internet of Things (IoT), things become much more confusing.

Screen size and network connection now have a greater range of applications for developers to work on their UIs to solve these difficulties.

I.2.7 Back End :

The background known as the server-side includes a server that provides data on request a channeling application and the information database

For instance, if a consumer navigates shoes on a website, they interact with the front end.

After choosing the item, put it into the shopping cart, and allow the transaction, the information will remain in a database on the server.

Some days later, when the customer checks the delivery state, the server takes up the required information, updates, and shows tracking data.

Back End Tools

The main concerns of back-end developers include application building that finds and distributes data on the front end.

Many of the businesses use trustworthy databases such as Oracle, Microsoft SQL Server, Teradata, IBM DB2, and EnterpriseDB.

MySQL, NoSQL, and PostgreSQL are also accessible as other popular databases.

Apart from Ruby on Rails, Java, C++/C/C#, Python, and PHP, the applications have a wide choice of frames or languages.

Over the last many years, back end-to-service (BaaS) providers have been a feasible solution. They are particularly useful to design and operate on mobile apps in a tight schedule.

Chapter II

Solution Design & Technologies

In this chapter we are going to cover the process of designing our web platform and the technologies that we are going to use to release the web application.

II.1 Solution Design :

II.1.1 Use case diagrams :

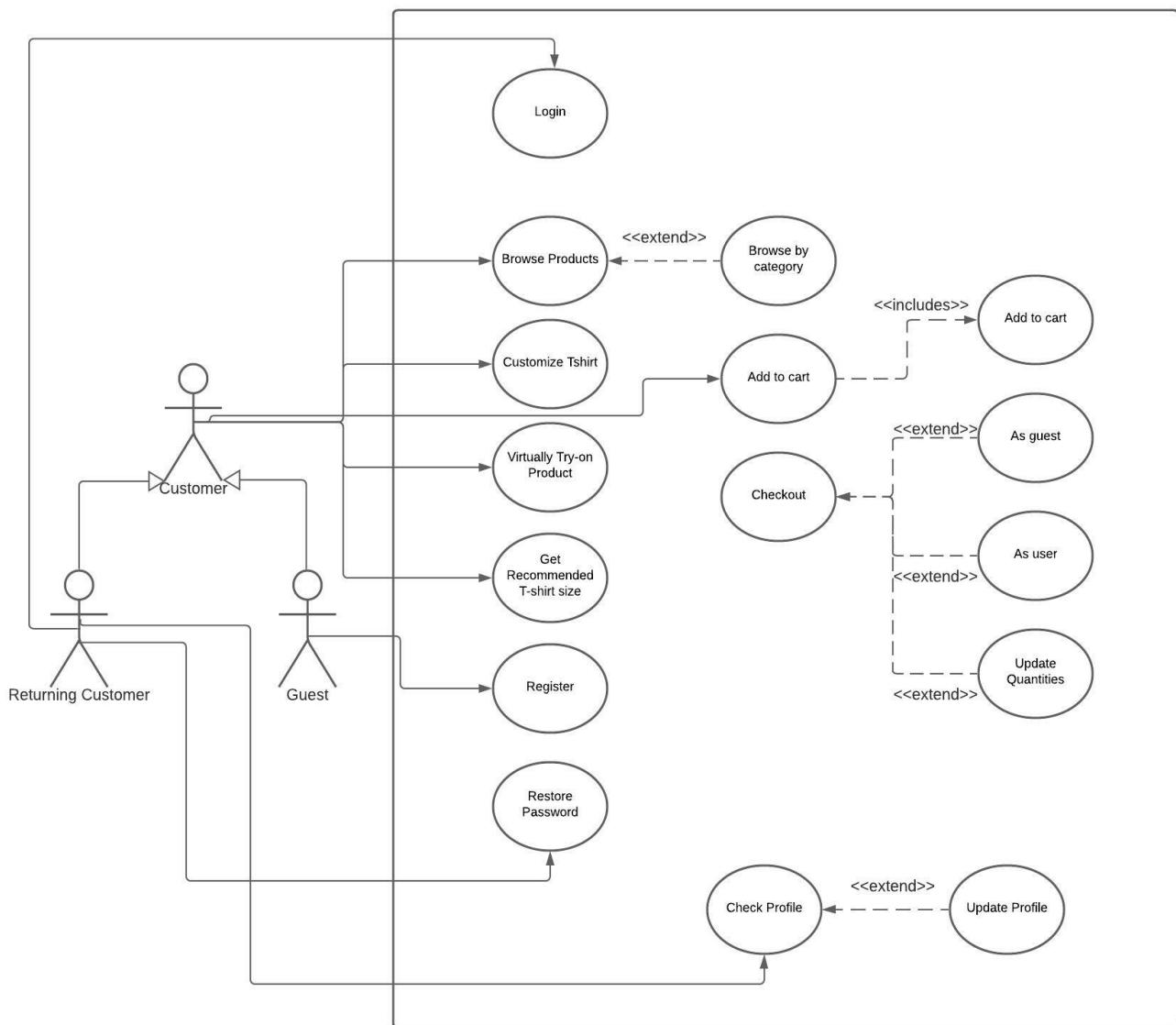


Figure 8. Client Use Case Diagram

We can see that the customer can use all the features that we have discussed in our solution description in the first part of this report such as customizing t-shirt, virtually try products...etc.

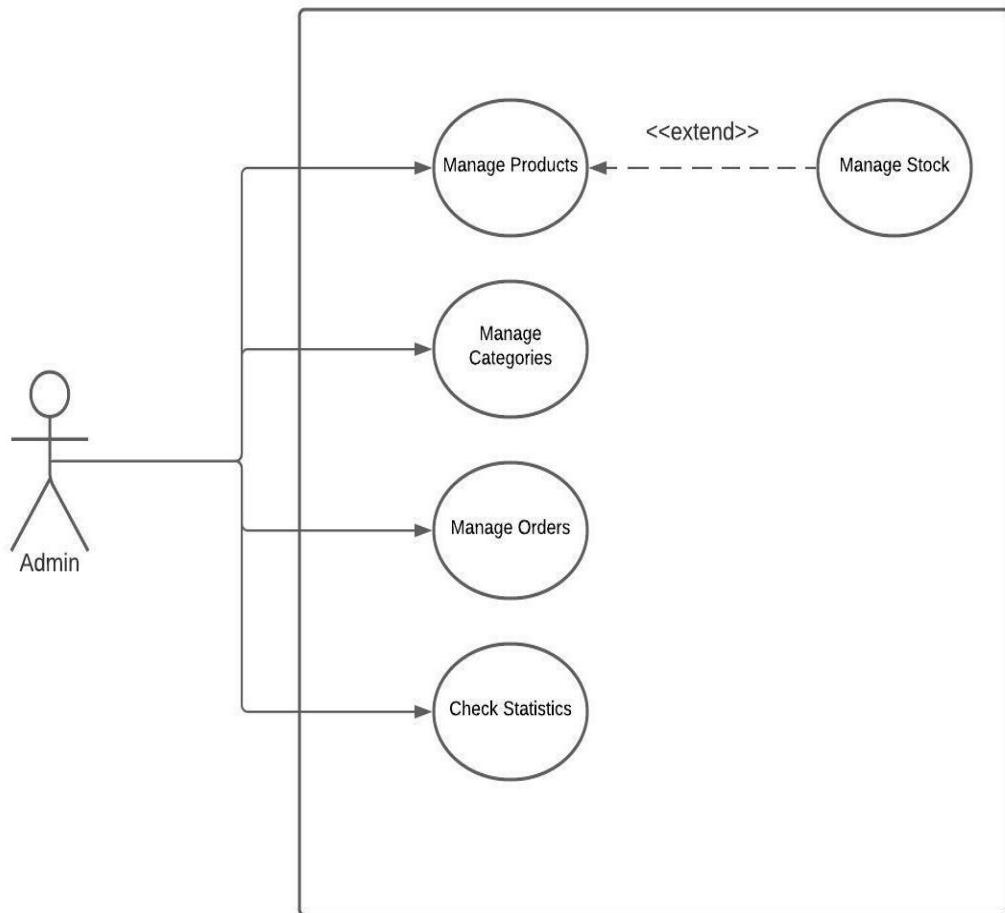


Figure 9. Administrator Use Case Diagram

About the administrator, he can manage everything in the website such as products and their stocks, categories products, orders and also check how the website is doing using the statistics such as how many users, the revenue..etc.

II.1.2 Component Diagram :

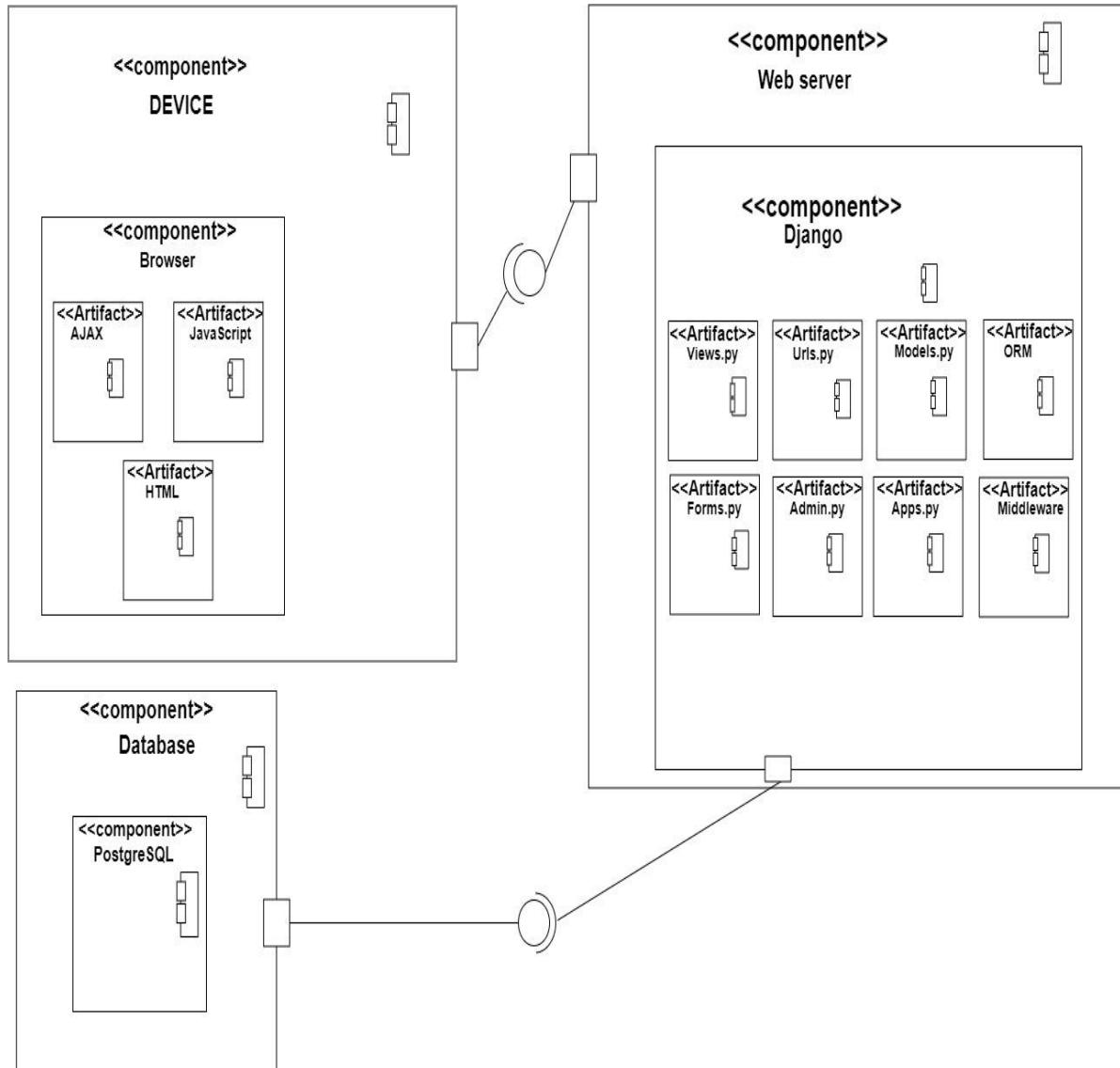


Figure 10. Component Diagram

Django component will have two distinct inner applications which are the Home application for the website and the Dashboard application for the Admin panel, this component will include the database to store and retrieve data. The user device will use the Webserver to interact with the Django component and can manipulate the website.

II.1.3 Deployment Diagram :

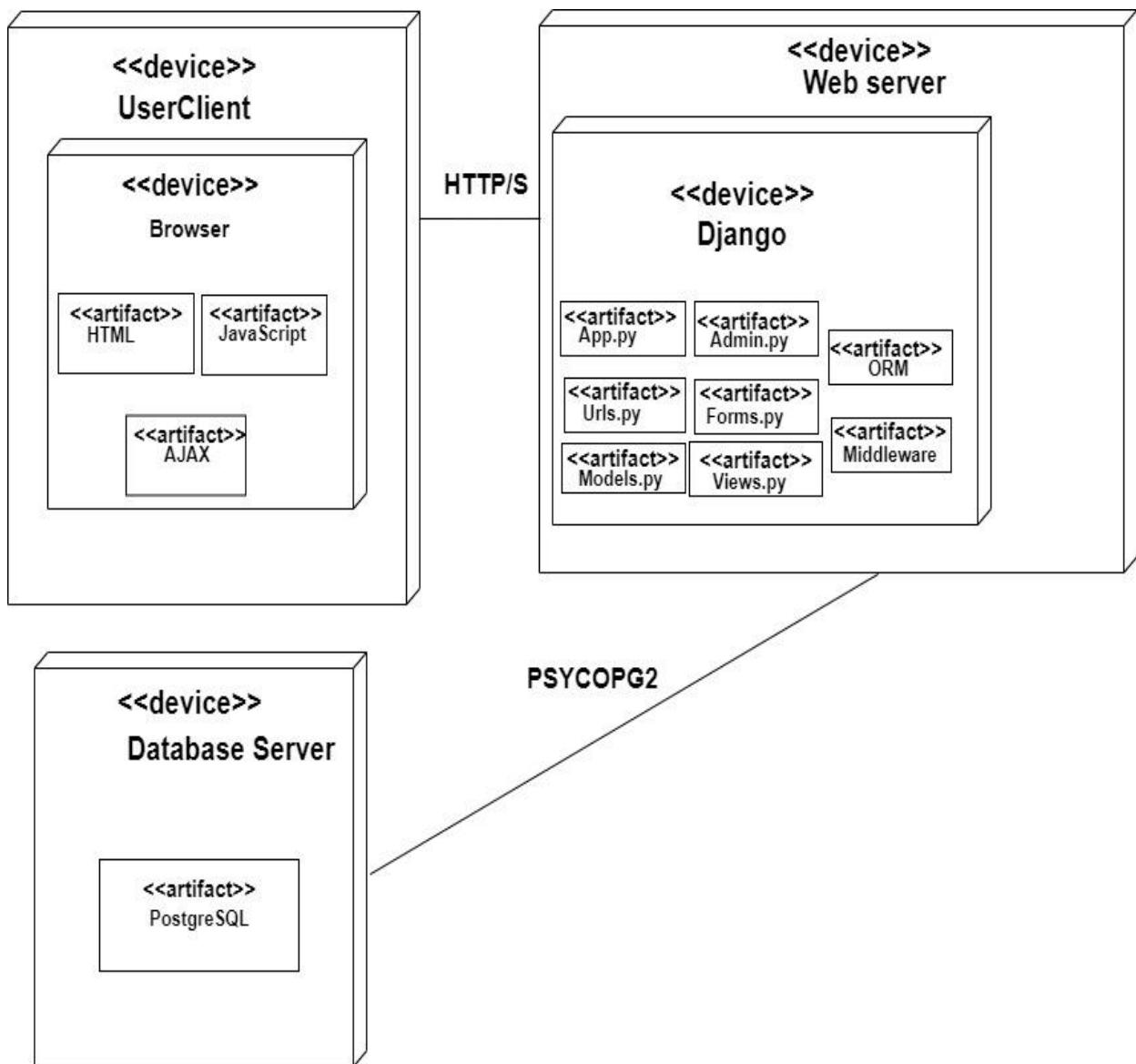


Figure 11. Deployment Diagram

To deploy this project we need two servers one for the web application where to hold the Django files and a second server to install our database and data store; to communicate between those servers they use **PSYCOPG2** protocol, which Django uses to communicate with a Postgresql server.

II.1.4 Class Diagram :

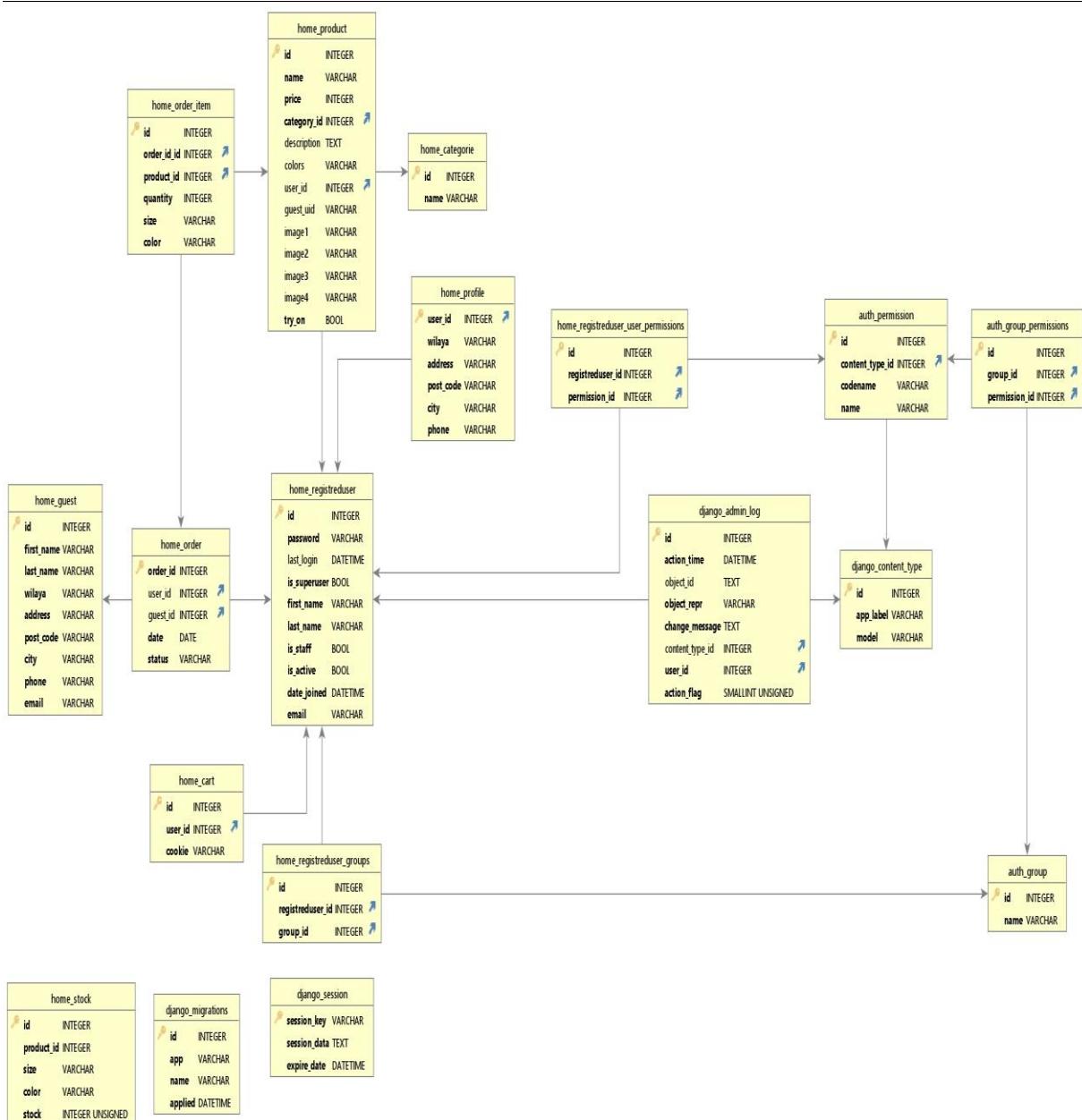


Figure 12 Class Diagram

This diagram is used to explain how the database layer is composed as we can see that each product has its own stock, each order has its own order items, the profile of both user and guest. Attributes in bold are required and they never can be nulled.

II.2 Technologies:

In this chapter, we will list all technologies that we will be using in our solution implementation.

II.2.1 Artificial Intelligence Technologies :

Environnement :

Python :



Python has become in a few years a major language in the world of applications focused on data processing. It is an object-oriented language for expressing concepts very complex in a few lines of code. And therefore greatly facilitates prototyping, which explains its popularity and uses in machine learning.

Libraries :

PyTorch :



PyTorch is a Torch Library based on Open Source machines, largely established by Facebook's AI Research Laboratory (FAIR). It is provided freely and under the modified BSD license. PyTorch also has a C++ interface, even though its Python interface is crisper and the focus of development.

Pandas :

The Pandas logo features a stylized graphic of vertical bars in various colors (blue, purple, yellow) followed by the word "pandas" in a lowercase sans-serif font.

Pandas is Python's data management and analysis software programming software package. The data structures and algorithms are included for manipulating numerical tables and time series. Free software is provided through the BSD tripartite license. The name is derived from the word "panel data," which is an econometric sentence for collections of data that include observations for one person over a period of time. It's a pun on the sentence "Python Data Analysis." From 2007 to 2010 WesMcKinney began to create what pandas are when he was a research fellow of AQR Capital.

Numpy :



NumPy is a Python language programming toolkit that supports a broad range of high-level mathematical operations, huge multi-dimensional arrays, and matrices. The ancestor of NumPy, Jim Hugunin, was first designed by Numeric for a few other developers. In 2005, Travis Oliphant manufactured NumPy with significant enhancements by combining the Numarray competitor qualities with Numeric. NumPy is an open-source project that has a lot of contributions.

Scikit-Learn :



Scikit-learn is a free Python programming machine learning package. The Python NumPy and SciPy NumSciPy NumBooks include several classifications, techniques of regression and classification, including vector support, the random forests, gradients boosts, k-means and DBSCAN.

OpenCV:



OpenCV is a programming library that has mostly been built for computer vision in real-time. Intel originally created the Willow Garage, then Itseez (which was later acquired by Intel). The library is cross-platform and can be used free of charge under the open-source Apache 2 License. OpenCV enables real-time GPU acceleration starting in 2011.

PIL :



PIL is the open-source library that enables many different image file kinds to be opened, manipulated, and stored. PIL is a free source library. PIL is a Python language-free library. Windows, Mac OS X, and Linux are included. The most current releases for PILs in September 2009 are 1.1.7 or Python 1.5.2-2.7.

Requests :



Requests is a programming package of HTTP Python language. The project aims to make HTTP requests easier and easier. The most recent version 2.26.0. Applications are released with the Apache License 2.0. Requests are one of Python's most popular non-Python libraries. Requests to be distributed with Python by default have been suggested.

Tools :

Anaconda :



Python and R are the science computer programming languages distributed for the administration and deployment of packages (data science, applied machine learning, large-scale data processing, predictive analytics, etc.). The distribution includes data science packages for Windows, Linux, and macOS. It was developed and maintained by Anaconda, Inc., which was established by Peter Wang and Travis Oliphant in 2012. Anaconda Inc. is also known as Anaconda Distribution and Anaconda Individual Edition, but both Anaconda Team Edition and Anaconda Enterprise Edition are not free.

Conda system package management maintains package versions of the Anaconda package. The Paket Manager was used as a separate open-source package for its own and other uses, other than for Python. An Anaconda, a small bootstrap called Miniconda, contains only conda, Python, the packages that they depend on, and few more packages.

Google Colab :



In short, the collaboratory is a product of Google Research. Colab enables any person through their browser to write or execute arbitrary Python code, suitable for machine learning, data analysis, and training. Colab is technically a Jupyter host notebook service which needs no configuration and provides free access to GPUs, including resources for processing.

II.2.2 Web Platform Technologies :

Django :

Django is a free, open-source Python Web framework, which follows the model's architectural model views (MTV). This organization, which is an independent US-based non-profit corporation, is maintained by the Django Software Foundation (DSF).

Django's major goal is to enable the building of sites based on a sophisticated database. The framework emphasizes reusability and "pluggability," reduced coding, low connectivity, rapid development, and a notion that does not recur. Python is used everywhere, even for settings, files, and models of data. For the administrative creation, reading, updating, and removal options, Django offer a dynamically created, introspective, and configurable interface via administrative models.

How's Django's code? A Web app awaiting HTTP Web browser programs on a basic data-driven Web site (or another client). On reception of the application, the application will be determined using the URL and possibly in POST or GET data. It can then read or write data from a database, or carry out any additional actions required to fulfill the request. The program gives a response to the web browser, often using an HTML template to dynamically construct an HTML page that may be shown by the browser.

Django Web apps often group code into distinct files for each of these phases:

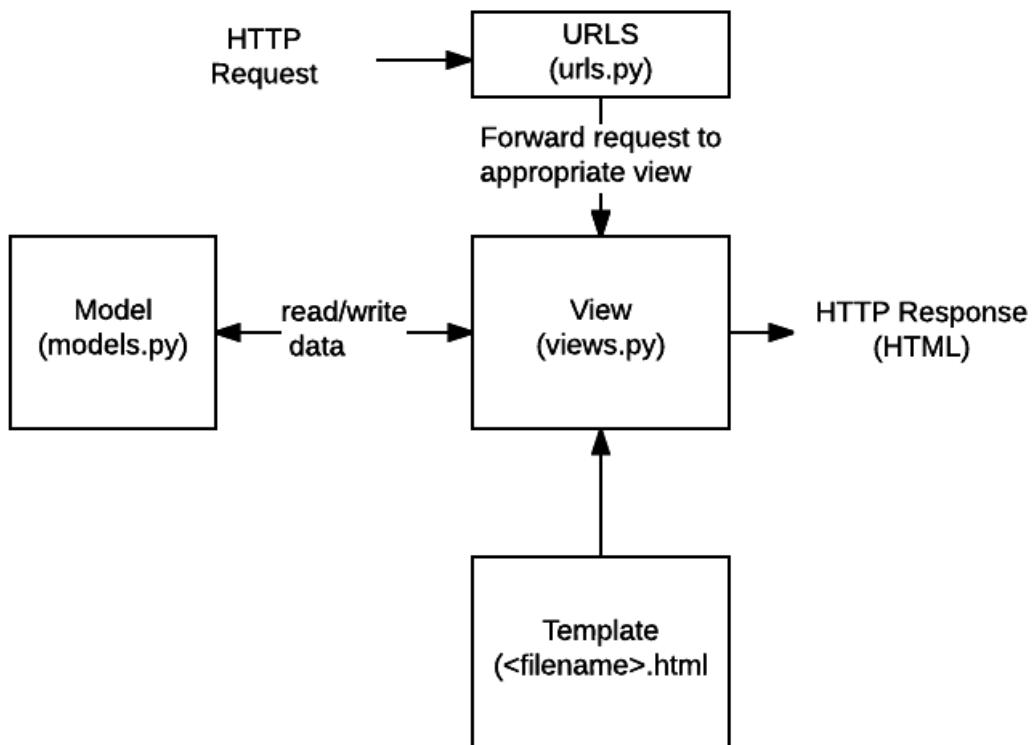


Figure 13. Django modelisation

- **URLs:** While requests for every URL can be processed with a single function, each source can be handled much easier with a distinct view function. Upon routing HTTP requests to the corresponding URL view, the URL mapper is employed. Additionally, in a URL that is displayed in the view function as data, the URL mapper can match specified strings of digits.
- **View:** A view is a request handler that receives HTTP requests and returns HTTP responses. Views access data needed for modeling and template formatting to satisfy requests.
- **Models:** Models are Python objects which determine the data structure of the application and give application administration tools, database query records (addition, updated, deleted).
- **Templates:** A template is a text file that uses placeholders to represent actual content, to define a file structure or style (such as the HTML page). An HTML template, reinforced by model data, allows the dynamic build of the HTML page. A template can define the structure of any file: this cannot be HTML!

Why Django :

- It's fast and simple

One of the key objectives of Django is to ease development labor. The Django context utilizes this:

- Rapid development principles that enable developers to perform more than one iteration at a time without starting the whole schedule;
- DRY philosophy — Don't Repeat Yourself, a philosophy that enables developers to reuse existing code and focus on a single one.

As a result, it takes much less time to get the project to market.

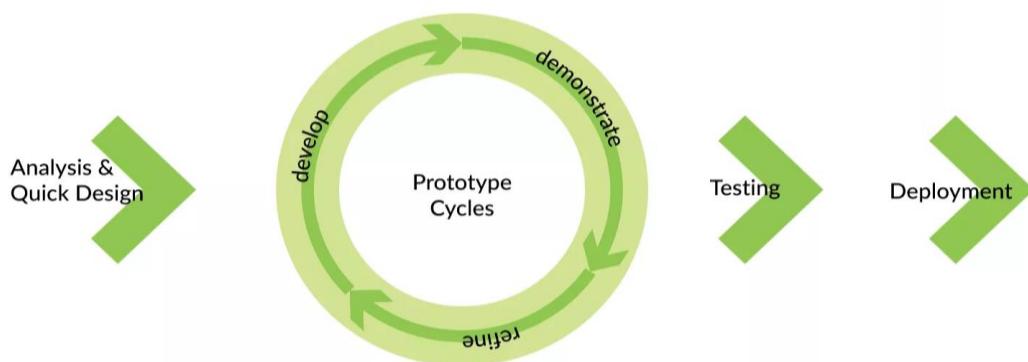


Figure 14. Django Development Methodology

- It's secure

For Django safety is also a top consideration. It offers one of the greatest safety systems out of the box out there and helps developers avoid typical security problems, including

- clickjacking,
- cross-site scripting
- SQL injection.

Django releases new security corrections quickly. In general, the first to react to vulnerabilities and warn of other frameworks is the first.

- It suits any web application project

Django allows you to handle projects of any size, whether they are an essential website or a wide web application. Why use Django for your project? Because:

- It is fully loaded and extendable to manage large volumes of data and heavy traffic applications;
- It is cross-platform to support your project with the Mac, Linux, or PC;
- It uses the most significant databases and simultaneously allows a more suitable database for a particular project or numerous databases.

- We can implement our AI models easily

Since our models are using the same language (Python) as Django it's quicker and easier to implement and link it with our backend.

- It's well-established

By time and crowd, Django was tried. It has access through many forums, channels, and websites to a huge, supportive community. If the code has an issue, help can be easily found, and you can find developers to base the next project on Django.

Django started the best of any other open source frameworks with superb documentation. And it is still kept high, updated with new functions and corrections, so that you can adjust easily to changes.

You may be confident that any problems with the framework are resolved as soon as they arise. The software is regularly updated and new packages are published to improve the convenience of working with Django.



PostgreSQL :

PostgreSQL is also known as Postgres, a free and open-source database system (RDBMS). The database at the University of California, Berkeley, was originally built as a successor to the Ingres database called POSTGRES. 1996 saw SQL support revamping the PostgreSQL Project. In 1996. The development team decided to retain its name after evaluating the PostgreSQL and the Postgres alias in 2007.

Transactions with Atomicity, Consistency, Isolation, Durability (ACID), automatically updateable views, materialized views, triggers, foreign keys, and storage processes are included in PostgreSQL. It is designed to handle a range of workloads with many competing users, from individual PCs to data warehouses or web services. It is the default macOS database and is available for Windows, Linux, FreeBSD and OpenBSD as well.

Database Performance

MySQL has always been known as a fast-reading-length working database, frequently in combination with writing operations at the price of the competition.

PostgreSQL, commonly known as Postgres, is proclaimed as "the strongest open-source relationship database in the world." It has been designed to be rich in features, consistent with extensions and standards. In the past, the performance of Postgres was more balanced – generally slow to read than MySQL; but, enormous volumes of data were more efficient to write and competitors were better handled.

Recent versions have largely eliminated the difference in performance between MySQL and Postgres. MySQL is highly quick to read data, only if the old MyISAM engine is used. When using InnoDB (which enables transactions, key limitations and other significant capabilities), the variation is insignificant (if they even exist). These functions are essential for businesses or consumers, so the use of the old engine is not an option. On the other side, the flaw in the writing of massive data has also been optimized by MySQL.

When deciding between MySQL and PostgreSQL, the performance of most running mill applications shouldn't be an issue – in any case it will be enough, however you consider anticipated growth in the future. Both platforms are entirely replicable and a large number of cloud providers have the management of both database versions. Therefore, before starting your next project using your default databases, you should take other benefits of Postgres over MySQL into consideration.

Postgres Advantages over MySQL

Postgres is an object-related database while MySQL is an object-related database. Postgres, therefore, covers the skills required for certain uses, such as table heritage and overload. Postgres is also closer to the SQL standards.

Postgres handles concurrency better than MySQL for multiple reasons:

Without reading locks, Postgres executes multiple version controllers (MVCC) Postgres enables simultaneous query strategies that can employ several CPUs. Postgres allows non-block indexes (using the syntax "BUTTER INDEX") and partial indexes can be created (for example, if you have a model with soft deletes, you can create an index that ignores records marked as deleted) Postgres is known at the transaction level for maintaining data integrity. Also, PostgreSQL is recommended by the Django creators “If you're not tied to any legacy system and have the freedom to choose a database back-end, we recommend PostgreSQL, which achieves a fine balance between cost, features, speed, and stability. (*The Definitive Guide to Django*, p. 15)”.

Libraries :

JS-Cookie :



Js-cookie is a simple, lightweight JavaScript API for handling cookies, we need to use it to handle our cookies and the shopping cart mechanism where it's all based on client-side cookies.

jQuery :



JQuery is a quick, compact, and feature-rich library of JavaScript. This reduces issues such as HTML documents, event handling, animation, and Ajax transmissibility and change by offering a user-friendly API for a range of browsers. With a combination of adaptability and extensibility, JQuery has changed the way millions of people write JavaScript.

Bootstrap :



Bootstrap is an open-source CSS framework that is responsive to mobile web development. It includes CSS design templates for JavaScript typography, forms, buttons, navigation, and other interface components.

Tools :

Visual Studio Code :



Visual Studio Code is an editor of the source code for Windows, Linux and MacOS made by Microsoft. Features include debugging aids, syntactic emphasis, smart code completion, snippets, code refactoring and built-in Git. Users can change subjects and keyboards, preferences and installations in order to give different benefits.

Deployment :

Nginx :



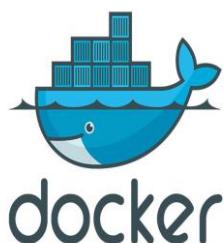
Nginx is a Web server, a load balancing machine, a Mail Proxy, and an HTTP cache. The software was developed by Igor Sysoev in 2004 and was publicly published. Nginx publishes free and open-source software under the 2-clause BSD licensing. Typically, NGINX is used as a load balancer by numerous web servers.

Gunicorn:



HTTP Server is known as the 'green unicorn' (or pistol unicorn) or Python web server gateway interface (WSGI). It is a pre-forked model, implemented by the Ruby Unicorn project. The Gunicorn server has a wide range of applications, is easy to deploy, and has cheap server needs and a range of web frames.

Docker:



Docker (Docker) is a collection of PaaS technologies that deliver the software using so-called container packages, leveraging OS-level virtualization. Containers are separated from one another and can communicate through clear channels, combining software, libraries and configuration files. Since all containers share the services of a single kernel, they have fewer resources than virtual machines.

Chapter III

III. Artificial Intelligence Algorithms

In this chapter, we will detail each AI algorithm that we have implemented in this project.

III.1 Virtual Try-On :

In virtual try-on, many projects and research papers exist talking about the same subject but the only problem is accuracy, every year there is CVPR (Conference on Computer Vision and Pattern Recognition); a conference to present the most recent results of research in the Computer vision field and Virtual Try-On is a subcategory of it.

Our solution is to bring the best accuracy model that exists which is **PF-AFN** [5] (Parser-Free Virtual Try-on via Distilling Appearance Flows) from **CVPR 2021**, At first, we have tried ACGPN (Realistic Virtual Try-On by Adaptively Generating Preserving Image Content) [6] but there was a huge difference after we have tested the one we have used PF-AFN.



Figure 15. Comparison of different virtual try-on models

PF-AFN is using DL (Deep learning) and VGG Model (PyTorch), when we say DL we say CNN (Convolutional Neural Network) since we are processing images as input and output. First, we need to collect data, luckily VITON (An Image-based Virtual Try-on Network) [7] made their data public so we can use them and won't lose time to keep collecting millions of pictures of humans and clothes. Now we need to train and test our model with this data.

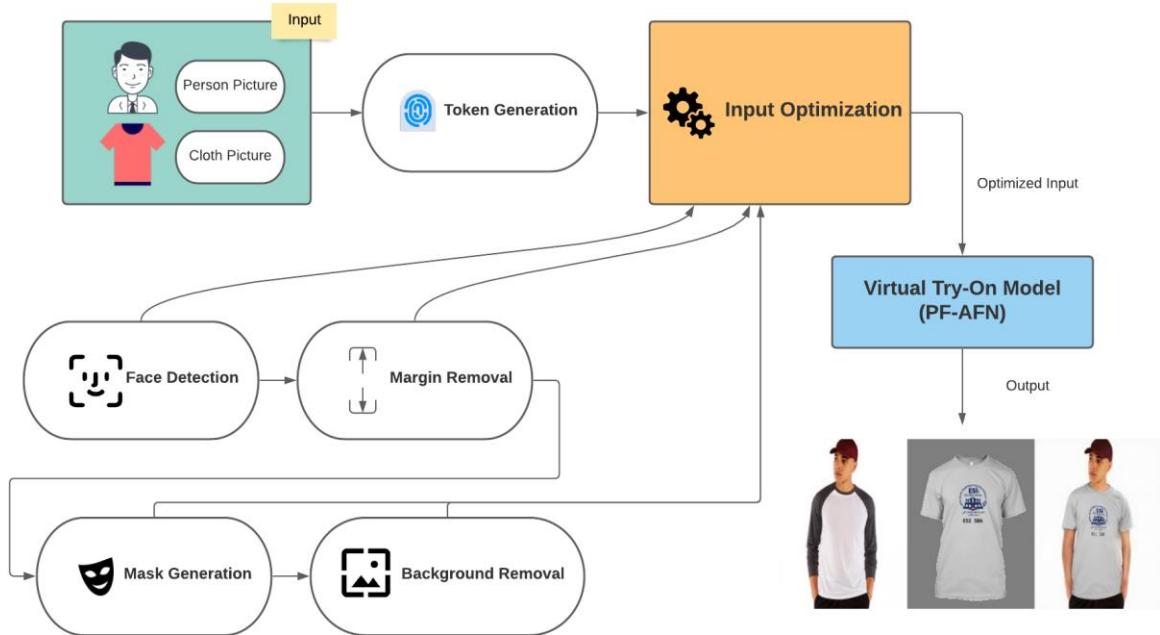


Figure 16. Virtual Try-On Structure

III.1.1 Data :

VITON Dataset is separated into three main folders train_img,train_clothes,train_edge where train_img contains human picture; train_clothes contains the cloth that the human is wearing; test_edge contains the mask of that cloth

Example :



Figure 17. Mask generation

III.1.2 Training :

Since our data is too big literally millions of data we need a big compute engine to train it otherwise we will keep waiting for months, fortunately, PF AFN provide the pre-trained model where they used 8 GTX1080 GPU as compute engine

The original results from PFA AFN weren't that satisfying so we decided to improve it by pre-treatment of the image input such as removing the background, scaling the image up to put the head exactly to the top of the image with a small margin.

To not mix up things the trained model is run under conda virtual environment we these specific requirements:

- Anaconda3
- pytorch 1.1.0
- torchvision 0.3.0
- cuda 9.0
- cupy 6.0.0
- opencv-python 4.5.1
- python 3.6

Anaconda & Installation :

Anaconda Individual Edition is the world's most popular Python distribution platform with over 25 million users worldwide. You can install and manage all packages in a simple way although it's built for Data Science & Machine learning. The installation is simple, just run the setup from <https://www.anaconda.com/products/individual> and everything will be done.

Environment Installation :

- conda create -n tryon python=3.6
- source activate tryon or conda activate tryon
- conda install pytorch=1.1.0 torchvision=0.3.0 cudatoolkit=9.0 -c pytorch
- conda install cupy
- pip install opencv-python

Now after installation, the training part is to download first the VITON Data and put it under the "dataset" folder and the VGG Model from PyTorch under the "models" folder and start the training by running the command **python scripts/train_PFAFN_stage1.sh** for the first stage, and second stage **python scripts/train_PFAFN_e2e.sh** .

When to stop the training? Mostly when the loss is very small.

III.1.3 Testing :

It is time for the inference after the training of our model, it does require at least one Nvidia GPU to test it and see the results predicted by the AI.

To run the test, we need to place our test inputs in the dataset folder under the PF-AFN_test folder, however, we had a problem generating the edge image of the cloth (mask image) after doing some thorough researches we found a pre-trained model that will generate the desired mask image for the input and it's called U-2-Net [9].

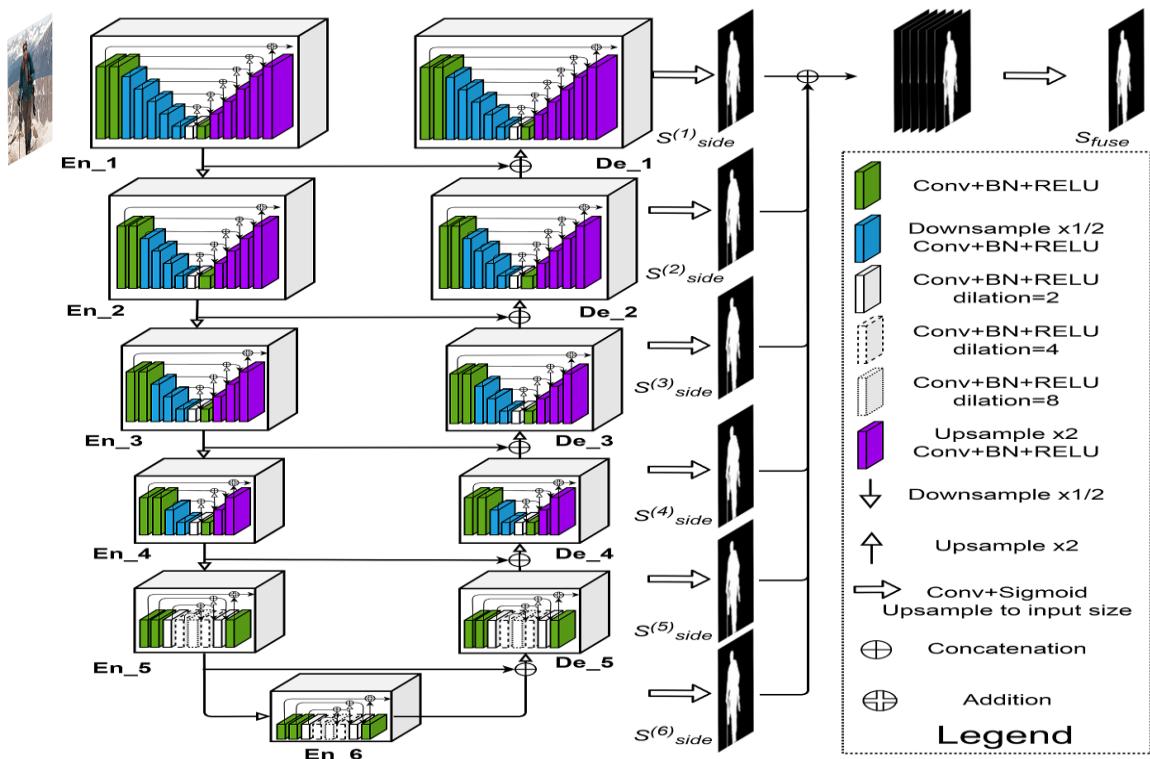


Figure 18. U-2-Net Architecture

U-2-Net is a very powerful model, it's architecture is very complex even that it's in top one between its competitors

Also, it allows us to remove the background & generate a mask of almost anything you want.



Figure 19. Human mask generation

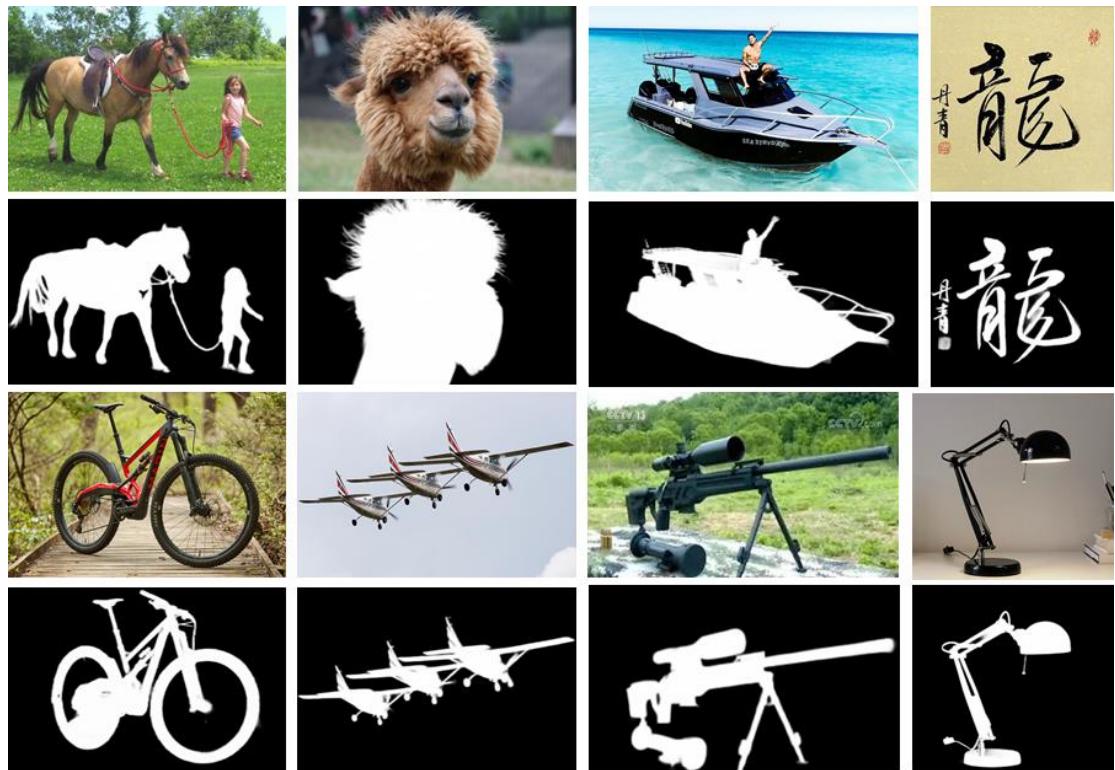


Figure 20. Object mask generation

After generating the mask for our cloth to test and run the model, unfortunately, the results were bad, below you will find the demonstration results.



Figure 21. Virtual Try-On results before the optimization

III.1.4 Optimizing the inference :

We have worked so hard to fix the problem and find solutions, after several tries and tests we have detected two main problems.

Background Problem :

If there is a background behind the person or cloth this will make a bigger issue for the model to give the results as expected, the only thing that fixes this problem is to fully remove the background and our input image transparent, So we used the U-2-Net model to solve this problem; we don't generate the mask only but both in onetime mask and no-background image.

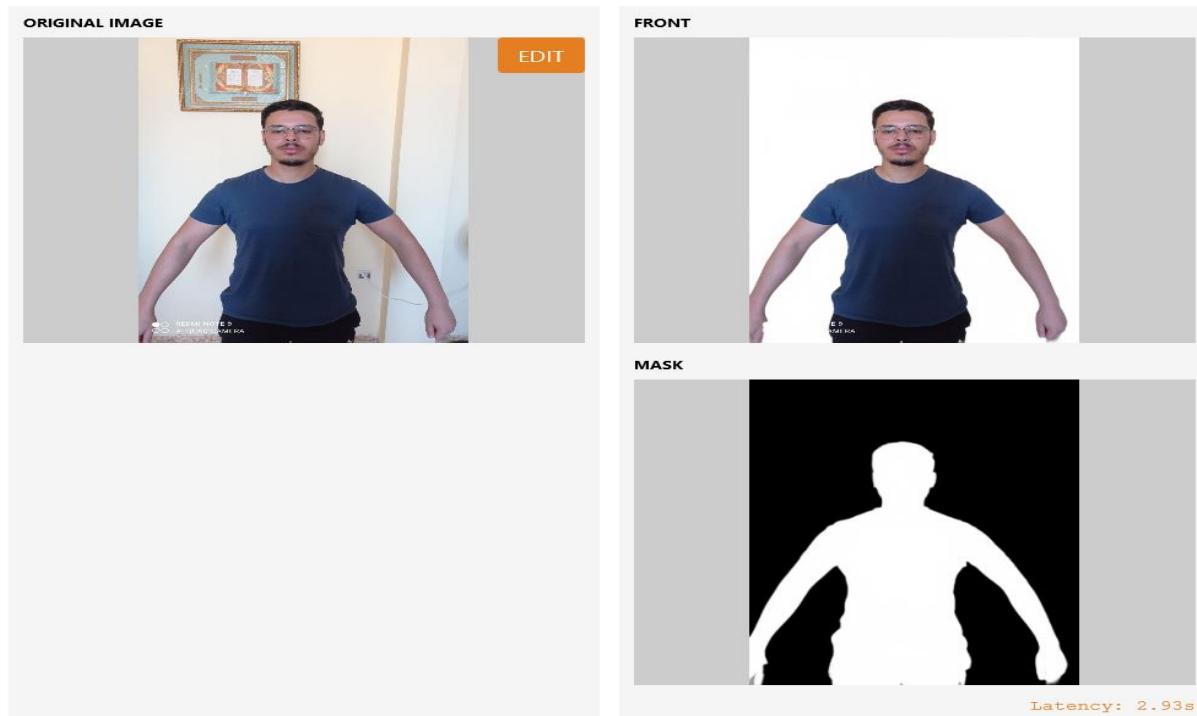


Figure 22. U-2-Net Background Removal

Margin Problem :

We have also detected that the margin on top of the head is a problem for our AI model, an example is in the figure below.

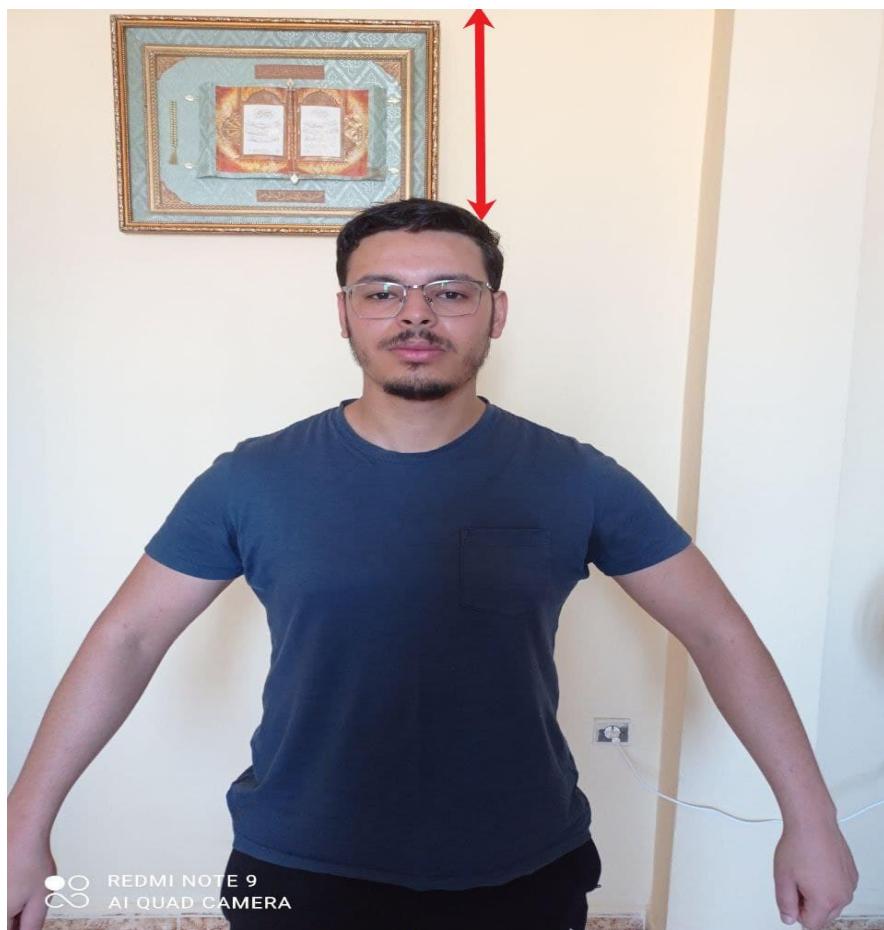


Figure 23. Margin Problem

So we have made an algorithm to recognize the face using open-cv and crop the margin on the top of the head, to crop the image we have used a very known library in python for images and it is Pillow.

```
[ ] from PIL import Image
import cv2

# Read the input image
img = cv2.imread('/content/drive/MyDrive/pfe/PF-AFN/PF-AFN_test/inputs/img/person.jpg')
# Convert into grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Load the cascade
face_cascade = cv2.CascadeClassifier('/content/drive/MyDrive/pfe/PF-AFN/PF-AFN_test/haarcascade_frontalface_alt2.xml')

# Detect faces
faces = face_cascade.detectMultiScale(gray, 1.1, 4)

# Draw rectangle around the faces and crop the faces
for (x, y, w, h) in faces:

    height_rec = h
    height_y = y

    cv2.imwrite('/content/drive/MyDrive/pfe/PF-AFN/PF-AFN test/inputs/img/person.jpg', img)
    img = cv2.imread('/content/drive/MyDrive/pfe/PF-AFN/PF-AFN_test/inputs/img/person.jpg')
    height, width, c = img.shape
    img = Image.open("/content/drive/MyDrive/pfe/PF-AFN/PF-AFN_test/inputs/img/person.jpg")
    w, h = img.size
    img.crop((0, height_rec , w, h)).save('/content/drive/MyDrive/pfe/PF-AFN/PF-AFN test/inputs/img/person.jpg')
```

Figure 24. Margin removal algorithm

Final results :

After fixing those problems we talked about, we did get better results than the first test.



Figure 25. Virtual Try-on Final results (PF-AFN)

III.2 Color Palettes Generator :

Customers in fashion always get confused to pick the best colors for the cloth according to their taste so the idea is to make an AI system that predicts the best colors palette based on the customer input which is the first color he picks for his customized t-shirt.

Now we need to choose which algorithm in artificial intelligence will respond to our needs. The idea is to predict the closest palette based on the distance between the user input and the data we have.

III.2.1 Data

Unfortunately, after a long lookup for color palettes data, we didn't find any so we decided to create our own data, we need our data to be accurate and reasonable where the palette should be consistent therefore it's a human-made palette. The best way to create this data as it was described is to grab from color websites such as colorhunt.co or color-hex.com, we've picked color-hex.com.

Data Collecting :

We need to create a script to collect those data for us, parse it, and save it into a CSV file and this is what we call web scraping with permission to scrape. Color-Hex has a big list of data which is 76720 color palettes, to collect all this data will be time-consuming so we need to speed up this process and the solution is multi-threading, we create 100 threads each it means that we will scrape 100 pages at the same time. Let's assume that each page to scrape will take almost one second which means to scrape all pages without multithreading will take approximately **21 hours** but using threads this time will be divided into a hundred so it will consume only **13 minutes**.

Data Scraping :

In order to scrape data we need a programming language, we will be using Python because it is fast and has several libraries that will help us process what we need. About the libraries, we will send **requests** to communicate with the website and **beautifulsoup4** as known as **bs4**, and finally **threading** to spawn our threads.

We have two main functions **extract_palette** take URL palette as a parameter, its purpose is to extract and parse the palette from the URL given; the second function is **extract_urls** and it takes page number to extract as a parameter, the idea of that function is to extract all palettes links for a specific page that holds many palettes URLs and to pass each URL to **extract_palette** function. Now we just need to loop through how many pages we want to scrape and call **extract_urls** which will eventually call **extract_palette** and saves the result to a file called **dataset2021-colorshex.csv** for HEX format and **dataset2021-colorsrgb.csv** for RGB format.

```

def extract_urls(page):
    links = []
    r = requests.get('https://www.color-hex.com/color-palettes/?page={}'.format(str(page)))
    soup = BeautifulSoup(r.text, 'lxml')
    data = soup.find_all('div', attrs={'class':'palettecontainerlist'})
    for d in data:
        if 'https://www.color-hex.com'+d.find('a').get('href') not in links:
            links.append('https://www.color-hex.com'+d.find('a').get('href'))
    for link in links:
        extract_palette(link)

    with open('log.txt','a') as file:
        file.write(str(page)+'\n')

```

Figure 26. extract_urls Function content

```

def extract_palette(url):
    global busy
    data = []
    hexa = []
    rgb = []
    r = requests.get(url)
    soup = BeautifulSoup(r.text, 'lxml')
    table = soup.find_all('table')
    try:
        for row in table:
            cols = row.find_all('td')
            cols = [ele.text.strip().replace('(', '').replace(')', '').replace(',', '-') for ele in cols]
            data.append([ele for ele in cols if ele])
        for i in range(10):
            if i % 2 == 0:
                hexa.append(data[0][i])
            else:
                rgb.append(data[0][i])
        while busy:
            time.sleep(0.1)
        busy = True
        with open('dataset2021-colorshex.csv', 'a+') as file:
            file.write(','.join(hexa)+'\n')

        with open('dataset2021-colorsrgb.csv', 'a+') as file:
            file.write(','.join(rgb)+'\n')

        busy = False
        print(','.join(hexa))
        print(','.join(rgb))
    except:
        pass

```

Figure 27. extract_palette Function content

Data Parsing & Filtering :

After collecting the data, we have to filter it and clean incompatible palettes or the non-consistent one, to detect inconsistencies is when we found the same color for each color in one palette is duplicated, for example, if white color is found twice in that palette we will remove it. We will use **Counter** from the **collections** library.

```

from collections import Counter

with open('../colors datacollect/dataset2021-colorshex.csv', 'r') as file:
    data = file.read().splitlines()

with open('../colors datacollect/dataset2021-colorshex-filtred-2.csv', 'a') as file:
    for d in data:
        dlist = d.split(',')
        delete = False
        counts_colors = dict(Counter(dlist))
        for color in counts_colors:
            if counts_colors[color] > 1:
                delete = True
                break
        if not delete:
            file.write(d+'\n')

```

Figure 28. Data filtering algorithm

```

69461 #597f80,#597f80,#597f80,#597f80,#597f80
69462 #aaed94,#557e55,#5f5fa4,#ff9a9a,#6a5252
69463 #bc4b4b,#140e00,#0000ff,#eeeeee,#000000
69464 #885159,#645188,#886451,#528881,#000000
69465 #295f48,#204c39,#18392b,#eeeeee,#000000
69466 #52bf90,#49ab81,#419873,#398564,#317256
69467 #8a7aa3,#83789e,#a69eb0,#877a96,#807d9e
69468 #a69eb0,#f2e2cb,#a6b57c,#effeff2,#7f9124
69469 #a69eb0,#effeff2,#704b5b,#f2e2cb,#945158
69470 #a69eb0,#effeff2,#f2e2cd,#dadae3,#000000
69471 #325da7,#f68c06,#f7f7f7,#ffffff,#181818
69472 #273609,#679317,#a6cd1b,#d9e3c5,#ffffff
69473 #0073e5,#7ddc1f,#f5f5f5,#444444,#000000
69474 #6c4e4e,#797d79,#626271,#593636,#c63636
69475 #77aaff,#99ccff,#bbeeff,#5588ff,#3366ff
69476 #aaff77,#ccff99,#efffb8,#88ff55,#66ff33
69477 #ff77aa,#ff99cc,#ffbbbe,#ff5588,#ff3377
69478 #ff77aa,#aaff77,#77aaff,#ffffff,#000000
69479 #3b5998,#8b9dc3,#dfe3ee,#f7f7f7,#ffffff
69480 #999999,#777777,#555555,#333333,#111111
69481 #55a8a3,#222222,#272e34,#f4f3f3,#aad6cc
69482 #c6c386,#c3eee7,#5b391e,#87c6c3,#66bbae
69483 #c6c386,#7df9ff,#0095b6,#0892d0,#000000
69484 #FF66cc,#ccff66,#66ccff,#ffffff,#000000
69485 #f1201f,#ff66cc,#0095b6,#0892d0,#b75a50

```

Figure 29. Data collected and filtered results

III.2.2 Training & Testing :

We didn't pick our algorithm yet, so we need the algorithm to fit our needs. An algorithm that uses the distance to train and predict and it's supervised then we will be using KNN (k-Nearest-Neighbors). The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

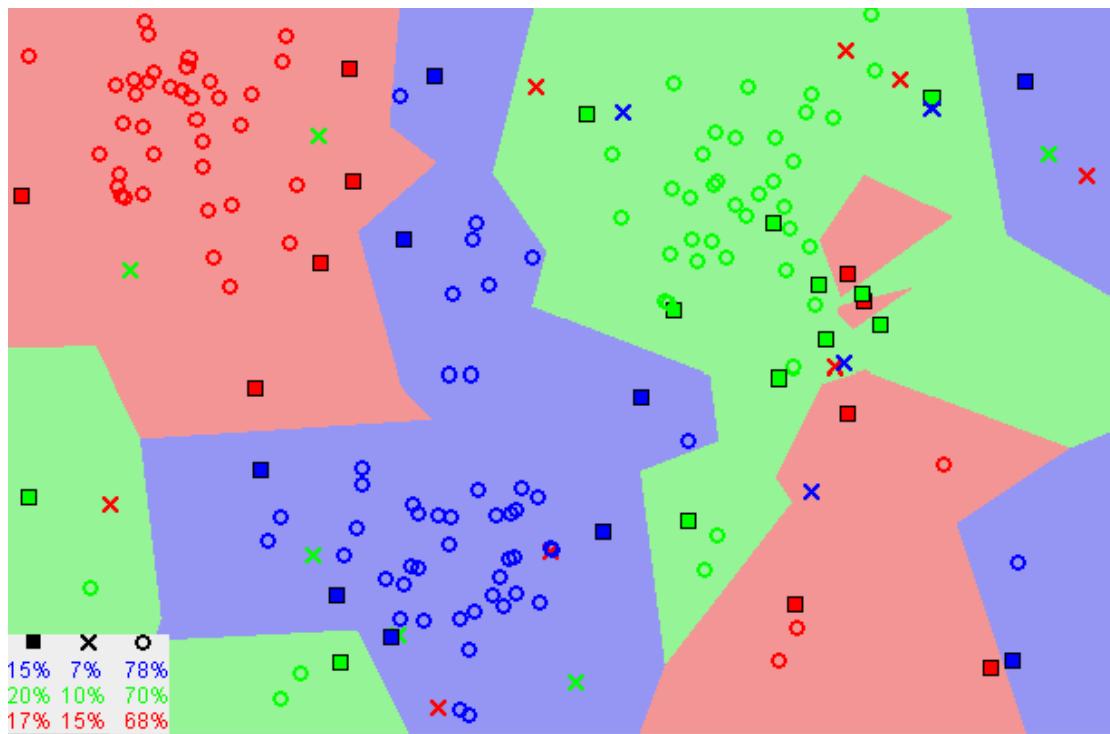


Figure 30. Similar data points typically exist close to each other

Notice in the image above that most of the time, similar data points are close to each other. The KNN algorithm hinges on this assumption being true enough for the algorithm to be useful. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the distance between points on a graph.

KNN Steps

1. Load the data
2. Initialize K to your chosen number of neighbors
3. For each example in the data
 - a. Calculate the distance between the query example and the current example from the data.
 - b. Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels
8. If classification, return the mode of the K labels

However we will not use the original KNN but we will tweak it, why tweaked? because the KNN usually predicts a class from our dataset based on features and distance between the input and each data in our dataset, but what we need is not the class because we don't have one, we need the full row of data which is our target palette. we will be using the **euclidean distance** to calculate the distance. The user input is a hex format and our dataset

is in hex format it will be hard if we keep the same format and try to calculate the distance so we convert it to RGB (Red Green Blue) format, for example, **#9C5BA9** will be a tuple **(156,91,169)** now it's easier.

```
def get_palette(data_distance, hexa):
    temp_colors_filtered = []
    rgb = (tuple(int(hexa[i:i+2], 16) for i in (0, 2, 4)))
    for palette in data_distance:
        colors = palette[0].split(',')
        if '#' + hexa in colors:
            temp_colors_filtered.append(palette)
    if len(temp_colors_filtered) == 0:
        temp_colors_filtered = data_distance

    for palette in temp_colors_filtered:
        colors = palette[0].split(',')

        #converting to RGB
        distances_each_rgb = []
        for i in range(len(colors)):
            colors[i] = (tuple(int(colors[i].lstrip('#')[j:j+2], 16) for j in (0, 2, 4)))
            #calculating distance for each color in palette and input color
            distances_each_rgb.append(sqrt((colors[i][0] - rgb[0])**2 + (colors[i][1] - rgb[1])**2 + (colors[i][2] - rgb[2])**2))
        palette[1] = sum(distances_each_rgb)

    #Calculating median:
    distances = [i[1] for i in temp_colors_filtered]
    mean_distance = np.median(distances)
    all_distances_to_mean = []
    for data in temp_colors_filtered:
        all_distances_to_mean.append([data[0],sqrt((data[1] - mean_distance)**2)])
    colors_palette = min(all_distances_to_mean, key=lambda x: x[1])
    colors_palette.pop()
    colors_palette = colors_palette[0].split(',')
    return colors_palette
```

Figure 31. Tweaked KNN algorithm for palette prediction (n=1)

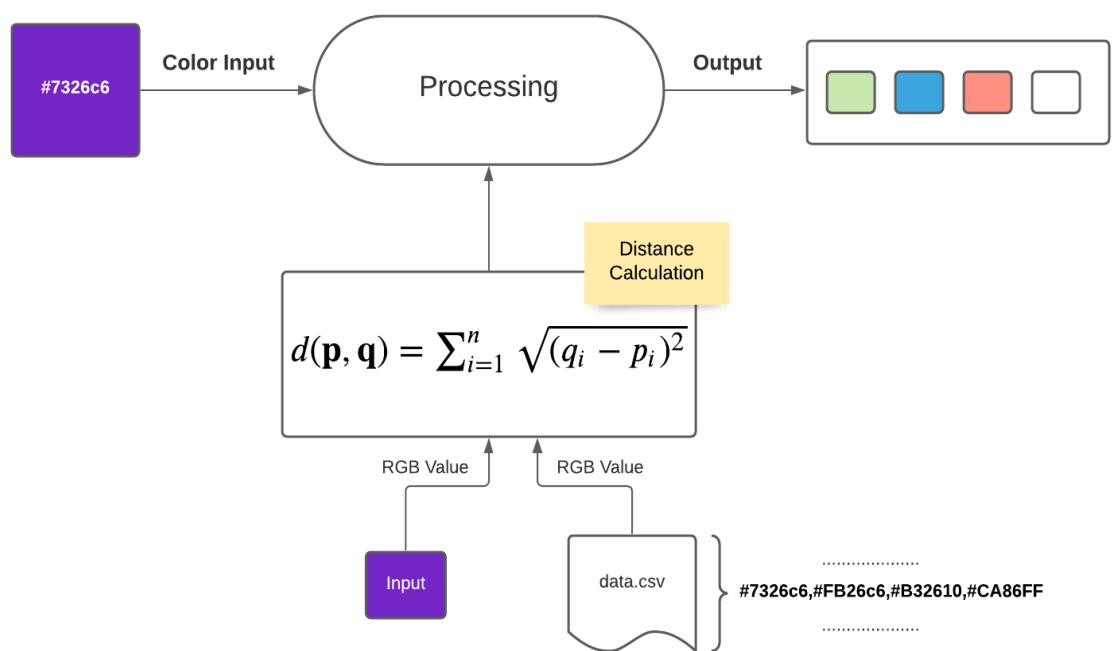


Figure 32. Colors Palette Generator Structure

III.2.3 AI Palettes :

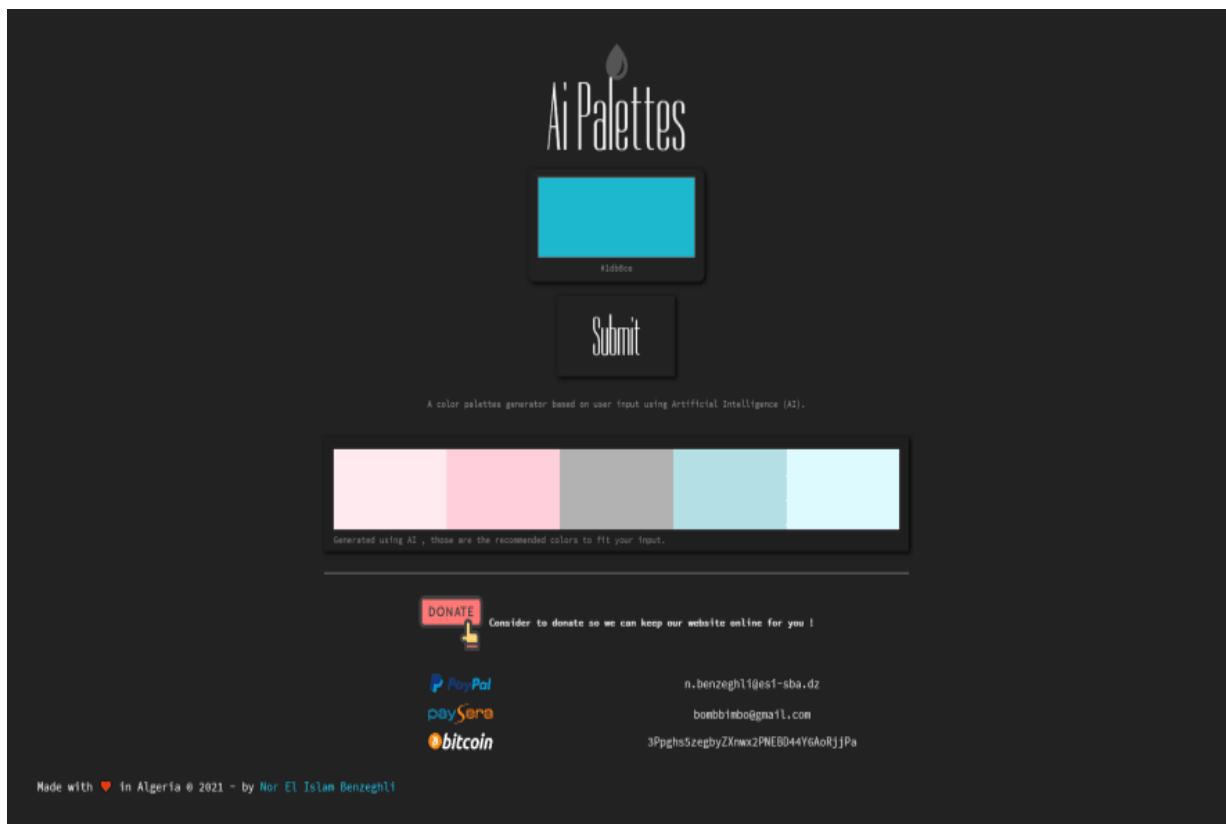


Figure 33. aipalettes.com

While our color palettes prediction algorithm is unique we had an idea to make it public and can be used by everyone. AI Palettes, an online website where you can choose one color to generate the best-fit colors palette that is compatible with your input, is A recommendation system based on Artificial Intelligence. You can copy each color code by clicking on it. We made it for free use to everyone! If you are a Designer / Painter / Art Creator this will help you a lot. it can be accessed by this link <https://aipalettes.com>.

III.3 T-shirt Size Recommender :



Figure 34. T-shirt Size Recommendation Structure

We want to recommend to the customer the best-fit t-shirt size using only his height and weight. It's all like the other AIs we did present, we need data, we did found some websites that provide data for T-shirt size, unfortunately, it didn't include height and weight also it's for only one gender "female" therefore we need to collect or create our own data.

III.3.1 Data :

Our data needs to be accurate and based on a unisex basis so our recommendation will fit both genders female and male, We've done some researches and trying to find the best size guide (Europe size guide) and we did find one.

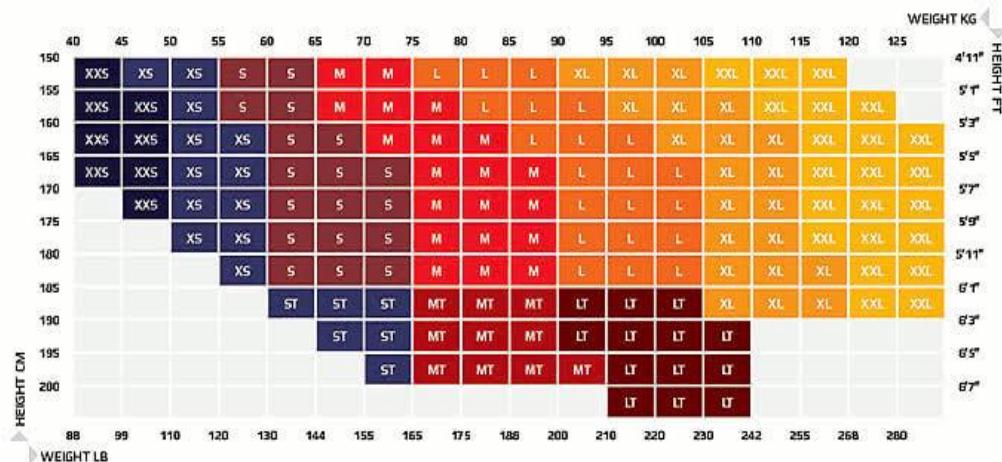


Figure 35. T-shirt size guide based on height and weight

Now we only have to create a dataset based on that table, we made a python script that will do it for us.

9426	109, 204, L
9427	110, 204, L
9428	111, 204, L
9429	112, 204, L
9430	113, 204, L
9431	114, 204, L
9432	115, 204, L
9433	116, 204, XL
9434	117, 204, XL
9435	118, 204, XL
9436	119, 204, XL
9437	120, 204, XL
9438	121, 204, 2XL
9439	122, 204, 2XL
9440	123, 204, 2XL
9441	124, 204, 2XL
9442	125, 204, 2XL
9443	126, 204, 2XL
9444	127, 204, 2XL
9445	128, 204, 2XL
9446	129, 204, 2XL
9447	130, 204, 2XL
9448	131, 204, 2XL
9449	132, 204, 2XL
9500	133, 204, 2XL
9501	134, 204, 2XL

Figure 36. T-shirt size recommendation dataset sample

III.3.2 Training & Testing :

Now we need to train our model and pick the most accurate and fast algorithm, consequently, we will try several models, happily, there is a new method to make this happen and it's called **Lazypredict**, it will train and test almost all famous algorithms for us and bring us back the results detailed.

```
1 import pyforest
2 import lazypredict
3 from lazypredict.Supervised import LazyClassifier
4 from sklearn.model_selection import train_test_split
5 import pandas as pd
6
7 train = pd.read_csv('new_dataset2.csv')
8 train['tshirt_size'] = pd.factorize(train['tshirt_size'])[0] + 1
9 #test['tshirt_size'] = pd.factorize(test['tshirt_size'])[0] + 1
10
11 X = train.drop(['tshirt_size'], axis=1)
12 y = train.tshirt_size
13 X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
14 clf = LazyClassifier(verbose=0, ignore_warnings=True)
15 models, predictions = clf.fit(X_train, X_test, y_train, y_test)
16 print(models)
17 |
```

Figure 37. Lazypredict Classifier

Results are stunning, not for all the algorithms but the ones that we want.

Model	Accuracy	Time Taken
XGBClassifier	1.00	2.54
DecisionTreeClassifier	1.00	0.09
BaggingClassifier	1.00	0.26
QuadraticDiscriminantAnalysis	0.97	0.53
GaussianNB	0.95	0.05
ExtraTreesClassifier	1.00	2.08
ExtraTreeClassifier	1.00	0.05
LinearDiscriminantAnalysis	0.94	0.34
SVC	1.00	9.17
KNeighborsClassifier	0.99	0.65
RandomForestClassifier	1.00	1.93
NearestCentroid	0.76	0.07
LogisticRegression	0.96	3.43
LGBMClassifier	0.87	1.03
SGDClassifier	0.78	0.38
AdaBoostClassifier	0.70	1.64
CalibratedClassifierCV	0.77	51.37
Perceptron	0.61	0.27
LinearSVC	0.77	16.47
PassiveAggressiveClassifier	0.63	0.26
BernoulliNB	0.55	0.11
RidgeClassifier	0.58	0.08
RidgeClassifierCV	0.58	0.12
DummyClassifier	0.32	0.06

[24 rows x 5 columns]
root@pop-os:~/Documents/lazypredict\$]

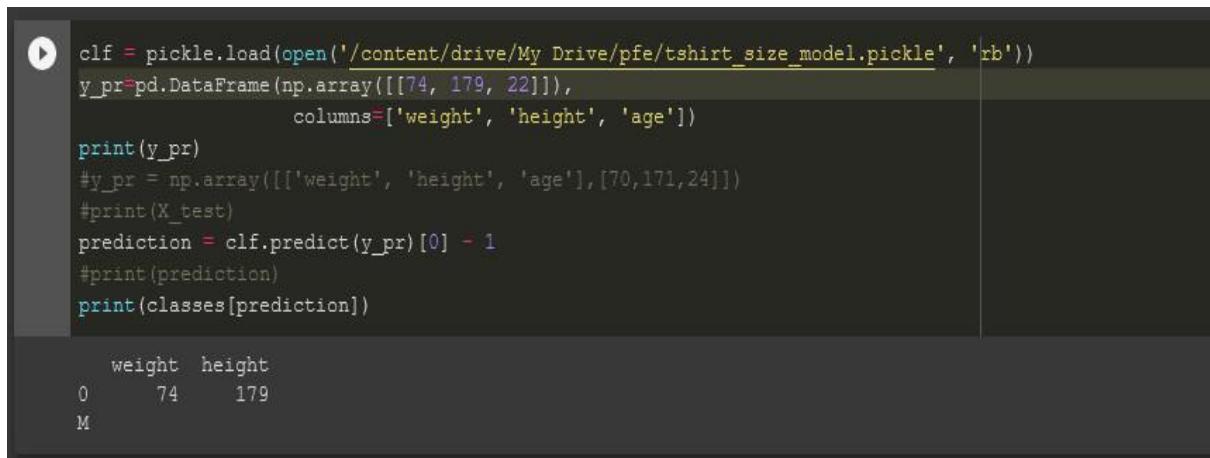
Figure 38. Lazypredict Classifier Results

Time taken by some of the models we trained and tested :

Model	Accuracy (f-measure)	Time Taken (second)
XGBClassifier	100%	2.54
DecisionTreeClassifier	100%	0.09
BaggingClassifier	100%	0.26
SVC	100%	9.17
RandomForestClassifier	100%	1.93

Concluding from the results of the `lazypredict` classifier we can clearly see that **DecisionTreeClassifier** is the best algorithm according to its accuracy **of 100%** and speed which has the least time taken **0.09 sec.**

Now we would like to test an input height, weight and see what is the prediction of course after compiling our model into a pickle file.



```
clf = pickle.load(open('/content/drive/My Drive/pfe/tshirt_size_model.pickle', 'rb'))
y_pr=pd.DataFrame(np.array([[74, 179, 22]]),
                  columns=['weight', 'height', 'age'])
print(y_pr)
#y_pr = np.array([['weight', 'height', 'age'], [70,171,24]])
#print(X_test)
prediction = clf.predict(y_pr)[0] - 1
#print(prediction)
print(classes[prediction])

      weight  height
0       74     179
M
```

Figure 39. T-shirt Size Recommendation Results

Our input is 74kgs weight and 179cm height the prediction class is “M” which is logical and our model is working perfectly.

Chapter IV

IV. Web Platform

In this chapter, we will detail how we will build our web platform used the technologies mentioned above.

IV.1 Project creation :

While we are using django, project creation is the first thing that we should do to start our project, in order to create the project django provides us a set of tools for the administration, and the module which hold those tools is called **django-admin**, it's a module where it accepts command line arguments. To create it, from the command line, cd into a directory where you'd like to store your code, then run the following command: **django-admin startproject jilshirt** , now that the project has been created, we will create applications inside our project but first what is a django application? A Django *application* is a Python package that is specifically intended for use in a Django project. An application may use common Django conventions, such as having models, tests, urls, and views submodules. We will have two main applications, one is for the user/customer we call it "**home**" and the other is for the administrator we call it "**dashboard**" where the administrator can manage his website and all of this can be achieved by the command **django-admin startapp <appname>** .

IV.2 Project Structure :

After creating the project, we create two other folders inside our project folder "**model**" to put our AI models and "**data**" to put our colors dataset.

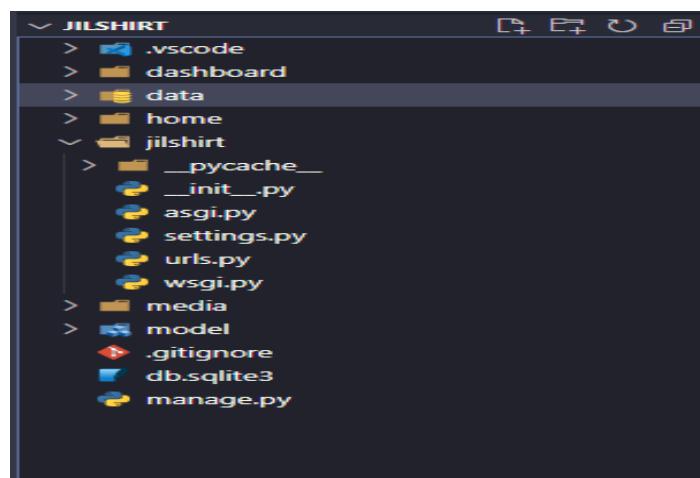


Figure 40. Project Structure

IV.3 Project Files :

- Settings.py: Django configurations.
- Urls.py: All routing urls must be defined in that file.
- Wsgi.py: An entry point for WSGI-compatible web servers to serve

- the project.
- Admin.py: Register models to be displayed on the Django admin panel.
- Apps.py: Contains the application configuration code
- Models.py: Define my models here. These are the tables used to store data and define their relationships.
- Forms.py: It does contain all HTML forms to be rendered.
- Tests.py: Writing tests for the application

IV.4 APIs:

We need to create APIs to make requests easier and our website real-time, here is the list of APIs that we should create :

- Get products

This API will return to us all products that exist in our database serialized in a JSON format.

- Get products based on cookies

Same as the first API but filtered by cookies which means only the products added to the cart by the customer.

- Check stock of a product

This API will check the product if it's in stock or not before adding it to the user shopping cart.

IV.5 Web Design :

IV.5.1 User Front :

The screenshot displays the homepage of the Jilshirt website, featuring a light brown header bar with the brand name 'Jilshirt' in a stylized script font. Below the header, a large banner features the slogan 'EMPOWER FASHION WITH DESIGNER AI' in bold, black, uppercase letters, accompanied by a red and white abstract graphic of a person's head and shoulders. The main navigation menu includes links for 'LONG SLEEVED SHIRTS', 'TSHIRTS', 'HOODIES', 'MHHM', 'TEE', 'MEN', 'Search' (with a magnifying glass icon), 'Cart' (with a shopping cart icon), and 'Log In'. A teal-colored sidebar on the left contains icons for 'AI', 'Payment', and 'Delivery'. The central content area is divided into several sections: 'Why Jilshirt?' (with a list of three bullet points: '1. Design your own product, use our artificial intelligence to predict your best-fit size', '2. Colors palette recommendation system based on your taste', and '3. Try it on virtually to see how it looks!'), 'Design your own clothes' (with a preview of a t-shirt on a computer screen), a 'GET STARTED' button, 'Latest Products' (showing three items: 'Long Sleeved Shirts product 1' (200 DZD), 'Long Sleeved Shirts product 2' (300 DZD), and 'T shirts product 3' (400 DZD)), and 'Shop by category' (displaying four categories: men's t-shirts, hoodies, women's long-sleeved shirts, and a 'Your design here' section). The footer contains links for 'Our Stores', 'Blog Posts', 'Support', and various site policies.

Our Stores

9870 St Vincent Place,
Glasgow, G14 4Ls.
Telephone: +1 800 545 2470
Fax: +1 800 658 5784
E-mail: Info@jilshop.com

Blog Posts

Sed ut perspiciatis unde omnis
The standard chunk of Lorem Ipsum used since the 1500s is
reproduced.

Sed ut perspiciatis unde omnis
The standard chunk of Lorem Ipsum used since the 1500s is
reproduced.

Sed ut perspiciatis unde omnis
The standard chunk of Lorem Ipsum used since the 1500s is
reproduced.

Support

Terms & Conditions
FAQ
Payment
Refunds
Track Order
Services

Press
Blog
About Us
Contact Us

Figure 41. Home Page



Figure 42. Product Page

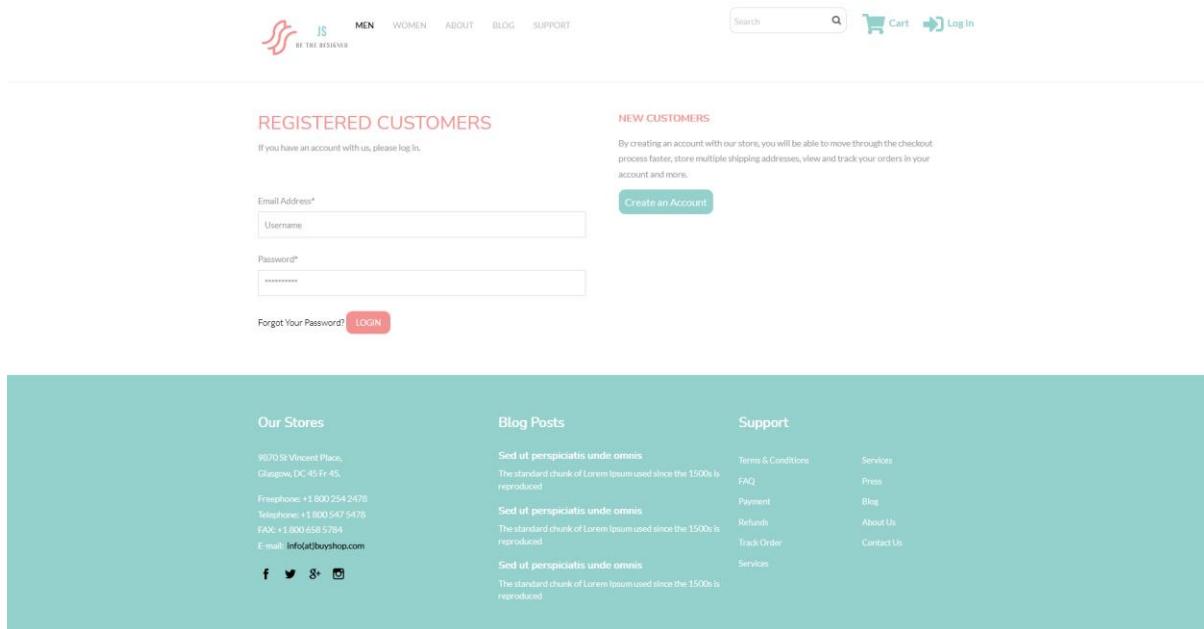


Figure 43. Login Page

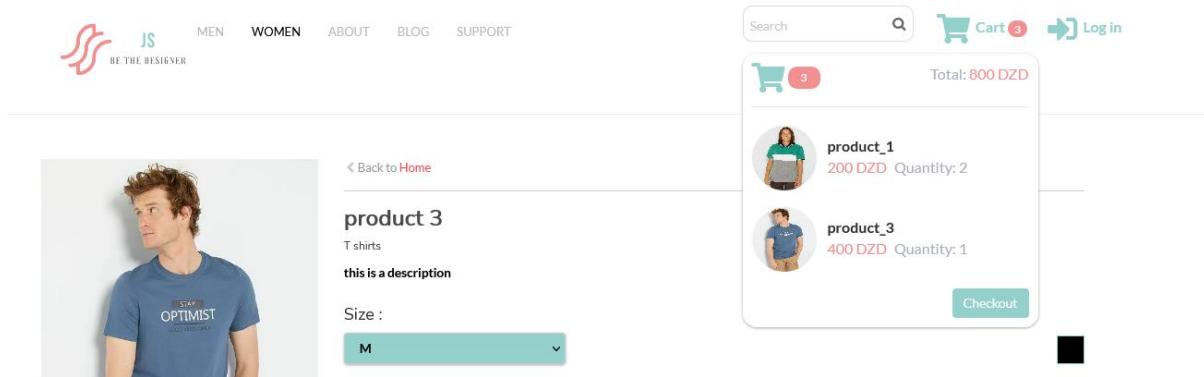


Figure 44. Shopping Cart

Price Details	
Total	0.00
Discount	—
Delivery Charges	150.00
TOTAL 800 DZD	
Place Order	

Figure 45. Checkout Page

Customize T-Shirt

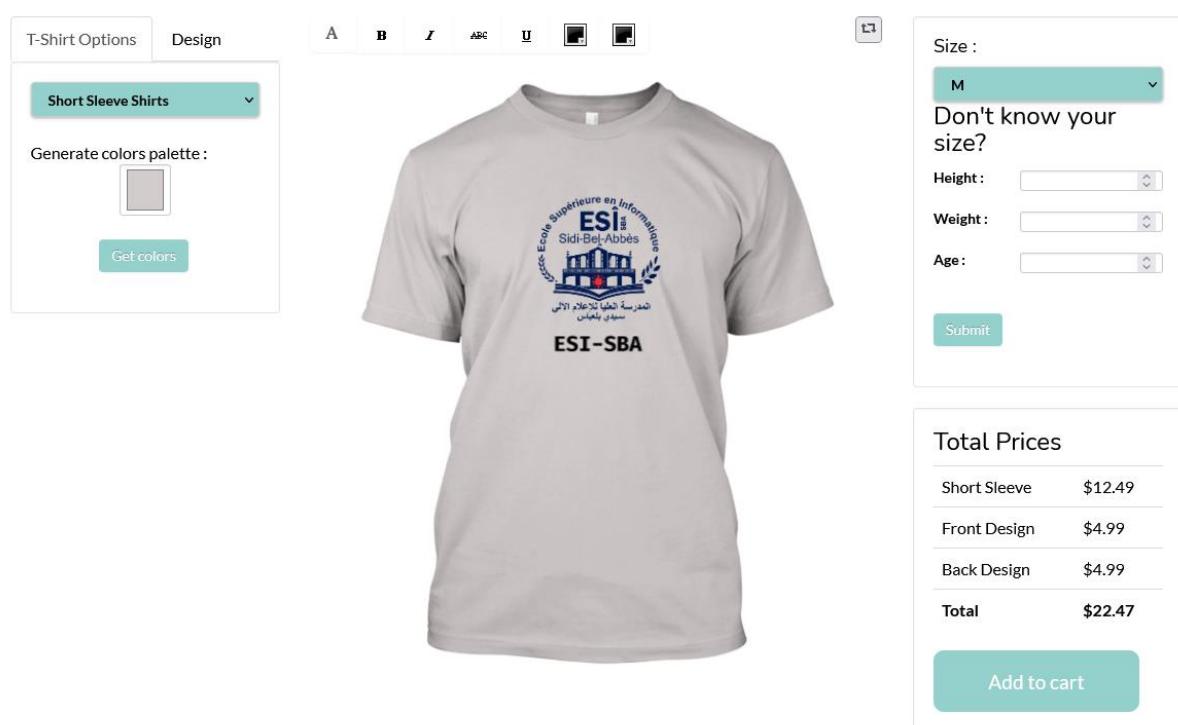


Figure 46. Customize T-Shirt Page

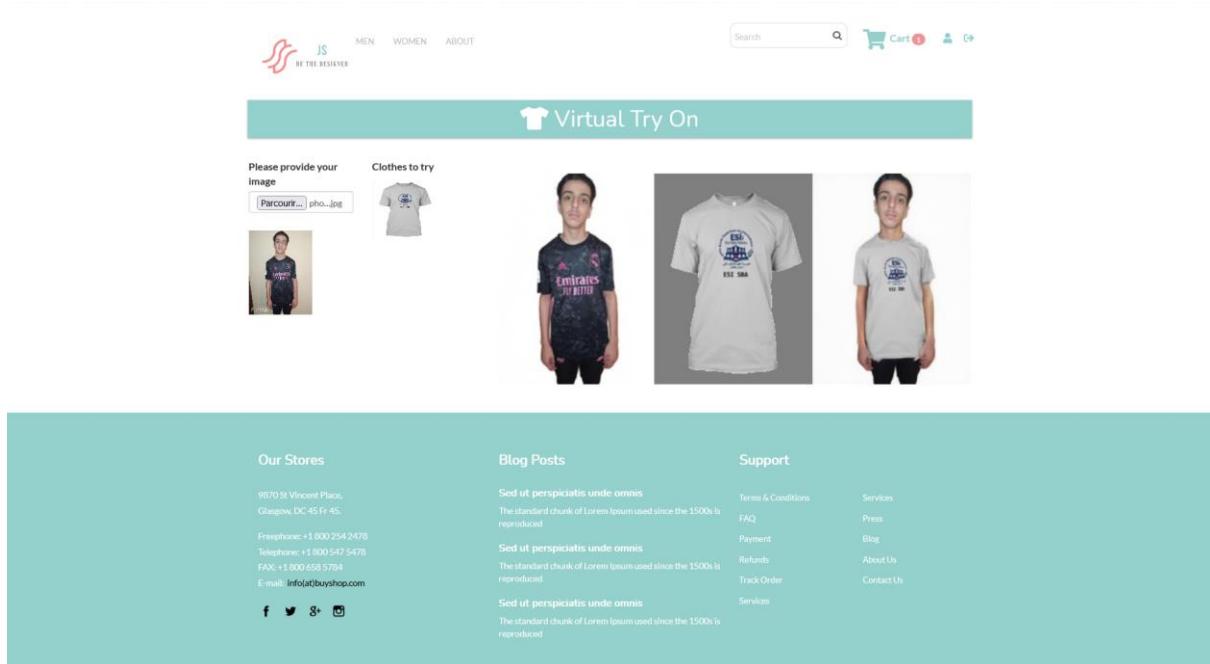


Figure 47. Virtual Try-on Page

IV.5.2 Admin Front :

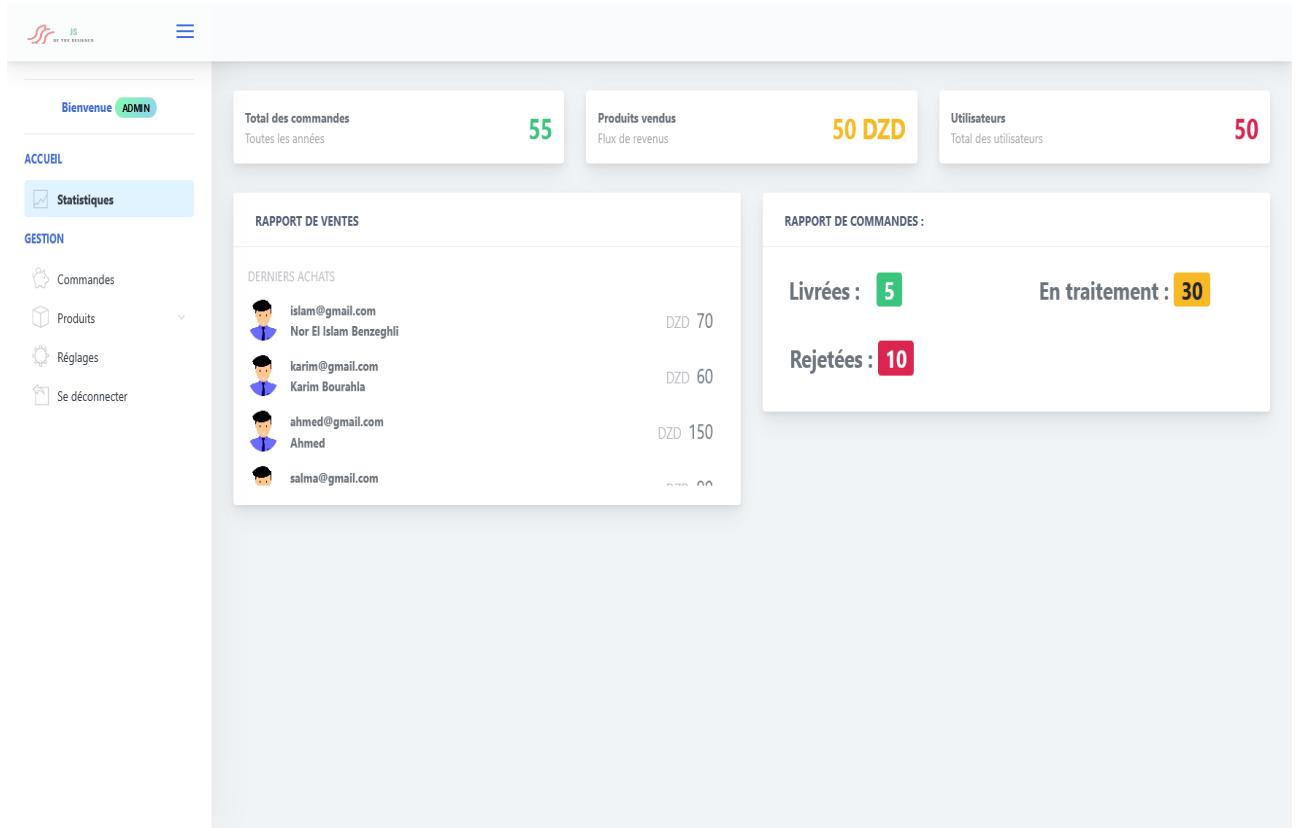


Figure 48. Statistics Page

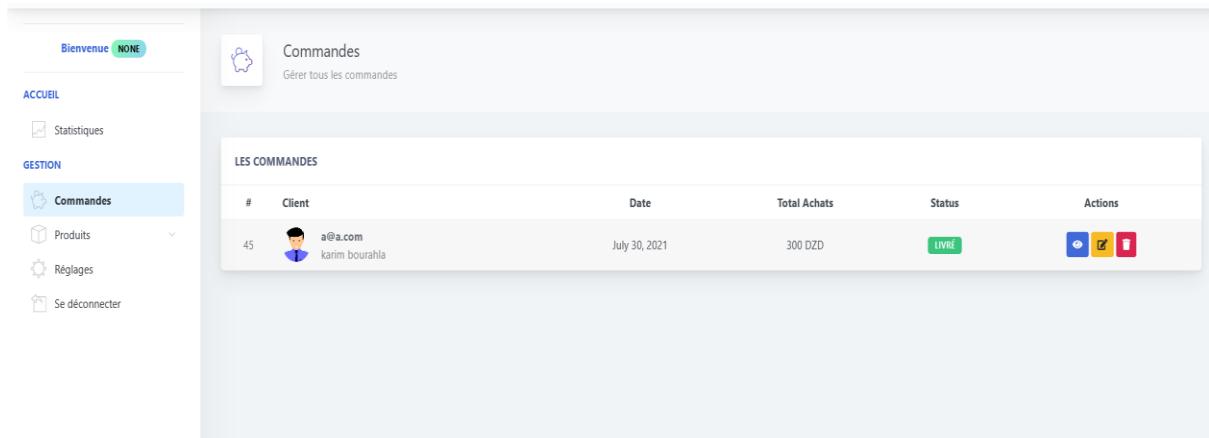


Figure 49. Orders Page

The screenshot shows a web-based administration interface for managing products. The top navigation bar includes a logo, a "Bienvenue" message, and an "ADMIN" button. On the left, a sidebar menu lists "ACCUEIL", "Statistiques", "GESTION" (selected), "Commandes", "Produits" (selected), "Règlages", and "Se déconnecter". The main content area is titled "Produits" and "Gérer tous mes produits". A sub-section titled "LES PRODUITS" displays a table with columns "Nom", "Prix", and "Actions". The table contains five rows of product data:

Nom	Prix	Actions
product 1 Long Sleeved Shirts	200 DZD	[Edit, Delete, Details]
product 2 Long Sleeved Shirts	300 DZD	[Edit, Delete, Details]
product 3 T shirts	400 DZD	[Edit, Delete, Details]
Customized-Product Customized Tshirts	50 DZD	[Edit, Delete, Details]
Customized-Product Customized Tshirts	50 DZD	[Edit, Delete, Details]

A small navigation bar at the bottom indicates pages 1, 2, and >.

Figure 50. Products Management Page

The screenshot shows a web-based administration interface for managing stock. The layout is similar to Figure 50, with a sidebar menu and a main content area titled "Produits" and "Gérer tous mes produits". A sub-section titled "LES STOCKS" displays a table with columns "Nom", "Color", "Size", and "Stock". The table contains three rows of product data:

Nom	Color	Size	Stock
product 1	[Color swatches: pink, blue]		[Edit, Delete, Details]
product 2	[Color swatches: black, red]		[Edit, Delete, Details]
product 3	[Color swatch: black]		[Edit, Delete, Details]

Figure 51. Stock Management Page

IV.6 Artificial Intelligence Linking :

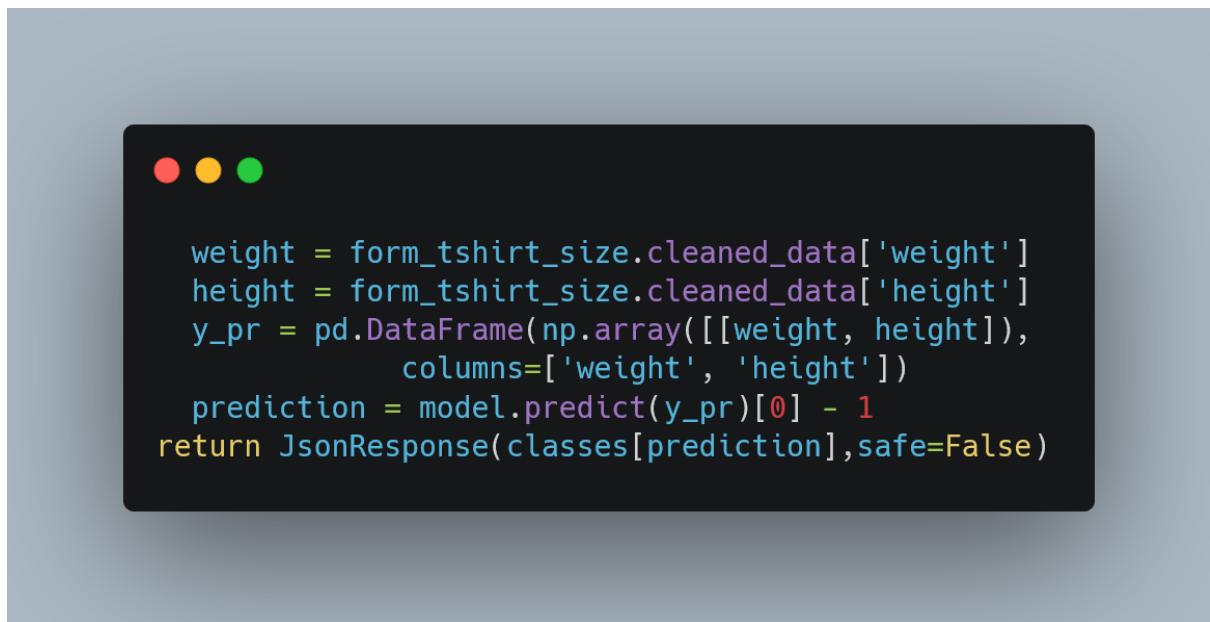
IV.6.1 T-shirt Size Recommendation :

Implementing the t-shirt size recommendation model isn't that hard because it's compiled into a pickle file (compressed) we just need to load the pickle file and interact with it and we return the results to our web page.



```
#model load
model = pickle.load(open('model/tshirt_size_model.pickle', 'rb'))
classes = ['S', 'M', 'L', 'XL', '2XL', '3XL']
SIZES = {
    'S':1, 'M':2, 'L':3, 'XL':4, '2XL': 5
}
```

Loading the model.

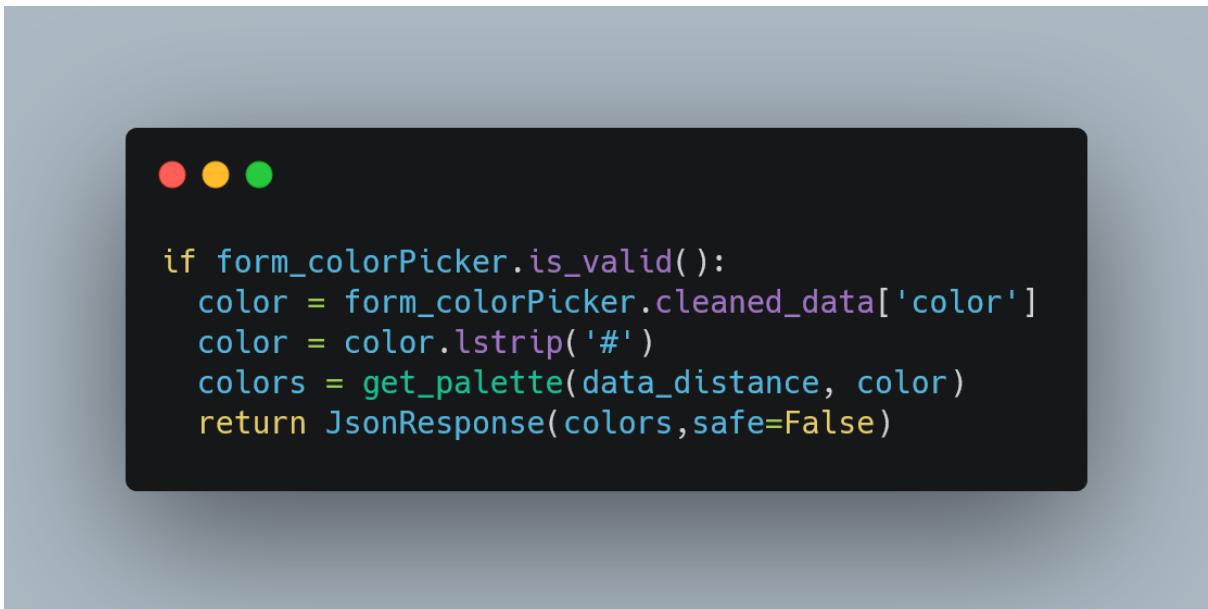


```
weight = form_tshirt_size.cleaned_data['weight']
height = form_tshirt_size.cleaned_data['height']
y_pr = pd.DataFrame(np.array([[weight, height]]),
                     columns=['weight', 'height'])
prediction = model.predict(y_pr)[0] - 1
return JsonResponse(classes[prediction], safe=False)
```

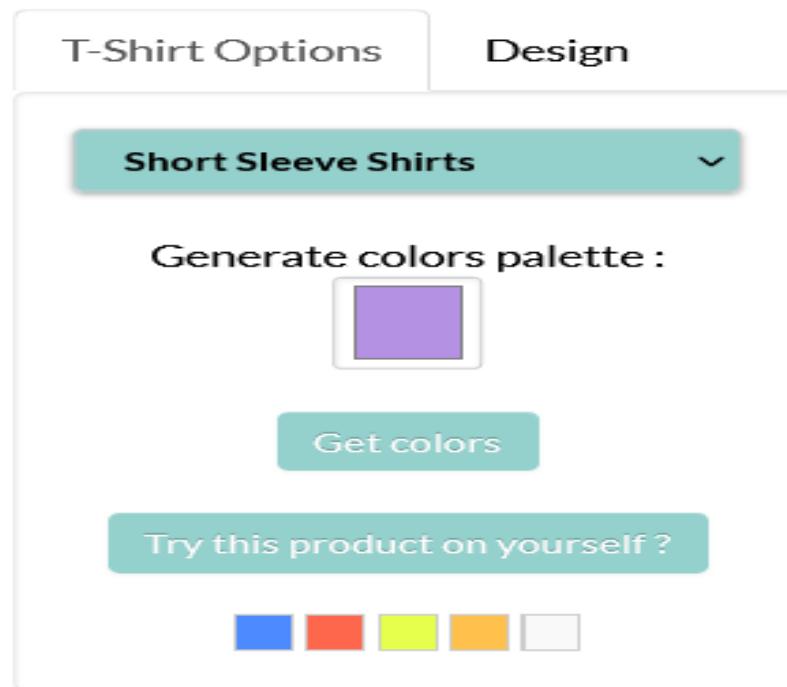
Returning Model Prediction as a response.

IV.6.2 Color Palettes Generator :

In this algorithm, we will be directly calling our function get_palette and return the response.



Colors Palette Generator as a response.



Rendering the response on the template.

IV.6.3 Virtual Try-on :

For virtual try-on algorithm, it needs to be passed by 5 different steps which are:

Crop the image for the margin :

```
crop_image()

filename = f"home/tryon/inference/inputs/img/person_{token}.jpg"
# load image from file
pixels = pyplot.imread(
    f"home/tryon/inference/inputs/img/person_{token}.jpg")
# create the detector, using default weights
detector = MTCNN()
faces = detector.detect_faces(pixels)

for result in faces:
    # get coordinates
    x, y, width, height = result['box']
    img = cv2.imread(filename)
    # Convert into grayscale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    height_rec = y
    img = Image.open(filename)
    w, h = img.size
    img.crop((0, height_rec-50, w, h)).save(f"home/tryon/inference/inputs/img/person_{token}.jpg")
```

Detect Face & Cropping image algorithm.

As we can see we have used **MTCNN** instead of **OpenCV** since it's deprecated and not too accurate to detect the face and crop the image using **Pillow** and save the results.

Remove Background :

```
remove_background(token):
    infer(path + f"/home/tryon/inference/inputs/img/person_{token}.jpg")
    os.rename(os.getcwd() + '/home/tryon/inference/inputs/img/result_0.png',
              os.getcwd() + f"/home/tryon/inference/inputs/img/result_0_{token}.png")

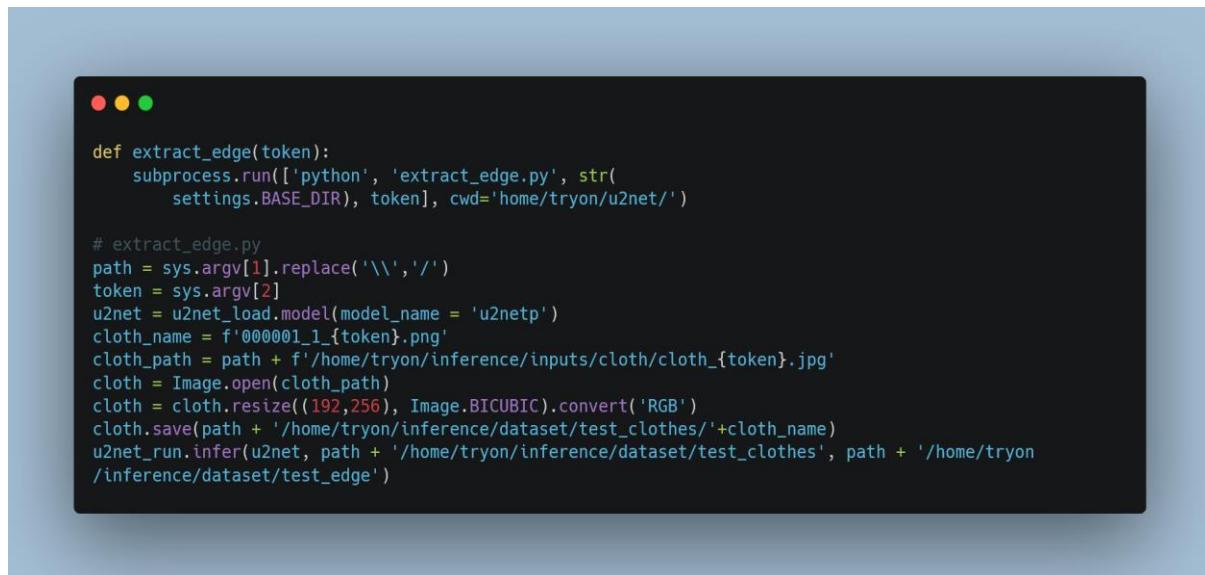
def infer(img):
    result = model.Segmentation(images=[
        cv2.imread(img)],
        paths=None,
        batch_size=1,
        input_size=320,
        output_dir=(path + '/home/tryon/inference/inputs/img/'),
        visualization=True)
    return (result[0]['front'][ :, :, ::-1], result[0]['mask'])
```

Remove background algorithm.

In this algorithm we used U2-Net as we mentioned above, removing the background and saving the results for the next step.

Extract Edge :

We need to extract the edge of the t-shirt so it can be detected.



```
def extract_edge(token):
    subprocess.run(['python', 'extract_edge.py', str(
        settings.BASE_DIR), token], cwd='home/tryon/u2net/')

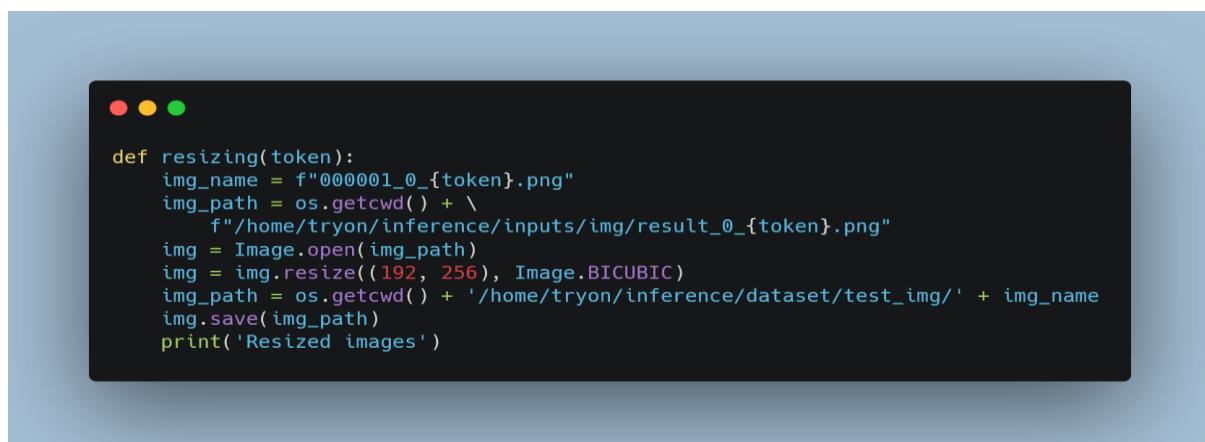
# extract_edge.py
path = sys.argv[1].replace('\\', '/')
token = sys.argv[2]
u2net = u2net_load.model(model_name = 'u2netp')
cloth_name = f'000001_1_{token}.png'
cloth_path = path + f'/home/tryon/inference/inputs/cloth/cloth_{token}.jpg'
cloth = Image.open(cloth_path)
cloth = cloth.resize((192,256), Image.BICUBIC).convert('RGB')
cloth.save(path + '/home/tryon/inference/dataset/test_clothes/' + cloth_name)
u2net_run.infer(u2net, path + '/home/tryon/inference/dataset/test_clothes', path + '/home/tryon/inference/dataset/test_edge')
```

Extracting Edge algorithm.

We used subprocess to spawn a new process for the u2-net removes edging model otherwise it won't work as a thread.

Resizing input :

We need to resize the input the same as the cloth image and the edge image.



```
def resizing(token):
    img_name = f"000001_0_{token}.png"
    img_path = os.getcwd() + \
               f"/home/tryon/inference/inputs/img/result_0_{token}.png"
    img = Image.open(img_path)
    img = img.resize((192, 256), Image.BICUBIC)
    img_path = os.getcwd() + '/home/tryon/inference/dataset/test_img/' + img_name
    img.save(img_path)
    print('Resized images')
```

Resizing Input algorithm.

Virtual Try-On (PF-AFN) :

Since everything is ready and all the problems are solved we just need to pass the results we get after the previous steps and run our model.

```
def virtual_try_on_modal(token):
    try:
        subprocess.run(['python', 'test.py', '--token', token],
                      cwd='home/tryon/inference/')

        return True
    except:
        return False
```

Virtual Try-on model spawn.

We also need to spawn a process for this model where test.py contains the whole inference algorithm where you can find it in our GitHub repository

Probably you are wondering about the **token** that is mentioned in every step, basically, it's a unique id generated for each request to not confuse different requests at the same time and everything should be working fine in parallel.

The **subprocess** module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes. This module intends to replace several older modules and functions. We used the function **.run** where it allows us to process the result until the process we spawned finished its work. **cwd** is an argument to change the current directory where the file we want to execute should exist.

Chapter V

V. Project Management

V.1. Collaborative Work

V.1.1 Introduction :

To make this project happen and be successful we need to work all together and collaborate in everything we could help. Parallel tasks are also a great aspect for time-boosting and project-boosting while we need real tools to apply them. so in this chapter, we will detail those tools and project monitoring.

V.1.2 Tools :

Git :



Git is a technology used in software development to track changes in any file size to coordinate programming activities. Their aims include speed, data integrity, and support for distributed and non-linear workflows (thousands of parallel branches running on different systems).

In 2005 Linus Torvalds invented Git for the Linux kernel development, which was supported by other kernel engineers.

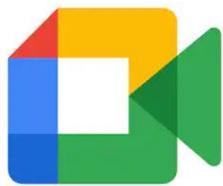
He has been the core maintainer since 2005. Each Git directory is a full-service, full versioned repository, independent of network access or central server, like most other distributed version control systems and other client-server systems. Git is the free and open-source GNU General Public License version 2 software.

Google Drive :



Google Drive is a file storage and sync service that is developed by Google. On 24 April 2012, Google Drive was introduced to allow users to store cloud files (on Google servers), to sync and share data across devices. Google Drive offers Windows/macOS PC and Android/IOS offline programs as well as a web interface and cellphones and tablets. Google Drive comprises Google Docs, Google Sheets and Google Slides that appear in the Google Docs editor's office suite that collaborates on the editing of documents, tablets, presentations, drawings, forms and other features. Google Drive preserves files generated and modified via the Google Docs Suite. So we saved our test files and our data and training files.

Google Meet :



Google Meet is a video service developed by Google (previously known as Hangouts Meet). It's one of two software solutions that replace Google Hangouts. We used Google to communicate with one another and to meet online in a remote manner when we wanted to be online.

Google Colab:

We have used Google Colab to share our notebooks for artificial intelligence algorithms and synchronize our work & files from google drive together and share results.

V.1.3 Project Monitoring :

About project monitoring, we were in touch with the Classbox supervisor almost every week and we were doing meets to see the progress perhaps each 2 to 3 weeks. For our university supervisor, we were doing online meetings whenever we needed him, he was always available to help and provide solutions and opinions.

V.2 Project Management Methodology

V.2.1 Introduction :

In the past few years, the use of project management methodologies has become increasingly common. These include traditional alternatives, such as PMP or Prince 2, as well as agile methodologies, such as SCRUM, Kanban, DSDM, and Extreme Programming.

Being certified in one of these management methodologies is an added value for individuals as well as companies, and it has even formed part of the criteria in deciding to work with certain clients.

Why are project management methodologies needed? What benefits can be gained by working in line with their recommendations?



Figure 52. Project management methodology

The general aim of project management methodology is to be able to standardize, structure, and organize work methods. This helps focus all projects in the same way and allows us to repeat successful aspects and learn from mistakes, resulting in a continuous improvement process. In other words, a methodology is a great tool for generating efficiency as it is used.

A project management methodology is used to obtain certain benefits:

- Organizing project times
- Providing tools to estimate times and costs correctly
- Helping to manage and minimize project risks
- Improving the cost-benefit relationship of resources
- Developing the team's skills

In terms of resources, a methodology helps shorten the team's learning curve and, as it is used in projects, it is improved and transformed in line with the company's personal style. With an adapted and standardized focus, it is possible to lower implementation risks and improve the work.

A decisive factor in applying management methodologies is that they are not all suitable for all projects, so it is essential to understand the strengths of each one so it may be applied properly.

V.2.2 Method used :

Since there are different methodologies, we have chosen the best one that could fit with software development and which is **Iterative and Incremental Project Management Methodology**, where we develop each phase and test it and monitor it then we pass to the second phase..etc until we build the full project.



Figure 53. Incremental Methodology

Conclusion

1. Conclusion :

In this project, we have solved one of the biggest problems in the world of fashion and humanity, no need to go to the store and shop especially in this COVID-19 period you can order your clothes from home and try it from home before even purchasing the product, customize your own design and there is an artificial intelligence system where it recommends you the colors based on your taste; a lot of customers return the product in fashion e-commerce websites because of the wrong size, well we have solved that problem too, you just have to enter your height and weight and it will recommend you the best fit size based on those entries. the website we have created is also responsive on mobiles, which means that you can try all our features via your mobile phone only without facing any issues. The Django framework has helped us a lot with releases that we needed such as linking the artificial intelligence algorithms with the website and also processing requests like creating APIs so quickly to interact with AJAX in real-time. For our benefit, we have learned several important things in the professional field, how the meetings go, how to progress, what are the priority tasks..etc. Also, we have learned different things in our computer science field like data pre-treatment, advanced Django, multiprocessing, and more essentially how to collaborate and work as a team.

2. Future Work :

Since our project is widely touching the world, it can be more developed and inspiring so here is a list of what we can develop in the future.

2.1 Virtual Try-on in Augmented Reality:

We can project our Virtual Try-on system to an augmented reality dimension where the user can see himself wearing the cloth in the real world via his camera phone or special glasses in the store so it's all about 3D modeling.



2.2 Size Estimation via Image :

Recommending size via weight and height is powerful but what about recommending it via only a picture of the person which is the same picture he will use for the virtual try-on and everything will be done via one single button, this will be very unique.



2.3 Outfit matching :

When a customer select a specific product we will recommend the correspond outfit that match the product that he selected so his full outfit will be compatible.



2.4 Popularity prediction :

A system that will predict if a product will be popular in the near future, To help the customer stay trendy ,fashionable and living the future.



References

- [1] Anne Bonner, The Complete Beginners Guide to Deep Learning, 2019.
- [2] Jake Frankenfield, Artificial Intelligence (AI), Investopedia, 2021.
- [3] Tom M. Mitche, McGraw-Hill Science/Engineering/Math; Page 1, 1997.
- [4] Sumit Saha, A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way, 2017.
- [5] Yuying Ge, Yibing Song, Ruimao Zhang, Chongjian Ge, Wei Liu, Ping Luo, Parser-Free Virtual Try-on via Distilling Appearance Flows (CVPR), 2021.
- [6] Yang, Han, and Zhang, Ruimao and Guo, Xiaobao and Liu, Wei and Zuo, Wangmeng and Luo, Ping, Towards Photo-Realistic Virtual Try-On by Adaptively Generating-Preserving Image Content, 2020.
- [7] Qin, Xuebin and Zhang, Zichen and Huang, Chenyang and Dehghan, Masood and Zaiane, Osmar and Jagersand, Martin, U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection, 2020.