SYSTEM REQUIREMENT SPECIFICATION FOR T-SHIRT DESIGN LAB

T-shirt Design Lab
Graduation Project
Dr. Taysir Hassan

By:

1-Ahmed Mohammed Mohammed Ahmed IS back-end Developer

Chapter I
Introduction

1.1 Introduction

creating an E-Commerce Web Site for marketing clothes “T-shirts” with Two layers of using The Site **(Normal User)** who buy a products or customize one if nothing like he\she, or can open a shop in Site of he\she market his products, **Adminstration** who see an admin panal for analysis some data, orders, shopping carts, updated Product and set his name, price and descount price, see users profiles and his orders, carts, customizations products, and he can Run data mining algorithm for see related products ex. See if user buy product X he\she maybe buy product Y and so .

Normal User he can let his\here review on a products “as a commint and rating of love it ”, he\she can pay cash on Delivary or any payment methods, can add many products or design it to his shopping carts for ordring it later time or same time .

For users who own shopps in my site they some heigh roles in their shops “CRUD Operation and additional functions in their shop and Only their shop like see mining in data and see knowledge ” .

Make an customization of the product if the user did not like any products or want to update a specific product and order it or save it in his profile, Update Product:- Open a product in another and Update it by adding some logos, add a text to design or remove logo, Design Product :- he\she can open a blank product and design it from scratch by the same functionality of Update product, Ex. If a user want to design a T-shirt he open a blank shirt and upload content of design and move it .

1.2 Scope of the project

- it can use by Users from any browsers or downloading phone Application
- customization can used by Person who had less time in real life for see all products
- t can easily understandable by every person because it’s content very Understandable by a normal person
- customization can used by persons ho had less time in real life for browsing all products
- It is based on web application we can easily access it from any where

1.3 Overall Description

Description “System features”

- create acount:- User want to buy or customize product or let review .
- Login : Users and Admins
- See user profile : the user can see only his profile, the admin can see all users profiles.
- Customization products : users or admins can customize products and save it or ordering it.
- Admin can see all orders, Users, Admins, mining results of related products .
- User can see only his orders and items in his cart and customized products .
- Admin can see any thing of users .
- User who buy product can pay cash or any payment methods .
- Decrease time for buy by customize the specific .

Chapter II
Requirement analysis and specification

2.1 product perspective

2.1.1 Interfaces

There are many types of interfaces as such supported by the E-learning software system namely; User Interface, Software Interface and Hardware Interface

2.1.2 Hardware Interfaces

- The hardware requirement at the end user is any PC and any browser
- for the server requirement it must have at least 20 GB RAM, hard disk of 50 TB and 2GHZ+ (5 cores) processor .

2.1.4Software interfaces

The application should support all major web browsers that will make it convenient for the user to access our system with ease .the back-end i.e. the database services will be used to a great extent and hence it will be quiet efficient designed

2.1.5 Communications interfaces

System shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol

2.2 constraints

2.2.1 Technology Constraints:

web application will be implemented PHP for back-end, for database with MySql

2.2.2 Interface Constraints:

Since this is a web based application so it should work on major browsers like Mozilla Firefox, Google Chrome, Opera etc and not work correct on internet Explorer.

2.2.3 Safety and Security Constraint:

for security if a user try to login in an upnormal place the acount will colse and concate with the user to check if an acount hacked or he just use VPN

2.3 System Requirement

For the website to run on the computer, the device is expected to meet the following system requirements.
The system requirements were categorized into hardware and software requirements.

- Client Software Architecture
  - For best result with PC, we recommend:
    Firefox (up to date).
    Chrome (up to date).

  - For best result with MAC, we recommend:
    Safari (up to date).
    Chrome (up to date).

  - For best result with Android phones :
    just install App from Google Play

- Server Requirement
  - Minimum Server Hardware Requirement:
    - 2GHZ+ (5 cores) processor
    - 20+GB RAM.
    - 50 TB free hard drive space.
  - Server Software Requirement:
    - Linux server 64-bit.
    - Server API apache 2.0 handler.
    - Php version 7.3.11
    - Apache version apache/2.4.41 (Unix) openSSL 1.1.11d.
    - MySQL Server MariaDB Server version10.4.8-MariaDB.

* Server Software Requirement
  - Linux Server 64-bit
  - Server Configration

Tool	Version
PHP	7.3.11
Composer	1.9.1
laravel-framework	6.5.0
Python	3.7.5
pip	18.1
Flask	1.1.1
OpenCV	4.1.2.30
Flask-RESTful	0.3.7

2.4 User Requirement

Create an web Application that marketing the colthes to Users with the folowing constrain:

I. User Can see products and Users reviews on it and see categories .
II. User can create account before buy, Customize, review or add products to his cart “IF he\she has acount must login first ”
III. For Login user can login with E-Mail or phone number in the same feild .
IV. Every user should have an acount for him and only he\she and the admin can see the acount .
V. In user profile show orders, items in cart and customized products.
VI. Create an Admin panal for System.

2.5 Functional Requirement

On our website prototype can be divided into three parts.
Each of them involving different functionalities in our website as illustrated in the use case diagram.

- Part for the customer.
- Part for the admin.

2.5.1 Functional Requirement for User “customer”

- **Registration:**
  - If the customer did not have an account on the site, the customer should have an account on the site to allow him to make transaction on the system.

- **Login:**
  - The customer should be able to login using his e-mail and password.

- **Browsing the website:**
  - The customer should be able to see all products in the site and see more details of any product he/she like.

- **Search for the product:**
  - The customer can search for any product and filter the products by his/her interests.

- **Buy the product:**
  - The customer can by any product match his/her interests by adding the product to the card.

- **Customize specific product:**
  - The customer should customize a specific by design it by

the way he/she like and send his/her design then the factory makes it.

- **Write the review:**
  - The customer should write his review about any product he/she buy it or customize it and rate it.

- **Adding product to card:**
  - The customer adds any product he/she like to the card and manage his/her card.

- **Payment:**
  - The system should be able to ask the user if he wants to pay in cash or visa.
  - If the customer chooses to visa the system should be able to ask the customer to enter his/her visa details and with-draw money from it money.
  - If the customer chooses cash the system should ask him/her the location of the customer to pay the money.

- **Log out:**
  - The customer should log out after finishing his/her transactions on the site.

2.5.2 Functional Requirement for Admin

- **Login:**
  - The admin should be able to login using his e-mail and password.

- **Manage the product:**
  - The admin should be able to manage the product the admin can update product data or /and search for the product or/and delete products or/and add products

- **Manage the customer:**
  - The admin should be able to manage the customer the admin can search for the customer data or/and delete customer data and so.

- **Manage the reviews:**
  - The admin should be able to manage the customer reviews to increasing benefits of the system

- **Show mining result of reviews:**
  - The admin should be able to show mining result for reviews which the customer writes to increasing benefits of the system

- **Log out:**

The admin should log out after finishing his/her transactions on the site.

2.6 Non-Functional Requirement

- **Availability:**
  - The designed system should have little or no down time.
  - It should always be up and running

- **usability:**
  - The system and the user interface should be easy to use and easy to understand by most user.
  - in order to allow customers to use the system within short period of timeframe. moreover, the user interface should be designed to be simple and suitable to display concrete information on the limited screen of mobile device.

- **speed:**
  - The system should have fast response time.
  - The system takes more than 40 seconds minus loading

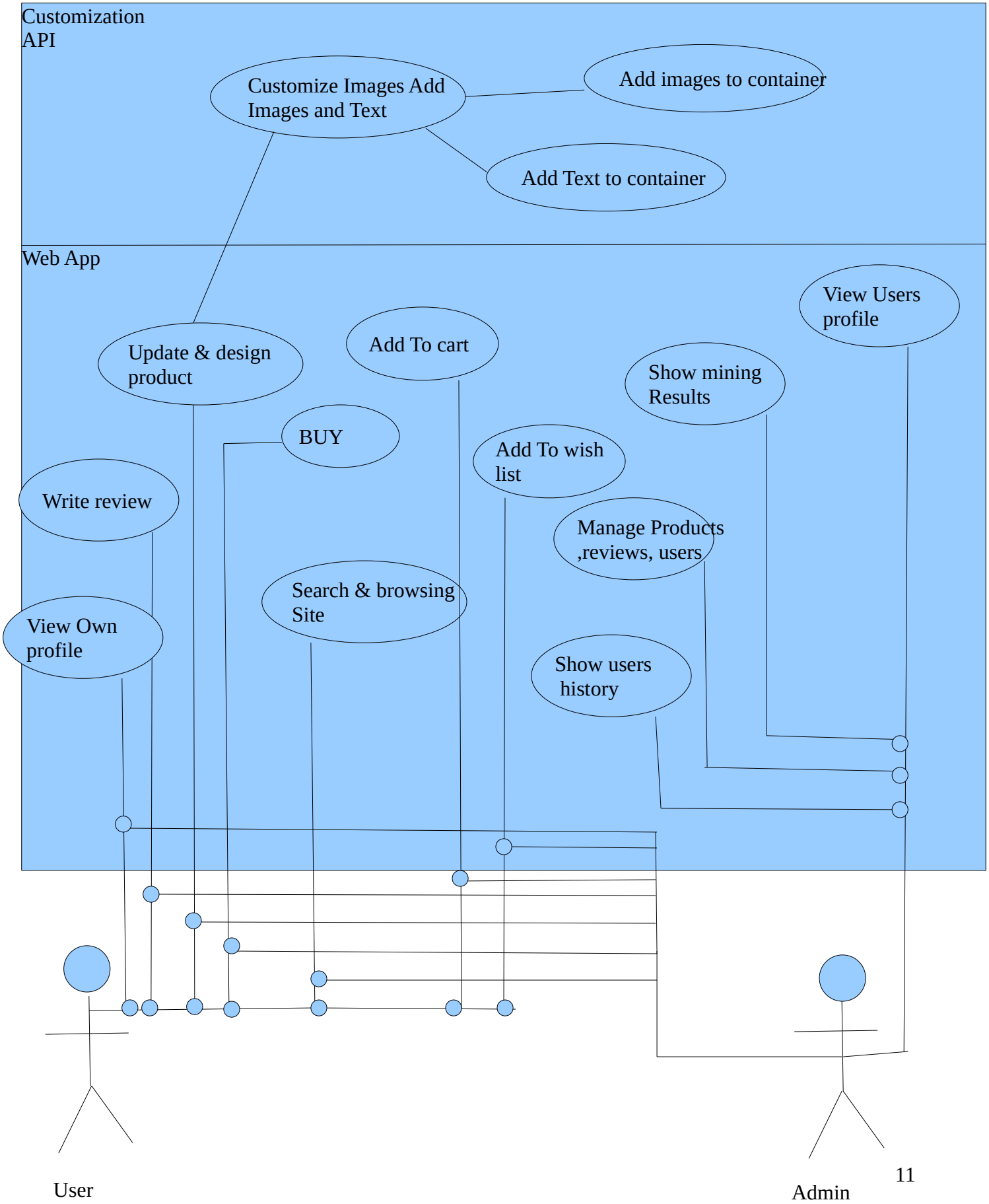
- **scalability:**
  - The system should be scalable.
  - Even with an increasing number of users, system should be able to perform effectively.

- **reliability:**
  - The system should be reliable.in case of the system failure, the system should be able to recover quickly and continue working normally.

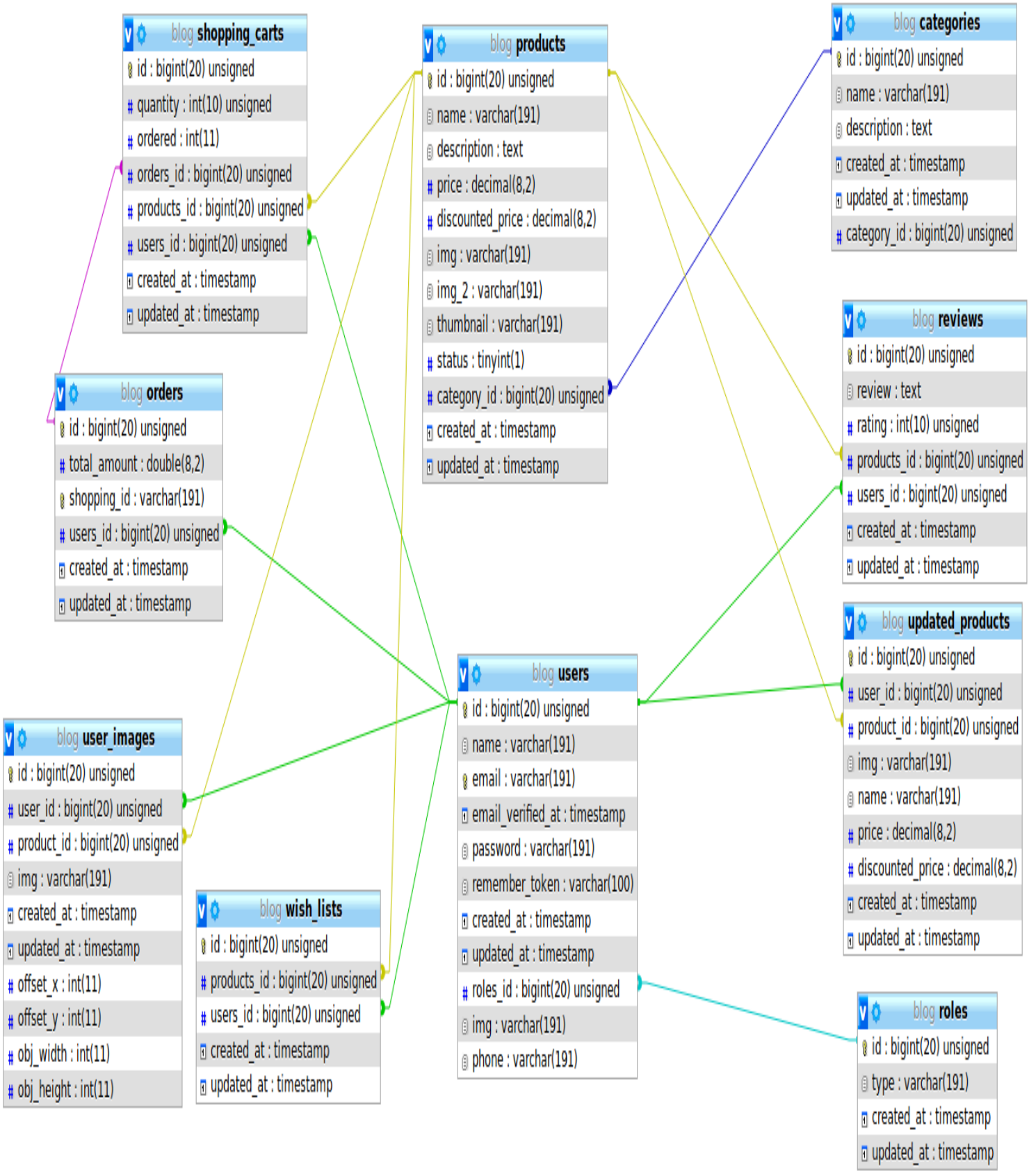
- **User friendly:**
  - The system should be user friendly with ability to show users where they are in the system and guide them on some processes through programmed controls

Chapter III
System Design

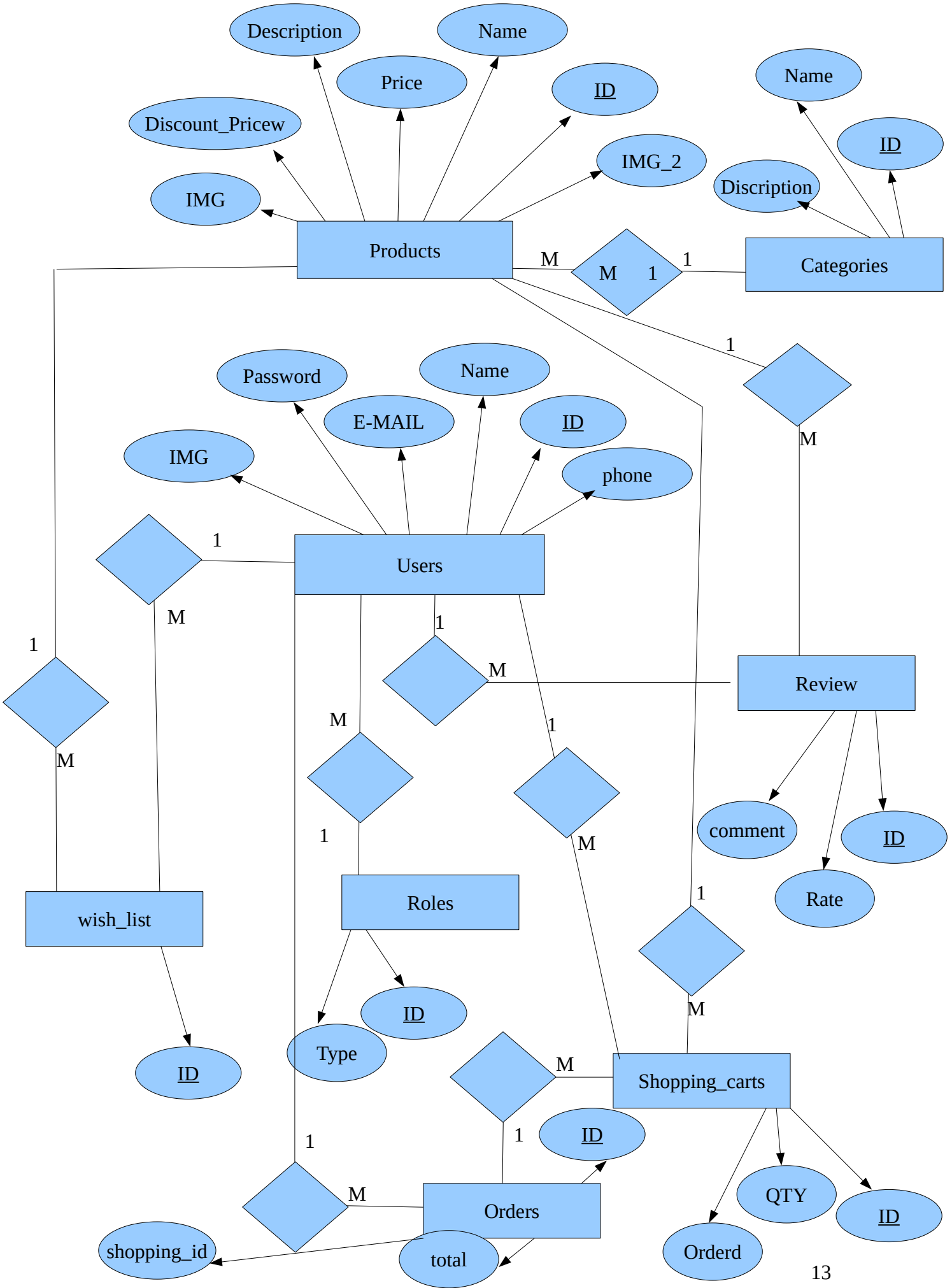
3.1 Use case diagram



3.2 Class diagram



3.3 ERD diagram



Entity Relation ship Diagram (ERD), a database design tool that provides graphical representation of database tables, their columns and their relationships.

3.4 System Architect

The T-shirt Design Lab is Implemented and deployed Using:

- PHP \ laravel framework back-end technology
- MySql \ ORM Object Relation Mapper
- HTML \ CSS front-end technology
- JavaScript \ Jquery UI technology
- Bootstrap
- AJAX \ JSON
- Python
- Flask
- OpenCV
- API techniqe

3.4.1 Database

the database will store all users data, products, carts, orders, updated products. All data will be accessed and manipulated by back-end System deployed using ORM “Object Relation Mapper”

3.4.2 Front-end

The front-end will used by implementation with HTML, CSS, JavaScript, jQuery and Bootstep.it includes functions allowing user to use the system as discussed previously.

3.4.3 Back-end

The web-based client will be used by customer and admin .it will be implemented PHP and Laravel.

3.4.4 Customization API

using an python with OpenCV, Flask Packages for appending images into a container and add text to a container .

3.5 Customization API Guide

1. Insert Images To an Image .

URL	***Methods***
`http://127.0.0.1:5000/api/`	POST

*Parameters Send JSON Notation

```
{
	"l_img":{  // container image object
		"img_src":"path Of the container Image",
		"width":width for resizing ,
		"height":height for resizing
	},
	"s_img":[ // array of images to append it in container image
		{
			"img_src":"BImg/! Fannan NewLook Ani (1).jpg",
			width":width for resizing ,
			"height":height for resizing ,
			"x":position in container x-coordinates,
			"y": position in container y-coordinates
		},
		{
			"img_src":"BImg/! Fannan NewLook Car.jpg",
			width":width for resizing ,
			"height":height for resizing ,
			"x":position in container x-coordinates,
			"y": position in container y-coordinates
		}
		]
}
```

*Responses

response_code	**Status**	**Message**
1	Success	name of new Image
2	Failed	parameters not send
3	Failed	Parameters is not complete
4	Failed	failed to load an Image from source

1. insert an text to an Image

URL	***Methods***
http://127.0.0.1:5000/api/addText/	POST

** Parameters Send Over JSON Notation

```
{
	"img":{
		"img_src":"BImg/! ! Fannan NewLook.jpg",
		" width":width for resizing ,
		"height":height for resizing ,
		"x":position of text to append in container x-coordinates,
		"y": position of text to append in container y-coordinates,
		"font_size":1,
		"font_family":8,//index of fonts avaliable
		"font_color":[225,225,225]//font color must Be In Array RGB(blue, green, read)
	},
	"text":"TEXT To Add It In JSON"
}
```

response_code	**Status**	**Message**
5	Success	Add Text Success
6	Failed	Failed No data Send
7	Failed	Failed Parameters Is Not Complete
8	Failed	Failed index of Font Family in not successfully
9	Failed	Failed color format in not successfully

I. Get Fonts Avalable in Python

URL	***Methods***
http://127.0.0.1:5000/api/addText/font_family/	GET

** response

```
{  "0": "cv.FONT_HERSHEY_COMPLEX",
  "1": "cv.FONT_HERSHEY_COMPLEX_SMALL",
  "2": "cv.FONT_HERSHEY_DUPLEX",
  "3": "cv.FONT_HERSHEY_PLAIN",
  "4": "cv.FONT_HERSHEY_SCRIPT_COMPLEX",
  "5": "cv.FONT_HERSHEY_SCRIPT_SIMPLEX",
  "6": "cv.FONT_HERSHEY_SIMPLEX",
  "7": "cv.FONT_HERSHEY_TRIPLEX",
  "8": "cv.FONT_ITALIC"  }
```


I. Insert Images and word to an Image

URL	***Methods***
http://127.0.0.1:5000/api/save/result/	POST

** parameters


```
{
  "container": {
    "height": 500,
    "img_src": "BImg/! ! Fannan NewLook.jpg",
    "width": 500
  },
  "images": [
    {
      "height": 100,
      "img_src": "BImg/! Fannan NewLook Ani (1).jpg",
      "width": 100,
      "x": 50,
      "y": 450
    },
    {
      "height": 120,
      "img_src": "BImg/! Fannan NewLook Car.jpg",
      "width": 120,
      "x": 0,
      "y": 0
    }
  ],
  "word": {
    "font_color": [
      225,
      225,
      225
    ],
    "font_family": 3,
    "font_size": 1,
    "text": "Value Added",
    "x": 50,
    "y": 50
  }
}
```

** Responses

response_code	Status	Message
10	Success	item Save Success
6	Failed	Failed No data Send
7	Failed	Failed Parameters Is Not Complete
8	Failed	Failed index of Font Family in not successfully
9	Failed	Failed color format in not successfully
4	Failed	failed to load an Image from source

3.6 Database Schema

Below are tables contained in the T-shirt Design Lab System Database:

- Categories

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**name**	varchar(191)	utf8mb4_unicode_ci		No	*None*		
3	**description**	text	utf8mb4_unicode_ci		No			
4	**created_at**	timestamp			Yes	*NULL*		
5	**updated_at**	timestamp			Yes	*NULL*		
6	**category_id**	bigint(20)		UNSIGNED	No	*None*		

- Orders

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**total_amount**	double(8,2)			No	*None*		
3	**shopping_id**	varchar(191)	utf8mb4_unicode_ci		No	*None*		
4	**users_id**	bigint(20)		UNSIGNED	No	*None*		
5	**created_at**	timestamp			Yes	*NULL*		
6	**updated_at**	timestamp			Yes	*NULL*		

- Products

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
name	varchar(191)	utf8mb4_unicode_ci		No	*None*		
description	text	utf8mb4_unicode_ci		No			
price	decimal(8,2)			No	*None*		
discounted_price	decimal(8,2)			No	*None*		
img	varchar(191)	utf8mb4_unicode_ci		No	*None*		
img_2	varchar(191)	utf8mb4_unicode_ci		No	*None*		
thumbnail	varchar(191)	utf8mb4_unicode_ci		No	*None*		
status	tinyint(1)			No	0		
category_id	bigint(20)		UNSIGNED	No	*None*		
created_at	timestamp			Yes	*NULL*		
updated_at	timestamp			Yes	*NULL*		

- Reviews

Name	Type	Collation	Attributes	Null	Default	Comments	Extra
id	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
review	text	utf8mb4_unicode_ci		No			
rating	int(10)		UNSIGNED	No	*None*		
products_id	bigint(20)		UNSIGNED	No	*None*		
users_id	bigint(20)		UNSIGNED	No	*None*		
created_at	timestamp			Yes	*NULL*		
updated_at	timestamp			Yes	*NULL*		

- roles

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**type**	varchar(191)	utf8mb4_unicode_ci		No	*None*		
3	**created_at**	timestamp			Yes	*NULL*		
4	**updated_at**	timestamp			Yes	*NULL*		

- Shopping_carts

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**quantity**	int(10)		UNSIGNED	No	*None*		
3	**ordered**	int(11)			No	-1		
4	**orders_id**	bigint(20)		UNSIGNED	Yes	*NULL*		
5	**products_id**	bigint(20)		UNSIGNED	No	*None*		
6	**users_id**	bigint(20)		UNSIGNED	No	*None*		
7	**created_at**	timestamp			Yes	*NULL*		
8	**updated_at**	timestamp			Yes	*NULL*		

- Updated_products

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**user_id**	bigint(20)		UNSIGNED	No	*None*		
3	**product_id**	bigint(20)		UNSIGNED	Yes	*NULL*		
4	**img**	varchar(191)	utf8mb4_unicode_ci		No	*None*		
5	**name**	varchar(191)	utf8mb4_unicode_ci		Yes	*NULL*		
6	**price**	decimal(8,2)			Yes	*NULL*		
7	**discounted_price**	decimal(8,2)			Yes	*NULL*		
8	**created_at**	timestamp			Yes	*NULL*		
9	**updated_at**	timestamp			Yes	*NULL*		

- Users

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**name**	varchar(191)	utf8mb4_unicode_ci		No	*None*		
3	**email**	varchar(191)	utf8mb4_unicode_ci		No	*None*		
4	**email_verified_at**	timestamp			Yes	*NULL*		
5	**password**	varchar(191)	utf8mb4_unicode_ci		No	*None*		
6	**remember_token**	varchar(100)	utf8mb4_unicode_ci		Yes	*NULL*		
7	**created_at**	timestamp			Yes	*NULL*		
8	**updated_at**	timestamp			Yes	*NULL*		
9	**roles_id**	bigint(20)		UNSIGNED	No	1		
10	**img**	varchar(191)	utf8mb4_unicode_ci		Yes	noImage.jpg		
11	**phone**	varchar(191)	utf8mb4_unicode_ci		No	*None*		

- User_images

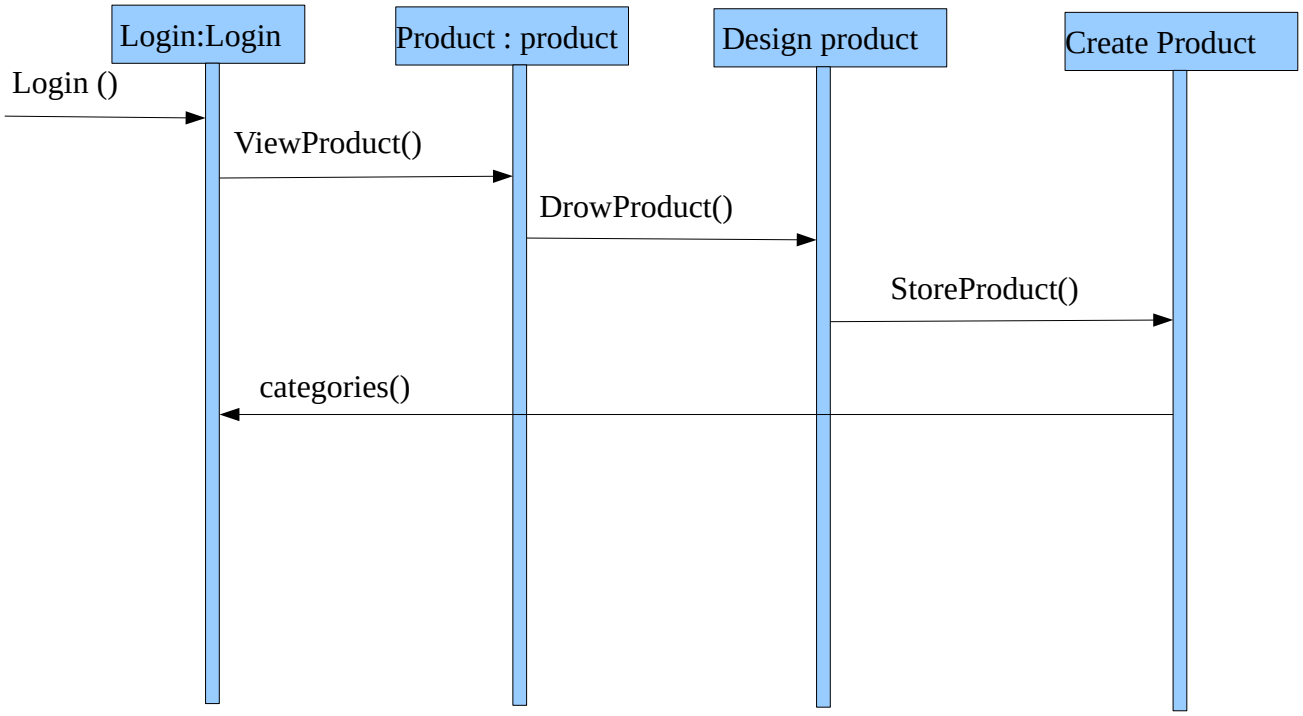
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**user_id**	bigint(20)		UNSIGNED	No	*None*		
3	**product_id**	bigint(20)		UNSIGNED	No	*None*		
4	**img**	varchar(191)	utf8mb4_unicode_ci		No	*None*		
5	**created_at**	timestamp			Yes	*NULL*		
6	**updated_at**	timestamp			Yes	*NULL*		
7	**offset_x**	int(11)			Yes	-1		
8	**offset_y**	int(11)			Yes	-1		
9	**obj_width**	int(11)			Yes	-1		
10	**obj_height**	int(11)			Yes	-1		

- wish_lists

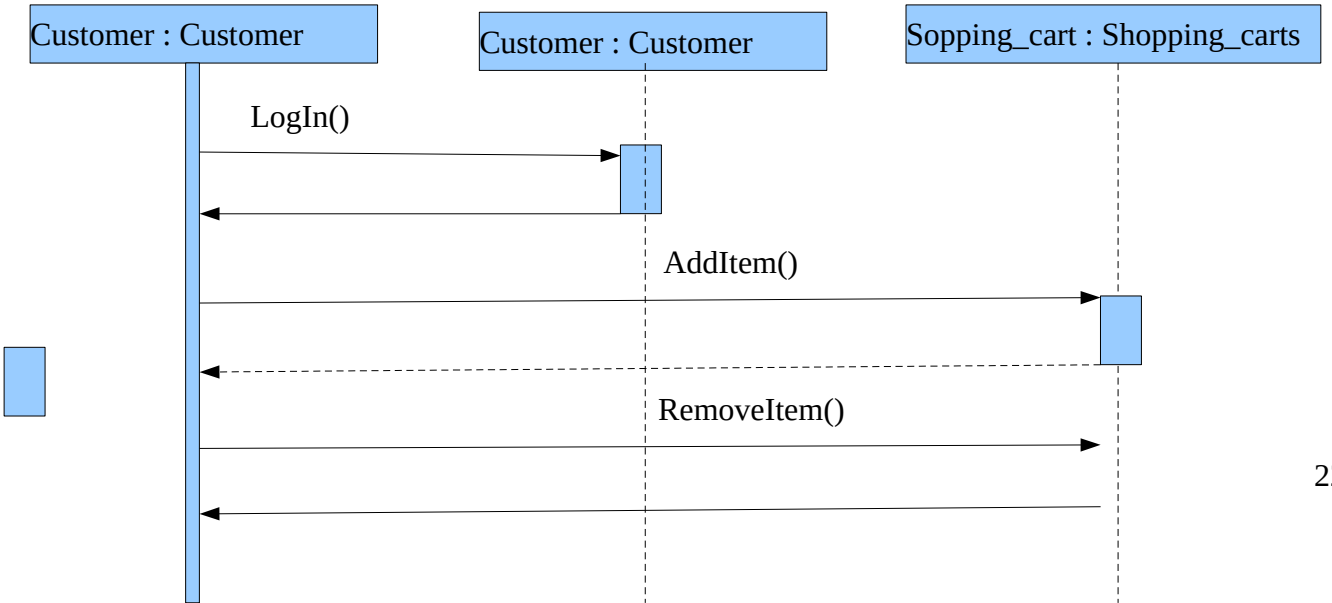
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	bigint(20)		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**products_id**	bigint(20)		UNSIGNED	No	*None*		
3	**users_id**	bigint(20)		UNSIGNED	No	*None*		
4	**created_at**	timestamp			Yes	*NULL*		
5	**updated_at**	timestamp			Yes	*NULL*		

3.7 Sequence Diagrame

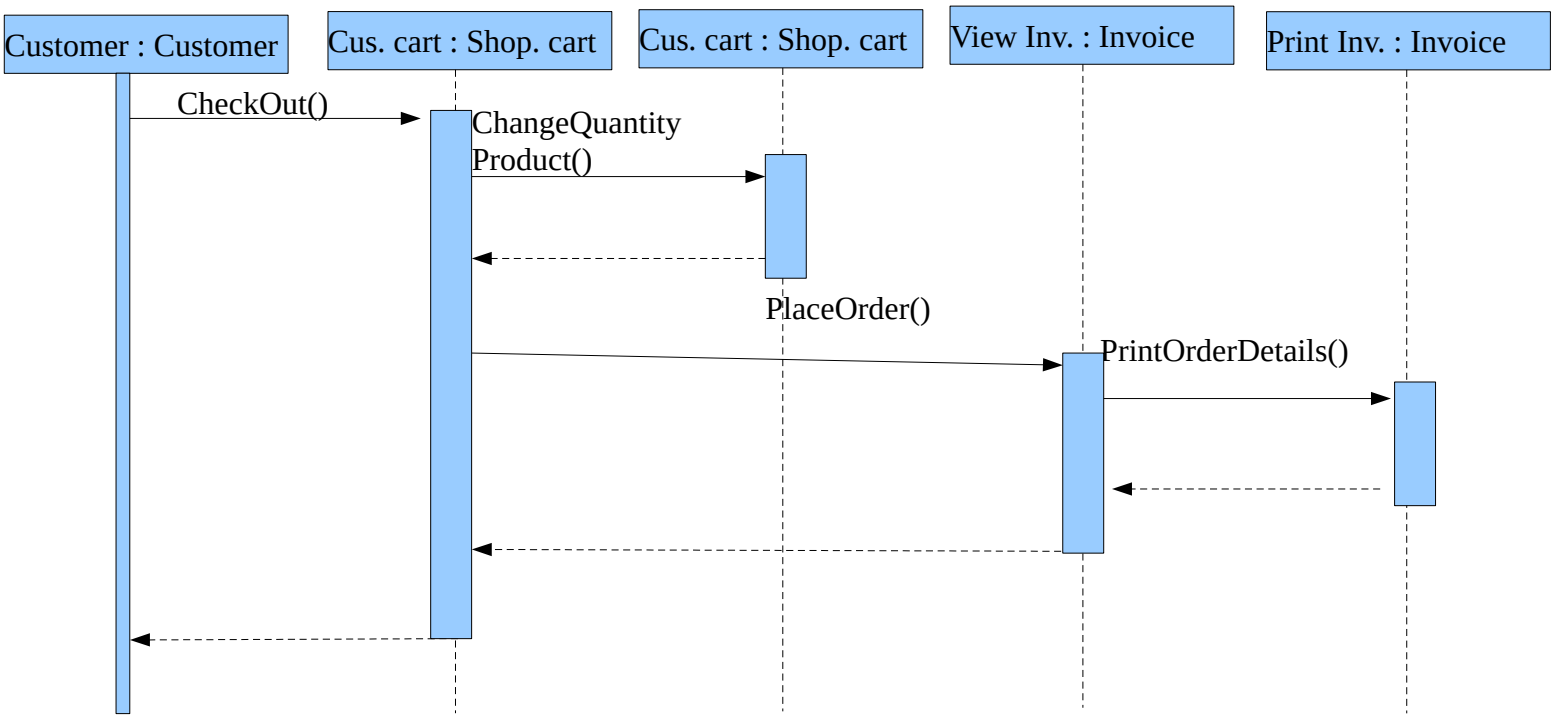
- Customize Product



- Customer Add & Remove Shopping carts

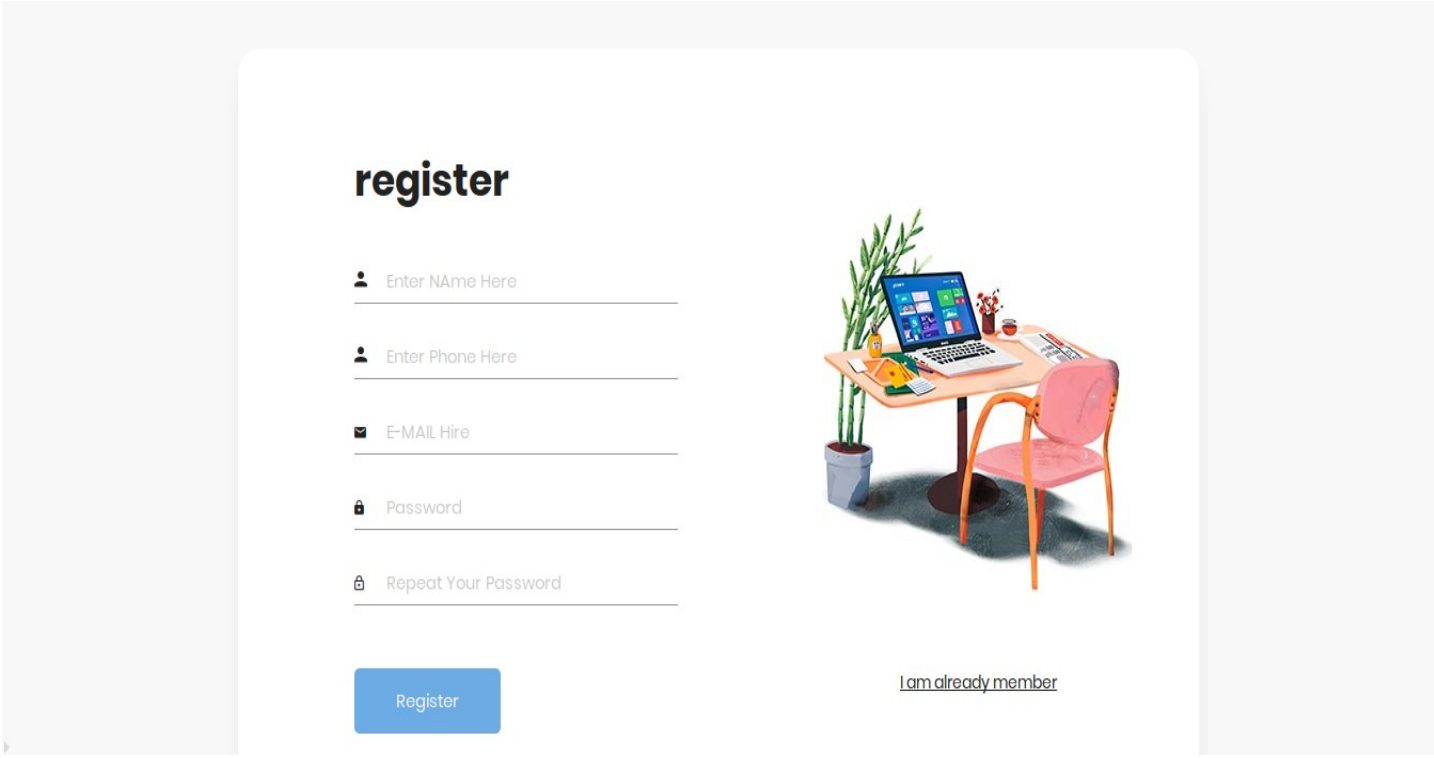


- Customer checkout

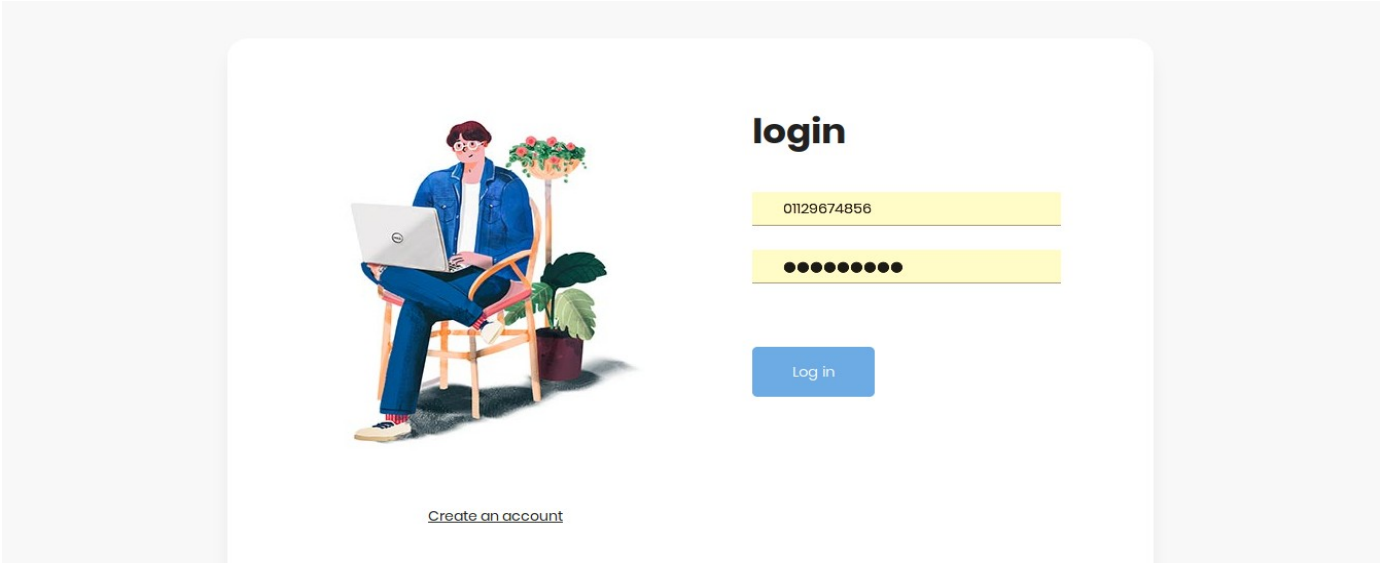


Chapter IV
Result Of Implementation

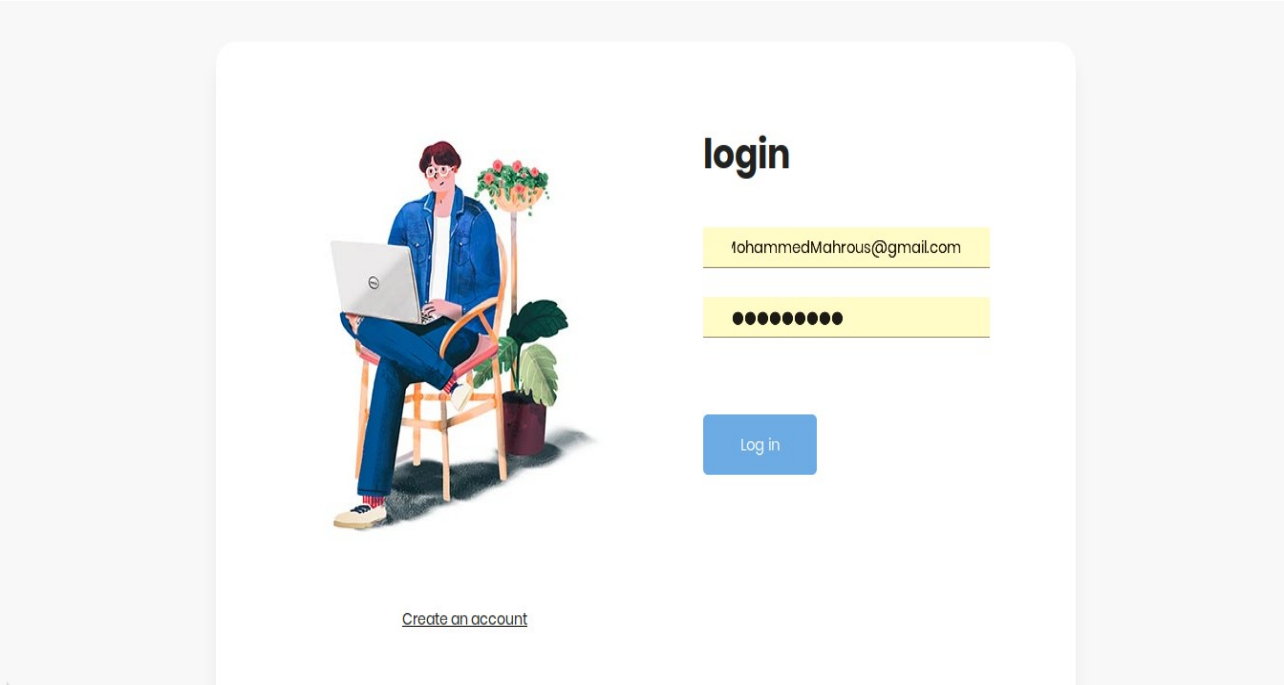
- Registeration



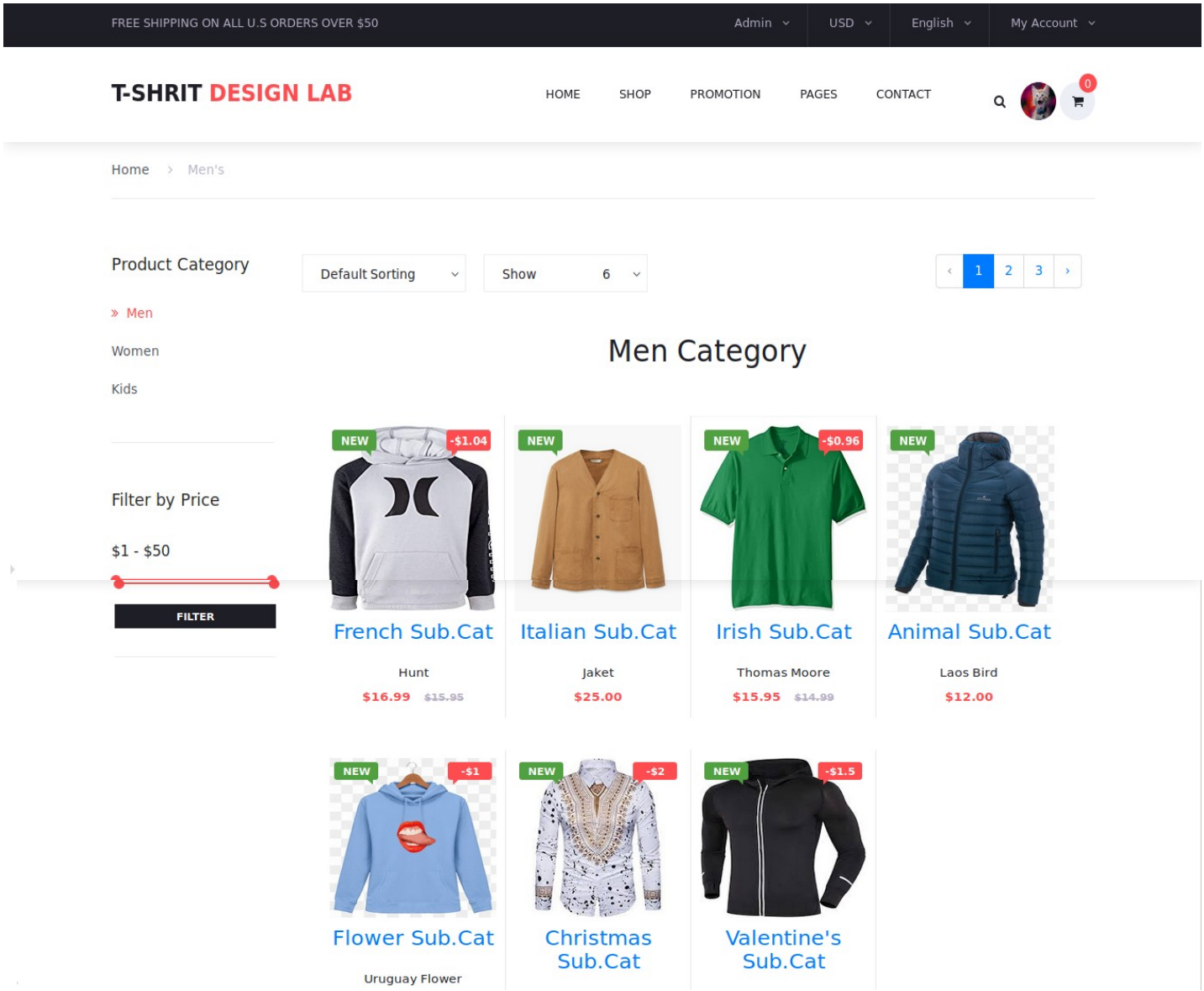
- Login
  with phone number



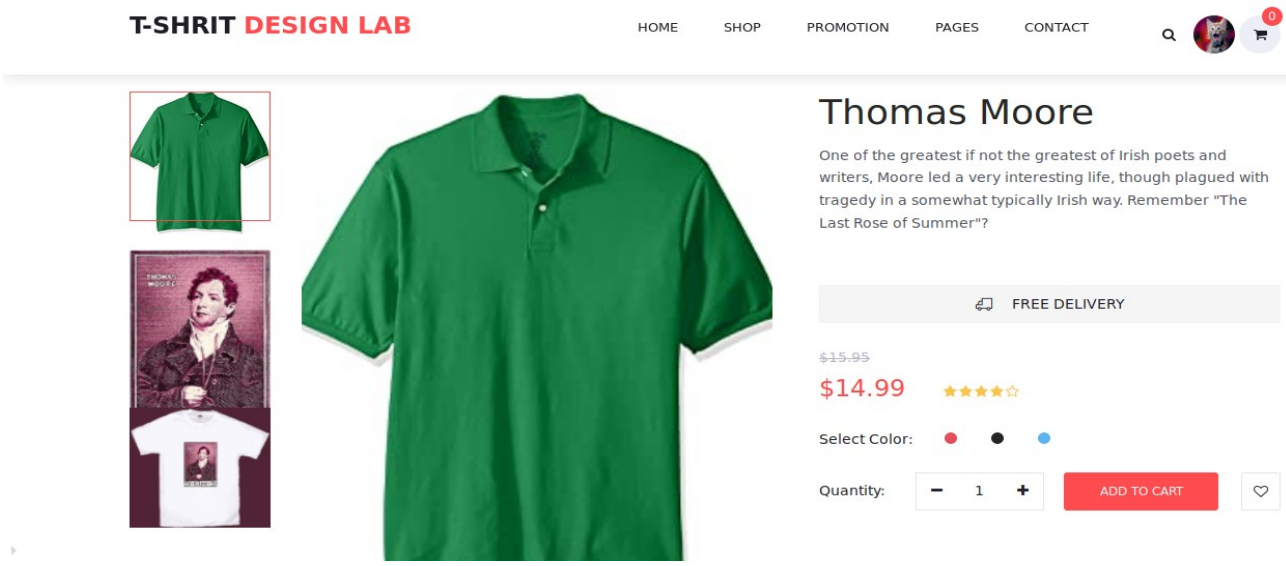
with E-Mail



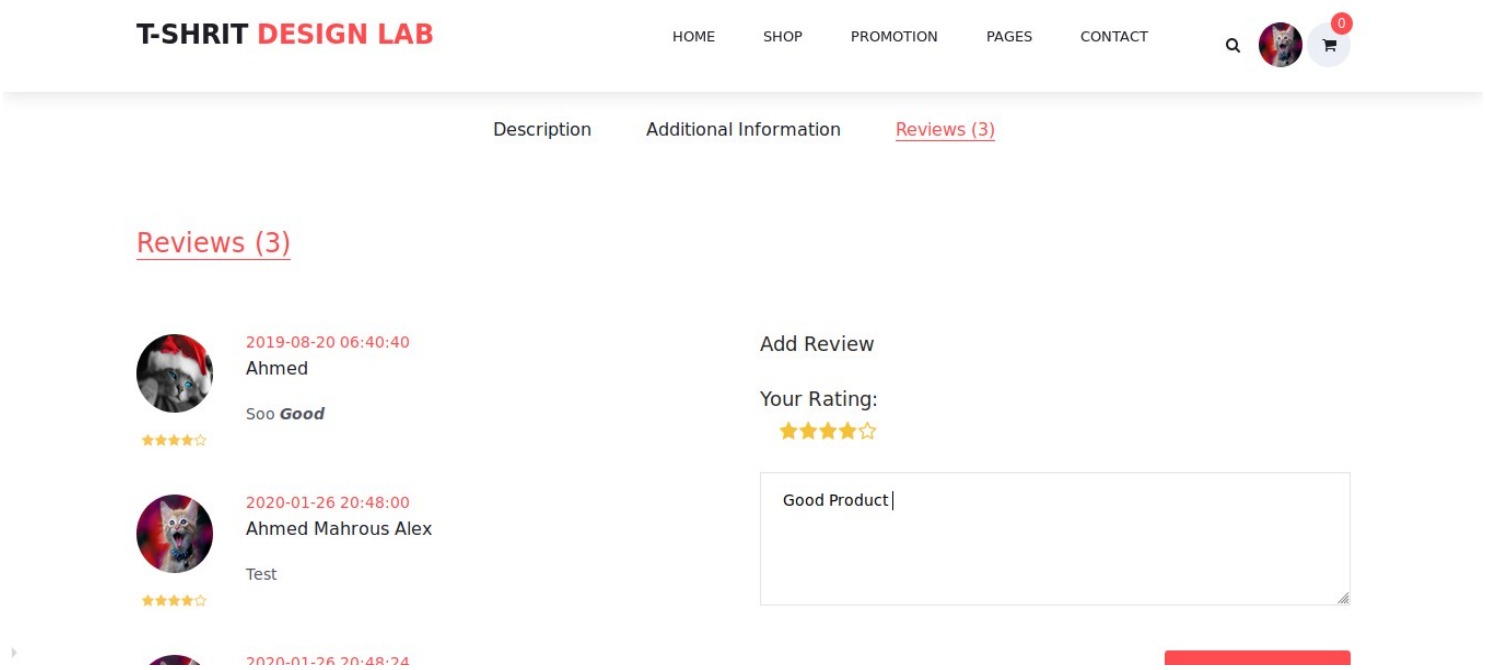
- Home Page



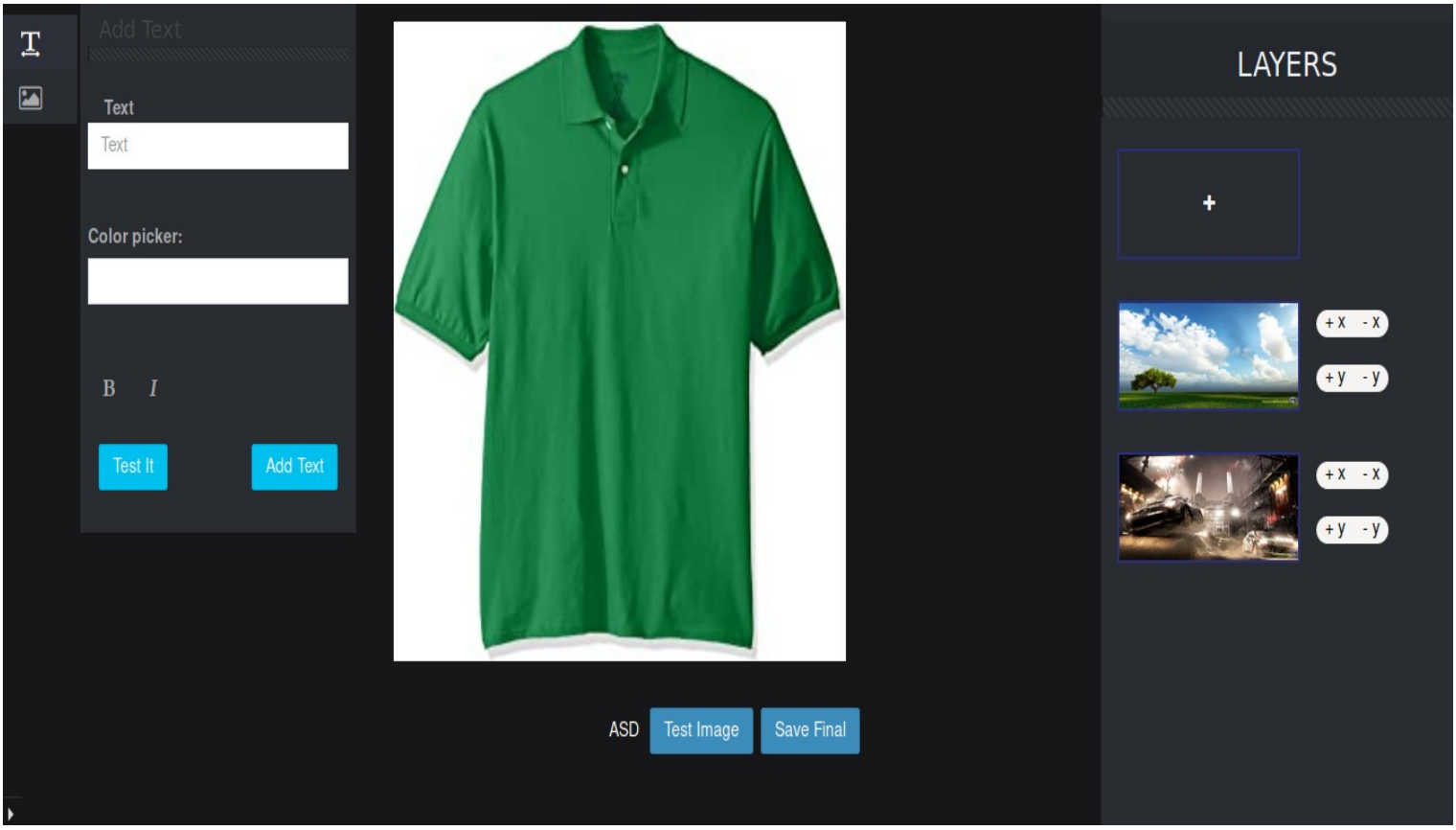
- single Product



- Reviews On Products

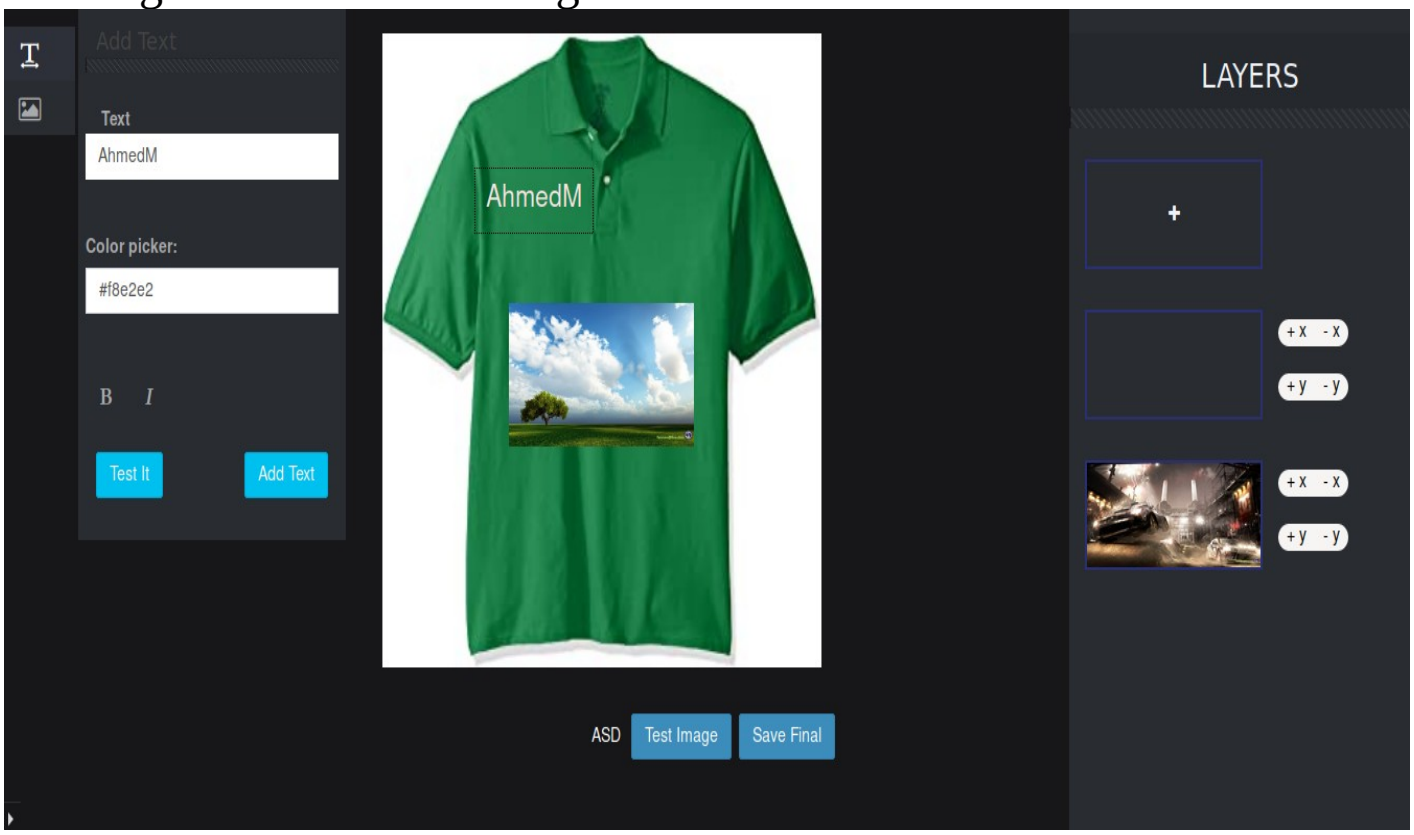


- Update On A specific Product

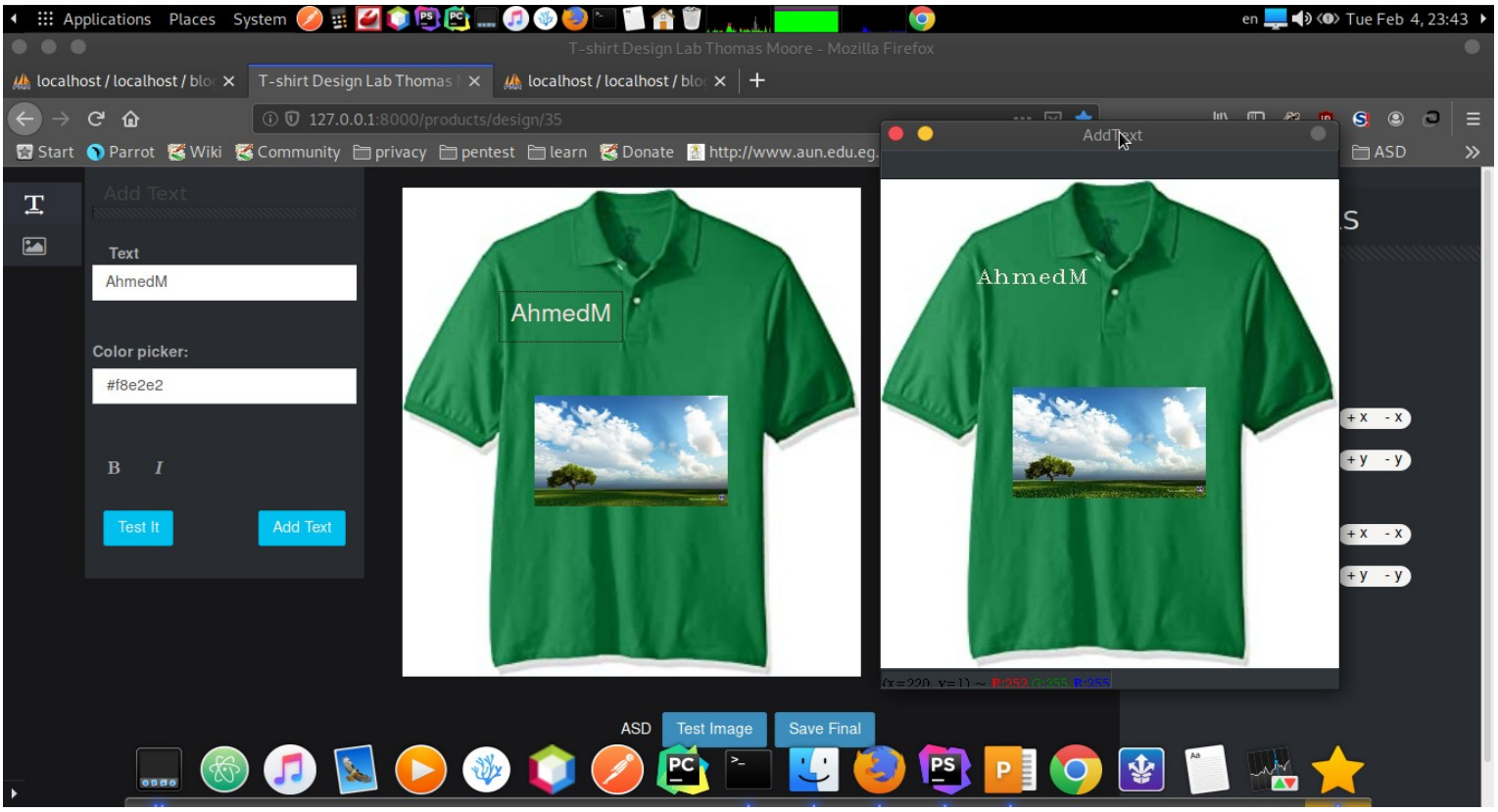


- Design Product UI

Images And Text Are Dragable Elements

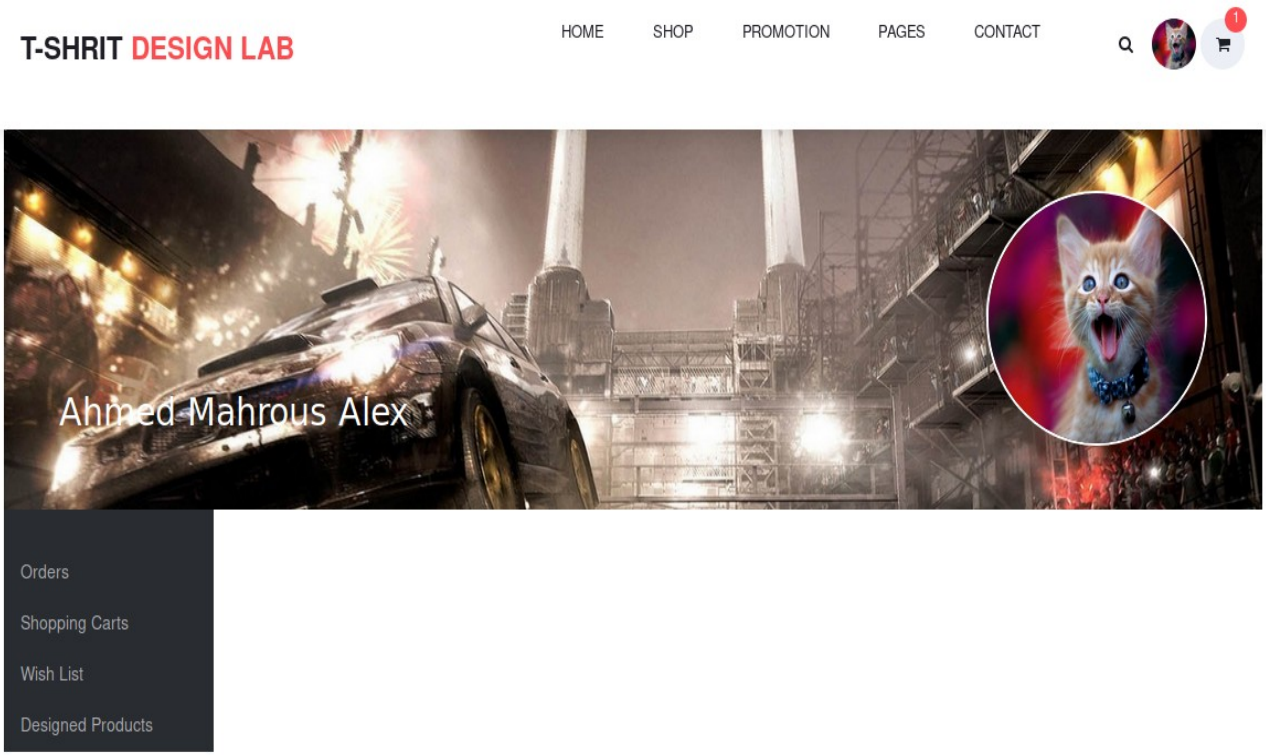


- Design Product Result



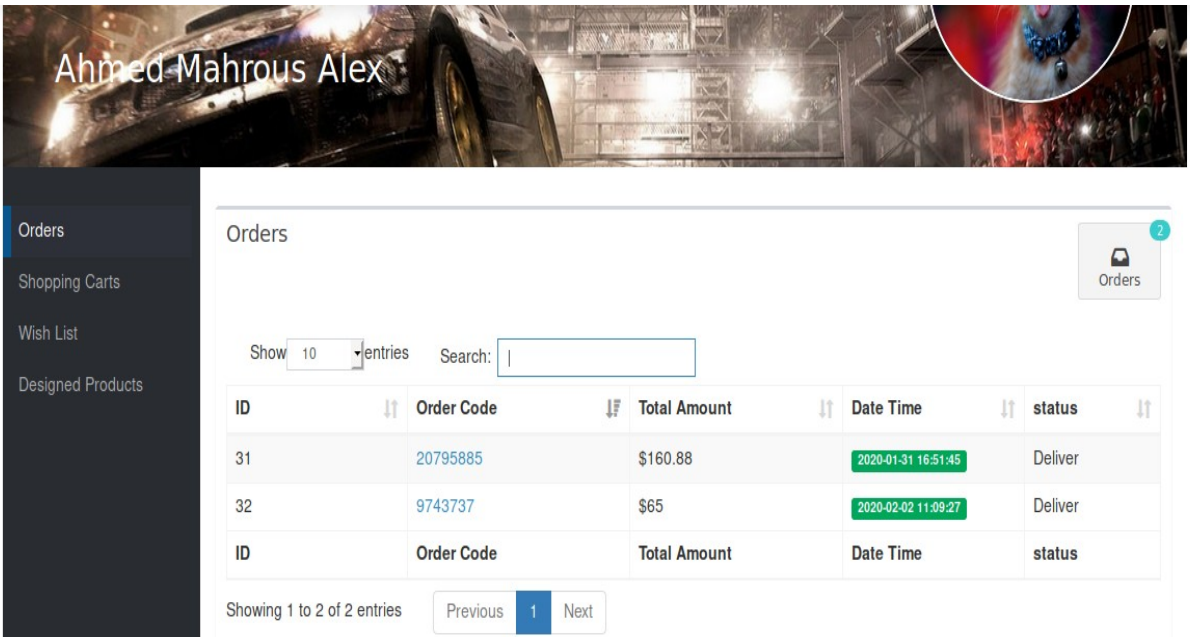
- User Profile

the smale images represent profile image for comments and user can change it click in image for changing



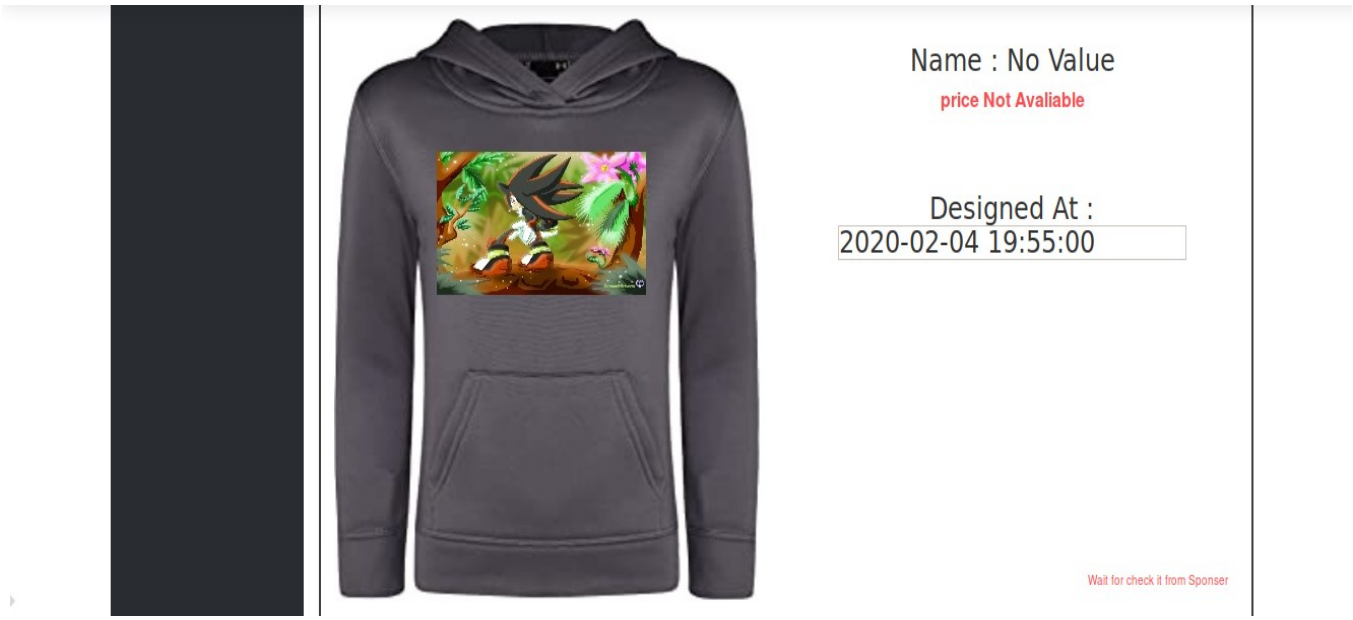
- User Orders

only the user and admin can see it

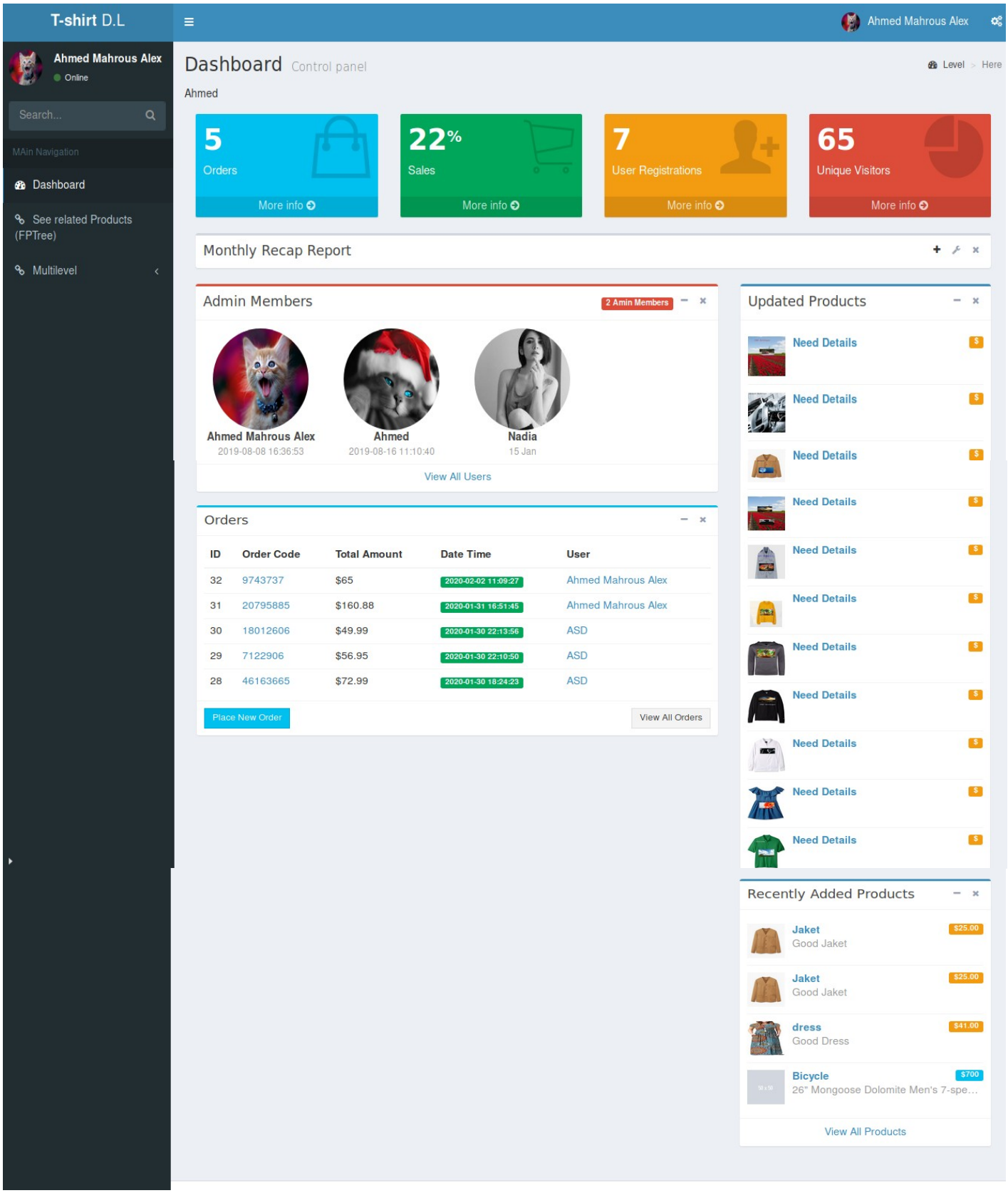


- Designed product

the user and admin can see it only them

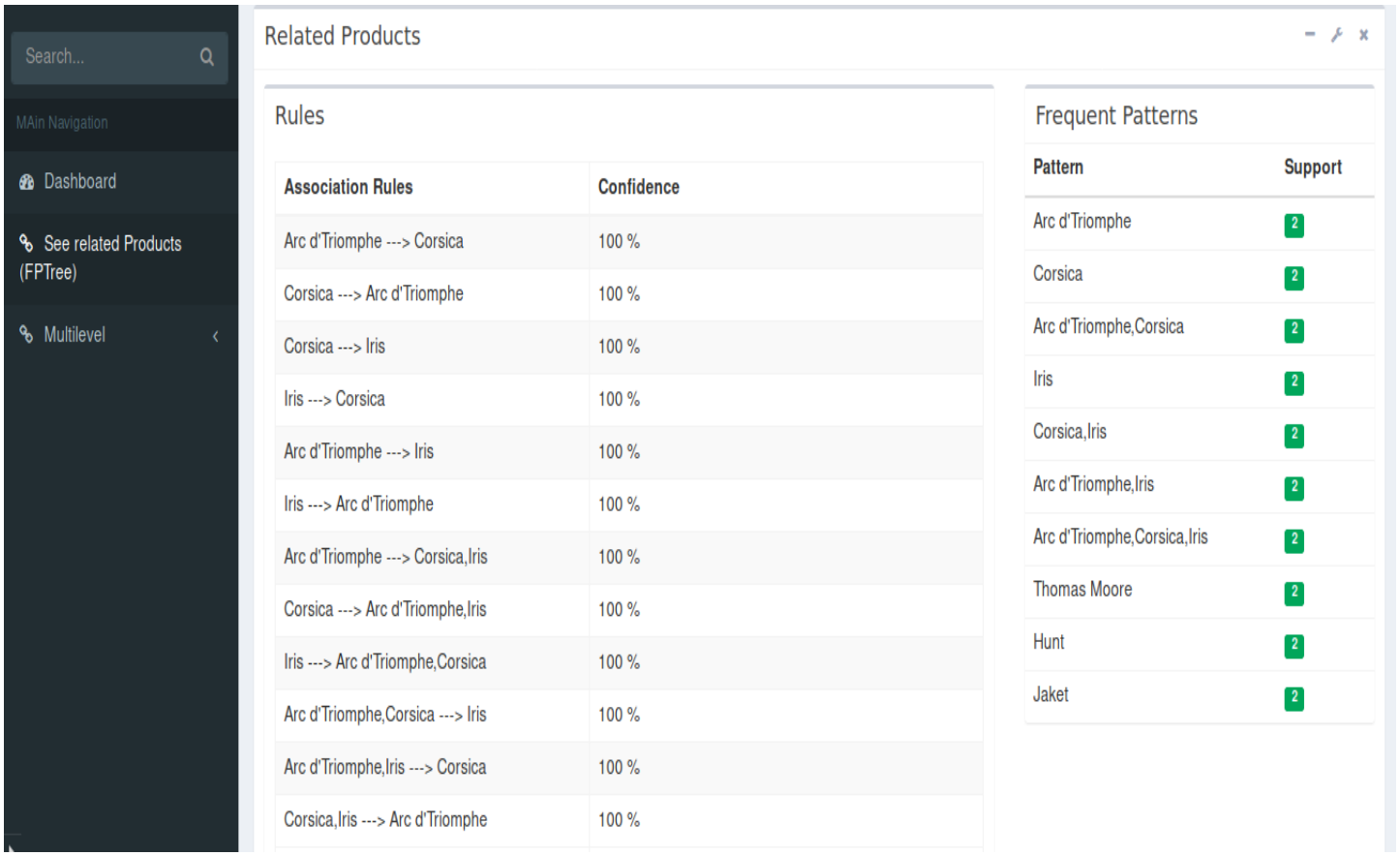


- Admin Panal


T-shirt D.L
Ahmed Mahrous Alex
Online
Search...
MAin Navigation
Dashboard
See related Products (FPTree)
Multilevel
Ahmed Mahrous Alex
Dashboard Control panel
Level > Here
Ahmed
5
Orders
More info
22%
Sales
More info
7
User Registrations
More info
65
Unique Visitors
More info
Monthly Recap Report
Admin Members
2 Amin Members
Ahmed Mahrous Alex
2019-08-08 16:36:53
Ahmed
2019-08-16 11:10:40
Nadia
15 Jan
View All Users
Orders
ID
Order Code
Total Amount
Date Time
User
32
9743737
$65
2020-02-02 11:09:27
Ahmed Mahrous Alex
31
20795885
$160.88
2020-01-31 16:51:45
Ahmed Mahrous Alex
30
18012606
$49.99
2020-01-30 22:13:56
ASD
29
7122906
$56.95
2020-01-30 22:10:50
ASD
28
46163665
$72.99
2020-01-30 18:24:23
ASD
Place New Order
View All Orders
Updated Products
Need Details
Need Details
Need Details
Need Details
Need Details
Need Details
Need Details
Need Details
Need Details
Need Details
Recently Added Products
Jaket
Good Jaket
$25.00
Jaket
Good Jaket
$25.00
dress
Good Dress
$41.00
Bicycle
26" Mongoose Dolomite Men's 7-spe...
$700
View All Products

- related Products

after applay FPTree algorithm see Frequent Pattern and Associasion rules



Index

BY:

I. Ahmed Mohammed Mohammed Ahmed - PHP\laravel Developer (coded the customization API using Python OpenCV Flask)
II. Magdy Abdel Samea Ali – Android Developer
III. Ahmed Mohammed Ahmed Ali -
IV. Osama Nabile Mahmoud -
V. Ahmed Meshaal Kamel -