**A**

**PROJECT REPORT**

**on**

**FavCart**

**at TECHPILE TECHNOLOGY PVT. LTD., LUCKNOW**



**Submitted Towards Partial Fulfillment of**

**Three-Year diploma in**

# Computer Science & Engineering

**Under the supervision of**

**Mr. Rahul Soni**

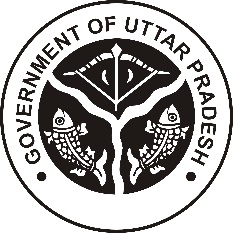
**SUBMITTED TO: SUBMITTED BY:**

**VIVEK SRIVASTAVA SAURABH KUMAR MISHRA**

**Roll No. 2252283893002**

**GOVERNMENT POLYTECHNIC CHOPAN SONEBHADRA**

**Session 2022-2023**



**Enroll: Techpile-ST221701**

**COMPLETION CERTIFICATE**

This is to certify that Saurabh Kumar Mishra of **DIPLOMA (Computer Science & Engineering [ Lateral Entry ])** from **Government Polytechnic Chopan Sonebhadra** (Institute/University) was working on the project entitled “**FavCart**” developed on **"Python with Django"** in Techpile Technology Pvt. Ltd. he was engaged with us during 21th July to 6th Septemberfor a period of **45 days.**

he has done an excellent job during his engagement with the Software Development & Testing Division of the company. he has completed his project during the training tenure. His performance has been good and satisfactory.

I would like to take this opportunity to express my appreciation to SAURABH KUMAR MISHRA for his work and wish his all the very best for his future endeavors.

**Regards,**

**Divya Rai**

**HR MANAGER**

**Techpile Technology Pvt. Ltd.**

**Lucknow (U.P.) Signature**

# PREFACE

Summer training is an important part of the engineering curriculum. The Diploma course summer training helps a student in getting acquainted with the manner in which his knowledge is being practically used outside his institute and this is normally different from what he has learnt from books. Hence, when the student switches from the process of learning to that of implementing his knowledge, he finds an abrupt change. This is exactly why summer training session during the Diploma curriculum becomes all the more important. Summer training is prescribed for the student of Technical College as a part of the three-year degree course of engineering by the AICTE. We are required to undergo summer training for a period of 45 days after the completion of the 2nd year.

This training report describes in detail the training after the 2nd year session, which I completed at the ***Techpile Technology Pvt. Ltd.*** This report also gives the information about the organization and it’s working along with the project undertaken in the training period.

The fundamental step used in **SDLC** process is based on the ISO 9001 guidelines. My aim was to follow the ISO guidelines and develop a perfect system.

The system development was organized into 5 major parts:

1. **Requirement Gathering**
2. **Documentation/Design**
3. **Development**
4. **Coding**
5. **Testing**

**ACKNOWLEDGEMENT**

Apart from my effort, the success of the project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who express have been instrumental in the successful completion of this project.

I would like to express my deep and sincere gratitude to my supervisor Mr. **Rahul Soni** Sir **(**Techpile Technology Pvt. Ltd...), who gave me his full support and encouraged me to work in an innovative and challenging project for educational field. His wide knowledge and logical thinking gave me right direction all the time.

I am deeply grateful my project coordinator for his help and support provided at every step of the project.

Last but not the least, I thank to all employees of **Techpile Technology Pvt. Ltd.** for their support and co-operation.

***SAURABH KUMAR MISHRA***

**DECLARATION**

This is to certify that the project report entitled “**FavCart**” is done by me is an authentic work carried out for the partial fulfillment of the requirements for the award of the Diploma in **“(Computer Science & Engineering [ LATERAL ENTRY ])”** under the guidance of Mr. **Rahul Soni**. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

`

**SAURABH KUMAR MISHRA**

**INDEX**

**PAGE**

1. **Introduction……………………………………………………………………...10-13**
   1. Overview of Organization………………………………………………………….…........10
   2. Introduction…………………………………………………………………………….…..10
   3. Objectives……………………………………………………………………....………...10-11
   4. Project Overview…………………………………………………………………………....11
   5. Project Scope……………………………………………………………….…………….…11
   6. Problem Definition…………………………………………………………...………..…11-13

**2.System Analysis………………………………………………...........………......14-24**

2.1 Objective………………………………………………………………………………….…14

2.2 SDLC Phases………………………………………………………………………..……….14

2.2.1 Preliminary Investigation…………………………………………………………14-16

2.2.2 System Analysis…………………………………………………………………..16

2.2.3 System Design……………………………………………………………………16

2.2.4 Coding……………………………………………………………………………17

2.2.5 Testing………………………………………………………………....................17

2.2.6 Implementation………………………………………………………..................17

2.2.7 Maintenance……………………………………………………………………...18

2.3 Process Description………………………………………………………………………...19

2.4 Project Model Used…………………………………………………………................…19-20

2.5 ER-Diagram………………………………………………………………………...…….21-22

2.6 Data Flow Diagram………………………………………………………………...……..23-24

**3.Software Hardware Requirement Specification……………………..........…..25-26**

3.1 Server-Side Hardware Requirement…………………………………………………………….25

3.2 Server-Side Software Requirement……………………………………………………...……...25

3.3 Client-Side Hardware Requirement……………………………………………………………..26

3.3.1 Software Resource…………………………………………………………………...26

3.4 Support Maintenance……………………………………………………………………..……..26

**4.System Design Approach…………………………………………………….......27-28**

4.1 Top-Down Designing………………………………………………………..……...………...27

4.2 Bottom –Up Designing…………………………………………………….………….............28

4.3 Following Approach……………………………………………………….………….............28

**5.Backend Design……………………………………………………......................29-36**

5.1 Code of view.py……………………………………………………………………………29-34

5.2 Description of Classes and Methods (model.py) ……………………………..................…34-35

5.3 Defined urls (urls.py) ………………………………………………………………………35-36

5.3.1 Project level urls.py……………………………………………………………35-36

5.3.2 App level urls.py…………………………………………………………………36

**6.Data Modeling…………………………………………………………..…….....37-41**

6.1 List of Tables………………………………………………………………,……………….…..37

6.2 Structure of Tables…………………………………………………..………..……………...38-41

**7.Testing……………………………………………………………..……….…….42-46**

**8.Input-Output Forms…………………………………………..…………………47-56**

8.1 Project Screenshot…………………………………………………………………………..47-48

8.2 Project Coding………………………………………………………………………………48-56

**9.Future Scope……………………………………………………………………..…57**

**10.Conclusion………………………………………………………………………....57**

**LIST OF TABLES**

**PAGE**

1. **Group Table…………………………………………………………………… .38**
2. **User Table ……………………………………………………………………...38**
3. **Category Table ………………………………………………………………….39**
4. **Contact Us Table…………………………………………………………….….39**
5. **MainCate Table………………………………………………………...…….….40**
6. **MCartTable…………………………………………………………...…………40**
7. **MOrder Table……………………………………………………….……..….…41**
8. **MyProduct Table…………………………………………………………..…….41**
9. **Register Table……………………………………….………….….………...…..42**

**LIST OF FIGURES**

**PAGE**

**1**. SOFTWARE DEVELOPMENT LIFE CYCLE -----------------------------------------------------18

**2.** DEVELOPMENT PHASES ----------------------------------------------------------------------------20

**3.** E-R DIAGRAM ------------------------------------------------------------------------------------------21-22

**4**. ZERO LEVEL DATA FLOW DIAGRAM ------------------------------------------------------------23-24

**1. INTRODUCTION**

**1.1 Overview of Organization**

Techpile is founded by some young engineers who have mastered the IT sector, whose objective is to achieve the highest position in the IT sector across the country. who are trying to achieve this objective by cooperating in various fields.

Techpile is an organization working in both software development and software training. which aims to make all the specials of its client successful through their coding as well as to make students more competent to work with a well reputed organization.

We are proud of our high-quality standards. These standards allow us to provide our customers with reliable and error-free software applications, regardless of complexity. Our top-notch developers use the latest software methodologies and technologies. This means that they can concentrate on our clients' business goals and keep them involved in every stage through the entire project. Our meticulous approach has helped us build our excellent track record with no failed or aborted projects. We are in the business of change, managing complexity with an unparalleled insight, looking beyond the horizon of IT with resources focused on solutions. Becoming successful is a skill but one cannot perfect it without practice.

**1.2 INTRODUCTION**

Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using an android device. Thus, the customer will get the service of online shopping and home delivery from his favorite shop.

## 1.3 OBJECTIVES:

The objective of the project is to make an application in android platform to purchase items in an existing shop. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project. The web application can be implemented in the form of an android application with web view.

## 1.4 PROJECT OVERVIEW:

The central concept of the application is to allow the customer to shop virtually using the Internet and allow customers to buy the items and articles of their desire from the store. The information pertaining to the products are stores on an RDBMS at the server side (store).

The Server process the customers and the items are shipped to the address submitted by them. The application was designed into two modules first is for the customers who wish to buy the articles. Second is for the storekeepers who maintains and updates the information per training to the articles and those of the customers. The end user of this product is a departmental store where the application is hosted on the web and the administrator maintains the database. The application which is deployed at the customer database, the details of the items are brought forward from the database for the customer view based on the selection through the menu and the database of all the products are updated at the end of each transaction. Data entry into the application can be done through various screens designed for various levels of users. Once the authorized personnel feed the relevant data into the system, several reports could be generated as per the security.

## 1.5 PROJECTSCOPE:

This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24\*7 and a home delivery system which can make customers happy.

**1.6 PROBLEM DEFINITION**

In this section we shall discuss the limitation and drawback of the existing system that forced us to take up this project. Really that work was very typical to manage the daily errors free records and adding or removing any node from server. This problem produces a need to change the existing system. Some of these shortcomings are being discussed below: -

* **Low Functionality**

With the existing system, the biggest problem was the low functionality. The problem faced hampered the work. For small task like adding any new node to server or deleting a node or keeping daily record we have to appoint minimum two or three employee.

* **Erroneous Input and Output**

In the existing system, humans performed all the tasks. As in the human tendency, error is also a possibility. Therefore, the inputs entered by the person who is working in the Company, in the registers may not be absolutely foolproof and may be erroneous. As a result of wrong input, the output reports etc. Will also be wrong which would in turn affect the performance.

* **Portability Problem**

System that existed previously was manual. As a result, the system was less portable. One has to carry the loads of many registers to take the data from one place to another. A big problem was that the system was less flexible and if we wanted to calculate yearly or monthly maintenance report or efficiency report, then it was a big headache.

* **Security-**

Security concerns were also one of the motives of the Company for the need of software. In the registers, the data is not secure as anybody can tamper with the data written in the registers. While in this software, just a password makes it absolutely secure from the reach of unauthorized persons.

* **Data Redundancy**

In the case of manual system, the registers are maintained in which, a lot of data is written.

* **Processing Speed**

In manual system maintaining a register and performing the necessary calculation has proved to be a troublesome job, which takes a lot of time and may affect the performance of the Company. But with this software we can have all the tasks performed in a fraction of second by a single click thus making the troublesome job much easier.

* **Manual Errors**

When a number of tough tasks are prepared by the humans like preparation of reports, performing long calculation then some human errors are obvious due to a number of factors like mental strain, tiredness etc. But as we all know that computer never get tired irrespective of the amount of work it has to do. So, this software can nullify the probability of manual error that improve the performance.

* **Complexity in Work**

In manual system whenever a record is to be updated or to be deleted a lot of cutting and overwriting needs to be done on the registers that are concerned that are deleted or updated record, which makes the work very complex

**2. SYSTEM ANALYSIS**

**2.1 Objective:**

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

**Analysis specifies what the system should do.**

**2.2 SDLC Phases:**

System Development Life Cycle (SDLC) mainly consists of the following 7 phases which can be detailed: -

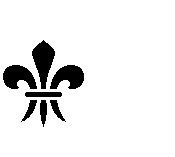
**2.2.1Preliminary Investigation:**

This is the first phase of the system development life cycle. In this phase we tend to find out the needs of the client –what exactly does the client want? Before the development of any system the important point is to know the needs, objectives and scope of the system

* **Feasibility Study**:

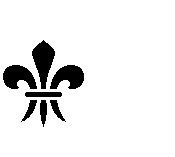
Feasibility study is the step of preliminary study of the system development life cycle. Things are always easy at the beginning in any software process. In fact, nothing is in feasible with unlimited time and resources. But it is not the fact. So, practically we have to do in limited resources in a restricted time margin. So, for the system to be feasible, following points we have to consider.

The feasibility study is conducted to check whether the candidate system is feasible. The system which is selected to be the best against the criteria is there after designed and developed. The feasibility study takes in to consideration, the risks involved in the project development beforehand. Therefore, in this phase we have to do feasibility study which is the test of the website according to its work ability, impact on the organization, ability to meet user need and effective use of resources. We do the feasibility study for website to analyze the risks, costs and benefits relating to economics, technology and user organization. There are several types of feasibility depending on the aspect they cover. Import of these includes:

**Technical Feasibility:**

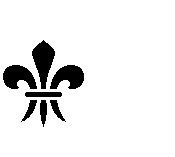
This is an important outcome of preliminary investigation. It comprises of following questions:

* Can the work of project bed one with the current equipment, existing software and available man power resource?
* If Technology is required what are the possibilities that it can be developed?

**Economic Feasibility:**

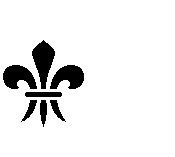
It deals with question related to the economy. It comprises of the following questions: -

* + Are there sufficient benefits in creating the system to make the cost acceptable?
  + Are the costs of not creating the system so great that the project must be undertaken?

**Legal Feasibility:**

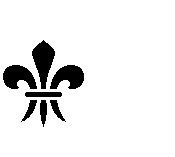
It deals with the question related to the legal issues. It comprises of the following questions: -

* Contract Signing
* Software License agreement
* Issues related to cyber laws.
* Legal issues relating to the man power contract.

**Operational Feasibility:**

The operational feasibility consists of the following activity: -

* Will the system be useful if it is developed & implemented?
* Will there be resistance from employee?

**Social & Behavioral Feasibility:**

It deals with the various issues related to the human behavior like: -

* Whether the user be able to adapt a new change or not?
* Whether the ambiance we are providing suits the user or not?
* **Request Approval:**

Request approval is the preliminary steps of the system development lifecycle. Request approval is the phase in which all the requirements which would be provide in the system are stated. The request approval is a sort of agreement between the client and the company which is building this software. Both the parties should be mutually agreed on the stated requirements.

**2.2.2SystemAnalysis:**

System analysis is the phase following the phase of the request approval. In this phase we tend to analyze the overall system which we have to build. System analysis is the crucial part in SDLC.

**2.2.3System Design:**

System design means the designing of the system. The System can be done in either of the following two ways: -

* + Logical System Design
  + Physical System Design

**2.2.4Coding:**

Coding is the phase in which a developer codes using any programming languages. Coding constitutes only20 % of the whole project and which is easier to write. The coding work is also done in the teams; development of the system is usually done under the modular programming style, which can be either top-down approach or bottom-up approach.

**2.2.5 Testing:**

Testing is the phase in which the system that has been developed is tested. Testing comprises of the 60%ofthe overall development of the system. Testing of the system is important because testing aims to uncover the different errors in the system. There are various different testing techniques that can be used for the testing of the system.

**2.2.6 Implementation:**

Implementation process involved the installation of software on user’s side. Implementation process actually depends on type of a system & various. Opting for suitable conversion approach is a step implementation. The conversion processes are as follows:-

* + - Parallel Conversion
    - Direct Conversion Approach
    - Pilot Conversion Approach
    - Phase In Conversion Approach

**2.2.7Maintenance**:

Merely developing the system is not important but also maintenance is important. The company that has built the system provides for some time free of cost maintenance to the client and after that period it is usually a paid service.

s

**2.3 Process Description**

Gantt charts mainly used to allocate resources to activities. The resources allocated to activities include staff, hardware, and software. Gantt charts (named after its developer Henry Gantt) are useful for resource planning. A Gantt chart is special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The length of each bar is proportional to the duration of the time planned for the corresponding activity.

Gantt chart is a project scheduling technique. Progress can be represented easily in a Gantt chart, by coloring each milestone when completed. The project will start in the month of January and end after 4 months at the beginning of April.

**2.4 PROJECTMODEL USED**

**Iterative Enhancement Model**

* This model has the same phases as the waterfall model, but with fewer restrictions. Generally, the phases occur in the same order as in the waterfall model, but they may be conducted in several cycles.
* Useable product is released at the end of the each cycle, with each release providing additional functionality. Customers and developers specify as many requirements as possible and prepare a SRS document. Developers and customers then prioritize these requirements. Developers implement the specified requirements in one or more cycles of design, implementation and test based on the defined priorities.

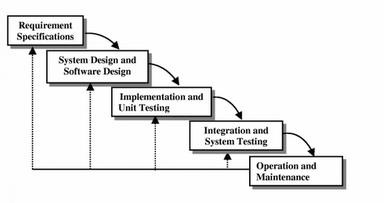
The procedure itself consists of the initialization step, the iteration step, and the Project Control List. The initialization step creates a base version of the system. The goal for this initial implementation is to create a product to which the user can react. It should offer a sampling of the key aspects of the problem and provide a solution that is simple enough to understand and implement easily. To guide the iteration process, a project control list is created that contains a record of all tasks that need to be performed. It includes such items as new features to be implemented and areas of redesign of the existing solution. The control list is constantly being revised as a result of the analysis phase.

The iteration involves the redesign and implementation of iteration is to be simple, straightforward, and modular, supporting redesign at that stage or as a task added to the project control list. The level of design detail is not dictated by the iterative approach. In a light-weight iterative project the code may represent the major source of [documentation](http://en.wikipedia.org/wiki/Software_documentation) of the system; however, in a critical iterative project a formal [Software Design Document](http://en.wikipedia.org/wiki/Software_Design_Document) may be used. The analysis of an iteration is based upon user feedback, and the program analysis facilities available. It involves analysis of the structure, modularity, [usability](http://en.wikipedia.org/wiki/Usability), reliability, efficiency, & achievement of goals. The project control list is modified in light of the analysis results.

**PHASES:**

Incremental development slices the system functionality into increments (portions). In each increment, a slice of functionality is delivered through cross- discipline work, from the requirements to the deployment. The unified process groups increments/iterations into phases: inception, elaboration, construction, and transition.

* Inception identifies project scope, requirements (functional and non-functional) and risks at a high level but in enough detail that work can be estimated.
* Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.
* Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.
* Transition delivers the system into the production operating environment.



**2.5 ER-Diagram**

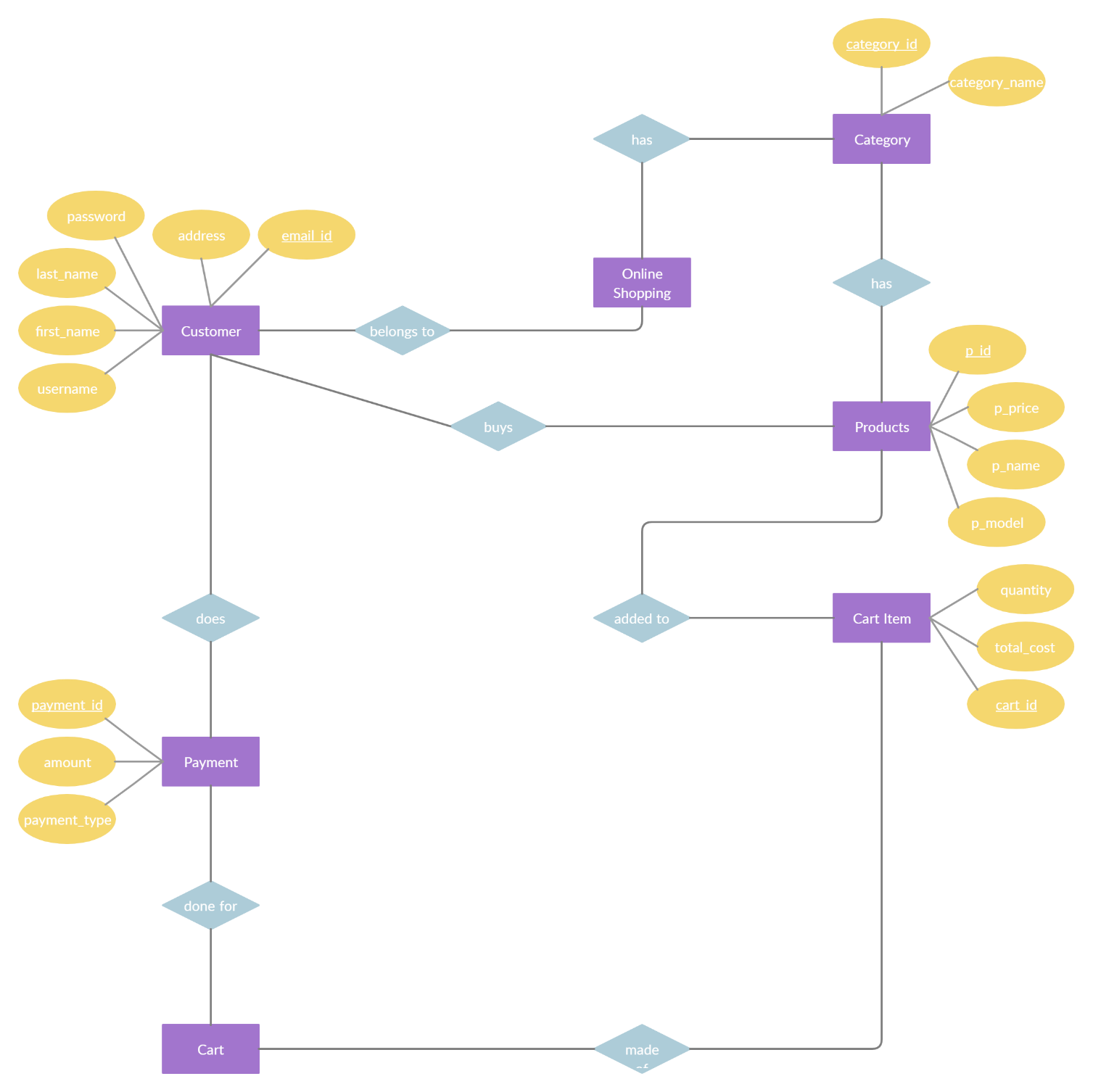
**Introduction:**

I[n software engineering, an](http://en.wikipedia.org/wiki/Software_engineering) entity-relationship model (ERM) is an abstract and conceptual representation of [data.](http://en.wikipedia.org/wiki/Data) Entity-relationship modeling is a [database modeling](http://en.wikipedia.org/wiki/Database_model) [method,](http://en.wikipedia.org/wiki/Database_model) used to produce a type [of conceptual schema](http://en.wikipedia.org/wiki/Conceptual_schema) o[r semantic data model of](http://en.wikipedia.org/wiki/Semantic_data_model) a system, often [a relational database, a](http://en.wikipedia.org/wiki/Relational_database)nd its require[ments in a top-down](http://en.wikipedia.org/wiki/Top-down) fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs. ER Diagrams depicts relationship between data objects. The attribute of each data objects noted in the entity-relationship diagram can be described using a data object description. Entity relationship diagram is very basic, conceptual model of data and it is fundamental to the physical database design. This analysis is then used to organize data as relations, normalizing relations, and obtaining a Relational database.

The entity-relationship model for data uses three features to describe data. These are:

1. Entities which specify distinct real-world items in an application.
2. Relationship, which connect entities and represent meaningful dependencies between them.
3. Attributes which specify properties of entities & relationships.

**E-R Diagram**



**2.6 Data Flow Diagram**

**Introduction:**

DFD is an acronym for the word Data Flow Diagram. DFD is ppictorial representation of the system. DFD is a graphical representation of the ―flow of data through the information system. DFD are also used for the visualization of data processing (structured design). ADFD provides no information about the timings of the process, or about whether process will operate in parallel or sequence. DFD is an important technique for modeling system’s high-level detail by showing how input data is transformed to output results through a sequence of functional transformations. DFD reveal relationships among between the various components in a program or system. The strength of DFD lies in the fact that using few symbols we are able to express program design in an easier manner. ADFD can be used to represent the following:-

* External Entity sending and receiving data. Process that changes the data.
* Flow of data within the system. Data Storage locations.

**Uses of DFD:**

The main uses of data flow diagrams are as follows: -

DFD is a method of choice for representation of showing f information through a system because of the following reasons:-

* + DFDs are easier to understand: - by technical and non-technical audiences.
  + DFDs can provide a high-level system overview, complete with boundaries and connections to other system.
  + DFDs can provide a detailed representation of system components.

1. **Level DFD**

Request

Online Shopping

USER

ADMIN

Request for Registration

Response

Response

Request for Login

1. **Level DFD**

Generate

Shopping Report

shopping

Management

Shopping Cart

Management

Generate

Shopping Cart Report

Online

Shopping

system

Product

Management

Generate

Product Cart

Shipment

Management

System User

Management

Login

Management

Generate

Shipment Report

Generate System

User Report

Check user login details

**3. SOFTWARE HARDWARE REQUIREMENT SPECIFICATION**

A requirements specification for a software system is a complete description of the behavior of a system to be developed and it includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements.

Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).Requirements are a sub-field of software engineering that deals with the elicitation, analysis, specification, and validation of requirements for software.

The software requirement specification document enlists all necessary requirements for project development. To derive the requirements, we need to have clear and thorough understanding of the products to be developed. This is prepared after detailed communications with project team and the customer.

**3.1 SERVER-SIDE HARDWARE REQUIREMENT:**

 AMD Athlon 64 with processor speed 2.8 or more

256 DDR Ram

  40 GB Hard disks

 Network Interface card

IIS

CD-Drive

**3.2 SERVER-SIDE SOFTWARE REQUIREMENT:**

* Windows
* Python 3.7 and PyCharm IDE 2021.3
* SQLite

**CLIENT-SIDE HARDWARE REQUIREMENT:**

* Processor Dual core-based computer
* 2 GB Minimum RAM
* 20 GB HDD
* 100 Mbps LAN
* Web Browser

**TO DEVELOP THIS PROJECT THE VARIOUS SOFTWARE RESOURCES ARE USED.**

* Front End                - HTML-CSS &Bootstrap
* Back End                - SQLite
* Web Server              -  Apache Server
* Technology               - Python technology
* Code-Behind Language   - Python
* IDE                                 - PyCharm

**3.3 SUPPORT AND MAINTENANCE: -**

One-year free support for rectifying system bugs including front end and beck end will be provided. During warranty period Software Engineers will be responsible for removing bugs and improving it. After one year support can be extended @ 20% of the total product deployment cost.

**4. SYSTEM DESIGN APPROACH**

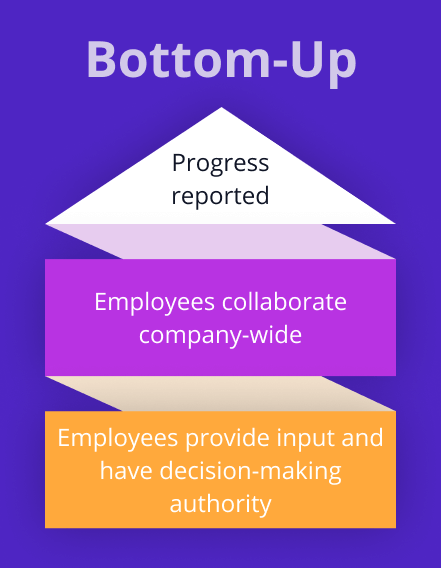
**4.1 Top – Down designing:**

The top - down designing approach started with major components of the system. It is a stepwise refinement which starts from an abstract design, in each steps the design is refined two or more concrete levels until we reach a level where no – more refinement is possible or not needed.



**4.2 Bottom – Up designing:**

In bottom – up designing the most basic and primitive components are designed first, and we proceed to higher level components. We work with layers of abstractions and abstraction are implemented until the stage is reached where the operations supported by the layer is complete.



**4.3 Following Approach:**

In this project we are following **Mixed Approach** i.e. A combination of top – down and bottom – up. We are developing some of the components using top – down designing approach (e.g., the Web pages) and some components in bottom – up designing approach (e.g., the middle tier classes).

**5. BACKEND DESIGN**

**5.1 Description of Classes and Methods (model.py)**

**5.1.1 All code of views.py**

from django.shortcuts import render  
from .models import \*  
from django.http import HttpResponse  
from datetime import datetime  
from django.db import connection  
  
*# Create your views here.*def index(request):  
 user=request.session.get('userid')  
 ct=""  
 if user:  
 ct=mcart.objects.all().filter(userid=user).count()  
 *#request.session['cart']=ct* x=category.objects.all().order\_by('-id')[0:6]  
 pdata=myproduct.objects.all().order\_by('-id')[0:7]  
 md={"data":pdata}  
 mydict={"data":x,"prodata":pdata , "cart":ct}  
 return render(request,'user/index.html',context=mydict)  
  
def about(request):  
 user = request.session.get('userid')  
 ct = ""  
 if user:  
 ct = mcart.objects.all().filter(userid=user).count()  
 *# request.session['cart']=ct* return render(request,'user/aboutus.html',{"cart":ct })  
  
def product(request):  
 user = request.session.get('userid')  
 ct = ""  
 if user:  
 ct = mcart.objects.all().filter(userid=user).count()  
 *# request.session['cart']=ct* return render(request,'user/product.html',{"cart":ct })  
  
def myorder(request):  
 user=request.session.get('userid')  
 oid=request.GET.get('oid')  
 if user:  
 if oid is not None:  
 morder.objects.all().filter(id=oid).delete()  
 return HttpResponse("<script>alert('your order has been cancelled');location.href='/user/myorder'</script>")  
 cursor=connection.cursor()  
 cursor.execute("select p.\*,o.\* from user\_myproduct p , user\_morder o where p.id=o.pid and o.userid='"+str(user)+"'and o.remarks='Pending'")  
 pdata=cursor.fetchall()  
 cursor.execute("select p.\*,o.\* from user\_myproduct p , user\_morder o where p.id=o.pid and o.userid='" + str(  
 user) + "'and o.remarks='delivered'")  
 ddata = cursor.fetchall()  
 mydict={"pdata":pdata , "ddata":ddata}  
 return render(request,'user/myorder.html',mydict)  
  
def myordr(request):  
 user=request.session.get('userid')  
 pid=request.GET.get('msg')  
 if user:  
 if pid is not None:  
 morder(userid=user,pid=pid,remarks="Pending",odate=datetime.now().date(), status=True).save()  
 return HttpResponse("<script>alert('your order confirmed');location.href='/user/viewproduct'</script>")  
 else:  
 return HttpResponse("<script>alert('you have to login first');location.href='/user/signin'</script>")  
 return render(request,'user/myordr.html')  
  
def mycart(request):  
 p=request.GET.get('pid')  
 user=request.session.get('userid')  
 if user:  
 if p is not None:  
 mcart(userid=user,pid=p,cdate=datetime.now().date(),status=True).save()  
 return HttpResponse('<script>alert("your item is added cart");location.href="/user/index/"</script>')  
 else:  
 return HttpResponse("<script>alert('you have to login first');location.href='/user/signin'</script>")  
  
 return render(request,'user/mcart.html')  
  
def enquiry(request):  
 status=False  
 if request.method=='POST':  
 a=request.POST.get('name')  
 b=request.POST.get('email')  
 c=request.POST.get('mob')  
 d=request.POST.get('msg')  
 contactus(Name=a,Email=b,Mobile=c,Message=d).save()  
 status=True  
 *#mdict={"Name":a,"Email":b,"Mobile":c,"Message":d }* msg={"m":status}  
 return render(request,'user/enquiry.html' ,context=msg)  
  
def signup(request):  
 user = request.session.get('userid')  
 ct = ""  
 if user:  
 ct = mcart.objects.all().filter(userid=user).count()  
 *# request.session['cart']=ct* if request.method=='POST':  
 a=request.POST.get('name')  
 b=request.POST.get('email')  
 c=request.POST.get('passwd')  
 d=request.POST.get('mob')  
 e=request.POST.get('address')  
 f=request.FILES.get('pic')  
 x=register.objects.all().filter(email=b).count()  
 if x==0:  
 register(name=a,email=b,mobile=d,passwd=c,address=e,ppic=f).save()  
 return HttpResponse("<script>alert('you are registered successfully ...');location.href='/user/signup'</script>")  
 else:  
 return HttpResponse("<script>alert('your email id is already registered ...');location.href='/user/signup'</script>")  
  
 return render(request,'user/signup.html',{"cart":ct })  
  
def myprofile(request):  
  
  
 user = request.session.get('userid')  
 ct = ""  
 if user:  
 ct = mcart.objects.all().filter(userid=user).count()  
 *# request.session['cart']=ct* user = request.session.get('userid')  
 if user:  
 if request.method=="POST":  
 a = request.POST.get('name')  
  
 c = request.POST.get('passwd')  
 d = request.POST.get('mob')  
 e = request.POST.get('address')  
 f = request.FILES.get('pic')  
 register(name=a,mobile=d,email=user,ppic=f,address=e,passwd=c).save()  
 return HttpResponse("<script> alert('your profile updated successfullt');location.href='/user/myprofile'</script>")  
  
 x = register.objects.all().filter(email=user)  
 d = {"mdata": x, "cart": ct}  
  
 return render(request,'user/myprofile.html', d)  
  
def signin(request):  
 if request.method=="POST":  
 email=request.POST.get('email')  
 Passwd=request.POST.get('passwd')  
 x= register.objects.all().filter(email=email,passwd=Passwd).count()  
 y=register.objects.all().filter(email=email,passwd=Passwd)  
 if x==1:  
 request.session["userid"]=email  
 request.session["userpic"]=str(y[0].ppic)  
  
 return HttpResponse("<script>alert('login succesfuly');location.href='/user/index/'</script>")  
 else:  
 return HttpResponse("<script>alert('your userid or password incorrect');location.href='/user/signin'</script>")  
  
 return render(request,'user/signin.html')  
  
def signout(request):  
 if request.session.get('userid'):  
 del request.session['userid']  
 return HttpResponse("<script>alert('signout successfuly');location.href='/user/index'</script>")  
  
def mens(request):  
 user = request.session.get('userid')  
 ct = ""  
 if user:  
 ct = mcart.objects.all().filter(userid=user).count()  
 *# request.session['cart']=ct* cid=request.GET.get('msg')  
 cat=category.objects.all().order\_by('-id')  
 d=myproduct.objects.all().filter(mcategory=1)  
 if cid is not None:  
 d=myproduct.objects.all().filter(mcategory=1,pcategory=cid)  
 mydict={"cats":cat,"data":d,"a":cid ,"cart":ct }  
 return render(request,'user/mens.html',mydict)  
  
def womens(request):  
 user = request.session.get('userid')  
 ct = ""  
 if user:  
 ct = mcart.objects.all().filter(userid=user).count()  
 *# request.session['cart']=ct* cid = request.GET.get('msg')  
 cat = category.objects.all().order\_by('-id')  
 d = myproduct.objects.all().filter(mcategory=2)  
 if cid is not None:  
 d = myproduct.objects.all().filter(mcategory=2, pcategory=cid)  
 mydict = {"cats": cat, "data": d, "a": cid ,"cart":ct }  
 return render(request,'user/womens.html',mydict)  
  
def kids(request):  
 user = request.session.get('userid')  
 ct = ""  
 if user:  
 ct = mcart.objects.all().filter(userid=user).count()  
 *# request.session['cart']=ct* cid = request.GET.get('msg')  
 cat = category.objects.all().order\_by('-id')  
 d = myproduct.objects.all().filter(mcategory=3)  
 if cid is not None:  
 d = myproduct.objects.all().filter(mcategory=3, pcategory=cid)  
 mydict = {"cats": cat, "data": d, "a": cid ,"cart":ct }  
 return render(request,'user/kids.html',mydict)  
def viewproduct(request):  
 user = request.session.get('userid')  
 ct = ""  
 if user:  
 ct = mcart.objects.all().filter(userid=user).count()  
 *# request.session['cart']=ct* a=request.GET.get('msg')  
 x=myproduct.objects.all().filter(id=a)  
 return render(request,'user/viewproduct.html',{"pdata":x , "msg":a , "cart":ct } )  
  
def showcart(request):  
 user=request.session.get('userid')  
 md={}  
 a=request.GET.get('msg')  
 cid=request.GET.get('cid')  
 pid=request.GET.get('pid')  
  
 if user:  
 if a is not None:  
 mcart.objects.all().filter(id=a).delete()  
 return HttpResponse("<script>alert('your item is deleted from cart...');location.href='/user/showcart'</script>")  
 elif pid is not None:  
 mcart.objects.all().filter(id=cid).delete()  
 morder(userid=user,pid=pid,remarks="Pending",status=True,odate=datetime.now().date()).save()  
 return HttpResponse("<script>alert('your order has been placed successfully');location.href='/user/myorder'</script>")  
  
  
 cursor=connection.cursor()  
 cursor.execute("select p.\*,c.\* from user\_myproduct p , user\_mcart c where p.id=c.pid and c.userid='"+str(user)+"' ")  
 cdata=cursor.fetchall()  
 md={"cdata":cdata}  
  
  
 return render(request,"user/showcart.html",md)  
  
def cpdetail(request):  
 c= request.GET.get('cid')  
 p=myproduct.objects.all().filter(pcategory=c)  
 return render(request,'user/cpdetail.html',{"pdata":p})

**5.1.2 *Description of Classes and Methods (model.py)***

from django.db import models

*# Create your models here.*class contactus(models.Model):  
 Name=models.CharField(max\_length=100)  
 Mobile=models.CharField(max\_length=50)  
 Email=models.CharField(max\_length=50)  
 Message=models.TextField()  
 def \_\_str\_\_(self):  
 return self.Name  
class category(models.Model):  
 Name=models.CharField(max\_length=40)  
 CPic=models.ImageField(upload\_to='static/category/',null=True)  
 def \_\_str\_\_(self):  
 return self.Name  
class maincate(models.Model):  
 Name=models.CharField(max\_length=20)  
 picture=models.ImageField(upload\_to='static/mcategory/',null=True)  
 cdate=models.DateField()  
 def \_\_str\_\_(self):  
 return self.Name  
  
class myproduct(models.Model):  
 pprice=models.FloatField()  
 dprice=models.FloatField()  
 psize=models.CharField(max\_length=20)  
 pcolor=models.CharField(max\_length=40)  
 pdes=models.TextField()  
 pdel=models.CharField(max\_length=60)  
 ppic=models.ImageField(upload\_to='static/product/',default="")  
 pdate=models.DateField()  
 pcategory=models.ForeignKey(category,on\_delete=models.CASCADE ,null=True)  
 mcategory=models.ForeignKey(maincate,on\_delete=models.CASCADE,null=True)  
  
class register(models.Model):  
 name=models.CharField(max\_length=100)  
 email=models.CharField(max\_length=100,primary\_key=True)  
 mobile=models.CharField(max\_length=30)  
 ppic=models.ImageField(upload\_to='static/profile/',null=True)  
 passwd=models.CharField(max\_length=60)  
 address=models.TextField()  
  
  
class morder(models.Model):  
 userid=models.CharField(max\_length=70)  
 pid=models.IntegerField()  
 remarks=models.CharField(max\_length=20)  
 odate=models.DateField()  
 status=models.BooleanField()

class mcart(models.Model):  
 userid=models.CharField(max\_length=70)  
 pid=models.IntegerField()  
 cdate=models.DateField()  
 status=models.BooleanField()

**5.2 Defined urls (urls.py)**

**5.2.1 Project Level – URL.PY**

*"""MyProject URL Configuration  
  
The `urlpatterns` list routes URLs to views. For more information please see:  
 https://docs.djangoproject.com/en/3.2/topics/http/urls/  
Examples:  
Function views  
 1. Add an import: from my\_app import views  
 2. Add a URL to urlpatterns: path('', views.home, name='home')  
Class-based views  
 1. Add an import: from other\_app.views import Home  
 2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')  
Including another URLconf  
 1. Import the include() function: from django.urls import include, path  
 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))  
"""*from django.contrib import admin  
from django.urls import path,include  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('user/',include('user.urls')),  
 path('',include('user.urls')),  
]

**5.2.2 APP LEVEL- URLS.PY**

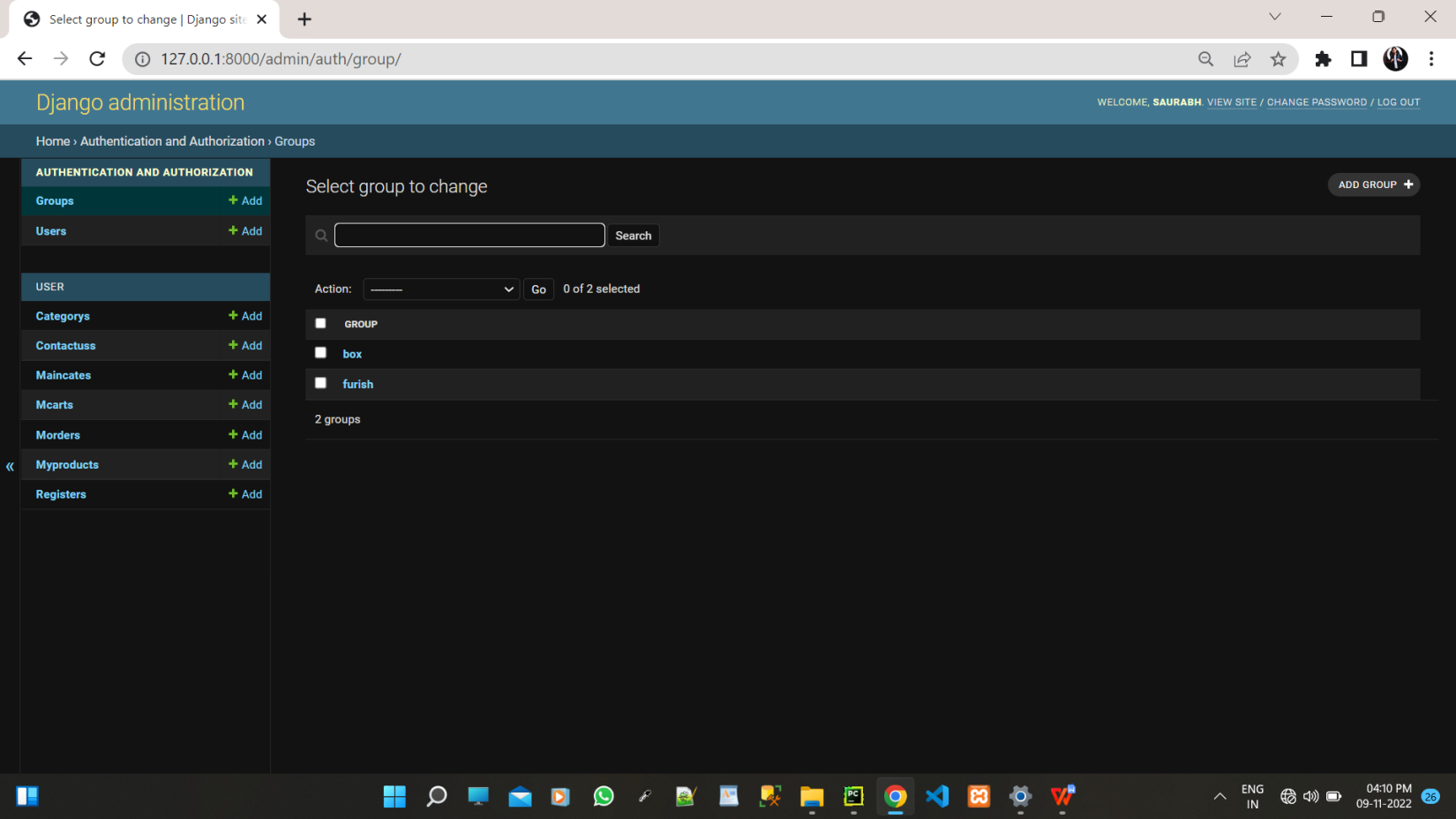
from django.urls import path  
from .import views  
urlpatterns=[  
 path('index/',views.index),  
 path('',views.index),  
 path('home',views.index),  
 path('about',views.about),  
 path('product',views.product),  
 path('myorder',views.myorder),  
 path('myordr',views.myordr),  
 path('mcart',views.mycart),  
 path('showcart',views.showcart),  
 path('cpdetail',views.cpdetail),  
 path('enquiry',views.enquiry),  
 path('signup',views.signup),  
 path('signin',views.signin),  
 path('signout',views.signout),  
 path('myprofile',views.myprofile),  
  
 path('mens',views.mens),  
 path('womens',views.womens),  
 path('kids',views.kids),  
 path('viewproduct',views.viewproduct),  
]

**6. DATA MODELING**

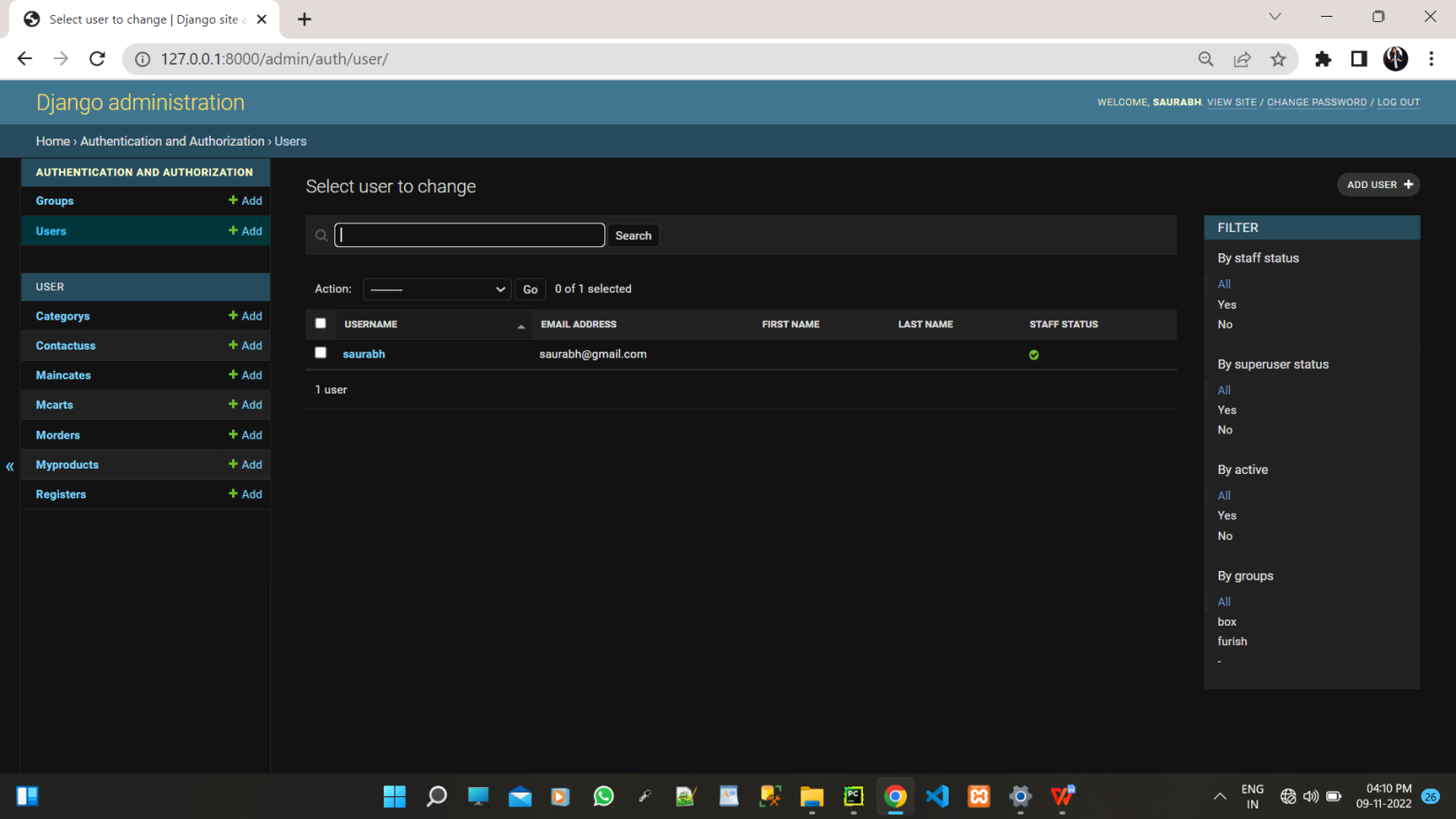
**6. 1 LIST OF TABLES:**

**6.1.1 Group Table**

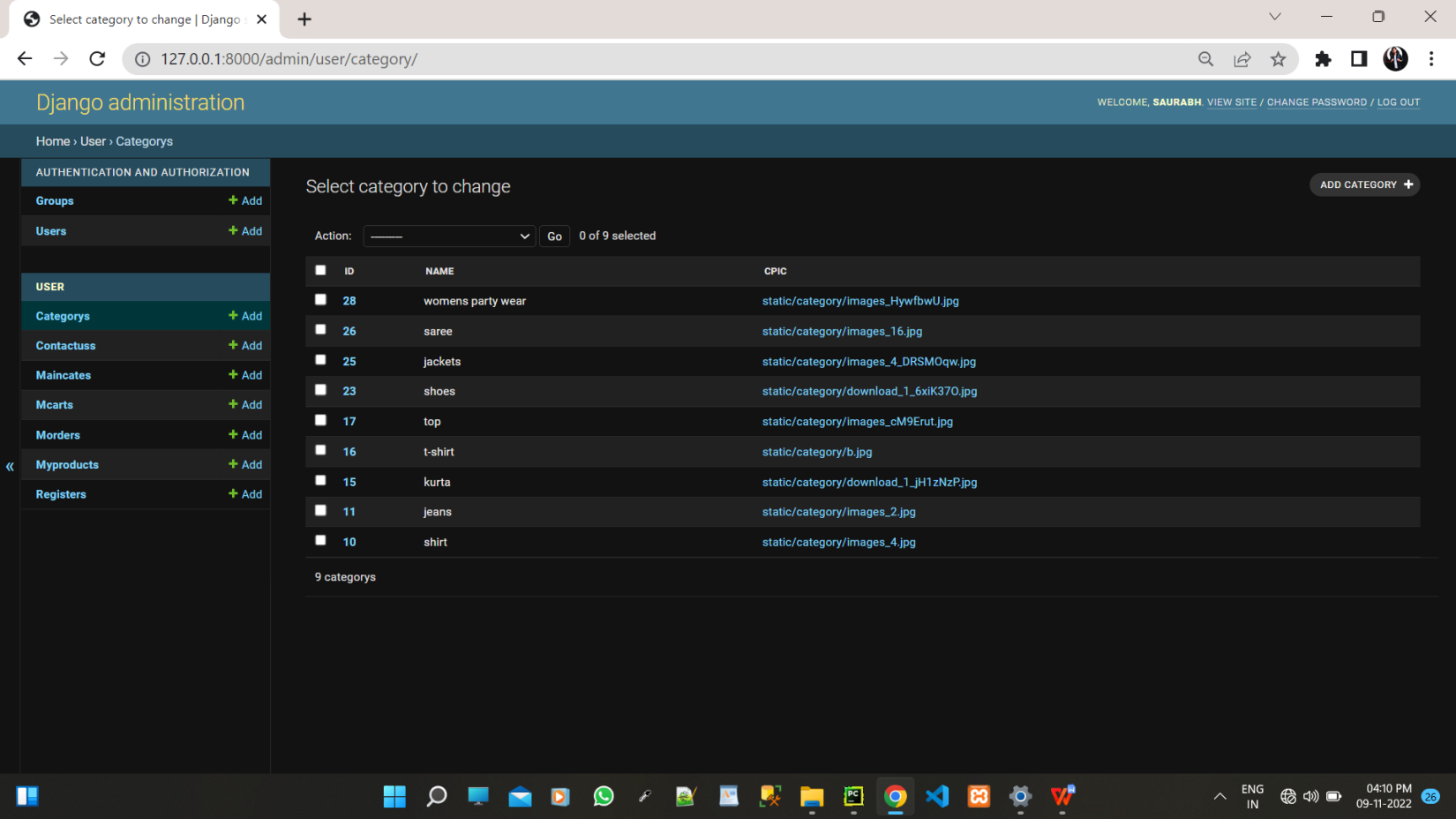
* + 1. **User Table**
    2. **Category Table**
    3. **Contact Us Table**
    4. **MainCate Table**
    5. **MCart Table**
    6. **MOrder Table**
    7. **MyProduct Table**
    8. **Register Table**
  1. **STRUCTURE OF TABLES:**
     1. **Group Table**

****

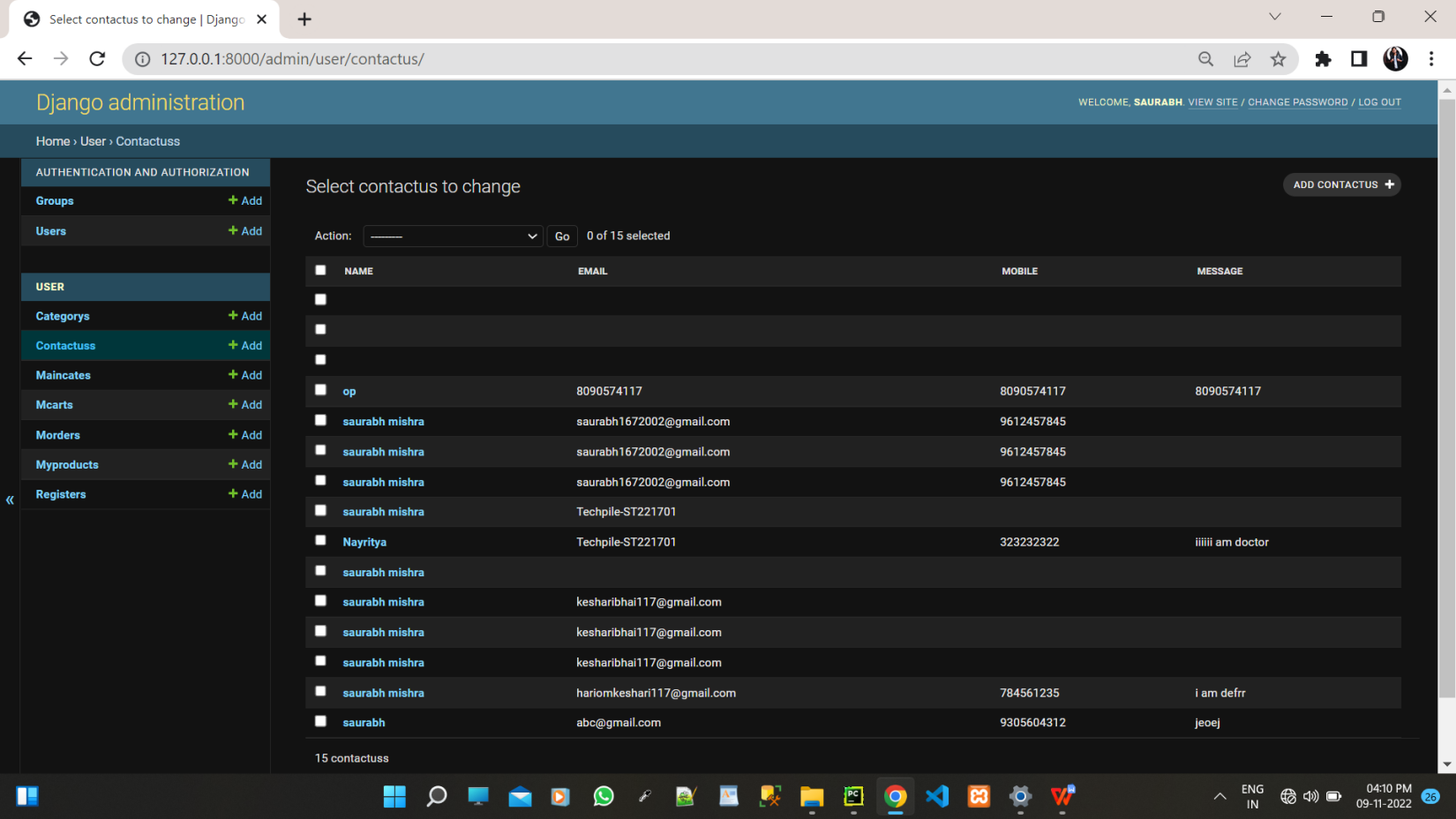
**6.2.2 User Table**

****

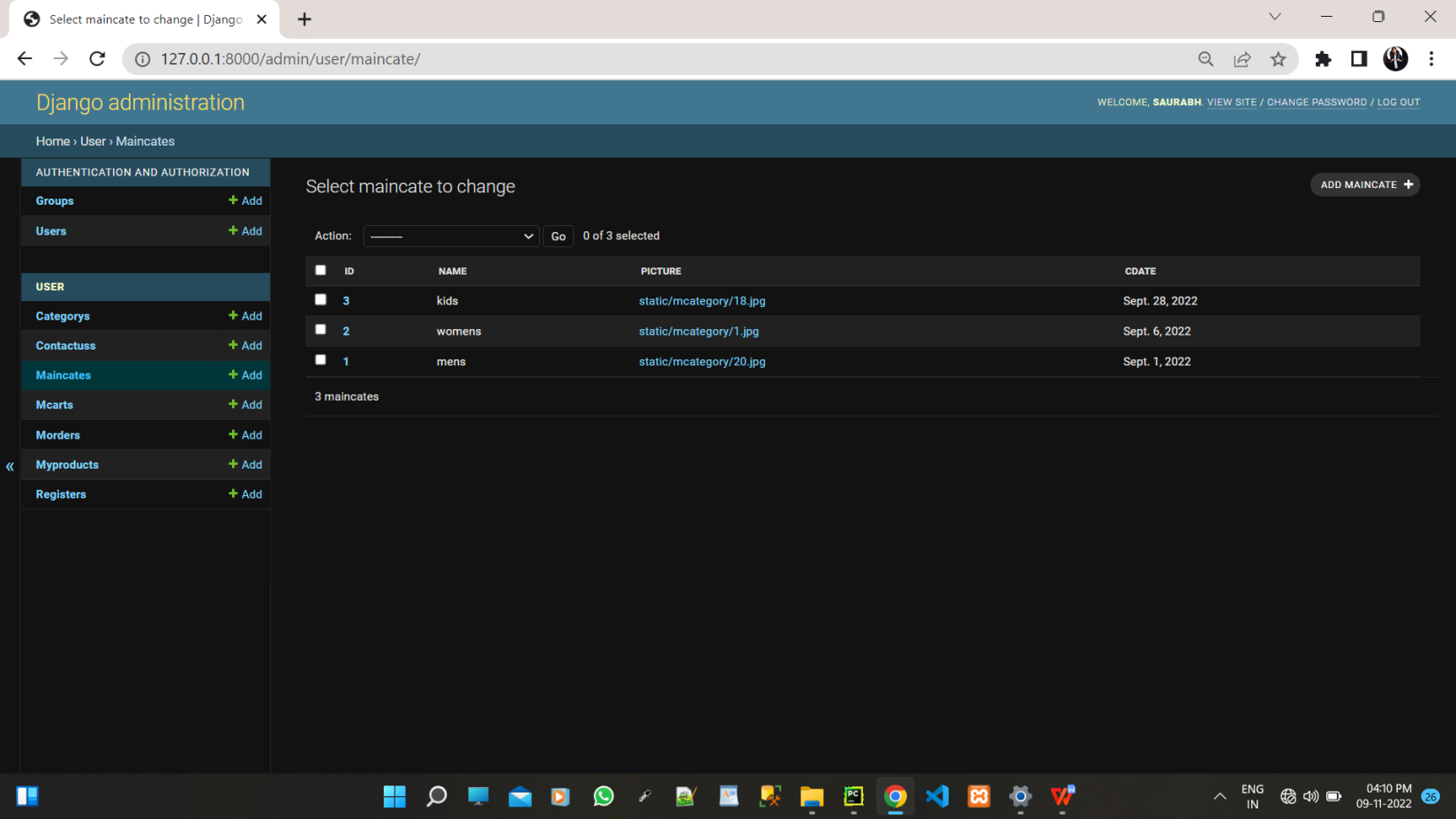
* + 1. **Category Table**

****

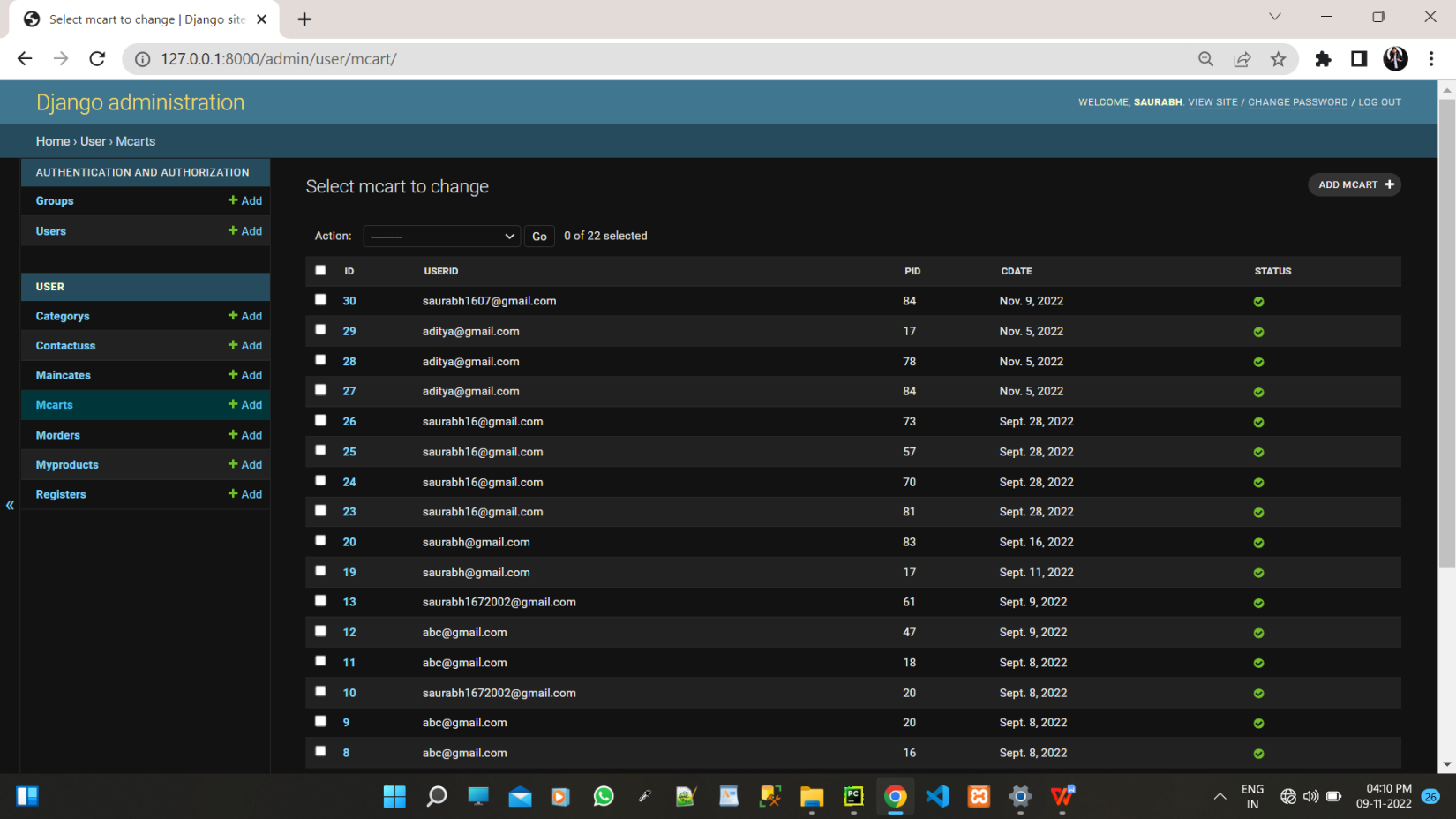
* + 1. **Contact Us Table**

****

* + 1. **MainCate Table**

****

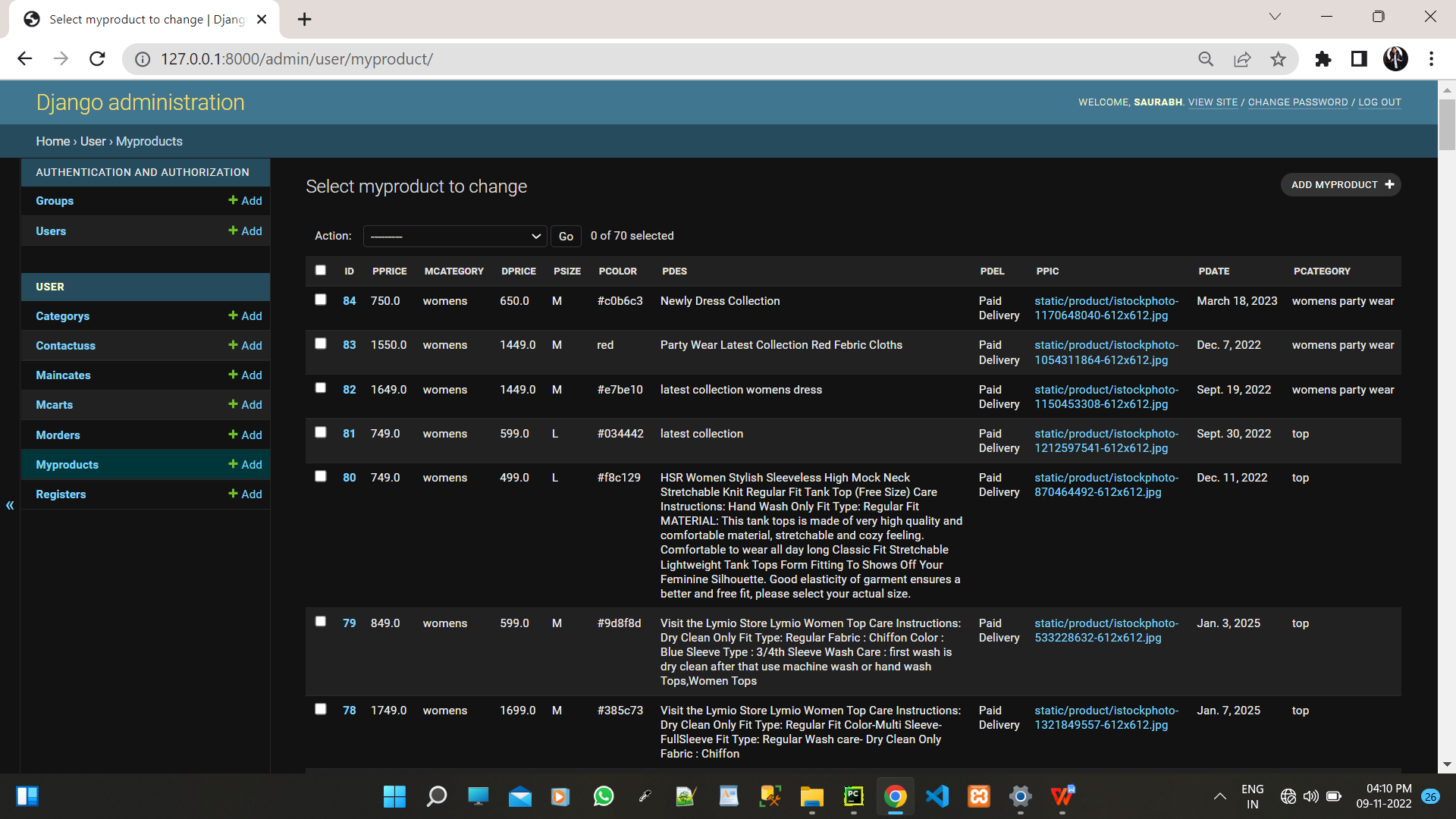
* + 1. **MCart Table**

****

* + 1. **MOrder Table**

****

* + 1. **MyProduct Table**

****

* + 1. **Register Table**

****

1. **TESTING**

Testing is the integral part of any System Development Life Cycle insufficient and interested application tends to crash and result in loss of economic and manpower investment besides user’s dissatisfaction and downfall of reputation.

“Software Testing can be looked upon as one among much process, an organization performs, and that provides the last opportunity to correct any flaws in the developed system. Software Testing includes selecting test data that have more probability of giving errors.” The first step in System testing is to develop the plan that all aspect of system. Complements, Correctness, Reliability and Maintainability.

Software is to be tested for the best quality assurance, an assurance that system meets the specification and requirement for its intended use and performance.

System Testing is the most useful practical process of executing the program with the implicit intention of finding errors that makes the program fail.

**Types of Testing:**

**Black Box (Functional) Testing:**

Testing against specification of system or component. Study it by examining its inputs and related outputs. Key is to devise inputs that have a higher likelihood of causing outputs that reveal the presence of defects. Use experience and knowledge of domain to identify such test cases. Failing this a systematic approach may be necessary. Equivalence partitioning is where the input to a program falls into a number of classes, e.g., positive numbers vs. negative numbers. Programs normally behave the same way for each member of a class. Partitions exist for both input and output. Partitions may be discrete or overlap.

Invalid data (i.e., outside the normal partitions) is one or more partitions that should be tested.

Internal System design is not considered in this type of testing. Tests are based on requirements and functionality.

This type of test case design method focuses on the functional requirements of the software, ignoring the control structure of the program. Black box testing attempts to find errors in the following categories:

* + Incorrect or missing functions.
  + Interface errors.
  + Errors in data structures or external database access.
  + Performance errors.
  + Initialization and termination errors.

**White Box (Structural) Testing:**

Testing based on knowledge of structure of component (e.g., by looking at source code). Advantage is that structure of code can be used to find out how many test case need to be performed. Knowledge of the algorithm (examination of the code) can be used to identify the equivalence partitions. Path testing is where the tester aims to exercise every independent execution path through the component. All conditional statements tested for both true and false cases. If a unit has n control statements, there will be up to 2n possible paths through it. This demonstrates that it is much easier to test small program units than large ones. Flow graphs are a pictorial representation of the paths of control through a program (ignoring assignments, procedure calls and I/O statements).

Use flow graph to design test cases that execute each path. Static tools may be used to make this easier in programs that have a complex branching structure. Tools support. Dynamic program analyzers instrument a program with additional code. Typically, this will count how many times each statement is executed. At end print out report showing which statements have and have not been executed. Problems with flow graph derived testing:

* Data complexity could not take into account.
* We cannot test all paths in combination.
* In really only possible at unit and module testing stages because beyond that complexity is too high.

This testing is based on knowledge of the internal logic of an application’s code. Also known as a Glass Box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

**Unit Testing:**

Unit testing concentrates on each unit of the software as implemented in the code. This is done to check syntax and logical errors in programs. At this stage, the test focuses on each module individually, assuring that it functions properly as a unit. In our case, we used extensive white-box testing at the unit testing stage.

A developer and his team typically do the unit testing do the unit testing is done in parallel with coding; it includes testing each function and procedure.

**Incremental Integration Testing:**

Bottom-up approach for testing i.e., continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately done by programmers or by testers.

**Integration Testing:**

Testing of integration modules to verify combined functionality after integration

.Modules are typically code modules, individual applications, client and server and distributed systems.

**Functional Testing:**

This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black box type testing geared to functionality requirements of an application.

**System Testing:**

Entire system is tested as per the requirements. Black box type test that is based on overall requirement specifications covers all combined parts of a system.

**End-to-End Testing:**

Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with hardware, applications, or system if appropriate.

**Regression Testing:**

Testing the application as a whole for the modification in any module or functionality. Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

**Acceptance Testing:**

Normally this type of testing is done to verify if system meets the customer specified requirements. User or customers do this testing to determine whether to accept application.

**Performance Testing:**

Term often used interchangeably with “stress” and “load” testing, To check whether system meets performance requirements, used different performance and load tools to do this.

**Alpha Testing:**

In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing.

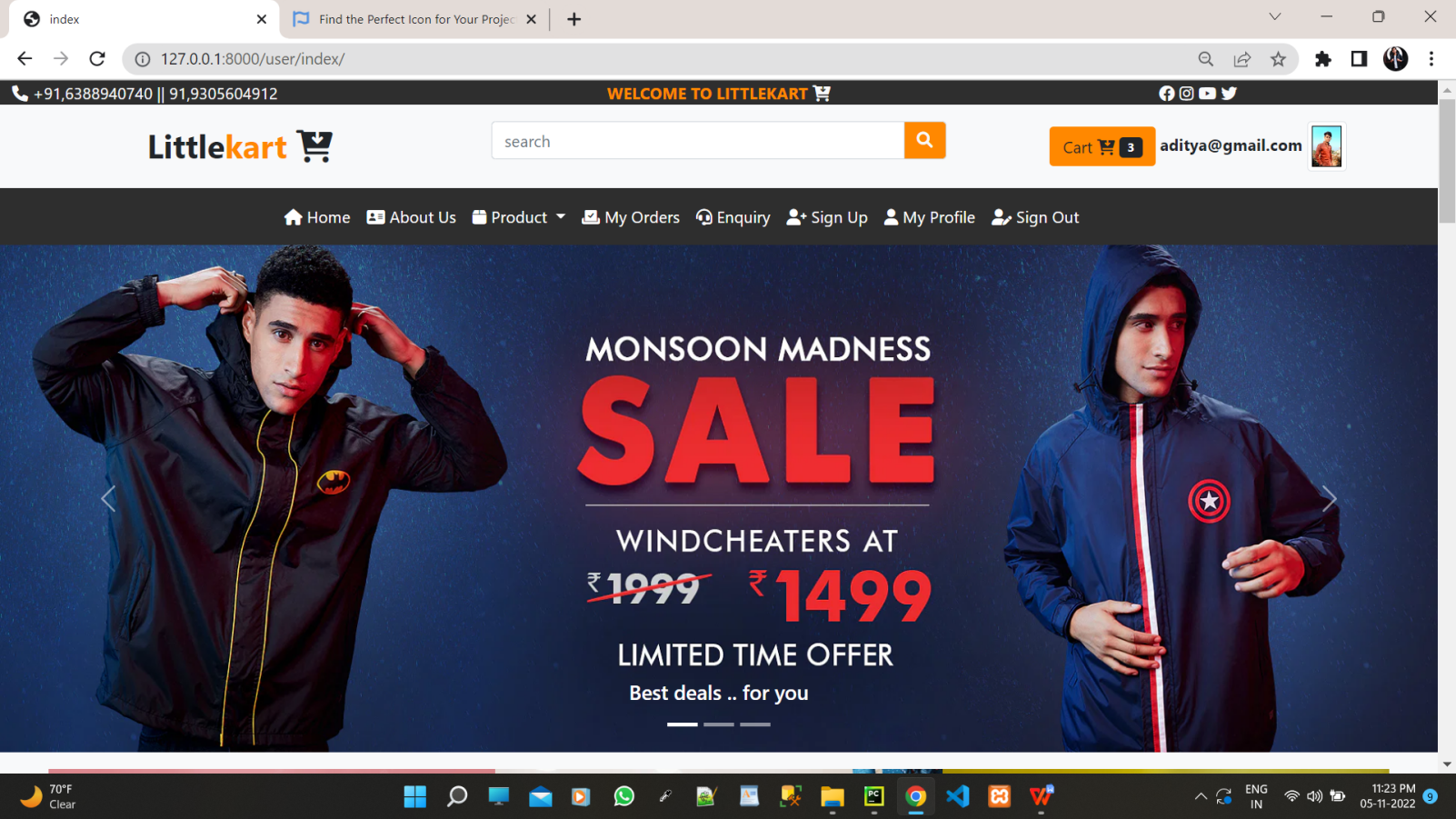
**Beta Testing:**

Testing typically done by end-users or others. This is final testing before releasing application for commercial purpose.

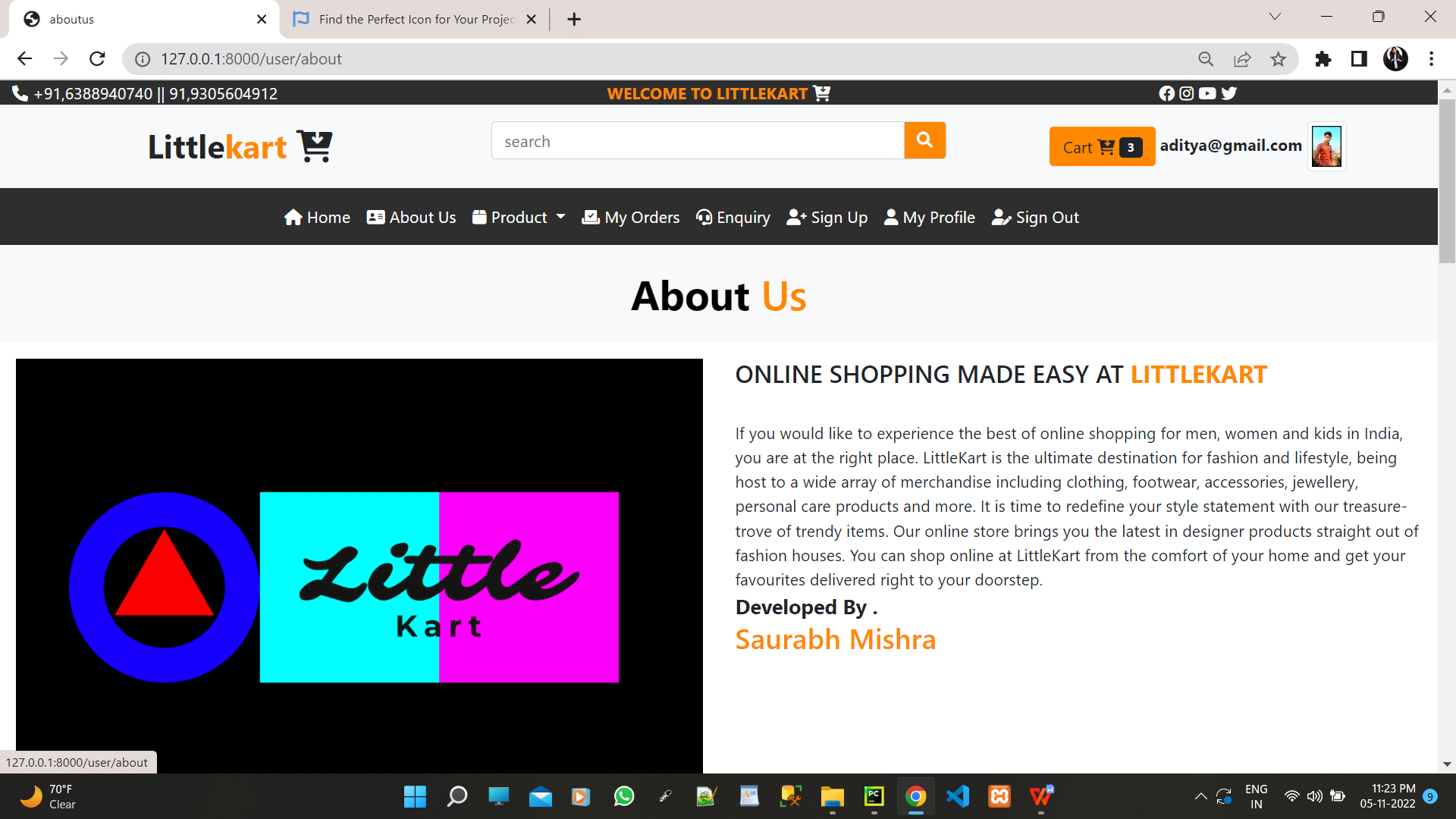
**8. Input-output Forms(SCREENSHOTS AND CODING)**

**8.1 project Screenshot**

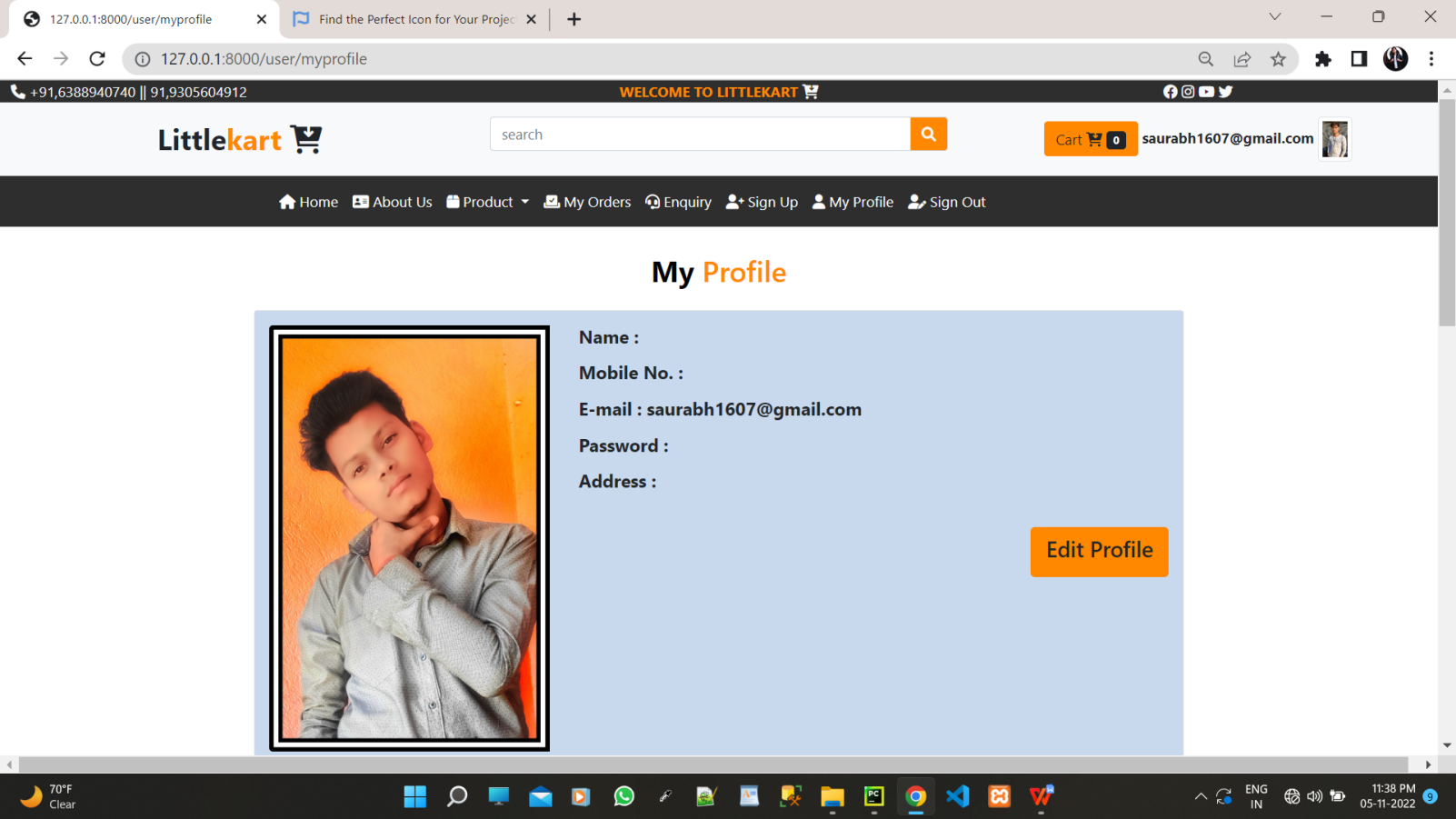
**Home Page Screenshot -**

****

**About Page Screenshot -**

****

**Profile Page Screenshot -**

****

**8.2 project Coding**

**Coding of Base.html file [similar code use to many category] -**

<!DOCTYPE html>  
{% load static %}  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <link href="{% static ''%}" rel="icon">  
 <title>  
 {% block title %}  
  
 {% endblock %}  
 </title>  
 <link href="{% static 'css/bootstrap.css'%}" rel="stylesheet"/>  
 <link href="{% static 'css/all.css' %}" rel="stylesheet"/>  
 <script src="{% static 'js/bootstrap.js' %}"></script>  
 <script src="{% static 'js/bootstrap.bundle.min.js'%}"></script>  
 <style>  
 .bg-color  
 {  
 background:#2E2E2E;  
 }  
 .bg-ocolor  
 {  
 background:#FF8800;  
 }  
 .bg-ocolor:hover  
 {  
 background:white;  
 border:1px solid black;  
 }  
 .text-ocolor  
 {  
 color:#FF8800;  
 }  
 #menu ul li a  
 {  
 color:white;  
 }  
  
 .ufooter  
 {  
 min-height:300px;  
 border-bottom:1px solid white;  
 }  
  
 .ufoot  
 {  
 min-height:250px;  
 float:left;  
 color:white;  
  
 }  
 .bfooter  
 {  
 height:50px;  
 font-size:22px;  
 text-align:center;  
 line-height:50px;  
 color:white;  
 }  
 .bfooter a  
 {  
 text-decoration:none;  
 }  
  
 .bfooter a:hover  
 {  
 color:white;  
 }  
 .w  
 {  
 background:url("{% static 'images/12.jpg'%}");  
 background-size:100% 100%;  
 min-height:230px;  
 padding-top:100px;  
 }  
 .w:hover  
 {  
 border:6px solid white;  
 }  
 .m  
 {  
 background:url("{% static 'images/5.jpg'%}");  
 background-size:100% 100%;  
 min-height:230px;  
 padding-top:100px;  
 }  
 .m:hover  
 {  
 border:6px solid white;  
 }  
 .k  
 {  
 background:url("{% static 'images/10.jpg'%}");  
 background-size:100% 100%;  
 min-height:230px;  
 padding-top:100px;  
 }  
 .k:hover  
 {  
 border:6px solid white;  
 }  
 .btn1  
 {  
 background:#FF8800;  
  
 }  
 .btn1:hover  
 {  
 background:white;  
 border:2px solid black;  
 text:bolder;  
 }  
  
 .product  
{  
background:light;  
}  
  
 .product:hover  
{  
background:white;  
border:6px solid white;  
}  
 .product  
 {  
 min-height:500px;  
 }  
 .big  
 {  
 min-height:200px;  
 border:3px solid white;  
 }  
 .message1  
 {  
 min-height:200px;  
 }  
  
 .zoom:hover  
 {  
 border:4px solid white;  
 }  
 </style>  
</head>  
<body class="bg-light">  
<div class="container-fluid">  
 <div class="row bg-color text-light" >  
 <div class="col-sm-4">  
 <i class="fa-solid fa-phone"></i>  
 +91,6388940740 || 91,9305604912  
 </div>  
 <div class="col-sm-4 text-center ">  
 <b class="text-ocolor"> WELCOME TO LITTLEKART</b>  
 <i class="fa-solid fa-cart-arrow-down"></i>  
 </div>  
 <div class="col-sm-4 " style="text-align:center;">  
 <i class="fa-brands fa-facebook"></i>  
 <i class="fa-brands fa-instagram"></i>  
 <i class="fa-brands fa-youtube"></i>  
 <i class="fa-brands fa-twitter"></i>  
 </div>  
 </div>  
 <div class="row header py-3 text-center">  
 <div class="col-sm-4 fs-2 text-center">  
 <b>Little</b><b class="text-ocolor">kart</b>  
 <i class="fa-solid fa-cart-arrow-down"></i>  
 </div>  
 <div class="col-sm-4">  
 <div class="input-group">  
 <input type="text" class="form-control" placeholder="search"/>  
 <i class="input-group-text bg-ocolor text-light">  
 <i class="fa-solid fa-magnifying-glass"></i></i>  
 </div>  
 </div>  
 <div class="col-sm-4">  
 <a href="/user/showcart" class="btn bg-ocolor">  
 Cart <i class="fa-solid fa-cart-arrow-down"></i>  
 <b class="badge bg-dark">  
 *<!-- {{request.session.cart}} -->* {{cart}}  
 </b>  
 </a>  
 {% if request.session.userid %}  
 <b>{{request.session.userid}}</b>  
 <img src="/{{request.session.userpic}}" class="img-thumbnail" style="max-height:50px; max-width:50px;"/>  
 {% else %}  
 <b>Guest User</b>  
 <img src="{% static 'profile/1.jpg'%}" class="img-thumbnail" style="max-height:50px;"/>  
 {% endif %}  
 </div>  
 </div>  
 <div class="row menu bg-color " style="min-height:45px;">  
 <div class="col-sm-2"></div>  
 <div class="col-sm-8">  
  
 *<!-- start menu -->*<nav class="navbar navbar-expand-lg " id="menu">  
 <div class="container-fluid">  
 <button class="navbar-toggler" type="button" data-bs-toggle="collapse"  
 data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">  
 <span class="navbar-toggler-icon"></span>  
 </button>  
 <div class="collapse navbar-collapse" id="navbarSupportedContent">  
 <ul class="navbar-nav me-auto mb-2 mb-lg-0">  
 <li class="nav-item">  
  
 <a class="nav-link active" aria-current="page" href="/user/index"><i class="fa-solid fa-house"></i>

Home</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link" href="/user/about"><i class="fas fa-address-card"></i> About Us</a>  
 </li>  
 <li class="nav-item dropdown">  
 <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">  
 <i class="fas fa-box"></i> Product  
 </a>  
 <ul class="dropdown-menu bg-dark ">  
 <li><a class="dropdown-item " href="/user/mens"><i class="fad fa-male"></i> Men's</a></li>  
 <li><a class="dropdown-item" href="/user/womens"><i class="fad fa-female"></i>

Womens</a></li>  
 <li><a class="dropdown-item" href="/user/kids"><i class="fad fa-baby"></i> Kids</a></li>  
 </ul>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link" href="/user/myorder"><i class="fas fa-vote-yea"></i> My Orders</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link " href="/user/enquiry"><i class="fas fa-headset"></i> Enquiry</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link " href="/user/signup"><i class="fa-solid fa-user-plus"></i> Sign Up</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link " href="/user/myprofile"><i class="fa-solid fa-user"></i> My Profile</a>  
 </li>  
 {% if request.session.userid %}  
 <li class="nav-item">  
 <a class="nav-link " href="/user/signout"><i class="fas fa-user-edit"></i> Sign Out</a>  
 </li>  
 {% else %}  
 <li class="nav-item">  
 <a class="nav-link " href="/user/signin"><i class="fas fa-user-edit"></i> Sign In</a>  
 </li>  
 {% endif %}  
 </ul>  
  
 </div>  
 </div>  
</nav>  
 *<!-- end menu -->* </div>  
 <div class="col-sm-2"></div>  
 </div>  
  
  
 <div class="row product">  
 <div class="col-sm-12 ">  
 {% block content %}  
 {% endblock %}  
 </div>  
 </div>  
 <div class="row bigsale big">  
 <div class="col-sm-6 text-center py-5 px-4 ">  
 <div class="row">  
 <div class="col-sm-12 zoom">  
 <div class="row">  
 <h2 class="text-ocolor text-center">....BIG SALE....</h2><br>  
 <b> 50% OFF on Men's Cloths </b><br>  
 <b>20% OFF on Every Kid's Cloths</b><br>  
 <b> Gety 10% Discount Every Product if buying above 2000 rupees </b>  
 </div>  
 <div class="row">  
 <img src="{% static 'images/30.jpg'%}" class="h-20% w-20% py-5 px-5 zoom"/>  
 </div>  
 </div>  
 </div>  
 </div>  
 <div class="col-sm-6 h-80%">  
 *<!-- start slider -->*<div class="row">  
 <div class="col-sm-12">  
  
  
 <div id="carouselExampleCaptions" class="carousel slide" data-bs-ride="true">  
 <div class="carousel-indicators">  
 <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>  
 <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1" aria-label="Slide 2" ></button>  
 <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="2" aria-label="Slide 3" ></button>  
 </div>  
 <div class="carousel-inner">  
 <div class="carousel-item active">  
 <img src="{% static 'images/32.jpg'%}" class="d-block w-100 " style="height:80%;" alt="...">  
 <div class="carousel-caption d-none d-md-block">  
 <h5>HOT DEALS </h5>  
  
 </div>  
 </div>  
 <div class="carousel-item">  
 <img src="{% static 'images/6.jpg'%}" class="d-block w-100 " style="height:80%;" alt="...">  
 <div class="carousel-caption d-none d-md-block">  
 <h5></h5>  
 </div>  
 </div>  
 <div class="carousel-item">  
 <img src="{% static 'images/31.jpg'%}" class="d-block w-100 " style="height:80%;" alt="...">  
 <div class="carousel-caption d-none d-md-block">  
  
 </div>  
 </div>  
 </div>  
 <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide="prev">  
 <span class="carousel-control-prev-icon" aria-hidden="true"></span>  
 <span class="visually-hidden">Previous</span>  
 </button>  
 <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide="next">  
 <span class="carousel-control-next-icon" aria-hidden="true"></span>  
 <span class="visually-hidden">Next</span>  
 </button>  
</div>  
  
</div>  
</div>  
 *<!-- end slider -->* </div>  
  
 </div>  
 <div class="row message">  
 <div class="col-sm-4 message1 px-5 py-5">  
 <div class="row ">  
 <div class="col-sm-12 bg-dark text-light text-center " style="height:80%;" >  
 <i class="fa-solid fa-people-carry-box text-center fs-2"></i>  
 <br><b class="text-ocolor fs-3">Delivery Services</b>  
 Read All Terms and <br>Condition's  
 </div>  
 </div>  
 </div>  
  
 <div class="col-sm-4 message1 px-5 py-5">  
 <div class="row ">  
 <div class="col-sm-12 bg-dark text-light text-center " style="height:80%;" ><i class="fa-regular fa-clock text-center fs-2 "></i>  
 <br><b class="text-ocolor fs-3">Delivery Time</b><br>  
 Check Current Status<br> Of Our Product</div>  
 </div>  
 </div>  
 <div class="col-sm-4 message1 px-5 py-5">  
 <div class="row ">  
 <div class="col-sm-12 bg-dark text-light text-center " style="height:20%;">  
 </div>  
 </div>  
 </div>  
 </div>  
 <div class="row ufooter bg-color border border-bottom border-1">  
 <div class="col-sm-4 ufoot text-center py-5 px-5"><img src="{% static 'images/23.png'%}" class="h-30% w-30%"/></div>  
 <div class="col-sm-4 ufoot text-center py-5 px-5">  
 <center>  
 <table style="padding:4px 4px;" >  
 <tr><th><b>Custmer</b> <b class="text-ocolor">Policies</b></th></tr>  
 <tr>  
 <td>Contact Us</td>  
 </tr>  
 <tr><td>FAQ</td></tr>  
 <tr><td>T&C</td></tr>  
 <tr><td>Terms of Use</td></tr>  
 <tr><td>Track Order</td></tr>  
 <tr><td>Shipping</td></tr>  
 <tr><td>Return</td></tr>  
 <tr><td>Cancellation</td></tr>  
 <tr><td>Privacy</td>  
 </tr>  
 </table>  
 </center>  
 </div>  
 <div class="col-sm-4 ufoot text-center py-5 px-5">  
 <center><table>  
 <tr><th>  
 <i class="fa-solid fa-person-walking-arrow-loop-left h-100% w-70% fs-1 " ></i></th></tr>  
 <tr><td><h2 class="text-ocolor">Return within 30days</h2> of receiving your order</td></tr>  
 <tr><th><b class="text-bg-ocolor">100% ORIGINAL guarantee</b><br> for all products at favcart.com</th></tr>  
 </table></center ><br><table><center>  
 <tr><th><h4>Registered<b class="text-ocolor"> Office Address</b></h4></th></tr>  
 <tr><td>Buildings Alyssa,</td></tr>  
 <tr><td>Outer Ring Road,Kalyanpur</td></tr>  
 <tr><td>Vikash Nager HP Petrol Pump</td></tr>  
 <tr><td>Lucknow</td></tr>  
 <tr><td>Help-Line No. <b class="text-ocolor">6388940740</b></td></tr>  
 </center></table></div>  
 </div>  
 <div class="row bfooter bg-color ">  
 <div class="col-sm-6">  
 Developed By : <a href="#" class="text-ocolor">Saurabh Kumar Mishra</a>  
 </div>  
 <div class="col-sm-6">  
 &copy; Copyright to : <a href="https://www.techpile.in" target="" class="text-ocolor">Techpile Technology Pvt. Ltd. </a>  
 </div>  
 </div>  
</div>

</body>  
</html>

## 9. FUTURE SCOPE

*Following modification or upgrades can be done in system.*

* 1. More than one company can be integrated through this software.
  2. Web services can be used to know exact donation status of packets.
  3. Client can check there donation delivery status online.

**10. CONCLUSION:**

At the last the Conclusion of project is to develop a web-application which the help programmer to get help from the site, so that they can develop their project and application Different Technologies and make a group of programmer. A **Group** is a social unit of any size that shares common values, ideas and code and queries. The portal doesn’t have to be expensive. It supports multiple programmer goals.

**FavCart** is not only a web portal; it is a live product of board of technical education. In future we will add more and more features on it.