WEB APPLICATION MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

ANBARASU.S
(Reg. No: 19BIR003)

TARUN.N
(Reg. No: 19BIR052)

in partial fulfilment of the requirements for the
Award of the degree of

BACHELOR OF SCIENCE
IN
INFORMATION SYSTEMS

DEPARTMENT OF COMPUTER TECHNOLOGY-UG

KONGU ENGINEERING COLLEGE
(Autonomous)
PERUNDURAI ERODE-638 060

MAY 2022

DEPARTMENT OF COMPUTER TECHNOLOGY-UG

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE-638 060

MAY 2022

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **WEB APPLICATION MANAGEMENT SYSTEM** is the bonafide record of project work done by **ANBARASU S (19BIR003)** and **TARUN N (19BIR052)** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Science in Information Systems of Anna University Chennai during the year of 2021-2022.

SUPERVISOR **HEAD OF THE DEPARTMENT**

(Signature with Seal)

Date:

Submitted for the End Semester Viva Voce examination held on ________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DEPARTMENT OF COMPUTER TECHNOLOGY-UG

KONGU ENGINEERING COLLEGE

(Autonomous)

PERUNDURAI ERODE-638 060

MAY 2022

DECLARATION

We affirm that the project report titled **WEB APPLICATION MANAGEMENT SYSTEM** being submitted in partial fulfilment of the requirements for the award of Bachelor of Science is the original work carried out by us. It has not formed the part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion of any other candidate.

Date: **ANBARASU.S (Reg.No:19BIR003)**

TARUN.N (Reg.No:19BIR052)

I certify that the declaration made by the above candidates is true to the best of my knowledge.

Date: Name and Signature of the Supervisor with seal

ABSTRACT

The main objective of the project entitled, “**WEB APPLICATION MANAGEMENT SYSTEM**” is to design and develop a system to maintain and manage the product sales in an organization. This is an attempt to computerize the day-to-day activities of a Circuit manufacturing company.

The aim of this project is to reduce the tedious work of managing, maintaining and updating the database. It enables the features such as automating the billing of customer, items available and monthly report. It updates the stock according to the changes made in supply. Details of customer billing also maintained along with all.

The process of this project is to the company to perform daily product sales and transaction with ease. The system designed to monitor various product sales within an organization. The design collects the information such as Customer details, product details, product price, Number of Item Sales, Report of the Item, Product Reviews and Ratings, etc., One of the aspects of this project is to create a smooth UI Environment which improves the Customer satisfaction

The Implementation of this project is done using HTML, CSS, JAVASCRIPT as client-side Technologies and python as a server-side Technology. To Enhance the architecture of this project, Django Framework is utilized.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved correspondent **Thiru.P. SACHITHANANDAN** and other philanthropic Trust members of the Kongu Vellalar Institute of Technology Trust for having provided with necessary resources to complete this project in a successful manner.

We are always grateful to our beloved visionary principal **Dr.V. BALUSAMY B.E(Hons)., M.Tech., Ph.D.,** and thank him for his motivation and moral support.

We express our deep sense of gratitude and profound thanks to **Dr.P. NATESAN M.E., Ph.D.** Head of the Department, Computer Technology-UG for his invaluable commitment and guidance for this project.

We are in immense pleasure to express our hearty thanks to our beloved project coordinator **Dr. RENUKADEVI MCA., M.Phil., Ph.D** and our guide **Dr.S. KALAISELVI M.E., Ph.D** for providing valuable guidance and constant support throughout the course of our project. We also thank the teaching, non-teaching staff members, fellow students, and our parents who stood with us to complete our project successfully.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
JSON	JavaScript Object Notation

CHAPTER 1

INTRODUCTION

The “**WEB APPLICATION MANAGEMENT SYSTEM”** is a web-based application designed on python, HTML and JavaScript to manage real-time inventory database. This may be used to control the allocation of items between several users of a larger franchise or to oversee the inventory of a single firm. The system, on the other hand, just collects purchases as well as refilling information and transfer out low-stock messages. Rather of handling all store upkeep, the idea is to lessen the load of tracking. Additional capabilities may include the capacity to create daily or weekly sales and inventory reports, although interpretation is left to management.

Still, maintaining direct sales and manual record details. Now, the project is developed like an online shopping and maintaining a purchase and sales records automatically. Now days, all purchasing and transaction are online. In future maximum people buy products in online shopping. It includes some tasks such as Customer details, product details, product price, No. of Item Sales, Report of the Item, Product Reviews and Ratings and etc. It is very difficult to maintain historical data. It’s cost effective and expensive. This application offers a general organization profile, sales information, purchase details, and the organization's remaining stock. This programme also displays the remaining stock balance as well as transaction balance data. Each new stock is entered and labelled with its name and entry date, and it may be modified at any moment as needed according to the transaction. The login page is built in this case to secure the organization's inventory management against threats and abuse of the inventory.

1.1 ABOUT AUTOMATION COMPONENTS

In industrial automation control, many processes such as temperatures, pressure, flow rate, range, as well as liquids levels can indeed be tracked at the same time. All of the data is recorded, processed, and controlled by complicated microcontroller, computers and PC-based data processing controllers.

1.2 ABOUT E-COMMERCE

E-commerce is the selling and marketing of goods over the web. And services, as well as the free flow of capital and information required to finish the transaction It's also known as an internet-based electronic transaction.

1.2.1 Business to Consumer (B2C):

Business-to-consumer e-commerce is the most frequent (B2C). A transaction between a firm and a client, except when we acquire a sensor from an online store, is referred to as business to consumer.

1.2.2 Business to Business (B2B):

A firm providing a service or product towards another organization, other than a supplier and a buyer, or a distribution company and B2B e-commerce occurs when a company sells to another company. E-commerce marketing is not for consumers, and it frequently requires raw materials, software, or a mix of goods. Manufacturers can sell directly to shoppers through B2B e-commerce. This site will examine the e-commerce industry in depth, including how it grew, what types of shopkeepers operate, or what mechanisms allow online selling. We'll also shed light on notable e-commerce successful projects and failures to give you a clearer sense of how much it takes to succeed in this industry.

1.2.3 Advantages of this Project:

In this application we using high secure payment for transaction, and sending order and payment details through mail services and additionally chat bot are enabled for customers to ask queries and we can generate report in different types. It guarantees that goods are provided swiftly and with little effort on the client's part.

Customer concerns are also addressed swiftly. It also saves a lot of time, effort, and energy for both consumers and enterprises. Another big benefit is the simplicity it offers. The products are available for inspection 24/7 a day, 7 days a week. From any location, the seller may readily monitor the order.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The present system being carried out manually. There are lots of difficulties available in the existing system and due to that checking for the stock available is taken in consideration. All the records have to be calculated and written manually. It is not user friendly and interactive. Accurate result is not calculated because of the manual calculations. So, there is need for a proposed system to rectify the drawbacks of the existing system and so it is computerized.

2.1.1 Drawbacks of Existing System

The disadvantages of existing system are:

- Time consumption is more for preparing information of current services and bill preparation.
- More number of hard copies is to be maintained to avoid data loss.
- It does not ensure security mechanisms.
- Chances for data redundancy or data omitted.

2.2 PROPOSED SYSTEM

The proposed system helps in managing fast data processing and report generation. The drawbacks of the existing system such as checking for the orders available are taken into consideration and it shows the stock details. Project is to ensure user friendly and more interactive to the administrator and users.

2.2.1 Advantages of Proposed System

The proposed system has following advantages:

- Reduces the human effort compared to other conventional methods.
- Accuracy and error free transaction process.
- Soft copies can be easily maintained to prevent data loss.
- Production reports are easily prepared. Report generation is easier and efficient.

2.3 FEASIBILITY STUDY

The feasibility study encompasses every one of the research goes into developing the concept. This aids in deciding whether or not to pursue the issue. When developing the project, each structure must be evaluated since it must service the end consumer in a consumer manner. System analysis comprises acquiring, arranging, and evaluating data on a system and its surroundings. A feasibility study is carried out to determine whether the proposed system is viable to build with the existing funds and whatever the financial constraints should be.

2.3.1 Economic Feasibility

The economic feasibility analysis compares the cost of software development to the ultimate income or advantages obtained from the established system. The organization must purchase computers, keyboards, and mouse, as well as set up a local area network; this is a direct expenditure. There are several immediate advantages to switching from a manual to a digital system. The user can receive solutions to inquiries. The reason for any capital expenditures is that it would lower spending or enhance the quality of service or commodities, which in turn may be expected to bring improved profits.

2.3.2 Operational Feasibility

This study includes determining how it will function once installed as well as evaluating the leadership culture in which it will be applied. The current proposal accessing process resolves issues that arose in the previous system. This system can accommodate the company's present inventory management processes. The main focus of operational feasibility should be an examination of how the proposed system would affect organizational structures and procedures.

2.3.3 Technical Feasibility

Technical feasibility considers whether the technology is marketable for development. To establish technical feasibility, comprehensive system design needs in terms of inputs, output documents, programs, and technique must be employed. The current system attempts to address the inadequacies of the prior system. The current solution is intended to reduce the technical ability required so that more individuals may participate in the programme.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE SPECIFICATION:

System: I5

Hard Disk: 500 GB

Monitor: Led

Keyboard: Logitech

Mouse: Optical Mouse

Ram: 8 GB

3.2 SOFTWARE REQUIREMENTS

Operating System: Windows 7 (minimum)

Front End: Html, CSS

Back End: Django

3.2.1 Front End

HTML

HTML is a language made up of components that may be applied to text in a document to give it alternative meaning, arrange a document into logical sections, and integrate material such as photos and videos into a website. This lesson will cover the first two, as well as the essential principles and syntax required to understand HTML.

A web page is a document that's also usually written in HTML and is subsequently translated by a web browser. A URL could be used to find a website. A Web page might be static or dynamic. We may create static web pages solely with HTML.

INTRODUCTION OF HTML

Hypertext markup language (Html). It's used to make websites by utilising a markup language. Hypertext distinguishes the relationship between pages. A scripting language is used to create a textual reporting inside the term that indicates site page development This vocabulary is used to simplify information so that it can be understood and processed by a machine (note for the PC). Most of markup dialects (for instance, HTML) are comprehensible. Labels are used in the language to specify what text treatment is necessary. Html tags that programme use to format text, images, and other content so that it displays correctly. Tim Berners-Lee designed HTML in 1991. HTML 1.0 was the underlying rendition, while HTML 2.0, gave in 1999, was the main standard variant.

HTML parts are the groundwork of HTML pages. Images and other items, such as sensible structures, can be incorporated into the supplied page using HTML structures. HTML allows you to create coordinated archives by expressing fundamental meanings for text components such as headers, passages, entries, links, citations, and so on. HTML components are assigned via identifiers, that are expressed in point sections. Labels like directly embed material in the page Extra labels, such as enclose and provide data about the archives content, and it may contain other labels as sub-components. HTML labels are not shown by programmes, but rather use them to decipher the page's content.

HTML may include pre-programmed scripts, such as JavaScript, that alter the behaviour and content of website pages. CSS determines how material appears and is laid up. Since roughly 1997, the Wide Web Consortium (W3C), the original supervisor of a Html5 is the latest and the current custodian of the Style standard, has promoted the use of CSS over unambiguous presentational HTML. CSS is a design language that is used to display the highlights of a report written in a modelling language such as HTML. CSS, like JavaScript, is a fundamental invention of the Online World. CSS is intended to separate show from content by separating formatting, colours, and text styles. This division can increase material availability, give additional control and flexibility in describing display highlights, and allow different sites to exchange arranging by providing the required CSS in a separate. CSS document, with less complication and obvious repetition in the core essence.

The separation of design and content enables an identical markup page to be presented in several forms for various delivery methods, for example, on-screen, on paper, by voice (through discourse-based programme or display reader), and on Braille-based material devices. CSS also retains rules for different arranging while reviewing data for just a cell phone. The word flowing is derived from the requirement instrument used to determine which style rule applies, presuming that more than one rule matches a single component. This need structure with flowing needs is predictable.

SCRIPTING LANGUAGE

Python is an open, significant level programming language. Its plan idea focuses on code lucidness by utilizing weighty space. Its linguistic components and article-organized methodology are intended to assist developers in recording clear, valid code for both small and large-scale projects.

Python is trash gathered and powerfully composed. It adheres to a wide range of programming principles, Structured (particularly mechanistic), object-oriented, and pragmatic programming are all examples of programming. It's sometimes referred to it as a "rechargeable batteries included" language because of its large executable.

Guido van Rossum began developing Python in the late 1980s as a replacement for the ABC computer language, and first released in 1991 as Py 0.9.0. Python 2.0 was released in 2000, with new features such as list cognizance, cycle-identifying garbage collection, reference counting, and Unicode compatibility. Python 3.0, released in 2008, was a massive update which was not completely backward compatible with previous variants. Python 2 was retired in 2020, with version 2.7.18.

FEATURES

Python may be used to write scripts for web apps such as module wsgi for such Apache web server. To assist these applications, a standard API, Web Server Connector Interface, has emerged. Web structures such as Django, Gantries, Pyramids, Component based, web2py, Tornado, Flask, Bottles, or Zope allow developers to create and maintain sophisticated frameworks. Pyjs and IronPython may be used to create client-side Ajax apps. SQLAlchemy might be used as a social data base information mapper. Contorted is a programming framework for PC communiques that is utilised by Dropbox and others. Python may be used to write scripts for web apps such as module wsgi for such Apache web server. To assist these applications, a standard API, Web Server Connector Interface, has emerged. Web structures such as Django, Gantries, Pyramids, Component based, web2py, Tornado, Flask, Bottles, or Zope allow developers to create and maintain sophisticated frameworks. Pyjs and IronPython may be used to create client-side Ajax apps. Contorted is a programming framework for PC communiques that is utilised by Dropbox and others.

The NumPy, and Matplotlib packages make it possible to use Python for logical registration, while Bio Py and Astropy provide area-specific features. SageMath is a Scripting language PC variable-based math platform with a notebook interface that includes variable-based math, combinatorial optimization, mathematical science, number theory, and analytics. Python OpenCV ties provide a wide variety of utility for Computer vision and image processing.

Python is frequently used in artificial intelligence and consciousness applications due to its extensive library, for example, TensorFlow, Keras, Pytorch, and Scikit-learn. Python is every now and again utilized for regular language handling as a result of its measured plan, simple grammar, and broad text handling highlights. Python may likewise be utilized to build games, on account of bundles like Pygame, which can produce 2D games.

Python has been successfully incorporated into a variety of product items as a planning language, particularly for finished utilizing technique programmers, such as Abaqus, 3D parametric modelling, such as FreeCAD, and 3D activity bundles, such as 3ds Max, Food processor, Movie theatre simulation, Eponymously, Nightcrawler, Maya, modo, Motion Building contractor, Softimage, and the enhanced visualisations typesetter. GIMP, Nuke, and other 2D imaging applications Python has been successfully installed as a prearranging language in a variety of product items, including limited component strategy programming such as Abaqus, 3D parametric modellers such as FreeCAD, 3D movement bundles such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the enhanced visualisations typesetter Nuke, 2D imaging programmes such as GIMP, Inkscape, Scribus, and Paint Shop Pro, and Python is used by GNU Debugger as a good printer for displaying complicated designs. Like C++ holders.

Esri recommends Python as the finest language for writing scripts in ArcGIS. It has also been used in several games' consoles, and that it's the first among 3 programming languages available in Google Application Servers, the other two are Java and Python. Score writer and Capella are melodic documentation applications, as are Inkscape, Scribus, and Paint Shop Pro. Python is used by GNU Debugger as a gorgeous printer to display sophisticated designs such as C++ holds. Python is recommended by Esri as the best language for writing scripts in ArcGIS. It's also been used in a few video games, and it's now the first of three programming languages access to Google App Engine (both other two being Jupiter and Go).

Many functioning frameworks include Python as a standard component. It's included in most Linux distributions, as well as AmigaOS 4 (which uses Py 3.8), FreeBSD (as a package), NetBSD, FreeBSD (as a package), and macOS, and may be launched from the command line (terminal). Python installers are used by several Linux distributions: Ubuntu uses the Ubiquity installation, while Red Hat Linux as well as Fedora Linux use the Anaconda installer. Python is used in Gentoo Linux's Portage package management infrastructure. Python is widely used in the realm of data security, particularly in the creation of attacks.

Sugar Labs is now working on the majority of the Sugar software in Python only for One Laptop per Child XO. Python has been chosen as the primary user-programming language for the Arduino Microchip central computer project. Python is included with LibreOffice and is intended to take the role of Java.

3.2.2 Back End

Django is a Python web structure that takes into account the speedy structure of protected and kept up with sites. Django, which was made by proficient designers, deals with the vast majority of the migraine of web improvement, permitting you to zero in on fostering your application as opposed to rehashing an already solved problem. It is free and open source, has a strong and dynamic local area, amazing documentation, and a few free and paid-for help choices.

Django can (and has been) utilized to make practically every type of site, from content administration frameworks and wikis to informal communities and news destinations. It is viable with any client-side system and can serve material in almost any configuration (counting HTML, RSS channels, JSON, XML, and so on.)

Inside, while it offers choices for essentially every capacity you would need (e.g., different significant data sets, templating motors, etc.), it can likewise be altered to utilize extra parts if important.

Django helps designers in staying away from numerous runs of the mill security bumbles by offering a system as a result, efforts have been made to "do the right things" in order to protect the site. and passwords, staying away from common traps like keeping meeting data in treats (all things being equal, treats convey a key, while the genuine information is saved in the data sets) or essentially putting away passwords rather than a secret phrase hash.

Django utilizes a "shared-nothing" part-based plan (each piece of the engineering is free of the others, and can subsequently be supplanted or changed if necessary). Since the various components are obviously isolated, it may grow traffic by adding hardware at any level, including backup servers, shared by various servers, and application servers. The most well-known websites have successfully scaled Django to handle their problems (for example Instagram as well as Disqus, to name only two).

Django writing computer programs is made as per plan standards and examples that advance the development of viable and reusable code. It explicitly utilizes the Don't Repeat Yourself (DRY) reasoning to kill unnecessary reiteration, henceforth bringing down how much code. Django additionally energizes the association of equivalent usefulness into reusable "applications" and, at a more profound level, the association of related programming onto parts.

VISUAL STUDIO CODE

Visual Studio Code is an origin editor that supports Java, Facebook, Go, Node.js, Python, C++, and Fortran, among other programming languages. It's built on the Electron platform, which is used to create Node.js Web apps that employ the Blink layout engines. The same editor component (codenamed "Monaco") that is used in Azure DevOps is utilized in Visual Studio Code

Extensions for Visual Studio Code are available through a common repository. This features editor enhancements as well as language support. The Appointed To serve Protocol allows users to write modifications that add capital for future language, themes, and debuggers, as well as perform static analysis and add code linters.

Visual Studio Code contains several FTP extensions, allowing it to be used for a free web development alternative. Without downloading any additional software, code may be synchronized between both the browser and the server. Users may customize the code page where the active project is stored, the delimiter character, and the active document's programming language in Visual Studio Code. This enables it to run on any system, in any location, and with any programming language.

CHAPTER 4

PROJECT DESCRIPTION

4.1 OVERVIEW OF THE PROJECT

“**WEB APPLICATION MANAGEMENT SYSTEM**” is an application for online shopping. The owner of this project may access and change stock data as well as keep customer information. The owner will be provided with a user id and a password. In this application, the owner can log in with their user id and password and they can view the details of automation components. If the stock decreases the owner can update the inventory stock details. The profit and loss for the annual year can be calculated through software. The owner can access the application from any place because it is stored in the centralized database. This application will be very useful as it displays all the details in the page itself when we log in with the password and username.

4.2 MODULE DESCRIPTION

The project “ONLINE ENTERPRISE MANAGEMENT SYSTEM” contains the following modules,

- customer details
- Item details
- Order details
- Report details

4.2.1 Customer Details

This module is designed to collect information about the customers. Customers are an important asset for any company. Details such as the customer number, name, address, and contact information are gathered. A separate table is designed to store information in the database. Customers can order the bed and mattress from the menu displayed so that the order can be processed accordingly.

4.2.2 Product Details

This module maintains information about the items the company deals with. The information like item no, item name, description, quantity, and amount of the item are maintained. A separate table will be designed to store this information in the database. These can be aligned and can be modified by the admin.

4.2.3 Order Details

This order information module stores product information of customers and goods to the database and displays the information to the administrator.

4.2.4 Report Details

Reports are designed to make a good analysis of the sales data. The generated reports will help the user to make planning for the future. Following are the various reports supported.

1. Item wise report
2. Date wise sales report
3. Maximum and minimum sale for Item and Area

4.3 SYSTEM FLOW DESIGN

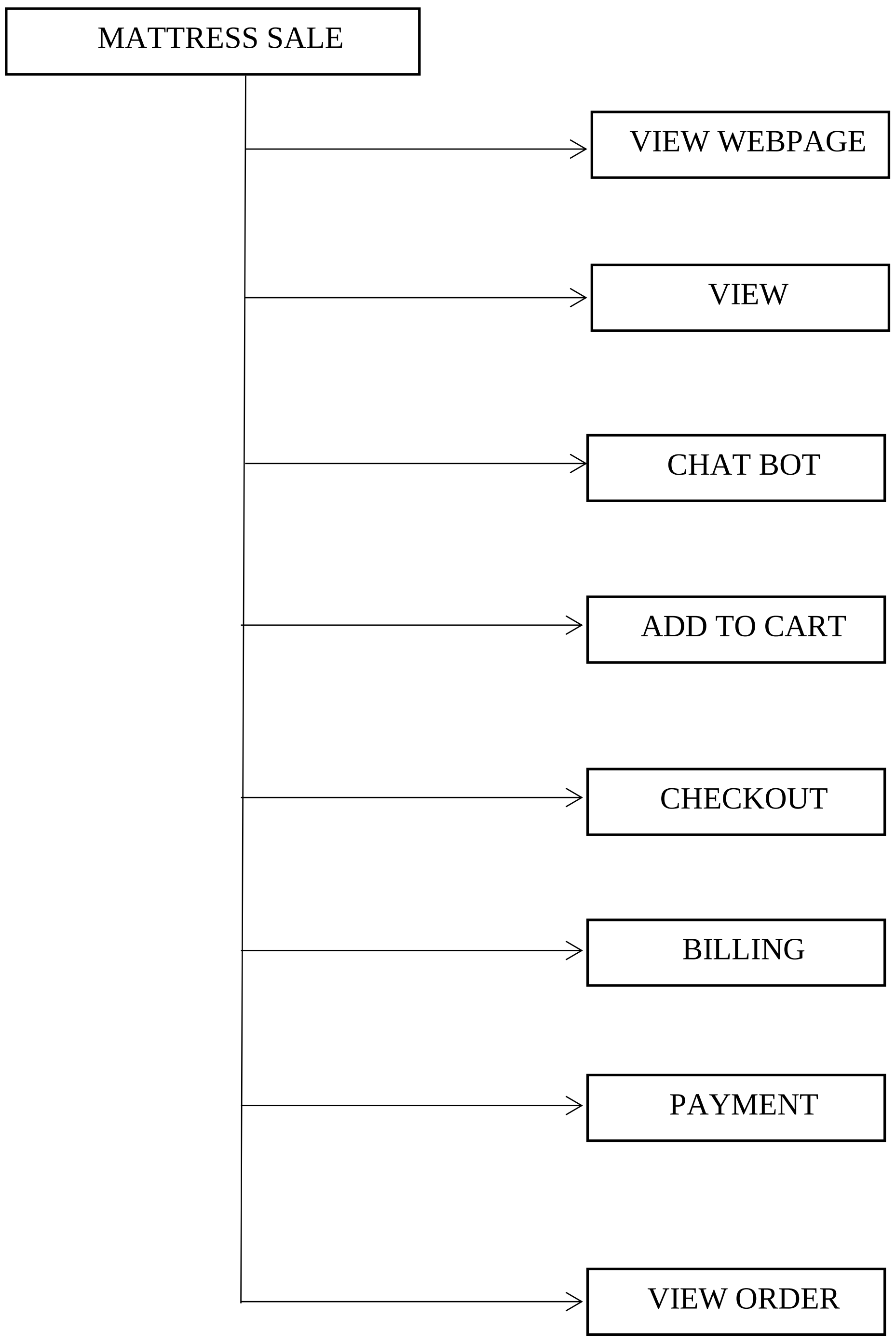


Fig 4.1 System Flow Diagram - User

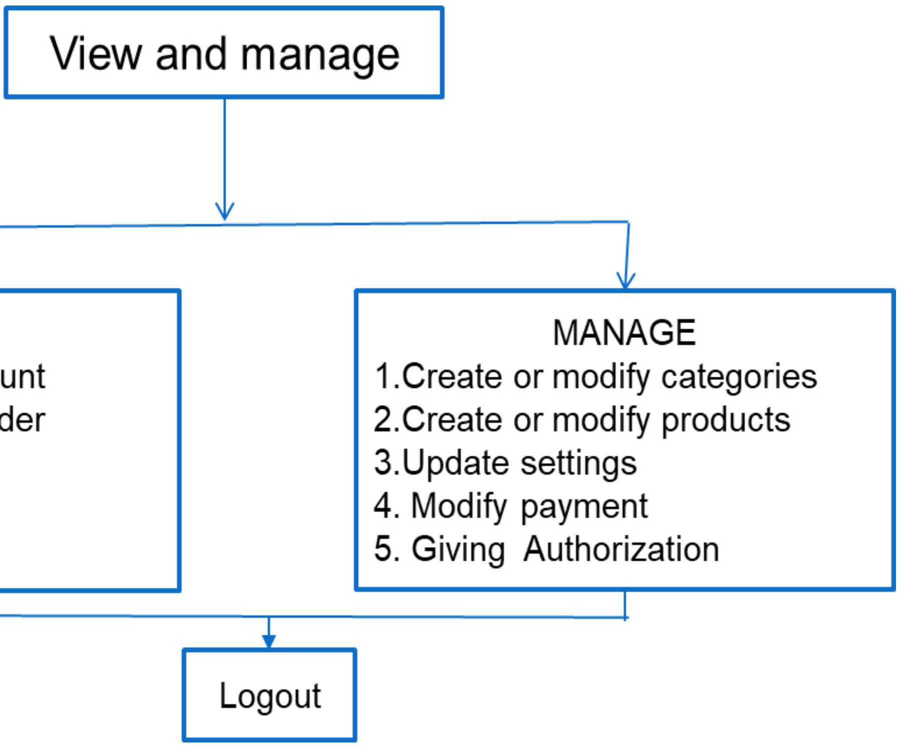


Fig 4.2 System Flow Diagram - Admin

4.4 DATA FLOW DESIGN

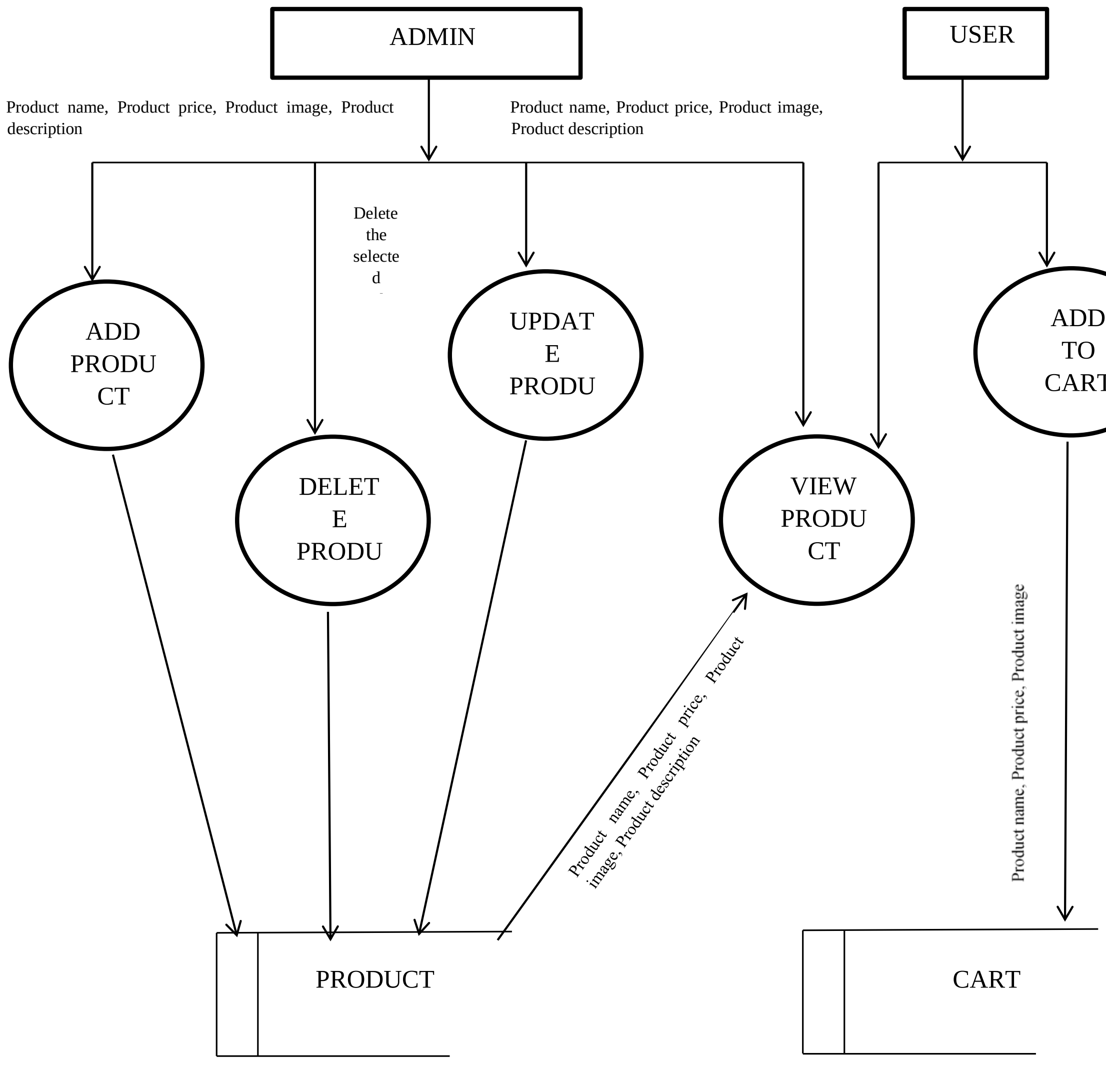

ADMIN
USER
Product name, Product price, Product image, Product description
Product name, Product price, Product image, Product description
Delete the selecte d
ADD PRODU CT
DELET E PRODU
UPDAT E PRODU
VIEW PRODU CT
ADD TO CART
Product name, Product price, Product image, Product description
Product name, Product price, Product image
PRODUCT
CART

Fig 4.3 Level 0 – Data Flow Diagram

4.5 DATABASE DESIGN

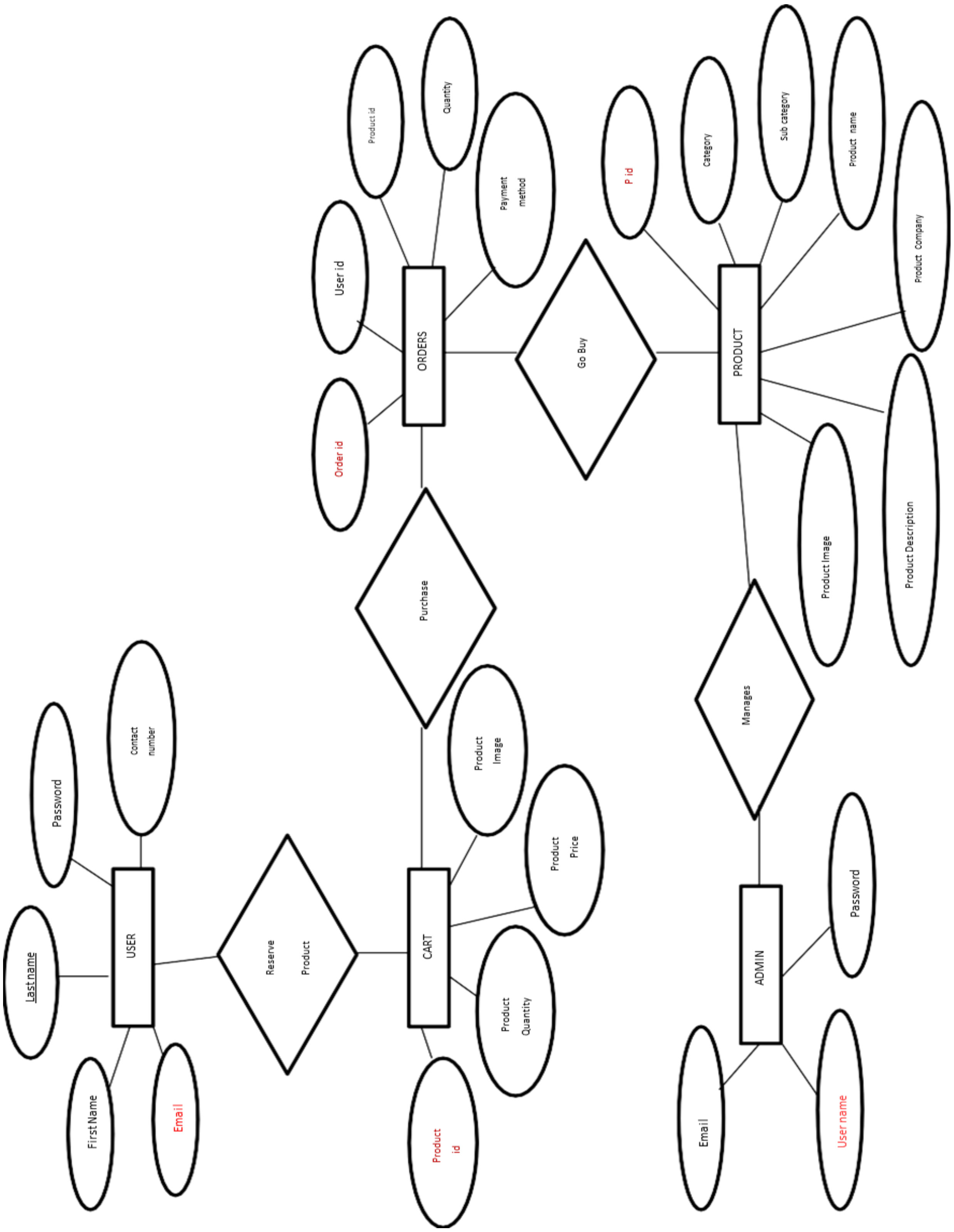


Fig 4.4 Entity Relationship Diagram

Table creation is a step in the databases design process. Databases are physically represented as stored files in the database. They exist independently. Tables are made up of rows and columns. Each column represents information about a certain member. The database design for this project is shown below.

CUSTOMERS

S.NO	FIELD NAME	DATA TYPE	CONSTRAINT
1	First_name	CharField()	
2	Last_name	CharField()	
3	Phone	IntegerField()	
4	Email	CharField()	PrimaryKey
5	Password	CharField()	

PRODUCT

	FIELD NAME	DATA TYPE	CONSTRAINT
1	Product_id	IntegerField()	Primary key
2	name	CharField()	
3	Price	FloatField()	
4	Category	CharField()	
5	Sub_category	CharField()	
6	stockQuan	IntegerField()	
7	Description	TextField()	
8	Image	ImageField()	

ORDER ITEM

S.NO	FIELD NAME	DATA TYPE	CONSTRAINT
1	order_id	CharField()	Primary Key
1	customer	OneToOneField()	Foreign Key
2	Ordered	BooleanField()	
3	Product	ManyToManyField()	Foreign Key
4	quantity	IntegerField()	

ORDER

S.NO	FIELD NAME	DATA TYPE	CONSTRAINT
1	customer	OneToOneField()	Foreign Key
2	Date_ordered	DateTimeField()	
3	complete	BooleanField()	
4	Transaction_id	CharField()	Primary Key
5	Orderitems	ManyToMany()	Foreign Key
6	ShippingAddress	OneToOne()	Foreign Key

ADDRESS

S.NO	FIELD NAME	DATA TYPE	CONSTRAINT
1	order	ManyToMany()	Foreign Key
2	address	CharField()	
3	City	CharField()	
4	country	CountryField()	
5	zipcode	IntegerField()	
6	Date_added	DateTimeField()	

CUSTOMER MESSAGE

S.NO	FIELD NAME	DATA TYPE	CONSTRAINT
1	User	OneToOneField()	Foreign Key
2	Email	EmailField()	
3	topic	CharField()	
4	Message	TextField()	

4.7 INPUT DESIGN

The input design process involves converting input requirements into a machine-readable format. The goal of input design is to develop an easy-to-follow, user-friendly input layout that avoids operator errors.

The most common data processing errors produced by data entry operators are caused by inaccurate data. The user can enter the required and formatted date with the help of the error message.

Formatted input entries such as date, drop down, file, textfield, window allow the user to enter data quickly and easily even if they are unfamiliar with the product. Here, too, great effort is taken to ensure that GUI-based development follows the same set of standards and guidelines.

The Menu based product helps even the native user work with the product. The successes are designs in such a way to help the user to get the information whenever necessary.

4.7 OUTPUT DESIGN

The Output designs are displayed in different report formats. Different output designs will improve the clarity and performance of output. The output designs are classified into individuals and groups of tables.

And also display the reports in lab tests, cross-matching, issue details are available in my project. It is used to check the collection of particular times of the period.

Report 1 Customer Report

By using admin _User Id we can view all details of the customer about the particular customer and delete particular customer details.

Report 2 Purchase Report

We can see when the mattress stock was last purchased and can delete particular bed stock details.

Report 3 Stock Report

The user can view the current availability of the mattress stock.

CHAPTER 5

SYSTEM TESTING

Demonstrated as related data structures when the source code has been completed. When the project is finished, it must go through testing and validation, where there is a systematic and definite endeavor to find flaws. The project developer treads carefully, writing and executing tests that indicate that the software works rather than discovering mistakes; yet, defects will exist, and if the contract administrator does not find them, the user will. The contract administrator is always in charge of testing the program's various pieces, or modules. In many circumstances, the developer will also do integration testing, which is the testing process that leads to the building of the entire programme structure.

5.1 UNIT TESTING

This is a computing system's unit testing is a software unit, and it is often tested via white box testing. Unit tests are often performed by the program's creator. First, technique level testing is performed. By providing incorrect inputs, the mistakes that occurred are identified and eliminated. The custom web level testing is then performed. For example, the right preservation of data toward the table.

The zero-length name and password are supplied and validated in both the company and the seeker registration form. In addition, the duplication username is provided and verified. Individual input fields are validated for upper and lower limits valid data ranges. The dates being entered incorrectly and checked.

Test 1

Module	: Login
Mail id	: admin@abc.com
Password	: 123456
Output	: Logged in successfully
Event	: Button click.
Analysis	: mail id and password has been verified.

Test 2

Module	: Login
Mail id	: admin@abc.com
Output	: wrong-password
Event	: Button click.
Analysis	: Password field is checked and error is shown.

Analysis: In this form, the admin id and admin password have been tested in the incorrect format. If the form admin id or admin password is mismatched to login into the database means, here the data mismatch error can occur.

5.2 INTEGRATION TESTING

The many elements of the system are incorporated with each other to form the full system in integration testing, and this sort of testing evaluates the network to also ensure that it would do what it is meant to do. An integrated model can be tested top-down, upper, or big-bang style. Some elements will be examined using white box testing procedures, while others will be tested using black box testing methodologies. This form of testing is critical to boosting the system's productivity. We tested our system utilizing integration testing approaches.

Test case 1

Module: customer menu

Test type: Payment Process

Input: giving incorrect credentials

Expected output: It shows a warning message.

Sample test

Input: On clicking, card number text field and giving incorrect numbers

Output: It gives a warning and display the fields.

Analysis: Respective menus open

5.3 VALIDATION TESTING

Examining software during or after development to check that it complies with stated business needs. The testing stage verifies that the product meets the requirements client's requirements. It may also be described as proving When employed in the appropriate environment, the product operates as predicted.

Test case

Module: Register

Test type: Register new user

Input: Input to all fields

Expected outputs: No required field should not be empty.

Sample test

Input: Input for a required field is not provided.

Output: Provide all required fields.

Analysis: The expected output is the same, so the form passed the validation test.

CHAPTER 6

SYSTEM IMPLEMENTATION

When the first design for the system was completed, the client was contacted for design acceptance so that the system development process could continue. Following the implementation of the prototype, they were given a demonstration of how the system worked. The goal of the system visualization was to detect any system flaws.

After the system's management was accepted, the system was installed in the company, initially running in parallel with the old manual system. When tested with live data, the system was deemed to just be error-free as well as user-friendly.

Implementation is the process of turning a novel or updated design method into a functional one. After the system completed the original design, a demonstration of the functional system was delivered to the end user.

By feeding various permutations of test data, this technique is being used to check and discover any logical problems with the system's operation. The system was introduced once it had been approved by both the end user and management. Many actions are involved in system implementation.

6.1 SIX KEY PHASES:

6.1.1 Coding:

Coding refers to the procedure by which mechanical design requirements supplied by the analytical team are transformed into usable programming code mostly by group of programmers.

6.1.2 Testing:

Each software module may be verified as the development process begins and continued in parallel.

6.1.3 Installation:

Installation is indeed the procedure of replacing the original system with a new system. This requires changing outdated data, software, documents, and work practices to ones that are compatible with the most recent system.

6.1.4 Documentation:

As a consequence of the installation procedure, user manuals explain how to use the system as well as its flow.

6.1.5 Training And Support:

A learning control is a blueprint for teaching users so that they can learn the new system rapidly. The training strategy was most likely developed early in the project.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

The project "WEB APPLICAYION MANAGEMENT SYSTEM" was planned and built-in accordance with the specifications. The project is quite easy to use. The system is evaluated using a variety of sample data. The project has been verified successfully, as defined in the abstract with system addition to reporting. In the event of manual administration, there is a risk of making mistakes, losing data, and having difficulties obtaining and analyzing the data. It reduces the time required for manual stock services and management record keeping.

It is determined that the application functions well and meets the needs of the user. The programme has been thoroughly tested, and any faults have been thoroughly debugged. The programme is accessible from several systems at the same time. The entire procedure is tested, from login to exit. It lessens the task of keeping records. The end-users are required to have less working experience to run this project and also create a user-friendly application to access.

7.2 FUTURE WORK:

Enhancement, modification, or redevelopment of code to support changes in the specification. It is important to stay up with changing user needs and the operating environment.

Following are some of the features that can be considered as future enhancement.

1. AR Model
2. Making Mobile Application
3. Inventory Updating
4. Order Tracking

APPENDIX

settings.py/ecommerce

```
ALLOWED_HOSTS = []
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'crispy_forms',
    'store.apps.StoreConfig',
]
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'ecommerce.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ["templates"],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
WSGI_APPLICATION = 'ecommerce.wsgi.application'
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

urls.py/ecommerce

```
rom django.contrib import admin
from django.urls import path, include
from django.conf.urls.static import static
from django.conf import settings
from django.contrib.auth import views as auth

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('store.urls')),
]

urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

urls.py/store

```
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [    path('index', views.store, name ='index'),
    path('register/',register.as_view(), name="register"),
    path('login/',Login.as_view(), name="login"),
    path('logout' ,logout ,name='logout'),
    path('', views.store, name="home"),
    path('cart/', views.cart, name="cart"),
    path('checkout/', views.checkout, name="checkout"),
    path('update_item/', views.updateItem, name="update_item"),
    path('process_order/', views.processOrder, name="process_order"),
    path('process_contact/', views.processContact, name="process_contact"),
    path('product/<int:productId>', views.product, name="store-product"),
    path('catalog/', views.catalog, name='store-catalog'),
    path('category/<int:categoryId>', views.category, name='category'),
    path('subcategory/<int:subcategoryId>', views.subcategory,
name='subcategory'),
    path('search_result/', views.search, name='store-search'),
    path('pricefilter/<int:subcategoryId>', views.priceFilter, name='store-
filter'),
    path('contacts/', views.contacts, name='store-contacts'),
    path('about/', views.about, name='store-about'),
    path('fqa/', views.fqa, name='store-fqa'),
    path('stock', stock.as_view(), name='stock'),
    path('report', views.report, name='report'),
]
```

views.py/store (Login and SignUp)

```
 class register(View):
    def validateCustomer(self,customer,error):
        error_message = ''
        if (not customer.first_name):
            error_message =error_message+ "First Name Required !!\n"
        elif len(customer.first_name) < 4:
            error_message = error_message +'First Name must be 4 char long or
more\n'
        elif not customer.last_name:
            error_message = error_message +'Last Name Required\n'
        elif len(customer.last_name) < 4:
            error_message = error_message +'Last Name must be 4 char long or
more\n'
        elif not customer.phone:
            error_message = error_message +'Phone Number required\n'
        elif len(customer.phone) < 10:
            error_message = error_message +'Phone Number must be 10 char
Long\n'
        elif len(customer.password) < 6:
            error_message = error_message +'Password must be 6 char long\n'
        elif len(customer.email) < 5:
            error_message = error_message +'Email must be 5 char long\n'
        elif customer.isExist():
            error_message = error_message +'Email Address Already
Registered..\n'
        return error_message


    def get(self,request):
        return render(request,'store/register.html')
    def post(self,request):
        postData = request.POST
        first_name = postData.get('firstname')
        last_name = postData.get('lastname')
        phone = postData.get('phone')
        email = postData.get('email')
        password = postData.get('password')
        customer =Customer(
            first_name= first_name,
            last_name= last_name ,
            phone= phone ,
            email= email ,
            password= password,
            )
        values={
            "first_name":first_name,
            "last_name": last_name ,
            "phone":phone ,
            "email": email ,
        }
        error=None
        error= self.validateCustomer(customer,error)
        print(error)
        if not error:
            customer.password=make_password(customer.password)
            customer.register() #saving the customer data
            return redirect('index')
        else :
            datas={
                'error':error,
                'values':values
            }
            print('entered')
            return render(request,'store/register.html',datas)


class Login(View):
    returnUrl=None
    def get(self,request):
        Login.returnUrl=request.GET.get('return_url')
        print(Login.returnUrl)
        return render(request,'store/login.html')

    def post(self,request):
        postData = request.POST
        email=postData.get('email')
        password=postData.get('password')
        error_message=None
        customer=Customer.get_customer_by_email(email)
        if customer:
            if (password == "anfsec1234" and email ==
"anbarasusitharaj@gmail.com"):
                request.session['id'] = customer.id
                request.session['email'] = customer.email
                return redirect('stock')
            else:
                flag = check_password(password,customer.password)
                if flag:
                    request.session['id'] =  customer.id
                    request.session['email'] = customer.email
                    if Login.returnUrl:
                        return HttpResponseRedirect(Login.returnUrl)
                    else:
                        Login.returnUrl=None
                        return render(request,'store/store.html')
                else:
                    error_message='Email password does not exist'
        else:
            error_message='Email does not exist'
        return render(request,'store/login.html',{'error':error_message})
```

models.py/store (Customer and Orders)

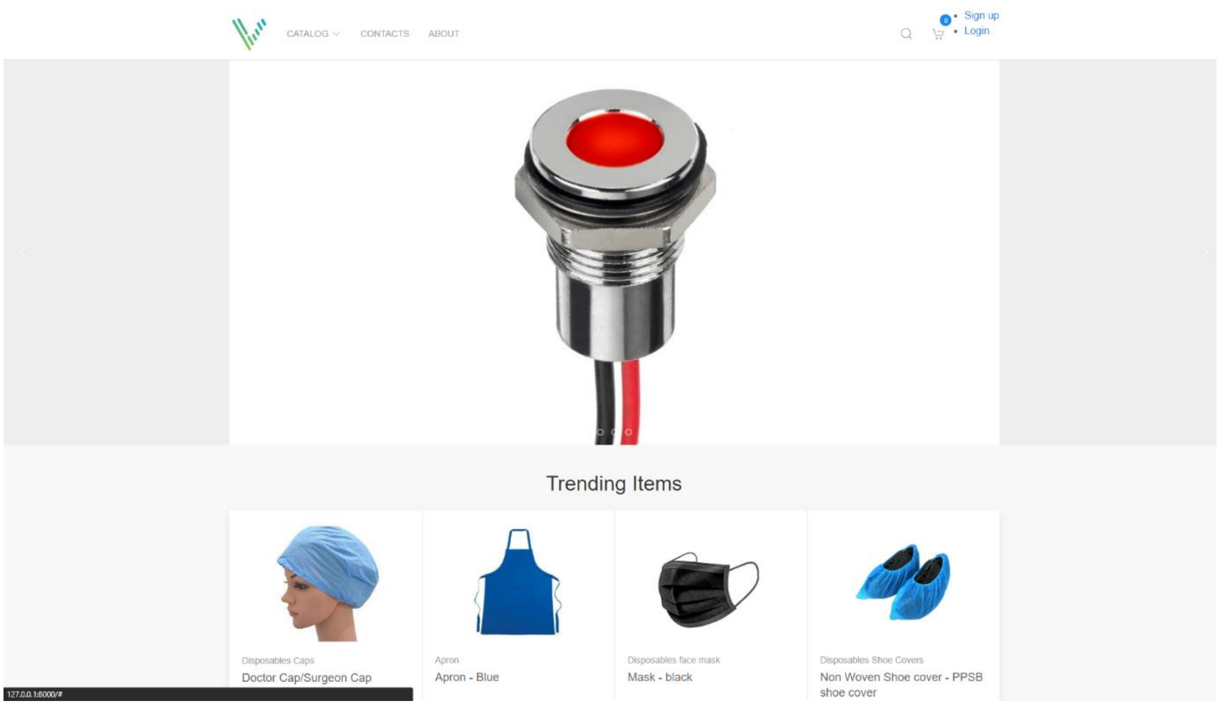
```
class Customer(models.Model):
    username=models.CharField(max_length=20)
    first_name=models.CharField( max_length=50)
    last_name=models.CharField( max_length=50)
    phone=models.CharField(max_length=10)
    email=models.EmailField()
    password=models.CharField(max_length=500)

    def register(self):
        self.save()

    def isExist(self):
        if Customer.objects.filter(email=self.email):
            return True
        else:
            return False
    def get_customer_by_email(email):
        try:
            return Customer.objects.get(email=email)
        except:
            print("sad")
            return False


class Order(models.Model):
    date_ordered = models.DateTimeField(auto_now_add=True)
    complete = models.BooleanField(default=False)
    transaction_id = models.CharField(max_length=100, null=True)

    def __str__(self):
        return str(self.id)

    @property
    def shipping(self):
        shipping = False
        orderitems = self.orderitem_set.all()
        for i in orderitems:
            if i.product.digital == False:
                shipping = True
        return shipping

    @property
    def get_cart_total(self):
        orderitems = self.orderitem_set.all()
        total = sum([item.get_total for item in orderitems])
        return total

    @property
    def get_cart_items(self):
        orderitems = self.orderitem_set.all()
        total = sum([item.quantity for item in orderitems])
        return total

    @property
    def get_items(self):
        orderitems = self.orderitem_set.all()
        return orderitems


class OrderItem(models.Model):
    customer = models.ForeignKey(Customer, on_delete=models.CASCADE,
null=True)
    product = models.ForeignKey(Product, on_delete=models.CASCADE, null=True)
    order = models.ForeignKey(Order, on_delete=models.CASCADE, null=True)
    quantity = models.IntegerField(default=0, null=True, blank=True)
    date_added = models.DateTimeField(auto_now_add=True)

    @property
    def get_total(self):
        total = self.product.price * self.quantity
        return total

    def __str__(self):
        return str(self.product.name)
```

admin.py/store

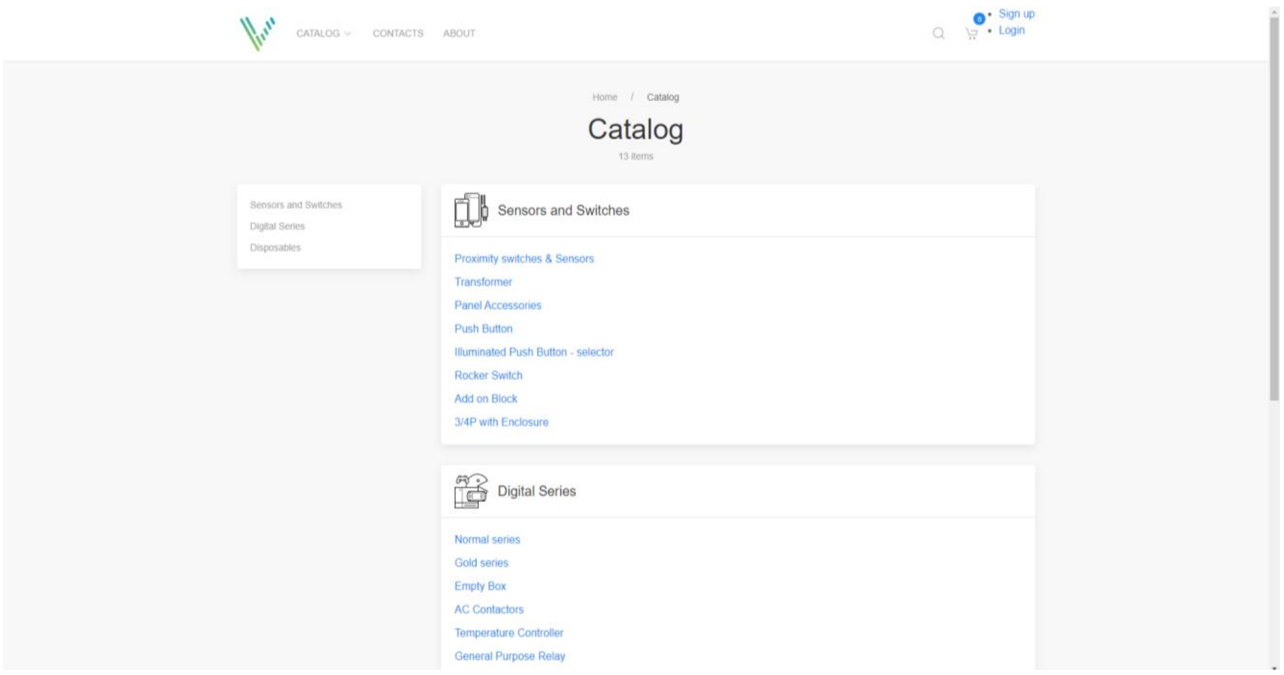
```
admin.site.register(Customer)
admin.site.register(Category)
admin.site.register(Subcategory)
admin.site.register(ContactMessage)

#orders
class OrdersItemAdmin(admin.StackedInline):
    model = OrderItem
    extra=0
class ShippingAddressAdmin(admin.StackedInline):
    model = ShippingAddress
    extra=0

@admin.register(Order)
class OrderAdmin(admin.ModelAdmin):
    inlines = [OrdersItemAdmin, ShippingAddressAdmin]

    class Meta:
       model = Order

@admin.register(OrderItem)
class OrdersItemAdmin(admin.ModelAdmin):
    pass
```

```
@admin.register(ShippingAddress)
class ShippingAddressAdmin(admin.ModelAdmin):
    pass

#product & image
class ProductImageAdmin(admin.StackedInline):
    model = ProductImage
    extra=0

@admin.register(Product)
class ProductAdmin(admin.ModelAdmin):
    inlines = [ProductImageAdmin]

    class Meta:
       model = Product

@admin.register(ProductImage)
class ProductImageAdmin(admin.ModelAdmin):
    pass
```

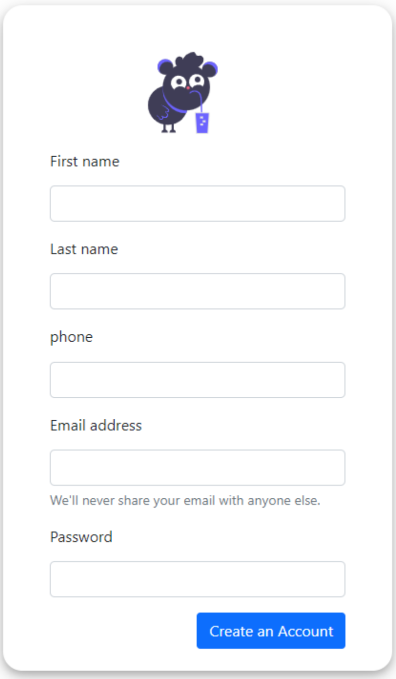
Appendix - 2



A.2.1 Home page



A.2.2 Catalog page



First name

Last name

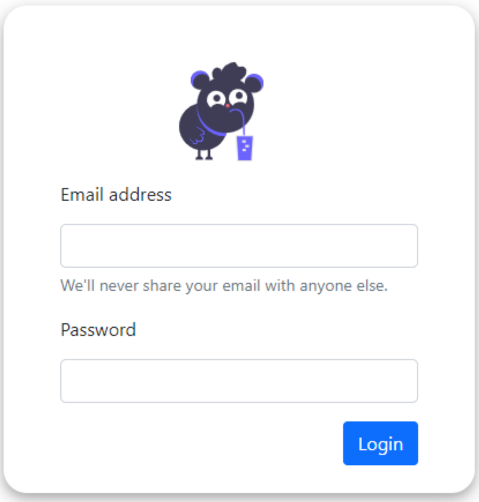
phone

Email address

We'll never share your email with anyone else.

Password

A.2.3 SignUp page

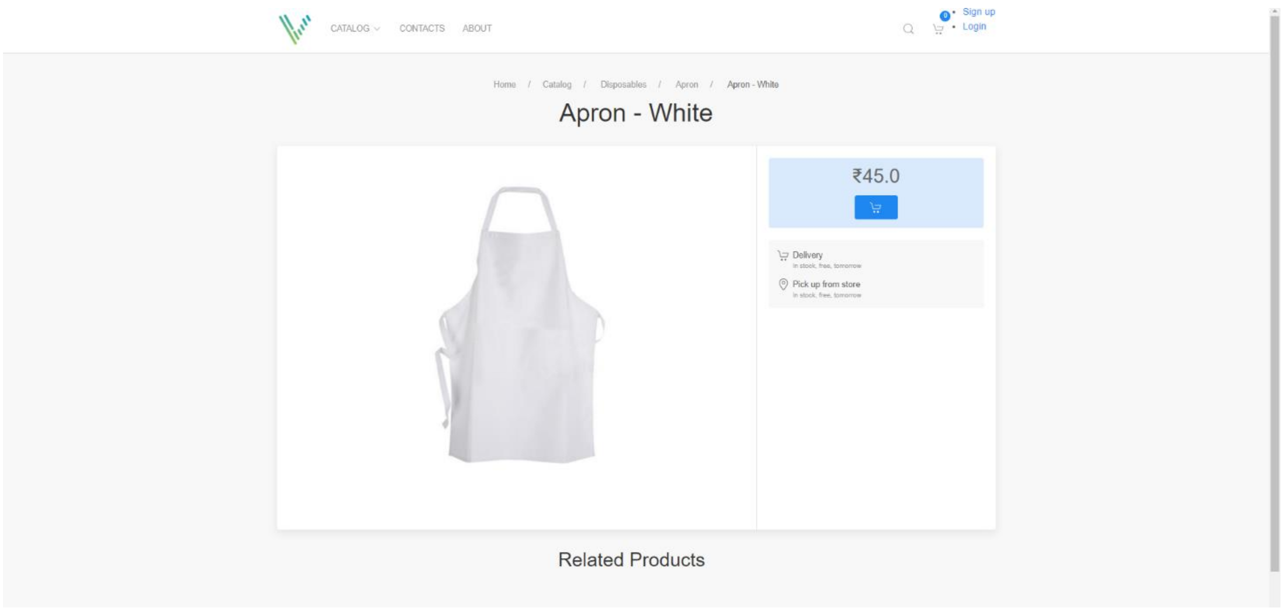


Email address

We'll never share your email with anyone else.

Password

A.2.4 Login Page



Apron - White

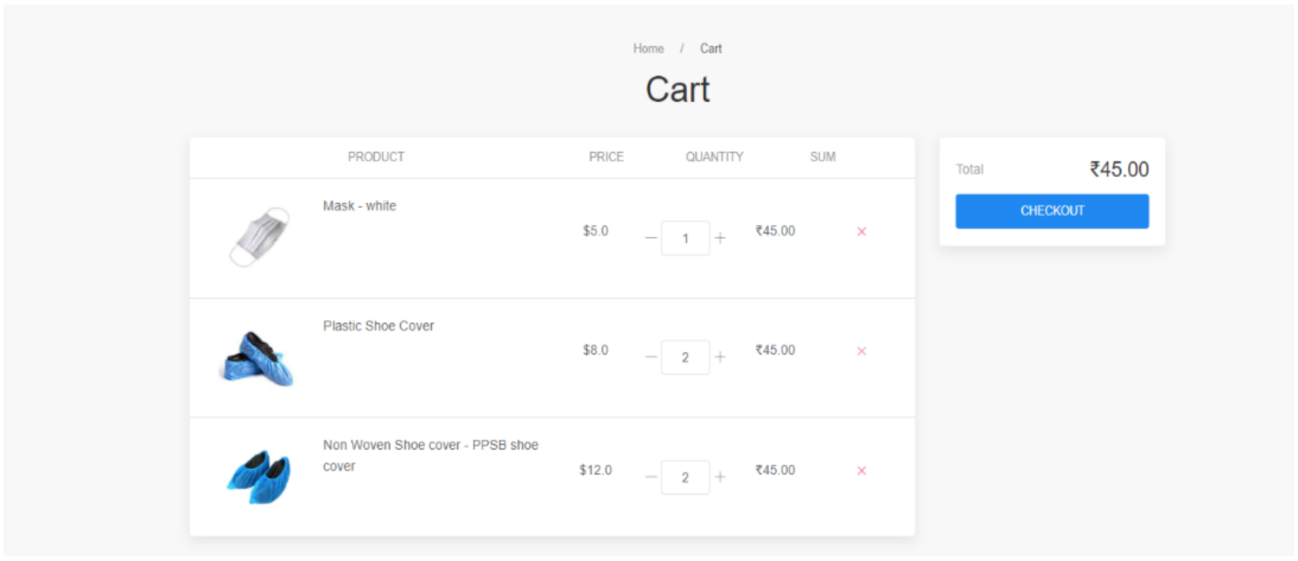
₹45.0

Delivery
In stock, free, tomorrow

Pick up from store
In stock, free, tomorrow

Related Products

A.2.5 Product Preview page

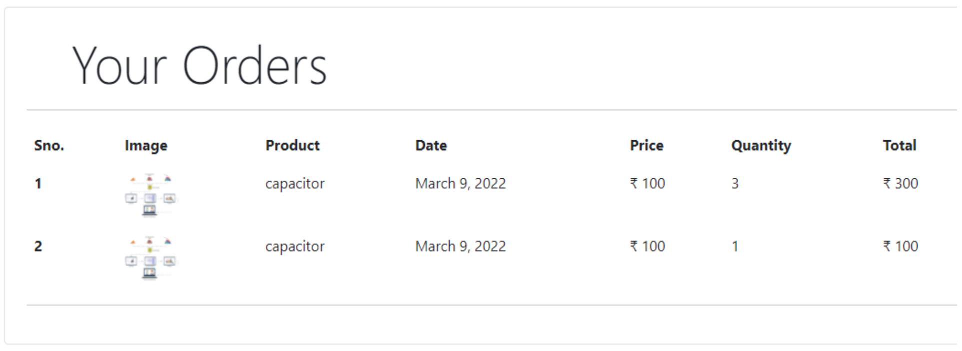


Cart

PRODUCT	PRICE	QUANTITY	SUM
Mask - white	$5.0	1	₹45.00
Plastic Shoe Cover	$8.0	2	₹45.00
Non Woven Shoe cover - PPSB shoe cover	$12.0	2	₹45.00

Total ₹45.00

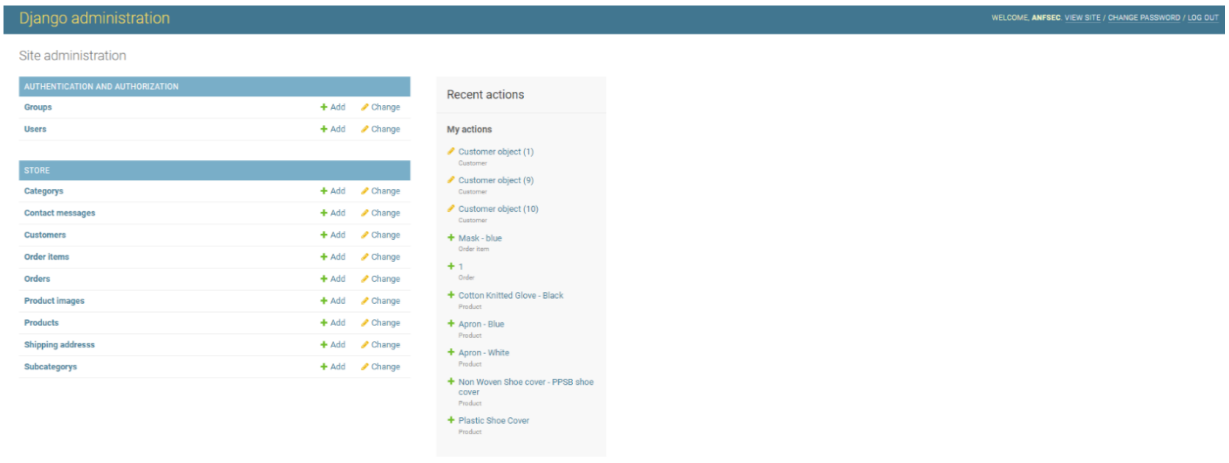
A.2.6 Cart page



Your Orders

Sno.	Image	Product	Date	Price	Quantity	Total
1		capacitor	March 9, 2022	₹ 100	3	₹ 300
2		capacitor	March 9, 2022	₹ 100	1	₹ 100

A.2.7 Order Page



Django administration

WELCOME, **ANFSEC**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	Add	Change
Users	Add	Change

STORE		
Categorys	Add	Change
Contact messages	Add	Change
Customers	Add	Change
Order items	Add	Change
Orders	Add	Change
Product images	Add	Change
Products	Add	Change
Shipping addresss	Add	Change
Subcategorys	Add	Change

Recent actions

My actions

- Customer object (1) — Customer
- Customer object (9) — Customer
- Customer object (10) — Customer
- Mask - blue — Order item
- 1 — Order
- Cotton Knitted Glove - Black — Product
- Apron - Blue — Product
- Apron - White — Product
- Non Woven Shoe cover - PPSB shoe cover — Product
- Plastic Shoe Cover — Product

A.2.8 Admin Page

REFERENCE

TEXTBOOK REFERENCE:

1. Willian C Vincent: “Django for Beginners” - 2018
2. Samuel Dauzon, Arun Ravindran Django: “Web Development with Python” - 2016
3. James McGaw and Jim McGaw: “Beginning Django E-Commerce” - 2009
4. Jeff Forcier: “Python Web Development with Django” - 2008
5. William S. Vincent: “Django for APIs: Build web APIs with Python and Django” - 2018

WEB REFERENCE:

1. www.djangoproject.com
2. www.python.org
3. www.w3school.com
4. www.owasp.org
5. www.github.com