

BS Software Engineering Final Year Project Report



Submitted By

MUHAMMAD SANAN	CU-481-2018
MUSTAFA HAIDER	CU-470-2018
HAMZA AYAZ	CU-524-2018

Supervised By

Engr. Zainab Jamil
Lecturer

DEPARTMENT OF COMPUTER SCIENCE
CECOS University of IT and Emerging Sciences Hayatabad Peshawar
July 2022

DATABOT (Hybrid Application)

Submitted By

Muhammad Sanan	CU-481-2018
Mustafa Haider	CU-470-2018
Hamza Ayaz	CU-524-2018

A Final Year Project Report submitted in partial fulfillment of the requirements for the degree of

BS Software Engineering

Final Year Project Report Supervisor

Engr. Zainab Jamil

Lecturer (Computer Science)

Supervisor Signature: _____

DEPARTMENT OF COMPUTER SCIENCE
CECOS University of IT and Emerging Sciences Hayatabad Peshawar
July 2022

ABSTRACT

Electronic Commerce is process of doing business through computer networks. A person sitting on his chair in front of a computer can access all the facilities of the Internet to buy or sell the products. Unlike traditional commerce that is carried out physically with effort of a person to go & get products, ecommerce has made it easier for human to reduce physical work and to save time. E-Commerce which was started in early 1990's has taken a great leap in the world of computers, but the fact that has hindered the growth of e-commerce is security. Security is the challenge facing e-commerce today & there is still a lot of advancement made in the field of security. The main advantage of e-commerce over traditional commerce is the user can browse online shops, compare prices and order merchandise sitting at home on their PC. For increasing the use of e-commerce in developing countries the B2B e-commerce is implemented for improving access to global markets for firms in developing countries. For a developing country advancement in the field of e-commerce is essential. The research strategy shows the importance of the e-commerce in developing countries for business applications.

In today's fast-changing business environment, it's extremely important to be able to respond to client needs in the most effective and timely manner. If your customers wish to see your business online and have instant access to your products or services.

This Project titled "DATABOT" provides the details of the product that is search by the user by one click. The software will gather the details of the specific product that the user need to buy in his range from different stores.

Keywords: Data Mining, Hybrid Application, Application Programming Interface, E-commerce Etc.

UNDERTAKING

I certify that research work titled DATABOT is our own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged / referred.

Muhammad Sanan	CU-481-2018	_____
Mustafa Haider	CU-470-2018	_____
Hamza Ayaz	CU-524-2018	_____

ACKNOWLEDGEMENTS

As a core of our belief (Iman), all praises to Allah Almighty Whose countless blessings are always showering on us. Alhamdulillah, another great blessing of Allah for granting us the ability and potential to complete this Final year project, which is an essential but partial fulfillment of requirements for the degree.

While accomplishing our project work on “DATABOT” We owe special thanks to Engr. Zainab Jamil our supervisor for this encouragement and facilitation. Our future life will always be enlightened with his fatherly way of education and communication.

Our special acknowledgment goes to our supervisor Engr. Zainab Jamil whose knowledge and guidance has enabled us to conduct and compile this project work. We would consider all our efforts as successful if this research may be of our help for future research workers for whom it is meant for.

Table of Contents

ABSTRACT	i
UNDERTAKING	ii
ACKNOWLEDGEMENTS	iii
CHAPTER-1	1
INTRODUCTION	1
1.1 Introduction.....	1
1.2 Problem Statement	1
1.3 Motivation	2
1.4 Aims and Objectives	2
1.5 Project Scope	3
1.6 Project Details.....	3
1.6.1 Tools	3
1.6.2 Framework	4
1.6.3 Languages	5
1.6.4 Database.....	6
1.6.5 Hardware.....	6
1.7 Implementation Issues and Challenges	6
CHAPTER-2	7
LITERATURE REVIEW	7
2.1 Literature Review.....	7
CHAPTER-3	9
REQUIREMENT ANALYSIS	9
3.1 Functional Requirements.....	9
3.1.1 Registration	9
3.1.2 Login Form	9
3.1.3 Forgot Password Form	9
3.1.4 Change Profile Settings	9
3.1.5 3th Party Products integration.....	10
3.1.6 View Products	10
3.1.7 Related Products.....	10
3.1.8 Search Product	10

3.1.9	Filter Products	10
3.2	Non-Functional Requirements.....	10
CHAPTER-4	12
LOGICAL DESIGN	12
4.1	Project Design.....	12
4.2	Process Model.....	12
4.2.1	Data Flow Diagram (DFD)	12
4.3	System Architecture	13
4.4	Use Cases.....	14
4.5	Flow Chart	15
4.6	Block Diagram.....	16
CHAPTER-5	17
PHYSICAL DESIGN	17
5.1	Database Design.....	17
CHAPTER-6	18
USER INTERFACE DESIGN	18
6.1	Registration.....	18
6.2	Login	18
6.3	Home Page Screen	19
6.4	Update Profile.....	20
6.5	Search Page.....	20
6.6	Shop Page	21
6.7	Men Page.....	21
6.8	Women Page	22
CHAPTER-7	23
CODING	23
7.1	Registration Functionality	23
7.2	Login Functionality.....	29
7.3	Products Slider.....	33
7.4	Forget Password.....	40
7.5	Product Details.....	44
7.6	Rendering Code	55

CHAPTER-8	58
SYSTEM TESTING	58
8.1 Testing	58
8.1.1 Usability Testing	58
8.1.2 Functionality Testing	58
8.1.3 Performance Testing.....	58
8.1.4 Compatibility Testing.....	59
8.1.5 Security Testing	59
CHAPTER-9	60
CONCLUSION AND FUTURE WORK	60
9.1 Conclusion	60
9.2 Future Work	60
REFERENCES	61

List of Figures

Figure 4.1 Data Flow Diagram	13
Figure 4.2 System Architecture.....	13
Figure 4.3 Use Cases Diagram.....	14
Figure 4.4 Flow chart Diagram	15
Figure 4.5 Block Diagram	16

CHAPTER-1

INTRODUCTION

1.1 Introduction

Just look the most valuable companies in the world. In 2006 these were the oil and energy companies. In 2012 the list included Google, Amazon and Facebook. The world's most valuable resource is no longer oil, but data. This Project provides the details of the product that is search by the user by one click. The software will gather the details of the specific product that the user need to buy in his range from different stores.

This project sets out to conduct research in the area of web mining and how it can be used as an application for extracting data of the products at different stores. The main focus of this project is to extracting the products details from different stores. To do this, a web scraper will be created to automate the process of extracting data of the products from different stores. Besides the scraping of prices, it can extract all product data: descriptions, image URLs, metadata, etc. And that data is automatically saved in Database. While figuring out the perfect price range is important, prices are directly related to the production process and details of your product. Scraping outfitters and other stores product through databot includes price information as well as the product description. Databot will gather the product information as Brand Name, Product Price, Product Attributes, Product Description and Product Name.

1.2 Problem Statement

In Pakistan there are too many e-commerce stores and there are also millions of products available in different market place. For a user it is difficult to check all the sites and compare their products. It is time consuming process for the user to find the products within their range from a different stores. There is no proper application that gathers all the Local brands products from different stores.

The problem arises when the user uses different stores for product listing as well as a specific product. There are multiple problems that the user faces in searching of a specific product where

the prices are different from each product at different stores, where the user wants the details of the product from multiple stores, so therefore, it is difficult and also time consuming for the customer to find the details of the product in a different store, due to which, it also requires lots of efforts. There is no generic web application that gathers the details of products on a single platform.

1.3 Motivation

E-commerce stores in Pakistan are increasing day by day and many of them are not making their ways to reach to the present era technologies and get some awareness on these new technologies that have not been yet explored by them. Only few e-commerce stores making the use of some advanced level technologies to operate their operations and tasks. However, there are still some issues that they might face and are still not capable to correct them and get the best of their outputs. We were motivated with the fact that there is a need to reach to these all Ecommerce stores and help them by making most of the use of latest technologies available and making a fast, efficient, and a secure system that can carry out their day to day tasks easily providing them with accurate and effective results.

1.4 Aims and Objectives

In our experience, price intelligence is the biggest use case for web mining. Extracting product and pricing information from e-commerce websites, then turning it into intelligence is an important part of modern e-commerce companies that want to make better pricing/marketing decisions based on data. The solution that thought of is that building software that gathers the details of specific product from different stores. The details are automatically saved in the database due to which it is less time consuming, minimum effort on the software as a result they will be more accurate. There isn't such an application that gathers the details of product at one time. If the user searches for the specific product that user wants to purchase, user will definitely search the different stores manually for the details which is very timing consuming. Following are the objectives

- To search product details from different stores automatically.
- Gather details of product effectively and efficiently with minimum effort in minimum time.
- Redirect the customer to specific store to purchase the desired item.

1.5 Project Scope

Following are the scope of the developed system.

- User can registered them self and login to the system.
- User is allow to update their profile information.
- Third party products integration.
- User can see all the products that are coming from APIs.
- Display detail information of the selected product.
- Related Products has been shown with selected product.
- If user buy product that will redirected to their original website.
- Mostly products will be shown on the basis of user gender.
- User can search products.
- Products Categorization.
- User can filter the products on the basis of gender, brand, price and category.

1.6 Project Details

In this section we present the architecture of our proposed solution as well as the tools required to implement the system. Additionally, we provide details of challenges and final deliverable of the project.

1.6.1 Tools

Visual Studio Code, Heroku, Netlify, Github and Git.

Visual Studio Code, also commonly referred to as VS Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it.

Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python,

PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages. Heroku was acquired by Salesforce in 2010 for \$212 million.

Netlify is a San Francisco–based cloud computing company that offers hosting and serverless backend services for web applications and static websites. The company provides hosting for websites whose source files are stored in the version control system Git and then generated into static web content files served via a Content Delivery Network. Given the limitations of the purely static model, the company later expanded services to include content management systems, and features of serverless computing to handle websites with interactive features.

GitHub Inc is a provider of Internet hosting for software development and version control using git. It offers the distributed version control and source code management functionality of git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuous integration, and wikis for every project. Headquartered in California, it has been a subsidiary of Microsoft since 2018. It is commonly used to host open-source projects. As of June 2022, GitHub reports having over 83 million developers and more than 200 million repositories (including at least 28 million public repositories). It is the largest source code host as of November 2021.

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

1.6.2 Framework

ReactJS, ExpressJS and NodeJS.

React JS is a declarative, efficient, and flexible JavaScript library for building user interfaces. It's 'V' in MVC. ReactJS is an open-source, component-based frontend library responsible only for the view layer of the application. It is maintained by Facebook. React uses a declarative paradigm that makes it easier to reason about your application and aims to be both efficient and

flexible. It designs simple views for each state in your application, and React will efficiently update and render just the right component when your data changes. The declarative view makes your code more predictable and easier to debug.

ExpressJS or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

Node JS is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36. The definition of Node.js as supplied by its official documentation is as follows – Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. Node.js is an open source, cross platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

1.6.3 Languages

HTML, CSS and JavaScript.

HTML Stand for Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheet (CSS) and scripting language such as JavaScript.

CSS Stands for “Cascading Style sheet” cascading style sheet are used to format the layout of web pages. This can be used to define text styles, table size, and other aspects of web pages that previously could only be defined in a page’s HTML. CSS helps Web developer create a uniform look across several pages of a web sites. Instead of defining the style of every page.

JavaScript often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries.

1.6.4 Database

MongoDB is an open source, Non-Relational database management system (DBMS) that uses flexible documents instead of tables and rows to process and store various forms of data. As a NoSQL solution, MongoDB does not require a relational database management system (RDBMS), so it provides an elastic data storage model that enables users to store and query multivariate data types with ease. This not only simplifies database management for developers but also creates a highly scalable environment for cross-platform applications and services. MongoDB documents or collections of documents are the basic units of data. Formatted as Binary JSON (Java Script Object Notation), these documents can store various types of data and be distributed across multiple systems. Since mongoDB employs a dynamic schema design, users have unparalleled flexibility when creating 6 data records, querying document collections through mongoDB aggregation and analyzing large amounts of information.

1.6.5 Hardware

Personal Computers, Desktops, Laptops, Mobile Phones.

1.7 Implementation Issues and Challenges

Our main goal is to design, implement and test a Web application for Pakistani users. Hence we face some challenges in order to meet our objective. Some of these challenges are.

- Gathering information of clothing brands products data of Pakistan.
- Implementing Data Mining Algorithm for related products.
- Integrating email to reset password using Gmail.
- Requires a developer with multiple expertise.
- Reliability on libraries of third-parties.
- Forced to execute custom designs in the native language.

CHAPTER-2

LITERATURE REVIEW

2.1 Literature Review

Farfetch [1] is a British-Portuguese online luxury fashion retail platform that sells products from over 700 boutiques and brands from around the world.[1] The company was founded in 2007 by the Portuguese entrepreneur José Neves with its headquarters in London and main branches in Lisbon and Porto. There are many offices worldwide in Guimarães, Braga, New York, Los Angeles, Tokyo, Shanghai, Hong Kong, São Paulo, Dubai, New Delhi and Moscow. The e-commerce company operates local-language websites and mobile apps for international markets in English, Spanish, French, Japanese, Chinese, Arabic, German, Portuguese, Korean, Italian and Russian. Farfetch has offices in 14 cities and employs over 4,500 staff.

ZASIMO Pakistan's [2] largest online destination for the latest and best high street and international fashion, footwear and accessory brands, providing customers fast order fulfilment, an attractive and personalized shopping experience with exceptional customer service. ZASIMO will continue to expand its unique offering of the best fashion brands from the UK, USA and Europe from season to season, bringing our customers an ever increasing range of fashion styles and top brands, delivered next day. Customers will have access to a continually growing range of brands and product categories via online store and mobile app. ZASIMO will offer local customer support, fast and secure deliveries and multiple payment options for customer convenience.

Lyst [3] operates an inventory free platform, and uses a proprietary data engine to provide each customer with a highly personalized shopping experience. Orders are fulfilled by the thousands of brands and stores which partner with Lyst. Brands available on lyst include Balenciaga, Balmain, Bottega Veneta, Burberry, Fendi, Gucci, Moncler, Off-White, Prada, Saint Laurent and Valentino. We're a mix of fashion lovers, technologists and data experts. Two thirds of lyst talent are engineers, scientists and product people. We're focused on ways technology can delight both our customers and our brand & store partners. Lyst is also a unique source of

fashion data insights, and is responsible for the lyst Index, the industry leading report of the hottest brands and products worldwide. Lyst was founded in London in 2010, and is backed by investors including certain funds managed by Fidelity International, C4 Ventures, Draper Esprit, Steadfast Financial, Balderton Capital and LVMH.

Applications	Targeted Brands	Platforms
FARFETCH [1]	Alexander McQueen, Balenciaga, <u>Bottega Veneta</u> , Gucci, Givenchy, Prada, Saint Laurent, Versace	Web Application
ZASIMO [2]	New Look, Burton, <u>Jack&Jones</u> , <u>Quizman</u> , <u>Bravl</u> Soul, Only, Blend	Web Application
LYST [3]	Nike, Gucci, <u>Moncler</u> , Barbour, Diesel, The north face, <u>Kenzo</u> , <u>Balmain</u>	Web Application
DATABOT	J., <u>Almirah</u> , <u>Heels</u> , <u>Leisure Club</u> , <u>Metro</u> , <u>Ndure</u> , <u>Shoeplant</u> , <u>urbansole</u> , <u>Servis</u> , <u>Edenrobe</u> , <u>Bareeze</u> , <u>Breakout</u> , <u>Sanasafinaz</u> , <u>Outfitters</u> . <u>Alkaram</u> etc....	Web, Android



FARFETCH

CHAPTER-3

REQUIREMENT ANALYSIS

3.1 Functional Requirements

A Functional Requirement (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Following are the some top most functional requirement of “DATABOT” that clearly elaborate all the functional requirements of our application.

3.1.1 Registration

The application will have a registration form which user can register themselves by their username, email, gender and password.

3.1.2 Login Form

The application will have a login form which will make sure that only authentic and valid users can access the application by entering their email and password. So on the bases of gender when male is login the application will show the data of all male products or if female is login then the application will show all the products of female.

3.1.3 Forgot Password Form

The application will have forgot password form which will ask the user to make a request for a new password if the user is unable to remember his password. This request will be sent to the user E-mail and then further action will be taken.

3.1.4 Change Profile Settings

The Profile Settings Form will be available for the users so that they can update their personal information.

3.1.5 3th Party Products integration

Our application have no their own products. It will integrate the products from different stores API. By calling product API of different stores in our application the products is display all the information of Products such as Brand name, Categories, Availability, Gender, Price and Title.

3.1.6 View Products

User can view all the products which will scrap from the different stores. User can also view the detail information of product which will scrap from different stores such as Brand name, Categories, Availability, Gender, Price and Title.

3.1.7 Related Products

Our application uses the data mining technique called (Association) for the related product so when the user is buying some product the application will also show the products related to that buying product so there is a chance the user will interested in buying that related product too.

3.1.8 Search Product

User can search the products and when they click on search product they will redirect to the next page where all the product is display to the user on the basis of keyword which the user searched.

3.1.9 Filter Products

User can filter the products on the basis of Gender, Brand, Price and Category.

3.2 Non-Functional Requirements

Non-Functional Requirement (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of non-functional requirement, “how fast does the website load?” Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

Here in this section we clearly mentioned some basic non-functional requirements of our application and we applied it.

- Robustness.
- Flexibility.
- User Friendly.

- Usability.
- Portability.
- Availability.
- Maintainability.
- Security.
- Performance.
- Scalability.
- Adaptability.
- Configurability.

CHAPTER-4

LOGICAL DESIGN

4.1 Project Design

In order to design an application, the database must be designed first. Conceptual design can be divided into two parts: The data model and the process model. The data model focuses on what data should be stored in the while the process model deals with how the data is processed. To put this in the context of the database, the data model is used to design the tables. The process model is used to design the queries that will access and perform operations on those tables.

4.2 Process Model

A Process Model tells us about how the data is processed and how the data flows from one table to another to gather the required information. This model consists of the Functional Decomposition Diagram and Data Flow Diagram.

4.2.1 Data Flow Diagram (DFD)

Data Flow Diagrams show the flow of data from external entities into the system, and from one process to another within the system. There are four symbols for drawing a DFD:

- Rectangles representing external entities, which are sources or destinations of data.
- Ellipses representing processes, which take data as input, validate and process it and output it.
- Arrows representing the data flows, which can either, be electronic data or physical items.
- Open-ended rectangles or a Disk symbol representing data stores, including electronic stores such as databases or JSON files and physical stores such as filing cabinets or stacks of paper.

Data Flow Diagrams for the current system. Each process within the system is first shown as a Context Level DFD and later as a Detailed DFD. The Context Level DFD provides a conceptual view of the process and its surrounding input, output and data stores. The Detailed DFD provides a more detailed and comprehensive view of the interaction among the sub-processes within the system.

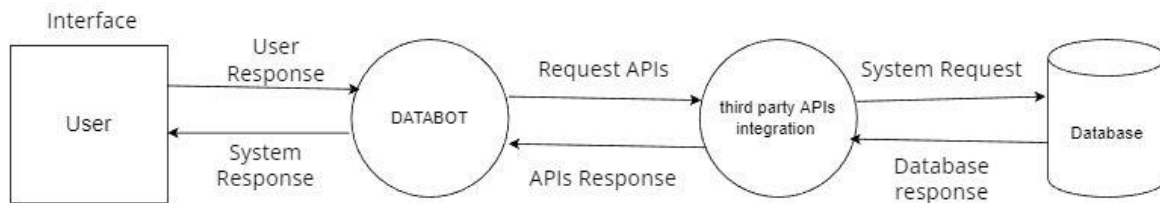


Figure 4.1 Data Flow Diagram

4.3 System Architecture

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

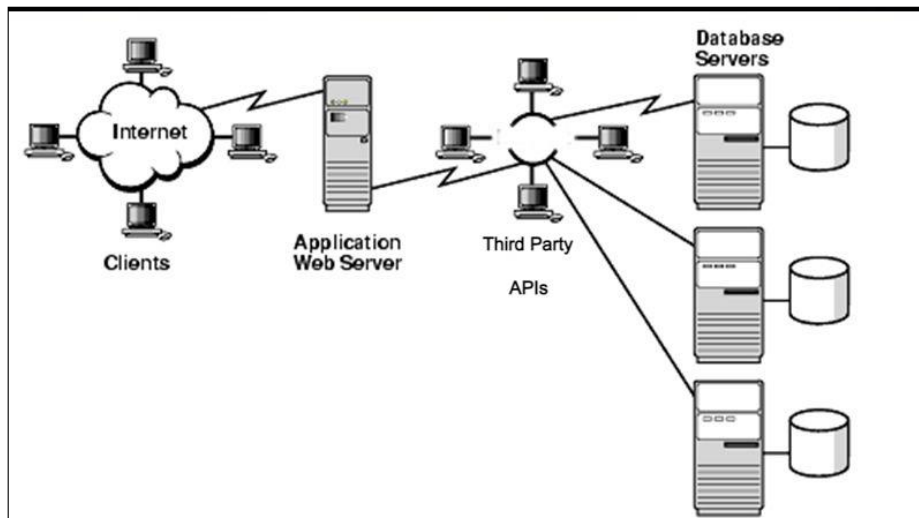


Figure 4.2 System Architecture

4.4 Use Cases

In software and systems engineering, the phrase use case is a polysemy with two senses: A usage scenario for a piece of software; often used in the plural to suggest situations where a piece of software may be useful. A potential scenario in which a system receives an external request and responds to it.

A use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language (UML) as an actor) and a system to achieve a goal. The actor can be a human or other external system.

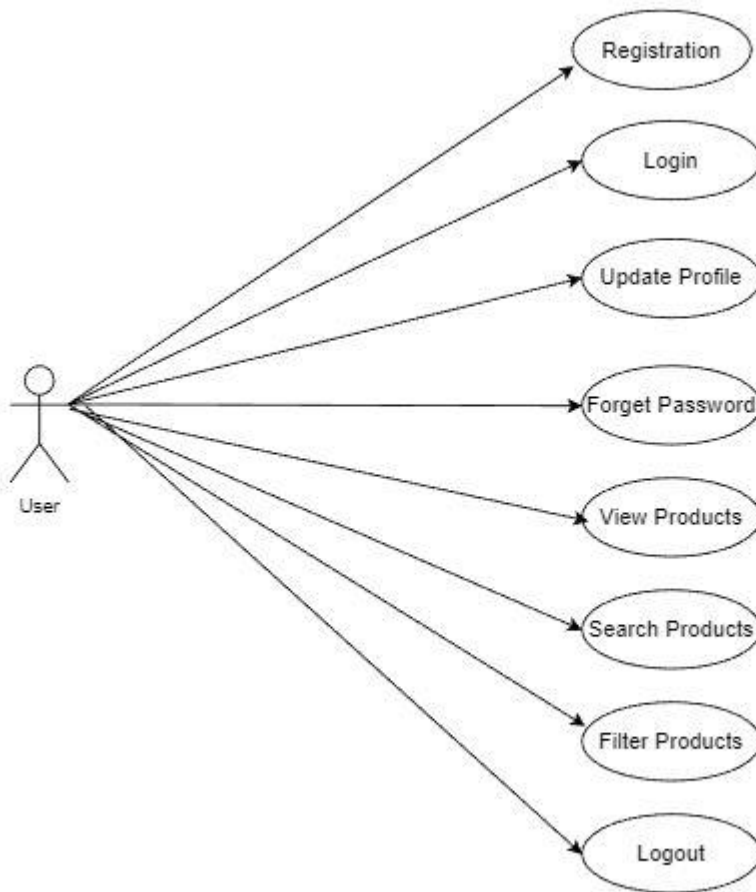


Figure 4.3 Use Case Diagram

4.5 Flow Chart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

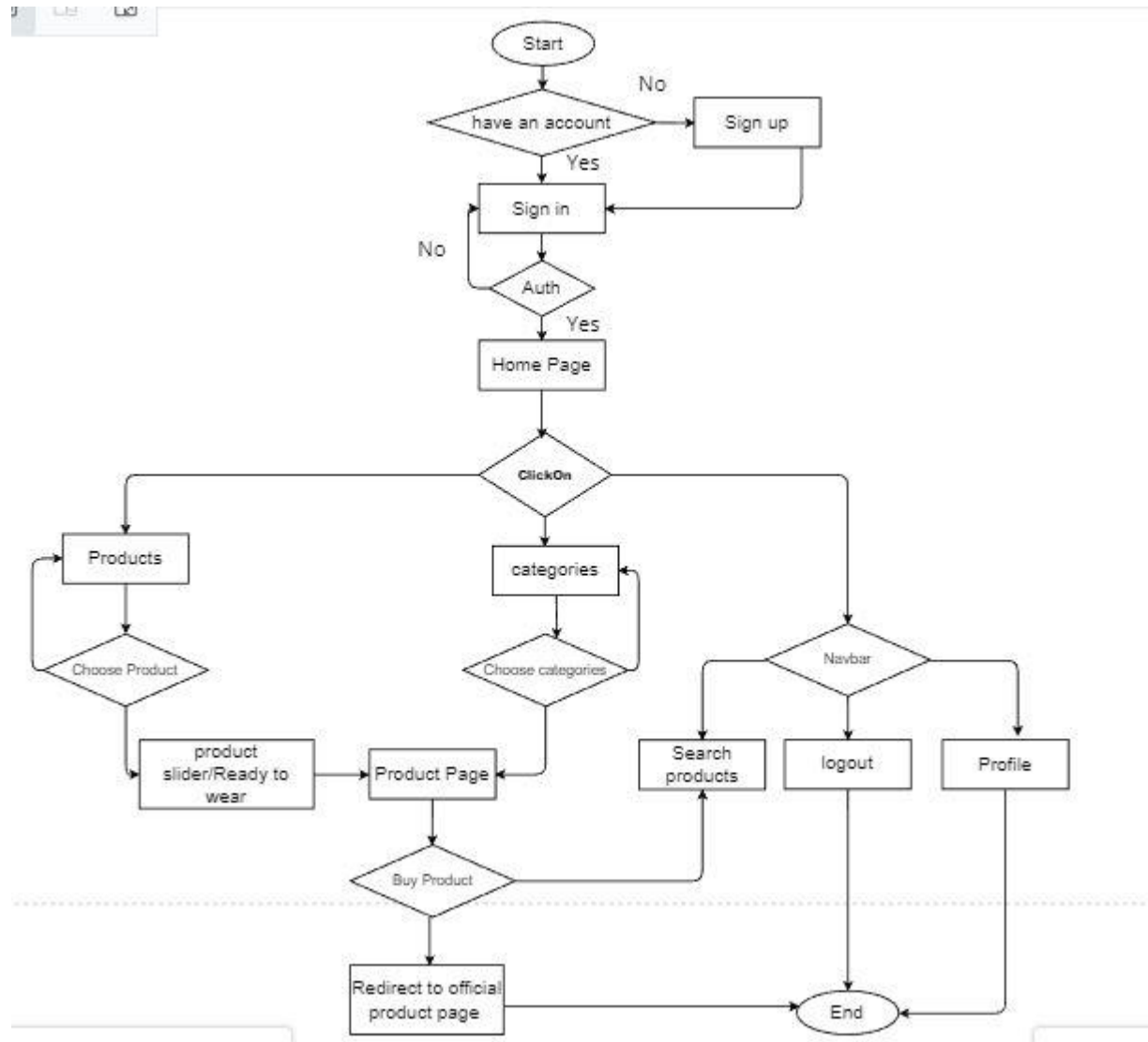


Figure 4.4 Flow Chart Diagram

4.6 Block Diagram

A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic design, software design, and process flow diagrams.



Figure 4.5 Block Diagram

CHAPTER-5

PHYSICAL DESIGN

5.1 Database Design

- The database used is mongoDB because it is characterized as a free, fast, reliable open source relational database.
- It lacks some facilities, but it has an active development team and, as it goes from release to release, more capabilities are added.
- This database is chosen because of its unique storage engine architecture mongoDB performance is very high and lightweight.
- Supports a large number of embedded applications which makes mongoDB very flexible.

```
mongo.Schema({
  username:{
    type:String,
    require:[true,"Please fill the username field"],
    trim:true,
    minlength:[3,"Minimum three letters required"],
    maxlength:15,

  },

  email:{
    type:String,
    require:[true,"Please fill the email field"],
    unique: [true, "email already exists in database!"],
    trim:true,
    match: [/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/, 'Please enter a valid email address'],


  },
  password:{
    type:String,
    require:[true,"Please fill the password field"],
    trim:true,
    minlength:[9,"Minimum nine letters required"],
    maxlength:15,
  },
  gender:{
    type:String,
    require:[true,"Please select the gender field"],
    //enum:["male", "female"],
  }
})
```

CHAPTER-6

USER INTERFACE DESIGN

Following are some snapshots of “DATABOT”

6.1 Registration



Sign up

Username *

Email Address *

Format: example@yahoo.com

Password *

Confirm Password *

Gender


☐ Female ☐ Male

SIGN UP

[Already have an account? Sign in](#)

Activate Windows
Go to Settings to activate Windows

6.2 Login



Sign in

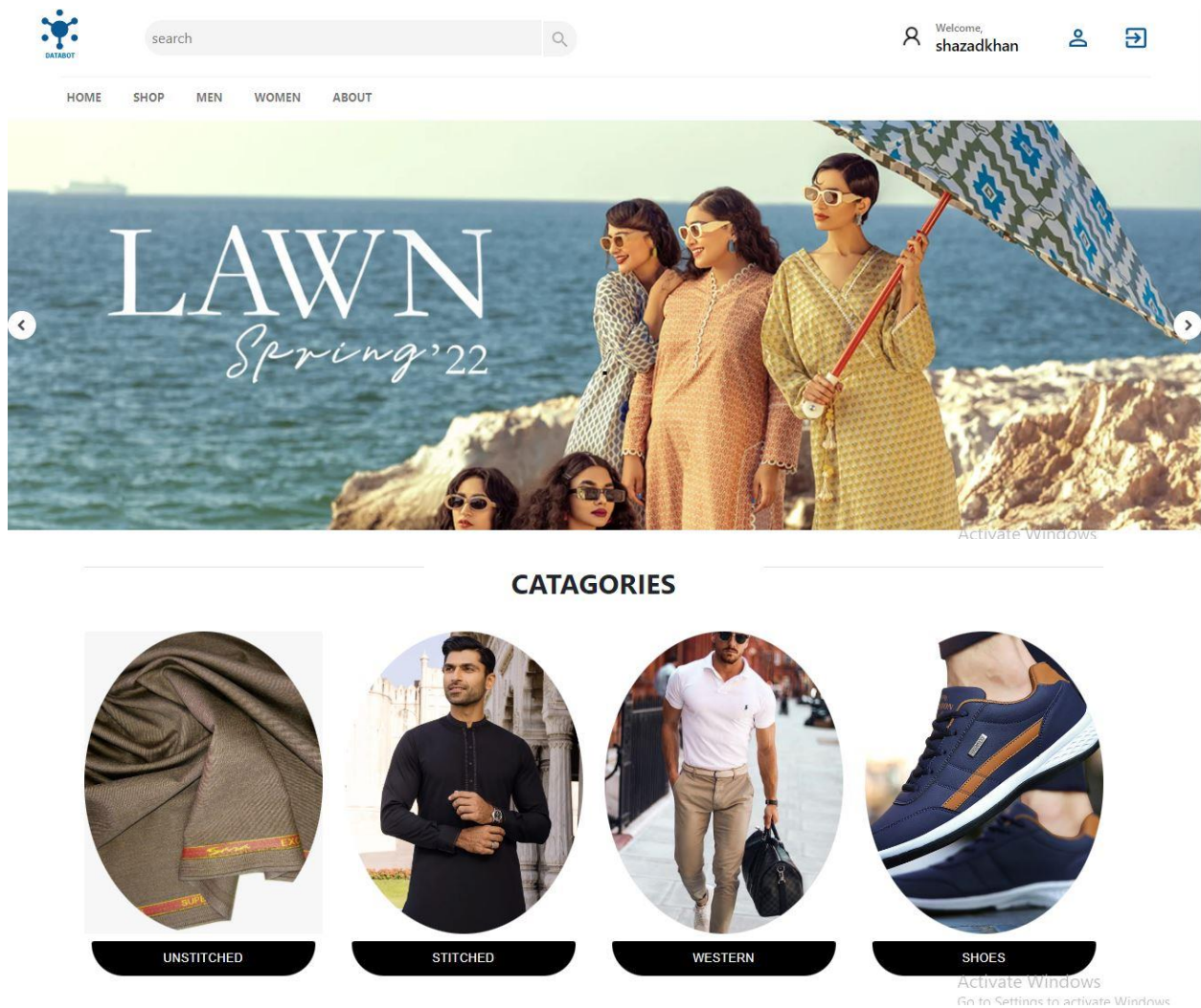
Email Address *

Password *


SIGN IN

[Don't have an account? Sign Up](#) [Forget Password ?](#)

6.3 Home Page Screen




6.4 Update Profile



Welcome, kinzakhan

HOME SHOP MEN WOMEN ABOUT


Profile

User Name *

kinzakhan

Email Address *

kinzakhan1998@gmail.com

Password *


Confirm Password *

☐ Show Password

UPDATE PROFILE

Activate Windows
Go to Settings to activate Windows.

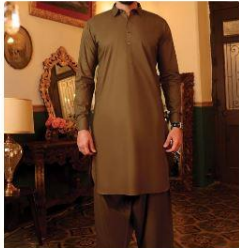
6.5 Search Page



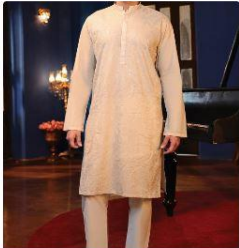
Welcome, sanankhan16

HOME SHOP MEN WOMEN ABOUT


Search Results : lawn




STITCHED
Dyed Lawn Shalwar Kameez
PKR 6392
ALKARAM STUDIO



STITCHED
Dyed Embroidered Lawn Kurta
PKR 4136
ALKARAM STUDIO




STITCHED
Dyed Embroidered Lawn Kurta
PKR 4136
ALKARAM STUDIO



STITCHED
Dyed Embroidered Lawn Kurta
PKR 3196
ALKARAM STUDIO

Activate Windows
Go to Settings to activate Windows.

6.6 Shop Page



Welcome, kinzakhan

HOME SHOP MEN WOMEN ABOUT

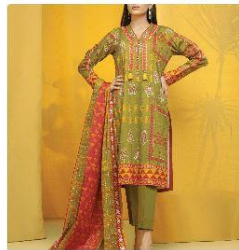
Gender
female

Brand
All

category
All

Price
All

SHOP



UNSTITCHED
EWU22A1-23008 - Mehndi - 3
Piece



UNSTITCHED
EWU22A1-20722 - Maroon - 3
Piece




UNSTITCHED
EWU22A1-20773 - Light
Brown - 3 Piece



UNSTITCHED
EWU22A1-23468 - Yellow - 3
Piece

6.7 Men Page



Welcome, kinzakhan

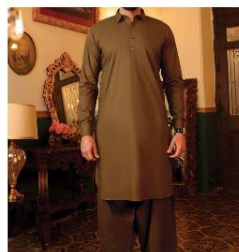
HOME SHOP MEN WOMEN ABOUT

Brand
All

category
All

Price
All

MEN



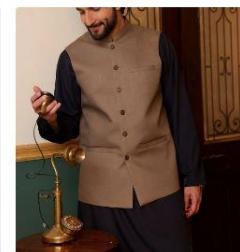
STITCHED
Dyed Lawn Shalwar Kameez
PKR 6392



STITCHED
Dyed Cambric Shalwar
Kameez




STITCHED
Dyed Cambric Shalwar
Kameez



STITCHED
Dyed Casual Waistcoat
PKR 6392



6.8 Women Page



search

Q

Welcome,
kinzakhan




HOME SHOP MEN WOMEN ABOUT

Brand: Alkaram Studio


category: All

Price: All


WOMEN




UNSTITCHED
3 Pc Embroidered Lawn Suit
With Organza Dupatta



UNSTITCHED
3 Pc Embroidered Lawn Suit
With Chiffon Dupatta



UNSTITCHED
3 Pc Embroidered Lawn Suit
With Chiffon Dupatta



UNSTITCHED
3 Pc Embroidered Lawn Suit
With Chiffon Dupatta

Activate Windows
Go to Settings to activate Windows.

CHAPTER-7

CODING

Following is the code of some functionalities and modules.

7.1 Registration Functionality

```
import React, { useState, useEffect } from "react";
import logo from "./logo.png";
import axios from "axios";
import Avatar from "@material-ui/core/Avatar";
import Button from "@material-ui/core/Button";
import CssBaseline from "@material-ui/core/CssBaseline";
import TextField from "@material-ui/core/TextField";
import FormControlLabel from "@material-ui/core/FormControlLabel";
import Checkbox from "@material-ui/core/Checkbox";
import Grid from "@material-ui/core/Grid";
import LockOutlinedIcon from "@material-ui/icons/LockOutlined";
import Typography from "@material-ui/core/Typography";
import { makeStyles } from "@material-ui/core/styles";
import Container from "@material-ui/core/Container";
import { useNavigate, Link, Navigate } from "react-router-dom";
import API from "./api";
import {
  Radio,
  FormControl,
  FormLabel,
  RadioGroup,
  Snackbar,
} from "@material-ui/core";
import MuiAlert from "@material-ui/lab/Alert";

const useStyles = makeStyles((theme) => ({
  paper: {
    marginTop: theme.spacing(2),
    display: "flex",
    flexDirection: "column",
    alignItems: "center",
  },
  avatar: {
```



```

    margin: theme.spacing(1),
    backgroundColor: theme.palette.secondary.main,
  },
  form: {
    width: "100%", // Fix IE 11 issue.
    marginTop: theme.spacing(3),
  },
  submit: {
    margin: theme.spacing(3, 0, 2),
  },
});

```

```

export default function SignUp() {
  const classes = useStyles();

  const [msg, setmsg] = useState(null);
  const [open, setopen] = useState(false);
  const [data, setdata] = useState({
    username: "",
    email: "",
    password: "",
    cpassword: "",
    gender: "",
  });

```

```

  const change = (e) => {
    setdata({
      ...data,
      [e.target.name]: e.target.value,
    });
  };

```

```

  let history = useNavigate();
  useEffect(() => {
    if (localStorage.getItem("user")) {
      history("/homepage");
    }
  }, []);
  const submit = (e) => {
    e.preventDefault();

    if (data.password !== data.cpassword) {

```

```

    setmsg("Password Is Not Match");
    console.log("Password Is Not Match");
    setopen(true);
  } else if (data.gender == "") {
    setmsg("Select the gender.");

    setopen(true);
  } else {
    // console.log(data);

    axios.post(`${API}/userregistration`, data).then((res) => {
      setmsg(res.data);
      //console.log(res.data)
      setopen(true);
      if (res.data == "User Registered") {
        history("/");
      }
    });
  }
};
if (localStorage.getItem("user")) {
  return <Navigate to="/homepage" replace />;
}
const handleClick = () => {
  setopen(false);
};
return (
  <Container component="main" maxWidth="xs">
    <div className={classes.paper}>
      <img src={logo} style={{ width: 110 }} />

      <Typography component="h1" variant="h5">
        Sign up
      </Typography>
      <form onSubmit={submit} className={classes.form}>
        <Grid container spacing={2}>
          <Grid item xs={12}>
            <TextField
              autoComplete="Username"
              name="username"
              variant="outlined"
              required

```

```

        inputProps={{ maxLength: 25, minLength: 5 }}
        value={data.username}
        onChange={change}
        fullWidth
        id="username"
        label="Username"
      />
    </Grid>

```

```

    <Grid item xs={12}>
      <TextField
        variant="outlined"
        required
        fullWidth
        value={data.email}
        onChange={change}
        id="email"
        label="Email Address"
        name="email"
        helperText="Formet: example@yahoo.com"
        type="email"
        autoComplete="email"
      />
    </Grid>

```

```

    <Grid item xs={12}>
      <TextField
        variant="outlined"
        required
        value={data.password}
        onChange={change}
        fullWidth
        name="password"
        label="Password"
        type="password"
        inputProps={{ maxLength: 25, minLength: 9 }}
        id="password"
        autoComplete="current-password"
      />
    </Grid>

```

```

    <Grid item xs={12}>
      <TextField

```

```

variant="outlined"
required
fullWidth
value={data.cpassword}
onChange={change}
name="cpassword"
label="Confirm Password"
type="password"
id="cpassword"
inputProps={{ maxLength: 25, minLength: 9 }}
autoComplete="current-password"
/>
</Grid>

<FormControl component="fieldset">
  <FormLabel component="legend">Gender</FormLabel>
  <RadioGroup
    aria-label="gender"
    name="gender"
    value={data.gender}
    onChange={change}
  >
    <Grid container>
      <Grid item xs={6}>
        <FormControlLabel
          value="women"
          control={<Radio />}
          label="Female"
        />
      </Grid>

      <Grid item xs={6}>
        <FormControlLabel
          value="men"
          control={<Radio />}
          label="Male"
        />
      </Grid>
    </Grid>
  </RadioGroup>
</FormControl>
</Grid>

```

```

<Button
  type="submit"
  fullWidth
  variant="contained"
  color="primary"
  className={classes.submit}
>
  Sign Up
</Button>
<Grid container justifyContent="flex-end">
  <Grid item>
    <Link
      to="/"
      variant="body2"
      style={{ color: "#0A5995", textDecoration: "none" }}
    >
      Already have an account? Sign in
    </Link>
  </Grid>
</Grid>
</form>
</div>

<Snackbar open={open} autoHideDuration={4000} onClose={handleClick}>
  {msg == "User Registered" ? (
    <MuiAlert severity="success" elevation={6} variant="filled">
      {msg}
    </MuiAlert>
  ) : (
    <MuiAlert severity="error" elevation={6} variant="filled">
      {msg}
    </MuiAlert>
  )}
</Snackbar>
<br />
<br />
<br />
</Container>
);
}

```

7.2 Login Functionality

```
import React, { useEffect, useState } from "react";
import Avatar from "@material-ui/core/Avatar";
import Button from "@material-ui/core/Button";
import CssBaseline from "@material-ui/core/CssBaseline";
import TextField from "@material-ui/core/TextField";
import FormControlLabel from "@material-ui/core/FormControlLabel";
import Checkbox from "@material-ui/core/Checkbox";
import Grid from "@material-ui/core/Grid";
import Box from "@material-ui/core/Box";
import MuiAlert from "@material-ui/lab/Alert";
import { Snackbar } from "@material-ui/core";
import LockOutlinedIcon from "@material-ui/icons/LockOutlined";
import Typography from "@material-ui/core/Typography";
import { makeStyles } from "@material-ui/core/styles";
import Container from "@material-ui/core/Container";
import axios from "axios";
import API from "../api";
import {
  BrowserRouter,
  Route,
  Routes,
  Link,
  useNavigate,
  useRouteMatch,
  Redirect,
  useParams,
  useLocation,
  Switch,
} from "react-router-dom";
import logo from "../logo.png";

const useStyles = makeStyles((theme) => ({
  paper: {
    marginTop: theme.spacing(4),
    display: "flex",
    flexDirection: "column",
    alignItems: "center",
  },
  avatar: {
```

```

margin: theme.spacing(1),
backgroundColor: theme.palette.secondary.main,
},
form: {
width: "100%", // Fix IE 11 issue.
marginTop: theme.spacing(1),
},
submit: {
margin: theme.spacing(3, 0, 2),
},
});

export default function SignIn() {
const classes = useStyles();

const [msg, setmsg] = useState(null);
const [open, setopen] = useState(false);
const [data, setdata] = useState({
email: "",
password: "",
});

const change = (e) => {
setdata({
...data,
[e.target.name]: e.target.value,
});
};

const history = useNavigate();

const submit = (e) => {
e.preventDefault();

axios.post(`${API}/signin`, data).then((res) => {
setmsg(res.data.msg);
setopen(true);
if (res.data.msg === "log In Successful") {
localStorage.setItem("user", JSON.stringify(res.data.user));
setmsg(res.data.msg);
setopen(true);
history("/homepage");
} else {

```

```

    setmsg(res.data.msg);
    setopen(true);
  }
});
};

const handleClick = () => {
  setopen(false);
};
useEffect(() => {
  if (localStorage.getItem("user")) {
    history("/homepage");
  }
}, []);
return (
  <Container component="main" maxWidth="xs">
    <CssBaseline />
    <div className={classes.paper}>
      <img src={logo} style={{ width: 130 }} />

      <Typography component="h1" variant="h5">
        Sign in
      </Typography>
      <form onSubmit={submit} className={classes.form}>
        <TextField
          variant="outlined"
          margin="normal"
          required
          value={data.email}
          onChange={change}
          fullWidth
          id="email"
          label="Email Address"
          name="email"
          type="email"
          // inputProps={{ pattern: "" }}

          autoComplete="email"
        />
        <TextField
          variant="outlined"
          margin="normal"

```



```

    required
    fullWidth
    name="password"
    label="Password"
    type="password"
    id="password"
    value={data.password}
    onChange={change}
    autoComplete="current-password"
  />
  <Button
    type="submit"
    fullWidth
    variant="contained"
    color="primary"
    className={classes.submit}
  >
    Sign In
  </Button>
  <Grid container spacing={3}>
    <Grid item>
      <Link
        to="/signup"
        variant="body2"
        style={{ color: "#0A5995", textDecoration: "none" }}
      >
        {"Don't have an account? Sign Up"}
      </Link>
    </Grid>

    <Grid item>
      <Link
        to="/forgetpassword"
        variant="body2"
        style={{ color: "#0A5995", textDecoration: "none" }}
      >
        {"Forget Password ?"}
      </Link>
    </Grid>
  </Grid>
</form>
</div>

```

```

<Snackbar open={open} autoHideDuration={3000} onClose={handleClick}>
  {() => {
    if (msg === "Incorrent Password") {
      return (
        <MuiAlert severity="error" elevation={6} variant="filled">
          {msg}
        </MuiAlert>
      );
    } else if (msg === "This Email is not registered") {
      return (
        <MuiAlert severity="error" elevation={6} variant="filled">
          {msg}
        </MuiAlert>
      );
    } else {
      return (
        <MuiAlert severity="success" elevation={6} variant="filled">
          {msg}
        </MuiAlert>
      );
    }
  }}()
</Snackbar>
</Container>
);
}

```

7.3 Products Slider

```

import React, { useEffect, useState } from "react";
import Slider from "react-slick";
import "slick-carousel/slick/slick.css";
import "slick-carousel/slick/slick-theme.css";
import "./Product_slider.css";
import ArrowBackIcon from "@material-ui/icons/ArrowBackIcon";
import ArrowForwardIcon from "@material-ui/icons/ArrowForwardIcon";
import Card from "@material-ui/core/Card";
import CardActionArea from "@material-ui/core/CardActionArea";
import CardActions from "@material-ui/core/CardActions";
import CardContent from "@material-ui/core/CardContent";

```

```

import CardMedia from "@material-ui/core/CardMedia";
import Button from "@material-ui/core/Button";
import Typography from "@material-ui/core/Typography";
import { Grid, Hidden, Divider, Container } from "@material-ui/core";
import { makeStyles } from "@material-ui/core/styles";
import {
  BrowserRouter as Router,
  Switch,
  Route,
  Link,
  useParams,
} from "react-router-dom";

import API from "../api";

const useStyles = makeStyles((theme) => ({
  root1: {
    alignItems: "center",
    justifyContent: "center",
  },

  heading: {
    textAlign: "center",
    alignItems: "center",
    fontSize: "30px",
    fontFamily: "font-family: Poppins,sans-serif",
    fontWeight: 700,
    textTransform: "uppercase",
    [theme.breakpoints.down("md")]: {
      marginTop: "30px",
    },
  },
  line1: {
    marginTop: "15px",
    fontWeight: "bold",
    backgroundColor: "grey",
  },

  line2: {
    marginTop: "15px",
    backgroundColor: "grey",
  },
}));

```

```

root: {
  maxWidth: 345,
  textAlign: "center",
  alignItems: "center",
  border: "none",
  padding: "10px",
},
media: {
  width: "100%",
  height: "50vh",
  objectFit: "contain",
},
cat: {
  texAlign: "center",
  fontWeight: 400,
  fontSize: "11px",
  fontFamily: "Open Sans,sans-serif",
  lineHeight: 1.7,
  opacity: 0.8,
  textTransform: "uppercase",
  textOverflow: "ellipsis",
  overflow: "hidden",
  whiteSpace: "nowrap",
},
title: {
  maxWidth: "100%",
  fontWeight: 400,
  fontSize: "1rem",
  fontSamily: "Poppins,sans-serif",
  lineHeight: 1.35,
  letterSpacing: "0.005em",
  marginBottom: "0.4rem",
  textOverflow: "ellipsis",
  overflow: "hidden",
},
price: {
  fontWeight: 600,
  fontSize: "1.2rem",
  fontSamily: "Poppins,sans-serif",
  lineHeight: 1.35,
  letterSpacing: "0.005em",
  marginBottom: "0.4rem",

```

```

      textOverflow: "ellipsis",
      overflow: "hidden",
    },
    btn: {
      fontSamily: "Poppins,sans-serif",
      lineHeight: 1.35,
      letterSpacing: "0.005em",
      marginBottom: "0.4rem",
      textOverflow: "ellipsis",
      overflow: "hidden",
      background: "#0A5995",
      color: "white",
      textTransform: "none",
      paddingLeft: "15px",
      paddingRight: "15px",
      "&:hover": {
        background: "black",
        color: "#fff",
      },
    },
  },
  }));

function Feature_slider() {
  const classes = useStyles();
  const [loginuser, setloginuser] = useState(
    JSON.parse(localStorage.getItem("user"))
  );
  const [ProductData, SetProductData] = useState([]);

  useEffect(() => {
    fetch(`${API}/getproducts`)
      .then((response) => response.json())
      .then((data) => {
        // console.log(data);

        const products = data.map((i) => ({
          cat: i.category,
          title: i.title,
          description: i.description,
          price: i.price,
          gender: i.gender,
          id: i._id,

```

```

        img: i.image,
      }));
      // console.log(products);
      // SetProductData(products);
      SetProductData(
        products.filter((product) => product.gender === loginuser.gender)
      );

      //console.log(products.filter(product =>product.gender===loginuser.gender))
    });
  }, []);

const PreviousBtn = (props) => {
  const { className, onClick } = props;
  return (
    <div className={className} onClick={onClick}>
      <ArrowBackIcon style={{ color: "black" }} />
    </div>
  );
};

const NextBtn = (props) => {
  const { className, onClick } = props;
  return (
    <div className={className} onClick={onClick}>
      <ArrowForwardIcon style={{ color: "black" }} />
    </div>
  );
};

const settings = {
  autoplay: true,
  autoplaySpeed: 3000,
  dots: false,
  prevArrow: <PreviousBtn />,
  nextArrow: <NextBtn />,
  slidesToShow: 4,
  slidesToScroll: 2,
  responsive: [
    {
      breakpoint: 1024,
      settings: {

```

```

        slidesToShow: 3,
        slidesToScroll: 3,
        infinite: true,
        dots: true,
      },
    },
    {
      breakpoint: 600,
      settings: {
        slidesToShow: 2,
        slidesToScroll: 2,
        initialSlide: 2,
      },
    },
    {
      breakpoint: 425,
      settings: {
        slidesToShow: 1,
        slidesToScroll: 1,
      },
    },
  ],
};

return (
  <>
    <Container style={{ marginTop: "20px" }}>
      <div className={classes.root1}>
        <Grid container>
          <Hidden only={['xs', 'sm']}>
            <Grid item xs={12} md={4} lg={4}>
              <Divider className={classes.line1} />
            </Grid>
          </Hidden>

          <Grid item xs={12} md={4} lg={4}>
            <h4 className={classes.heading}>Feature Products</h4>
          </Grid>
          <Hidden only={['xs', 'sm']}>
            <Grid item xs={12} md={4} lg={4}>
              <Divider className={classes.line2} />
            </Grid>
          </Hidden>
        </Grid>
      </div>
    </Container>
  </>
);

```

```

    </Hidden>
  </Grid>
</div>

<div style={{ margin: "30px" }}>
  <Slider {...settings}>
    {ProductData.slice(70, 80).map((i) => (
      <div>
        <Link
          style={{ textDecoration: "none" }}
          to={` /products/${i.id}`}
        >
          <Card variant="outlined" className={classes.root}>
            <CardActionArea>
              <CardMedia
                className={classes.media}
                image={i.img}
                title={i.title}
              />
            </CardActionArea>
            <CardContent>
              <Typography
                className={classes.cat}
                variant="body1"
                color="textSecondary"
                component="p"
              >
                {i.cat}
              </Typography>
              <Typography
                className={classes.title}
                variant="body1"
                component="h3"
              >
                {i.title}
              </Typography>
              { /* <Typography className={classes.title} variant="body1" component="h3">
                {i.description.slice(0,40)+"..." }
              </Typography> */ }
              <Typography
                className={classes.price}
                variant="h6"

```



```

        component="h2"
      >
        {"PKR " + i.price}
      </Typography>
    </CardContent>
    <CardActions
      style={{
        justifyContent: "center",
        alignItems: "center",
        textAlign: "center",
      }}
    >
      <Button
        className={classes.btn}
        size="medium"
        color="primary"
      >
        Buy Now
      </Button>
    </CardActions>
  </Card>
</Link>
</div>
)))
</Slider>
</div>
</Container>
</>
);
}

```

```
export default Feature_slider;
```

7.4 Forget Password

```

import React, { useEffect, useState } from "react";
import Avatar from "@material-ui/core/Avatar";
import Button from "@material-ui/core/Button";
import CssBaseline from "@material-ui/core/CssBaseline";
import TextField from "@material-ui/core/TextField";
import FormControlLabel from "@material-ui/core/FormControlLabel";

```

```

import Checkbox from "@material-ui/core/Checkbox";
import Grid from "@material-ui/core/Grid";
import Box from "@material-ui/core/Box";
import MuiAlert from "@material-ui/lab/Alert";
import { Snackbar } from "@material-ui/core";
import LockOutlinedIcon from "@material-ui/icons/LockOutlined";
import Typography from "@material-ui/core/Typography";
import { makeStyles } from "@material-ui/core/styles";
import Container from "@material-ui/core/Container";
import axios from "axios";
import API from "../api";
import {
  BrowserRouter,
  Route,
  Routes,
  Link,
  useNavigate,
  useRouteMatch,
  Redirect,
  useParams,
  useLocation,
  Switch,
} from "react-router-dom";
import logo from "../logo.png";

const useStyles = makeStyles((theme) => ({
  paper: {
    marginTop: theme.spacing(8),
    display: "flex",
    flexDirection: "column",
    alignItems: "center",
  },
  avatar: {
    margin: theme.spacing(1),
    backgroundColor: theme.palette.secondary.main,
  },
  form: {
    width: "100%", // Fix IE 11 issue.
    marginTop: theme.spacing(1),
  },
  submit: {
    margin: theme.spacing(3, 0, 2),

```

```

    },
  });

export default function Fpassword() {
  const classes = useStyles();

  const [msg, setmsg] = useState(null);
  const [open, setopen] = useState(false);
  const [data, setdata] = useState({
    email: "",
  });

  const change = (e) => {
    setdata({
      ...data,
      [e.target.name]: e.target.value,
    });
  };

  const history = useNavigate();

  const submit = (e) => {
    e.preventDefault();

    axios
      .post(`${API}/forgetpassword`, data)

      .then((res) => {
        setmsg(res.data.msg);
        setopen(true);

        if (res.data.msg == "Check your email for password") {
          setdata({ email: "" });
        }
      });
  };

  const handleClick = () => {
    setopen(false);
  };

  useEffect(() => {
    if (localStorage.getItem("user")) {
      history("/homepage");
    }
  }, []);
}

```

```

    }
  }, []);
  return (
    <Container component="main" maxWidth="xs">
      <CssBaseline />
      <div className={classes.paper}>
        <img src={logo} style={{ width: 110 }} />

        <Typography component="h1" variant="h5">
          Forget Password
        </Typography>
        <form onSubmit={submit} className={classes.form}>
          <TextField
            variant="outlined"
            margin="normal"
            required
            value={data.email}
            onChange={change}
            fullWidth
            id="email"
            label="Email Address"
            name="email"
            type="email"
            // inputProps={{ pattern: "" }}

            autoComplete="email"
          />
          <Button
            type="submit"
            fullWidth
            variant="contained"
            color="primary"
            className={classes.submit}
          >
            Send login details
          </Button>
          <Grid container>
            <Grid item>
              <Link
                to="/"
                variant="body2"
                style={{ color: "#0A5995", textDecoration: "none" }}

```

```

        >
        {"Back to Login..."}
      </Link>
    </Grid>
  </Grid>
</form>
</div>

<Snackbar open={open} autoHideDuration={6000} onClose={handleClick}>
  {(() => {
    if (msg === "This email is not exist") {
      return (
        <MuiAlert severity="error" elevation={6} variant="filled">
          {msg}
        </MuiAlert>
      );
    } else {
      return (
        <MuiAlert severity="success" elevation={6} variant="filled">
          {msg}
        </MuiAlert>
      );
    }
  })()}
</Snackbar>
</Container>
);
}

```

7.5 Product Details

```

import {
  Grid,
  Card,
  CardMedia,
  Typography,
  Divider,
  Container,
  Button,
  IconButton,

```

```

CardActionArea,
CardContent,
CardActions,
Link,
} from "@material-ui/core";
import React, { useState, useEffect } from "react";
import Breadcrumbs from "@material-ui/core/Breadcrumbs";
import NavigateNextIcon from "@material-ui/icons/NavigateNext";
import { Hidden } from "@material-ui/core";
import API from "../api";
import {
  BrowserRouter as Router,
  Link as Linknew,
  useParams,
} from "react-router-dom";
import { makeStyles } from "@material-ui/core/styles";
import Header1 from "../Header1";
import FacebookIcon from "@material-ui/icons/Facebook";
import TwitterIcon from "@material-ui/icons/Twitter";
import LinkedInIcon from "@material-ui/icons/LinkedIn";
import InstagramIcon from "@material-ui/icons/Instagram";
import Footer from "../Footer";

```

```

const useStyles = makeStyles((theme) => ({
  root: {
    maxWidth: 570,
  },

  media: {
    height: "75vh",
  },
  row: {},
  h1: {
    color: "#222529",
    fontSize: "2.4rem",
    fontWeight: 600,
    letterSpacing: "0rem",
    [theme.breakpoints.down("md")]: {
      fontSize: "1.8rem",
      marginTop: "10px",
    },
  },
},
),

```

```

Breadcrumbs: {
  marginTop: "10px",
  "& > * + *": {
    marginTop: theme.spacing(2),
    color: "#777",
  },
},
link: {
  textDecoration: "none",
  color: "#777",
  fontSize: "12px",
  lineHeight: "24px",
  textTransform: "uppercase",
  [theme.breakpoints.down("md")]: {
    fontSize: "10px",
  },
},
Divider: {
  width: "20%",
  background: "#e7e7e7",
  border: "1px solid ",
  marginTop: "15px",
},
body1: {
  marginTop: "25px",
  color: "#222529",
  fontSize: "1.5rem",
  fontWeight: 600,
  verticalAlign: "middle",
  lineHeight: ".8",
  [theme.breakpoints.down("md")]: {
    fontSize: "1.3rem",
  },
},
des: {
  color: "#777",
  marginTop: "20px",
  fontSize: "1.2rem",
  [theme.breakpoints.down("md")]: {
    fontSize: "1rem",
  },
},

```

```

span: {
  fontSize: "15px",
  fontWeight: 700,
  paddingLeft: "10px",
  textTransform: "uppercase",
},
div: {
  display: "flex",
  marginTop: "30px",
},
cataaa: {
  fontSize: "15px",
  color: "#777",
  textTransform: "uppercase",
},
Divider1: {
  marginTop: "20px",
},
btn: {
  marginTop: "20px",
  background: "black",
  color: "white",
  width: "30%",
  padding: "9px",
},
icon: {
  color: "black",
},
stock: {
  backgroundColor: "green",
  fontSize: "12px",
  color: "white",
  width: "15%",
  textAlign: "center",
  borderRadius: "12px",
  padding: "4px",
  [theme.breakpoints.down("md")]: {
    fontSize: "10px",
  },
},
div1: {
  display: "flex",

```



```

    marginTop: "10px",
  },
  relpro: {
    marginTop: "50px",
    fontSize: "1.2rem",
    fontWeight: 700,
    textTransform: "uppercase",
  },
  root1: {
    maxWidth: 345,
    textAlign: "center",
    alignItems: "center",
    border: "none",
    "&:hover": {
      border: "1px solid #eee",
      boxShadow: " 0px 15px 15px 2px lightGrey ",
      background: "offWhite",
    },
  },
  media1: {
    height: "40vh",
  },
  cat1: {
    texAlign: "center",
    fontWeight: 400,
    fontSize: "11px",
    fontFamily: "Open Sans,sans-serif",
    lineHeight: 1.7,
    opacity: 0.8,
    textTransform: "uppercase",
    textOverflow: "ellipsis",
    overflow: "hidden",
    whiteSpace: "nowrap",
  },
  title1: {
    maxWidth: "100%",
    fontWeight: 400,
    fontSize: "1rem",
    fontSamily: "Poppins,sans-serif",
    lineHeight: 1.35,
    letterSpacing: "0.005em",
    marginBottom: "0.4rem",
  },

```

```

textOverflow: "ellipsis",
overflow: "hidden",
[theme.breakpoints.down("md")]: {
  fontSize: "12px",
},
},
price1: {
  fontWeight: 600,
  fontSize: "1.2rem",
  fontSamily: "Poppins,sans-serif",
  lineHeight: 1.35,
  letterSpacing: "0.005em",
  marginBottom: "0.4rem",
  textOverflow: "ellipsis",
  overflow: "hidden",
[theme.breakpoints.down("md")]: {
  fontSize: "14px",
},
},
btn1: {
  fontSamily: "Poppins,sans-serif",
  lineHeight: 1.35,
  letterSpacing: "0.005em",
  marginBottom: "0.4rem",
  textOverflow: "ellipsis",
  overflow: "hidden",
  background: "#f4f4f4",
  color: "#6f6e6b",
  textTransform: "none",
  paddingLeft: "15px",
  paddingRight: "15px",
"&:hover": {
    background: "black",
    color: "#fff",
  },
},
});

function Product_detail() {
  const [loginuser, setloginuser] = useState(
    JSON.parse(localStorage.getItem("user"))
  );

```

```

const classes = useStyles();
const proid = useParams();
const [ProductData, SetProductData] = useState([]);
const [filterdata, Setfilterdata] = useState([]);

useEffect(() => {
  fetch(`${API}/getproducts`)
    .then((response) => response.json())
    .then((data) => {
      //console.log(data);

      SetProductData(data.filter((x) => x._id === proid.id));
      Setfilterdata(data);
      console.log(filterdata);
    });
}, [proid]);

var apidata = filterdata
  .filter(
    (x1) =>
      x1.brand === ProductData[0].brand &&
      x1.category === ProductData[0].category
  )
  .filter((x1) => x1.gender === ProductData[0].gender);
// console.log(apidata.length)
var arr = apidata.length - 5;
var randomdata = Math.floor(Math.random() * arr);
//console.log(randomdata)

console.log(apidata);
return (
  <>
    <Header1 />
    <Divider />
    <Divider />

    {ProductData.map((i) => (
      <>
        <Container>
          <div className={classes.Breadcrumbs}>
            <Breadcrumbs
              separator={<NavigateNextIcon fontSize="small" />}

```

```

    aria-label="breadcrumb"
  >
    <Link to="/shop" className={classes.link}>
      {i.category}
    </Link>
    <Typography className={classes.link} color="textPrimary">
      {i.title}
    </Typography>
  </Breadcrumbs>
</div>
</Container>

<div className={classes.row}>
  <Container>
    <Grid container style={{ marginTop: "15px" }}>
      <Grid xs={12} md={6} lg={6}>
        <Card className={classes.root}>
          <CardMedia
            className={classes.media}
            component="img"
            alt="Contemplative Reptile"
            height="140"
            image={i.image}
            title={i.title}
          />
        </Card>
      </Grid>
      <Grid xs={12} md={6} lg={6}>
        <div>
          <Typography variant="p" className={classes.h1}>
            {i.title}
          </Typography>
          <Typography variant="body1" className={classes.stock}>
            {i.availability}
          </Typography>
          <Divider className={classes.Divider} />

          <Typography variant="body1" className={classes.body1}>
            {"PKR " + i.price}
          </Typography>
          <Typography variant="body1" className={classes.des}>
            {i.description}

```

```

</Typography>
<div className={classes.div}>
  <Typography className={classes.cataaa}>
    Catagory :
  </Typography>
  <Typography className={classes.span}>
    {i.category}
  </Typography>
</div>
<div className={classes.div1}>
  <Typography className={classes.cataaa}>
    Brand :
  </Typography>
  <Typography className={classes.span}>
    {i.brand}
  </Typography>
</div>
<div className={classes.Divider1}>
  <Divider />
  <Divider />
</div>
<Link
  style={{ textDecoration: "none" }}
  href={i.url}
  target="_blank"
>
  <Button className={classes.btn} variant="contained">
    Buy Now
  </Button>
</Link>
<div className={classes.Divider1}>
  <Divider />
  <Divider />
</div>
</div>
</Grid>
</Grid>
</Container>

<Container>
  <div style={{ marginTop: "20px" }}>
    <Typography className={classes.relpro} variant="p">

```

Related Products
</Typography>

```
<div>
  <Divider />
  <Divider />
</div>
<Container style={{ padding: "40px" }}>
  <Grid container spacing={2}>
    {apidata.slice(randomdata, randomdata + 4).map((i) => (
      <>
        <Grid item xs={6} md={6} lg={3}>
          <Linknew
            onClick={() => {
              window.scrollTo({ top: 0, behavior: "smooth" });
            }}
            style={{ textDecoration: "none" }}
            to={` /products/${i._id}`}
          >
            <Card variant="outlined" className={classes.root1}>
              <CardActionArea>
                <CardMedia
                  className={classes.media1}
                  image={i.image}
                  title={i.title}
                />
                <CardContent>
                  <Typography
                    className={classes.cat1}
                    variant="body1"
                    color="textSecondary"
                    component="p"
                  >
                    {i.category}
                  </Typography>
                  <Typography
                    className={classes.title1}
                    variant="body1"
                    component="h3"
                  >
                    {i.title}
                  </Typography>
                </CardContent>
              </CardActionArea>
            </Card>
          </Grid item>
        </>
      )
    )}
```

```

        <Typography
          className={classes.title1}
          variant="body1"
          component="h3"
        >
          {i.description.slice(0, 100) + " ...."}
        </Typography>
        <Typography
          className={classes.price1}
          variant="h6"
          component="h2"
        >
          {"PKR " + i.price}
        </Typography>
      </CardContent>
    </CardActionArea>
    <CardActions
      style={{
        justifyContent: "center",
        alignItems: "center",
        textAlign: "center",
      }}
    >
      <Button
        className={classes.btn1}
        size="medium"
        color="primary"
      >
        Buy Now
      </Button>
    </CardActions>
  </Card>
</Linknew>
</Grid>
</>
  )}}
</Grid>
</Container>
</div>
</Container>
</div>
</>

```

```

    )))}

    <br />
    <br />

    { /* <Hidden only={['sm','xs']}>
    <Footer/>
    </Hidden> */}
    </>
  );
}

export default Product_detail;

```

7.6 Rendering Code

```

import React, { useEffect, useState, createContext } from "react";
import './App.css';
import Loading from './components/loading';
import SignIn from './components/signin';
import SignUp from './components/signup';
import ProtectedRoute from './ProtectedRoute';
import Searchpage from './components/Searchpage';
import Profile from './components/profile';
import Fourzeroofour from './components/Fourzeroofour';
//import fourzeroofour from './components/Fourzeroofour';
import Men from './components/men.js';
import Women from './components/women.js';
import { BrowserRouter, Route, Routes } from "react-router-dom";
import {
  createTheme,
  ThemeProvider,
  responsiveFontSizes,
} from "@material-ui/core/styles";
import Hidden from "@material-ui/core/Hidden";

import Homepage from './components/Homepage';
import Shop from './components/Shop';
import Product_detail from './components/Product_detail';
import Unstitched from './components/Unstitched';

```



```

import Stitched from "./components/Stitched";
import Western from "./components/Western";
import Shoes from "./components/Shoes";
import Mobilenav from "./components/Mobilenav";
import About from "./components/About";

import Fpassword from "./components/forgetpassword";
let theme = createTheme({
  palette: {
    primary: {
      main: "#0A5995",
    },
  },
});

theme = responsiveFontSizes(theme);
export const Context = createContext(null);

function App() {
  const [loginuser, setloginuser] = useState(
    JSON.parse(localStorage.getItem("user"))
  );

  return (
    <>
    <BrowserRouter>
    <ThemeProvider theme={theme}>
    <Context.Provider value={{ loginuser, setloginuser }}>
    {/* <Route path="/" element={<Mobilenav/>} /> */}
    <Hidden only={['xl', 'lg', 'md']}>
    <Mobilenav />
    </Hidden>

    <Routes>
    <Route path="/" element={<SignIn />} />
    <Route path="/signup" element={<SignUp />} />
    <Route path="/forgetpassword" element={<Fpassword />} />
    <Route element={<ProtectedRoute />}>
    <Route path="/homepage" element={<Homepage />} />
    <Route path="/shop" element={<Shop />} />
    <Route path="/men" element={<Men />} />
    <Route path="/women" element={<Women />} />

```

```

    <Route path="/profile" element={<Profile />} />
    Women
    <Route path="/Search" element={<Searchpage />} />
    <Route path="/products/:id" element={<Product_detail />} />
    <Route path="/unstitched" element={<Unstitched />} />
    <Route path="/stitched" element={<Stitched />} />
    <Route path="/Western" element={<Western />} />
    <Route path="/shoes" element={<Shoes />} />
    <Route path="/about" element={<About />} />
  </Route>
  <Route path="/*" element={<Fourzerofour />} />
</Routes>
</Context.Provider>
</ThemeProvider>
</BrowserRouter>
</>
);
}
export default App;

```

CHAPTER-8

SYSTEM TESTING

8.1 Testing

After development the hybrid application was tested based on functional and non-functional requirements. The various tests are detailed as:

8.1.1 Usability Testing

When it comes to making application user-friendly and effective, its user interface should comply with the standards. We have followed all the global conventions and modern web standards while developing this web application. As everyone knows that Usability testing is suitable for the applications that are intended to streamline the manual process. However, we know that we should also keep in mind certain important aspects like proper navigation, proper color combinations, avoid using over-crowded content and more while approaching for the usability testing.

8.1.2 Functionality Testing

Our main goal of functional testing was to make sure that all the functions within our application are working smoothly without any technical glitches. In this project, functional testing could cover different things like whether all the modules are working properly or not, testing database for the security and so on. One should also ensure that test cases cover all the boundary conditions that need to be tested. Also, invalid inputs should give an appropriate error message.

8.1.3 Performance Testing

Performance testing will help us determine the performance of our project under various scenarios. Performance testing usually involves stress testing, scalability testing and load testing. In this testing method, software is usually tested for its functionality on different windows operating system, hardware platforms and more.

8.1.4 Compatibility Testing

Compatibility of application is one of the most crucial things. We have considered while testing the application. During compatibility testing we have checked our application on different Browser and different versions of android operating system.

8.1.5 Security Testing

Security Testing is a process which checks whether the confidential data stays confidential or not (i.e., it is not exposed to individuals/entities for which it is not meant for) and the users can perform only those tasks that they are authorized to perform.

This application is protected from following vulnerabilities:

- URL Manipulation
- SQL injection
- Cross-Site Scripting

CHAPTER-9

CONCLUSION AND FUTURE WORK

9.1 Conclusion

In general, today's businesses must always strive to create the next best thing that consumers will want because consumers continue to desire their products, services etc. to continuously be better, faster, and cheaper. In this world of new technology, businesses need to accommodate to the new types of consumer needs and trends because it will prove to be vital to their business' success and survival. E-commerce is continuously progressing and is becoming more and more important to businesses as technology continues to advance and is something that should be taken advantage of and implemented. From the inception of the Internet and e-commerce, the possibilities have become endless for both businesses and consumers. Creating more opportunities for profit and advancements for businesses, while creating more options for consumers. However, just like anything else, e-commerce has its disadvantages including consumer uncertainties, but nothing that cannot be resolved or avoided by good decision-making and business practices. There are several factors and variables that need to be considered and decided upon when starting an e-commerce business. Some of these include: types of e-commerce, marketing strategies, and countless more. If the correct methods and practices are followed, a business will prosper in an e-commerce setting with much success and profitability. This project sets out to conduct research in the area of web scraping and how it can be used as an application for extracting data of the products at different stores. The main focus of this project is to extracting the products details from different stores. To do this, a web scraper will be created to automate the process of extracting data of the products from different stores.

9.2 Future Work

As we know that not even a single project is ever considered as complete forever because our mind is always thinking new and our necessities are also growing time to time.

- In our project also, if you see at the first glance you will find it fully completed but still we want to make it more mature and add more automatic functionalities.
- Add more fashion brands.
- Enable google ads.
- Deploy to Play Store.
- Develop IOS application.

REFERENCES

[1] farfetch. (2022). *Farfetch*. Retrieved February 20, 2022, from <https://www.farfetch.com/al/shopping/men/items.aspx>

[2] zasimo. (2020). *Zasimo*. Retrieved March 14, 2022, from <https://zasimo.pk/>

[3] lyst. (2019). *Lyst*. Retrieved March 12, 2022, from <https://www.lyst.com/>