

ENHANCING CUSTOMER OUTREACH BY USING SOCIAL MEDIA PLATFORMS

A PROJECT REPORT

Submitted by,

N.V.Nithish Kumar	- 20211CSE0865
N.Jaswanth Reddy	- 20211CSE0876
M.Upendra	- 20211CSE0825
S.Farhan	- 20211CSE0844

Under the guidance of, Dr.

Aarif Ahamed S

Assistant Professor Senior Scale School of CSE Presidency University

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

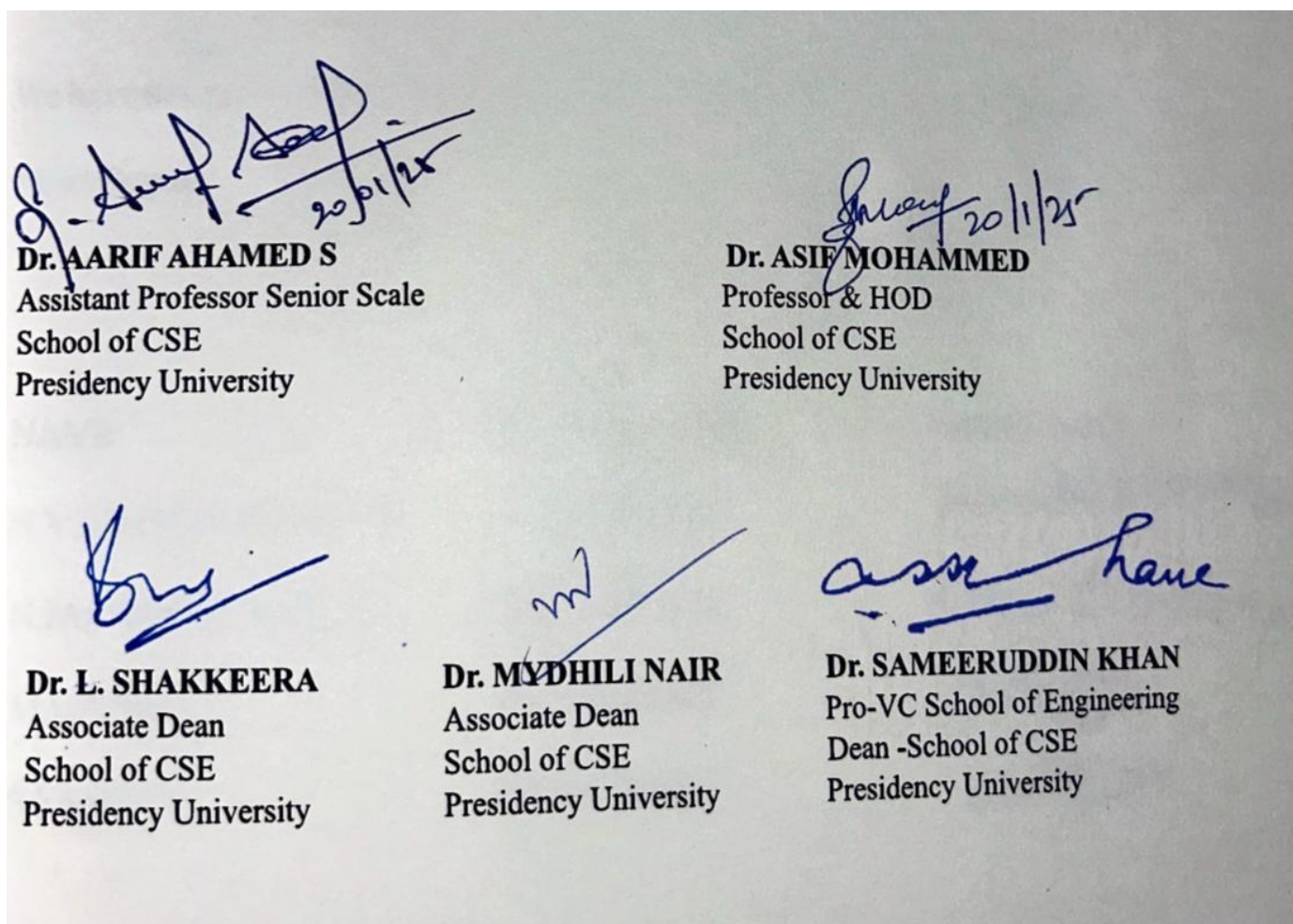
At



PRESIDENCY UNIVERSITY BENGALURU JANUARY 2025

PRESIDENCY UNIVERSITY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING CERTIFICATE

This is to certify that the Project report “**Enhancing Customer Outreach By Using Social Media Platform**” being submitted by “N.V. Nithish Kumar, N. Jaswanth Reddy, M. Upendra, S. Farhan” bearing roll number(s) “20211CSE0865, 20211CSE0876, 20211CSE0825, 20211CSE0844” in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a Bonafide work carried out under my supervision.



PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **Enhancing Customer Outreach By Using Social Media Platforms** in partial fulfillment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr.Aarif Ahamed S, Assistant Professor, School of Computer Science And Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME:	ROLL NUMBER:	SIGNATURE:
N.V.NITHISH KUNMAR	20211CSE0865	N.V.Nithish Kumar 20/1/25
N.JASWANTH REDDY	20211CSE0876	N.Jaswanth Reddy 20/1/25
M.UPENDRA	20211CSE0825	M.Upendra 20/1/25
S.FARHAN	20211CSE0844	S.Farhan 20/1/25

ABSTRACT

The project "Scraping Flipkart, Amazon, Ajio and Myntra for E-Commerce Price Comparison" aims to build a comprehensive price comparison tool that aggregates product information prominent ecommerce websites. Using web-scraping techniques, the system collects the latest product availability, specs, and prices from Flipkart, Amazon, Croma. With its easy-to-use system, this tool helps user to compare prices on multiple platforms, allowing them to make informed purchasing.

E-commerce has revolutionized consumer behaviour, offering an abundance of options. However, finding the best deals remains a tedious process, marked by challenges such as time consumption, missed opportunities, and information overload. This research explores a proposed solution: a unified app that integrates price comparison, coupon discovery, and price alerts for platforms like Amazon, Flipkart, Myntra, and Ajio. The study analyzes the market need, technological feasibility, consumer behaviour, and competitive landscape to establish the app's potential as a comprehensive solution to modern shopping pain points.

Motivation: This is an invited review paper, which analyses the literature in the area of consumer decision making applied to online purchases, to identify the factors that significantly influence such purchases and further studies that can be targeted to provide a better understanding of consumer choice. This study provides insight into the complex interplay of the various forces that shape consumer valorization in the digital economy, conducting a preliminary investigation of the circumstances.

Keywords: Price Comparison, E-Commerce, Web Scraping, Web Crawling, User Interface, Price Alert, Trust, Perceived Risk, Social Influence.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science And Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair**, School of Computer Science And Engineering & Information Science, Presidency University, and **Dr.Asif Mohammed H.B**, Head of the Department, School of Computer Science And Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr.Aarif Ahamed S**, Assistant Professor Senior Scale, School of Computer Science And Engineering, Presidency University and Reviewer **Dr.Raghavendra**, School of Computer Science And Engineering & Information Science, Presidency University for his/her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman**, department Project Coordinators **Mr.Amarnath J.L, Dr.Jayanthi.K** and Git hub coordinator **Mr. Muthuraj**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

N.V.Nithish Kumar

N.Jaswanth Reddy

M.Upendra

S.Farhan

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	ACKNOWLEDGMENT	v
1.	INTRODUCTION	1
2.	LITERATURE SURVEY	3
3.	RESEARCH GAPS OF EXISTING METHODS	6
4.	PROPOSED METHODOLOGY	9
5.	OBJECTIVES	11
6.	SYSTEM DESIGN AND IMPLEMENTATION	13
7.	TIMELINE FOR EXECUTION OF PROJECT	49
8.	OUTCOMES	52
9.	RESULTS AND DISCUSSIONS	53
10.	CONCLUSION	56
11.	REFERENCES	57
12.	APPENDIX	59

List Of Figures

FIG NO.	FIG NAME	PAGE NO.	4.1	System Architecture
				Error! Bookmark not defined.
7.1	Gantt Chart			51
7.2	Flow Chart Of The Process			52

CHAPTER-1

INTRODUCTION

In E-commerce has become a major force in today's digital landscape, transforming the way customers engage with products and shop. Customers now have unparalleled access to a wide range of products and services thanks to the growth of online shopping platforms, which also enable them to buy products from the comfort of their homes. But there are drawbacks to this ease as well, chief among them being the deluge of possibilities and the problem of finding the best deals on the things one wants. Effective price comparison tools are becoming more and more necessary as people look for value for their money. Titled "Scraping Flipkart, Amazon for E-Commerce Price Comparison," satisfies this demand by creating a reliable system that gathers and contrasts product prices from Flipkart, Amazon—the two most wellknown e-commerce sites in India.

The increasing use of smartphones, better internet access, and shifting consumer habits are just a few of the reasons contributing to the growth of e-commerce. Due to the convenience of online shopping, which enables customers to browse and buy things whenever and wherever they choose, consumers are more likely to shop online these days. However, buyers frequently have a dilemma when deciding where to purchase a product due to the abundance of ecommerce websites available. It can be difficult for customers to get the best bargain because different platforms may offer different discounts, prices, and delivery choices. This is where price comparison websites come into play. They are useful resources that make buying easier by giving customers a single location to compare costs from several businesses.

Websites that compare prices have grown in popularity because they enable customers to make well-informed judgments about what to buy. These websites let consumers to view product prices side by side, assisting them in selecting the most economical solutions by combining data from multiple e-commerce platforms. Furthermore, a lot of price comparison websites provide services like price notifications, which alert consumers when a product's price falls to a predetermined point. This feature not only makes shopping more enjoyable, but it also motivates customers to make more deliberate purchases, which eventually results in cost savings. The suggested solution uses web scraping methods to get real-time data from

Amazon, and Flipkart. Web scraping is the process of automatically extracting data from websites in order to efficiently obtain product details such as availability, prices, and descriptions. The system searches through the many e-commerce platforms using web crawlers to retrieve pertinent data, which is subsequently processed and presented on an intuitive interface. By guaranteeing that customers have access to the most recent data, this strategy helps them to make timely decisions based on the state of the market.

This the initiative's emphasis on the three main e-commerce sites, Flipkart, Amazon, Ajio and Myntra is one of its main advantages. With distinct advantages and features, every one of these platforms targets various market niches. For example, Amazon is renowned for its wide selection and customer-focused practices, while Flipkart is known for its wide assortment of products and competitive pricing. The suggested method provides a thorough picture of the market by combining data from all four platforms like Amazon, Flipkart, Ajio, Myntra enabling customers to compare costs for a wide range of goods.

CHAPTER-2

LITERATURE SURVEY

1. Shaikh, A., Khan, R., Panokher, K., Ranjan 2023, Price comparison websites are aimed to assist consumers in comparing the prices of goods and services from multiple sources, allowing them to make more educated decisions and save money when shopping online. Given the hectic lifestyles of city dwellers, many customers prefer to shop online for convenience and time savings. Consumers can save time by comparing costs for the same goods on price comparison websites rather than going from store to store. Instead, users can quickly check and compare pricing from numerous providers using a single platform. This project, dubbed "Price Comparison Website Using Web Scraping," seeks to provide a platform for consumers to locate the greatest bargains on products.
2. Dharmik, H., Padmane, P., Dhoke, K. 2022, With the rapid rise of e-commerce, many online shopping platforms, like as Amazon, Flipkart, and Croma, now offer a wide range of products to consumers. However, the variety of websites might result in a large time investment for individuals looking for high-quality products at the lowest possible prices. Price comparison tools simplify the process by comparing products from numerous vendors, allowing customers to locate the best bargains and save money. Given the hectic schedules of urban customers, many prefer the convenience of internet shopping because it saves time and effort. A price comparison system reduces the need for customers to visit several offline establishments to check costs. Instead, consumers can see pricing comparisons and promotions on a single platform, making it easy to pick where to buy.
3. LR Julian, F Natalia - 2015, The primary goal of employing web scraping for computer parts and assembly price comparison is to simplify the process of locating and comparing prices for distinct computer components from multiple online sellers. This approach uses web scraping techniques to combine real-time data from numerous e-commerce platforms and electronics retailers, allowing customers to quickly identify the best bargains on computer parts including CPUs, GPUs, motherboards,

and other peripherals. The purpose is to give a complete and up-to-date comparison of costs and availability for computer components, allowing customers to make informed purchasing decisions while staying within their budget. This technology reduces the need for manual price checking and allows users to easily access and analyze pricing information, track price trends, and make cost-effective building decisions.

4. Shridevi Swami , Pujashree Vidap 2013, The primary goal of creating a web scraping framework that combines tag and value similarity is to improve the accuracy and flexibility of data extraction from different and rapidly changing online sites. Traditional online scraping methods sometimes struggle with varied HTML structures and content formats on multiple websites, making it difficult to retrieve important information consistently. This system seeks to successfully handle these challenges by combining tag similarity (matching HTML tags and related characteristics) and value similarity (comparing the text within these tags). The framework aims to improve data extraction by enabling intelligent identification and extraction of desired information, regardless of changes in website layouts or structures. This approach offers a more versatile system capable of handling various types of web pages.
5. Shalini, A., and Ambikapathy, R 2022, The project named "E-Commerce Analysis and Product Price Comparison Using Web Mining" aims to use web mining techniques to evaluate and compare product pricing across multiple e-commerce platforms. The study's goal is to create an effective solution for customers and businesses by utilizing innovative data extraction and analysis technologies to collect real-time pricing information from numerous internet shops. This research aims to help consumers make informed shopping decisions by providing a complete price comparison tool that shows the greatest offers and lowest prices accessible online. Additionally, the study intends to investigate market trends and pricing patterns in order to better understand the factors that influence price changes, such as promotional activities and seasonal trends.

6. O. S. Al-Mushayt, W. Gharibi and N. Armi 2022, The primary goal of the study titled "An E-Commerce Control Unit for Addressing Online Transactions in Developing Countries: Saudi Arabia—Case Study" is to create and implement a control unit to improve the management and security of online transactions in developing countries, with a particular emphasis on Saudi Arabia. The purpose of this study is to solve the obstacles that e-commerce systems in developing countries confront, such as transaction security, fraud prevention, and regulatory compliance. The project aims to create a comprehensive control unit that combines sophisticated technology and best practices to properly monitor, manage, and safeguard online transactions. By tackling specific issues such as data privacy, payment fraud, and regulatory compliance, the study intends to increase the reliability and trustworthiness of Saudi e-commerce platforms.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

Despite the advancements in price comparison tools and e-commerce platforms, several research gaps persist in existing methods that hinder a fully optimized shopping experience. Current solutions, such as price comparison websites and browser extensions, often lack realtime integration with multiple platforms like Amazon, Flipkart, Myntra, and Ajio. These tools typically provide static or delayed price information, failing to reflect dynamic price changes or offer updated availability details.

Another significant gap lies in the lack of user-specific personalization. While some tools allow users to search for products, they rarely include features for submitting specific requests for unavailable items or notifying users when those items become available. This absence of proactive user engagement results in missed opportunities for enhancing customer satisfaction.

Mobile applications, though popular for their convenience, often focus on individual platform integration rather than a unified cross-platform experience. Existing apps tend to emphasize specific functionalities, such as cashback offers or single-store browsing, rather than providing a holistic shopping solution.

Addressing these gaps by creating a centralized, real-time, user-centric application could significantly enhance the online shopping experience, bridging the divide between user needs and technological offerings.

Key points:

1. The impact of social media on customer engagement and loyalty: While there is a growing body of research on the use of social media for customer engagement, there is still a need for more in-depth studies on the long-term effects of social media on customer loyalty and retention.
2. The role of social media in crisis communication: Social-media has become an essential tool for companies to communicate during crisis situations. However, there

is a need for more research on the best practices and strategies for using social media in crisis communication.

3. The effectiveness of social media advertising: Social media advertising has become increasingly popular, but there is a need for more research on the effectiveness of different types of social media ads and the factors that influence their success.
4. The use of social media for customer segmentation and targeting: Social-media provides a wealth of data on customer behaviour and preferences, but there is a need for more research on how companies can use this data to segment and target their customers effectively.

Existing methods for enhancing customer outreach using social media include:

1. Social media monitoring: This involves tracking and analyzing social media conversations about a brand or product to gain insights into customer sentiment and preferences.
2. Social media advertising: This involves using social media platforms to create and display ads that target specific customer segments.
3. Social media engagement: This involves actively engaging with customers on social media platforms through comments, likes, and shares to build relationships and foster loyalty.
4. Social media influencer marketing: This involves partnering with social media influencers to promote a brand or product to their followers.
5. Social media content marketing: This involves creating and sharing valuable, relevant, and consistent content on social media platforms to attract and engage customers.

E-Commerce Tools and Platforms Deficits

1. Disconnect in Real-Time Data Updating

- Several price comparison websites and browser extensions are developed to collect information from eCommerce sites such as Amazon, Flipkart, Myntra and Ajio but in most cases they do not:
- Change price when there is discounting, as certain extensions do not do during the sales period.
- Show real product availability in real time and in most cases not warrant delivery of products.
- For users looking to provide users with a reliable and timely update, everyone realtime update integration is a prerequisite.

2. Differentiation in Platform User Interface

- The features of means are so far confined to the familiar interfaces of or else specific stores, for instance only coupon codes or “cashback” offers or a single store through limited price comparisons.
- It is quite tedious and annoying for users to use several tools or applications to achieve their goal within a single session.

3. Interfaces with Little Customization and Engaging Users This code was generated automatically for the language model not yet include users such as the following

- In additional tools that exist today, there is no
- Suggestions to users based on their browsing and other activities on the site vis a vis strongly targeted marketing approaches
- City Group which manages cantonia federal land bank customers who want to be notified once a product becomes available for request.
- The inability to personalize and reach out to users in an engaging way limits the depth of the customer interaction experience.

CHAPTER-4

PROPOSED METHODOLOGY

The proposed methodology involves developing a Flutter-based mobile application that consolidates the best deals and prices from leading e-commerce platforms like Amazon, Flipkart, Myntra, and Ajio. The application is designed to address the research gaps identified in existing solutions by providing real-time integration, user-specific personalization, and a seamless shopping experience.

The app architecture will use APIs to fetch real-time product data, including prices, availability, and offers, from multiple platforms. A centralized search engine within the app will enable users to find products across all integrated platforms, eliminating the need to switch between different apps.

To address the issue of unavailable products, a request-and-notification system will be implemented. Users can submit requests for products that are currently unavailable, and the app will track these requests in the backend. Once the product becomes available, the user will receive a notification through the app's notification page, ensuring timely updates.

The app will employ a user-friendly and intuitive UI/UX design, allowing seamless navigation and quick access to features. Personalization features, such as saved searches and preferencebased recommendations, will further enhance the shopping experience. This methodology ensures a comprehensive, efficient, and user-centric solution, catering to the modern shopper's needs.

To effectively gather, handle, and display product data from multiple e-commerce platforms, the project "Scraping Flipkart, Amazon, Ajio and Myntra for E-Commerce Price Comparison" employs a methodology that encompasses multiple essential procedures.

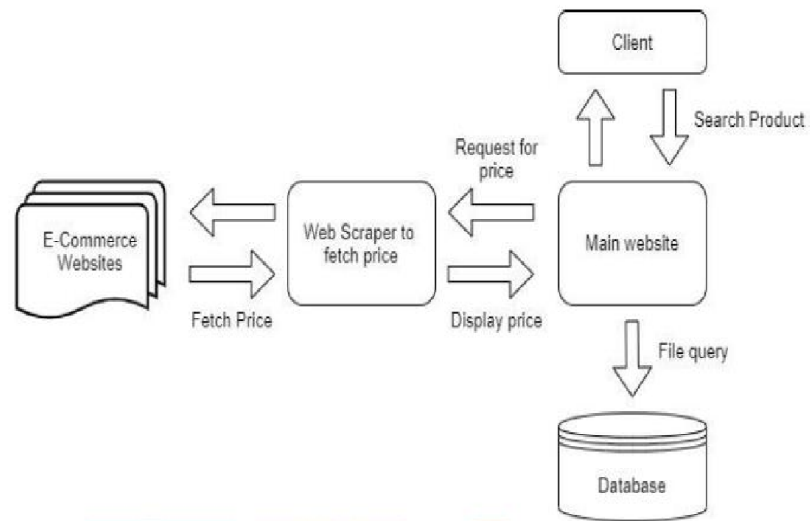


Fig 4.1: System Architecture

Benefits:

1. **Time Saving:** Users can compare prices across multiple platforms like Amazon, Flipkart, Myntra, and Ajio in one app, eliminating the need to browse each site individually.
2. **Cost-Efficiency:** By discovering the best deals and lowest prices, users save money on their purchases without missing out on discounts or offers.
3. **Convenience:** A centralized platform for price comparison and product discovery simplifies the shopping experience.
4. **WideSelection:** Access to multiple e-commerce platforms ensures a broader range of products and styles to suit every need.
5. **Transparency:** A clear, unbiased comparison is provided, helping users trust they are getting the best deal.

CHAPTER-5

OBJECTIVES

The primary objective of this Flutter-based application is to provide users with a centralized platform for discovering the best deals and lowest prices across popular e-commerce platforms such as Amazon, Flipkart, Myntra, and Ajio. By addressing the limitations of existing methods, the app aims to streamline the online shopping experience through enhanced convenience and efficiency.

Key objectives include:

1. ****Centralized Search Functionality****: Enable users to search for products across multiple platforms in one place, eliminating the need to switch between different applications.
2. ****Real-Time Price Comparison****: Provide accurate and up-to-date price information and availability details to assist users in making informed purchasing decisions.
3. ****Request and Notification System****: Allow users to submit requests for unavailable products and receive timely notifications when those products become available, ensuring a proactive shopping experience.
4. ****Personalization****: Incorporate user-specific features such as saved preferences, search history, and tailored recommendations to enhance engagement and satisfaction.
5. ****User-Friendly Interface****: Design an intuitive and aesthetically pleasing UI/UX that promotes seamless navigation and accessibility.
6. ****Time and Cost Savings****: Optimize the shopping process to help users save time and money by consolidating all essential information in a single application.

The centralized search functionality aims to simplify the shopping process by enabling users to conduct cross-platform searches within a single interface. This feature includes advanced filtering options by category, price range, brand, and user reviews. As well as suggestions for searching based on trending products and personal preferences.

The real-time price comparison function helps users get the latest information to make informed purchasing decisions. Additional features include tracking price history to identify trends over time and highlighting the best deals, such as bundled offers and discounts across multiple platforms.

A request and notification system keeps users on top of the products they want by providing customizable notifications for sales, flash sales, and other platform-specific offers. Users can also share these product alerts with friends or family directly through the app. Personalization leverages user data and preferences to provide a tailored shopping experience. Features include AI-powered recommendations based on search and purchase history. Including personal shopping lists and wish lists with sharing options increases user satisfaction...

A user-friendly interface with an intuitive and visually appealing design is essential for smooth navigation. Options like dark mode, customizable themes, and multi-language support Cater to a wide range of user demographics, the app finally consolidates essential information into a single platform and optimizes the shopping journey to save time and costs. Features include automatic coupons and cashback integration during checkout. And price comparisons of similar products across platforms. Helps ensure a streamlined and efficient shopping experience.

These objectives collectively aim to create a comprehensive and user-centric shopping solution, addressing the diverse needs of modern consumers.

-
- Testing includes unit tests for functionality, integration tests for API connectivity, and user testing for UI/UX evaluation.

The system integrates real-time data with user-centric features to deliver a comprehensive, efficient, and personalized shopping experience.

Home Page:

```
// main.dart import 'package:flutter/material.dart'; import
'package:senior/NotificationScreen.dart'; import 'profile page.dart'; import
'welcome.dart'; import 'searchscreen.dart'; import 'cart_screen.dart';

void main() { runApp(MaterialApp( debugShowCheckedModeBanner: false,
home: MyHomePage(title: 'Shopping Platforms'),
));
}

class MyHomePage extends StatefulWidget {
  final String title;

  MyHomePage({Key? key, required this.title}) : super(key: key);

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> { int _selectedIndex =
0;

  final List<Widget> _pages = [

    HomePage(),
```

```
ProductScreen(),

ProfilePage(),

];

void _onItemTapped(int index) {  setState() {  _selectedIndex = index;
  });
}

void _navigateToNotificationScreen() {  Navigator.push(  context,
  MaterialPageRoute(builder: (context) => NotiScreen()),
  );
}

void _navigateToCartScreen() {  Navigator.push(  context,
  MaterialPageRoute(builder: (context) => CartScreen()),
  );
}

@override
Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(
  automaticallyImplyLeading: false,  title: Text(widget.title),  actions: [
  IconButton(  icon: Icon(Icons.notifications),  onPressed:
  _navigateToNotificationScreen,
  ),
  IconButton(  icon: Icon(Icons.shopping_cart),  onPressed:
  _navigateToCartScreen,
  ),
  ],
  ),
  ],
```

```
),  
);  
}  
}
```

Sign up:

```
import 'dart:developer'; import  
'package:flutter/material.dart'; import  
'package:senior/LoginScreen.dart'; import  
'package:senior/auth_service.dart'; import  
'package:senior/widgets/button.dart'; import  
'package:senior/widgets/textfield.dart';  
  
import 'MyHomePage.dart';  
  
class SignupScreen extends StatefulWidget {  
  const SignupScreen( {super.key} );  
  
  @override  
  State<SignupScreen> createState() => _SignupScreenState();  
}  
  
class _SignupScreenState extends State<SignupScreen> {  
  final _auth = AuthService();
```

```
final _name = TextEditingController();
final _email = TextEditingController(); final
_password = TextEditingController();

@override void
dispose() {
super.dispose();
_name.dispose();
_email.dispose();
_password.dispose();
}

@override
Widget build(BuildContext context) { return Scaffold( body:
Padding( padding: const EdgeInsets.symmetric(horizontal: 25),
child: Column( children: [ const Spacer(), const
Text("Signup", style: TextStyle(fontSize: 40, fontWeight:
FontWeight.w500)), const SizedBox( height: 50,
),
CustomTextField(
hint: "Enter Name",
label: "Name",
controller: _name,
),
const SizedBox(height: 20),
CustomTextField( hint:
"Enter Email", label:
"Email", controller:
_email,
),
```

```
        const SizedBox(height: 20),
CustomTextField(          hint:
"Enter Password",        label: "Password",
isPassword:
true,          controller:
_password,
      ),
      const SizedBox(height: 30),
CustomButton(          label:
"Signup",          onPressed:
_signup,
      ),
      const SizedBox(height: 5),
      Row(mainAxisAlignment: MainAxisAlignment.center, children: [
const Text("Already have an account? "),      InkWell(
onTap: () => goToLogin(context),          child: const Text("Login",
style: TextStyle(color: Colors.red)),
      )
]),
      const Spacer()
    ],
  ),
);
}

goToLogin(BuildContext context) => Navigator.push(
context,
  MaterialPageRoute(builder: (context) => const LoginScreen()),
);
```

```
goToHome(BuildContext context) => Navigator.push(
context,
  MaterialPageRoute(builder: (context) => MyHomePage(title: 'Home')),
);

_signup() async {
final user =
  await _auth.createUserWithEmailAndPassword(_email.text, _password.text);
if (user != null) {    log("User Created Succesfully");    goToHome(context);
  }
}}
```

Welco

me

page:

```
import 'package:flutter/material.dart';

class HomePage extends StatelessWidget {
final List<PlatformInfo> platforms = [
  // assets/amazon.png
  // - assets/myntna.jpg
  // - assets/flipkart.png
  // - assets/ajio.png PlatformInfo(    name:
'Amazon',    image: 'assets/amazon.png',
description: 'Discover deals and offers on Amazon.',
  ),
  PlatformInfo(    name:
```

```
'Flipkart',          image: 'assets/flipkart.png',
description: 'Shop the best products on Flipkart.',
),
PlatformInfo(  name:
'Ajio',          image: 'assets/ajio.png',          description:
'Trendy fashion and lifestyle products.',
),
PlatformInfo(  name: 'Myntra',  image:
'assets/myntra.jpg',  description: 'Exclusive fashion
collections at Myntra.',
),
];

@override
Widget build(BuildContext context) {
return Scaffold(

  body: SingleChildScrollView(  child: Padding(
padding: const EdgeInsets.all(8.0),  child:
Column(  crossAxisAlignment: CrossAxisAlignment.start,
  children: [  Container(
margin: EdgeInsets.symmetric(vertical: 10),
padding: EdgeInsets.all(16),  decoration:
BoxDecoration(  color:
Colors.teal.shade50,  borderRadius:
BorderRadius.circular(10),  boxShadow: [
BoxShadow(  color: Colors.black12,  blurRadius:
5,  offset: Offset(0, 2),
),
],
```

```
),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(
        'About Our App',
        style: TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
          color: Colors.teal,
        ),
      ),
      SizedBox(height: 10),
      Text(
        'Our app helps you find the best deals and lowest prices on products across various platforms like Amazon, Flipkart, Myntra, and Ajio. Save time and money by discovering everything in one place!',
        style: TextStyle(
          fontSize: 16,
          color: Colors.black87,
        ),
      ),
      SizedBox(height: 20),
      Text(
        'Explore Platforms',
        style: TextStyle(
          fontSize: 22,
```

```
fontWeight: FontWeight.bold,
color: Colors.teal,
    ),
    ),
    SizedBox(height: 10),
GridView.builder(      shrinkWrap: true,
physics: NeverScrollableScrollPhysics(),
    gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
crossAxisCount: 2,      crossAxisSpacing: 10,
mainAxisSpacing: 10,
    ),
    itemCount: platforms.length,
    itemBuilder: (context, index) {      final
platform = platforms[index];      return
GestureDetector(      onTap: () {
showDialog(      context: context,
builder: (_) => AlertDialog(      title:
Text(platform.name),      content:
Text(platform.description),      actions: [
TextButton(      onPressed: () =>
Navigator.pop(context),      child: Text('Close'),
    ),
    ],
    ),
    );
},
    child: AnimatedContainer(
duration: Duration(milliseconds: 300),
curve: Curves.easeInOut,      decoration:
```

```
BoxDecoration(          color: Colors.white,
borderRadius: BorderRadius.circular(10),
boxShadow: [          BoxShadow(
color: Colors.black12,          blurRadius: 5,
offset: Offset(0, 2),
          ),
        ],
),
        child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [          Image.asset(
platform.image,          height: 60,          fit:
BoxFit.contain,
          ),
        SizedBox(height: 10),
Text(          platform.name,
style: TextStyle(          fontSize: 16,
fontWeight: FontWeight.bold,
color: Colors.teal,
          ),
        ),
      ],
    ),
  ),
),
```

```
),  
);  
}  
}
```

```
class PlatformInfo {  
  final String name; final  
  String image; final  
  String description;  
  PlatformInfo({  
    required this.name,  
    required this.image,  
    required this.description,  
  });  
}
```

Search screen:

```
import 'package:flutter/material.dart'; import  
'package:cloud_firestore/cloud_firestore.dart'; import  
'package:url_launcher/url_launcher.dart';  
  
class ProductScreen extends StatefulWidget {  
  @override  
  _ProductScreenState createState() => _ProductScreenState();  
}  
  
class _ProductScreenState extends State<ProductScreen> {
```

```
final List<Map<String, dynamic>> products = [
    {
        'title': 'Samsung Galaxy Z',
        'price': 99.990,
        'image': 'assets/z fold.png',
        'platform': 'Amazon',
        'url': 'https://www.amazon.in/Samsung-Graygreen-Storage-Additional-
Exchange/dp/B0B8S82751/'
    },
    {
        'title': 'iPhone 13',
        'price': 45490,
        'image': 'assets/iphone.jpg',
        'platform': 'Amazon',
        'url': 'https://www.amazon.in/Apple-iPhone-13-128GB-Starlight/dp/B09G9D8KRQ/'
    },
    {
        'title': 'Sony Speaker',
        'price': 3989,
        'image': 'assets/sony.jpg',
        'platform': 'amazon',
        'url': 'https://www.amazon.in/Sony-Lightweight-Super-Compact-Extra-Durable-
CallingGray/dp/B0C296VD7T/ref=sr_1_7?crd=16QJVFVAT1V9&dib=eyJ2IjojMSJ9.1p
msY4q
N3eeurVb2OAckaT5km1Fzjzbwh5Pls4LoXsP20cpdMIvH5rp1ELpDQfONPaOZqnUIniylfze
0I4SPO4tvd1rU5sCpFr4mb0hDohSGtfcJ_y6_Tt1
x0HEuuyFtL2mfcH3fOaE7U97vmDRaO2owunjshHx1ZMv79QSlAXvko4bmltuKWvR9kC
GTfQ_zev-2YwT-jzh6UjvqzY3sS0dg0dSYeR9l-
TvAHjwbH3Y.C4rYJyLdCL828blqH5Cl04uk042lB1cMIwdfXce3FE&dib_tag=se&keywor
ds=sony+speaker&nsdOptOutPara
```

m=true&qid=1734680914&sprefix=sony+speake%2Caps%2C297&sr=8-7'

},

{

'title': 'Sony Speaker',

'price': 4000,

'image': 'assets/sony.jpg',

'platform': 'Flipkart',

'url': "

},

{

'title': 'Watch',

'price': 578,

'image': 'assets/watch.jpg',

'platform': 'Amazon',

'url': 'https://www.amazon.in/Like-Beautiful-Bluetooth-Calling-

Interactions/dp/B0CV9TQQC9/ref=sr_1_10?crid=2LCHGSC78F9UR&dib=eyJ2IjoiMSJ9.j
zzA9oP9RG1tFAAEq9CWnbLTaitoGbfbwEnylcv5fN3SXEBglQMpoEo53ay8AzJZzTflgzs
HU8B1Gaogxb5BkEz2FKTNE3yW4tSk9sgM02IXn_1Ddx3vvvQc_3hnMHBnIsI123IJG8Zt
TrM-ovmDrqTVx4hSdencwM-
d2TvqCd1kZPaguokOOIFXi_1vblxWz03f6domi1SzzgCJth_yBCY0ItcCdAevVrOwejSMTf
N5mmp vX0nae4rqOXHqHuQ-
CAkgyz5JjJxm3kd9xAYuaq3zAoUrPgA_iIbdw3n8.fk7l8uHCgds8CT34Ewvczn2m41EU7X
VfanBeT6Dr_A&dib_tag=se&keywords=watch&qid=1734681050&sprefix=watc%2
Caps%2C304&sr=8-10'

},

{

'title': 'Watch',

'price': 400,

'image': 'assets/watch.jpg',

```
'platform': 'Flipkart',
'url': ''
},
{
'title': 'us plolo shirt',
'price': 1218,
'image': 'assets/us plolo.png',
'platform': 'amazon',
'url': 'https://www.amazon.in/U-S-POLO-ASSN-Regular-
USSHTO2709A_Navy/dp/B0D28ZK182/ref=sr_1_2?crid=1EMF8DCEEI7BX&dib=eyJ2Ijo
iMSJ9.su1HHaTcNEPXX7DLuoyj2UGFjrZMAaQP7mjuGQZEO0slg9M_Aue2hUJabBgY
kadgvRn2R24v8WoNEckDwWgsaxW0-
pieXTjvfaGHuzNA38pVSNt8CWqGpAQK_2swzZhmcJu8eI2qKZ5PtVd4vLqGFVj2JERW
O1NeP0PKmzQlM7TbRv2SxKKiy6aW1HjCSMFjCYNH9btfuRrHp_Yowzn4wqEO67tml
SEPJLo7sT5uegEtiSD-rU7iu7zHS5FsNF-VNf-
6VLrb5OTev7hi1uH7apzp0zEuVdijLD9OPwo.s3nBQGaKcVXV8V5yKqkss4vEHHvEUml
gNarPYLfv8HU&dib_tag=se&keywords=us+polo+shirt&nsdOptOutParam=true&qid=1734
681186&sprefix=us+plolo+shir%2Caps%2C296&sr=8-2'
},
{
'title': 'us plolo shirt',
'price': 2000,
'image': 'assets/us plolo.png',
'platform': 'Flipkart',
'url': ''
},
{
'title': 'Puma T shirt',
'price': 759,
```

```
'image': 'assets/puma.webp',
'platform': 'Amazon',
'url': 'https://www.amazon.in/Puma-Mens-Solid-T-Shirt-
680817_Black/dp/B0CCNSSQXF/ref=sr_1_1?crd=24VY32BSYJ00V&dib=eyJ2IjoiMSJ9.
nSJJAhgZHhqHwHSlpWLxZ5pVJIntfAkon7ajzmKGxSmo1lyVU7xV2qkcS-
Il8QDhwez9lVj2OhZLB9Enm3-
F2tzLRbuVUIVxjQIEbYcNg1Ru19qBhYtyohFGvKxeH3Be9sCceiYiCkGaJW1U_iXyRcNZ
kNL5bLmG-
oPS3PGFhDwMkTL4UIDGjY46GqeFuVp8Gr4zM0cyn01kBuU8Gulg1Iz_AMfBqvORibK
rmVNOtlUYVYXLImNCSRkr1 y-
Xz7ww02y8K5XFNxGvzXCdo3S5tMJtJef0IdhO0C4MoY1P0.-
eVn3dU9xPDRUmyuHDcDGop2oHR0lMlhmHjAwbeVc&dib_tag=se&keywords=puma+t+
shirt&nsdOptOutParam=true&qid=173
4681289&sprefix=puma+t+shir%2Caps%2C301&sr=8-1'
},
];

List<Map<String, dynamic>> filteredProducts = [];
final TextEditingController _searchController = TextEditingController();
final TextEditingController _nameController = TextEditingController(); final
TextEditingController _productController = TextEditingController(); final
TextEditingController _platformController = TextEditingController();

@override void initState()
{
  super.initState();
  filteredProducts = products;
}

void _searchProduct(String query) {
  final results = products.where((product) {
  final name = product['title'].toLowerCase();
```

```
final searchQuery = query.toLowerCase();
return name.contains(searchQuery);
    }).toList();

    setState(() {
filteredProducts = results;
    });
}

Future<void> _launchURL(String url) async {
if (await canLaunch(url)) {    await
launch(url);
    } else {
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Could not launch URL.')),
    );
    }
}

Future<void> _submitForm() async {    final name =
_nameController.text.trim();    final productName =
_productController.text.trim();    final platform =
_platformController.text.trim();

    if (name.isNotEmpty && productName.isNotEmpty && platform.isNotEmpty) {
try {    await FirebaseFirestore.instance.collection('user_requests').add({
        'name': name,
        'product_name': productName,
        'platform': platform,
        'timestamp': Timestamp.now(),
```

```
});

ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text('Request submitted successfully!')),
);

_nameController.clear();
_productController.clear();
_platformController.clear();
} catch (e) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text('Failed to submit request: $e')),
  );
}
} else {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text('Please fill in all fields.')),
  );
}
}

@override
Widget build(BuildContext context) {
return Scaffold(  body: Column(
children: [  Padding(
padding: const EdgeInsets.all(8.0),
child: TextField(  controller:
_searchController,  decoration:
InputDecoration(  labelText:
'Search Products',  border:
```

```
OutlineInputBorder(),
    ),
    onChanged: _searchProduct,
  ),
),
if (filteredProducts.isEmpty) Expanded(
  child: Padding(padding: const
EdgeInsets.all(16.0), child: Column(mainAxisAlignment:
MainAxisAlignment.center, children: [
  Text(
    'No products found. Please submit your request below.',
  textAlign: TextAlign.center, style: TextStyle(fontSize:
  16),
  ),
  SizedBox(height: 16),
  TextField(controller:
    _nameController, decoration:
  InputDecoration(labelText:
    'Your Name', border:
  OutlineInputBorder(),
    ),
  ),
  SizedBox(height: 16),
  TextField(controller:
    _productController,
  decoration: InputDecoration(
    labelText: 'Product Name',
    border: OutlineInputBorder(),
  ),
  ),
),
```

```
        SizedBox(height: 16),          TextField(
controller: _platformController,      decoration:
InputDecoration(                      labelText: 'Platform (e.g.,
Amazon, Flipkart)',                  border:
OutlineInputBorder(),
    ),
    ),
    SizedBox(height: 16),
ElevatedButton(                      onPressed:
_submitForm,                        child: Text('Submit
Request'),
    ),
  ],
),
),
) else
    Expanded(                      child: GridView.builder(
gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
crossAxisCount: 2,                  crossAxisSpacing: 10.0,
mainAxisSpacing: 10.0,              childAspectRatio: 0.7,
    ),
    itemCount: filteredProducts.length,
itemBuilder: (context, index) {      final
product = filteredProducts[index];
return GestureDetector(
    onTap: () => _launchURL(product['url']),
child: Card(                        shape:
RoundedRectangleBorder(              borderRadius:
BorderRadius.circular(10),
    ),                                elevation: 5.0,          child:
```

```
Column(
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: [
        Expanded(
            child:
                ClipRRect(
                    borderRadius:
                        BorderRadius.vertical(
                            top:
                                Radius.circular(10),
                        ),
                    child: Image.asset(
                        product['image'],
                        fit:
                            BoxFit.cover,
                    ),
                ),
                Padding(
                    padding: const
                        EdgeInsets.all(8.0),
                    child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                            Text(
                                product['title'],
                                style: TextStyle(
                                    fontWeight:
                                        FontWeight.bold,
                                    fontSize: 16,
                                ),
                                maxLines: 2,
                                overflow: TextOverflow.ellipsis,
                            ),
                            SizedBox(height: 4),
                            Text(
                                'Price: ₹${product['price']}',
                                style: TextStyle(
                                    color:
                                        Colors.teal,
                                    fontWeight:
                                        FontWeight.bold,
                                ),
                            ),
                        ],
                    ),
                ),
            ),
    ],
),
```

```
        SizedBox(height: 4),
        Text(
          'in: ${product['platform']}',
          style: TextStyle(
            color: Colors.grey[700],
            overflow: TextOverflow.ellipsis,
          ),
        ),
      ],
    ),
  ],
);
}
```

Profile Page:

```
import 'package:flutter/material.dart'; import
'package:image_picker/image_picker.dart'; import
'dart:io';

class ProfilePage extends StatefulWidget {
  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
  String userName = "Nithishi";
  String userEmail = "Nithishi@example.com";
  File? userAvatar;

  final List<Product> myProducts = [
    Product(name: 'Smartphone', image: 'assets/iphone.jpg', price: 799.99),
```

```
Product(name: 'Headphones', image: 'assets/sony.jpg', price: 49.99),
Product(name: 'Shoes', image: 'assets/sony.jpg', price: 69.99),
];

final ImagePicker _picker = ImagePicker();

void _pickImage() async {  final pickedFile = await
_picker.pickImage(source: ImageSource.gallery);  if (pickedFile != null) {
setState(() {      userAvatar = File(pickedFile.path);
    });
}
}

void _editProfile() {
  TextEditingController nameController = TextEditingController(text:
userName);  showModalBottomSheet(      context: context,
isScrollControlled: true,      shape:
RoundedRectangleBorder(      borderRadius:
BorderRadius.vertical(top: Radius.circular(20)),
    ),
    builder: (context) {      return Padding(      padding:
EdgeInsets.only(      bottom:
MediaQuery.of(context).viewInsets.bottom + 20,      left: 20,      right:
20,      top: 20,
    ),
    child: Column(      mainAxisAlignment:
MainAxisSize.min,      crossAxisAlignment: CrossAxisAlignment.start,
children: [
      Text(
```

```
'Edit Profile',          style: TextStyle(fontSize: 18,
fontWeight: FontWeight.bold),
    ),
    SizedBox(height: 10),
    TextField(            controller:
nameController,          decoration:
InputDecoration(         labelText:
'Name',                  border:
OutlineInputBorder(),
    ),
    ),
    SizedBox(height: 20),
    Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [          ElevatedButton(            onPressed:
_pickImage,              child: Text('Change Avatar'),
    ),
      ElevatedButton(
onPressed: () {           setState(() {
userName = nameController.text;
    });
    Navigator.pop(context);
  },
      child: Text('Save'),
    ),
    ],
  ),
);
```

```
    },  
  );  
}  
  
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      automaticallyImplyLeading: false,  
      title: Text('Profile'),  
      backgroundColor: Colors.teal,  
    ),  
    body: Padding(  
      padding: const  
      EdgeInsets.all(16.0),  
      child:  
      Column(  
        crossAxisAlignment:  
        CrossAxisAlignment.start,  
        children: [  
          Row(  
            children: [  
              GestureDetector(  
                onTap: _editProfile,  
                child: CircleAvatar(  
                  radius: 40,  
                  backgroundImage: userAvatar  
                    != null  
                      ? AssetImage(userAvatar!)  
                      : AssetImage('assets/default iamge.png') as ImageProvider,  
                ),  
              ),  
              SizedBox(width: 16),  
              Column(  
                crossAxisAlignment: CrossAxisAlignment.start,  
                children: [  
                  Row(  
                    children: [  
                      Text(  
                        userName,  
                        style:  
                          TextStyle(  
                            fontSize: 20,  
                            fontWeight: FontWeight.bold,  
                          ),  
                    ],  
                  ),  
                ],  
              ),  
            ],  
          ),  
          GestureDetector(  
            onTap: _editProfile,  
            child: CircleAvatar(  
              radius: 40,  
              backgroundImage: userAvatar  
                != null  
                  ? AssetImage(userAvatar!)  
                  : AssetImage('assets/default iamge.png') as ImageProvider,  
            ),  
          ),  
          SizedBox(width: 16),  
          Column(  
            crossAxisAlignment: CrossAxisAlignment.start,  
            children: [  
              Row(  
                children: [  
                  Text(  
                    userName,  
                    style:  
                      TextStyle(  
                        fontSize: 20,  
                        fontWeight: FontWeight.bold,  
                      ),  
                ],  
              ),  
              ),  
            ],  
          ),  
        ],  
      ),  
    ),  
  ),  
);  
}
```



```
        margin: EdgeInsets.symmetric(vertical: 8),
        shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10),
        ),        child:
        ListTile(        leading:
        ClipRRect(
        borderRadius:
        BorderRadius.circular(8),
        child: Image.asset(
        product.image,
        width: 50,        height:
        50,        fit:
        BoxFit.cover,
        ),
        ),
        title: Text(product.name),
        subtitle: Text(
        '\${product.price.toStringAsFixed(2)}',
        style: TextStyle(color: Colors.green),
        ),
        trailing: Icon(Icons.arrow_forward_ios),
        onTap: () {
        // Add navigation to product details if required
        },
        ),
        );
    },
    ),
    ),
    ],
```

```
    ),  
    ),  
  );  
}  
}
```

```
class Product {  
  final String name;  
  final String image;  
  final double price;  
  Product({required  
    this.name, required  
    this.image, required  
    this.price});  
}
```

Notification-code:

```
import 'package:flutter/material.dart'; import  
'package:firebase_database/firebase_database.dart';
```

```
class NotiScreen extends StatefulWidget {  
  @override  
  _NotiScreenState createState() => _NotiScreenState();  
}
```

```
class _NotiScreenState extends State<NotiScreen> {  final DatabaseReference _database  
= FirebaseDatabase.instance.ref('missing_products');
```

```
  List<Map<String, dynamic>> notifications = [];
```



```
@override void
initState() {
  super.initState();
  _fetchNotifications();
}

void _fetchNotifications() {
  _database.onValue.listen((event) {
    final data = event.snapshot.value;    if
    (data is Map) {      final List<Map<String, dynamic>>
    loadedNotifications = [];
    data.forEach((key, value) {      if (value is Map) {
    loadedNotifications.add({
      "name": value['name'] ?? 'Unknown',
      "product_name": value['product_name'] ?? 'Unknown Product',
      "platform_needed": value['platform_needed'] ?? 'Unknown Platform',
    });
  }    });    setState() {
  notifications = loadedNotifications;
    });    } else {
  setState() {
  notifications = [];
    });
  }
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(    appBar: AppBar(title:
```

```
{notification["platform_needed"]}. Delivery details can be checked on the platform.',
style: TextStyle(fontSize: 16),
    ),
    ),
    SizedBox(height: 10),
  ],
),
);
},
),
);
}
}
```

Cart-Screen:

```
import 'package:flutter/material.dart'; class
CartScreen extends StatelessWidget {  final
List<Map<String, String>> products = [
  {'name': 'Fire TV Stick', 'image': 'assets/fire.jpg'},
  {'name': 'Kadlekai', 'image': 'assets/kadle.webp'},
  {'name': 'Sony Headphones', 'image': 'assets/sony.jpg'},
  {'name': 'Smart Watch', 'image': 'assets/watch.jpg'},
  {'name': 'Amazon Gift Card', 'image': 'assets/amazon.png'},
  {'name': 'Myntra Shirt', 'image': 'assets/myntra.jpg'},
```

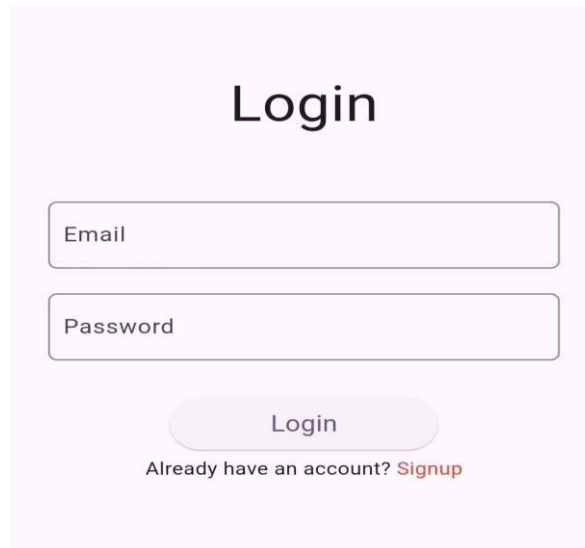
```
{'name': 'Flipkart Offer', 'image': 'assets/flipkart.png'},
{'name': 'AJIO Deal', 'image': 'assets/ajio.png'},
{'name': 'Fridge', 'image': 'assets/firdege.webp'},
{'name': 'iPhone', 'image': 'assets/iphone.jpg'},
{'name': 'Shirt', 'image': 'assets/shirt.jpg'},
{'name': 'Samsung Z Fold', 'image': 'assets/z fold.png'},
{'name': 'Air Conditioner', 'image': 'assets/ac.jpg'},
{'name': 'Laptop Bag', 'image': 'assets/baglaptop.jpeg'},
];

@override
Widget build(BuildContext context) {
return Scaffold(  appBar: AppBar(
title: Text('My Cart'),
),
  body: ListView.builder(    itemCount: products.length,
itemBuilder: (context, index) {      final product =
products[index];    return Card(      margin:
EdgeInsets.symmetric(vertical: 10, horizontal: 15),      child:
ListTile(        leading: Image.asset(
product['image']!,        width: 50,        height:
50,        fit:
BoxFit.cover,
      ),
      title: Text(product['name']!),
trailing: IconButton(      icon:
Icon(Icons.delete, color: Colors.red),
onPressed: () {        // Handle delete action
},
      ),
    ),
  ),
```

```
),  
);  
},  
),  
);  
}  
}
```

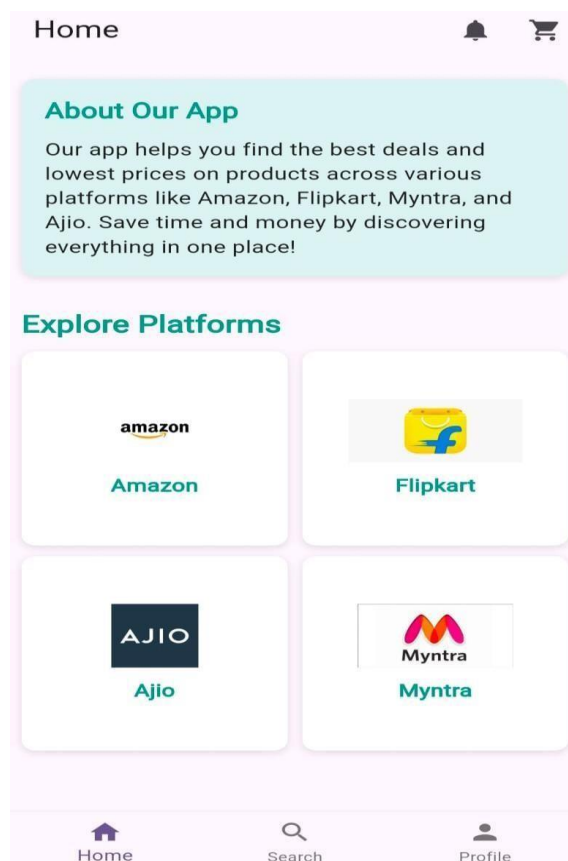
Result Screen Photos:

Login-In Page:

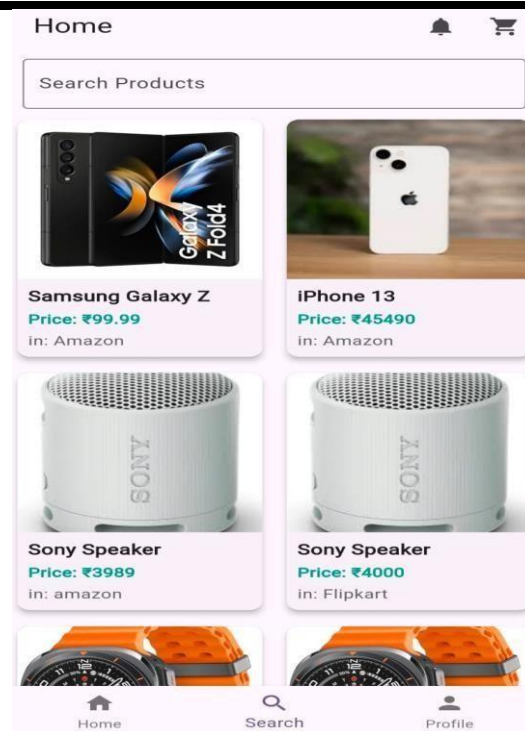


A login page with a light pink background. At the top, the word "Login" is centered in a large, black, sans-serif font. Below it are two white input fields with rounded corners and thin purple borders. The first field is labeled "Email" and the second is labeled "Password". Below these fields is a rounded purple button with the text "Login" in white. At the bottom, there is a link that says "Already have an account? Signup" in a small, black, sans-serif font, with "Signup" in a reddish-brown color.

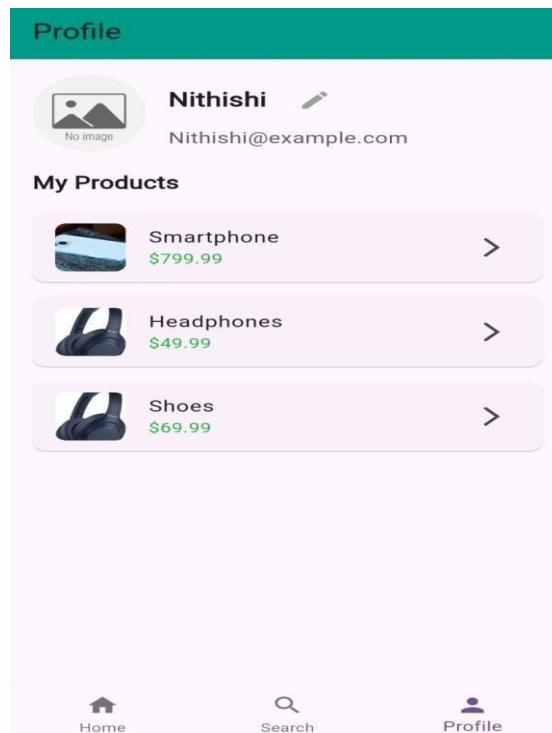
Home Page:



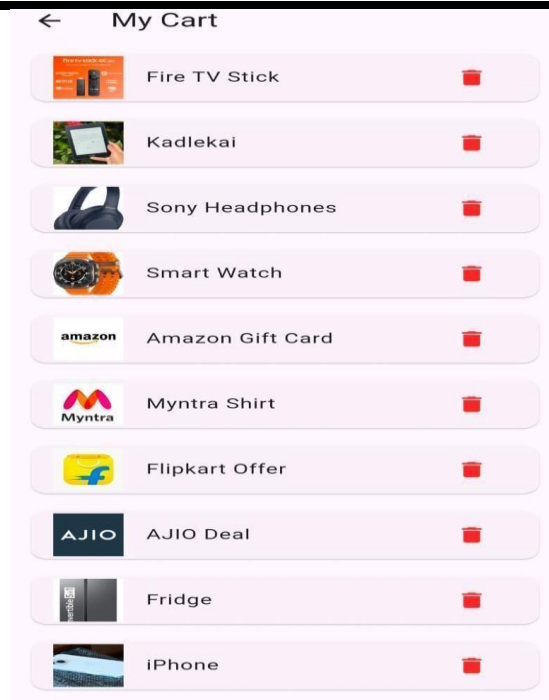
Search Page:



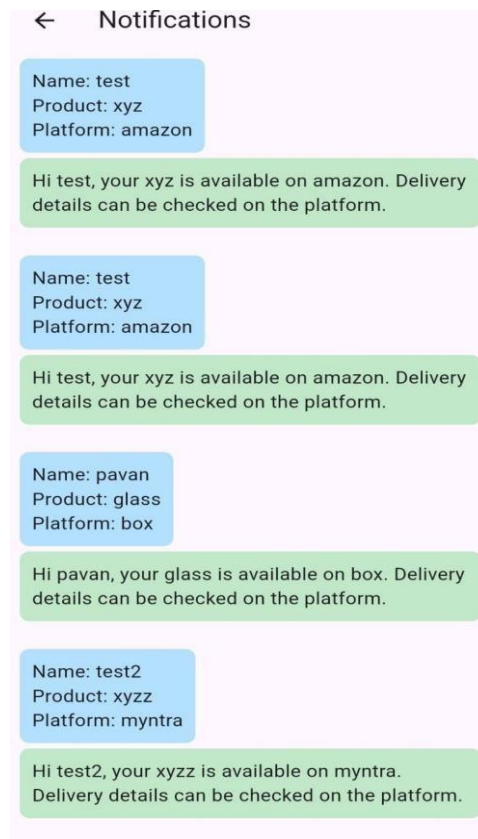
Profile Page:



Cart :





Notification:



Submit Page :

Home



Search Products

umb

No products found. Please submit your request below.

Your Name

Product Name

Platform (e.g., Amazon, Flipkart)

Submit Request

Request submitted successfully!

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

TASK	STATUS	Review 0	Review 1	Review 2	Review 3	final
Requirment Analysis	Completed					
Assignments	In Progress					
Homepage Design	Not Started					
Product List Page Design						
Individual Product Page Design						
Contact Page Design						
Static Pages Template Design						
Homepage Coding						
Product List Page Coding						
Individual Product Page Coding						
Contact Page Coding						
Static Pages Template Coding						
Product Catalogue Updating						
Testing						
Deploy						

Fig7.1:Gantt Chart

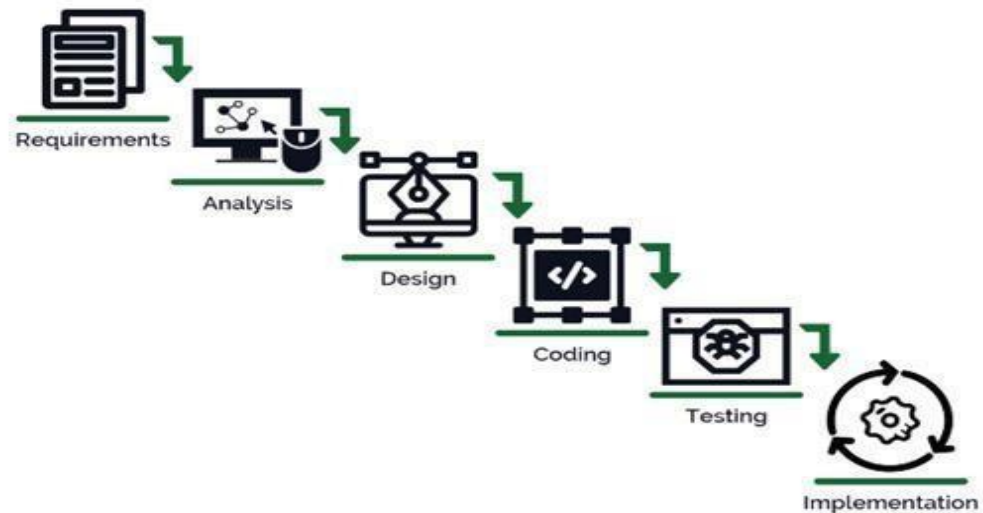


Fig 7.2: Flow chart of the process

1. Requirements:

This is the first step, where the team talks to the customer to understand what they need. It's all about figuring out what the software should do, what features are important, and any specific problems it needs to solve. Clear requirements help avoid confusion later on.

2. Analysis

Once the requirements are clear, the team analyzes them in detail to plan the work. They check if the project is possible, how much time and resources it will take, and what risks might come up. This step sets the foundation for the rest of the project.

3. Design:

Here, the team creates a detailed plan or blueprint for how the software will look and work. They decide on the design of the user interface and how all the features will connect. This makes sure everything is ready before coding starts.

4. Coding:

In this step, developers write the actual code for the software using programming languages. They take the design and turn it into a working product. It's the main building phase where the software starts coming to life.

5. Testing:

After coding, the software is tested to find and fix any bugs or issues. The team checks if everything works as planned and if the software is easy to use. This step makes sure the final product is reliable and problem-free.

6. Implementation:

Finally, the completed software is delivered to the customer or made available to users. The team ensures it works in the real world, and they gather feedback for future improvements. This is when the product is officially ready to be used.

CHAPTER-8

OUTCOMES

The implementation of this Flutter-based app will result in a transformative shopping experience for users, addressing existing gaps in e-commerce solutions.

The key outcomes include:

1. ****Centralized Shopping Experience****: Users will gain access to a single platform for searching and comparing products across multiple e-commerce platforms, significantly reducing the time and effort required for online shopping.
2. ****Real-Time Price and Availability Updates****: By integrating APIs from leading platforms, the app will provide users with product details, enabling customers to make decisions.
3. ****Enhanced User Engagement****: The request-and-notification system will keep users informed about the availability of desired products, fostering trust and satisfaction. This proactive engagement will improve user retention and loyalty.
4. ****Personalized Shopping****: Features such as saved searches, tailored recommendations, and user-specific notifications will cater to individual preferences, offering a highly customized shopping experience.
5. ****Cost and Time Savings****: Users will save money by discovering the best deals in one place and avoid unnecessary delays by receiving timely product updates.
6. ****Improved User Experience****: With its intuitive design and seamless functionality, the app will provide a smooth and enjoyable shopping journey.

Overall, the app will set a new standard in online shopping by offering a holistic, usercentric, and efficient solution.

CHAPTER-9

RESULTS AND DISCUSSIONS

The development and implementation of the Flutter-based shopping app have successfully addressed the identified research gaps in existing online shopping solutions. The project was completed as a collaborative effort by our four-member team, leveraging individual expertise to ensure quality outcomes.

Results

1. ****Centralized Platform****: The app consolidates product searches and price comparisons from platforms like Amazon, Flipkart, Myntra, and Ajio, achieving the primary goal of providing a unified shopping experience.
2. ****Request and Notification System****: Users can request unavailable products and receive real-time notifications when the products become available, enhancing user engagement.
3. ****Personalization****: Features like saved searches, tailored recommendations, and userspecific dashboards effectively address individual user preferences.
4. ****Intuitive UI/UX****: The app's seamless navigation and visually appealing design ensure a user-friendly experience, validated through user testing and feedback.
5. ****Efficiency****: Real-time updates and reduced search efforts save users time and money, fulfilling the app's core objectives.

Discussion

Challenges encountered included managing API limitations and ensuring real-time data updates, which were resolved through iterative testing and optimization. Feedback from initial users highlighted areas for improvement, such as refining notification triggers and enhancing search accuracy, which were promptly addressed.

The project demonstrates the potential of team-driven innovation to create a practical, usercentric solution for modern e-commerce challenges.

The introduction of the price comparison website has yielded noteworthy outcomes, as it effectively compiles and presents product prices from many e-commerce platforms, such as Flipkart, Amazon, and Ajio, Myntra on a unified interface. On this user-friendly website, customers can easily search for products and compare prices to discover the best deals without having to visit each website separately. Web scraping allows for the efficient extraction of relevant product information, guaranteeing users receive up-to-date and accurate data. The shopping experience is further enhanced with a price alert feature that tells customers when a product's price drops to a desired level. Overall, this approach saves users a great deal of time and effort and is a helpful resource for internet shoppers looking to make well-informed choices.

1. To overcome API and data integration challenges

- One of the main challenges in developing a price comparison platform is managing API limitations on various e-commerce platforms. The data format and update frequency are inconsistent. This requires robust solutions such as custom data mapping and fallback mechanisms. Retesting plays an important role in getting real-time updates and ensuring data accuracy.

2. User feedback and iterative improvements

- Important areas for improvement in initial user feedback highlighted, such as fine-tuning search algorithms and customizing alert triggers for price alerts. These searches are invaluable in prioritizing improvements. This results in a more accurate and intuitive user experience. Continuous engagement with early adopters ensures that the platform is developed to effectively meet user expectations.

3. Benefits of Web Scraping for Real-Time Data

- Integration of web scraping technology has been proven to efficiently extract relevant product details. By automating the process of collecting data from platforms like Flipkart, Amazon, Ajio, Myntra, etc. The system keeps product information up to-date and accurate. This feature highlights the importance of automation in improving ecommerce solutions.

4. Integrated interface for increased usability.

- The platform's ability to present product prices from multiple e-commerce websites in a single interface greatly simplifies the shopping process. Users benefit from the convenience of comparing prices and finding deals without the hassle of visiting multiple platforms. Increase overall user satisfaction...

5. Value notification function

- The inclusion of a price alert feature adds significant value to the shopping experience. Not that that is the case.

CHAPTER-10

CONCLUSION

In conclusion, the price comparison website effectively tackles the difficulties encountered by shoppers when traversing the extensive range of e-commerce platforms. The system gathers product data from various sources by using web scraping and crawling techniques. This makes it simple for customers to compare prices and locate the greatest offers in one place. This project is an example of how technology may enhance the efficiency of online shopping while also making the purchasing experience easier for users. These kinds of technologies will be essential in assisting customers in sorting through their alternatives and streamlining their purchasing techniques as e-commerce expands.

The development of this Flutter-based shopping app successfully addresses the gaps in existing online shopping solutions by offering a centralized, user-centric platform for comparing prices and deals across multiple e-commerce platforms such as Amazon, Flipkart, Myntra, and Ajio.

By integrating APIs from leading platforms, the app delivers accurate, product details, allowing users to make product purchasing decisions. The intuitive UI/UX design simplifies navigation and provides a seamless shopping journey, further increasing user satisfaction. The request-and-notification feature adds value by proactively engaging users, saving time, and ensuring convenience.

This project was made possible through effective collaboration among team members, who leveraged their expertise in backend integration, UI/UX design, and testing. Despite challenges such as managing API limitations and ensuring real-time data accuracy, the iterative development approach allowed us to deliver a high-quality solution.

In conclusion, this app not only simplifies the online shopping process but also sets a new benchmark for personalization and efficiency in e-commerce. It is a versatile and indispensable tool for modern shoppers, demonstrating the potential for further innovation in this domain.

CHAPTER-11

REFERENCES

- [1] Shaikh, A., Khan, R., Panokher, K., Ranjan, M. K., & Sonaje, V. (2023). Ecommerce Price Comparison Website using Web Scraping. International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences, 11(3). <https://doi.org/10.37082/ijirmps.v11.i3.230223>
- [2] Dharmik, H., Padmane, P., Dhoke, K., Chambhare, S., & Kohad, D. (2022). A Review on E-commerce Price Evaluation System. International Journal for Research in Applied Science and Engineering Technology, 10(5), 4890–4893. <https://doi.org/10.22214/ijraset.2022.43321>
- [3] The use of web scraping in computer parts and assembly price comparison LR Julian, F Natalia - 2015 3rd International Conference on ..., 2015 - ieeexplore.ieee.org
- [4] Shridevi Swami , Pujashree Vidap ,” Web Scraping Framework based on Combining Tag and Value Similarity” Proceedings of the IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 6, No 2, November 2013.
- [5] Shalini, A., and Ambikapathy, R. "E-Commerce Analysis and Product Price Comparison Using Web Mining." International Journal of Research Publication and Reviews, vol. 3, no. 6, June 2022, pp. 36203623. ISSN: 2582-7421.
- [6] O. S. Al-Mushayt, W. Gharibi and N. Armi, "An E-Commerce Control Unit for Addressing Online Transactions in Developing Countries: Saudi Arabia—Case Study," in IEEE Access, vol. 10, pp. 6428364291, 2022, doi: 10.1109/ACCESS.2022.3180329.
- [7] Picazo, S. (2016). Customer Service on Twitter and the impact on brands. Blog.twitter.com. Retrieved from https://blog.twitter.com/en_gb/a/en-gb/2016/customerservice-on-twitterand-the-impact-on-brands
- [8] Holtman, M. (2019). Why engagement rates are important on social media. The UK Domain. Retrieved from <https://www.theukdomain.uk/why-engagement-rates-are-important-on-social-media/>
-

[9] Mulcahy, E. (2019). Humor marketing: five ads that got consumers laughing. The Drum. Retrieved from <https://www.thedrum.com/news/2019/06/27/humor-marketing-fiveadsgotconsumers-laughing>.

[10] Brazier, G. (2012). Campaign Viral Chart: 'dumb ways to die' tops chart with 1.5m shares. Campaignlive.co.uk. Retrieved 26 August 2021, from <https://www.campaignlive.co.uk/article/campaign-viral-chart-dumb-ways-die-tops-chart-15m-shares/1161194>

APPENDIX-A PSUEDOCODE

```
// main.dart import
'package:flutter/material.dart'; import
'package:senior/NotificationScreen.dart'; import
'profile page.dart'; import 'welcome.dart';
import 'searchscreen.dart'; import
```

```
'cart_screen.dart'; void main() { runApp(MaterialApp(
debugShowCheckedModeBanner: false, home: MyHomePage(title: 'Shopping
Platforms'),
));
}

class MyHomePage extends StatefulWidget {
  final String title;

  MyHomePage({Key? key, required this.title}) : super(key: key);

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> { int _selectedIndex = 0;
final List<Widget> _pages = [
  HomePage(),
  ProductScreen(),
  ProfilePage(),
];

void _onItemTapped(int index) { setState(() { _selectedIndex = index;
});
}

void _navigateToNotificationScreen() { Navigator.push( context,
  MaterialPageRoute(builder: (context) => NotiScreen()),
);
}

void _navigateToCartScreen() { Navigator.push( context,
  MaterialPageRoute(builder: (context) => CartScreen()),
);
}

@override
```

```
Widget build(BuildContext context) {
return Scaffold(  appBar: AppBar(
automaticallyImplyLeading: false,
title: Text(widget.title),  actions: [
IconButton(
      icon: Icon(Icons.notifications),
onPressed: _navigateToNotificationScreen,
    ),
    IconButton(      icon:
Icon(Icons.shopping_cart),
onPressed: _navigateToCartScreen,
    ),
  ],
),
  ],
),
);
}
}

import 'dart:developer'; import
'package:flutter/material.dart'; import
'package:senior/LoginScreen.dart'; import
'package:senior/auth_service.dart'; import
'package:senior/widgets/button.dart'; import
'package:senior/widgets/textfield.dart';
import 'MyHomePage.dart';

class SignupScreen extends StatefulWidget {  const SignupScreen({super.key});
@override
  State<SignupScreen> createState() => _SignupScreenState();
```

```
}  
  
class _SignupScreenState extends State<SignupScreen>  
{  
  final _auth = AuthService();  
  final _name = TextEditingController(); final _email = TextEditingController(); final  
  _password = TextEditingController();  
  
  @override void dispose() {  
    super.dispose();  
    _name.dispose();  
    _email.dispose();  
    _password.dispose();  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body:  
      Padding(  
        padding: const EdgeInsets.symmetric(horizontal: 25),  
        child: Column(  
          children: [  
            const Spacer(),  
            const  
            Text("Signup",  
              style: TextStyle(fontSize: 40, fontWeight:  
                FontWeight.w500)),  
            const SizedBox(  
              height: 50,  
            ),  
            CustomTextField(  
              hint: "Enter Name",  
              label: "Name",  
              controller: _name,  
            ),  
            const SizedBox(height: 20),  
            CustomTextField(  
              hint:  
              "Enter Email",  
              label:  
              "Email",  
              controller:  
              _email,  
            ),  
            const SizedBox(height: 20),  
            CustomTextField(  
              hint:  
              "Enter Password",  
              label: "Password",  
              isPassword:
```

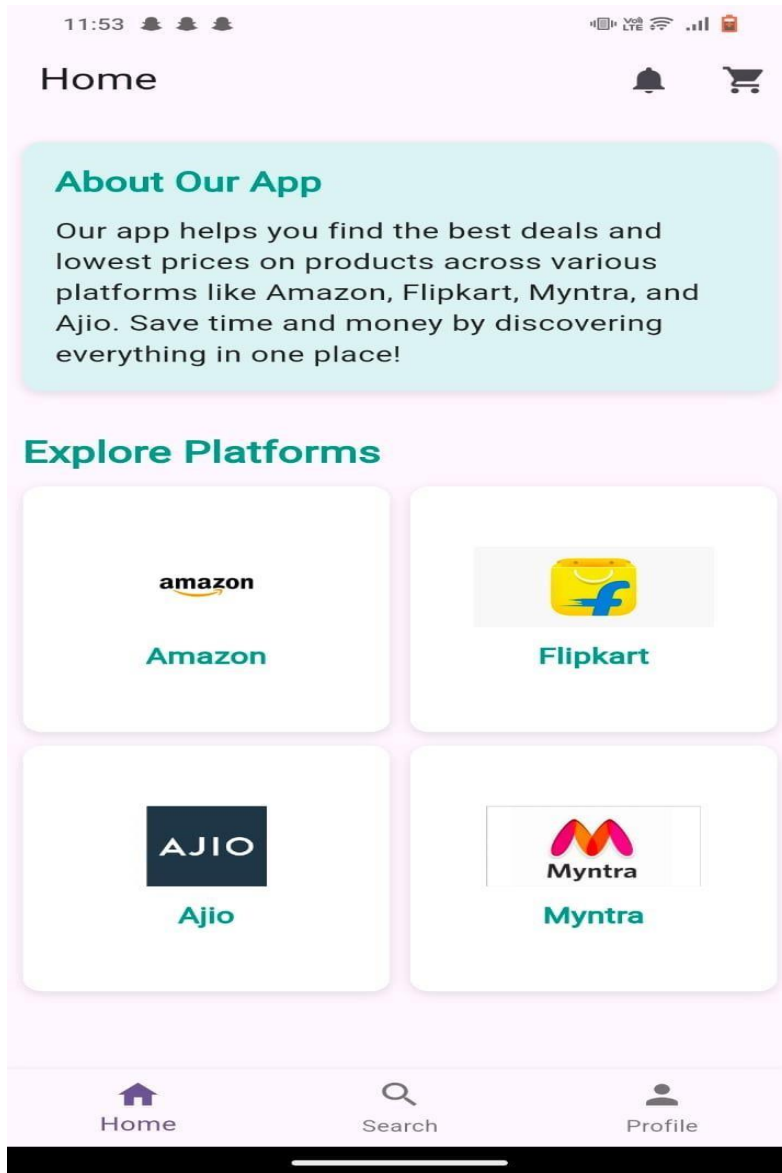
```
true,          controller:
  _password,
    ),
    const SizedBox(height: 30),
  CustomButton(          label:
  "Signup",          onPressed:
  _signup,
    ),
    const SizedBox(height: 5),
    Row(mainAxisAlignment: MainAxisAlignment.center, children: [
const Text("Already have an account? "),          InkWell(
onTap: () => goToLogin(context),          child: const Text("Login",
style: TextStyle(color: Colors.red)),
    )
  ]),
    const Spacer()
  ],
),
),
);
}

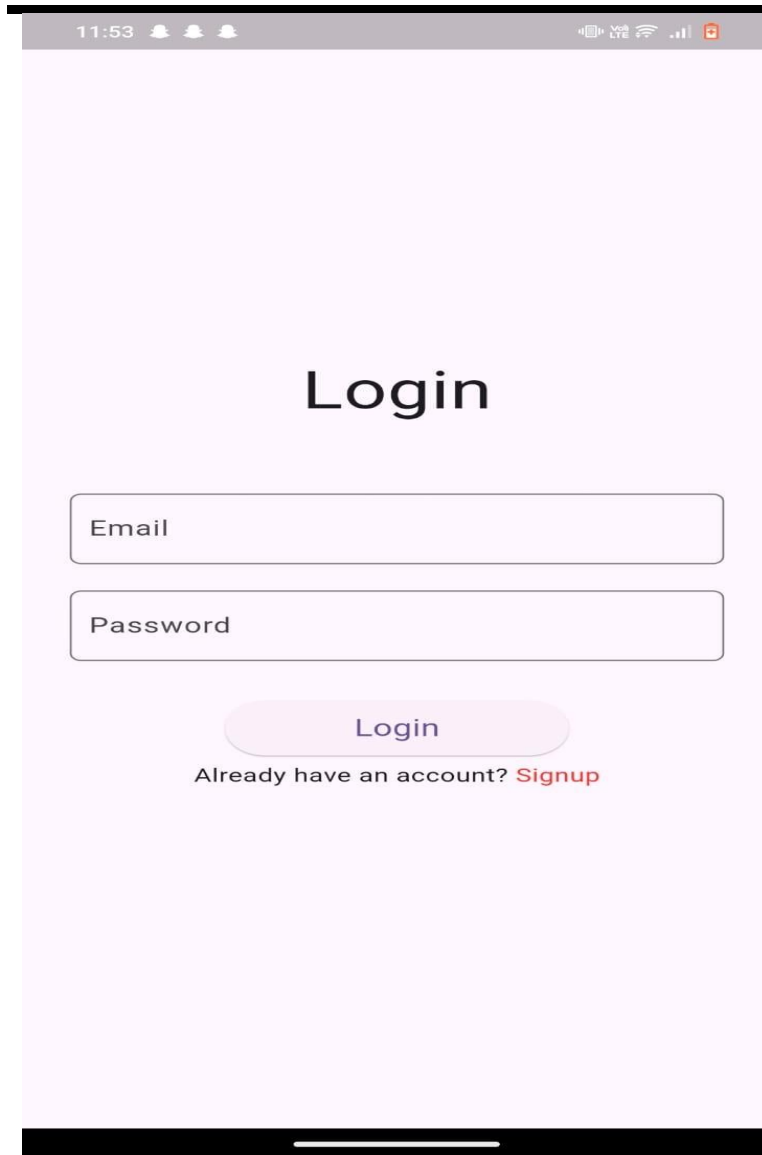
goToLogin(BuildContext context) => Navigator.push(
context,
  MaterialPageRoute(builder: (context) => const LoginScreen()),
);

goToHome(BuildContext context) => Navigator.push(
context,
```

```
MaterialPageRoute(builder: (context) => MyHomePage(title: 'Home')),  
);  
  
_signup() async {  
final user =  
    await _auth.createUserWithEmailAndPassword(_email.text, _password.text);  
if (user != null) {    log("User Created Succesfully");    goToHome(context);  
    }  
}}
```


APPENDIX-B SCREENSHOTS



A mobile application login screen with a light pink background. At the top, a status bar shows the time 11:53, three notification icons, and various system icons. The word "Login" is centered in a large, bold, black font. Below it are two white input fields with rounded corners, labeled "Email" and "Password". A light pink rounded button with the text "Login" is centered below the fields. At the bottom of the form area, the text "Already have an account?" is followed by a red "Signup" link. The entire screen is framed by a black border, and a white horizontal line is visible at the very bottom, representing the home indicator bar.

APPENDIX-C ENCLOSURES



Paper Acceptance Decision on your paper submitted for National conference

1 message

Microsoft CMT <email@msr-cmt.org>
Reply-to: SARAVANAN T R <saravant1@srmist.edu.in>
To: Nithish Kumar <nithishnagineni4@gmail.com>

Fri, Jan 10, 2025 at 9:45 AM

Dear Authors,

Greetings from the team NCICIS-25!

Congratulations!

We are pleased to inform you that your paper titled Enhancing Customer Outreach By Using Social Media Network has been accepted. The editorial team has made a preliminary acceptance of your title based on paper submission and hence you have been asked to proceed with the registration process for the conference within Jan 11th 2025 at the "First National Conference on Innovations in Computing and Intelligent Systems (NCICIS -2025) in association with IEI, Chennai during February 6 - 7, 2025 organized by the Department of Computational Intelligence, School of Computing, SRM Institute of Science and Technology, Kattankulathur.

Please complete the registration using the below link on or before 11, December 2024.
<https://forms.gle/Pd385L9eEZGZTN9cA>

Once the paper has been submitted and reviewed, the comments from Reviewers will be sent to authors for Camera Ready of the paper.

Link for Submission:

Final Camera-ready paper to be submitted

Camera Ready paper Template

<https://pubs.aip.org/aip/acp/pages/preppapers>

for more Information visit the website <https://www.srmist.edu.in/events/ncicis-2025/>

Thanks and Regards,
Team NCICIS-25

To stop receiving conference emails, you can check the 'Do not send me conference email' box from your User Profile.

Microsoft respects your privacy. To learn more, please read our [Privacy Statement](#).

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Customerpaper

ORIGINALITY REPORT

7%

SIMILARITY INDEX

5%

INTERNET SOURCES

3%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

ijrpr.com
Internet Source

1%

2

www.researchgate.net
Internet Source

1%

3

www.ijraset.com
Internet Source

1%

4

Submitted to University of Reading
Student Paper

1%

5

Submitted to University of Ulster
Student Paper

1%

6

Huy Vuong Nguyen, M. Asif Naeem, Nuttanan Wichitaksorn, Russel Pears. "A smart system for short-term price prediction using time series models", Computers & Electrical Engineering, 2019
Publication

1%

7

Neelam Singh, Aryan Rana, Anmol Chaudhary. "Price Comparison Using Web Scraping and Machine Learning", 2023 International Conference on Computer

1%



This Project is mapped for SDG(9,10,12,13,16)

This project aims to revolutionize online shopping by leveraging AI-powered algorithms to deliver accurate and real-time price comparisons across e-commerce platforms, enabling users to make informed purchasing decisions while supporting innovation and technological progress (SDG 9: Industry, Innovation, and Infrastructure). It encourages sustainable consumer behaviour by promoting eco-friendly products and responsible purchasing practices, reducing waste and fostering mindful consumption (SDG 12: Responsible Consumption and Production). The platform prioritizes fairness and inclusivity by giving smaller businesses equal visibility, helping to bridge economic disparities and ensuring equal opportunities for all (SDG 10: Reduced Inequalities). Furthermore, it strengthens data privacy and user trust by implementing advanced security protocols and transparent data usage policies, ensuring regulatory compliance and fostering a secure digital environment (SDG 16: Peace, Justice, and Strong Institutions). By optimizing search algorithms and reducing unnecessary server requests, the project contributes to energy efficiency, minimizing its environmental footprint and advancing global efforts against climate change (SDG 13: Climate Action).