WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

E-Commerce application with Flutter framework

Punam Amipara [1],Priyanshu Kher[2], Pragati Deo[3], Sanket Patel [4], Priyanshu Panchani [5], Nitin Pal[6], Ashish Dubey[7], RaviRanjan Kumar Pandey[8]

Computer science & engineering department, Parul University
Vadodara, Gujarat, India
210305105467@paruluniversity.ac.in
210305105464@paruluniversity.ac.in
210305105450@paruluniversity.ac.in
210305105499@paruluniversity.ac.in
210305105462@paruluniversity.ac.in

ABSTRACT:-

E-commerce has completely changed how companies run by making things more convenient and accessible for both buyers and sellers.. This paper presents the development of an e-commerce application using Flutter and Firebase, designed specifically for small businesses that struggle with the high costs of major e-commerce platforms. The application offers a cost-effective, one-time investment model while ensuring a seamless shopping experience. Secure transactions, product listing, order management, and user identification are important aspects. The app supports two profiles: users (customers), who can browse and purchase products, and business owners, who can manage their own stores and inventory.The application guarantees a consistent user experience on both iOS and Android devices by utilising Flutter's cross-platform features. Scalability, security, and real-time data synchronisation are all improved by the Firebase integration. This study highlights the application's development process, key functionalities, and the impact of using modern mobile technologies in empowering small businesses in the digital economy.

1. Introduction :

In the digital area, e-commerce applications have revolutionized the way businesses operate and consumers shop. With the rapid advancement of technology, mobile applications have become the backbone of online shopping, providing convenience, accessibility, and efficiency. Among various frameworks available for mobile app development, Flutter has emerged as a popular choice due to its cross-platform capabilities, high performance, and flexible UI components.

This research focuses on the development of an e-commerce application using Flutter and Firebase, tailored specifically for small businesses that cannot afford the high costs associated with major e-commerce platforms. The application provides a one-time investment model, allowing business owners to showcase and sell their products without recurring platform fees. The app ensures a seamless user experience by integrating essential e-commerce functionalities, including user authentication, product listing, order management, and payment processing.

The objective of this study is to explore how Flutter's single codebase and Firebase's backend services can enhance the efficiency and scalability of e-commerce applications. Additionally, the paper examines how such a solution can empower small businesses by reducing technical and financial barriers to digital commerce.

2. Background and Purpose :

2.1 Project Background

Electronic commerce, generally known as online business and trade, alludes to the acquiring and selling of products and services well over the internet. In the 1960s, the term "ecommerce" was initially used. Social networking has become a powerful tool for enhancing a website's strength as mobile device usage increases.

Salesforce The phrase "electronic commerce" refers to any online activity that involves the buying and selling of products and services. E-commerce, to put it another way, is the practice of conducting business online. When Phil Branden Berger purchased Sting's "Ten Summoner's Tales" online in 1994, e-commerce officially began. It is predicted that 27.2 percent of people worldwide will purchase online by 2021. (Oberlo, no date.)

To deliver information to individuals, several innovative strategies and innovations in e-commerce applications have been established. Globally, online shopping has experienced substantial growth. Twenty-eight percent of people worldwide shop online. You're missing out on a huge group of potential purchasers if you're not selling online. According to Salesforce, 73% of consumers explore over multiple channels, such as email, social media, and Google.

2.2 Object Projectives

In recent few years, online shopping has become very much popular. The preference of consumers worldwide for online shopping over physical stores is the reason behind the success of Amazon and other comparable online retailers, as well as the growing interest in online e-commerce. Darrell (2011) However, innovation is helping us overcome some of these obstacles in the modern world by employing computer frameworks to assist us with a variety of labour tasks. The world of internet shopping is expanding rapidly, as everyone is aware. Consequently, e-commerce applications have already created a number of novel strategies and technologies to provide people with information. However, humans face a lot of difficulties and worries as a result of these concepts and technology. (sgrlaw).

3. ANDROID APPLICATION REQUIREMENTS AND DEVELOPMENT :

3.1 Ecommerce Application Development

E-commerce has been around for more than 50 years. In the 1970s, early innovations like teleshopping and EDI (Electronic Data Interchange) were demonstrated, opening the door for modern e-commerce businesses. The history of the internet and e-commerce are intricately intertwined. Online shopping was made possible by the development of the internet in the 1990s. Since then, hundreds of e-commerce businesses have emerged as a result of Amazon becoming one of the most well-known e-commerce websites in the US. (No date, EverydayOnlineMarketing).

In order to illustrate e-shopping, Michael Aldrich connected a phone to a computer that processed transactions and an adjustable TV. (No date, Aldrich Archive)

The global market for online business transactions has risen at an annual pace of 33.7 percent since 2016, despite desktop spending remaining steady throughout the year. Customers' constant preparedness with every screen tap is the reason behind this. Simple logic suggests that the development of mobile applications for e-commerce businesses is a long-term trend. There is no indication that cellphones will soon be rejected by the global population. In contrast, it is estimated that by 2021, the value of diverse online purchasing would reach $3.5 trillion. (Shopify, undated)

The need to satisfy customer expectations in the face of growing online sales is driving the most noticeable expansion of flexible arrangements in the e-commerce industry. Consumers want to be able to buy goods and services whenever they want and from anywhere without having to travel. Because e-commerce apps and websites are more advanced, more comfortable, and more versatile, customers prefer them while making purchases.

For It's critical for developers of mobile applications to keep in mind the differences between mobile and traditional software development. Think about the different settings for using mobile apps vs desktop computers, such as acknowledging the limitations and complexity of a touch-based user interface and taking into account the particular security flaws that mobile technologies offer. The significance of user experience and the reality that most users are accustomed to the design standards of a certain operating system should also be kept in mind by a mobile developer. (No date, creativebloq)

3.2 Android App Development

The process of developing mobile applications for Android-powered devices is known as Android app development. Android applications are made using object-oriented languages (like Java), C, and the SDK (Android Software Development Kit). The Java programming language serves as the foundation for Android, which was first released in 2009. With the bulk of the market share and a global clientele, Android currently controls the entire portable app industry.

3.3 Functional Requirement

The following are the functional criteria for this application that must be met in order to finish this project:

- When the client clicks on the app icon, the application must launch.
- The application's home page must be displayed initially.
- The application must give the client the content list for the chosen category.
- The products must be viewable by the client.
- The customer must be able to see the product's picture.
- The application must give the customer a description of the chosen product.
- The product must be able to be added to the client's shopping cart via the application.
- The client's information form must be provided by the application.
- The client needs to send the application the credentials they have provided.

3.4 Non-Functional Requirements

The following are the non-functional requirements for this application that must be met in order to finish this project:

- An interface for displaying products to customers should be included in the application.
- An interface for displaying a list of categories should be included in the application.
- A list of categories should be displayed by the application so that users can click on them with ease.
- The description of a product should be able to be displayed by the application.
- An interface for adding products to the shopping cart should be included in the application.

4. Technology :

4.1 Flutter

Flutter technology is the foundation of this application. Since Flutter is a Google product, it has a number of benefits. Here is a brief introduction to Flutter. Flutter transforms the process of developing apps. Writing one-time code allows you to create, test, and deploy beautiful desktop, web, mobile, and embedded applications. Flutter was developed by Google as an open-source project. It is used to create hybrid programs from a single codebase for Google Fuchsia, Windows, Linux, macOS, iOS, Android, and the web. (No date, Flutter)

Hot Reload allows for rapid development and iteration. As soon as you update the code, you will notice changes without any loss of state. To produce distinctive, responsive designs that feel and look fantastic on all devices, take care to every pixel.

In Flutter, each component is called a widget. It is based purely on widgets. Widgets can be either stateful or stateless. A stateful widget has multiple states, not just one. Depending on the user's interest, they can change their status. Conversely, a stateless widget is a static widget used for static functionality or data. It has no state at all. Because we deal with static data, there are situations in which we must use it. (AppDividend; date unknown)

Anyone can use and contribute to Flutter at any level because it is an open-source project. Being a Google contributor is a better way to use it and share your enthusiasm for Google technology. Since Google is the most popular search engine in the world and Flutter is a Google product, it has a huge user base.

When it comes to pricing, Flutter performs better than other hybrid solutions. It's among Flutter's many benefits. With the help of the Common Crypto library and the iOS SecKey API, Flutter developers may encrypt app data using both symmetric and asymmetric keys. Dart, which contains 13 encryption libraries and numerous cryptos that use different cryptographic hashing and encryption methods, is used to write Flutter code.

4.2 Android

Android is a multipurpose mobile operating system that may be used on a variety of devices, including televisions, tablets, smartphones, and even household robots. Developed by 'Android Inc' and built on Linux, this operating system was acquired and sold by Google. Since its 2008 launch as an operating system (OS), Android has gained popularity among smart devices. Because to this operating system, contemporary smartphones and tablets could be regarded as tiny minicomputers.

The Android platform is expected to have over 35 billion users globally by 2020, making it the most widely used mobile operating system globally. This operating system powers many smart devices, including smartphones, tablets, and wearables. These days, any brand or company in the globe needs an Android e-commerce app to expand their clientele, revenue, and brand recognition. (Android, no date)

Getting to know Android takes less than an hour, and it has a tremendously fascinating and adaptable system. Any client can quickly configure OS settings because so many critical programs are available. Beyond recognition, you can change everything: just download a meaningful software from the Google Play Store if you do not like the design, symbols, or ringtone, and you may rapidly customize everything to your desire.

4.3 Dart Language

A client-oriented programming language called Dart makes it possible to create apps quickly on any platform. Developing one of the most productive languages across several platforms is its main objective. It will be advantageous to the user as well as the server. The dart2js tool, which creates the JavaScript version of a Dart script so that it may run on websites that don't support Dart, and the Dart VM compiler are both included in the Dart SDK. C++ and this object-oriented programming language are extremely similar. Dart is a well-liked programming language for making online apps and single-page webpages. (Dart, no date).

Dart is made to offer logic and, consequently, a visually appealing user interface. For backend, desktop, and mobile applications, gather specific

Benefits

The first benefit is that learning the language is quite simple. Because the grammar is similar, those of us who are familiar with C or C++ can pick up this language quickly. The second thing to consider is the large community, which offers a wealth of documentation. Additionally, they make things as easy as possible because it's a Google product. Its outstanding performance is its third benefit. Programs built with Dart run more quickly than those built with JavaScript. (Code Carbon, 2020)

Drawback

Of course, we have to weigh the disadvantages as well as the benefits. The first drawback is its lack of resources, which makes problem-solving more difficult. This is because there aren't many bigger, more unified development communities to help you.

Dart is being developed at the moment. That's great, and there's nothing "wrong" with Dart. However, if you start working in Dart right now, the API might change, the documentation might not be complete or correct, and there might not be as much information available online as there is on jQuery, for instance. Carbon Code, 2020

4.4 Google Firebase

Google's Firebase initiative enables developers to make apps for web-based apps, iOS, and Android, among other platforms. Firebase is a rapid tool for creating custom APIs and other things that would take a long time in a typical database 15 system. Firebase comes with built-in APIs that let us quickly build applications. The fact that Firebase provides us with tools for monitoring company reports and testing out various products is one of its best advantages.

Clients can access data instantly thanks to the Firebase database's real-time functionality. You can still use your application with all of your previous data sets even if you are not online. Firebase gives consumers the ability to use data in a safe and secure manner.

The Firebase database is a NOSQL database, similar to the MYSQL database used in PHP applications. The real-time database API was created exclusively for rapid operations and capabilities. (Firebase, no date).

4.5 Android Studio

It's a Google-created integrated development environment (IDE). The goal of Android is to speed up progress and make it easier for users to create high-quality apps for Android devices. Commonly used operating frameworks, such as Mac and Windows, allow variations of this IDE. It also provides a development kit and plugins for developers. cross-platform application support(IDE).

5. Technical Implementation :

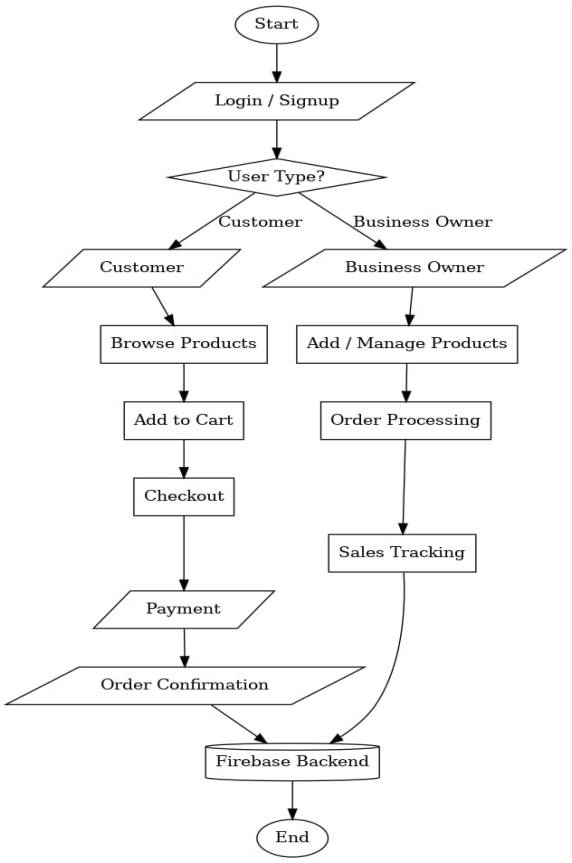
5.1 Planing

Planning what we wish to construct that will benefit our society is the first step in the technological execution of our project. Up till now, we have made an effort to consider every possibility that the software might touch:

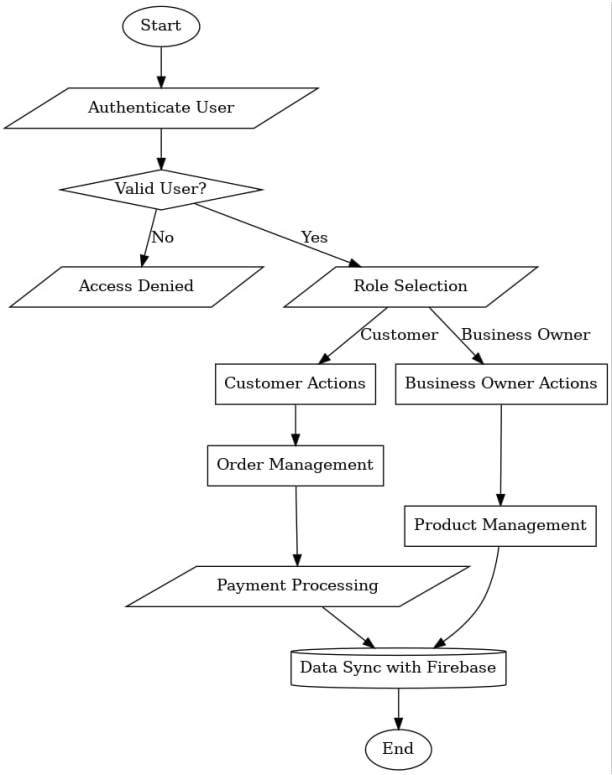
- Will the development happen all at once or in phases?
- How long is it going to take?
- Will there be additional security difficulties?

Lastly, our goal is to develop e-commerce software that enables customers to buy products without physically visiting a store or shop. This is a great way to alleviate the challenges that many people who are unable to visit the market to buy items experience. The only thing customers need to do is browse the entire store and select the items that best fit their needs.

5.1.1 Data Flow Diagram



5.1.2 Control Diagram



5.2 MVVM (Model-View-View Model)

The MVVM paradigm, which uses a well-known provider pattern, is the foundation of the app's present design. In this design, views keep a careful eye on the view model and respond to any changes they notice. In our example, the Data View Model has a reference to the home-view that is observing the Data View Model.

Similar to populating categories, the Data View Model's filling category method is in charge of adding categories to the list of categories and alerting the listener. A Horizontal List View widget in the Home-view keeps an eye on the category list and modifies the list View's content if the list is changed.

This also applies to filling out and classifying goods. Three methods are available in the Data View Model: getShoes(), getPurse(), and getShirts(). When we select one of the 17 categories, the Grid view widget, which is in charge of product presentation, gets notified and queries the items in that category. Product information are displayed on the product details page using the DataViewModel's Selected Product reference variable.

When we select a product from the Home Page, we make the necessary changes to the detail page's content and update the product's reference variable, which is displayed in the product details page. Additionally, we monitor the listOfCartProducts in DataViewModel. When we add a product from the details page to the cart, we update the listOfCartProducts in DataViewModel. When the list changes, this variable, which is monitored on the cart screen, updates the listView widget.

Additionally, we use Firebase authentication to handle users. Only verified users are able to place orders. For firebase configuration, we created a project for Angel. Within that project, we created an app for iOS and Android based on the bundle id for iOS and the package name for Android, respectively. Firebase creates an info-plist file for iOS and a google-services.json file for Android. After putting the files in the proper locations, we used the Firebase APIs to construct the authentication mechanism.

5.3 Configuration and Customization

Installing the programmer and setting up the features that can be utilised immediately are now necessary. Actually, before creating any processes or hastily customising any module, this stage should always be completed.

- Installing and configuring Flutter is the first step. The installation will then be done by following the guidelines on the official Flutter website.
- Before installing the flutter and dart plugins to use flutter in Android Studio, the second step is to install and set up Android Studio.
- In order to launch and test our app during this phase, we created an emulator in Android Studio.
- Making a home screen is the fourth stage. The home screen was then made using the Dart programming language.
- Following the concept of the home screen, we created goods and their product description pages. In the same vein, we have developed every design element independently.
- We have included add to cart functionality to our app so that users may select many items at once. You may also purchase multiple things at once by using the add to cart feature.
- Before placing an order, users must first register and log in using the new login and registration form. Flutter and Firebase were used for this.

5.3.1 Flutter Firebase working

When an Android or iOS client queries a Firebase API, the client submits its query to the Firebase database, which replies to both the client and the Firebase API.

This is a lib folder in Flutter that allows us to manage every screen of our application. The working section is also in this folder. 90% of our app development is done in this folder.

- Our library files are all managed using Pubspec.yaml. This part of our software is extremely secret. It is also limited by space.
- Firebase_core: ^1.16.0 and Firebase_auth: ^3.3.1719 are the two libraries in charge of Firebase integration.
- We store all of the assets for our Flutter application in the Flutter asset folder. Pictures, videos, static data, and further resources
- All iOS development activities are managed by the IOS folder. Being a hybrid technology, Flutter is capable of handling multiple platforms.
- Like iOS, it handles every task needed for Android development.

5.3.2 Firebase Configuration for Android and IOS:

We must first establish a Firebase console project in order to leverage Firebase with Flutter. On Firebase, we have created a project named Angel. We then included packages for iOS and Android. In the root directory of both iOS and Android, we placed an info plish file for iOS and a Google JSON file for Android. We configured our Firebase app for iOS and Android in this manner. Additionally, we have added authentication packages to the pub spec files.

The project id, which is basically our project's unique identity, was given to us. In order to set up integrations with Firebase or other third-party services, we also assigned a project number. Additionally, we acquired the public-facing name that will show up in emails sent to users following their account creation in our application.

When we were configuring and customising, we encountered the following problems.

- When setting up the Android application, we put the Google Services JSON file somewhere other than the root directory. As a result, unanticipated issues arise during application testing. But it doesn't function when we put the Google Services JSON file in the Android root directory.
- While managing sessions, we encountered an authentication issue with the application. The home screen is displayed when a user exits an application, saving them from having to log in again. This is an authentication issue. We've fixed it in our app.
- We are having trouble with the letters in our passwords. A password that contains fewer than six letters is not examined, and no error message appears. Now that the issue has been fixed, the password needs to have more than six letters.
- A product shows up in the cart screen when we add it. For us, this has been a major issue. An array's indexes were examined in order to solve this issue. Indexes are not stored in arrays. The cart consequently shows an empty value.
- Before placing an order, users are not required to log in. We were able to use session and authentication to fix this issue. In order to place an order or view the main home page, users must now be logged in.

6. Code Explanation :

6.1 Assistance from the open sources and friend

In addition to other sources, one of my friends helped me with the design. A friend made the splash screen, intro screens, and shipping details page. The design was taken from an online website and combined into one.

In order to design a portion of the code, we obtained a fairly good open-source code from GitHub. We then utilised portions of that code and modified it by following YouTube tutorials to include more components. (No date, Monsieur ZbanowanYY)
With YouTube's assistance, we replaced the search bar on the upper right corner of the application with the cart feature. Additionally, we have a shipment details box where we request buyer details like address and phone number. (2020, The Flutter Way)

6.2 Whole Application background Function calling

We configure a background function to run once when a new user launches an application for the first time. To keep people occupied, it features three or four guides. To obtain product data and show it in a grid on the home screen, the home screen widget calls the state widget when the home screen is activated.

We can navigate data by category with the help of a program called Navigator. We obtain a product's array index and fetch the desired object detail when a user touches on it.The product details page appears when a user taps on a product (which is sourced from an open source project). We match a product's ID with the array on the detailed screen, and when the id and array index match, we display data. (Unsplash)

We have fields for description, price, add to cart, and buy now on the detail screen. When the user hits "add to cart," it will be added to the cart. A user will be directed to the checkout page after clicking the "buy now" button. If the user is logged in, they can place an order; if not, they have to register for an app, log in, and then log in to their account.

The user will check the cart to see which goods have been added after adding products. After reviewing the things, he or she will check out. He or she must first log in before placing an order, as the login session is checked when they tap to check out again.

Additionally, we show products beneath each product's details screen so that the user can choose items from the same category. Libraries are used to manage user interface processes. Once the order has been placed, we simply show a Thank You screen. Due to the large number of screens in our program, we will go over the process for each screen and the bundled packages to clarify the code.

6.3 Home Screen

The following categories are available on our home screen: purses, shoes, and shirts. Each product belongs to a particular category. Angel_Home_Screen.Dart is the name of the dart file we created. Our primary home categories are controlled in this file.Next, we give the file the name prepare_data. We create a class called Data_View_Model in this file.

We are managing all of our goods, categories, and carts in this class. To minimise the number of files and optimise the code, we are keeping everything in one file. We create a method for every capability in this class. To retrieve every item in a purse, we create the get purse function in this class. We also provide get-shoe and get-shirts in a same logic.

For each category in which the products are maintained, we inserted loops. If there are ten products in a category, the loop will repeat ten times to display every product in that category.

Because this application uses provider patterns, the change notifier is extended in the main class. Listen to the list that has been prepared for this class by utilising this provider pattern. Any changes that take place in this class will be communicated to the provider. In essence, this is the greatest feature offered by providers for application data management.

We use the init function in our program. We are invoking our Data_View_Model class in the init function. Before running, everything is ready. Imagine that you launch an application. By default, just one category of products is visible. Note the change and send it to the listener that is calling the init() function when we click on another category notification. After then, the home screen data index is updated, and products from each index-specific category are called up and shown on the home screen.

6.4 Login Registration

We use a stateful widget on the login and signup page. We call the state of a stateful widget. We are using the scaffold widget in its current state. The scaffold is the arrangement of our entire screen that provides us with the App Bar, Body, and many other features.To align vertically, we are utilising a column widget. We have children in the column. We use text fields in Children. One for the password and another for the email.

We used create user with email and password APIs for registration, and we used sign in with email and password.
when a person inputs their password and email. A query is created to determine whether or not the user is registered in the Firebase database. When credentials match, a true response is sent, and the user is redirected to the screen where they can add details.We save the user's email and assign it a unique key when they check in to their account. We will offer them a new key when they attempt to log in again after pressing the logout button. We authenticate our users in this method.

We also manage sessions, which begin when a user logs into their account and end when they log out of an application. We are using Flutter's shared settings to manage sessions. We can control user sessions with a set time interval thanks to shared preferences. For use in Flutter, we have included a package in the pub standard.

shared_preferences: ^2.0.13

Applications utilise these packages to complete registration and login.We have access to the Flutter database through Firebase Core. We can also carry out initialisation tasks with Firebase Core. Firebase auth, on the other hand, enables us to verify users when they register or log in. These packages allow us to leverage Firebase APIs for registration or login.

firebase_core: ^1.16.0
firebase_auth: ^3.3.17

6.5 Add to Card Screen :

We use a stateful widget in the Add to Cart screen. We call the state of a stateful widget. We are using the scaffold widget in its current state. The scaffold is the arrangement of our entire screen that provides us with the App Bar, Body, and many other features. To align vertically, we are utilising a column widget. We have children in the column.We use the list view in Children. All of the selected items are arranged in the list view widget.
The add to cart feature also includes a session check. A user will use the add to cart feature to make a purchase after logging into their account and starting the session. Using sessions to purchase points and authenticate users is a really smart practice.

In our list view, we are retrieving data that has an index tagged as true. We are creating edit and delete buttons for each item in the list view. To proceed to checkout, we use the elevation button below the list display. There is an on-tap feature for the elevation button. We added on-tap capability to the checkout page path.

7 Discussion and Further Development :

Online shopping has become more and more common in recent years. Amazon and other similar online stores are not only successful but also attracting interest in online business because people worldwide prefer to make purchases online rather than in physical stores. In order to provide users with information, e-commerce apps have developed a number of innovative tactics and technologies. The global online e-commerce industry is expanding significantly.

They will require time and resources, but we have answers. Users will receive modules from this app to assist them in their task. Prioritising quality above quantity was the player's main objective when creating the app. As previously said, further modules and features will be added soon to strengthen and expand the software.

This program's main objective is to facilitate and support individuals in using e-commerce platforms, which are now widely used globally for online shopping and the purchasing of goods and commodities. As a result, people can feel completely assimilated into society and enjoy, readily access, and buy anything through online shopping. Our work will not stop here; we will keep refining the program and adding new features for persons who are totally blind and cannot see the world from any perspective. Our goal is to include a feature that will allow blind people to use this app completely; they will be able to dress themselves as they choose and operate it with voice commands. We're going to add more features as time goes on.

REFERENCES :

1. Abilitynet. 2013. How to use TalkBack in Android Jelly Bean. Read on 25.05.2022. https://mcmw.abilitynet.org.uk/talkback-android-jelly-bean-4-2-2
2. Android Accessibility Help. 2022. Get started on Android with TalkBack. Read on 11.05.2022. https://support.google.com/accessibility/android/answer/6283677?hl=en
3. AppDividend. no date. Flutter Mobile Development Guide For Beginners. Read on 08.05.2022. https://appdividend.com/category/flutter/
4. Code Carbon. 2020. Major Advantages And Disadvantages Of Dart Language. Read on 15.05.2022. https://codecarbon.com/pros-cons-dart-language/
5. Collin, C. 2021. 5 Critical Steps for Your Software Implementation Plan. Read on 11.05.2022. https://www.softwareadvice.com/resources/software-implementation-plan/
6. Dart. No date. Dart programming language | Dart. Read on 05.05.2022. https://dart.dev/
7. Darrell, R. 2011. The Future of Shopping. Harvard Business Review. Read on 08.05.2022. https://hbr.org/2011/12/the-future-of-shopping
8. Everyday. No date. Essential reads: Where marketing meets strategy - Everyday Online Marketing. Read on 05.05.2022. https://everydayonlinemarketing.com/
9. Firebase. No date. Firebase documentation: Learn the fundamentals. Read on 01.05.2022.28 https://firebase.google.com/docs
10. Flutter. no date. Flutter: Build apps for any screen. Read on 10.05.2022. https://flutter.dev/
11. GeeksforGeeks. No date. A computer science portal for geeks. Read on 06.05.2022. https://www.geeksforgeeks.org
12. Google. 2022. Everything you need to build on Android. Read on 24.05.2022. https://developer.android.com/studio/features
13. James, H. 2022. What is Java? Definition, Meaning & Features of Java Platforms. Read on 12.05.2022. https://www.guru99.com/java-platform.html
14. Lockwood, B. no date. 10 Technology Challenges. Read on 03.05.2022. https://sgrlaw.com/ttl-articles/10-technology-challenges/
15. Michael Aldrich Archive. No date. Welcome to the Michael Aldrich Archive. Read on 09.05.2022. https://www.aldricharchive.co.uk/
16. MonsieurZbanowanYY. 2022. E-Commerce-App-UI-Flutter. Read on 27.05.2022. https://github.com/MonsieurZbanowanYY/E-Commerce-App-UI-Flutter
17. Nazmun Nessa Moon , Shaheena Sultana, Fernaz Narin Nur & Mohd Saifuzzaman. 2017. A Literature Review of the Trend of Electronic Commerce in Bangladesh Perspective. Read on 23.05.2022. https://globaljournals.org/GJMBR_Volume17/3-A-Literature-Review-of-the-Trend.pdf

18. Oberlo. 2022. FULFILL YOUR DROPSHIPPING DREAMS. Read on 21.05.2022. https://www.oberlo.com/

19. Pravine, V. no date. A 5-STEP PLAN FOR SOFTWARE IMPLEMENTATION SUCCESS. Read on 18.05.2022. https://www.chetu.com/blogs/technical-perspectives/software-implementationplan.php

20. Richard F. Schmidt. 2013. Chapter 19 - Software Implementation. Read on 25.05.2022. https://www.sciencedirect.com/science/article/pii/B9780124077683000197

21. Salesforce, Inc. 2022. CRM Software; Cloud Computing Solutions - Salesforce EMEA. Read on 05.05.2022. https://www.salesforce.com/eu/