A Project Report

on

Comprehensive Online Shopping Platform Utilizing AR

Submitted to

Amity University Dubai

In partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology in
Computer Science & Engineering

by

Mr. Christon Andrew D'souza & Mr. Danish Shaikh

Under the guidance of

Ms. Lipsa Sadath

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We, Christon Andrew D'souza and Danish Shaikh, students of B. Tech (CSE) hereby declare that the project titled "Comprehensive *Online Shopping Platform"* which is submitted by me to Department of Computer Science and Engineering, Amity University Dubai, in partial fulfillment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering, has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

Date: May 30, 2023

Danish Shaikh

Enrollment No.:

A40105219029

Christon Andrew D'Souza

Enrollment No.:

A40105219010

CERTIFICATE

This is a certification that Christon Andrew D'souza and Danish Shaikh, students of B. Tech Computer Science & Engineering, have carried out the work presented in the Major Project report titled Comprehensive Online Shopping Platform as a part of the Fourth Year program of Bachelor of Technology (Computer Science and Engineering) from Amity University Dubai, UAE.

Ms. Lipsa Sadath

Department of Computer Science and

Engineering Amity University Dubai.

ACKNOWLEDGEMENT

We must begin by expressing our heartfelt gratitude and profound appreciation to the Almighty, for bestowing upon us His boundless blessings throughout our arduous journey of study. It is with an overwhelming sense of gratitude that we humbly acknowledge the Divine guidance and unwavering support that have propelled us forward, culminating in the successful completion of this profound research endeavor.

We want to convey our profound gratitude to Ms. Lipsa Sadath, our faculty mentor, for her wisdom, direction, and supervision while we completed our project.

We would also like to express our heartfelt gratitude to the University and all of the professors for their unwavering support, resources, and continual monitoring, as well as for allowing us to work on this project.

Finally, We would like to express my gratitude to our parents and friends for their unwavering support and guidance throughout.

ABSTRACT

The world of today has seen the quick, constant growth and expansion of technology, which has changed society's lifestyle and way of existence. One of the many benefits of technology is the efficient execution of any activity that would otherwise involve human contact. The online shopping method, which involves purchasing products from the comfort of anywhere in the world, has been around for a while. However, the systems' execution through online platforms has been mediocre to say the least. We developed this platform with an intuitive User Interface, cleaner slate and bolder look, and zero buggy occurrences in the purchasing system. However, our standout feature is the presence of an Augmented Reality based view for commodities placed on sale. Instead of the traditional photo-or-video format description of an item, we have the ability to place the object as a 3D model in the real world environment so that the shopper can have a detailed look of the product.

TABLE OF CONTENTS

1. INTRODUCTION........ 10
2. LITERATURE REVIEW........ 11
3.METHODOLOGY........ 13
3.A. Problem Statement........13
3.B. Objective........ 14
3.C. Augmented Reality........16
3.D. E-Authentication........21
3.E. Software Design........23
3.F. Gathering the requirements/Planning........ 27
3.G. Designing the UI........28
3.H. Augmented Reality integration........29
3.I. Frontend implementation........30
3.J. Backend design........31
3.K. Sign In request........ 34
4.RESULTS AND DISCUSSION........35
5. CONCLUSION........41
6. FUTURE PROSPECTS........ 42
7. LIMITATION........43
8. INDIVIDUAL CONTRIBUTION........44
9. REFERENCES........45

TABLE OF IMAGES

Figure 1: UML Diagram……………………………………………………………………….. 12

Figure 2: Risk Analysis Matrix……………………………………………………………… 13

Figure 3: User Interface XML……………………………………………………………….. 29

Figure 4: Thumbnail Images………………………………………………………………… 31

Figure 5: Homepage Code…………………………………………………………………... 31

Figure 6: Android Virtual Device…………………………………………………………….. 33

Figure 7: HomePage 1……………………………………………………………………….. 34

Figure 8: HomePage 2……………………………………………………………………….. 34

Figure 9: Landscape Orientation…………………………………………………………… 35

Figure 10: Buy Now/Add To Cart……………………………………………………….. 36

Figure 11: Product Page……………………………………………………………………. 37

Figure 12: AR Enabled……………………………………………………………………….. 38

1. INTRODUCTION

It has long been practiced to purchase goods using the traditional process, which includes going to a store and physically analyzing the products before making a purchase decision. However, the systems' performance on online platforms has been, to put it mildly, unsatisfactory. This project was started with the goal of creating an online shopping platform that prioritized user friendliness. We designed this platform with a simple User Interface, a blank canvas and a bolder appearance, and no instances of problematic behavior in the purchasing mechanism. However, our unique feature is the availability of AR views for goods offered for sale. We have the option to place an object as a 3D model in a real world scenario so that the shopper may get a close-up view of the item instead of the conventional photo or video format description. A major impediment with online shopping is the fear of getting scammed, getting a product that contains defects and discrepancies or not liking the scale or structure of a product in person. As the shopper needs to rely on images or video of the product shot by the seller, there is no verification for the condition of the product. However, our AR view provides aid to this barrier by presenting the product as it is, in all its forms and dimensions. Furthermore, a shopper cannot grasp the dimensions of a product, specifically products that range from phones to furniture. AR view, through computational expertise, solves this problem, by providing a proper scale for the dimensions of the product.

2. LITERATURE REVIEW

Augmented Reality (AR) is a fast expanding field with applications ranging from gaming and entertainment to education, healthcare, and other fields. The purpose of this literature review is to provide an overview of significant works on AR, highlighting their contributions, techniques, and findings. Azuma, Ronald T's "A Survey of Augmented Reality" provides a comprehensive overview of the early advancements in AR, examining the obstacles, essential technology, and future applications [1]. It describes the core components of augmented reality systems and gives information on tracking, display devices, and user interface design.

The paper by Hollerer, Tobias gives an overview of markerless AR techniques, with a focus on spatial interaction in AR settings [2]. It discusses a variety of tracking strategies, including SLAM-based systems, 3D reconstruction, and sensor fusion techniques. The authors cover the benefits, limitations, and prospective applications of markerless AR systems, whereas "Handheld Augmented Reality" by Billinghurst, Mark, and Kato, Hirokazu delves into handheld AR, discussing the problems and design considerations for mobile AR experiences. The authors describe a handheld AR framework that addresses concerns such as registration, tracking, user interfaces, and interaction strategies [3]. The article emphasizes the promise of handheld augmented reality in a variety of application domains, including gaming, tourism, and remote collaboration.

In “Augmented reality: Applications, challenges and future trends'' by Mekni, M. and Lemieux, It is hypothesized that future study could go in a number of different areas [4]. There is a need for many HMDs made expressly with AR in mind. HMDs still have a poor field of view, poor contrast, and poor resolution. The usage of HMDs and other wearable devices, such as datasets and data gloves, is restricted. All wearable technology needs to be made lighter, smaller, and more user-friendly. Researchers working on AR systems must also take into account other difficulties like response time lags and hardware or software malfunctions. Registration mistake is one drawback of AR systems. The study of occlusion detection in AR systems is ongoing.

Yovcheva, Z., Buhalis, D. and Gatzidis, C. discovered that in order to effectively support the mobile on-site requirements of tourists, current smartphone AR applications offer access [5] to location-based information pertinent to the tourists' immediate surroundings; enable access to variable content; are flexible in terms of delivering text, video, or images; and offer interactive annotations which are integrated with map-based services and additional information.

In the paper "Personalized in-store e-commerce with the promopad: an augmented reality shopping assistant. Electronic Journal for E-commerce Tools and Applications", the idea of a shopping assistant is presented, which makes use of augmented reality technology to deliver tailored advertising and in-store shopping assistance based on dynamic contextualization [6]. The PromoPad device was a step toward ubiquitous and pervasive computers in the incredibly profitable grocery shopping sector. A pleasant and inviting shopping experience that is facilitated by a tablet computer that uses augmented reality was the development aim. The technological implementation of the video see-through augmentation system was discussed in the article, as well as how this technology enables the idea of dynamic contextualization, which modifies the context to guide users' interest flows. With augmentations and reductions of the apparent visual context, augmented reality technology can enable dynamic contextualization, the real-time alteration of context.

The paper "Enhancing brick-and-mortar store shopping experience with an augmented reality shopping assistant application using personalized recommendations and explainable artificial intelligence. Journal of Research in Interactive Marketing" stated a lot of promise for improving the brick-and-mortar store shopping experience with the development of an augmented reality (AR) shopping assistant application employing personalized recommendations and explainable artificial intelligence (AI) [7]. Customers can become more engaged with the application, increase shopping efficiency, and develop trust by taking advantage of its capacity to offer personalized product recommendations and transparent decision-making insights. It is clear from user assessments and comments that the AR shopping assistant has the potential to fundamentally alter how consumers engage with real establishments. Retailers can remain competitive by adopting these technologies and providing a distinctive and personalized shopping experience that smoothly combines the ease of online shopping with the tactile sensation of in-store browsing.

3. METHODOLOGY

3.A Problem Statement

Customers are unable to physically engage with products during the traditional online purchasing experience, which leads to restricted product understanding, probable unhappiness, and greater product return rates. In addition, diminished engagement and a less memorable consumer experience are results of the absence of immersive and interactive features during the purchasing process. The development of an Android shopping app with augmented reality (AR) features that enables users to view and interact with digital 3D models of objects in their actual environment is required to address these problems. Through the use of augmented reality (AR) technology, this app will work to provide a smooth and engaging shopping experience that bridges the gap between online and offline purchasing.

Reduced engagement and probable customer discontent result from the typical online buying experience's lack of physical touch and product visualization. By allowing users to view and interact with virtual 3D models of objects in their actual surroundings, an Android shopping app with augmented reality (AR) capabilities can solve these problems. By offering immersive product visualization, customized recommendations, and enhanced decision-making, this software seeks to improve the buying experience and revolutionize how users interact with online shopping.

3.B. Proposed System

The suggested system is a shopping software for Android that includes augmented reality (AR) elements to improve the online buying experience. Customers may digitally view and interact with 3D models of products in their real-world environment via the app, which makes use of augmented reality (AR) technology. Customers can explore and choose things from a sizable catalog utilizing the app's integration with ecommerce platforms, and they can then use their mobile devices to visualize those products in their own environment. Customers may then evaluate a product's size, design, and compatibility with their surroundings before making a purchase, creating a more immersive and customized purchasing experience.

Customers can visually engage with 3D models of products in their real-world environment by integrating AR elements into the app. They can use their mobile devices to overlay virtual things into their environment, allowing them to evaluate a product's size, design, and aesthetic appeal in relation to their surroundings.

Customers can rotate, zoom, and move the virtual product models using the app's AR capability, creating a realistic and dynamic visualization experience. As a result, they have a better understanding and perspective of the products, which enables them to make wiser purchasing decisions. Customers can experiment with numerous color schemes, scrutinize minute details, and even do virtual simulations like putting virtual furniture in their homes or trying on virtual clothes.

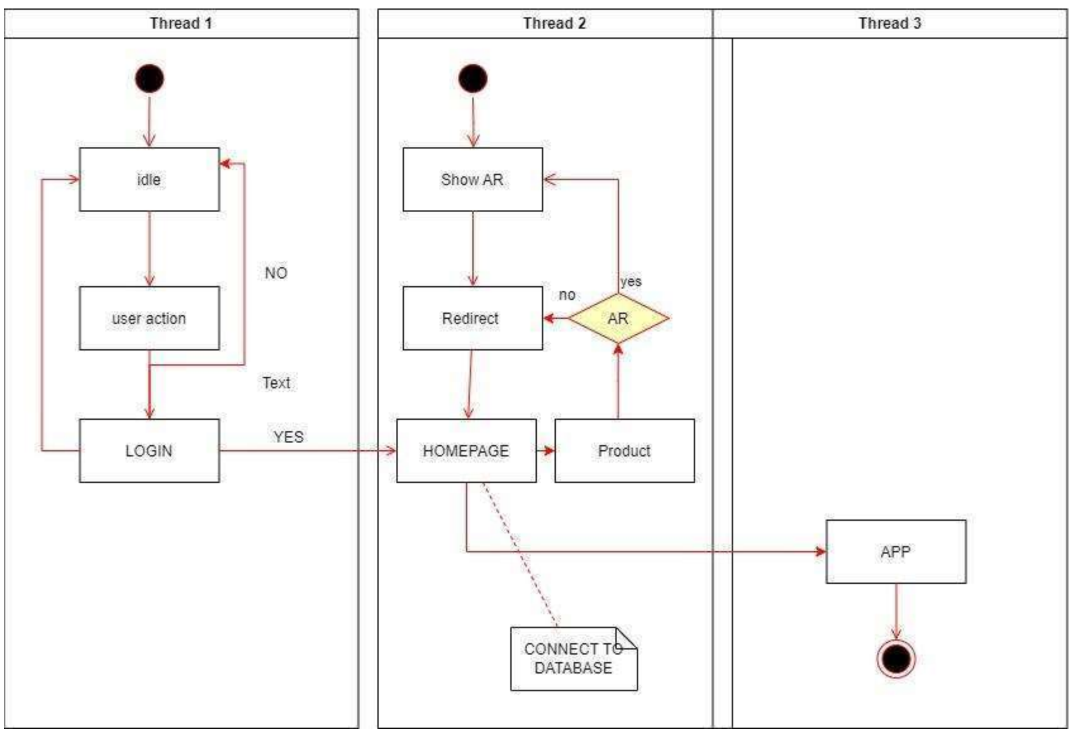

Fig. 1: UML Diagram

Risk Assessment Matrix

LIKELIHOOD \ IMPACT	ACCEPTABLE	TOLERABLE	GENERALLY UNACCEPTABLE
NOT LIKELY	PROJECT MANAGEMENT	TECHNICAL CHALLENGES	QUALITY ASSURANCE
POSSIBLE	UI/UX DESIGN	USER ADOPTION	AR
PROBABLE	DATA PRIVACY	INTEGRATION/COMPABILITY	LOGIN

Fig 2: Risk Assessment Matrix

The Risk Analysis Matrix for the Android AR Shopping App lists a number of risks that are grouped according to their likelihood and potential impact. Data protection, UX/UI design, project management difficulties, technological difficulties, user acceptance, integration/compatibility problems, quality assurance, and reliance on integration are some of these hazards. While some risks, like data privacy and UX/UI design, have a manageable impact and low likelihood, others, like project management and technological issues, may have a medium impact and high possibility. Risks with a high impact and likelihood include those related to quality control and reliance on third-party services. The development team can adopt appropriate mitigation methods and manage resources efficiently to ensure the success of the app development process by identifying and comprehending these risks. To handle these risks, assure the Android AR Shopping App's seamless performance, user satisfaction, and overall success, regular monitoring and proactive risk management are crucial. The login mechanism risk stands out as the most serious due to the potential consequences it may have. Any vulnerability or breach in the system could have significant repercussions, including unauthorized access, privacy breaches, data theft, and potential account misuse, as the login procedure requires processing sensitive user information. Such security flaws expose the app and its creators to legal and regulatory responsibilities in addition to directly endangering user confidence and privacy.

3.C. Augmented Reality

Our vision and interaction with the physical world are improved by augmented reality (AR), a cutting-edge technology that seamlessly merges digital data and virtual objects with the actual world. This is achieved by superimposing computer-generated content (CGC) over our actual environment, resulting in an immersive and participatory experience. Since AR enriches the real world by incorporating virtual aspects, it differs from Virtual Reality (VR) in that it does not entirely replace it.

A combination of hardware and software components are used to operate AR. The hardware often consists of gadgets with cameras, sensors, and displays, such as smartphones, tablets, or smart glasses. These gadgets record the actual surroundings, identify things, and follow their motions. To process the data that has been taken, identify surfaces or objects, and precisely align the virtual material with the real environment, the software component makes use of complex algorithms. With the help of this synchronization, the virtual objects can convincingly pass for real-world items in the environment. A dynamic and immersive experience is made possible by the user's ability to interact with the virtual objects through gestures, touch, voice commands, or other input techniques.

AR has found uses in a wide range of fields, including gaming, education, healthcare, architecture, manufacturing, and retail. AR allows players to play interactive games that incorporate virtual characters and objects into their real-world settings. AR in education allows for interactive learning by superimposing instructional content, simulations, or 3D models onto textbooks or actual items. In healthcare, augmented reality (AR) aids surgeons by offering real-time advice during surgery, and also aids medical training through virtual patient simulations. AR helps architects to visualize and manipulate virtual architectural models in real-world settings. Retailers can utilize augmented reality (AR) to improve the shopping experience by allowing customers to visually try on items or visualize furniture in their homes.

While augmented reality has made considerable progress, there are still issues that must be addressed. These include the creation of more compact and wearable hardware, the protection of privacy and ethics, the reduction of integration complexity, and the establishment of standardization and interoperability across various AR platforms.

Nonetheless, AR has enormous potential to transform different industries and redefine human-computer interaction. AR has the promise of new advancements and uses as technology evolves, making it an interesting topic of study and research with far-reaching ramifications for the future.

3.C. i. Object Detection and Tracking

Object detection and tracking are key components of Augmented Reality (AR) technology, allowing for the identification and location of real-world objects in a given context. These procedures are critical for precisely overlaying virtual content onto the physical world and ensuring seamless integration.

This is accomplished by utilizing computer vision algorithms that analyze visual data captured by the device's cameras. To assess the existence and identity of specified objects, these algorithms compare collected pictures or video frames to predefined patterns or reference images. To improve the accuracy and robustness of object recognition in AR applications, advanced approaches like feature extraction and matching, machine learning, and deep neural networks are frequently used.

For object detection and tracking, machine learning techniques such as Support Vector Machines (SVM), Random Forests, and Convolutional Neural Networks (CNN) are used. These algorithms learn to recognize and classify objects based on their visual properties using a labeled training dataset. CNNs, in particular, have demonstrated tremendous effectiveness in object identification tasks and are widely used in a variety of AR applications.

Object tracking, on the other hand, requires constant real-time monitoring of the position and movement of recognized objects. After identifying an object, the AR system employs several tracking algorithms to follow its spatial position and orientation as the camera or the object moves. This is critical for keeping the virtual content and the real-world object aligned and ensuring that the overlay remains constant and cohesive.

Marker-based tracking, in which specific markers or fiducial markers are placed on objects to provide reference points, and markerless tracking, in which visual features and motion estimation are used to track things without markers, are two common tracking approaches.

3.C. ii. Scene Rendering and Positioning

Scene rendering and positioning are critical components of Augmented Reality (AR) technology because they entail properly placing virtual objects in the real-world environment and rendering them realistically to provide a smooth merger of the virtual and physical worlds.

The act of producing and displaying virtual objects or content onto the real-world view acquired by the device's camera is referred to as scene rendering in AR. This necessitates the use of powerful computer graphics techniques to depict the virtual objects with accurate lighting, shading, and perspective, mirroring the visual features of the real world. Real-time rendering is required to maintain the illusion of virtual and real things coexisting in the scene. To do this, AR systems use rendering techniques and optimizations that take into consideration the device's processing capabilities and limits, resulting in seamless and interactive AR experiences.

Positioning virtual objects accurately in the real world is accomplished through a combination of tracking and spatial mapping techniques. The AR system decides the viewpoint from which the virtual objects should be rendered by tracking the camera's location and orientation in real-time.

Spatial mapping techniques are used to build a computer representation of the physical world, including surfaces, objects, and their spatial relationships. This mapping assists AR systems in comprehending physical space and allows virtual objects to interact accurately with their environment. AR systems ensure that virtual material seems anchored and integrated into the real environment by aligning virtual items with the tracked camera and employing spatial mapping data.

Scene rendering and placement are critical in AR applications because they allow users to perceive and interact with virtual things as if they were real. Continuous advances in rendering techniques, spatial mapping algorithms, and hardware capabilities are paving the way for more realistic and immersive augmented reality experiences in a variety of industries ranging from gaming and entertainment to education, healthcare, and beyond.

AR systems use rendering algorithms that are commonly used in computer graphics, such as rasterization or ray tracing. Rasterization is a popular approach for converting 3D geometric data into 2D graphics by projecting vertices onto the screen and filling in the pixels in between. To build realistic and visually appealing virtual objects, this procedure employs techniques such as vertex transformation, clipping, and shading. Ray tracing, on the other hand, precisely calculates lighting, shadows, reflections, and refractions by simulating the behavior of light rays in a virtual environment. This technique generates highly realistic renderings but necessitates a large amount of computer power.

Spatial mapping techniques generate a digital representation of the physical world. These algorithms recreate the physical space in 3D using sensor data such as depth maps from cameras or point clouds from LiDAR. Volumetric reconstruction, point cloud registration, and mesh generation are some of the techniques used to build a detailed and accurate depiction of surfaces, objects, and their spatial relationships. Spatial mapping is critical for precisely locating virtual items and ensuring their interaction with the physical world.

3.C. iii. LiDAR

Lidar, an abbreviation for Light Detection and Ranging, is a remote sensing technique that employs laser light to detect distances and build highly accurate 3D maps of the surroundings. Lidar devices fire laser pulses and time how long it takes for the light to reflect back after colliding with objects in its path. Lidar systems can determine distances to various objects by precisely measuring the round-trip duration of laser pulses, resulting in a detailed point cloud representation of the scene [8]. Cloud data can be used in a variety of applications, such as autonomous cars, terrain mapping, urban planning, forestry, and others, where precise and high-resolution spatial information is required for reliable analysis and decision-making.

3.C.iv. ARCore

ARCore is a platform developed by Google that enables us, as developers, to create immersive and interactive augmented reality (AR) experiences on Android devices. The problem statement and specific objectives for integrating ARCore into the Android shopping app are identified in the first phase of the process.

The statement of the problem focuses on the drawbacks of conventional internet buying methods, such as the absence of actual product engagement and visualization. To get over these restrictions and give customers an immersive and engaging shopping experience, ARCore technology will be used.

An extensive analysis of the requirements is carried out in this phase to identify the precise use cases and functionalities where ARCore will be included. Identifying the supported devices, required operating system versions, and hardware specifications for ARCore compatibility are all part of the requirements collecting process. In addition, the intended AR capabilities and interactions, such as object placement and 3D product viewing, are specified.

This entails adding to the project the relevant parameters and dependencies. For AR capabilities, such as motion tracking, environmental awareness, and light estimates, the ARCore SDK offers crucial features and APIs. The SDK gives the app the ability to smoothly overlay virtual items, detect real-world surfaces, and monitor the location and orientation of the device with accuracy.

To integrate ARCore into our Android Studio project, we need to add the ARCore SDK as a dependency in the project's build.gradle file. By implementing the ARCore features, such as motion tracking, environmental understanding, and light estimation, using the ARCore APIs provided, we can leverage the camera feed and sensor data to track the position and orientation of the device, detect real-world surfaces, and seamlessly overlay virtual objects. It's important to test the AR features on compatible devices and iterate as needed to optimize the AR experience for our users. With ARCore, we can unlock the potential of AR technology and create captivating experiences within our Android applications.

3.D. E-Authentication

The process of electronically validating the identity of individuals or entities in digital transactions or interactions is referred to as e-authentication. It entails employing numerous security procedures and approaches to ensure that the user's or entity's stated identification is valid and dependable. In today's digital landscape, e-authentication is critical for establishing trust, protecting sensitive information, preventing fraud, and enabling safe online transactions.

To create and verify identification, the e-authentication procedure often involves numerous criteria. These elements can include something the user knows (like a password or PIN), something the user has (like a smart card or token), or something the user has (like biometric features like fingerprint or facial recognition). A multi-factor authentication approach improves the security and reliability of the authentication process by combining several factors. To protect the transmission and storage of user credentials and personal information, e-authentication systems may additionally use encryption, digital certificates, secure protocols, and other security techniques.

E-authentication is used in a variety of industries, including e-commerce, online banking, government services, healthcare, and others. It allows for secure access to digital resources, the protection of sensitive data, and the assurance that only authorized users can access and interact with digital systems. As the digital ecosystem evolves, e-authentication technologies and processes change to solve new security challenges and deliver robust identity verification solutions.

3.D.i. OAuth 2.0

OAuth 2.0 is an open standard protocol that enables users to offer limited access to protected resources on one website (referred to as the "resource server") to another website or application (referred to as the "client") without exposing their credentials, such as usernames and passwords. It provides a secure and standardized framework for access rights authorization and delegation in distributed systems. The resource owner (the user who owns the protected resources), the client (the application or website that wants access to the resources), and the authorization server (the server responsible for authenticating the resource owner and issuing access tokens) interact in a series of interactions.

The procedure begins when the client redirects the resource owner to the authorization server, where they authenticate themselves [9]. After authentication, the resource owner provides the client access to their protected resources. The authorization server then provides the client with an access token. The client can use this access token to acquire access to the requested resources on behalf of the resource owner. The resource server validates the access token and, if it is genuine, presents the client with the requested resources.

Google has widely implemented and employs OAuth 2.0. It provides a standardized and safe method for third-party programs to access user data without requiring sensitive credentials to be shared, hence improving user privacy and security.

3.D.ii. Firebase

Google Firebase is a complete mobile and online application development platform. Firebase provides real-time database, cloud storage, authentication, hosting, cloud functions, and other services that may be easily integrated into applications using SDKs and APIs. Developers can use Firebase to create scalable and dependable applications, handle user authentication and data storage, send notifications, and analyze user behavior using built-in analytics.

Firebase's user-friendly UI, thorough documentation, and strong community support make it a popular choice for developers wishing to expedite their development process while leveraging Google's solid infrastructure.

Google uses Firebase for a variety of reasons. For starters, Firebase supports Google's objective of organizing the world's information and making it universally accessible. Google helps developers worldwide to create creative and accessible applications that leverage Google's infrastructure and services by providing a complete development platform. Furthermore, Firebase connects smoothly with other Google services such as Google Cloud Platform, Google Analytics, and Google Ads, giving developers a cohesive environment for building, analyzing, and monetizing their apps [12]. The scalability, dependability, and real-time capabilities of Firebase make it an excellent choice for Google to support and empower developers, further growing Google's footprint in the mobile and online application development market.

3.E. Software Design

3.E.i. Kotlin

JetBrains created Kotlin, a new programming language that combines the finest characteristics of object-oriented and functional programming. Kotlin, which was designed to be succinct, expressive, and interoperable with current Java code, has grown in popularity, particularly in the field of Android app development.

In comparison to Java, Kotlin has a more streamlined and intuitive syntax, which reduces boilerplate code and improves code readability. It has sophisticated language features like null safety, extension functions, smart casts, and coroutines that boost developer productivity and code resilience. Kotlin's easy integration with the Java Virtual Machine (JVM) enables developers to leverage existing Java libraries and frameworks, making it an adaptable choice for Android development.

Kotlin is completely compatible with Java. Kotlin and Java can coexist in the same project, easing migration and allowing developers to take advantage of Kotlin's current capabilities without sacrificing compatibility. Second, as compared to Java, Kotlin's compact syntax and language features dramatically minimize the amount of code required to achieve the same functionality.

This results in higher developer productivity, improved code maintainability, and less opportunities for error introduction. Furthermore, by providing type safety and enforcing non-nullability by default, Kotlin's null safety feature aids in the elimination of null pointer exceptions, a typical cause of defects in Java. This feature improves the stability and dependability of Android apps, lowering crashes and enhancing user experience [10].

Kotlin is intended to be fully compatible with Java, which implies that Kotlin code can interact with existing Java codebases effortlessly. Kotlin compiles into Java bytecode, allowing it to run on the Java Virtual Machine (JVM) and take advantage of Java's enormous ecosystem of libraries, frameworks, and tools. Kotlin can call Java code directly without any further preparation or conversion.

Java classes and methods are treated as first-class citizens in Kotlin, allowing you to import and utilize them in Kotlin code. When working with Java classes, Kotlin provides increased syntax and capabilities such as null safety and smart casts that can be used to make the code more resilient and succinct. Kotlin's nullable types can be used with Java's optional types, and Kotlin's collections can be used with Java's collections seamlessly.

Because of its more streamlined and expressive syntax than Java, Kotlin is a popular choice for Android development. It removes boilerplate code, resulting in more compact and readable code. Developers may accomplish the same functionality with fewer lines of code, increasing productivity and shortening development cycles.

By offering null safety, Kotlin tackles one of Java programming's primary pain points. The type system in Kotlin distinguishes between nullable and non-nullable types, which reduces the occurrence of null pointer exceptions, which are a common cause of crashes in Java projects. This feature improves the stability and dependability of Android apps, resulting in fewer bugs.

Google officially embraced Kotlin as an Android programming language in 2017. This Google support improved Kotlin's popularity among Android developers tremendously. Google actively promotes Kotlin, providing official documentation and tools, and encouraging developers to utilize Kotlin in their Android projects [11].

3.E.ii. SDK

Android Studio is the official Integrated Development Environment (IDE) for developing Android apps. It includes a full set of tools and capabilities developed exclusively for developing Android applications. Android Studio is a rich and user-friendly interface that allows developers to write, debug, and test code quickly. It includes a variety of development tools, including as layout editors, code analyzers, emulators, and profilers, to boost productivity and make the development process easier. Android Studio also works in tandem with other Android development tools and libraries, creating a unified environment for creating high-quality Android apps.

For a variety of reasons, Android Studio is regarded as the superior choice for Android app development. To begin with, it provides exceptional support and interoperability with the

Android platform. It includes a powerful emulator for testing programs on multiple Android device setups, as well as performance analysis and debugging tools.

Furthermore, Android Studio connects with the Android SDK and supports the most recent Android APIs, ensuring compatibility and access to the most recent features and advancements. Furthermore, Android Studio's support for the Gradle build system streamlines the build and dependency management processes, enabling rapid and adaptable app development.

Google's Android SDK (Software Development Kit) provides a collection of tools, frameworks, and resources for creating Android applications. It includes the Android API libraries, build tools, emulator, debugger, and documentation needed for developing, testing, and deploying Android apps. The Android SDK gives developers access to the Android platform's features and capabilities, as well as the tools they need to create and optimize Android applications.

We will often use the Android SDK in conjunction with Kotlin while developing Android apps. Kotlin is fully compatible with and integrates with the Android SDK. Kotlin includes language features, libraries, and tools that aid with the creation of Android apps, such as Kotlin Android Extensions and Kotlin Coroutines. When developing Android apps using Kotlin, these features enable you to write clear and efficient code.

Android Studio fully supports Kotlin, the recommended programming language for Android development. Kotlin may be smoothly incorporated into Android Studio projects by either creating new Kotlin projects or adding Kotlin files to existing Java ones. Kotlin-specific capabilities, such as syntax highlighting, code completion, and refactoring support, are provided by Android Studio, making Kotlin development in Android Studio smooth and quick. Because Kotlin and Java are interoperable, developers may work with existing Java codebases while gradually adopting Kotlin and leveraging the benefits of both languages within the same project. The robust support for Kotlin in Android Studio allows developers to leverage the power and productivity of Kotlin in their Android app development process.

3.F. Gathering The Requirements / Planning

We started by defining our AR shopping app's objectives and needs, specifying the major features and functions the app should have, such as product browsing, AR product visualization, user identification, shopping cart, and (future) payment integration. After gathering the requirements, the following stage was to establish a strong development strategy. During this stage, it was critical to consider any problems and risks that could arise during development, such as AR implementation complexities, compatibility issues with various Android devices, or security concerns linked to user authentication.

A precise development roadmap was essential. Breaking down the project into discrete development tasks allows for time to be allocated for UI design, AR functionality implementation, Firebase integration, user authentication, shopping cart development, and testing. The development plan aids in creating clear goals, effectively managing resources, and ensuring that all components of the app were built methodically.The app included several components.

The prominent of these was the User Interface. The app's user interface should be well-designed to allow for easy navigation and interaction. Product listings, search functionality, product details view, shopping cart, and checkout screens are examples of UI elements. The user interface should be simple, visually appealing, and suited for both AR and regular touch-based interactions.

Because we were working with Android, the app required us to include an AR framework such as ARCore to enable AR product visualization. These frameworks provide APIs and tools for detecting surfaces, locating virtual items in the real world, and rendering augmented reality scenes. Using the device's camera and sensors, users may view and interact with virtual products in their physical environment.

Firebase offers various critical app components, including Firebase Authentication for user registration, login, and authentication. It enabled users to create accounts, sign in, and access personalized features in a secure manner. Product data, such as product photos, descriptions, and pricing, can be stored and retrieved using Firebase Realtime Database or Firestore. Firebase Cloud Functions or Firebase Cloud Messaging can also be used to provide push notifications about order progress or promotional changes.

The software required to offer shopping cart management functionality, allowing users to add products, adjust quantities, and remove items. The data from the shopping cart can be kept locally on the device or synced with a backend server through Firebase Realtime Database or Firestore.

User authentication is critical for ensuring safe access and personalized experiences. User registration, login, password recovery, and account management may all be handled via Firebase Authentication. Features such as password reset, email verification, and profile management are included.

The app required a system for managing and displaying product data, such as photos, descriptions, prices, and availability. This information can be saved in a backend database, such as Firebase Realtime Database or Firestore, and retrieved dynamically as users navigate the app. To efficiently handle product data and provide fast retrieval and changes, proper data modeling and database structure should be adopted. For development, we used Android Studio and the Kotlin programming language.

3. G. Designing the User Interface

We used Figma to develop wireframes and mockups during this phase, which allowed us to visualize the app's layout, navigation flow, and interaction patterns. The user interface should stress simplicity, consistency, and convenience of use, allowing customers to browse items, examine AR visualizations, and manage their shopping basket with ease.

To develop a seamless AR experience within the UI, it's critical to consider the special requirements for AR interactions, such as AR object positioning, gesture-based controls, and 3D rendering.

With the growth of different Android smartphones with varying screen sizes, resolutions, and aspect ratios, it was critical to guarantee that the app's user experience is responsive and compatible across a variety of devices. The UI design had to adapt to numerous screen sizes fluidly, providing consumers with a consistent and optimum experience regardless of the device they used.

The XML layout files in Kotlin allow you to specify responsive design attributes like weighted layouts, constraint-based layouts, and dimension qualifiers to ensure that UI elements are correctly positioned and presented across multiple device settings. Compatibility testing on various Android devices and screen resolutions is critical for identifying and addressing any UI rendering issues.

3. H. Augmented Reality Integration

We went with ARCore, Google's augmented reality platform. APIs and tools are provided by ARCore to identify surfaces, track camera movement, and position virtual items in the actual environment. ARCore integration entails including the required dependencies in your project's build configuration file (e.g., Gradle), implementing ARCore-specific classes and methods in your code, and handling AR-related events and interactions.

To precisely put virtual items, AR capability mainly relies on recognizing and tracking real-world surfaces. Utilize the features of the AR framework to implement surface detection and tracking. Surface detection APIs, for example, are provided by ARCore, allowing your app to identify horizontal and vertical surfaces such as floors, tables, and walls.

You can determine acceptable surfaces for object placement by analyzing the camera feed and depth information. Implement logic to display the identified surfaces in your app's AR view and provide users with visual feedback. Surface tracking guarantees that virtual objects keep their position in relation to real-world surfaces even when the camera moves.

After detecting and tracking surfaces, we began to place virtual objects in the AR environment, such as 3D models or overlays. Use the AR framework's APIs to precisely position and orient these items in 3D space. Allow users to engage with virtual items by using gesture-based interactions such as tapping or dragging. Users can, for example, tap on a virtual product to access details or drag and drop it onto a surface for display. Consider including visual signals, animations, or effects to improve the user experience and provide feedback during object placement and interaction.

3. I. Frontend Implementation

The AR feature is mostly handled on the client-side, within the app's frontend code, in a frontend implementation. This method is widespread in smartphone apps, as the AR experience is integrated directly into the user interface. The frontend code makes use of AR frameworks and libraries such as ARKit for iOS and ARCore for Android to accomplish activities such as surface detection, object placement, and AR content rendering.

The AR logic runs on the user's device, utilizing the camera and sensors to create real-time AR experiences. To gather camera input, detect elements in the environment, overlay virtual objects, and manage user interactions, the frontend code interacts with the AR framework's APIs.

To organize and show product listings, search results, and shopping cart information, we use Android layout components like ConstraintLayout, LinearLayout, and RecyclerView. In order to assure consistency and deliver a familiar experience to users, include new design ideas such as Material Design guidelines. In addition, we added custom views and animations to improve the user experience and generate compelling interactions. To promote simple access to different program areas, the user interface should contain intuitive navigation patterns such as bottom navigation or side drawers. We also used AR-specific UI components, such as on-screen instructions and object placement buttons, to help consumers through the AR purchasing experience.

Using Kotlin and the ARCore framework, we easily integrated AR capability into the UI of your project. We have to leverage ARCore APIs to detect and track surfaces in order to implement an AR view where the camera feed and AR elements are rendered, allowing users to place virtual objects on compatible surfaces in the real world. We also used intuitive movements like tap and drag to interact with AR items. To visualize the products in the user's environment, overlay virtual product overlays or 3D models on top of the camera stream. To improve the AR experience, we provided visual feedback such as highlighting recognized surfaces or providing loading indicators during item insertion.

Using Kotlin and Android style components, we created an easy-to-use and visually appealing Google Sign-In page. To commence the Google Sign-In procedure, we used buttons or custom UI components. To give a familiar and consistent user experience, we presented relevant branding components such as our app logo or bespoke branding. Our responsive design responded to various screen sizes and orientations. In the event of authentication failures or connectivity challenges, we implemented error handling techniques to display necessary error messages. We also offered clear instructions or tooltips to walk users through the Google Sign-In process, guaranteeing a smooth and user-friendly experience.

Using the Kotlin and Google Sign-In APIs, we incorporated Google Sign-In functionality into the frontend of our project. In the Google Cloud Console, we configured our app with the appropriate credentials and permissions to allow Google Sign-In. When a user clicks the Sign-In button, we launch the Google Sign-In flow and retrieve the user's authentication data, such as the user ID and email. We managed the asynchronous nature of the authentication process by leveraging Kotlin features like as coroutines and asynchronous programming techniques. We also included error handling and handled instances when authentication failed or was canceled by the user politely. We received and processed the authentication result before moving on to the next steps in the workflow of our app.

We integrated Google Sign-In functionality into our project's frontend using the Kotlin and Google Sign-In APIs. We configured our app in the Google Cloud Console with the necessary credentials and permissions to allow Google Sign-In. When a user clicks the Sign-In button, the Google Sign-In flow is launched, and the user's authentication data, such as the user ID and email address, is retrieved.

We handled the authentication process's asynchronous nature by utilizing Kotlin features such as coroutines and asynchronous programming approaches. We also incorporated error handling and handled cases where authentication failed or was canceled graciously by the user. We received and processed the authentication result before proceeding to the next steps in our app's workflow.

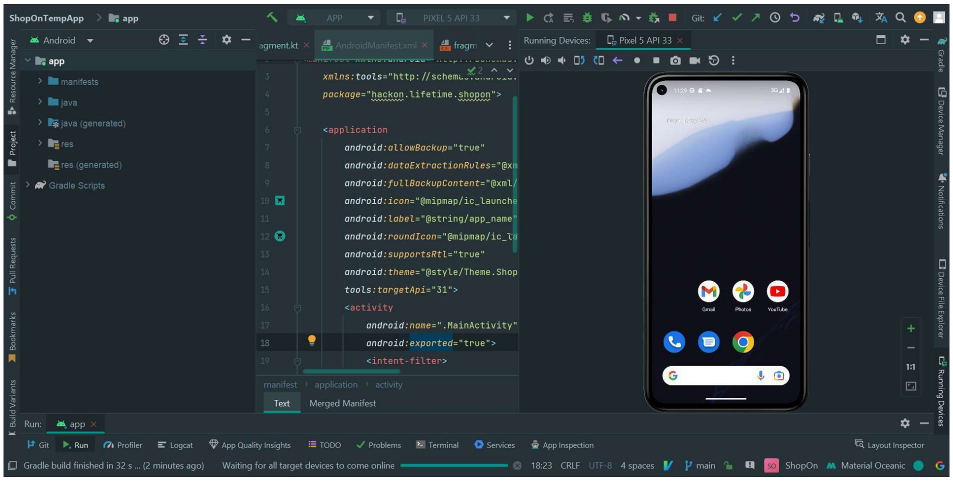

Fig. 3: User Interface XML

3. J. Backend Design

We handled the authentication process's asynchronous nature by utilizing Kotlin features such as coroutines and asynchronous programming approaches. We also incorporated error handling and handled cases where authentication failed or was canceled graciously by the user. We received and processed the authentication result before proceeding to the next steps in our app's workflow.

Using Kotlin and relevant server-side technologies, we built a secure authentication backend. To handle user registration, login, and session management, we built an authentication service or used existing frameworks. This included securely storing user credentials including usernames and passwords, as well as adopting strong encryption and hashing algorithms to protect sensitive data. We also integrated the necessary APIs and libraries to enable Google Sign-In authentication, which allows users to sign in using their Google accounts. To provide a smooth and safe AR purchasing experience, we evaluated and processed authentication requests, securely exchanged tokens with the frontend, and maintained user sessions.

We created a backend system for AR data processing in order to allow AR functionality in our app. Configuring server-side components to receive and handle AR-related data, such as product information, 3D models, and augmented reality overlays, was part of the procedure.

We created APIs or implemented relevant server-side logic to handle frontend requests and deliver the data required for displaying AR experiences. As needed, we stored and retrieved AR-related data from databases or other storage systems. Furthermore, we enhanced the backend infrastructure to accommodate the computing demands of AR processing, guaranteeing that our users have smooth and rapid AR experiences.

To precisely locate virtual objects in the actual world, camera images are matched with depth information or point cloud data. Sensor fusion techniques, such as simultaneous localization and mapping (SLAM), are frequently used to track the location and orientation of the device in relation to its surroundings.

We ensured that our app's frontend and backend components communicated and integrated seamlessly. APIs and protocols for data sharing and synchronization between client-side (frontend) and server-side (backend) components were developed.

To protect data transmission, secure communication channels were established employing encryption and authentication procedures. To reduce latency and optimize speed during data transfer, we created efficient data models and protocols. We also developed error handling and logging systems to track and troubleshoot any communication issues that arose. We built a coherent AR shopping experience where the frontend effortlessly interacts with the backend to present users with accurate and engaging AR experiences through effective integration and communication.

We chose to store the thumbnail images directly on our server's local file system. When a new product was added or updated, we generated a unique filename for the thumbnail image and saved it in a designated folder on our server. This approach gave us direct control over the storage and retrieval of the images. We ensured proper backup and maintenance of the file storage system to prevent any data loss.

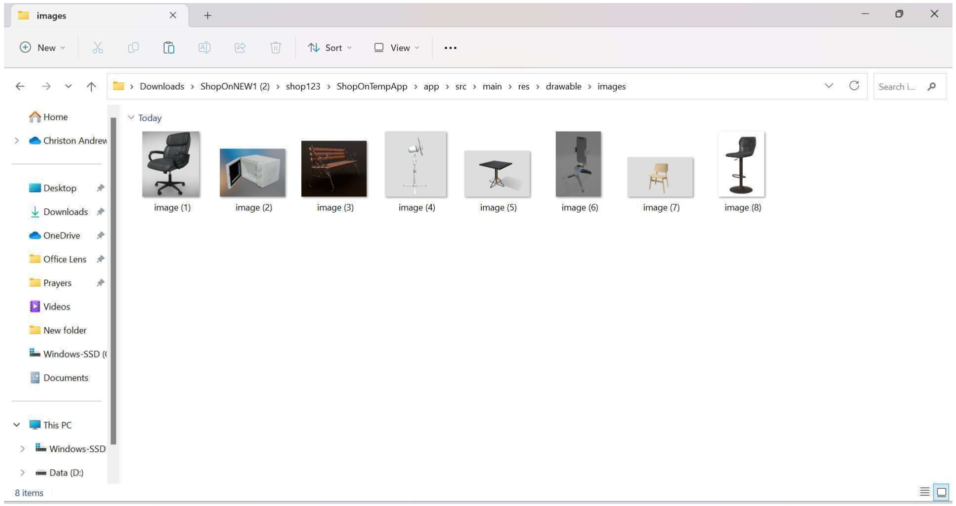

Fig. 4: Thumbnail Images

We considered storing the thumbnail images directly in our database. This involved converting the images into binary data and storing them in a BLOB (Binary Large Object) column [13]. While this approach simplified data management by keeping the images within the database, it could increase the database size and potentially impact performance if there were a large number of images or concurrent requests for image retrieval. Ultimately, we decided against this approach due to the potential drawbacks.

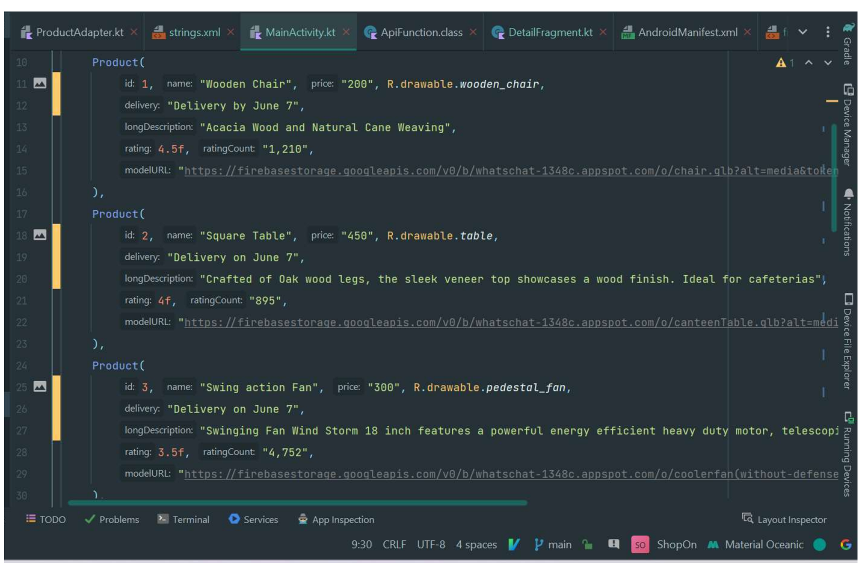

Fig 5: Homepage Code

3. K. Sign In Request

We followed a set of steps to integrate Google Sign-In into our Android app. First, we created our project on the Google Developers Console and obtained the necessary credentials for authentication, such as the client ID and client secret. The Google Sign-In API was then integrated into our app by adding the necessary dependencies to our project's build.gradle file and syncing the project. In addition, we included the SHA-1 fingerprint of our app's signing certificate in the Google Developers Console credentials.

Following that, we added a user interface element, often a button, to commence the Google Sign-In process. We handled the sign-in flow in our code when the user clicked the button. This entailed presenting the user with a consent page and requesting permission to access their Google account information, including their basic profile details and email address. Our app received a sign-in response including an ID token and an access token after the user granted consent. The ID token was used to validate the user's identity on our backend server, and the access token was used to connect with Google APIs on the user's behalf.

Furthermore, if our app had a backend server, we sent the ID token to the server for verification. Our server utilized Google's token verification API to verify the authenticity of the token and extract relevant user information. Once the user's identity was successfully authenticated, we handled their account within our app. We stored the user's information locally, associated it with their app-specific account, and provided personalized experiences based on their Google profile data. By implementing Google Sign-In, we were able to offer our users a seamless and secure authentication option, eliminating the need for them to create separate accounts and enhancing their overall app experience.

4. RESULT AND DISCUSSION

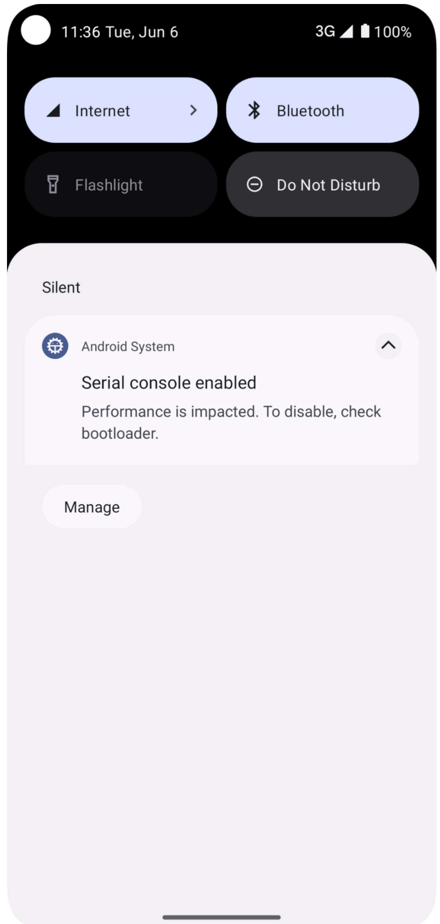

Fig 6: Android Virtual Device

By using the Pixel 5 as our Android virtual device in Android Studio, we could create a testing environment that closely resembles the functionality and behavior of actual devices. The Pixel 5 is the best option for reliably evaluating app compatibility and performance across various Android versions thanks to its cutting-edge hardware specifications and optimized software. We could guarantee a flawless user experience and solid app functionality on this well-liked and widely-used Android handset by using the Pixel 5 as our virtual device.

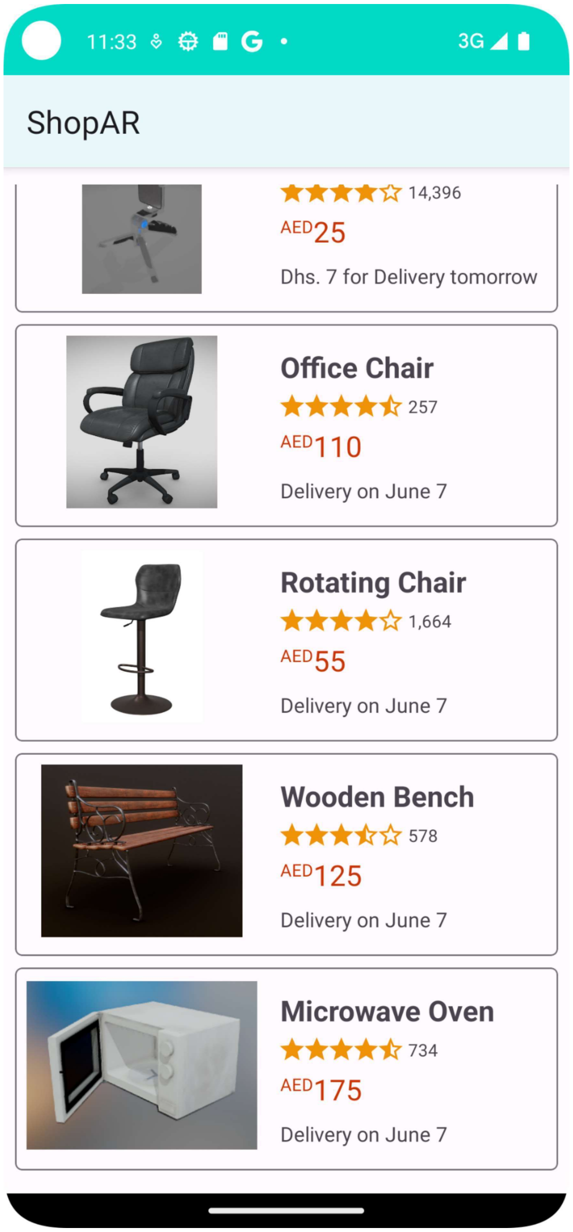

Fig 7: HomePage

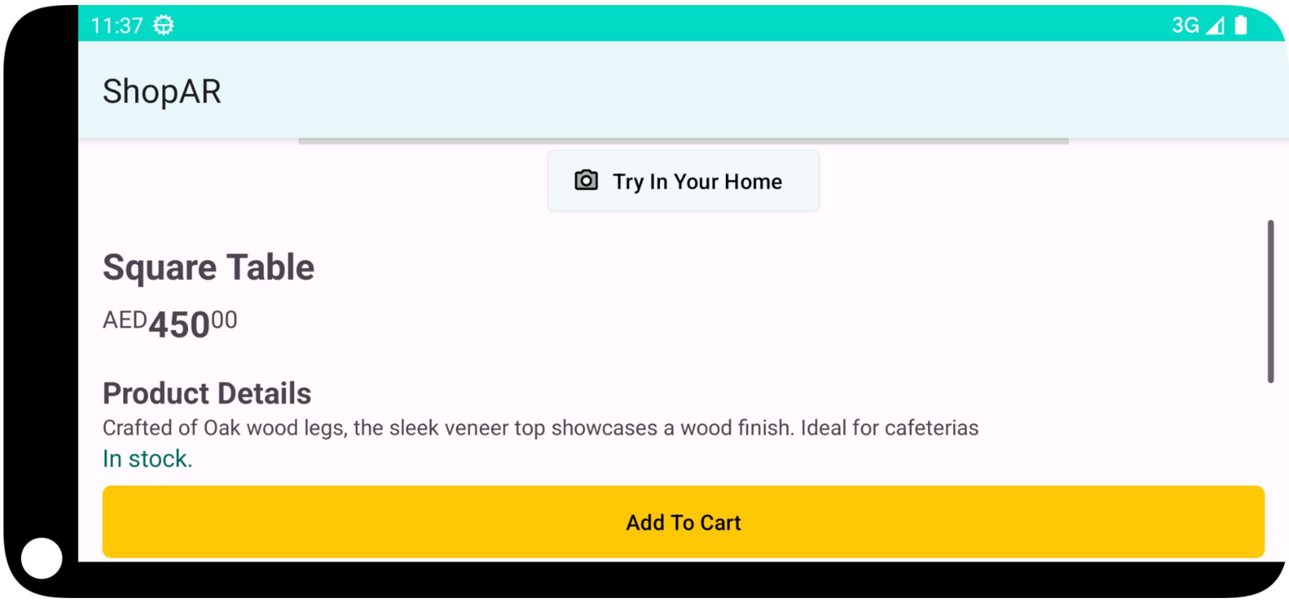

Fig 9: Landscape Orientation

In our project, we incorporated elements of the Amazon shopping app style to create a familiar and trustworthy user experience. We adopted the classic blue and white UI scheme, reminiscent of the Amazon brand, to instill a sense of familiarity and credibility among users. By leveraging the recognizable design elements, we aimed to provide a seamless transition for users and establish a sense of trust and reliability within our app.

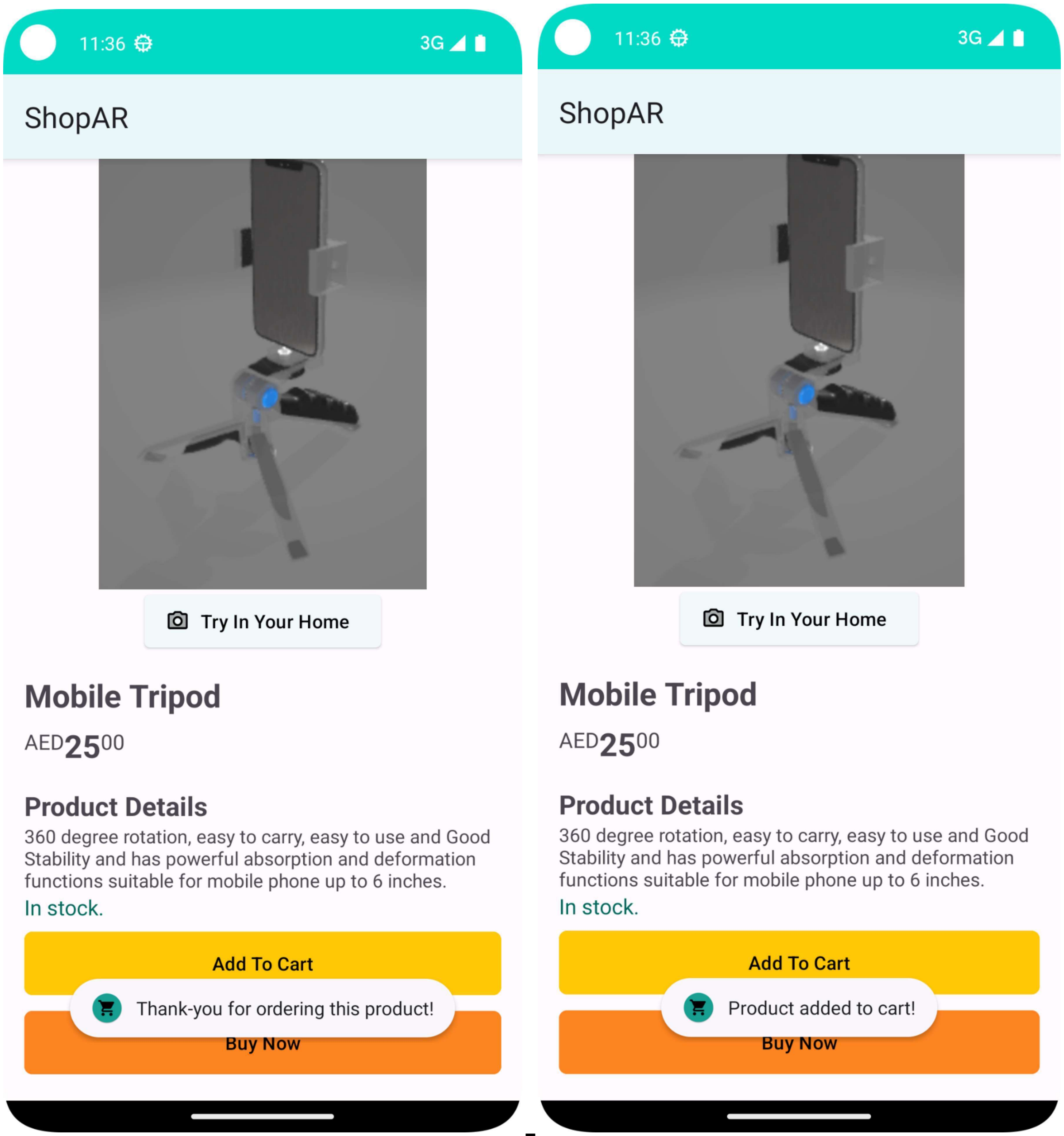

Fig 10: Buy Now Enabled/Add to Cart Feature Enabled

The “Buy Now” feature allows for instant purchase without needing to go through the hassle of adding to cart and moving forth to payment. The order is instantaneously placed. The “Add To Cart" feature is subdued and generalized with its typical function, allowing you to continue navigating the app and adding more items to your order.

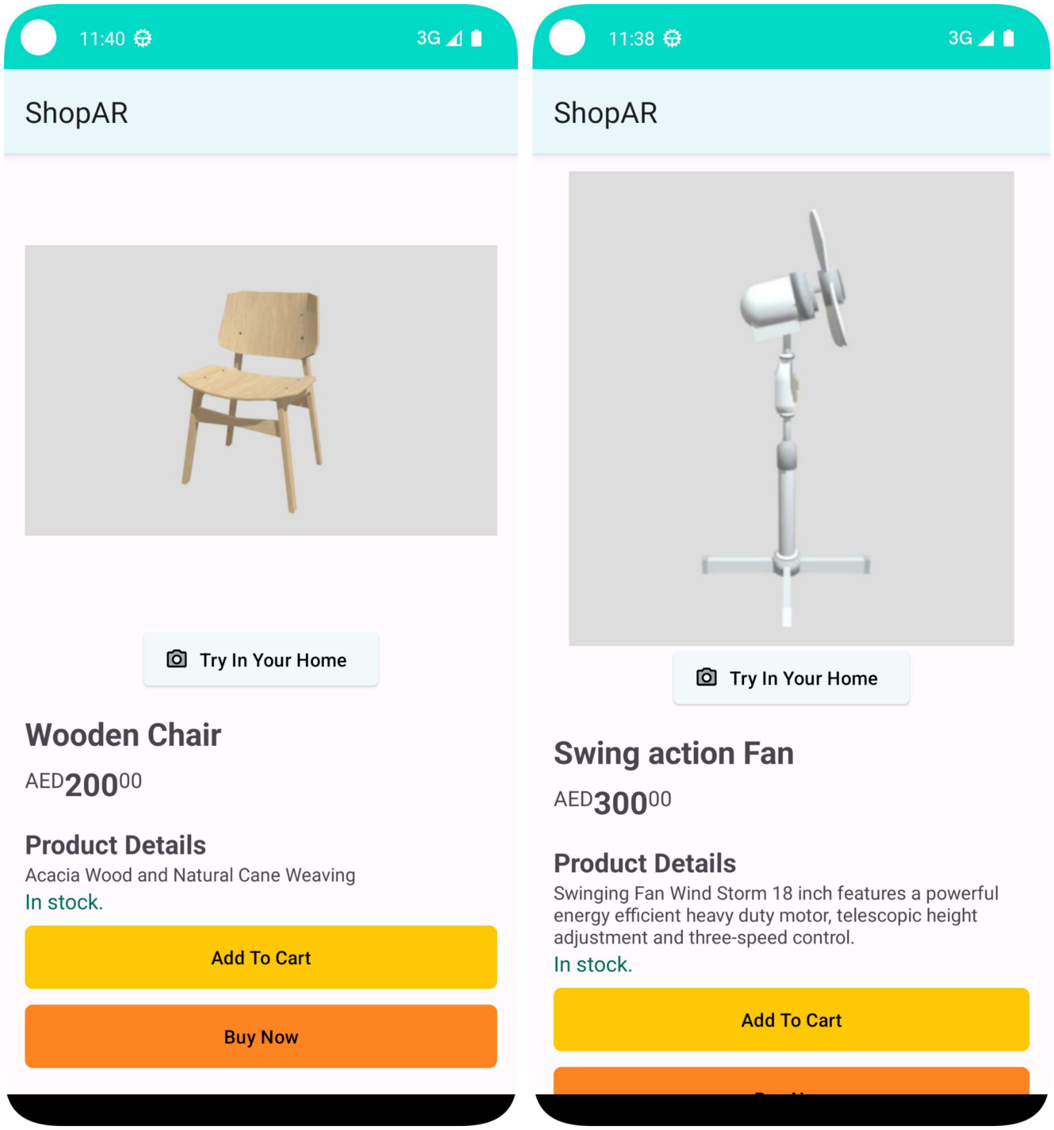

Fig 11: Product Page

The Product page contains the image of the product, followed by a short description of it. Above the name of the product, is the “Try In Your Home” button that enables the AR feature. You can see the AR enabled for these two products in the Fig 12 given below.

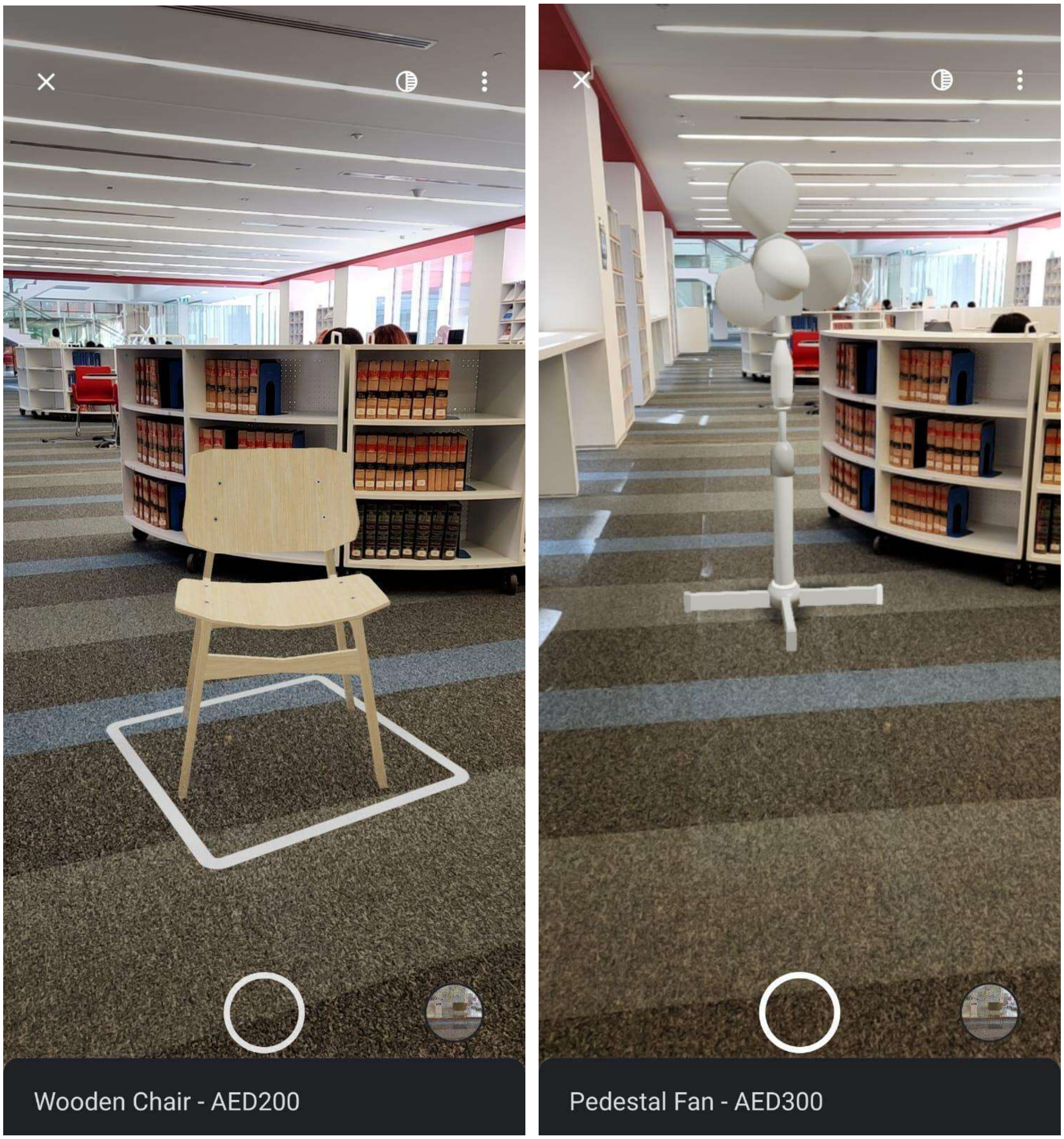

Fig 12: AR Enabled

Once the AR feature is enabled, the 3D model of the product, accurate in its scale and dimensions, is placed in the real environment using the phone's camera. A quick 5 second scan of the environment helps in locating all the elements present while the initial white box pinpoints the accurate location for locating the product. Product name and price are given below. You can adjust the color or exit the AR page using controls provided at the top right corner.

5. CONCLUSION

In conclusion, the purchasing experience has been transformed by the incorporation of Augmented Reality (AR) technology in our Android app. Users can interact with products in a personalized and immersive way by using AR, enabling wise purchasing decisions. The seamless integration of virtual goods with the physical surroundings heightens the thrill and engagement of the shopping experience. The usage of augmented reality in our app results in better decision-making, better product visualization, and more engaged users. Users may visually test out products, determine their compatibility, and comprehend their features better thanks to the power of AR. While dependable backend systems offer a dependable framework for a flawless user experience, our secure login system guarantees user privacy.

Our Android app's usage of augmented reality has revolutionized purchasing by fusing cutting-edge technology with a user-centric strategy. Our software stands out thanks to the immersive AR experience, secure login, and tailored purchasing experience it offers consumers. As we adopt new technologies, we stay constant in our goal to provide customers with a superior purchasing experience that goes above and beyond their expectations.

6. FUTURE PROSPECTS

We intend to include a personalized recommendation engine into our app. We can deliver personalized product suggestions to each user by leveraging user data such as purchase history, browsing tendencies, and preferences. The whole user experience will be improved by delivering relevant and curated recommendations, making the shopping process more efficient and entertaining.

Recognizing the importance of data security, we are thinking about adopting Two-Factor Authentication or Face Unlock as an extra layer of security throughout the sign-in process. Users must supply two types of authentication credentials, often a combination of something they know (e.g., a password) and something they have (e.g., a verification number texted to their mobile device).

We intend to take advantage of new breakthroughs and features to improve our app. Incorporating more realistic and immersive AR experiences, such as enhanced object detection, real-time interaction with virtual items, and integration with wearable AR devices, is part of this. We can give cutting-edge features to our users and solidify our position as a leader in AR-based retail apps by remaining at the forefront of AR advancements.

By making our software available to independent sellers, we hope to empower ordinary people by giving them a platform to promote and sell their wares. This adds a distinct feature to our program by fostering a thriving marketplace where users can discover a varied selection of things, from professional shops to individual sellers.

7. LIMITATIONS

Our app's AR functionality is strongly reliant on the device's hardware capabilities, including its camera and processing capability. As a result, on older or lower-end smartphones, the app may not provide the same degree of performance or AR experience. Users who have devices that do not meet the minimal system requirements may have restricted functionality or face performance issues. While augmented reality technology allows virtual objects to be placed in real-world settings, it is crucial to note that some physical limits may have an impact on the AR experience. Inadequate illumination, shiny surfaces, and cluttered backdrops can all impede accurate object recognition or tracking. To maximize the AR capabilities of the app, users may need to ensure they are in an appropriate setting.

8. INDIVIDUAL OUTCOME

My contribution to the app has been driven by an exploratory strategy as a passionate AR developer. I've excitedly experimented with several AR frameworks and tools as I've dug deeper into the field of augmented reality. Through practical exploration, I have learned a lot about the potential and constraints of AR technology. This inquisitive mentality has helped me find novel methods to employ augmented reality (AR) within the app, pushing the bounds of what is practical and opening up fresh possibilities for user engagement.

The integration of Kotlin into the AR development process has been a problem for me, despite the fact that my expertise in this area is still in its infancy. As Kotlin becomes more well-known as a flexible programming language for Android development, I realized how useful it would be to use its expressive syntax and powerful features to improve the app's augmented reality capabilities. I have improved the performance and maintainability of the AR implementation by experimenting with Kotlin and writing cleaner, more succinct code.

In order to overcome obstacles and provide engaging AR experiences, I have embraced innovative problem-solving throughout my career as an AR developer. I have applied my problem-solving abilities to come up with creative solutions for everything from addressing AR tracking issues to enhancing performance. To improve the AR capabilities of the app, I have investigated a number of methods, including introducing markerless tracking and improving object identification algorithms. I have worked hard to give consumers engaging and smooth experiences within the app by constantly improving and iterating on the AR features.

9. REFERENCES

[1] Azuma, R.T., 1997. A survey of augmented reality. Presence: teleoperators & virtual environments, 6(4), pp.355-385.

[2] Lee, T. and Hollerer, T., 2008, March. Hybrid feature tracking and user interaction for markerless augmented reality. In 2008 IEEE Virtual Reality Conference (pp. 145-152). IEEE.

[3] Billinghurst, M. and Kato, H., 2002. Collaborative augmented reality. Communications of the ACM, 45(7), pp.64-70.

[4] Mekni, M. and Lemieux, A., 2014. Augmented reality: Applications, challenges and future trends. Applied computational science, 20, pp.205-214.

[5] Yovcheva, Z., Buhalis, D. and Gatzidis, C., 2012. Smartphone augmented reality applications for tourism. E-review of tourism research (ertr), 10(2), pp.63-66.

[6] Zhu, W., Owen, C.B., Li, H. and Lee, J.H., 2004. Personalized in-store e-commerce with the promopad: an augmented reality shopping assistant. Electronic Journal for E-commerce Tools and Applications, 1(3), pp.1-19.

[7] Zimmermann, R., Mora, D., Cirqueira, D., Helfert, M., Bezbradica, M., Werth, D., Weitzl, W.J., Riedl, R. and Auinger, A., 2023. Enhancing brick-and-mortar store shopping experience with an augmented reality shopping assistant application using personalized recommendations and explainable artificial intelligence. Journal of Research in Interactive Marketing, 17(2), pp.273-298.

[8] Wang, Z. and Menenti, M., 2021. Challenges and opportunities in Lidar remote sensing. Frontiers in Remote Sensing, 2, p.641723.

[9] Argyriou, M., Dragoni, N. and Spognardi, A., 2017. Security flows in OAuth 2.0 framework: a case study. In Computer Safety, Reliability, and Security: SAFECOMP 2017 Workshops, ASSURE, DECSoS, SASSUR, TELERISE, and TIPS, Trento, Italy, September 12, 2017, Proceedings 36 (pp. 396-406). Springer International Publishing.

[10] Billinghurst, M. and Kato, H., 2002. Collaborative augmented reality. Communications of the ACM, 45(7), pp.64-70.

[11] Petrović, V. and Popović-Božović, J., 2016. Towards Real-Time Blob Detection in Large Images with Reduced Memory Cost. threshold, 255, p.0.

[12] Khawas, C. and Shah, P., 2018. Application of firebase in android app development-a study. International Journal of Computer Applications, 179(46), pp.49-53.

[13] Wang, L. ed., 2005. Support vector machines: theory and applications (Vol. 177). Springer Science & Business Media.

[14] Yamashita, R., Nishio, M., Do, R.K.G. and Togashi, K., 2018. Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9, pp.611-629.

[15] Klein, G. and Murray, D., 2007, November. Parallel tracking and mapping for small AR workspaces. In 2007 6th IEEE and ACM international symposium on mixed and augmented reality (pp. 225-234). IEEE.

Report

ORIGINALITY REPORT

9%	5%	5%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dokumen.pub Internet Source	1%
2	www.emerald.com Internet Source	1%
3	Wei Zhu, Charles B. Owen. "Design of the PromoPad", Journal of Organizational and End User Computing, 2008 Publication	<1%
4	Submitted to CSU, San Jose State University Student Paper	<1%
5	Submitted to University of Greenwich Student Paper	<1%
6	Submitted to Liberty University Student Paper	<1%
7	Submitted to Institute For Tourism Studies Student Paper	<1%
8	patents.google.com Internet Source	<1%

Submitted to University of Aberdeen

9	Student Paper	<1 %
10	Prabath Siriwardena. "Advanced API Security", Springer Science and Business Media LLC, 2020 Publication	<1 %
11	docshare.tips Internet Source	<1 %
12	Submitted to Texas A&M University, College Station Student Paper	<1 %
13	lib.iitta.gov.ua Internet Source	<1 %
14	Submitted to TED Ankara College Student Paper	<1 %
15	Submitted to University of Central Lancashire Student Paper	<1 %
16	Submitted to Florida International University Student Paper	<1 %
17	Submitted to HELP UNIVERSITY Student Paper	<1 %
18	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1 %

19	Submitted to Waterford Institute of Technology Student Paper	<1 %
20	eprints.utar.edu.my Internet Source	<1 %
21	Submitted to Georgia Gwinnett College Student Paper	<1 %
22	aisel.aisnet.org Internet Source	<1 %
23	link.springer.com Internet Source	<1 %
24	medium.com Internet Source	<1 %
25	newindua.blogspot.com Internet Source	<1 %
26	Cheol-Joo Chae, Kwang-Nam Choi, Kiseok Choi, Yong-Hee Yae, YounJu Shin. "The Extended Authentication Protocol using E-mail Authentication in OAuth 2.0 Protocol for Secure Granting of User Access", Journal of Internet Computing and Services, 2015 Publication	<1 %
27	www.tib.eu Internet Source	<1 %
28	core.ac.uk Internet Source	

<1%

29	kipdf.com Internet Source	<1%
30	sciences.uodiyala.edu.iq Internet Source	<1%
31	Ali Samini, Karljohan Lundin Palmerius, Patric Ljung. "A Review of Current, Complete Augmented Reality Solutions", 2021 International Conference on Cyberworlds (CW), 2021 Publication	<1%
32	Hui-Fei Lin, Hsin-yi Sandy Tsai, Benjamin Yeo. "Augmented Reality Advertising in Entertainment Programming: An Exploration Across Cultures", Journal of Creative Communications, 2023 Publication	<1%
33	New Frontiers of Educational Research, 2013. Publication	<1%
34	Submitted to University of Birmingham Student Paper	<1%
35	Submitted to University of Northumbria at Newcastle Student Paper	<1%
36	deepblue.lib.umich.edu	

	Internet Source	<1 %
37	hrcak.srce.hr Internet Source	<1 %
38	ir.nmu.org.ua Internet Source	<1 %
39	jwcn-eurasipjournals.springeropen.com Internet Source	<1 %
40	publikationen.sulb.uni-saarland.de Internet Source	<1 %
41	scholar.archive.org Internet Source	<1 %
42	www.cse.ust.hk Internet Source	<1 %
43	www.researchgate.net Internet Source	<1 %
44	www.scitepress.org Internet Source	<1 %
45	www.tandfonline.com Internet Source	<1 %
46	Lecture Notes in Computer Science, 2014. Publication	<1 %
47	Elena Marchiori, Lorenzo Cantoni. "Including Augmented Reality in Tourism Education	<1 %

Programs", Emerald, 2015
Publication

48	www.dropbox.com Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		