BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY)
COLLEGE OF ENGINEERING, PUNE
2024-2025

DEPARTMENT OF COMPUTER ENGINEERING
PROJECT BASED LEARNING
ON
TOPIC NAME: - "**ONLINE SHOPPING APP**"

NAME: - Abhijeet Panda	**PRN: -**2214110174	**ROLL NO.: -**40
NAME: - Akshat Kumar	**PRN: -**2214110124	**ROLL NO.: -**01
NAME: - Shlok Shukla	**PRN: -**2214110190	**ROLL NO.: -**41
NAME: - Siddhi Chaudhari	**PRN: -**2214110136	**ROLL NO.: -**11
NAME: - Priyanshi Bhootda	**PRN: -**2214110134	**ROLL NO.: -**09

Department: -COMPUTER ENGINEERING

Under the Guidance of: - **Prof. Gauri Rao**

Date:-
Place: Pune

(Prof Gauri Rao)
Project Guide

(Prof Gauri Rao)
Head of Department

CERTIFICATE

This is to certify that the under-Project report titled **"ONLINE SHOPPING APP"** **submitted by-**

Abhijeet Panda	**2214110174**
Akshat Kumar	**2214110124**
Shlok Shukla	**2214110190**
Siddhi Chaudhari	**2214110136**
Priyanshi Bhootda	**2214110134**

Under the guidance and supervision in partial fulfilment of requirement for the degree of "Bachelor of technology in Information Technology ", Bharati Vidyapeeth (Deemed to be) University College of Engineering Pune during the academic year 2024-2025

Date:

Semester: VI

Place: Pune

ACKNOWLEDGEMENT

We would like to express deepest appreciation towards the Principal, Bharati Vidyapeeth (Deemed to Be University) College of Engineering, Pune, India and Prof. Gauri Rao Head of the Department (CE), who invaluable supported us in completing this project.

We are profoundly grateful to Prof. Gauri Rao, Project guide for her expert guidance and continuous encouragement throughout to see that this project rights, its target since its commencement to its completion.

At last, we must express our sincere heartfelt gratitude to all the staff members of CE Department who helped us directly or indirectly during this course of work.

Abhijeet Panda
Akshat Kumar
Shlok Shukla
Siddhi Chaudhari
Priyanshi Bhootda

B.Tech. (CE) Semester – VI

INTRODUCTION

In a fast-paced world where convenience and efficiency are paramount, online shopping has revolutionized the way consumers engage with retail. The Online Shopping App is designed to streamline this experience, providing users with an intuitive platform to explore a vast array of products from the comfort of their homes. Whether searching for the latest fashion trends, essential household items, or unique gifts, our app offers a seamless shopping experience tailored to individual preferences.

Harnessing advanced technology and a user-friendly interface, the Online Shopping App connects consumers to a diverse marketplace, ensuring access to high-quality products at competitive prices. Users can effortlessly browse through various categories, utilize advanced search filters, and discover personalized recommendations based on their shopping history and interests.

The app features a secure and simplified checkout process, complete with multiple payment options and real-time order tracking. To enhance user engagement, the Online Shopping App incorporates customer reviews and ratings, fostering a community-driven shopping environment that empowers users to make informed choices.

As the e-commerce landscape continues to evolve, the importance of a reliable and efficient shopping solution cannot be overstated. The Online Shopping App not only simplifies the purchasing process but also elevates the overall shopping experience, allowing users to shop with confidence and convenience in a dynamic retail environment.

OBJECTIVE

The primary objective of the Online Shopping App is to provide users with a seamless and convenient shopping experience that caters to their diverse needs and preferences. By leveraging advanced technology, the app aims to deliver a user-friendly interface that simplifies product discovery, enabling customers to browse, search, and compare items effortlessly.

Additionally, the app seeks to enhance user engagement through personalized recommendations, allowing shoppers to find products tailored to their interests. A key objective is to ensure a secure and efficient checkout process, minimizing friction and fostering trust in online transactions.

Furthermore, the app aims to build a community-driven environment by incorporating customer reviews and ratings, helping users make informed purchasing decisions. Ultimately, the Online Shopping App strives to empower consumers by providing them with the tools and information they need to enjoy a safe, efficient, and enjoyable shopping experience in an increasingly digital world.

TECHNOLOGY STACK

The "Online Shopping app" is developed using Kotlin, a modern programming language that has become increasingly popular for Android app development. Kotlin offers several advantages, including concise syntax, enhanced readability, and null safety, which significantly reduce the risk of runtime errors. By leveraging Kotlin's features, developers can write cleaner and more maintainable code, leading to improved productivity and a faster development cycle. This focus on code quality is essential for delivering a robust and reliable weather forecasting application.

Kotlin is fully interoperable with Java, which allows the integration of existing Java libraries and frameworks into the app seamlessly. This interoperability enables developers to take advantage of a rich ecosystem of tools and libraries while still utilizing the benefits of Kotlin's modern programming paradigms. By adopting Kotlin, the "Weather Forecast" app can utilize powerful libraries for network calls, data storage, and user interface design, ensuring a rich and feature-rich experience for users.

In addition to Kotlin, the app leverages various Android frameworks and libraries to enhance its functionality. For instance, Retrofit is used for efficient network communication, allowing the app to fetch realtime weather data from reliable meteorological APIs. The use of libraries like Gson for data serialization ensures that the app can parse and manage the incoming data efficiently. Moreover, Android Jetpack components, such as LiveData and ViewModel, are utilized to manage UI-related data in a lifecycle-conscious way, enhancing the app's performance and responsiveness.

Overall, the choice of Kotlin as the primary programming language, combined with a selection of robust libraries and frameworks, positions the "Online Shopping app" to provide users with an efficient, reliable, and user-friendly experience.

SYSTEM DESIGN

The system design of the “Online Shopping App” is structured to ensure scalability, reliability, and user-friendliness. At its core, the architecture follows a client-server model, where the app serves as the client that communicates with a remote server to fetch real-time weather data. This design allows for efficient updates and the ability to handle a large number of users concurrently without compromising performance. The use of RESTful APIs enables the app to request and receive in a standardized format, ensuring compatibility with various external data sources.

The user interface (UI) is designed with a focus on simplicity and intuitiveness, utilizing Material Design principles to create a visually appealing experience. The main screen displays current weather conditions, hourly and daily forecasts, and relevant alerts. Navigation is streamlined, allowing users to quickly access different sections of the app, such as settings, location management, and educational resources about weather phenomena. Responsive design principles ensure that the app is optimized for various screen sizes and orientations, providing a seamless experience across different Android devices.

To manage data efficiently, the app employs a Model-ViewViewModel (MVVM) architecture. This design pattern separates the app's logic into distinct components, allowing for better organization and maintainability of code. The Model component interacts with the API to fetch weather data, while the ViewModel handles the presentation logic and communicates with the View (UI) to update the display based on user interactions and data changes. This separation of concerns enhances the app's responsiveness and makes it easier to implement features and fixes in the future.

For data storage, the app utilizes Room, an SQLite-based persistence library, to cache weather data locally. This allows for offline access to previously fetched forecasts and minimizes the need for constant API calls, thereby reducing network usage and enhancing performance. By combining these elements sclient-server architecture, a user-centric UI, MVVM design, and local data storage—the "Weather Forecast" app is built to provide users with a fast, reliable, and enjoyable weather forecasting experience.

ARCHITECTURE

The architecture of the "Online Shopping App" is designed to facilitate seamless communication between the client and server while ensuring a smooth and responsive user experience. It employs a layered architecture that consists of several key components: the Presentation Layer, the Business Logic Layer, and the Data Layer. This separation of concerns enhances maintainability, scalability, and the ability to integrate new features efficiently.

At the top of the architecture is the **Presentation Layer**, which is responsible for the user interface (UI) and user interactions. Built using Android's Jetpack Compose and Material Design principles, this layer ensures a clean and responsive UI that adapts to various screen sizes and orientations. It handles user inputs, such as location settings and preferences, and communicates with the View Model to display weather data and updates dynamically. The UI components are designed to be modular, allowing for easy updates and enhancements without impacting the overall functionality of the app.

The **Business Logic Layer** is implemented through the Model View View Model (MVVM) design pattern. The View Model serves as an intermediary between the Presentation Layer and the Data Layer, encapsulating the app's business logic. It retrieves weather data from the repository and prepares it for presentation in the UI. This layer also manages user interactions, such as refreshing data or responding to alerts, ensuring that the app remains responsive to user needs. By separating the business logic from the UI, this architecture promotes a cleaner codebase that is easier to test and maintain.

At the bottom is the **Data Layer**, which handles data storage and retrieval. This layer utilizes a combination of local and remote data sources. Room, an SQLite-based persistence library, is used to store cached weather data locally, allowing the app to provide offline access to previously fetched information. For real-time updates, the app connects to a remote weather API using Retrofit, which facilitates efficient network calls to fetch the latest forecasts. This dual approach to data management not only enhances performance by reducing network dependency but also improves the overall user experience by ensuring that critical information is readily available.

In summary, the architecture of the "Online Shopping App" is structured to provide a robust, scalable, and maintainable solution for delivering weather information. By employing a layered approach, integrating the MVVM design pattern, and utilizing both local and remote data sources, the architecture ensures that users receive accurate and timely weather updates while enjoying a seamless and engaging app experience.

IMPLEMENTATION

The implementation of the “Online Shopping App” involves several key stages, including planning, coding, testing, and deployment. The process begins with thorough planning, where the requirements and features of the app are defined. This stage involves identifying user needs, selecting appropriate technologies, and designing the app’s architecture and user interface. By focusing on user experience from the outset, the development team ensures that the app is both functional and intuitive.

The coding phase primarily utilizes Kotlin, which is the official programming language for Android development. The development team leverages Android Studio, the integrated development environment (IDE) for Android, to build the app. Utilizing the MVVM architecture, developers create distinct components for the user interface, business logic, and data handling. The user interface is designed using Jetpack Compose, allowing for a modern, responsive, and customizable layout. Developers implement various screens, such as the home screen displaying current weather, hourly forecasts, and alerts, ensuring smooth navigation throughout the app.

During implementation, the app integrates several essential libraries and APIs. Retrofit is used for network communication, facilitating efficient data retrieval from the Weather API. This library simplifies the process of making HTTP requests and parsing JSON responses, enabling seamless interaction with external weather data sources. Additionally, Room is employed as the local database solution, allowing the app to cache weather data for offline access. This integration ensures that users can view previously fetched weather information even without an internet connection.

Testing is a critical phase of the implementation process, ensuring that the app functions correctly and meets user expectations. The development team conducts both unit testing and user acceptance testing (UAT) to identify and resolve any bugs or usability issues. Unit tests are

written to validate individual components, while UAT involves real users interacting with the app to provide feedback on performance and functionality. Once testing is complete and any necessary adjustments are made, the app is prepared for deployment on the Google Play Store.

In summary, the implementation of the “Online Shopping App” is a comprehensive process that emphasizes user experience, robust architecture, and thorough testing. By utilizing modern technologies and best practices, the app aims to deliver a reliable and engaging weather forecasting solution that meets the needs of its users.

API INTEGRATION

API integration is a crucial component of the “Online Shopping App”, enabling it to fetch real-time weather data from external sources. The app relies primarily on a RESTful weather API, which provides access to a wide range of meteorological data, including current conditions, hourly forecasts, and long-term predictions. This integration allows users to receive accurate and timely weather information tailored to their specific locations.

The integration process begins with the selection of a reliable weather API. The chosen API typically requires an API key for authentication, which is included in each request to ensure secure access to the data. Popular weather APIs, such as OpenWeatherMap, Weatherstack, or AccuWeather, are commonly used due to their extensive data coverage and ease of use. Developers review the API documentation to understand the available endpoints, data formats, and usage limits, which helps in planning the app’s data retrieval strategies.

In the implementation phase, Retrofit, a powerful HTTP client for Android, is utilized for making network requests to the weather API. Retrofit simplifies the process of constructing requests, handling responses, and parsing JSON data. Developers define data models that represent the structure of the weather data returned by the API. By using annotations, Retrofit converts the API responses into Kotlin objects, making it easy to work with the data within the app. This streamlined process enhances the efficiency of data retrieval and minimizes the likelihood of errors in data handling.

To ensure a smooth user experience, the app implements caching mechanisms using the Room database. When the app fetches weather data from the API, it also stores a local copy in the database. This allows the app to provide offline access to previously fetched data, reducing the dependency on network connectivity.

UI DESIGN

The user interface (UI) design of the “Online Shopping App” is centered around providing an intuitive and engaging experience for users. The design follows Material Design principles, ensuring a modern and visually appealing layout that is both functional and easy to navigate. The primary goal is to make essential weather information readily accessible while ensuring a seamless interaction flow throughout the app.

The main screen of the app prominently displays the current weather conditions, including temperature, humidity, and wind speed. This information is presented using large, easily readable fonts and eye catching icons that reflect the current weather status (e.g., sunny, rainy, cloudy). To enhance readability, a color scheme is employed that contrasts well with the text, ensuring that users can quickly grasp the information, even in varying lighting conditions.

In addition to the current weather display, the UI includes a detailed hourly and daily forecast section. Users can swipe through these forecasts to see how the weather will change throughout the day and week. Each forecast card features relevant icons and a color-coded background that aligns with the weather conditions. This design choice not only makes the information visually appealing but also reinforces the app’s dynamic background color feature, enhancing the overall aesthetic experience.

CHALLENGES FACED

During the development of the “Online Shopping App”, the team encountered several challenges that required creative problem-solving and adaptability. One of the primary challenges was ensuring accurate and timely data retrieval from the weather API. Integrating a thirdparty API often comes with limitations such as rate limiting, data availability, and varying response times. To address this, the team implemented robust error handling mechanisms that notify users when data retrieval fails, along with caching strategies using Room to store previously fetched data. This ensures that users can still access weather information even if the app is unable to connect to the API at that moment.

Another significant challenge was optimizing the app’s performance, particularly when handling large datasets for hourly and daily forecasts. As weather data can include numerous parameters, loading this information efficiently while maintaining a responsive user interface became crucial. The development team employed pagination and lazy loading techniques to display data incrementally rather than all at once. This not only improved the app’s loading speed but also enhanced the user experience by providing a smoother interaction as users scrolled through forecast details.

User experience also posed a challenge during the design phase. Striking the right balance between a visually appealing interface and functional usability required extensive user testing and feedback. Initial designs that looked aesthetically pleasing often sacrificed usability, leading to confusion among users about how to navigate the app effectively. To overcome this, the team conducted multiple rounds of usability testing, incorporating user feedback to refine the design iteratively. This process ensured that the final design was both attractive and intuitive, ultimately enhancing user satisfaction.

Lastly, adapting to the rapid changes in technology and user expectations presented an ongoing challenge. The mobile app development landscape evolves quickly, with new frameworks,

libraries, and design trends emerging regularly. The team had to stay updated with the latest developments in Kotlin and Android development best practices to ensure that the app remained relevant and competitive. Continuous learning and adaptability became integral parts of the development process, enabling the team to implement the best solutions available.

In summary, the development of the "Weather Forecast" app involved navigating various challenges related to data integration, performance optimization, user experience design, and staying abreast of technological advancements. Through

proactive problem-solving and a focus on user feedback, the team successfully addressed these challenges, resulting in a robust and user-friendly weather forecasting application.

FUTURE ENHANCEMENT

As the “Online Shopping App” continues to evolve, several enhancements are planned to improve functionality, user experience, and overall performance. These future enhancements aim to leverage user feedback and emerging technologies to ensure that the app remains competitive and meets the changing needs of its users.

AI-Powered Recommendations: Integrating machine learning algorithms to provide more personalized product recommendations based on users' browsing and purchasing history, improving customer satisfaction and increasing sales potential.

Augmented Reality (AR) Integration: Implementing AR technology to allow users to visualize products, such as clothing, furniture, or home decor, in their own environments before making a purchase, enhancing their shopping experience.

Voice Search and AI Chatbot: Adding voice search functionality for hands-free browsing and implementing AI-driven chatbots to assist users with queries, product suggestions, and order tracking, offering a more interactive and user-friendly experience.

Loyalty Programs and Gamification: Introducing loyalty programs, reward points, and gamified features, such as daily challenges or special promotions, to increase user engagement and retention.

Enhanced Security with Blockchain: Exploring the use of blockchain technology to secure transactions, improve transparency, and safeguard sensitive customer information, especially in terms of payment processing and data privacy.

Social Shopping and Reviews: Expanding social sharing features, allowing users to share their favourite products and reviews on social media, while also incorporating real-time social proof, such as product reviews from friends or trending items, to build trust and community.

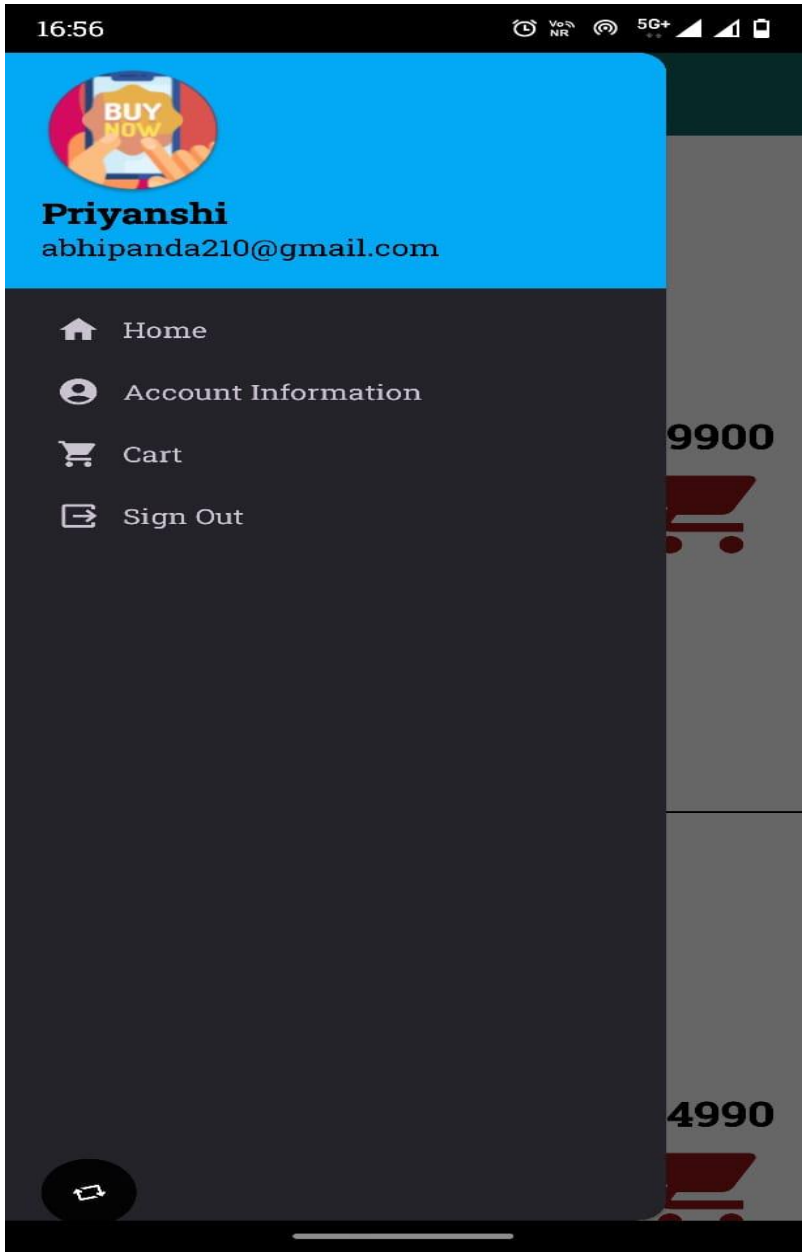
Advanced Analytics for Sellers: Offering advanced analytics and reporting tools to vendors and sellers, enabling them to track their sales, customer behavior, and inventory, allowing for more data-driven decisions to optimize their business.

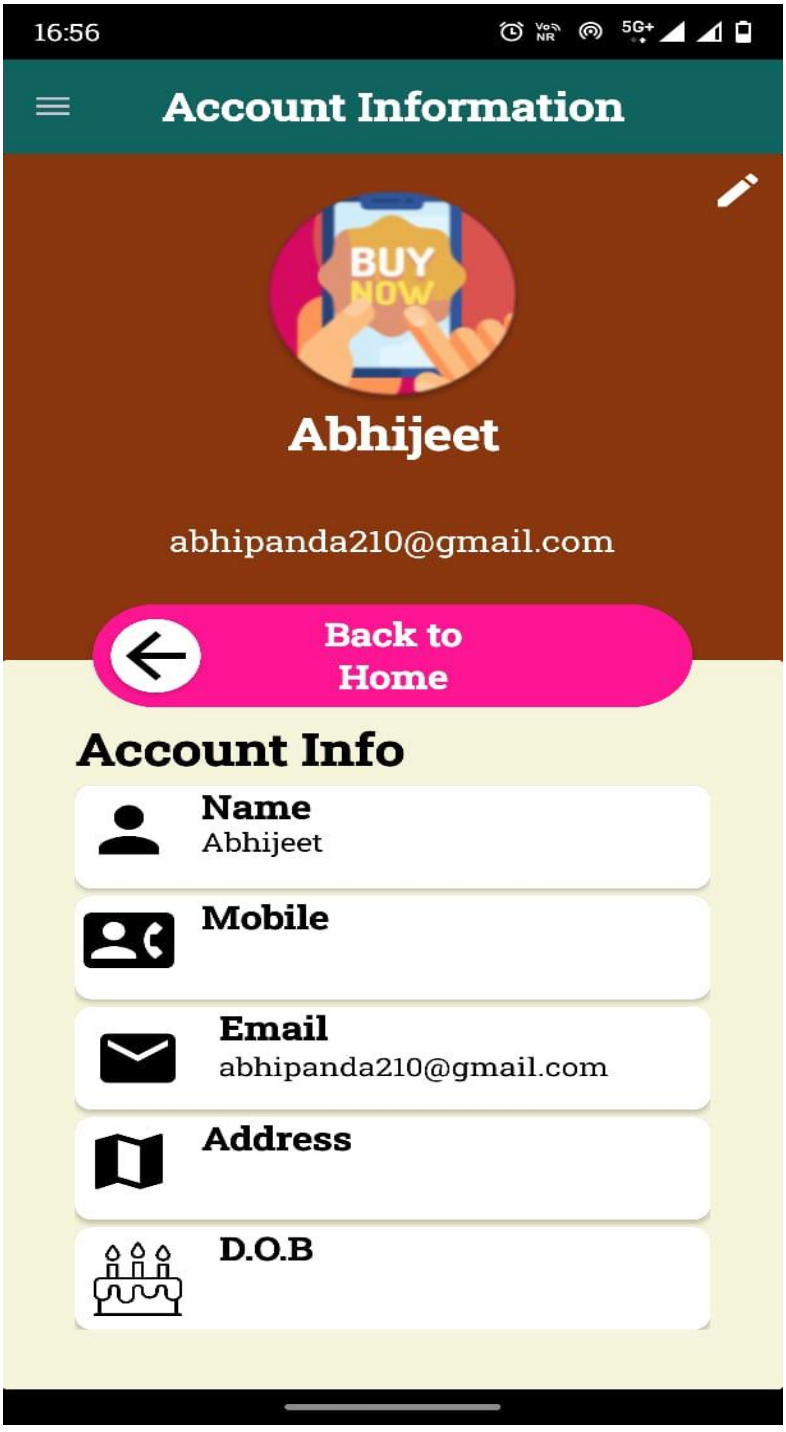
Global Expansion with Multi-Language and Multi-Currency Support: Adding support for multiple languages and currencies to cater to a global user base, ensuring a seamless experience for international shoppers.

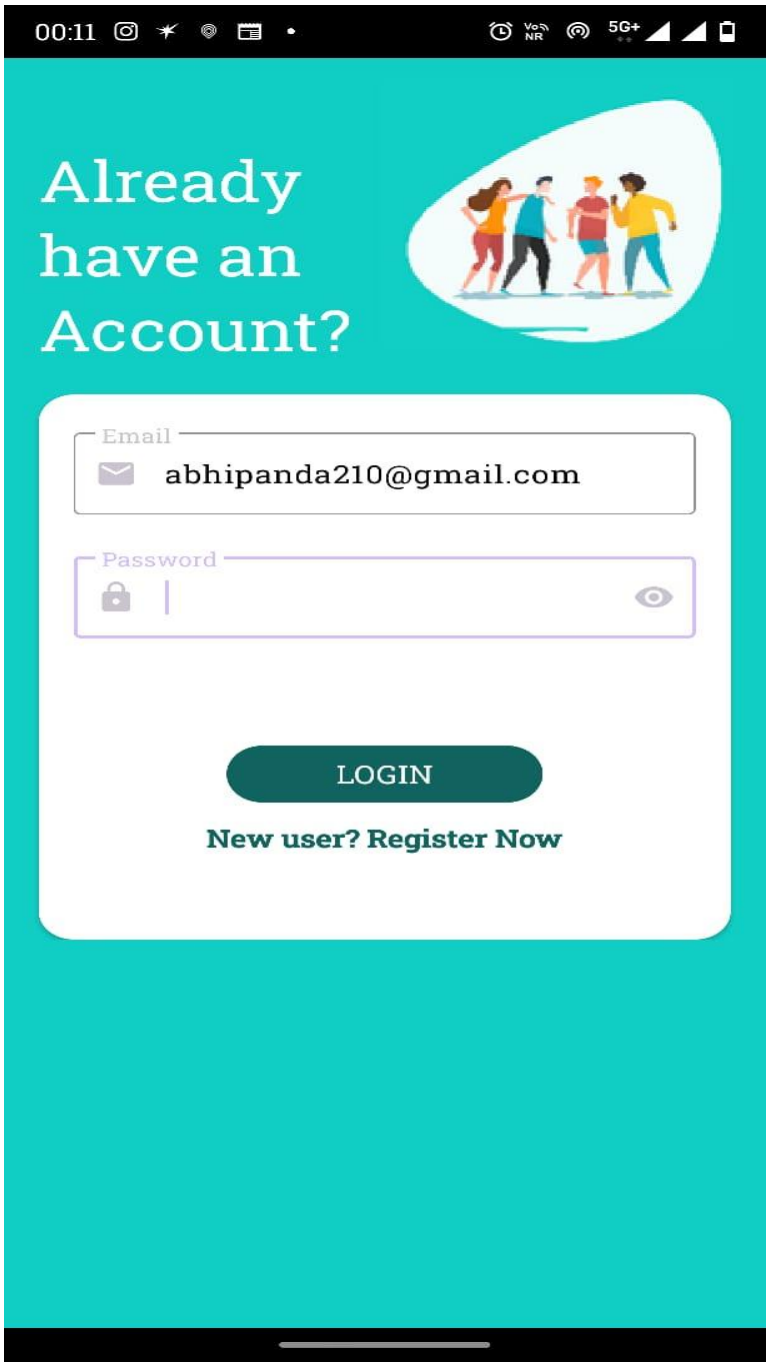
One-Click Checkout and Buy Now Features: Simplifying the purchasing process with a one-click checkout option, making it easier and faster for returning users to complete their transactions without re-entering payment or shipping details.

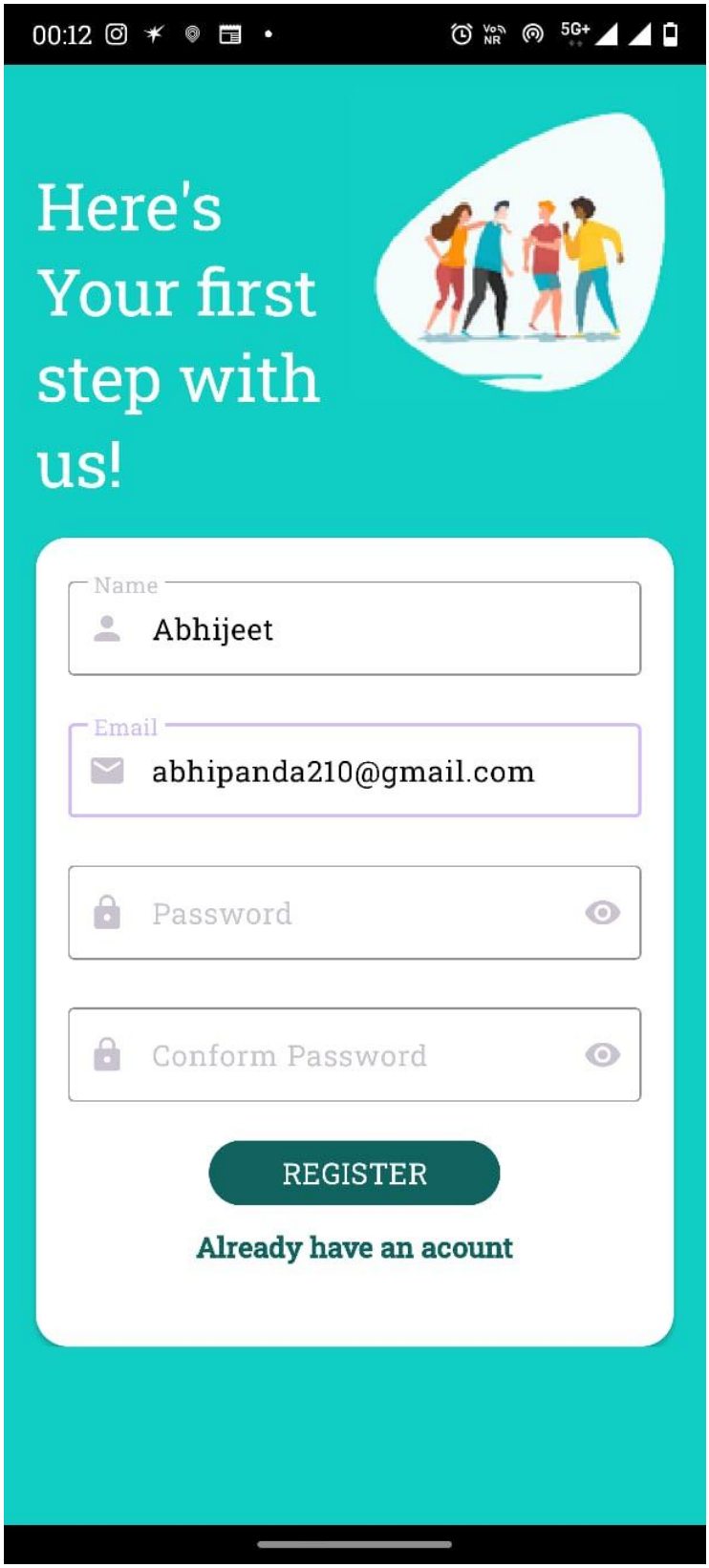
Subscription-Based Shopping: Introducing subscription-based services where users can sign up for automatic reorders of specific products (e.g., groceries or personal care items) at regular intervals, improving customer convenience.

PROJECT SNIPPETS




16:56
Account Information
BUY NOW
Abhijeet
abhipanda210@gmail.com
Back to Home
Account Info
Name
Abhijeet
Mobile
Email
abhipanda210@gmail.com
Address
D.O.B


00:11
Already have an Account?
Email
abhipanda210@gmail.com
Password
LOGIN
New user? Register Now


00:12
Here's Your first step with us!
Name
Abhijeet
Email
abhipanda210@gmail.com
Password
Conform Password
REGISTER
Already have an acount

12:27 2.95 KB/S 67%

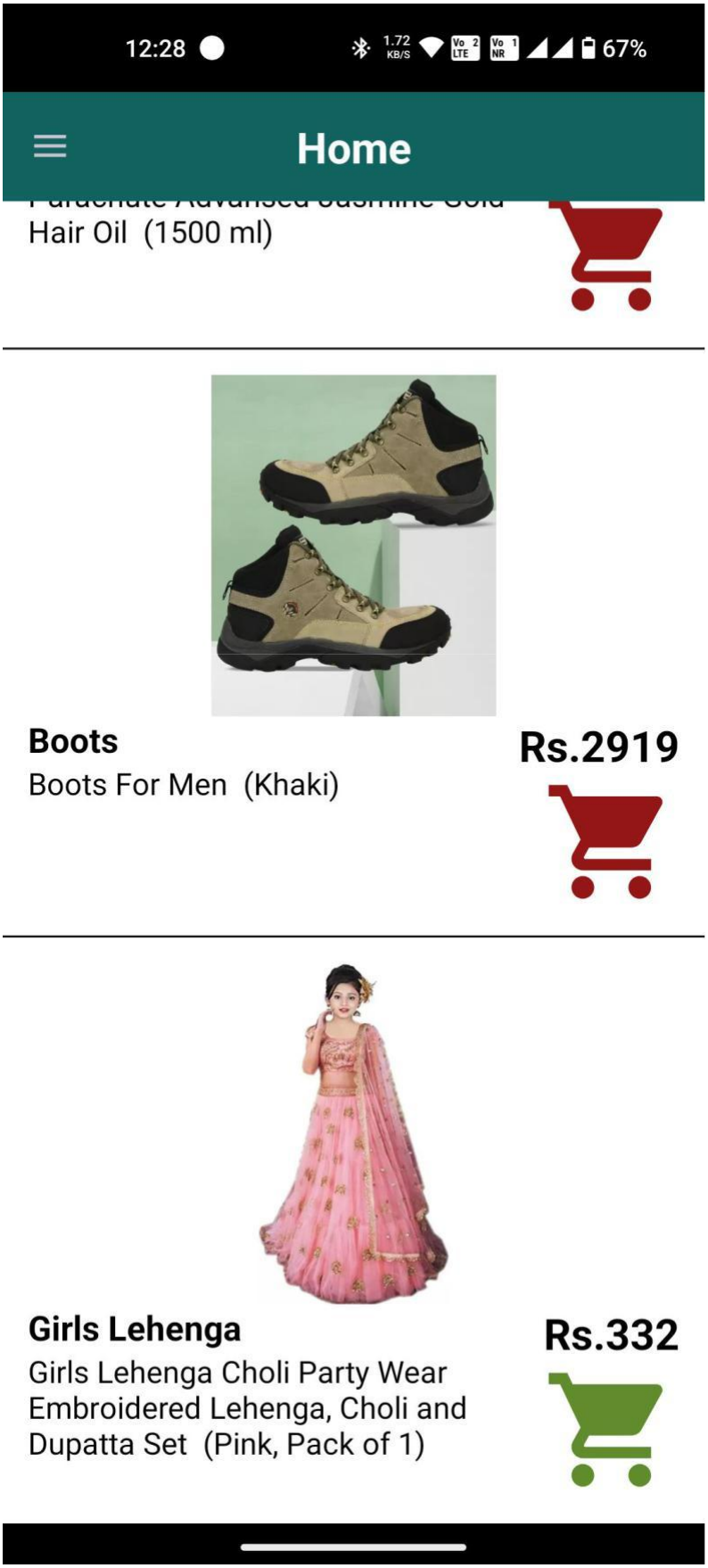
Home

iPhone 15 (Blue, 128 GB) **Rs.79900**

RAM | ROM
Processor A16 Bionic Chip, 6 Core Processor | Hexa Core
Front Camera
48MP + 12MP
Front Camera
12MP
Display
6.1 inch All Screen OLED Display

APPLE AirPods Pro **Rs.24990**

APPLE AirPods Pro (2nd generation) with Active Noise Cancellation, Spatial Audio Bluetooth Headset (White, True


12:28
67%
Home
Hair Oil (1500 ml)
Boots
Boots For Men (Khaki)
Rs.2919
Girls Lehenga
Girls Lehenga Choli Party Wear Embroidered Lehenga, Choli and Dupatta Set (Pink, Pack of 1)
Rs.332


12:28
67%
Cart
iPhone 15 (Blue, 128 GB)
Rs.79900
RAM | ROM
Processor A16 Bionic Chip, 6 Core Processor | Hexa Core
Front Camera
48MP + 12MP
Front Camera
12MP
Display
6.1 inch All Screen OLED Display
MacBook AIR M2
Rs.97990
APPLE 2022 MacBook AIR M2 - (8 GB/256 GB SSD/Mac OS Monterey) MLY13HN/A (13.6 Inch, Starlight, 1.24 kg)
PLEASE ORDER
Girls Lehenga
Rs.332

CONCLUSION

The “Online Shopping app” has been developed with a strong focus on user experience, functionality, and reliability. By leveraging modern technologies such as Kotlin, Retrofit, and Room, the app provides users with real-time weather information in an engaging and intuitive format. Through thoughtful UI design and the integration of dynamic features like color-changing backgrounds, the app enhances user engagement while ensuring that essential weather data is easily accessible.

Throughout the development process, the team faced several challenges, including data integration, performance optimization, and user experience design. However, by implementing effective solutions and prioritizing user feedback, the app has successfully overcome these obstacles, resulting in a robust and user-friendly application. The incorporation of features like personalized weather alerts and improved data visualization has further enriched the user experience, demonstrating the team’s commitment to continuous improvement. Looking forward, the planned enhancements will position the

“Online Shopping app” for sustained success in a competitive market. By focusing on personalization, expanding geographic coverage, and integrating with emerging technologies, the app aims to meet the evolving needs of its users. The development team remains dedicated to refining the app and ensuring that it provides accurate, timely, and relevant weather information.

In conclusion, the “Online Shopping app” represents a significant step forward in delivering a comprehensive weather forecasting solution. Through its innovative features and commitment to user satisfaction, the app not only meets the current demands of users but also sets the foundation for future growth and enhancements.