UNIVERSITY OF BUEA

REPUBLIC OF CAMEROON

PEACE-WORK-FATHERLAND

P.O. Box 63,
Buea, South West Region
CAMEROON
Tel: (237) 3332 21 34/3332 26 90
Fax: (237) 3332 22 72

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

ARCHIVAL AND RETRIEVAL OF MISSING OBJECTS USING IMAGE MATCHING ALGORITHMS AND ADVANCED MACHINE VISION TECHNIQUES

PHASE 5: USER INTERFACE DESIGN AND IMPLEMENTATION

By:

Group 6

Supervisor:

Dr Nkemeni Valery

University of Buea

LIST OF PARTICIPANTS

NAMES	MATRICULE	Speciality
KOUETE TEPE KENNETH	FE21A220	Software
ZELEFACK MARIE	FE21A330	Software
EFUETAZOH ASONG RODERIC	FE21A179	Network
TAKO DIEUDONNEVOJONG	FE21A310	Software
BUINFUN MONIE JULIUS	FE21A154	Software

Table Of Figures

INTRODUCTION

The "FindIt" application is a cutting-edge platform designed to help users locate lost items efficiently. In today's fast-paced world, misplacing personal items can lead to significant inconvenience and stress. "FindIt" aims to alleviate this problem by providing a reliable and user-friendly tool for locating lost belongings. This report details the comprehensive journey of designing and implementing the user interface (UI) for the "FindIt" application, emphasizing the importance of UI/UX design in creating a user-friendly and intuitive experience.

The project encompasses both design and implementation phases, leveraging Figma for the design process and React Native with Expo for development. The UI/UX design is a crucial aspect of this project as it directly impacts the user experience, engagement, and overall effectiveness of the application. A well-designed interface not only makes the application more appealing but also ensures that users can navigate and utilize the app's features with ease.

This report will cover the following aspects in detail:

1. **Basic Notions of Design Applied**: An introduction to fundamental design principles that guide the creation of user-friendly interfaces.
2. **Design Process**: A step-by-step walkthrough of the design phases, from initial concept to final prototype, illustrating how ideas were transformed into visual designs.
3. **Characteristics of Good Design**: Key attributes that define an effective and user-friendly design, ensuring the application meets high usability standards.
4. **The design implementation: design principles**
5. **Application Implementation Process**: The technical journey of transforming designs into a functional application using modern tools and technologies, focusing on how design elements were brought to life.

THE DESIGN PHASE

The design phase is a critical component in the development of the "FindIt" application, serving as the foundation for creating a user-friendly and intuitive user interface (UI). This phase involves a meticulous process of transforming ideas and requirements into visual

representations that not only meet the needs of the users but also enhance their overall experience with the application.

Designing a UI is more than just arranging visual elements on a screen; it is about crafting an experience that is both functional and aesthetically pleasing. A well-designed interface can significantly influence how users perceive and interact with the application, which is why this phase is given utmost importance. The design process encompasses several stages, each aimed at refining the UI to ensure it is effective, engaging, and accessible to all users.

The primary goals of the design phase were to:

1. **Understand User Needs**: Gather and analyse user requirements to ensure the design aligns with their expectations and needs.
2. **Create an Intuitive Layout**: Develop a layout that is easy to navigate, reducing the learning curve for new users and enhancing the overall usability.
3. **Ensure Visual Appeal**: Design a visually appealing interface that attracts users and encourages interaction.
4. **Enhance Usability**: Focus on usability principles to make the application intuitive, reducing the chances of user errors and improving satisfaction.
5. **Maintain Consistency**: Ensure consistency across all design elements to create a cohesive and predictable user experience.
6. **Incorporate Feedback**: Iterate on designs based on feedback from stakeholders and users to continuously improve the UI.

Tools

- Figma

What is design?

“Most people make the mistake of thinking design is what it looks like. People think it’s this veneer that the designers are handed this box and told, ‘Make it look good!’ That’s not what we think design is. **It’s not just what it looks like and feels like**. Design is how it works”

What is a good design?

“A solution that serves the users and satisfies the client.”

The good design:

- Does what the user needs and want.
- Is natural to use and
- Is easy to understand. In other words, help the users avoid troubles

Basic Concept of Design(usability Principle).

Understanding the basic concepts of design is essential for creating a user interface (UI) that is not only functional but also visually appealing and user-friendly. The "FindIt" application leverages these fundamental principles to ensure an intuitive and engaging experience for its users. Below are the key concepts of design that guided the creation of the "FindIt" application:

User-Cantered Design

User-cantered design (UCD) places the user at the heart of the design process. This approach ensures that the needs, preferences, and limitations of end-users are considered at every stage of the design. The goal is to create an intuitive interface that requires minimal learning and enhances user satisfaction. Key practices include:

- **User Research**: Conducting interviews, surveys, and usability tests to gather insights directly from users.
- **Personas**: Creating detailed user personas to represent different segments of the target audience and their specific needs.
- **User Scenarios and Journey Maps**: Mapping out the user's journey to understand how they will interact with the application.

Simplicity and Clarity

A good design should be simple and clear, avoiding unnecessary complexity. Simplicity helps users to focus on the primary functions without being overwhelmed. This includes:

- **Minimalism**: Using only essential elements and eliminating anything that does not serve a functional purpose.
- **Whitespace**: Utilizing whitespace effectively to create a clean and uncluttered interface, which helps in highlighting important elements and improving readability.

Consistency

Consistency in design ensures that similar elements behave in predictable ways. This helps users learn the interface quickly and reduces confusion. Consistency is maintained in:

- **Visual Elements**: Consistent use of colors, fonts, and icons throughout the application.
- **Functionality**: Similar functions are accessed in the same way across different parts of the application.

Visual Hierarchy

Visual hierarchy is the arrangement of elements in a way that implies importance. It helps users to focus on what is most important first. Techniques to establish visual hierarchy include:

- **Size and Scale**: Larger elements draw more attention than smaller ones.
- **Color and Contrast**: Bright colors and high contrast can make elements stand out.
- **Typography**: Using different font sizes, weights, and styles to distinguish between headings, subheadings, and body text.

Aesthetic Integrity

Aesthetic integrity refers to the visual appeal of the design and how well it fits with the overall purpose and tone of the application. This includes:

- **Brand Alignment**: Ensuring the design aligns with the brand's identity and values.
- **Pleasant Appearance**: Creating an interface that is visually pleasing and enjoyable to use.

Provide all necessary help

Help should be Searchable, context sensitive, task sensitive, concrete, short, and most important **NOT NEEDED**. In one word, accurate. Integrate help with the application

By incorporating these basic design concepts, the "FindIt" application aims to provide a seamless, engaging, and efficient user experience. The design principles guide every aspect of the UI, ensuring that the final product is not only functional but also intuitive and aesthetically

pleasing. This strong foundation in design concepts helps create a product that meets user needs and enhances their overall experience with the application.

The Design Process

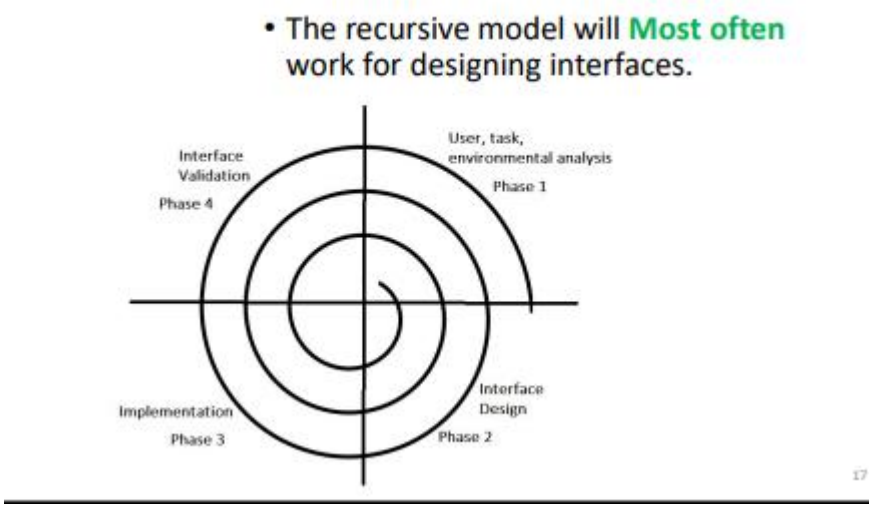


Figure 1. Method used

Phase 1: Requirements analysis

The design process for the "FindIt" application follows a structured approach to ensure that the final product is user-centric, visually appealing, and functionally robust. This process involves several stages, each contributing to the development of an intuitive and effective user interface (UI). Below are the key stages of the design process:

This phase collects all of the necessary requirements for an interactive system or device and yields a **requirements specification or document** as its outcome.

Requirements documents written specifically for user experience and interaction design aspects are specified in terms of three main components for requirement analysis:

Functional requirements define specific behaviour that the system should support (often captured in so-called use cases);

Non-functional requirements specify overall criteria governing the operation of the interactive system without being tied to a specific action or behaviour (hardware, software, system performance, reliability, etc.); and

User experience requirements explicitly specify non-functional requirements for the user interaction and

Phase 2: Preliminary and detailed design

- The design phase consists of two stages: a preliminary stage where the high-level design or architecture of the interactive system is derived, a detailed stage, where the specifics of each interaction are planned out.
- The outcome from the design phase is a detailed design document.

Phase 3: Build and implementation

- The implementation phase is where all the planning gets turned into actual, running code.
- The outcome from this phase is a working system, albeit not necessarily the final one
- Different platforms are used for interactive applications (Canvas, Figma etc.) we have used Figma in our case.
- Regardless of platform, make sure to evaluate tool capabilities, ease of use, ease to learn, cost, and performance.

Phase 4.1: Evaluation

In the final phase of the design cycle, developers test and validate the system implementation to ensure that it conforms to the requirements and design set out earlier in the process.

- The outcome of the validation process is a validation report specifying test performance.
- A straightforward approach to validate a system specified using use cases is simply to check that each use case can be completed successfully
- Depending on this outcome, the design team can decide to proceed with production and deployment of the

Phase 4.2: Validation

Validation is a vital part of the design process. In addition to a variety of expert review methods, tests with the intended users, surveys, and automated analysis tools are proving to be valuable.

Procedures vary greatly depending on the goals of the usability study and the number the number of expected users, the danger of errors, and the level of investment.

Design framework Used

While the design process discussed above generally should remain the same for all your projects, the approach to performing it may vary following different design frameworks.

In other words, the design framework is the specific flavour and approach the designer takes to conducting the design process.

Three known design frameworks:

- User-centred design
- Participatory design
- Agile interaction design (nascent)

User-Centred Design (UCD)- What we used

- Many software development projects fail to achieve their goals.
- Much of this problem can be traced to poor communication between developers and their business clients or between developers and their users.
- The result is often systems and interfaces that force the users to adapt and change their behaviour to fit the interface rather than an interface that is customized to the needs of the users.
- User-centred design (UCD) prescribes a design process that primarily takes the needs,

Why we Chose this Framework ?

- Gives designers a much-needed understanding of what their users actually need since they are directly involved in the process
- Dramatically reduce both development time and cost when a careful attention is paid to UCD at the early stage of software development.
- UCD leads to systems that generate fewer problems during development and have lower maintenance costs over their lifetimes.
- UCD is easy to learn, result in faster performance, reduce user errors substantially, and encourage users to explore features that go beyond the minimum required to get by.

- UCD reduces the risk of designers building the “wrong system”: a system that the end users neither need nor asked for.
- UCD practices help organizations align system functionality with their business needs and priorities

Desing Implementation: The Principles

Implementing the design for the "FindIt" application involves adhering to core design principles that ensure the user interface (UI) is not only visually appealing but also functional, intuitive, and user-friendly. These principles guide the design decisions and ensure consistency and coherence throughout the application. Below are the key design principles considered during the implementation phase:

1. **Unity**

   Unity creates a sense of harmony between all the elements in the page. A page with elements that are visually or conceptually arranged together will create a sense of unity.

   So, in applying this, we made sure Elements have clear relationships between them
2. **Hierarchy**

   Hierarchy is the perceived order of the various elements in a composition: “who come first and who needs to be read/seen first?”
   - Colour and size are the most common ways to create hierarchy.
3. **Rhythm**

   When all of the elements of a UI have a consistent size and shape that creates a rhythm.

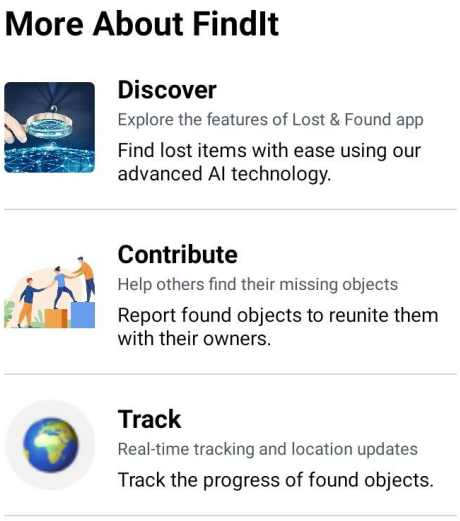


Figure 2: Application of Rhythm principle in FindI

4. **Harmony**

When visual elements are in harmony, they relate to and complement each other.

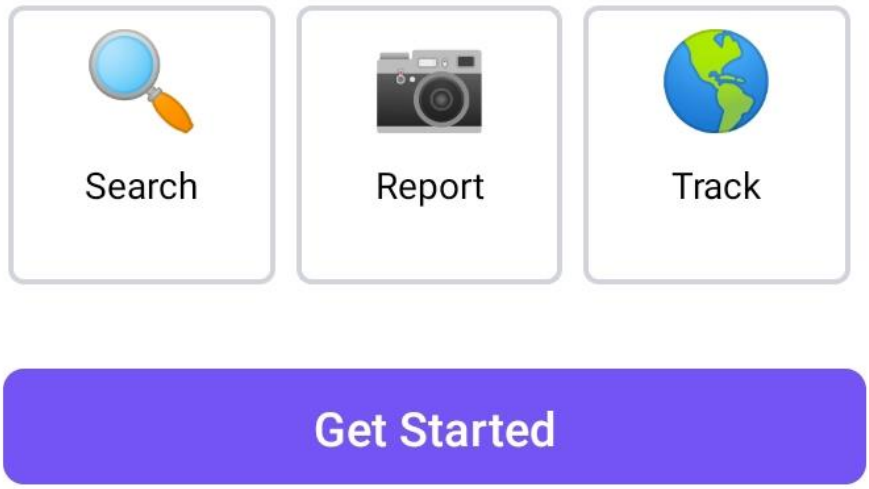


Figure 3: Application of Harmony principle in FindI

5. **Dominance**

- Dominance is a very similar concept to Hierarchy, but it places importance on the available content. E.g., a call-to-action button needs to be bigger and better readable than the others, thus answering to visual hierarchy.
- Dominance can be achieved by making the desired object both larger or more contrast or both.

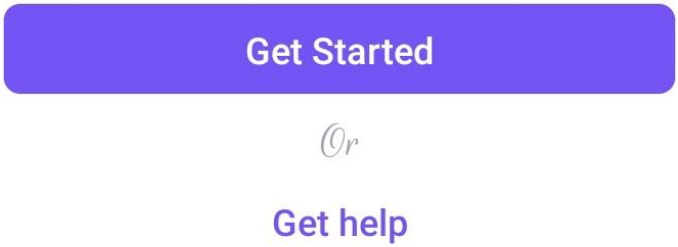


Figure 4: Application of Dominance principle in FindIt

6. **Alignment**

- Alignment is the most important visual design principle.
- Without alignment, a design will be discomforting for a user just at the first glance.

- Even if you do nothing but use proper alignment, your designs will be definitely more useful, usable, and understandable which also makes them more valuable.

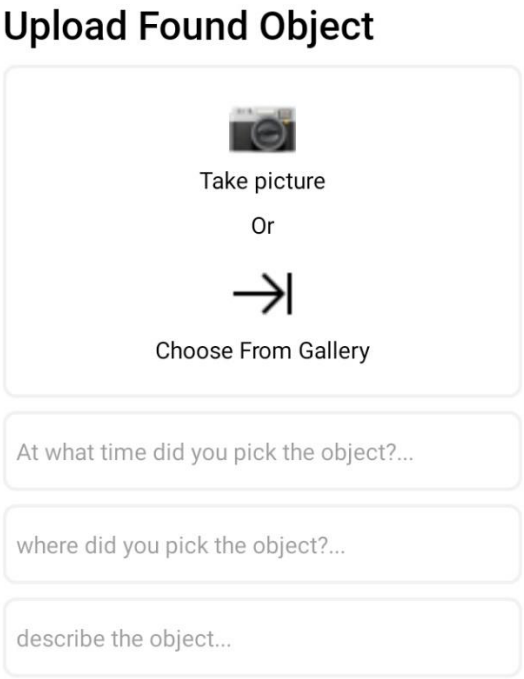


Figure 5: Application of Alignment principle in FindIt

Different Applications Screens

Login Screen

This page permit us then identify users of the system and give only authorise access.

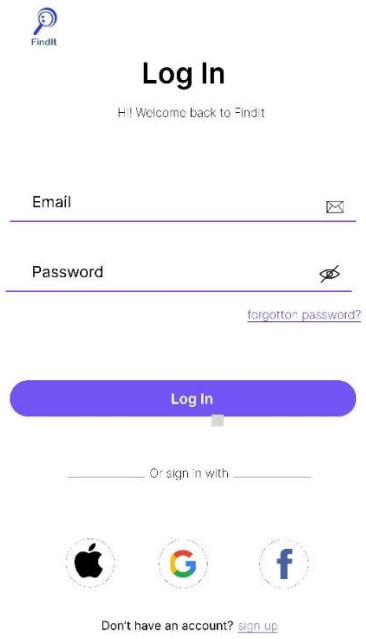


Figure 6: Login page of FindIt

Sign-up Screen

This page permit us to keep Track of the users of our system

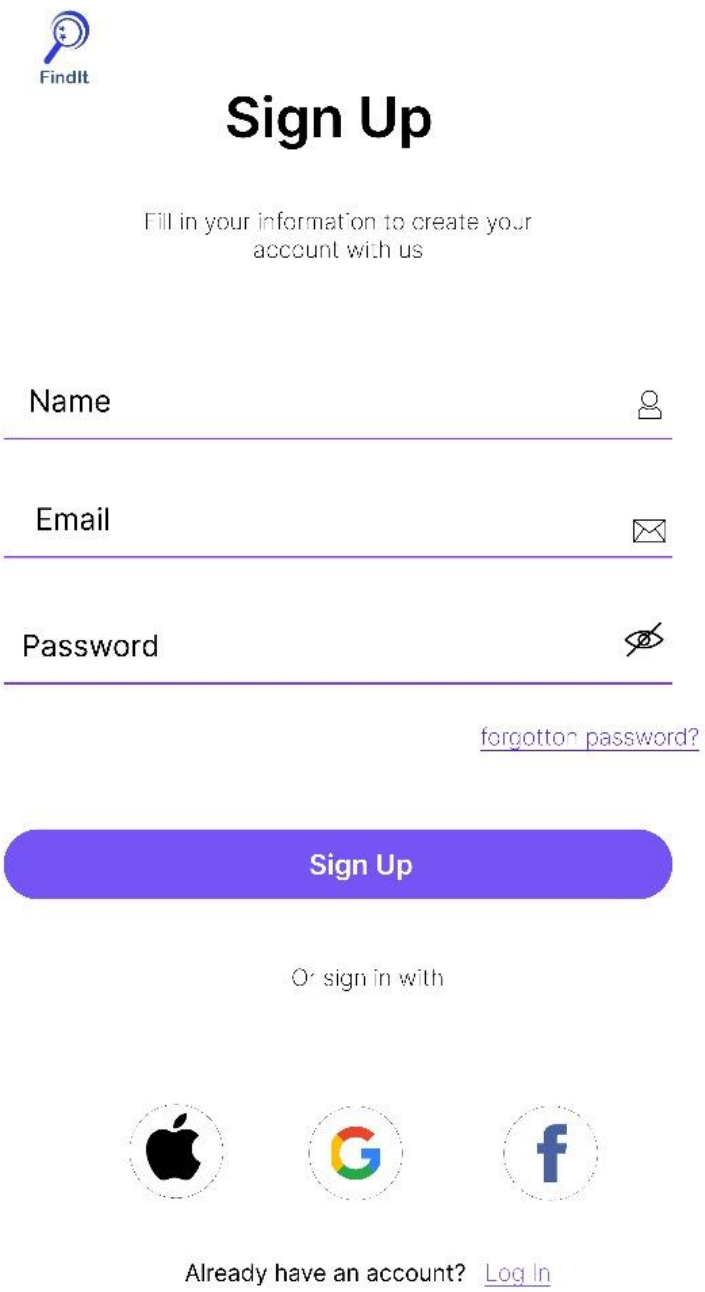


Figure 7: Sign-up page for FindIt

Landing Screen

This page permit us to introduce the Users (especially the new users) to our App before he gets started

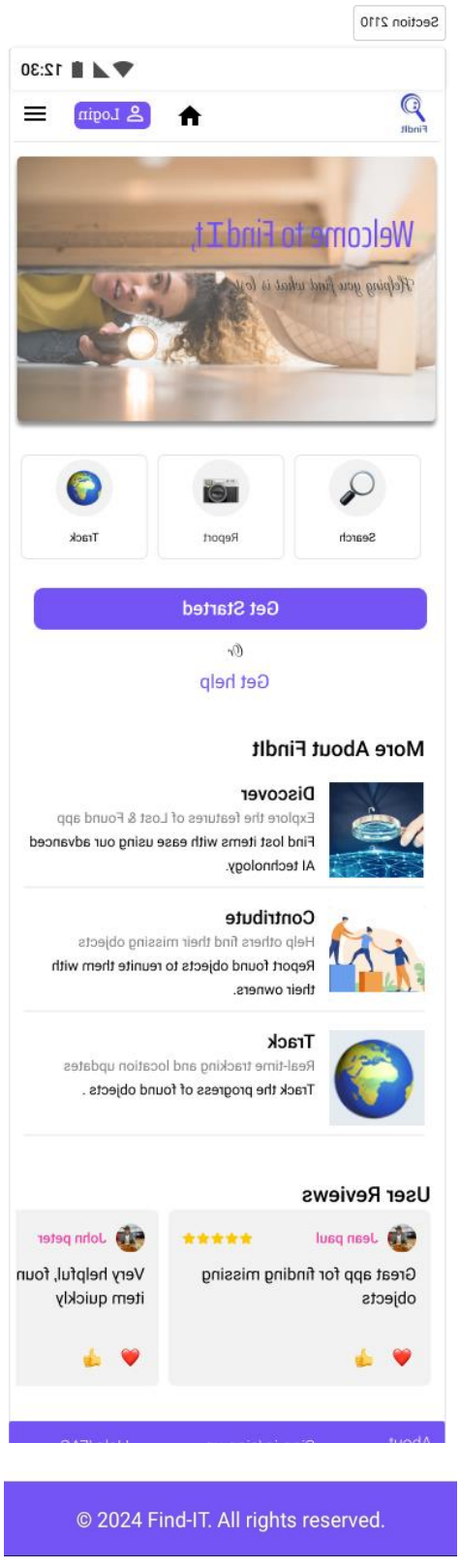
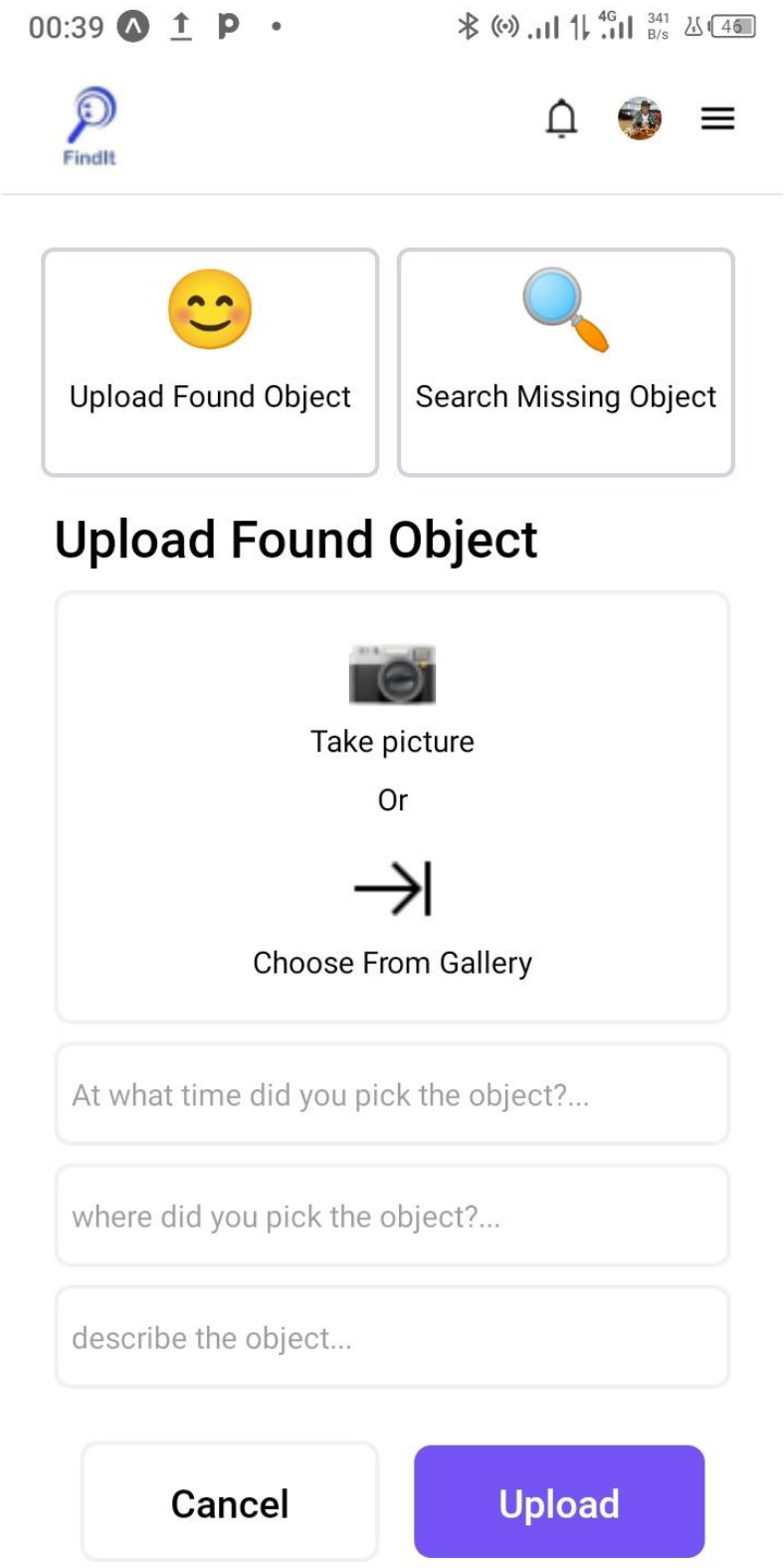
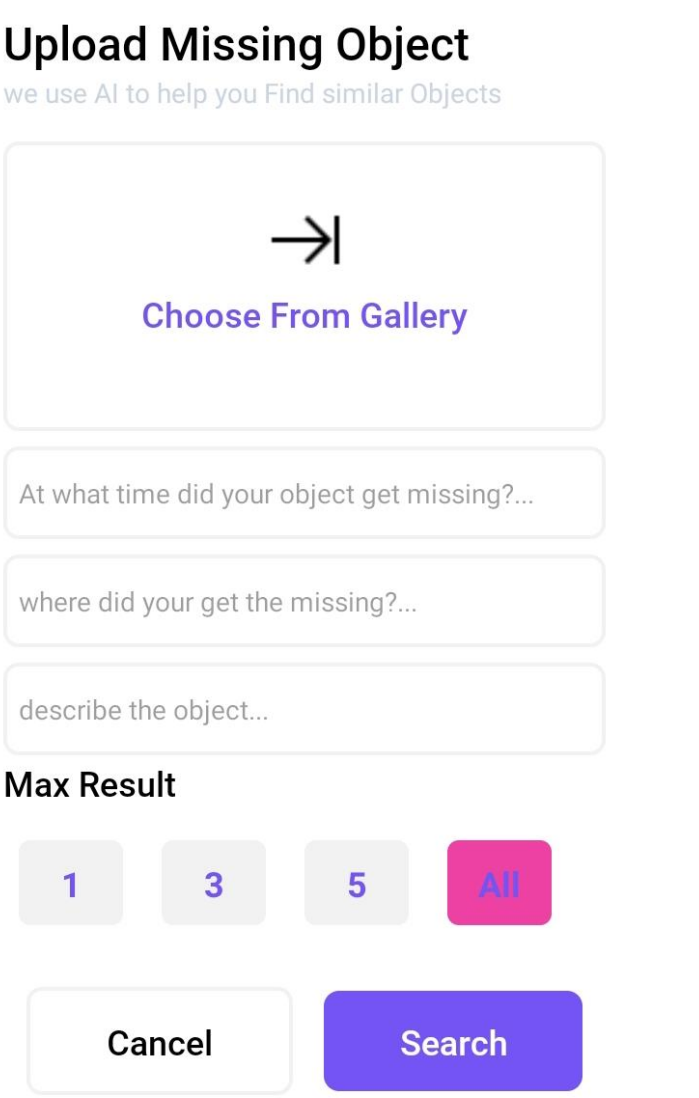


Figure 8: Landing Screen.

Home Screen

Here the User can carry out the main Functionalities such as, Uploading found objects, searching and tracking missing objects.




Upload Missing Object
we use AI to help you Find similar Objects
Choose From Gallery
At what time did your object get missing?...
where did your get the missing?...
describe the object...
Max Result
1
3
5
All
Cancel
Search

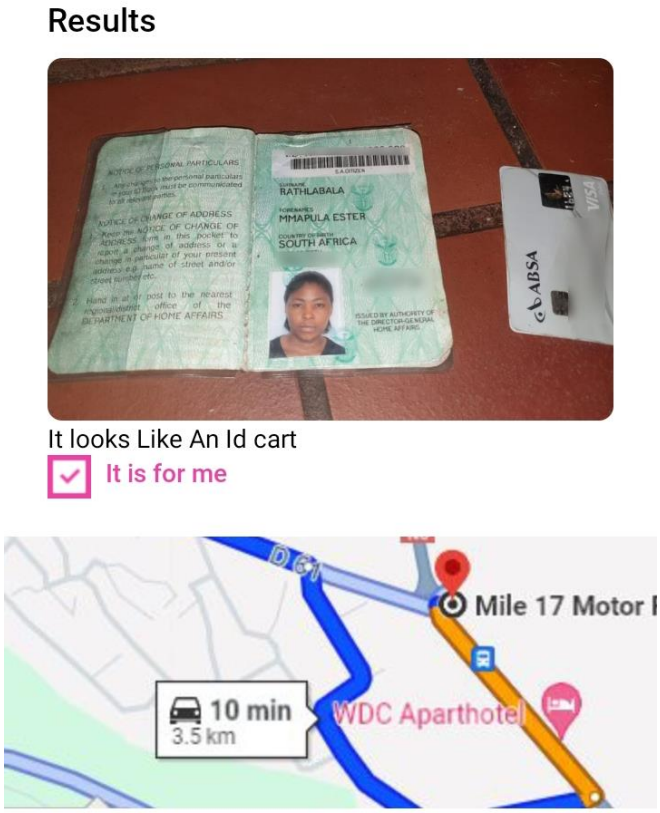


Figure 9: Home Screen

APLICATION IMPLEMENTATION

The implementation process for the "FindIt" application involved translating the design concepts into a functional and interactive mobile application. This technical journey required the integration of modern tools and technologies, along with a meticulous focus on bringing the design elements to life. Below is a detailed report on the implementation process:

Setup and Tooling

Development Environment

The development environment was established using the following tools and technologies:

- **React Native**: A popular framework for building cross-platform mobile applications using JavaScript and React.
- **Expo**: A powerful set of tools and services built around React Native, facilitating the development, building, and deployment processes.
- **Node.js and npm**: Essential for managing dependencies and running the development server.
- **Visual Studio Code**: The preferred code editor, enhanced with extensions for React Native development.

Setting Up The Application

Setting up the "FindIt" application involves several key steps to ensure a smooth and efficient development process. This section will outline the initial setup of the project, including environment configuration, project creation, and initial setup of essential tools and libraries.

Prerequisites

Before starting the setup process, ensure that the following prerequisites are met:

- **Node.js and npm**: Install Node.js and npm, which are essential for running and managing the project dependencies.
- **Expo CLI**: Install Expo CLI globally using npm by running the command.

  npm install -g expo-cli

Creating a New Expo Projec

To create a new Expo project, we followed these steps:

Initialize the Project: Open your terminal and run the following command to create a new Expo project:

expo init FindIt

Navigate to the Project Directory:

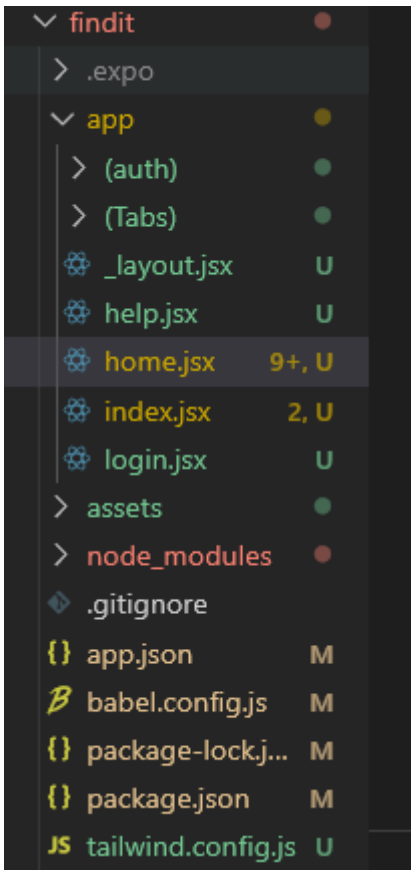
cd FindIt

Installing Dependencies

Install the necessary dependencies for the project. Here are the key libraries and tools used in the "FindIt" application:

- **React Navigation**: For navigation and routing.
- **NativeWind**: For styling the components.
- **React Native Safe Area Context**: For handling safe areas on devices with notches.
- **React Native Screens**: For optimizing navigation performance.
- **Expo Modules**: Various Expo modules for additional functionalities.

Setting Up Project Structure



Why We Chose React Native For The implementation

1. Cross-Platform Development

React Native enables the development of applications that run on both iOS and Android platforms from a single codebase. This significantly reduces development time and costs, ensuring that both platforms receive updates and new features simultaneously.

2. Efficient Development Process

React Native allows for a more efficient development process due to its hot-reloading feature. Developers can see the results of their code changes in real time without needing to recompile the entire application. This rapid feedback loop enhances productivity and accelerates the debugging process.

3. Rich Ecosystem and Community Support

React Native has a vast ecosystem of libraries and tools, which accelerates the development of common features. Additionally, its strong community support means that developers can find solutions to issues quickly and access a wealth of resources and tutorials.

4. Seamless Integration with Native Code

React Native provides the flexibility to integrate native code when necessary. This ensures that the application can leverage platform-specific features and performance optimizations, which is crucial for creating a high-quality user experience.

5. Performance

While React Native uses JavaScript, it renders using native components, which results in better performance compared to hybrid frameworks that use WebView. This balance between JavaScript and native code ensures a smoother and more responsive user experience.

6. Maintainability

Using a single codebase for both platforms simplifies maintenance and updates. Bug fixes, new features, and performance improvements can be implemented once and deployed across both iOS and Android, ensuring consistency and reducing the potential for platform-specific issues.

7. Cost-Effective

By allowing a single team of developers to build and maintain applications for both iOS and Android, React Native reduces the overall development cost. There's no need to hire separate teams for each platform, which is particularly beneficial for startups and small to medium-sized enterprises.

8. Strong Backing by Facebook and Major Companies

React Native is backed by Meta (formerly Facebook), which ensures ongoing development and support for the framework. Additionally, many major companies, including Instagram, Airbnb, and Tesla, have successfully used React Native for their mobile applications, showcasing its robustness and scalability.

9. Component-Based Architecture

React Native's component-based architecture allows for reusable components, which can be beneficial for a modular and maintainable codebase. This approach aligns with modern software engineering practices and enhances the reusability and readability of the code.

Choosing React Native for the implementation of the "FindIt" application provided numerous benefits, from cross-platform compatibility and efficient development processes to strong community support and maintainability. This choice ensured that the development team could deliver a high-quality, per formant, and cost-effective solution that meets the needs of users on both iOS and Android platforms.

What is Expo??

Expo is an open-source platform designed to enhance the development process of React Native applications. It provides a suite of tools and services that streamline the creation, deployment, and maintenance of cross-platform mobile apps. Expo abstracts many complexities of native mobile development, allowing developers to focus on building feature-rich applications without getting bogged down by the intricate details of native code.

Why We Chose to Use Expo?

Simplified Setup and Development

Expo offers a streamlined setup process that simplifies the initial stages of mobile app development. By using Expo CLI, developers can quickly create new projects with a standardized structure and pre-configured settings. This ease of setup reduces the barrier to entry and allows the development team to start coding almost immediately.

2. Comprehensive Documentation and Support

Expo provides extensive and well-maintained documentation, which is invaluable for both beginners and experienced developers. The comprehensive guides, tutorials, and examples help developers understand how to utilize Expo’s features effectively. Additionally, Expo has a supportive community and active forums where developers can seek help and share knowledge.

3. Built-In Libraries and Components

Expo includes a rich set of pre-built libraries and components that cover a wide range of common functionalities, such as camera access, geolocation, push notifications, and more.

These built-in modules save developers time and effort, as they do not need to implement these features from scratch or integrate third-party libraries.

4. Over-the-Air Updates

One of the standout features of Expo is its support for over-the-air (OTA) updates. This allows developers to push updates and bug fixes directly to users without requiring them to download a new version from the app store. OTA updates streamline the deployment process and ensure that users always have the latest version of the app with minimal friction.

5. Cross-Platform Consistency

Expo ensures that the application behaves consistently across both iOS and Android platforms. This cross-platform consistency is crucial for delivering a unified user experience and reduces the need for platform-specific code, thereby simplifying the development and maintenance processes.

6. Development and Debugging Tools

Expo provides robust development and debugging tools that enhance the development experience. The Expo Go app allows developers to preview their application in real time on actual devices, making it easier to test and debug. Additionally, the Expo DevTools offer a web-based interface for managing and inspecting the app during development.

7. Integration with React Native

Expo seamlessly integrates with React Native, leveraging its powerful component-based architecture while adding additional layers of abstraction and convenience. This integration allows developers to use all the standard React Native components and APIs, while also benefiting from Expo’s enhancements.

Expo was chosen for the development of the "FindIt" application due to its ability to simplify and accelerate the development process. With its comprehensive set of tools, built-in libraries, and robust support system, Expo enables the development team to focus on building a high-quality user interface and user experience. The ability to push over-the-air updates and ensure cross-platform consistency further solidifies Expo as an ideal choice for this project.

Implementing Core Features

Navigation Bar

A custom navigation bar was created to enhance user experience:

- **Links**: Utilized the Link component for seamless navigation between screens.
- **Icons**: Integrated local icons and images, styled to be visually consistent and intuitive.
- **Responsiveness**: Ensured the navigation bar was responsive and functioned well on different screen sizes.

Hero Section

The hero section was designed to be visually striking:

- **Image Background**: Used ImageBackground to display a background image, styled with NativeWind.
- **Text Overlay**: Positioned text elements over the background image with appropriate styling for readability.

User Reviews Section

A dynamic user reviews section was implemented to display user feedback:

- **FlatList**: Employed FlatList for rendering a list of reviews horizontally.
- **Review Cards**: Created custom review cards to display user feedback, styled for clarity and visual appeal.

Screens List

- Login
- Register
- Help/FAQ
- Landing Sreen
- Home Screen
- Profile Screen.
- Tracking Screen

CONCLUSION

The "FindIt" application exemplifies how thoughtful design and strategic technology choices can create an effective solution to a common problem. By combining a user-centered design approach with robust implementation techniques, we have developed a platform that is not only functional but also delightful to use. This project stands as a testament to the power of modern mobile development frameworks and the importance of a well-structured design and implementation process.

APPENDICES

Appendix A: Design Mockups

1. **Welcome Screen:**
    - Contains the app logo, a navigation bar, hero section, and an introductory text.
    - Background image to create an engaging visual appeal.
    - User Reviews section for displaying user feedback in a carousel.
2. **Navigation Bar:**
    - Contains icons for Home, Login, and Menu.
    - Positioned at the top for easy access.
3. **User Reviews Carousel:**
    - Displays user reviews in a horizontal scrollable format.
    - Includes star ratings and user comments.

Appendix B: Code Snippets

Welcome Screen:

```
import { StatusBar } from 'expo-status-bar';
import { Text, View, ScrollView,ImageBackground, Image, Pressable,
TouchableOpacity } from 'react-native';
import { SafeAreaView } from 'react-native-safe-area-context';
import { Link } from 'expo-router';
import { Ionicons } from '@expo/vector-icons';
import { FlatList } from 'react-native'
import Swiper from 'react-native-swiper'
const reviews = [
    {
      name: 'Jean paul',
      review: 'Great app for finding missing objects',
      rating: 5,
      image: require('../assets/images/profile.jpg'), // replace with your
image path
    },
    {
      name: 'John peter',
      review: 'Very helpful, found item quickly',
      rating: 4,
      image: require('../assets/images/profile.jpg'), // replace with your
image path
    },
    {
        name: 'kouete laurant',
```

```
        review: 'Very helpful, found item quickly',
        rating: 4,
        image: require('../assets/images/profile.jpg'), // replace with your
image path
      },
    // Add more reviews as needed
  ];

  const renderReviewItem = ({ item }) => (
    <View className="flex-row items-center border-b border-gray-300 py-4 px-
2">
      <Image source={item.image} className="w-12 h-12 rounded-full mr-4" />
      <View className="flex-1">
        <Text className="font-semibold text-lg">{item.name}</Text>
        <Text>{item.review}</Text>
      </View>
      <View className="flex-row items-center">
        {[...Array(item.rating)].map((_, index) => (
          <Text key={index} className="text-yellow-400 text-xl">★</Text>
        ))}
      </View>
    </View>
  );

export default function App() {
    const features = [
        {
          title: 'Discover',
          description1: 'Explore the features of Lost & Found app',
          description2: 'Find lost items with ease using our advanced AI
technology.',
          image: require('../assets/images/search.jpg'), // replace with the
actual path to your image
        },
        {
          title: 'Contribute',
          description1: 'Help others find their missing objects',
          description2: 'Report found objects to reunite them with their
owners.',
          image: require('../assets/images/contribute-title.png'), // replace
with the actual path to your image
        },
        {
          title: 'Track',
          description1: 'Real-time tracking and location updates',
          description2: 'Track the progress of found objects.',
          image: require('../assets/icons/globe.png'), // replace with the
actual path to your image
```

```
        },
      ];
  return (
    <SafeAreaView className="flex-1 bg-white">
        {/* Navigation Bar */}
        <View className="flex-row justify-between items-center py-0 px-4
border-b border-gray-200" style={{ height: 60, width: '100%', backgroundColor:
'#ffffff' }}>
            {/* Logo */}
            <View className="flex items-center justify-center h-13 w-12 p-1
rounded-full">
                <Image
                    source={require('../assets/images/logo.png')}
                    className="rounded-full"
                    style={{ width: '90%', height: '90%', resizeMode: 'cover'
}}
                />
            </View>

            {/* Menu Icons */}
            <View className="flex-row gap-x-4 items-center p-1 rounded-lg">
                <View className="p-0 hover:bg-gray-200 transition-colors"
style={{height:30}}>
                    <Link href="/home">
                        <Image
                            source={require('../assets/icons/ic_baseline-
home.png')}
                            style={{ width: 20, height: 20,resizeMode:
'contain' }} // Adjust size as needed
                        />
                    </Link>
                </View>
                <View className="p-0 hover:bg-gray-200 transition-colors"
style={{height:30}}>
                    <Link href="/login" className="flex justify-center items-
center p-2 bg-primary h-8 rounded-md hover:bg-gray-200 transition-colors">
                        <Image
                            source={require('../assets/icons/profile_white.png
')}
                            style={{ width: 24, height: 24,resizeMode:
'contain' }} // Adjust size as needed
                        />
                        <Text className="font-bold text-xs text-bgsecondary
ml-1">Login</Text>
                    </Link>
                </View>
                <View className="p-0   hover:bg-gray-200 transition-colors"
style={{height:30}}>
```

```
                    <Link href="/menu">
                        <Image
                        className="mt-0"
                            source={require('../assets/icons/ic_baseline-
menu.png')}
                            style={{ width: 20, height: 20,resizeMode:
'contain', marginTop:0 }} // Adjust size as needed
                        />
                    </Link>
                </View>
            </View>
        </View>
        <ScrollView className="flex-col"
contentContainerStyle={{alignItems:'center'}}>
            {/* Hero Section */}
            <ImageBackground
            source={require('../assets/images/hero_image.jpg')} // Replace
with your local image path or URL
            className="w-full h-64 items-center justify-center"
            style={{ width: '100%', height: 300 }} // Ensure the height covers
the hero section
            imageStyle={{ resizeMode: 'cover' }}>
                <View className="bg-black absolute  opacity-40 w-full h-full">
                </View>
                <Text className="text-4xl text-primary font-bold font-
Poppins">
                    Welcome to Find <Text className="font-extrabold">I</Text>t
                </Text>
                <Text className="text-lg text-bgsecondary mt-2 font-
rRougeScript">
                    Helping you find what is lost
                </Text>
            </ImageBackground>
            <View className="flex-row gap-2 mt-0 ml-0 mr-3  py-3 px-4 justify-
evenly items-center" style={{width:'100%',height:130}}>
                {/* Home Tab */}
                <Pressable className=" flex-1 items-center border-2 rounded-md
border-gray-300" style={{height:'100%'}}>
                    <Text className="text-blue-500 text-4xl p-2">🔍</Text>
                    <Text className="text-sm">Search</Text>
                </Pressable>

                {/* Report Tab */}
                <Pressable className="items-center flex-1 border-2 rounded-md
border-gray-300" style={{height:'100%'}}>
                    <Text className="text-blue-500 text-4xl p-2">📷</Text>
                    <Text className="text-sm">Report</Text>
                </Pressable>
```

```
                {/* Track Tab */}
                <Pressable className="items-center flex-1 border-2 border-
gray-300 rounded-md" style={{height:'100%'}}>
                    <Text className="text-blue-500 text-4xl p-2">🌎</Text>
                    <Text className="text-sm">Track</Text>
                </Pressable>
            </View>
            <View className="flex-1 mt-6 w-full items-center justify-center
bg-white px-4">
                {/* Get Started Button */}
                <TouchableOpacity className="w-full  bg-primary py-2 rounded-
lg items-center">
                    <Link href="/home">
                     <Text className="text-white text-[#ffffff] text-lg font-
semibold">Get Started</Text>
                    </Link>
                </TouchableOpacity>

                {/* Or Text */}
                <Text className="text-gray-400 my-2 font-rRougeScript text-
2xl">Or</Text>

                {/* Get Help Link */}
                <Link href="/help" className="text-primary text-lg font-
semibold">
                    Get help
                </Link>
            </View>

            <View className="p-5 w-full">
            <Text className="text-2xl font-bold mb-4">More About FindIt</Text>
                {features.map((feature, index) => (
                <View key={index} className="flex-row text-sm items-center mb-
4 border-b border-gray-300 pb-4">
                    <Image
                    source={feature.image}
                    className="w-16 h-16 mr-4 rounded"
                    style={{ resizeMode: 'cover' }}
                    />
                    <View className="flex-1">
                    <Text className="text-lg font-bold">{feature.title}</Text>
                    <Text className="text-gray-600 text-
xs">{feature.description1}</Text>
                    <Text className=" text-text mt-
1">{feature.description2}</Text>
                    </View>
                </View>
```

```
                ))}
            </View>

            {/* User Reviews Section */}
            <View className="flex-1" style={{height:310}}>
            <View className="pl-5">
                <Text className="font-bold text-2xl">User Reviews</Text>
            </View>
            <Swiper
                style={{height:300}}
                showsPagination={false} // Disable default pagination dots
                horizontal={true} // Ensure horizontal scrolling
                showsButtons={true} // Enable navigation buttons
                nextButton={<Ionicons name="arrow-forward" size={24}
color="black" />}
                prevButton={<Ionicons name="arrow-back" size={24}
color="black" />} >
                {reviews.map((review, index) => (
                    <View key={index} className="flex-1 justify-center items-
center" style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
                        <View className="py-7 px-4 rounded-lg bg-bgsecondary">
                            <View className="flex-row justify-between items-
center">
                                <View className="flex-row justify-start gap-x-
2 items-center">
                                    <Image className="rounded-full"
source={review.image} style={{ width: 40, height: 40 }} />
                                    <Text className="text-
secondary">{review.name}</Text>
                                </View>
                                <View style={{ flexDirection: 'row' }}>
                                {[...Array(review.rating)].map((_, index) => (
                                <Ionicons key={index} name="star" size={10}
color="gold" />

                                ))}
                                </View>
                            </View>
                            <View className="mt-4">
                                <Text>{review.review}</Text>
                            </View>

                        </View>
                    </View>
                ))}
            </Swiper>
            </View>

            {/* Footer */}
```

```
            <View className="py-4 bg-primary w-full items-center">
            <Text className="text-white text-center text-base font-Roboto-
Regular">
                © 2024 Find-IT. All rights reserved.
            </Text>
            </View>
        </ScrollView>
      <StatusBar style="auto" />
    </SafeAreaView>
  );
}
```

GLOSSARY

- **UI/UX Design:** User Interface (UI) and User Experience (UX) design are critical components of software development focused on the appearance and functionality of the application, ensuring it is user-friendly and intuitive.
- **Figma:** A cloud-based design tool used for creating user interface designs, wireframes, prototypes, and more. It allows real-time collaboration and has become a staple in modern design workflows.
- **React Native:** A popular framework for building mobile applications using JavaScript and React. It allows for the development of cross-platform apps with native performance.
- **Expo:** A framework and platform for universal React applications. It provides a set of tools and services for building, deploying, and quickly iterating on apps for iOS, Android, and the web.
- **NativeWind:** A utility-first CSS framework for React Native that helps in styling applications efficiently.
- **Wireframe:** A basic visual guide used in the early stages of designing user interfaces. It represents the skeletal framework of a digital interface.
- **Prototype:** An early sample, model, or release of a product built to test a concept or process. It is used to evaluate design ideas before final implementation.
- **Carousel:** A UI component that allows users to browse through a collection of items (such as images or reviews) by swiping horizontally.

REFERENCES

https://docs.expo.dev/versions/latest/sdk/navigation-bar/#navigationbarbuttonstyle

https://www.ibm.com/design/thinking/page/toolkit

https://www.interaction-design.org/literature/topics/design-principles