

COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD CAMPUS

**Mobile Application Development semester project**

**Documentation for Clean cloths- Laundry App**

**Laiba Binta Tahir FA21-BSE-019-5A**

**Arfah Ali FA21-BSE-080-5A**

Submitted To:

**Sir Jawad Khan**

Date: 29 december-2023

Contents

[1. Introduction 3](#_Toc154762551)

[1.1 Purpose and Overview 3](#_Toc154762552)

[1.2 Target Audience 3](#_Toc154762553)

[2. Technology Stack 3](#_Toc154762554)

[2.1 Frontend 3](#_Toc154762555)

[2.2 Backend 3](#_Toc154762556)

[2.3 Additional Libraries/Tools 3](#_Toc154762557)

[3. Authentication Module 3](#_Toc154762558)

[3.1 Description 3](#_Toc154762559)

[3.2 Implementation 3](#_Toc154762560)

[3.3 Key Features 3](#_Toc154762561)

[4. Home Screen 4](#_Toc154762562)

[4.1 Description 4](#_Toc154762563)

[4.2 Navigation 4](#_Toc154762564)

[4.3 Screenshots 5](#_Toc154762565)

[5. Services Module 5](#_Toc154762566)

[5.1 Description 5](#_Toc154762567)

[5.2 Service Types 5](#_Toc154762568)

[5.3 Interfaces 5](#_Toc154762569)

[6. Service Details 6](#_Toc154762570)

[6.1 Description 6](#_Toc154762571)

[6.3 Visuals 7](#_Toc154762572)

[7. Scheduling and Checkout Module 7](#_Toc154762573)

[7.1 Description 7](#_Toc154762574)

[7.3 Screenshot 7](#_Toc154762575)

[8. User Profile Module 8](#_Toc154762576)

[8.1 Description 8](#_Toc154762577)

[8.2 Features 8](#_Toc154762578)

[8.3 Connection to Orders 9](#_Toc154762579)

[8. FAQs Module 9](#_Toc154762580)

[8.1 Description 9](#_Toc154762581)

[8.2 User Flow 9](#_Toc154762582)

[8.3 Sample FAQ Screenshots 10](#_Toc154762583)

[9. Conclusion 10](#_Toc154762584)

[10. Code Snippets 10](#_Toc154762585)

[10.1 app.js 10](#_Toc154762586)

[10.2 Login.js 11](#_Toc154762587)

[10.3 Reister.js 13](#_Toc154762588)

[10.4 home.js 16](#_Toc154762589)

[10.5 faq screen 18](#_Toc154762590)

[10.6 profile screen 20](#_Toc154762591)

[10.7 checkout screen.js 23](#_Toc154762592)

[10.8 services screen.js 25](#_Toc154762593)

[10.9 detailed services 26](#_Toc154762594)

[10.10 schedule pickup 28](#_Toc154762595)

[10.11 Success screen 30](#_Toc154762596)

# 1. Introduction

## 1.1 Purpose and Overview

Clean cloths is a mobile application designed to streamline and simplify laundry management for users. With a user-friendly interface and powerful features, it aims to address common challenges associated with traditional laundry services.

## 1.2 Target Audience

The app is tailored for busy professionals, families, and students who seek a convenient and efficient solution for their laundry needs.

# 2. Technology Stack

## 2.1 Frontend

* **Framework:** React Native
* **State Management:** Context API

## 2.2 Backend

* **Database:** Firebase Firestore
* **Authentication:** Firebase Authentication

## 2.3 Additional Libraries/Tools

* React Native Paper
* React Native Gesture Handler
* React Native Reanimated
* React Native Safe Area Context
* Async Storage

# 3. Authentication Module

## 3.1 Description

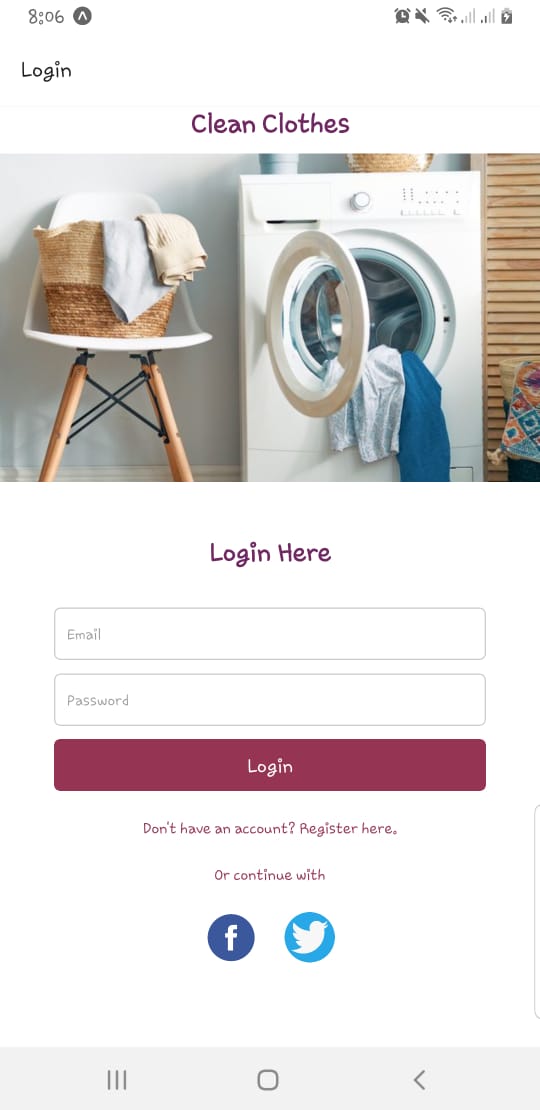
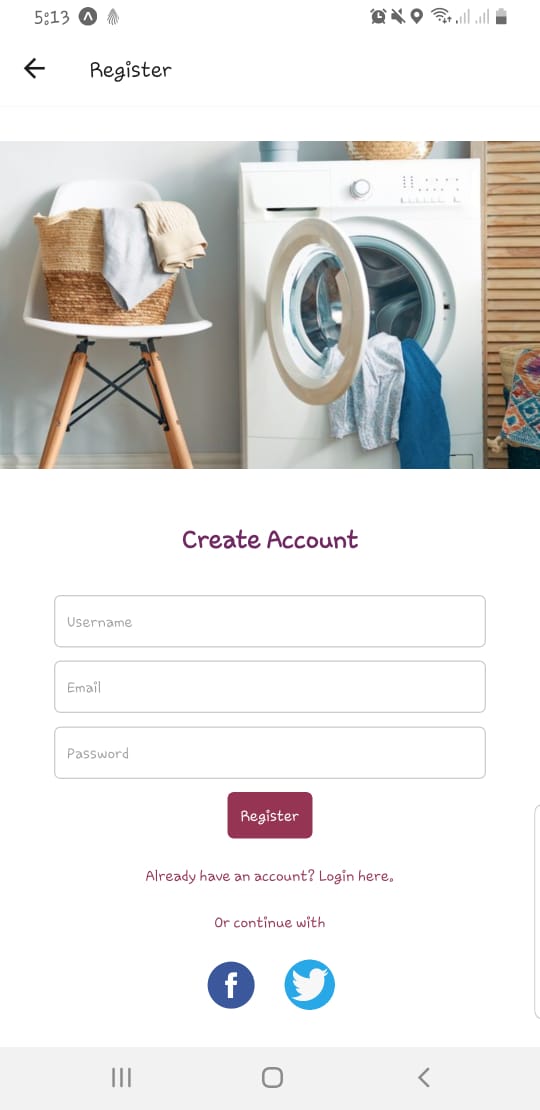
The authentication module ensures a secure and personalized experience for users. Leveraging the Context API, the app manages user authentication seamlessly.

## 3.2 Implementation

Explain the implementation details of user authentication, including the flow and security measures in place.

## 3.3 Key Features

* Secure user authentication.
* Integration with Firebase Authentication for enhanced security.

# 4. Home Screen

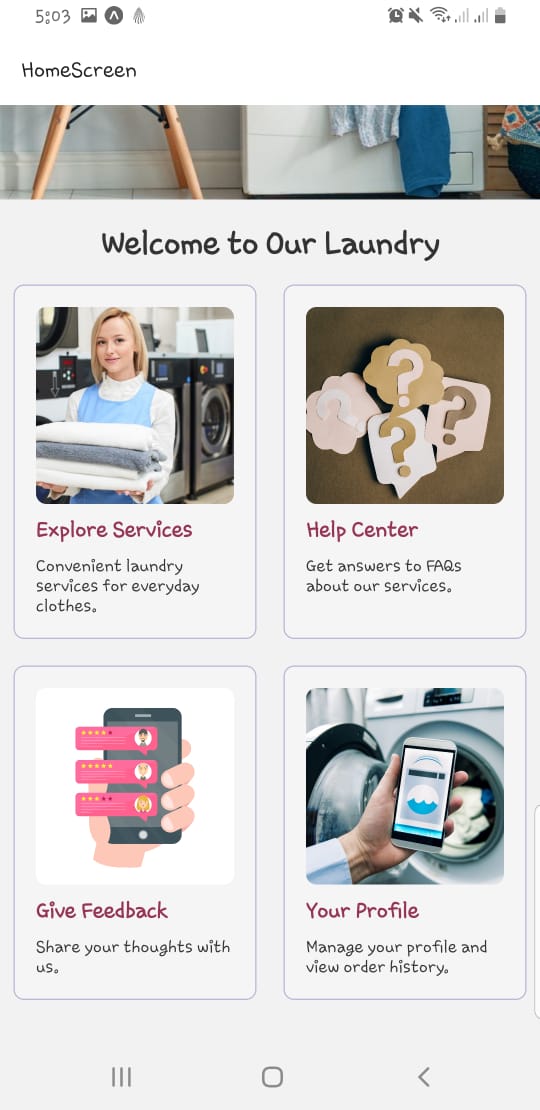
## 4.1 Description

The home screen serves as the central hub for navigation, providing quick access to essential features.

## 4.2 Navigation

* FAQs: Frequently Asked Questions
* Services: Explore and schedule laundry services and schedule any service
* Feedback: Provide feedback on services
* Profile: User account and settings

## 4.3 Screenshots



# 5. Services Module

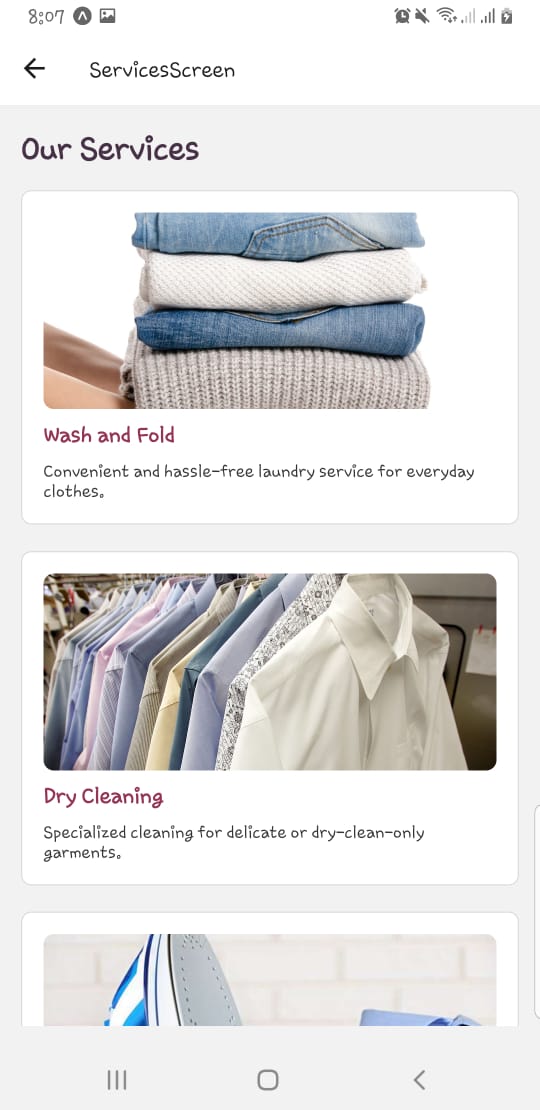
## 5.1 Description

The services module introduces users to the available laundry services, offering a variety of options to suit different needs.

## 5.2 Service Types

Dry Cleaning, Ironing, wash and fold

## 5.3 Interfaces

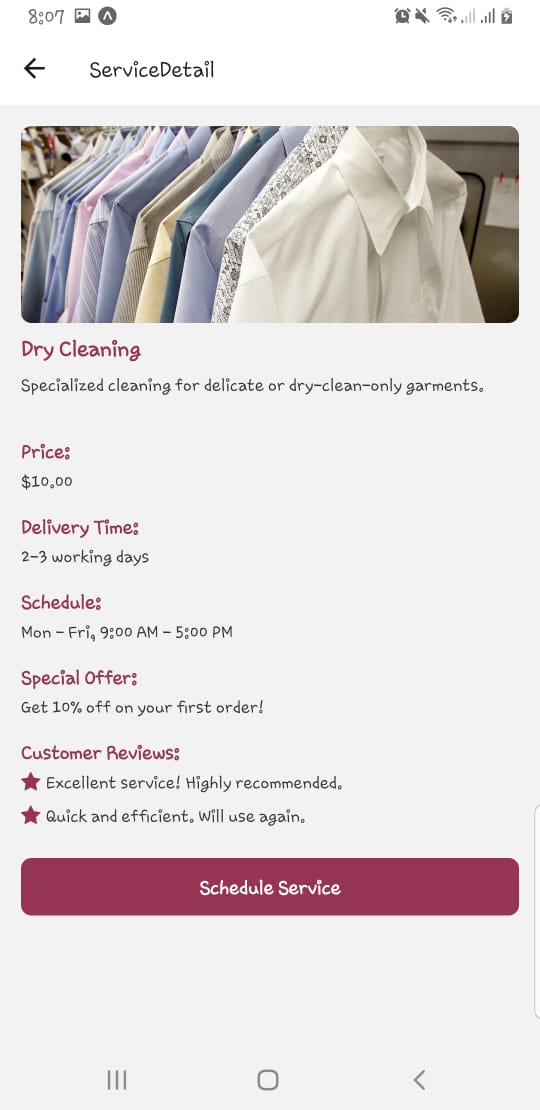


# 6. Service Details

## 6.1 Description

When a user selects a service, detailed information is presented, highlighting the unique features and benefits of each option.

## 6.3 Visuals

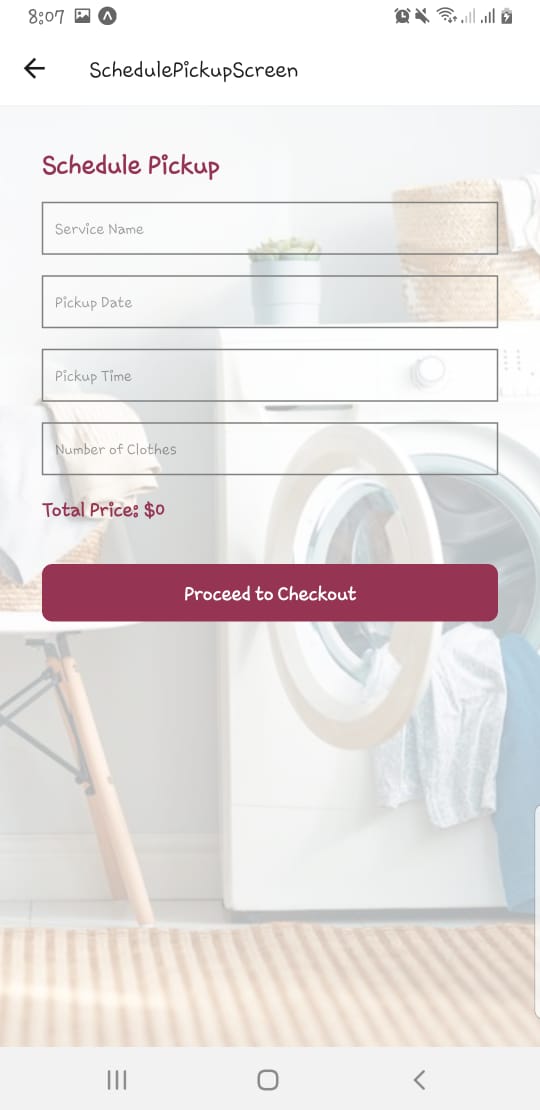
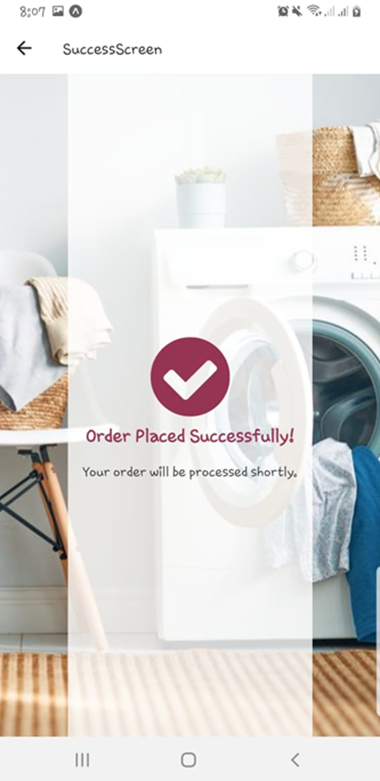


# 7. Scheduling and Checkout Module

## 7.1 Description

The scheduling and checkout module simplifies the process of booking a service and proceeding to checkout.

## 7.3 Screenshot

# 8. User Profile Module

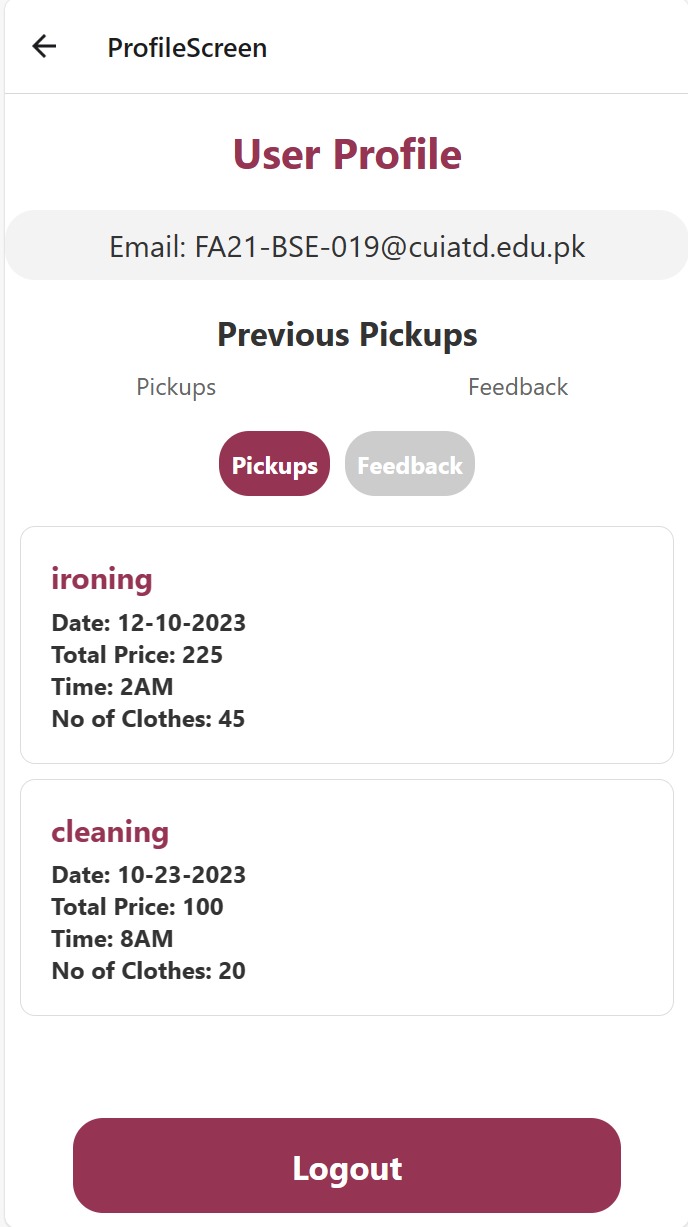
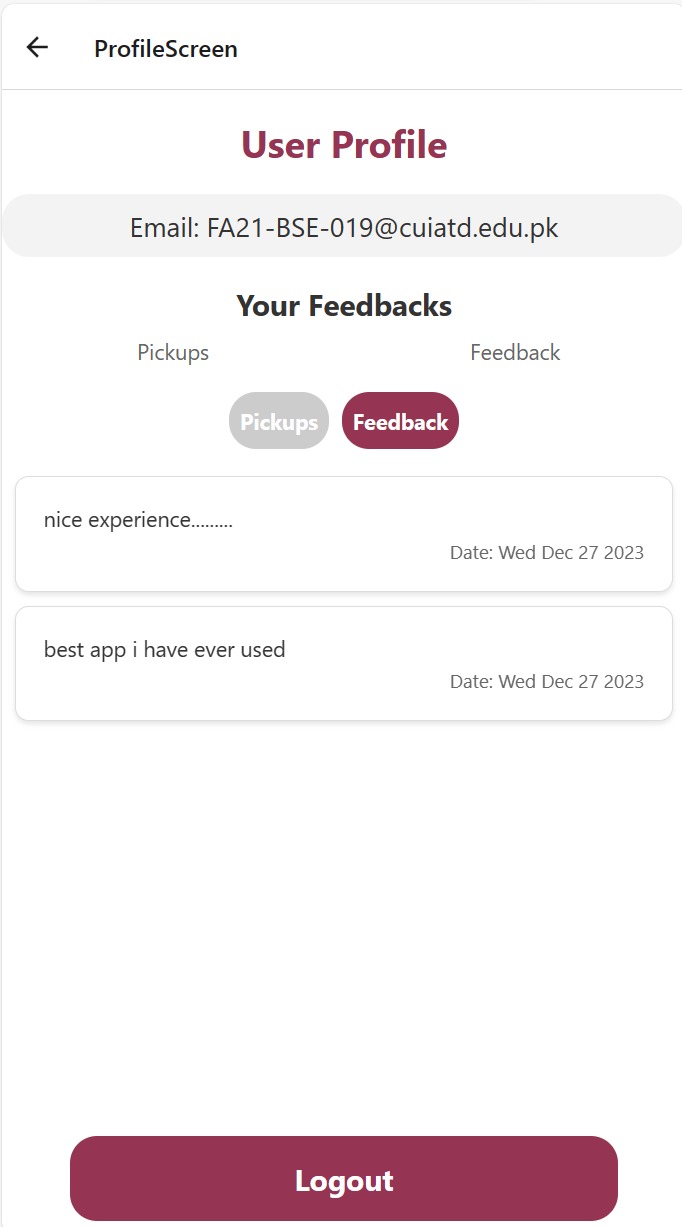
## 8.1 Description

The user profile module allows users to customize their experience and view order history.

## 8.2 Features

Highlight features such as personalization options, order history, and feedbacks made.

## 8.3 Connection to Orders

# 8. FAQs Module

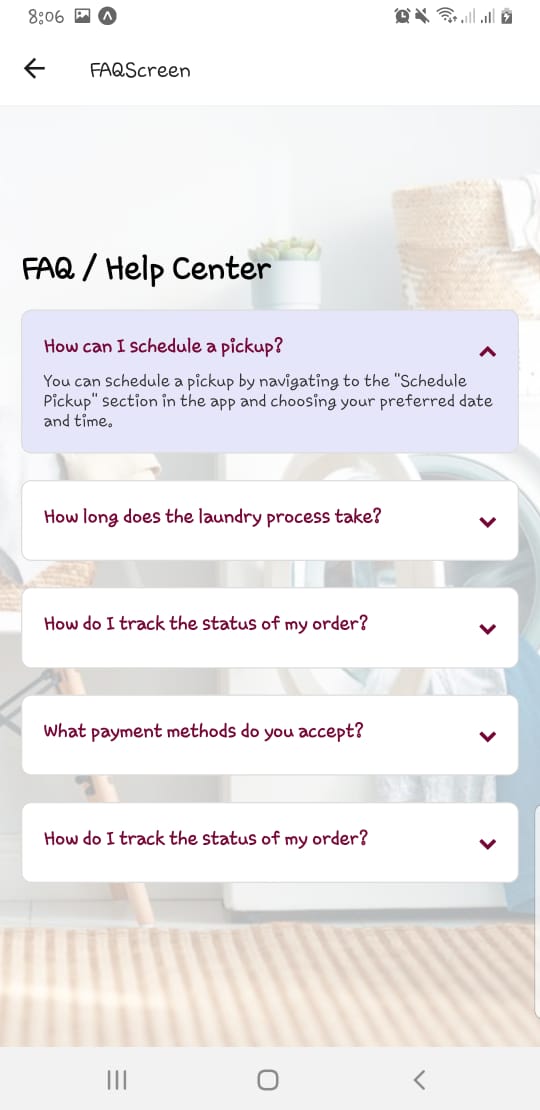
## 8.1 Description

The FAQs module serves as a comprehensive resource to address common queries and concerns that users may have about the app, its services, and functionalities.

## 8.2 User Flow

Provide users with an easy-to-follow guide on how to access and utilize the FAQs module. Include steps on navigating through the questions and finding relevant answers.

## 8.3 Sample FAQ Screenshots



# 9. Conclusion

In conclusion, clean cloths redefines the laundry experience by seamlessly blending cutting-edge technology with user-friendly design. From the convenience of scheduling services to the personalized touch of user profiles, our app is crafted to simplify and elevate the way you manage laundry.

# 10. Code Snippets

## 10.1 app.js

// App.js

import React from 'react';

import RootNavigation from './src/navigation';

import { AuthProvider } from './contexts/AuthContext';

const App = () => {

    return(

    <AuthProvider>

    <RootNavigation />

    </AuthProvider>

    );

};

export default App;

## 10.2 Login.js

import React, { useState } from 'react';

import { View, TextInput, TouchableOpacity, StyleSheet, Text, Image, ActivityIndicator } from 'react-native';

import useAuth from '../hooks/useAuth';

import { Alert } from 'react-native';

const LoginComponent = ({ navigation }) => {

  const [formData, setFormData] = useState({ email: '', password: '' });

  const [errors, setErrors] = useState({});

  const { Login, loading } = useAuth(); // useAuth hook

  const validateForm = () => {

    let newErrors = {};

    if (!formData.email.includes('@') || formData.email.length < 3) {

      newErrors.email = 'Please enter a valid email';

    }

    setErrors(newErrors);

    return Object.keys(newErrors).length === 0;

  };

  const handleLogin = async () => {

    if (validateForm()) {

      try {

        await Login(formData.email, formData.password); // Attempt to login

      } catch (error) {

        console.log(error.message); // Log the error message

        Alert.alert('Incorrect credentials'); // Display an alert with the error

      }

    }

  };

  return (

    <View style={styles.container}>

      <Text style={styles.appName}>Clean Clothes</Text>

      <Image source={require('../../assets/back.jpg')} style={styles.logo} />

      <Text style={styles.companyName}>Login Here</Text>

      <TextInput

        style={styles.input}

        placeholder="Email"

        keyboardType="email-address"

        onChangeText={(text) => setFormData({ ...formData, email: text })}

        value={formData.email}

      />

      {errors.email && <Text style={styles.errorText}>{errors.email}</Text>}

      <TextInput

        style={styles.input}

        placeholder="Password"

        secureTextEntry={true}

        onChangeText={(text) => setFormData({ ...formData, password: text })}

        value={formData.password}

      />

      <TouchableOpacity style={styles.loginButton} onPress={handleLogin}>

        {loading ? (

          <ActivityIndicator size="small" color="#fff" />

        ) : (

          <Text style={styles.loginButtonText}>Login</Text>

        )}

      </TouchableOpacity>

      <Text style={styles.switchText} onPress={() => navigation.navigate('Register')}>

        Don't have an account? Register here.

      </Text>

      <Text style={styles.switchText}>Or continue with</Text>

      <View style={styles.socialButtonsContainer}>

        <TouchableOpacity style={styles.socialButton}>

          <Image source={require('../../assets/facebook.png')} style={styles.socialIcon} />

        </TouchableOpacity>

        <TouchableOpacity style={styles.socialButton}>

          <Image source={require('../../assets/twitter.png')} style={styles.socialIcon} />

        </TouchableOpacity>

      </View>

    </View>

  );

};

export default LoginComponent;

## 10.3 Reister.js

import React, { useState, useContext } from "react";

import {

  View,

  TextInput,

  TouchableOpacity,

  StyleSheet,

  Text,

  Image,

  ActivityIndicator,

} from "react-native";

import { AuthContext } from "../../contexts/AuthContext";

import useFirestore from "../hooks/useFirestore";

import useAuth from "../hooks/useAuth";

const RegisterComponent = ({ navigation }) => {

  const [formData, setFormData] = useState({

    username: "",

    email: "",

    password: "",

  });

  const [errors, setErrors] = useState({});

  const { signUp } = useAuth();

  const { emailExists, userExsists, loading } = useFirestore();

  const validateForm = async () => {

    let newErrors = {};

    const emailRegex = /^[a-zA-Z0-9.\_-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;

    if (!emailRegex.test(formData.email)) {

      newErrors.email = "Please enter a valid email address";

    } else if (await emailExists(formData.email)) {

      newErrors.email = "Email already in use";

    }

    if (formData.password.length < 6) {

      newErrors.password = "Password must be at least 6 characters long";

    }

    if (await userExsists(formData.username)) {

      newErrors.username = "Username already taken";

    }

    setErrors(newErrors);

    return Object.keys(newErrors).length === 0;

  };

  const handleRegister = async () => {

    if (await validateForm()) {

      // Additional user data to be saved in Firestore

      const additionalUserData = {

        username: formData.username,

        email: formData.email,

        // Any other user data fields you want to store

      };

      const result = await signUp(

        formData.email,

        formData.password,

        additionalUserData

      );

      if (result.success) {

        console.log("Registration successful");

        navigation.navigate("Login"); // Replace with your success route

      } else {

        setErrors({ ...errors, general: result.message });

      }

    }

  };

  return (

    <View style={styles.container}>

      <Image source={require("../../assets/back.jpg")} style={styles.logo} />

      <Text style={styles.companyName}>Create Account</Text>

      {errors.general && <Text style={styles.errorText}>{errors.general}</Text>}

      {/\* Username Input \*/}

      <TextInput

        style={styles.input}

        placeholder="Username"

        onChangeText={(text) => setFormData({ ...formData, username: text })}

        value={formData.username}

      />

      {errors.username && (

        <Text style={styles.errorText}>{errors.username}</Text>

      )}

      {/\* Email Input \*/}

      <TextInput

        style={styles.input}

        placeholder="Email"

        keyboardType="email-address"

        onChangeText={(text) => setFormData({ ...formData, email: text })}

        value={formData.email}

      />

      {errors.email && <Text style={styles.errorText}>{errors.email}</Text>}

      {/\* Password Input \*/}

      <TextInput

        style={styles.input}

        placeholder="Password"

        secureTextEntry={true}

        onChangeText={(text) => setFormData({ ...formData, password: text })}

        value={formData.password}

      />

      {errors.password && (

        <Text style={styles.errorText}>{errors.password}</Text>

      )}

      <TouchableOpacity style={styles.signupButton} onPress={handleRegister}>

        {loading ? (

          <ActivityIndicator size="small" color="#fff" />

        ) : (

          <Text style={styles.registerButton}>Register</Text>

        )}

      </TouchableOpacity>

      <Text

        style={styles.switchText}

        onPress={() => navigation.navigate("Login")}

      >

        Already have an account? Login here.

      </Text>

      <Text style={styles.switchText}>Or continue with</Text>

      <View style={styles.socialButtonsContainer}>

        <TouchableOpacity style={styles.socialButton}>

          <Image

            source={require("../../assets/facebook.png")}

            style={styles.socialIcon}

          />

        </TouchableOpacity>

        <TouchableOpacity style={styles.socialButton}>

          <Image

            source={require("../../assets/twitter.png")}

            style={styles.socialIcon}

          />

        </TouchableOpacity>

      </View>

    </View>

  );

}}

export default RegisterComponent

## 10.4 home.js

import React, { useState } from "react";

import { View, Text, Image, StyleSheet, TouchableOpacity, TouchableHighlight, ScrollView } from "react-native";

const HomeScreen = ({ navigation }) => {

  const [hoveredBox, setHoveredBox] = useState(null);

  const handleNavigate = (screen) => {

    navigation.navigate(screen);

  };

  const boxesData = [

    {

      id: '1',

      name: 'Explore Services',

      image: require('../../assets/Services.jpg'),

      screen: 'ServicesScreen',

      description: 'Convenient laundry services for everyday clothes.',

    },

    {

      id: '2',

      name: 'Help Center',

      image: require('../../assets/faqs.webp'),

      screen: 'FAQScreen',

      description: 'Get answers to FAQs about our services.',

    },

    {

      id: '3',

      name: 'Give Feedback',

      image: require('../../assets/feedback\_icon.png'),

      screen: 'FeedbackScreen',

      description: 'Share your thoughts with us.',

    },

    {

      id: '4',

      name: 'Your Profile',

      image: require('../../assets/profile\_icon.png'),

      screen: 'ProfileScreen',

      description: 'Manage your profile and view order history.',

    },

  ];

  const navigateToScreen = (screen) => {

    navigation.navigate(screen);

  };

  const renderBox = ({ item }) => (

    <TouchableHighlight

      style={[styles.box, item.id === hoveredBox && styles.hoveredBox]}

      underlayColor="#E6E6FA"

      onPress={() => navigateToScreen(item.screen)}

      onHideUnderlay={() => setHoveredBox(null)}

      onShowUnderlay={() => setHoveredBox(item.id)}

    >

      <>

        <Image source={item.image} style={styles.boxImage} />

        <Text style={styles.boxText}>{item.name}</Text>

        <Text style={styles.Description}>{item.description}</Text>

      </>

    </TouchableHighlight>

  );

  return (

    <ScrollView>

      <View style={styles.container}>

        <Image source={require("../../assets/back.jpg")} style={styles.coverImage} />

        <Text style={styles.welcomeText}>Welcome to Our Laundry</Text>

        <View style={styles.boxContainer}>

          {boxesData.map((item) => renderBox({ item }))}

        </View>

      </View>

    </ScrollView>

  );

};

export default HomeScreen;

## 10.5 faq screen

import React, { useState } from "react";

import {

  View,

  Text,

  ScrollView,

  StyleSheet,

  TouchableWithoutFeedback,

  ImageBackground,

  TouchableOpacity,

  Modal,

} from "react-native";

import Icon from "react-native-vector-icons/FontAwesome";

const FAQScreen = () => {

  const faqData = [

    {

      question: "How can I schedule a pickup?",

      answer:

        'You can schedule a pickup by navigating to the "Schedule Pickup" section in the app and choosing your preferred date and time.',

    },

    {

      question: "How long does the laundry process take?",

      answer:

        "The duration of the laundry process depends on the type of service you choose. We provide estimated processing times for each service in the app. You can also track the progress in the 'Order History' section.",

    },

    {

      question: "How do I track the status of my order?",

      answer:

        'You can track the status of your order in the "Order History" section of the app. We provide real-time updates on the processing of your laundry.',

    },

    {

      question: "What payment methods do you accept?",

      answer:

        "We accept various payment methods, including credit cards and digital wallets. You can securely pay for your services through the app.",

    },

    {

      question: "How do I track the status of my order?",

      answer:

        'You can track the status of your order in the "Order History" section of the app. We provide real-time updates on the processing of your laundry.',

    },

  ];

  const [expandedIndex, setExpandedIndex] = useState(null);

  const [modalVisible, setModalVisible] = useState(false);

  const toggleAnswer = (index) => {

    setExpandedIndex(expandedIndex === index ? null : index);

  };

  return (

    <ImageBackground

      source={require("../../assets/back.jpg")}

      style={styles.backgroundImage}

    >

      <ScrollView contentContainerStyle={styles.container}>

        <Text style={styles.title}>FAQ / Help Center</Text>

        {faqData.map((item, index) => (

          <TouchableWithoutFeedback

            key={index}

            onPress={() => toggleAnswer(index)}

          >

            <View

              style={[

                styles.faqItem,

                expandedIndex === index && styles.faqItemExpanded,

              ]}

            >

              <View style={styles.questionContainer}>

                <Text style={styles.question}>{item.question}</Text>

                <Icon

                  name={expandedIndex === index ? "chevron-up" : "chevron-down"}

                  size={14}

                  color="#770737"

                />

              </View>

              {expandedIndex === index && (

                <Text style={styles.answer}>{item.answer}</Text>

              )}

            </View>

          </TouchableWithoutFeedback>

        ))}

      </ScrollView>

      <Modal

        animationType="slide"

        transparent={true}

        visible={modalVisible}

        onRequestClose={() => {

          setModalVisible(!modalVisible);

        }}

      >

        <View style={styles.overlay}>

          <Text style={styles.overlayText}>Additional Content</Text>

          <TouchableOpacity

            style={styles.closeButton}

            onPress={() => setModalVisible(!modalVisible)}

          >

            <Text style={styles.closeButtonText}>Close</Text>

          </TouchableOpacity>

        </View>

      </Modal>

    </ImageBackground>

  );

};

## 10.6 profile screen

import React, { useState, useEffect } from "react";

import { View, Text, StyleSheet, FlatList } from "react-native";

import useFirestore from "../hooks/useFirestore";

import { Button } from "react-native-elements";

import useAuth from "../hooks/useAuth";

const UserProfileScreen = ({ navigation }) => {

  const [viewMode, setViewMode] = useState("pickups");

  const { currentUser, signOutUser } = useAuth();

  const { getUserPickups } = useFirestore();

  const [previousPickups, setPreviousPickups] = useState([]);

  const { getUserFeedback } = useFirestore();

  const [userFeedback, setUserFeedback] = useState([]);

  useEffect(() => {

    if (currentUser) {

      if (viewMode === "pickups") {

        getUserPickups(currentUser.uid).then((pickups) => {

          setPreviousPickups(pickups);

        });

      } else if (viewMode === "feedback") {

        getUserFeedback(currentUser.uid).then((feedback) => {

          setUserFeedback(feedback);

        });

      }

    }

  }, [currentUser, viewMode, getUserPickups, getUserFeedback]);

  const renderItem = ({ item }) => (

    <View style={styles.orderItem}>

      <Text style={styles.orderService}>{item.serviceName}</Text>

      <Text style={styles.orderTotal}>Date: {item.pickupDate}</Text>

      <Text style={styles.orderTotal}>Total Price: {item.totalPrice}</Text>

      <Text style={styles.orderTotal}>Time: {item.pickupTime}</Text>

      <Text style={styles.orderTotal}>

        No of Clothes: {item.numberOfClothes}

      </Text>

    </View>

  );

  const renderFeedbackItem = ({ item }) => (

    <View style={styles.feedbackItem}>

      <Text style={styles.feedbackText}>{item.feedback}</Text>

      <Text style={styles.feedbackDate}>

        Date: {item.timestamp.toDate().toDateString()}

      </Text>

    </View>

  );

  return (

    <View style={styles.container}>

      <Text style={styles.title}>User Profile</Text>

      {currentUser && (

        <View>

          <Text style={styles.userInfo}>Email: {currentUser.email}</Text>

        </View>

      )}

      <Text style={styles.sectionTitle}>

        {viewMode === "pickups" ? "Previous Pickups" : "Your Feedbacks"}

      </Text>

      <View style={styles.toggleLabels}>

        <Text style={styles.toggleLabel}>Pickups</Text>

        <Text style={styles.toggleLabel}>Feedback</Text>

      </View>

      <View style={styles.toggleButtons}>

        <Button

          title="Pickups"

          onPress={() => setViewMode("pickups")}

          buttonStyle={[

            styles.toggleButton,

            viewMode === "pickups"

              ? styles.toggleButtonSelected

              : styles.toggleButtonUnselected,

          ]}

          titleStyle={styles.toggleButtonText}

        />

        <Button

          title="Feedback"

          onPress={() => setViewMode("feedback")}

          buttonStyle={[

            styles.toggleButton,

            viewMode === "feedback"

              ? styles.toggleButtonSelected

              : styles.toggleButtonUnselected,

          ]}

          titleStyle={styles.toggleButtonText}

        />

      </View>

      {viewMode === "pickups" ? (

        <FlatList

          data={previousPickups}

          keyExtractor={(item) => item.id}

          renderItem={renderItem}

        />

      ) : (

        <FlatList

          data={userFeedback}

          keyExtractor={(item) => item.id}

          renderItem={renderFeedbackItem}

        />

      )}

      <View style={styles.logoutButton}>

        <Button

          title="Logout"

          onPress={signOutUser}

          buttonStyle={styles.logoutButton}

          titleStyle={styles.logoutButtonText}

        />

      </View>

    </View>

  );

};

## 10.7 checkout screen.js

import React, { useState } from 'react';

import { View, Text, StyleSheet, TouchableOpacity, FlatList, Image } from 'react-native';

const CheckoutScreen = ({ route, navigation }) => {

  const { service, pickupDate, pickupTime, numberOfClothes } = route.params;

  const [items, setItems] = useState([]);

  const [totalAmount, setTotalAmount] = useState(0);

  const addItemToCheckout = () => {

    const newItem = {

      id: items.length + 1,

      name: service.name,

      quantity: parseInt(numberOfClothes, 10),

      price: 10.00,

    };

    setItems([...items, newItem]);

    setTotalAmount(totalAmount + newItem.quantity \* newItem.price);

  };

  return (

    <View style={styles.container}>

      <Image

          source={require('../../assets/back.jpg')}

        style={styles.backgroundImage}

        resizeMode="cover"

      />

      <View style={styles.overlay}>

        <Text style={styles.title}>Checkout</Text>

        <FlatList

          data={items}

          keyExtractor={(item) => item.id.toString()}

          renderItem={({ item }) => (

            <View style={styles.itemContainer}>

              <Text>{item.name}</Text>

              <Text>Quantity: {item.quantity}</Text>

              <Text>Price: ${item.price.toFixed(2)}</Text>

            </View>

          )}

        />

        <Text style={styles.totalAmount}>Total Amount: ${totalAmount.toFixed(2)}</Text>

        <TouchableOpacity

          style={styles.placeOrderButton}

          onPress={() => {

            navigation.navigate('SuccessScreen');

          }}

        >

          <Text style={styles.placeOrderButtonText}>Place Order</Text>

        </TouchableOpacity>

      </View>

    </View>

  );

};

export default CheckoutScreen;

## 10.8 services screen.js

// ServicesScreen.js

import React from 'react';

import { View, Text, Image, StyleSheet, FlatList, TouchableOpacity } from 'react-native';

import { useNavigation } from '@react-navigation/native';

const ServicesScreen = () => {

  const navigation = useNavigation();

  const servicesData = [

    {

      id: '1',

      name: 'Wash and Fold',

      description: 'Convenient and hassle-free laundry service for everyday clothes.',

      image: require('../../assets/wash\_and\_fold.jpg'),

    },

    {

      id: '2',

      name: 'Dry Cleaning',

      description: 'Specialized cleaning for delicate or dry-clean-only garments.',

      image: require('../../assets/dry\_cleaning.png'),

    },

    {

      id: '3',

      name: 'Ironing/Pressing',

      description: 'Professional ironing and pressing services to keep your clothes crisp.',

      image: require('../../assets/ironing\_pressing.jpg'),

    },

  ];

  const navigateToServiceDetail = (service) => {

    navigation.navigate('ServiceDetail', { service });

  };

  const renderItem = ({ item }) => (

    <TouchableOpacity

      style={styles.serviceItem}

      onPress={() => navigateToServiceDetail(item)}

    >

      <Image source={item.image} style={styles.serviceImage} />

      <Text style={styles.serviceName}>{item.name}</Text>

      <Text style={styles.serviceDescription}>{item.description}</Text>

    </TouchableOpacity>

  );

  return (

    <View style={styles.container}>

      <Text style={styles.title}>Our Services</Text>

      <FlatList

        data={servicesData}

        keyExtractor={(item) => item.id}

        renderItem={renderItem}

      />

    </View>

  );

};

export default ServicesScreen;

## 10.9 detailed services

import React from 'react';

import { View, Text, Image, StyleSheet, TouchableOpacity } from 'react-native';

import Icon from 'react-native-vector-icons/FontAwesome';

const ServiceDetailScreen = ({ route, navigation }) => {

  const { service } = route.params;

  const additionalDetails = {

    price: '$10.00',

    deliveryTime: '2-3 working days',

    schedule: 'Mon - Fri, 9:00 AM - 5:00 PM',

    specialOffer: 'Get 10% off on your first order!',

    customerReviews: [

      { id: '1', rating: 4.5, comment: 'Excellent service! Highly recommended.' },

      { id: '2', rating: 5, comment: 'Quick and efficient. Will use again.' },

    ],

  };

  const handleScheduleService = () => {

    navigation.navigate('SchedulePickupScreen', { service });

  };

  return (

    <View style={styles.container}>

      <Image source={service.image} style={styles.serviceImage} />

      <Text style={styles.serviceName}>{service.name}</Text>

      <Text style={styles.serviceDescription}>{service.description}</Text>

      <View style={styles.detailsContainer}>

        <Text style={styles.detailLabel}>Price:</Text>

        <Text style={styles.detailValue}>{additionalDetails.price}</Text>

        <Text style={styles.detailLabel}>Delivery Time:</Text>

        <Text style={styles.detailValue}>{additionalDetails.deliveryTime}</Text>

        <Text style={styles.detailLabel}>Schedule:</Text>

        <Text style={styles.detailValue}>{additionalDetails.schedule}</Text>

        <Text style={styles.detailLabel}>Special Offer:</Text>

        <Text style={styles.detailValue}>{additionalDetails.specialOffer}</Text>

        <Text style={styles.detailLabel}>Customer Reviews:</Text>

        {additionalDetails.customerReviews.map((review) => (

          <View key={review.id} style={styles.reviewContainer}>

            <Icon name="star" size={16} color="#953553" style={styles.starIcon} />

            <Text style={styles.reviewComment}>{review.comment}</Text>

          </View>

        ))}

      </View>

      <TouchableOpacity style={styles.scheduleButton} onPress={handleScheduleService}>

        <Text style={styles.scheduleButtonText}>Schedule Service</Text>

      </TouchableOpacity>

    </View>

  );

};

export default ServiceDetailScreen;

## 10.10 schedule pickup

import React, { useState, useContext, useEffect } from 'react';

import { View, Text, TextInput, StyleSheet, TouchableOpacity, ImageBackground } from 'react-native';

import useFirestore from '../hooks/useFirestore';

import { AuthContext } from '../../contexts/AuthContext';

import useAuth from '../hooks/useAuth';

import { addDoc } from 'firebase/firestore';

const SchedulePickupScreen = ({ route, navigation }) => {

    const [serviceName, setServiceName] = useState('');

    const [pickupDate, setPickupDate] = useState('');

    const [pickupTime, setPickupTime] = useState('');

    const [numberOfClothes, setNumberOfClothes] = useState('');

    const [totalPrice, setTotalPrice] = useState(0);

    const { addPickupDetails } = useFirestore();

    const { currentUser } = useAuth();

    const pricePerCloth = 5; // Static price per cloth

    useEffect(() => {

      // Calculate total price whenever the number of clothes changes

      calculateTotalPrice(numberOfClothes);

    }, [numberOfClothes]);

    const calculateTotalPrice = (quantity) => {

      const total = parseInt(quantity, 10) \* pricePerCloth;

      setTotalPrice(isNaN(total) ? 0 : total); // Update total price

    };

    const validateInputs = () => {

      return serviceName && pickupDate && pickupTime && numberOfClothes;

    };

    const handleProceedToCheckout = async () => {

        if (validateInputs()) {

            if (currentUser) {

              try {

                await addPickupDetails(currentUser.uid, {

                  serviceName, pickupDate, pickupTime, numberOfClothes, totalPrice

                });

                navigation.navigate('SuccessScreen', {

                  serviceName,

                  pickupDate,

                  pickupTime,

                  numberOfClothes,

                  totalPrice,

                });

              } catch (error) {

                console.error("Error saving pickup details: ", error);

                // Optionally show an alert or a message to the user

              }

            } else {

              // Handle the case when user is not authenticated

              console.log('User not authenticated');

            }

          } else {

            alert('Please fill in all fields correctly.');

          }

        };

  return (

    <ImageBackground source={require('../../assets/back.jpg')} style={styles.backgroundImage}>

      <View style={styles.overlay}>

        <View style={styles.container}>

          <Text style={styles.title}>Schedule Pickup</Text>

          <TextInput

            style={styles.input}

            placeholder="Service Name"

            value={serviceName}

            onChangeText={(text) => setServiceName(text)}

          />

          <TextInput

            style={styles.input}

            placeholder="Pickup Date"

            value={pickupDate}

            onChangeText={(text) => setPickupDate(text)}

          />

          <TextInput

            style={styles.input}

            placeholder="Pickup Time"

            value={pickupTime}

            onChangeText={(text) => setPickupTime(text)}

          />

          <TextInput

            style={styles.input}

            placeholder="Number of Clothes"

            keyboardType="numeric"

            value={numberOfClothes}

            onChangeText={(text) => setNumberOfClothes(text)}

          />

          <Text style={styles.totalPrice}>Total Price: ${totalPrice}</Text>

          <TouchableOpacity style={styles.proceedButton} onPress={handleProceedToCheckout}>

            <Text style={styles.proceedButtonText}>Proceed to Checkout</Text>

          </TouchableOpacity>

        </View>

      </View>

    </ImageBackground>

  );

};

export default SchedulePickupScreen;

## 10.11 Success screen

import React from 'react';

import { View, Text, StyleSheet, Image } from 'react-native';

import Icon from 'react-native-vector-icons/FontAwesome';

const SuccessScreen = () => {

  return (

    <View style={styles.container}>

      <Image

        source={require('../../assets/back.jpg')}

        style={styles.backgroundImage}

        resizeMode="cover"

      />

      <View style={styles.overlay}>

        <Icon name="check-circle" size={100} color="#953553" />

        <Text style={styles.title}>Order Placed Successfully!</Text>

        <Text style={styles.message}>Your order will be processed shortly.</Text>

      </View>

    </View>

  );

};

export default SuccessScreen;