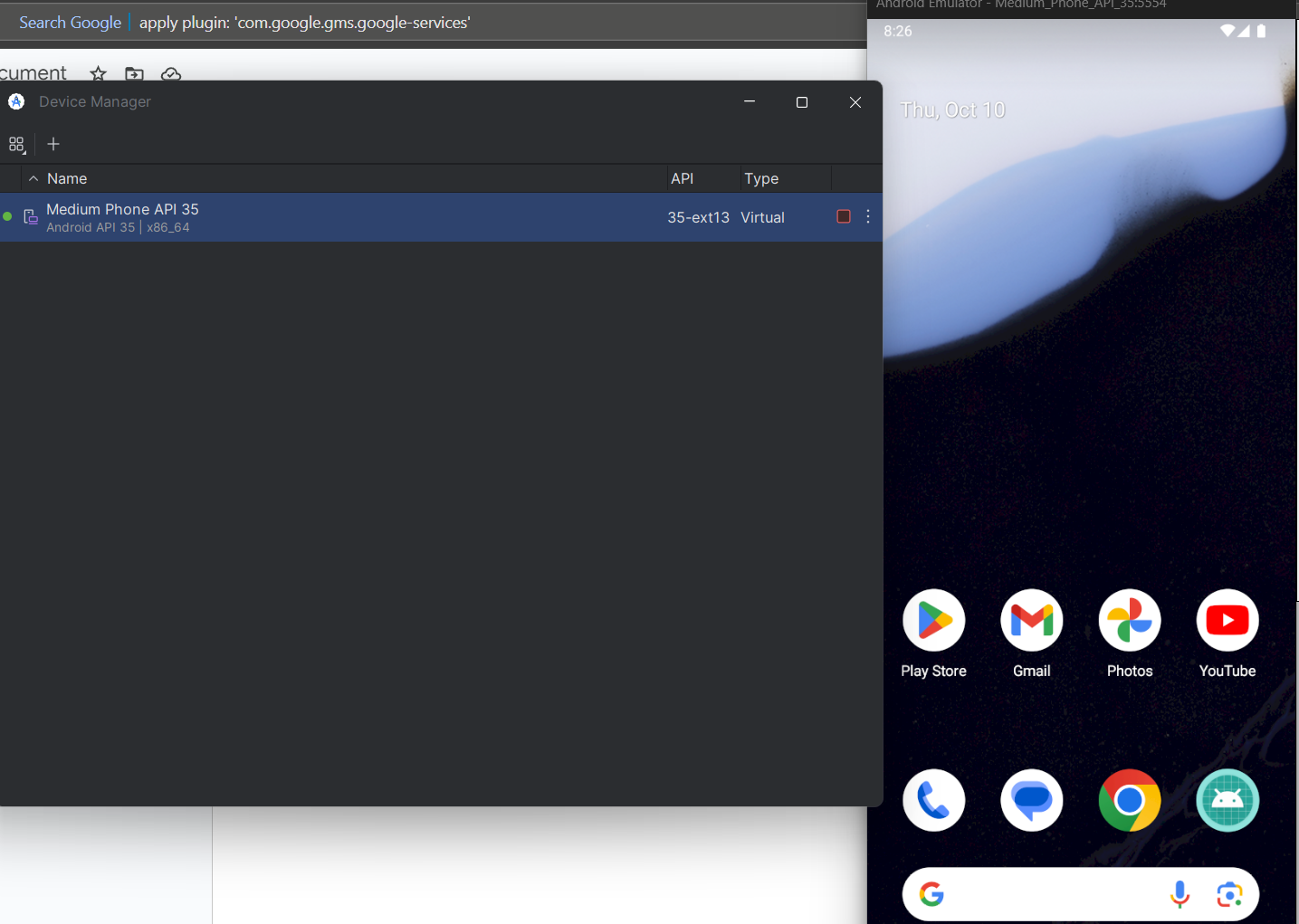
Họ và tên: Phạm Minh Trí

Mã sinh viên: 21IT590

**Các bước thực hiện thực hiện đề tài**

* Tạo ra thư mục ứng dụng bằng câu lệnh: **npx react-native init MidExam**
* Cài đặt các thư viện cần thiết: **npm i @react-native-firebase/app @react-native-firebase-auth @react-native-firebase/firesstore @react-navigation/native @react-navigation/stack react-native-gesture-handler react-native-image-picker react-native-safe-area-context react-native-screens**
* Đăng ký Firestore và Storage trên Firebase Console với tên com.midexam => tải file cấu hình **google-service.json** về và để vào trong thư mục **android/app/**
* Thêm dependencies **("com.google.gms:google-services:4.3.3")** vào file **build.gradle**
* Thêm plugin **apply plugin: 'com.google.gms.google-services'** vào file **android/app/build.gradle**
* Thêm các quyền sử dụng camera, internet và lưu trữ ở trong file **android/app/src/main/AndroidManifest.xml**
* <uses-permission *android:name*="android.permission.INTERNET" />
* <uses-permission *android:name*="android.permission.CAMERA" />
* <uses-permission *android:name*="android.permission.READ\_EXTERNAL\_STORAGE" />
* <uses-permission *android:name*="android.permission.WRITE\_EXTERNAL\_STORAGE" />
* Thêm các stack navigator cho ứng dụng:
* import React from 'react';
* import { NavigationContainer } from '@react-navigation/native';
* import { createStackNavigator } from '@react-navigation/stack';
* import ProductListScreen from './src/screens/ProductListScreen';
* import AddProductScreen from './src/screens/AddProductScreen';
* import EditProductScreen from './src/screens/EditProductScreen';
* import DetailProductScreen from './src/screens/DetailProductScreen';
* *const* Stack = createStackNavigator();
* *const* App = () => {
* return (
* <NavigationContainer>
* <Stack.Navigator *initialRouteName*="ProductList">
* <Stack.Screen
* *name*="ProductList"
* *component*={ProductListScreen}
* *options*={{ title: 'Product List' }}
* />
* <Stack.Screen
* *name*="AddProduct"
* *component*={AddProductScreen}
* *options*={{ title: 'Add Product' }}
* />
* <Stack.Screen
* *name*="EditProduct"
* *component*={EditProductScreen}
* *options*={{ title: 'Edit Product' }}
* />
* <Stack.Screen
* *name*="DetailProduct"
* *component*={DetailProductScreen}
* *options*={{ title: 'Product Details' }}
* />
* </Stack.Navigator>
* </NavigationContainer>
* );
* };
* export default App;
* Tạo file AddProductScreen.js  
  import React, { useState } from 'react';
* import { View, TextInput, Button, StyleSheet, Text, TouchableOpacity, Image } from 'react-native';
* import firestore from '@react-native-firebase/firestore';
* import storage from '@react-native-firebase/storage';
* import { launchImageLibrary } from 'react-native-image-picker';
* *const* AddProductScreen = ({ *navigation* }) => {
* *const* [name, setName] = useState('');
* *const* [type, setType] = useState('');
* *const* [price, setPrice] = useState('');
* *const* [imageUri, setImageUri] = useState('');
* *const* [uploading, setUploading] = useState(false);
* *const* handleAddProduct = *async* () => {
* *let* imageUrl = '';
* if (imageUri) {
* setUploading(true);
* *const* imageRef = storage().ref(`/products/${Date.now()}`);
* await imageRef.putFile(imageUri);
* imageUrl = await imageRef.getDownloadURL();
* setUploading(false);
* }
* firestore().collection('Products').add({
* name,
* type,
* price: parseFloat(price),
* imageUrl,
* }).then(() => {
* setName('');
* setType('');
* setPrice('');
* setImageUri('');
* navigation.navigate('ProductList');
* console.log('Product added!');
* }).catch((*err*) => console.error(err));
* };
* *const* selectImage = () => {
* launchImageLibrary({ mediaType: 'photo' }, (*response*) => {
* if (response.assets && response.assets.length > 0) {
* setImageUri(response.assets[0].uri);
* }
* });
* };
* return (
* <View *style*={styles.container}>
* <Text *style*={styles.header}>Add New Product</Text>
* <TextInput
* *style*={styles.input}
* *placeholder*="Product Name"
* *value*={name}
* *onChangeText*={setName}
* />
* <TextInput
* *style*={styles.input}
* *placeholder*="Product Type"
* *value*={type}
* *onChangeText*={setType}
* />
* <TextInput
* *style*={styles.input}
* *placeholder*="Price"
* *value*={price}
* *onChangeText*={setPrice}
* *keyboardType*="numeric"
* />
* <TouchableOpacity
* *style*={styles.imageButton}
* *onPress*={selectImage}
* >
* <Text *style*={styles.buttonText}>Select Image</Text>
* </TouchableOpacity>
* {imageUri ? (
* <Image
* *source*={{ uri: imageUri }}
* *style*={styles.image}
* />
* ) : null}
* <TouchableOpacity
* *style*={styles.button}
* *onPress*={handleAddProduct}
* *disabled*={uploading}
* >
* <Text *style*={styles.buttonText}>
* {uploading ? 'Uploading...' : 'Add Product'}
* </Text>
* </TouchableOpacity>
* </View>
* );
* };
* *const* styles = StyleSheet.create({
* container: {
* flex: 1,
* padding: 20,
* justifyContent: 'center',
* backgroundColor: '#f5f5f5',
* },
* header: {
* fontSize: 24,
* fontWeight: 'bold',
* marginBottom: 20,
* textAlign: 'center',
* color: '#333',
* },
* input: {
* backgroundColor: '#fff',
* padding: 15,
* borderRadius: 10,
* marginBottom: 20,
* shadowColor: '#000',
* shadowOffset: { width: 0, height: 2 },
* shadowOpacity: 0.1,
* shadowRadius: 4,
* elevation: 2,
* },
* button: {
* backgroundColor: '#007BFF',
* padding: 15,
* borderRadius: 10,
* alignItems: 'center',
* justifyContent: 'center',
* shadowColor: '#000',
* shadowOffset: { width: 0, height: 2 },
* shadowOpacity: 0.2,
* shadowRadius: 4,
* elevation: 5,
* },
* imageButton: {
* backgroundColor: '#28a745',
* padding: 15,
* borderRadius: 10,
* alignItems: 'center',
* justifyContent: 'center',
* marginBottom: 20,
* },
* buttonText: {
* color: '#fff',
* fontSize: 18,
* fontWeight: 'bold',
* },
* image: {
* width: 200,
* height: 200,
* borderRadius: 10,
* alignSelf: 'center',
* marginBottom: 20,
* },
* });
* export default AddProductScreen;
* Code file DetailProductScreen.js  
  import React, { useState, useEffect } from 'react';
* import { View, Text, StyleSheet, Image, TouchableOpacity, ActivityIndicator } from 'react-native';
* import firestore from '@react-native-firebase/firestore';
* *const* DetailProductScreen = ({ *route*, *navigation* }) => {
* *const* { id } = route.params; *// Lấy ID của sản phẩm từ route params*
* *const* [product, setProduct] = useState(null);
* *const* [loading, setLoading] = useState(true);
* *// Lấy chi tiết sản phẩm từ Firestore theo ID*
* useEffect(() => {
* *const* getProduct = *async* () => {
* *const* productDoc = await firestore().collection('Products').doc(id).get();
* if (productDoc.exists) {
* setProduct(productDoc.data());
* }
* setLoading(false);
* };
* getProduct();
* }, [id]);
* if (loading) {
* return (
* <View *style*={styles.loadingContainer}>
* <ActivityIndicator *size*="large" *color*="#007BFF" />
* </View>
* );
* }
* if (!product) {
* return (
* <View *style*={styles.loadingContainer}>
* <Text *style*={styles.errorText}>Product not found.</Text>
* </View>
* );
* }
* return (
* <View *style*={styles.container}>
* {*/\* Hiển thị ảnh sản phẩm \*/*}
* {product.imageUrl ? (
* <Image
* *source*={{ uri: product.imageUrl }}
* *style*={styles.productImage}
* />
* ) : (
* <Text *style*={styles.noImageText}>No image available</Text>
* )}
* {*/\* Hiển thị thông tin chi tiết sản phẩm \*/*}
* <Text *style*={styles.productName}>{product.name}</Text>
* <Text *style*={styles.productType}>Category: {product.type}</Text>
* <Text *style*={styles.productPrice}>Price: ${product.price}</Text>
* {*/\* Nút quay lại danh sách sản phẩm \*/*}
* <TouchableOpacity
* *style*={styles.backButton}
* *onPress*={() => navigation.goBack()}
* >
* <Text *style*={styles.buttonText}>Go Back</Text>
* </TouchableOpacity>
* </View>
* );
* };
* *const* styles = StyleSheet.create({
* container: {
* flex: 1,
* padding: 20,
* backgroundColor: '#f5f5f5',
* justifyContent: 'center',
* alignItems: 'center',
* },
* productImage: {
* width: 300,
* height: 300,
* borderRadius: 10,
* marginBottom: 20,
* },
* noImageText: {
* fontSize: 16,
* fontStyle: 'italic',
* color: '#666',
* marginBottom: 20,
* },
* productName: {
* fontSize: 24,
* fontWeight: 'bold',
* color: '#333',
* marginBottom: 10,
* textAlign: 'center',
* },
* productType: {
* fontSize: 18,
* color: '#666',
* marginBottom: 10,
* },
* productPrice: {
* fontSize: 22,
* fontWeight: '600',
* color: '#007BFF',
* marginBottom: 30,
* },
* backButton: {
* backgroundColor: '#007BFF',
* padding: 15,
* borderRadius: 10,
* alignItems: 'center',
* justifyContent: 'center',
* width: '100%',
* },
* buttonText: {
* color: '#fff',
* fontSize: 18,
* fontWeight: 'bold',
* },
* loadingContainer: {
* flex: 1,
* justifyContent: 'center',
* alignItems: 'center',
* },
* errorText: {
* fontSize: 18,
* color: '#ff0000',
* },
* });
* export default DetailProductScreen;
* Thêm code EditProductScreen.js  
  import React, { useState, useEffect } from 'react';
* import { View, TextInput, Button, StyleSheet, Text, TouchableOpacity, Image, ActivityIndicator } from 'react-native';
* import firestore from '@react-native-firebase/firestore';
* import storage from '@react-native-firebase/storage';
* import { launchImageLibrary } from 'react-native-image-picker';
* *const* EditProductScreen = ({ *route*, *navigation* }) => {
* *const* { id } = route.params;
* *const* [name, setName] = useState('');
* *const* [type, setType] = useState('');
* *const* [price, setPrice] = useState('');
* *const* [imageUri, setImageUri] = useState('');
* *const* [uploading, setUploading] = useState(false);
* *const* [imageUrl, setImageUrl] = useState('');
* useEffect(() => {
* *const* getProduct = *async* () => {
* *const* product = await firestore().collection('Products').doc(id).get();
* *const* data = product.data();
* setName(data.name);
* setType(data.type);
* setPrice(data.price.toString());
* setImageUrl(data.imageUrl); *// Hiển thị ảnh đã có*
* };
* getProduct();
* }, [id]);
* *const* selectImage = () => {
* launchImageLibrary({ mediaType: 'photo' }, (*response*) => {
* if (response.assets && response.assets.length > 0) {
* setImageUri(response.assets[0].uri);
* }
* });
* };
* *const* handleUpdateProduct = *async* () => {
* *let* updatedImageUrl = imageUrl;
* *// Nếu người dùng chọn ảnh mới, tải ảnh lên Firebase Storage*
* if (imageUri) {
* setUploading(true);
* *const* imageRef = storage().ref(`/products/${Date.now()}`);
* await imageRef.putFile(imageUri);
* updatedImageUrl = await imageRef.getDownloadURL();
* setUploading(false);
* }
* *// Cập nhật thông tin sản phẩm và URL ảnh trong Firestore*
* firestore().collection('Products').doc(id).update({
* name,
* type,
* price: parseFloat(price),
* imageUrl: updatedImageUrl,
* }).then(() => {
* navigation.navigate('ProductList');
* });
* };
* return (
* <View *style*={styles.container}>
* <Text *style*={styles.header}>Edit Product</Text>
* <TextInput
* *style*={styles.input}
* *placeholder*="Product Name"
* *value*={name}
* *onChangeText*={setName}
* />
* <TextInput
* *style*={styles.input}
* *placeholder*="Product Type"
* *value*={type}
* *onChangeText*={setType}
* />
* <TextInput
* *style*={styles.input}
* *placeholder*="Price"
* *value*={price}
* *onChangeText*={setPrice}
* *keyboardType*="numeric"
* />
* <TouchableOpacity
* *style*={styles.imageButton}
* *onPress*={selectImage}
* >
* <Text *style*={styles.buttonText}>Select New Image</Text>
* </TouchableOpacity>
* {*/\* Hiển thị ảnh sản phẩm \*/*}
* {imageUri ? (
* <Image
* *source*={{ uri: imageUri }}
* *style*={styles.image}
* />
* ) : (
* imageUrl && (
* <Image
* *source*={{ uri: imageUrl }}
* *style*={styles.image}
* />
* )
* )}
* <TouchableOpacity
* *style*={styles.button}
* *onPress*={handleUpdateProduct}
* *disabled*={uploading}
* >
* <Text *style*={styles.buttonText}>
* {uploading ? 'Uploading...' : 'Update Product'}
* </Text>
* </TouchableOpacity>
* {uploading && <ActivityIndicator *size*="large" *color*="#007BFF" />}
* </View>
* );
* };
* *const* styles = StyleSheet.create({
* container: {
* flex: 1,
* padding: 20,
* justifyContent: 'center',
* backgroundColor: '#f5f5f5',
* },
* header: {
* fontSize: 24,
* fontWeight: 'bold',
* marginBottom: 20,
* textAlign: 'center',
* color: '#333',
* },
* input: {
* backgroundColor: '#fff',
* padding: 15,
* borderRadius: 10,
* marginBottom: 20,
* shadowColor: '#000',
* shadowOffset: { width: 0, height: 2 },
* shadowOpacity: 0.1,
* shadowRadius: 4,
* elevation: 2,
* },
* imageButton: {
* backgroundColor: '#28a745',
* padding: 15,
* borderRadius: 10,
* alignItems: 'center',
* justifyContent: 'center',
* marginBottom: 20,
* },
* buttonText: {
* color: '#fff',
* fontSize: 18,
* fontWeight: 'bold',
* },
* button: {
* backgroundColor: '#007BFF',
* padding: 15,
* borderRadius: 10,
* alignItems: 'center',
* justifyContent: 'center',
* shadowColor: '#000',
* shadowOffset: { width: 0, height: 2 },
* shadowOpacity: 0.2,
* shadowRadius: 4,
* elevation: 5,
* },
* image: {
* width: 200,
* height: 200,
* borderRadius: 10,
* alignSelf: 'center',
* marginBottom: 20,
* },
* });
* export default EditProductScreen;
* Thêm code ProductLisProductListScreen.js  
  import React, { useState, useEffect } from 'react';
* import { View, Text, FlatList, StyleSheet, TouchableOpacity, Image } from 'react-native';
* import firestore from '@react-native-firebase/firestore';
* *const* ProductListScreen = ({ *navigation* }) => {
* *const* [products, setProducts] = useState([]);
* *// Lấy danh sách sản phẩm từ Firestore*
* useEffect(() => {
* *const* unsubscribe = firestore()
* .collection('Products')
* .onSnapshot((*querySnapshot*) => {
* *const* productsArray = [];
* querySnapshot.forEach((*documentSnapshot*) => {
* productsArray.push({
* ...documentSnapshot.data(),
* id: documentSnapshot.id,
* });
* });
* setProducts(productsArray);
* });
* return () => unsubscribe();
* }, []);
* *// Xóa sản phẩm*
* *const* deleteProduct = (*id*) => {
* firestore().collection('Products').doc(id).delete().then(() => console.log('Product deleted!'));
* };
* *const* renderProduct = ({ *item* }) => (
* <TouchableOpacity *onPress*={() => navigation.navigate('DetailProduct', { id: item.id })}>
* <View *style*={styles.productContainer}>
* {*/\* Hiển thị hình ảnh sản phẩm \*/*}
* {item.imageUrl ? (
* <Image
* *source*={{ uri: item.imageUrl }}
* *style*={styles.productImage}
* />
* ) : (
* <Text *style*={styles.noImageText}>No Image Available</Text>
* )}
* <Text *style*={styles.productName}>{item.name}</Text>
* <Text *style*={styles.productType}>Category: {item.type}</Text>
* <Text *style*={styles.productPrice}>Price: ${item.price}</Text>
* <View *style*={styles.buttonContainer}>
* <TouchableOpacity
* *style*={styles.editButton}
* *onPress*={() => navigation.navigate('EditProduct', { id: item.id })}
* >
* <Text *style*={styles.buttonText}>Edit</Text>
* </TouchableOpacity>
* <TouchableOpacity
* *style*={styles.deleteButton}
* *onPress*={() => deleteProduct(item.id)}
* >
* <Text *style*={styles.buttonText}>Delete</Text>
* </TouchableOpacity>
* </View>
* </View>
* </TouchableOpacity>
* );
* return (
* <View *style*={styles.container}>
* <FlatList
* *data*={products}
* *keyExtractor*={(*item*) => item.id}
* *renderItem*={renderProduct}
* />
* <TouchableOpacity
* *style*={styles.addButton}
* *onPress*={() => navigation.navigate('AddProduct')}
* >
* <Text *style*={styles.addButtonText}>Add Product</Text>
* </TouchableOpacity>
* </View>
* );
* };
* *const* styles = StyleSheet.create({
* container: {
* flex: 1,
* padding: 20,
* backgroundColor: '#f5f5f5',
* },
* productContainer: {
* backgroundColor: '#fff',
* padding: 15,
* marginBottom: 15,
* borderRadius: 10,
* shadowColor: '#000',
* shadowOffset: { width: 0, height: 2 },
* shadowOpacity: 0.2,
* shadowRadius: 4,
* elevation: 5,
* },
* productImage: {
* width: '100%',
* height: 150,
* borderRadius: 10,
* marginBottom: 10,
* },
* noImageText: {
* fontSize: 14,
* color: '#999',
* marginBottom: 10,
* fontStyle: 'italic',
* },
* productName: {
* fontSize: 18,
* fontWeight: 'bold',
* marginBottom: 5,
* color: '#333',
* },
* productType: {
* fontSize: 14,
* marginBottom: 5,
* color: '#666',
* },
* productPrice: {
* fontSize: 16,
* marginBottom: 10,
* color: '#007BFF',
* },
* buttonContainer: {
* flexDirection: 'row',
* justifyContent: 'space-between',
* },
* editButton: {
* backgroundColor: '#28a745',
* paddingVertical: 8,
* paddingHorizontal: 20,
* borderRadius: 5,
* },
* deleteButton: {
* backgroundColor: '#dc3545',
* paddingVertical: 8,
* paddingHorizontal: 20,
* borderRadius: 5,
* },
* buttonText: {
* color: '#fff',
* fontWeight: 'bold',
* textAlign: 'center',
* },
* addButton: {
* backgroundColor: '#007BFF',
* paddingVertical: 15,
* borderRadius: 30,
* alignItems: 'center',
* justifyContent: 'center',
* marginTop: 20,
* },
* addButtonText: {
* color: '#fff',
* fontSize: 18,
* fontWeight: 'bold',
* },
* });
* export default ProductListScreen;
* Chạy máy ảo trên Android Studio
* 
* Chạy ứng dụng bằng câu lệnh: **npm start**
* ****
* Chọn nền tảng muốn chạy - ở đây là Android thì sex nhấn nút **a**
* Chương trình sẽ build và cài đặt trên máy ảo.
* 