

BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG
TRƯỜNG ĐẠI HỌC THỦY LỢI



BÀI TẬP LỚN

PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG
ĐỀ TÀI: ỨNG DỤNG BÁN GIÀY THỂ THAO

Giáo viên hướng dẫn: ThS. Kiều Tuấn Dũng

Sinh viên thực hiện:

STT	Mã sinh viên	Họ và tên	Lớp
1	2251061875	Trần Đình Tâm	64CNTT1
2	2251061739	Phạm Tiến Đạt	64CNTT1
3	2251061773	Đặng Hoàng Hiệp	64CNTT1
4	2251061753	Nguyễn Tùng Dương	64CNTT1

Hà Nội, năm 2025

**BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG
TRƯỜNG ĐẠI HỌC THỦY LỢI**



BÀI TẬP LỚN

**PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG
ĐỀ TÀI: ỨNG DỤNG BÁN GIÀY THỂ THAO**

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bảng Số	Bảng Chữ
1	2251061875	Trần Đình Tâm	30/09/2004		
2	2251061739	Phạm Tiến Đạt	19/10/2004		
3	2251061773	Đặng Hoàng Hiệp	09/12/2004		
4	2251061753	Nguyễn Tùng Dương	26/10/2004		

CÁN BỘ CHẤM THI

Hà Nội, năm 2025

LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, việc ứng dụng các thiết bị di động vào hoạt động kinh doanh đã trở thành một xu hướng tất yếu. Nhận thức được điều đó, nhóm chúng em đã quyết định lựa chọn đề tài "Ứng dụng bán giày thể thao" cho bài tập lớn môn "Phát triển ứng dụng cho thiết bị di động".

Mục tiêu của dự án này là xây dựng một ứng dụng di động tiện lợi, thân thiện với người dùng, giúp khách hàng dễ dàng tìm kiếm, lựa chọn và mua sắm các sản phẩm giày thể thao một cách nhanh chóng và hiệu quả. Ứng dụng không chỉ mang đến trải nghiệm mua sắm trực tuyến tuyệt vời mà còn góp phần thúc đẩy sự phát triển của hoạt động kinh doanh giày thể thao trên nền tảng di động.

Trong quá trình thực hiện dự án, nhóm chúng em đã nỗ lực nghiên cứu, tìm hiểu các công nghệ, kỹ thuật lập trình ứng dụng di động, đồng thời áp dụng các kiến thức đã học để xây dựng và hoàn thiện sản phẩm. Báo cáo này là kết quả của quá trình làm việc nghiêm túc, trách nhiệm và đầy tâm huyết của tất cả các thành viên trong nhóm.

Chúng em xin gửi lời cảm ơn chân thành đến thầy **Kiều Tuấn Dũng** đã tận tình hướng dẫn, hỗ trợ và tạo điều kiện thuận lợi cho nhóm chúng em trong suốt quá trình thực hiện bài tập lớn này. Đồng thời, chúng em cũng xin gửi lời cảm ơn đến các bạn sinh viên đã đóng góp ý kiến và hỗ trợ nhóm chúng em hoàn thành báo cáo này.

Mặc dù đã cố gắng hết sức, song do hạn chế về thời gian và kinh nghiệm, báo cáo của nhóm chúng em không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự góp ý, đánh giá từ thầy/cô và các bạn để báo cáo được hoàn thiện hơn.

Xin chân thành cảm ơn!

MỤC LỤC

LỜI NÓI ĐẦU.....	3
MỤC LỤC	4
DANH MỤC CÁC TỪ VIẾT TẮT	6
Chương 1. TỔNG QUAN VỀ ĐỀ TÀI.....	7
1.1. Giới thiệu về đề tài.....	7
1.2. Mục tiêu của đề tài.....	7
1.3. Phạm vi của đề tài.....	7
1.4. Phân chia nhiệm vụ	8
Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ.....	9
2.1. Kiến trúc hệ thống.....	9
2.1.1Giới thiệu về mô hình MVVM:	9
2.1.2 Firebase Authentication (Xác thực người dùng)	9
2.1.3 Firebase Realtime Database (Cơ Sở Dữ Liệu)	9
2.2. Giới thiệu về Công nghệ phát triển	10
Chương 3. XÂY DỰNG ỨNG DỤNG.....	11
3.1. Thiết kế Figma	11
3.2. Thiết kế CSDL.....	11
3.3. Giao diện ứng dụng.....	11
3.3.1. Màn hình mở đầu	12
3.3.2. Màn hình đăng nhập và đăng ký	12
3.3.3. Màn hình trang chủ	13
3.3.4. Màn hình tìm kiếm sản phẩm	14
3.3.5. Màn hình chi tiết sản phẩm.....	15
3.3.6. Màn hình giỏ hàng.....	16

3.3.7. Màn hình trang cá nhân và chỉnh sửa thông tin	17
3.3.8. Màn hình quản lý banner	18
3.3.9. Màn hình quản lý hãng giày	18
3.3.10. Màn hình quản lý sản phẩm.....	19
3.3.11. Màn hình quản lý người dùng.....	19
3.4. Code minh họa các chức năng cốt lõi	20
3.4.1. Code xử lý lấy các dữ liệu từ realtime database	20
3.4.2. Code xử lý tìm kiếm sản phẩm	21
3.4.3. Code xử lý giỏ hàng	21
3.4.4. Code xử lý chỉnh sửa thông tin người dùng	23
3.4.5. Code xử lý đăng xuất người dùng.....	24
KẾT LUẬN	26
1. Kết quả đạt được.....	26
2.Nhược điểm	26
3.Hướng phát triển.....	26
TÀI LIỆU THAM KHẢO	28

DANH MỤC CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	MVVM	Model View View Model
2	UI	User Interface
3	API	Application Programming Interface

Chương 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Giới thiệu về đề tài

Trong thời đại công nghệ số phát triển mạnh mẽ, nhu cầu mua sắm trực tuyến ngày càng gia tăng, đặc biệt là trong lĩnh vực thời trang và giày dép. Với sự tiện lợi của các ứng dụng thương mại điện tử, người dùng có thể dễ dàng tìm kiếm, lựa chọn và đặt hàng các sản phẩm mong muốn một cách nhanh chóng.

Nhằm đáp ứng nhu cầu này, nhóm chúng tôi quyết định phát triển một ứng dụng shop bán giày trên nền tảng Android. Ứng dụng không chỉ giúp người dùng mua sắm dễ dàng mà còn hỗ trợ quản lý sản phẩm, đơn hàng, và cung cấp các chương trình khuyến mãi hấp dẫn.

Dự án này sẽ áp dụng các công nghệ hiện đại như Firebase Realtime Database để quản lý dữ liệu, sử dụng giao diện thân thiện để nâng cao trải nghiệm người dùng.

1.2. Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng một ứng dụng di động hỗ trợ mua sắm giày thể thao trực tuyến với các chức năng chính sau:

- Cung cấp giao diện thân thiện, dễ sử dụng cho khách hàng.
- Quản lý danh mục sản phẩm, hiển thị thông tin chi tiết về giày, bao gồm giá cả, mô tả, hình ảnh.
- Tích hợp giỏ hàng và đặt hàng.
- Quản lý tài khoản người dùng, bao gồm đăng ký, đăng nhập, và cập nhật thông tin cá nhân.
- Xây dựng hệ thống khuyến mãi và giảm giá nhằm thu hút khách hàng.
- Hỗ trợ quản lý đơn hàng và kiểm kê hàng hóa dành cho người quản trị.

1.3. Phạm vi của đề tài

Ứng dụng shop bán giày được xây dựng hướng đến các đối tượng người dùng bao gồm:

- Khách hàng:** Người mua giày có thể tìm kiếm sản phẩm, thêm vào giỏ hàng và đặt hàng.
- Người quản trị:** Quản lý danh sách sản phẩm, cập nhật thông tin sản phẩm và theo dõi đơn hàng.

Phạm vi của đề tài bao gồm:

- Nền tảng:** Ứng dụng được phát triển trên hệ điều hành Android.
- Dữ liệu:** Sử dụng Firebase Realtime Database để lưu trữ thông tin sản phẩm, người dùng và đơn hàng.
- Chức năng chính:** Xây dựng các tính năng cần thiết như đăng nhập, đăng ký, tìm kiếm sản phẩm, quản lý giỏ hàng và đặt hàng.

Dự án này tập trung vào việc hoàn thiện một ứng dụng có tính thực tiễn cao, giúp người dùng dễ dàng tiếp cận sản phẩm và thực hiện các giao dịch nhanh chóng, tiện lợi.

1.4. Phân chia nhiệm vụ

<<Bảng phân chia nhiệm vụ>>

Thành viên	Nhiệm vụ
Trần Đình Tâm	-Thiết kế và triển khai cơ sở dữ liệu Realtime Database -Tích hợp Firebase Authentication, bảo mật -Triển khai chức năng và màn hình Trang chủ -Xây dựng quản lý của Admin(web-server)
Phạm Tiến Đạt	-Triển khai chức năng và màn hình chi tiết sản phẩm -Xuất file báo cáo
Nguyễn Tùng Dương	-Triển khai chức năng và màn hình mở đầu. -Xử lý chức năng và màn hình của đăng nhập, đăng ký.
Đặng Hoàng Hiệp	-Triển khai chức năng và màn hình giỏ hàng. -Thiết kế giao diện (UI/UX)

Chương 2. KIẾN TRÚC VÀ CÔNG NGHỆ

2.1. Kiến trúc hệ thống

Ứng dụng shop bán giày được xây dựng theo mô hình **MVVM (Model - View - ViewModel)**. Đây là mô hình kiến trúc phổ biến trong phát triển ứng dụng Android, giúp tách biệt logic xử lý dữ liệu, giao diện và tương tác người dùng, từ đó cải thiện khả năng bảo trì và mở rộng ứng dụng.

2.1.1 Giới thiệu về mô hình MVVM:

MVVM là một kiến trúc phần mềm giúp tách biệt các thành phần trong ứng dụng để tăng tính linh hoạt và dễ bảo trì. Mô hình MVVM bao gồm ba thành phần chính:

1. **Model (M):**

- Là thành phần chịu trách nhiệm quản lý dữ liệu.
- Trong ứng dụng shop bán giày, Model sẽ làm việc trực tiếp với Firebase Realtime Database để truy xuất và cập nhật dữ liệu sản phẩm, đơn hàng và người dùng.
- Dữ liệu có thể được lấy về thông qua API hoặc giao tiếp với cơ sở dữ liệu.

2. **View (V):**

- Là giao diện người dùng (UI) của ứng dụng.
- Hiển thị thông tin sản phẩm, giỏ hàng, trạng thái đơn hàng và nhận tương tác từ người dùng.
- View chỉ tập trung vào hiển thị dữ liệu mà không xử lý logic nghiệp vụ.

3. **ViewModel (VM):**

- Chịu trách nhiệm xử lý logic nghiệp vụ và điều phối dữ liệu giữa Model và View.
- Giữ trạng thái của dữ liệu và đảm bảo rằng View luôn cập nhật khi có thay đổi.
- Sử dụng **LiveData** để lắng nghe và cập nhật dữ liệu theo thời gian thực mà không ảnh hưởng đến vòng đời của Activity/Fragment.

Luồng hoạt động của MVVM trong ứng dụng:

1. Người dùng thực hiện thao tác trên giao diện (View).
2. View gửi yêu cầu đến ViewModel để xử lý logic.
3. ViewModel truy vấn dữ liệu từ Model (Firebase Realtime Database).
4. Khi có dữ liệu, ViewModel sử dụng LiveData để cập nhật View một cách tự động.
5. Giao diện người dùng hiển thị dữ liệu mới theo trạng thái cập nhật.

2.1.2 Firebase Authentication (Xác thực người dùng)

- **Firebase Authentication** là dịch vụ xác thực người dùng do Google cung cấp trong Firebase. Nó giúp đăng nhập, đăng ký, đăng xuất, quản lý người dùng một cách đơn giản và an toàn.

2.1.3 Firebase Realtime Database (Cơ Sở Dữ Liệu)

- Là cơ sở dữ liệu Realtime NoSQL, lưu trữ dữ liệu dưới dạng cây JSON, phù hợp với các ứng dụng yêu cầu đồng bộ hóa dữ liệu theo thời gian thực, như:

chat, danh sách sản phẩm, thông báo, v.v.

- Dữ liệu được tổ chức dạng key-value (khóa – giá trị), dễ dàng truy cập thông qua các đường dẫn (path), nhưng không hỗ trợ subcollection như Firestore.
- Cung cấp khả năng đồng bộ hóa thời gian thực (Realtime Sync) giữa các thiết bị và nền tảng (Web, Android, iOS...), đảm bảo người dùng luôn nhận được dữ liệu mới nhất mà không cần tải lại ứng dụng.
- Hỗ trợ Firebase Security Rules, giúp phân quyền truy cập dữ liệu theo người dùng (user-based access control), đảm bảo an toàn dữ liệu khi nhiều người sử dụng hệ thống cùng lúc.

2.2. Giới thiệu về Công nghệ phát triển

Ngôn ngữ lập trình

- **Java:** Được sử dụng làm ngôn ngữ chính để phát triển ứng dụng Android.

Công nghệ và thư viện sử dụng

- **Firebase Realtime Database:** Cơ sở dữ liệu NoSQL giúp lưu trữ thông tin sản phẩm, đơn hàng và người dùng theo thời gian thực.
- **Firebase Authentication:** Hỗ trợ đăng nhập và xác thực người dùng bằng email, Google, Facebook.
- **Glide:** Thư viện giúp tải và hiển thị hình ảnh sản phẩm một cách tối ưu.
- **LiveData & ViewModel:** Quản lý dữ liệu giúp giao diện cập nhật theo thời gian thực mà không bị ảnh hưởng bởi vòng đời của Activity/Fragment.
- **RecyclerView:** Hiển thị danh sách sản phẩm theo cách tối ưu và có thể cuộn mượt mà.
- **Chip Navigation Bar:** Thư viện hỗ trợ tạo thanh điều hướng giao diện dễ dàng.
- **Dots Indicator:** Hỗ trợ hiển thị chỉ báo vị trí cho các banner quảng cáo hoặc slider hình ảnh sản phẩm.
- **Gson:** Hỗ trợ chuyển đổi dữ liệu JSON sang Java Object và ngược lại.
- **Room Database:** Cung cấp cơ sở dữ liệu cục bộ giúp lưu trữ dữ liệu tạm thời trên thiết bị, giảm tải truy vấn trực tiếp từ Firebase.

Công cụ phát triển

- **Android Studio:** Môi trường chính để viết mã và debug ứng dụng.
- **Git/GitHub:** Hỗ trợ quản lý mã nguồn, làm việc nhóm và kiểm soát phiên bản.
- **Firebase Console:** Cung cấp giao diện quản lý cơ sở dữ liệu, người dùng, xác thực và thông báo đẩy.

Với kiến trúc MVVM và các công nghệ hiện đại, ứng dụng shop bán giày đảm bảo khả năng mở rộng, bảo trì dễ dàng và tối ưu hiệu suất hoạt động.

Chương 3. XÂY DỰNG ỨNG DỤNG

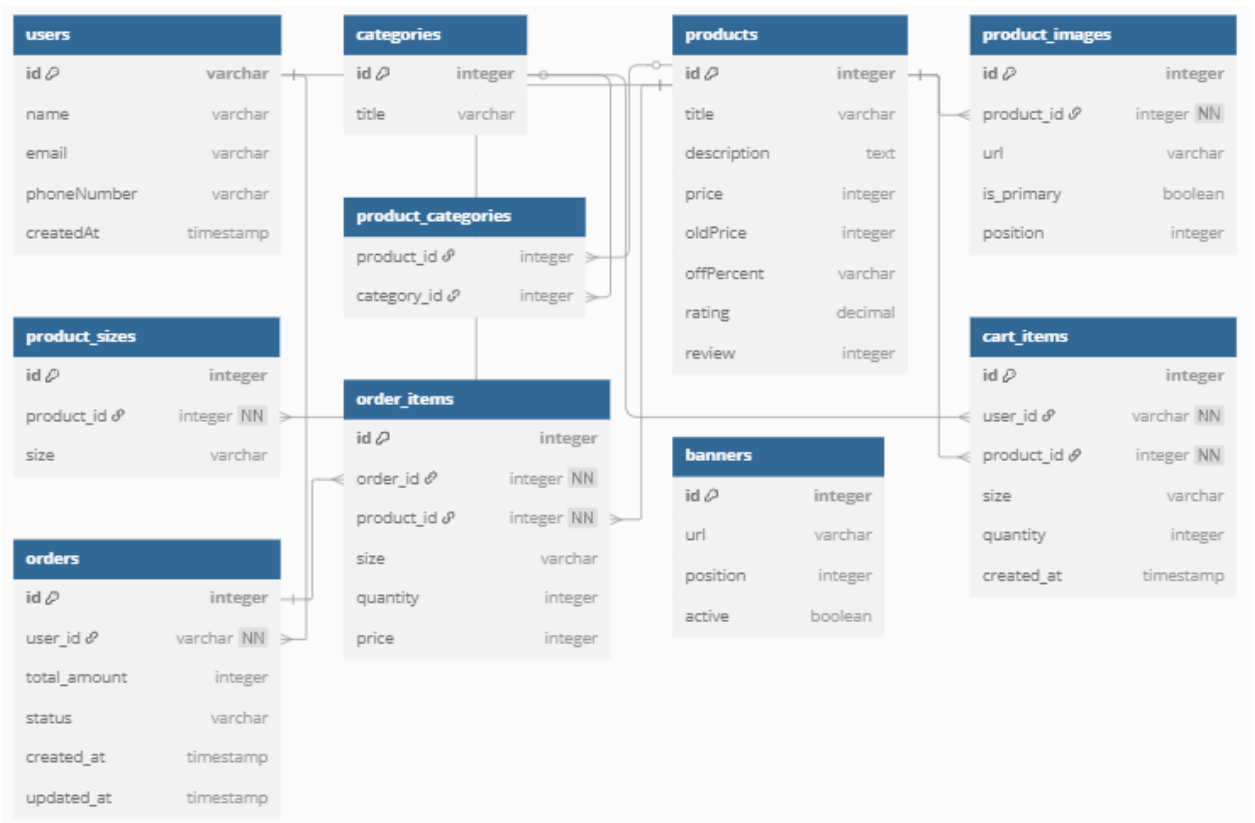
3.1. Thiết kế Figma

Link Figma:

https://www.figma.com/design/yME9qbfORqWT6bAUK0PHGv/Shop_b%C3%A1n_gi%C3%A0_vapp?node-id=0-1&m=dev&t=2LpgFrIErAMtZIAK-1

3.2. Thiết kế CSDL

<<Lược đồ>>

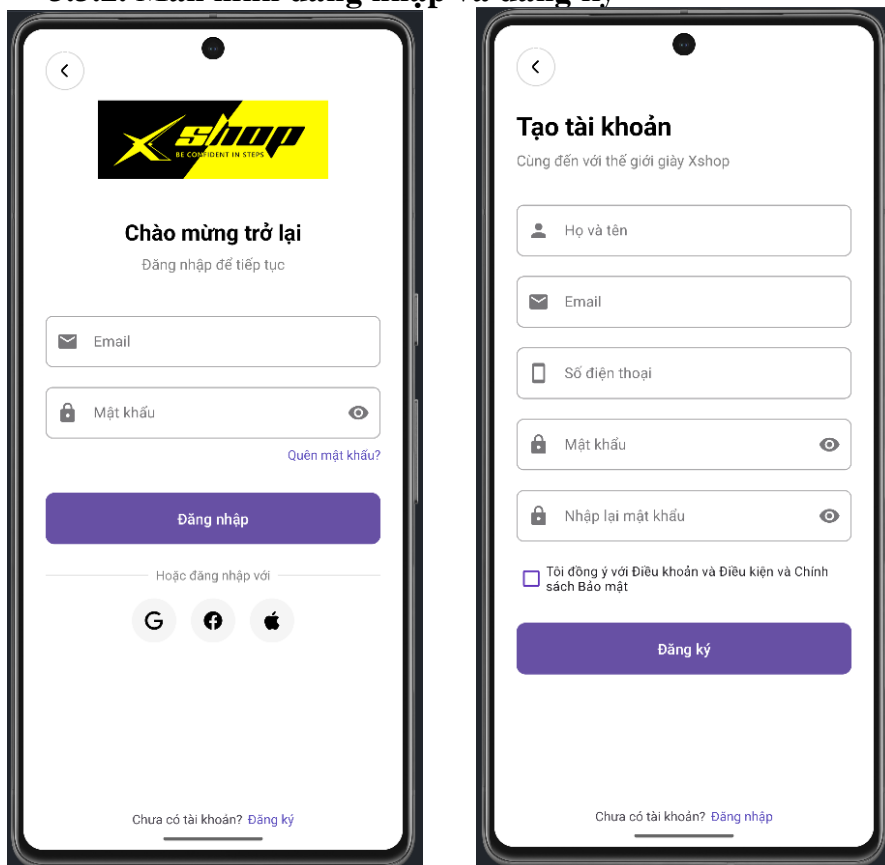


3.3. Giao diện ứng dụng

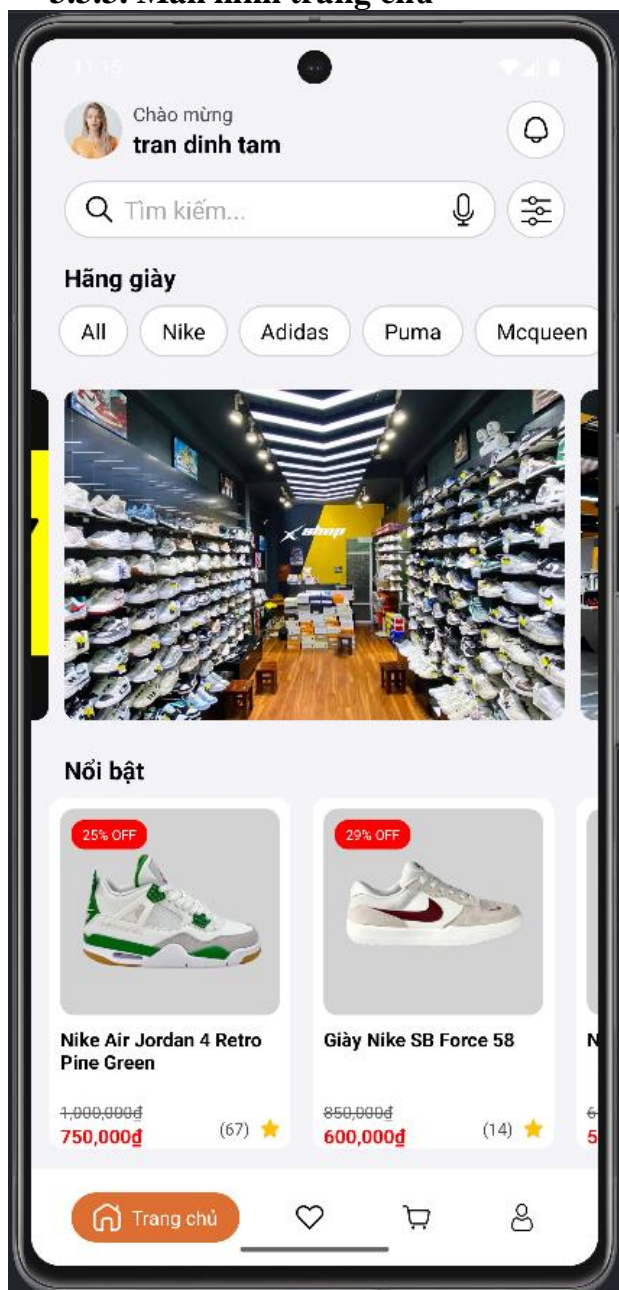
3.3.1. Màn hình mở đầu



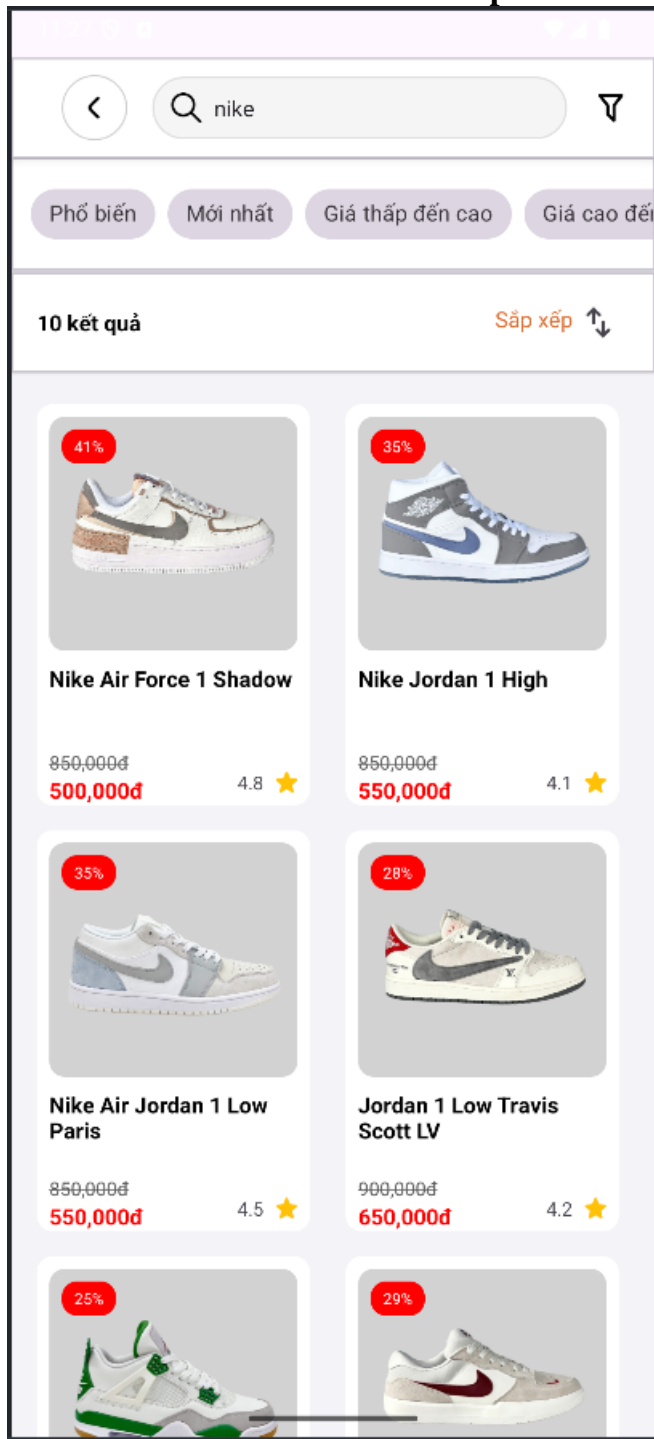
3.3.2. Màn hình đăng nhập và đăng ký



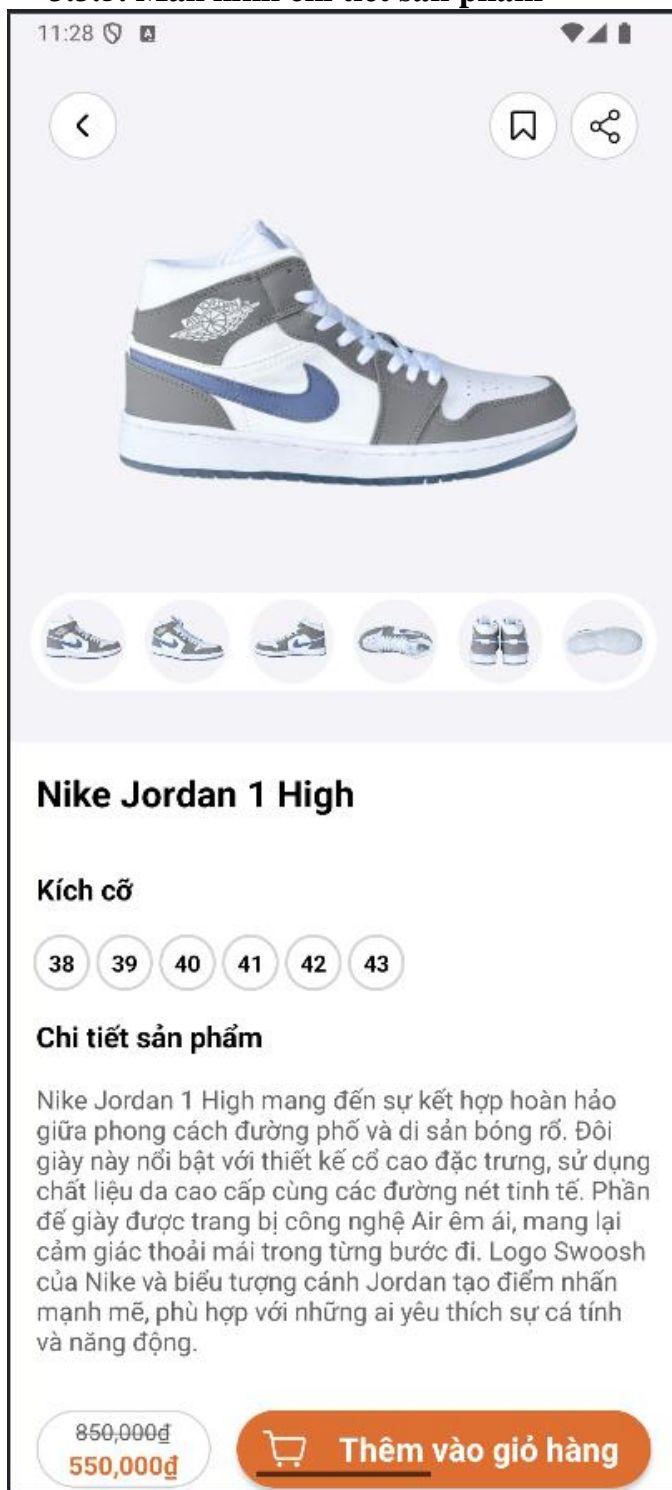
3.3.3. Màn hình trang chủ



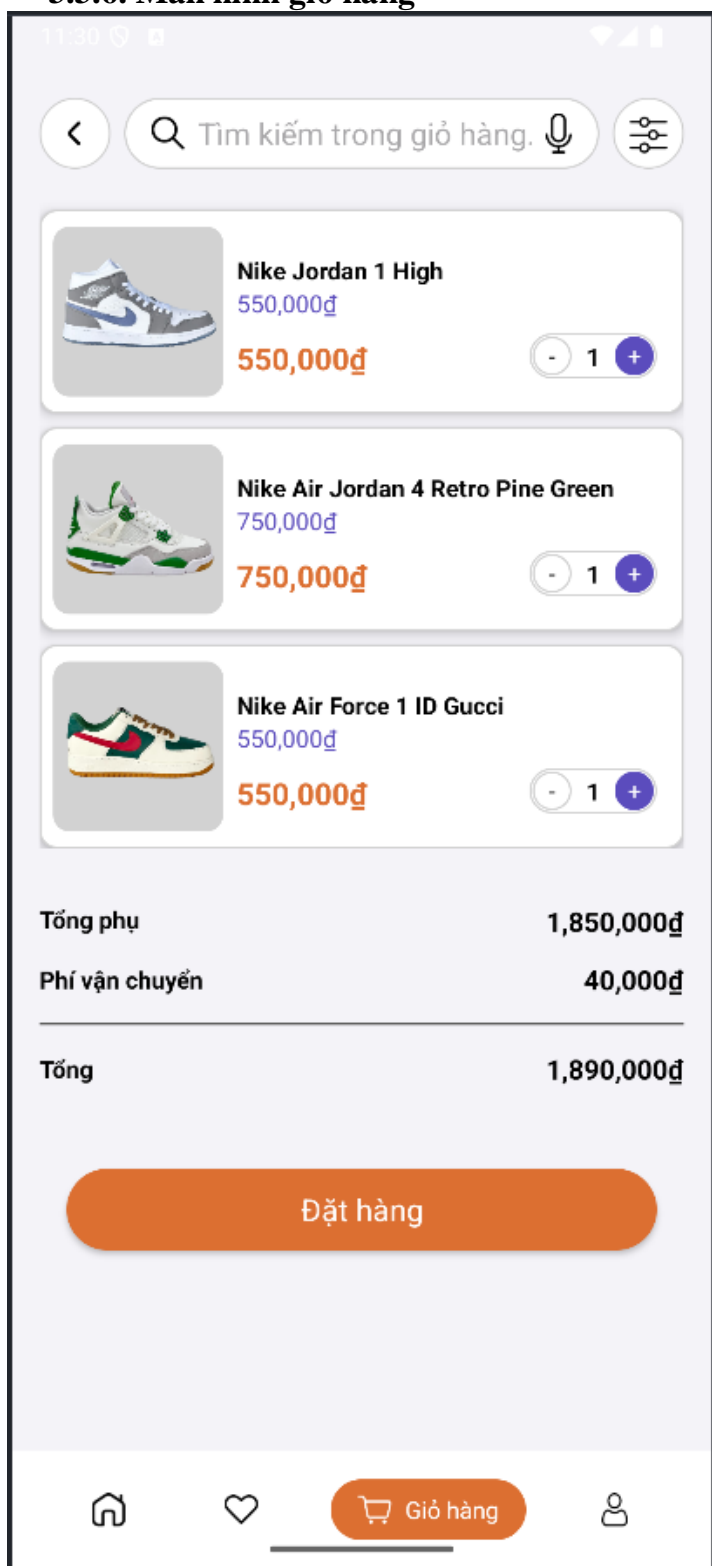
3.3.4. Màn hình tìm kiếm sản phẩm



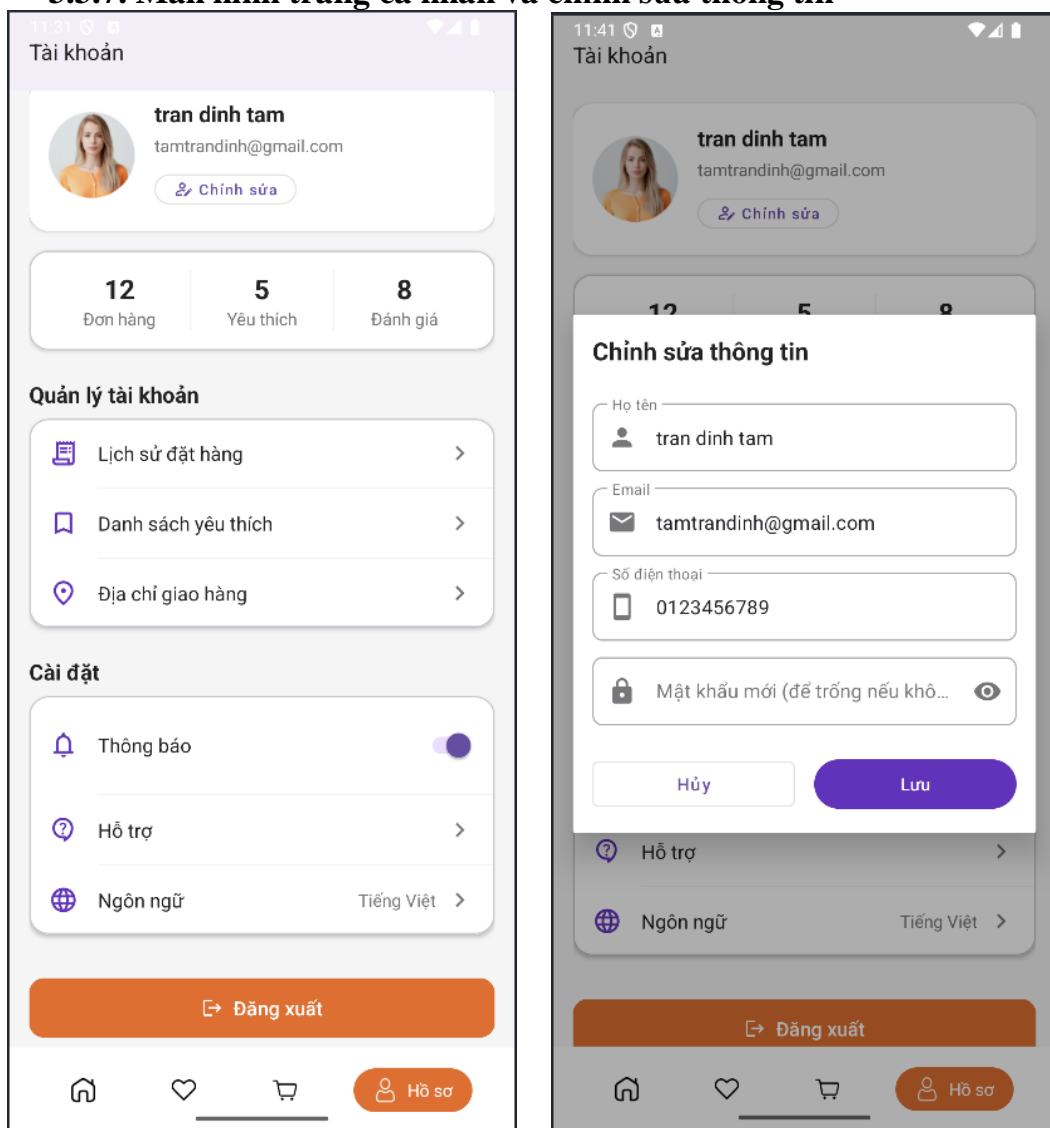
3.3.5. Màn hình chi tiết sản phẩm



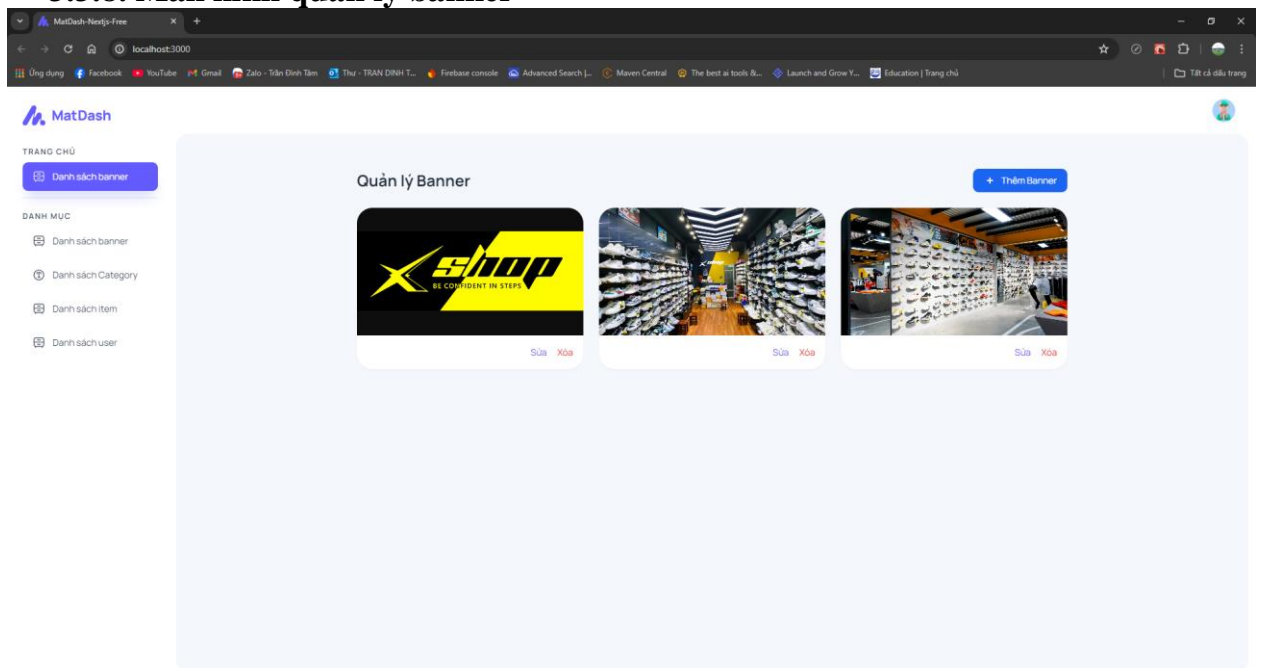
3.3.6. Màn hình giỏ hàng



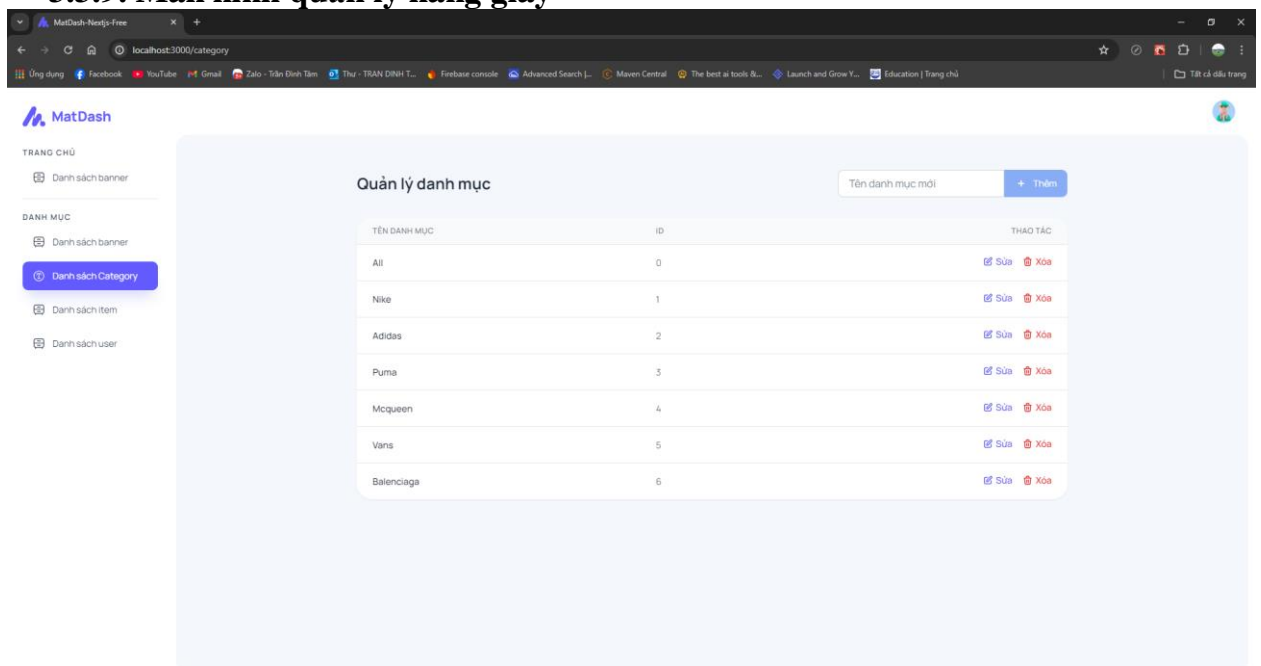
3.3.7. Màn hình trang cá nhân và chỉnh sửa thông tin



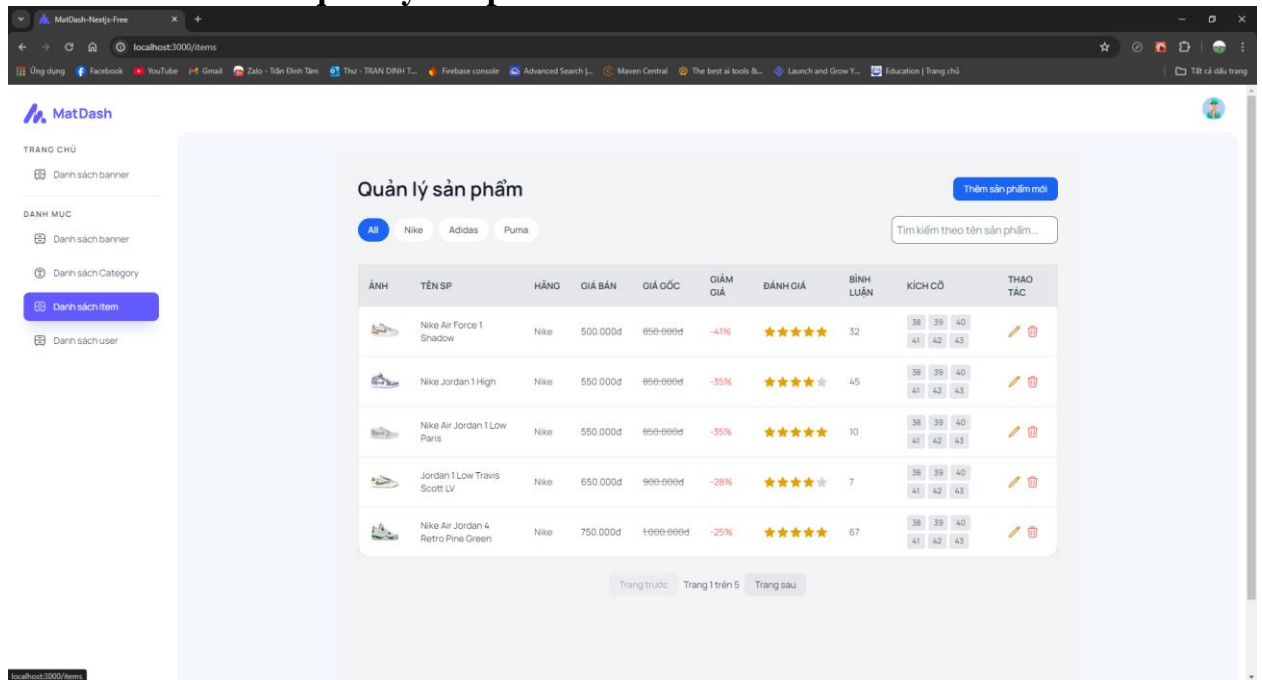
3.3.8. Màn hình quản lý banner



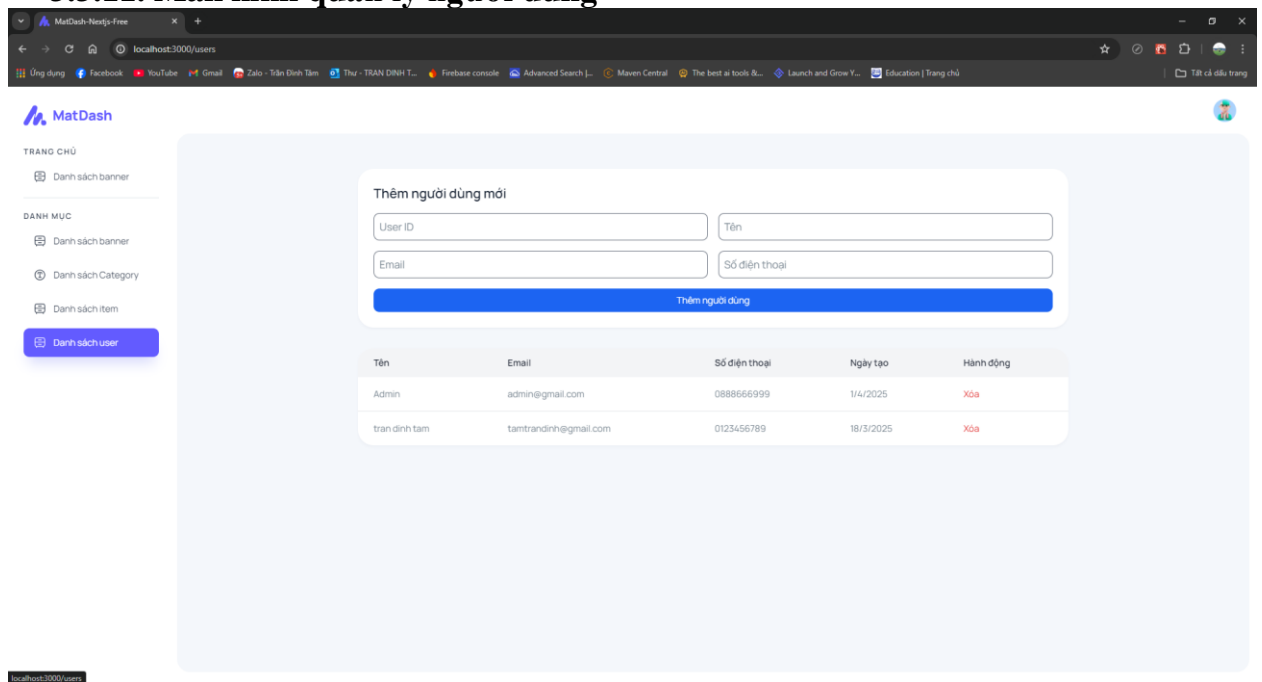
3.3.9. Màn hình quản lý hãng giày



3.3.10. Màn hình quản lý sản phẩm



3.3.11. Màn hình quản lý người dùng



3.4. Code minh họa các chức năng cốt lõi

3.4.1. Code xử lý lấy các dữ liệu từ realtime database

```
public class MainRepository {
    private final FirebaseDatabase firebaseDatabase = FirebaseDatabase.getInstance();
    private static final double MIN_POPULAR_RATING = 4.5;
    public LiveData<ArrayList<CategoryModel>> loadCategory(){
        MutableLiveData<ArrayList<CategoryModel>> listData = new MutableLiveData<>();
        DatabaseReference ref = firebaseDatabase.getReference("Category");
        ref.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                ArrayList<CategoryModel> list = new ArrayList<>();
                for (DataSnapshot childSnapshot : snapshot.getChildren()){
                    CategoryModel item = childSnapshot.getValue(CategoryModel.class);
                    if(item != null){
                        list.add(item);
                    }
                }
                listData.setValue(list);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
        return listData;
    }

    public LiveData<ArrayList<BannerModel>> loadBanner(){
        MutableLiveData<ArrayList<BannerModel>> listData = new MutableLiveData<>();
        DatabaseReference ref = firebaseDatabase.getReference("Banner");
        ref.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                ArrayList<BannerModel> list = new ArrayList<>();
                for (DataSnapshot childSnapshot : snapshot.getChildren()){
                    BannerModel item = childSnapshot.getValue(BannerModel.class);
                    if(item != null){
                        list.add(item);
                    }
                }
                listData.setValue(list);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
        return listData;
    }

    public LiveData<ArrayList<ItemsModel>> loadPopular(){
        MutableLiveData<ArrayList<ItemsModel>> listData = new MutableLiveData<>();
        DatabaseReference ref = firebaseDatabase.getReference("Items");
        ref.addValueEventListener(new ValueEventListener() {
```

```

@Override
public void onDataChange(@NonNull DataSnapshot snapshot) {
    ArrayList<ItemsModel> list = new ArrayList<>();
    for (DataSnapshot childSnapshot : snapshot.getChildren()){
        ItemsModel item = childSnapshot.getValue(ItemsModel.class);
        if(item != null && item.getRating() >= MIN_POPULAR_RATING ){
            list.add(item);
        }
    }
    listData.setValue(list);
}

@Override
public void onCancelled(@NonNull DatabaseError error) {

}
});
return listData;
}
}

```

3.4.2. Code xử lý tìm kiếm sản phẩm

```

private void setupSearchFunctionality() {
    binding.edtSearching.setOnEditorActionListener((v, actionId, event) -> {
        if(actionId == EditorInfo.IME_ACTION_SEARCH ||
            (event != null && event.getKeyCode() == KeyEvent.KEYCODE_ENTER)) {
            // Lấy query từ EditText
            String query = binding.edtSearching.getText().toString().trim();

            if(!query.isEmpty()) {
                // Ấn bàn phím
                InputMethodManager imm = (InputMethodManager)
                    requireContext().getSystemService(Context.INPUT_METHOD_SERVICE);
                imm.hideSoftInputFromWindow(v.getWindowToken(), 0);

                // Log để kiểm tra giá trị query
                Log.d("HomeFragment", "Search query: " + query);

                // Chuyển đến SearchResultsActivity với query
                Intent intent = new Intent(getActivity(), SearchResultsActivity.class);
                intent.putExtra("searchQuery", query);
                startActivity(intent);
            }
            return true;
        }
        return false;
    });
}

```

3.4.3. Code xử lý giỏ hàng

```

private void updateCartUI() {
    if(managmentCart.getListCart().isEmpty()) {
        // Hiển thị thông báo giỏ hàng trống
        binding.emptyTxt.setVisibility(View.VISIBLE);
        binding.emptyTxt.setText("Giỏ hàng của bạn đang trống");
    }
}

```

```

// Căn giữa thông báo giỏ hàng trống trên màn hình
FrameLayout.LayoutParams params = new FrameLayout.LayoutParams(
    ViewGroup.LayoutParams.WRAP_CONTENT,
    ViewGroup.LayoutParams.WRAP_CONTENT
);
params.gravity = android.view.Gravity.CENTER;
binding.emptyTxt.setLayoutParams(params);

// Ẩn nội dung giỏ hàng
binding.scrollViewCart.setVisibility(View.GONE);
} else {
    // Ẩn thông báo giỏ hàng trống
    binding.emptyTxt.setVisibility(View.GONE);

    // Hiển thị nội dung giỏ hàng
    binding.scrollViewCart.setVisibility(View.VISIBLE);

    // Thiết lập RecyclerView
    setupCartRecyclerView();

    // Tính toán và hiển thị tổng tiền giỏ hàng
    calculatorCart();
}
}

private void setupCartRecyclerView() {
    binding.cartView.setLayoutManager(new LinearLayoutManager(requireContext(),
        LinearLayoutManager.VERTICAL, false));
    cartAdapter = new CartAdapter(managmentCart.getListCart(), requireContext(), () -> {
        // Đây là callback khi các mục trong giỏ thay đổi
        calculatorCart();

        // Kiểm tra xem giỏ hàng có rỗng sau khi xóa mục không
        if(managmentCart.getListCart().isEmpty()) {
            updateCartUI();
        } else {
            // Cập nhật adapter với danh sách giỏ hàng mới nhất
            cartAdapter.updateCartItems(managmentCart.getListCart());
        }
    });
    binding.cartView.setAdapter(cartAdapter);
}

private void setVariable() {
    binding.btnOrder.setOnClickListener(v -> {
        if(!managmentCart.getListCart().isEmpty()) {
            // Xóa giỏ hàng sau khi đặt hàng
            clearCart();

            // Hiển thị thông báo thành công
            Toast.makeText(requireContext(), "Đặt hàng thành công! Cảm ơn bạn đã mua sắm.",
                Toast.LENGTH_SHORT).show();

            // Quay về màn hình chính
            navigateToHome();
        } else {
            Toast.makeText(requireContext(), "Giỏ hàng của bạn đang trống",
                Toast.LENGTH_SHORT).show();
        }
    });
}

```

```

});
}

private void navigateToHome() {
    if(getActivity() instanceof MainActivity) {
        MainActivity mainActivity = (MainActivity) getActivity();
        // Chọn tab trang chủ trong thanh điều hướng dưới
        mainActivity.selectTab(R.id.home);
    }
}

private void clearCart() {
    // Lấy các mục hiện tại trong giỏ hàng
    if(managmentCart != null) {
        // Xóa từng mục trong giỏ hàng
        while (!managmentCart.getListCart().isEmpty()) {
            managmentCart.minusItem(managmentCart.getListCart(), 0, 0) -> {

        }
    }
}

private void calculatorCart() {
    double delivery = 40000;
    double itemTotal = managmentCart.getTotalFee();
    double total = itemTotal + delivery;

    NumberFormat nf = NumberFormat.getNumberInstance(Locale.getDefault());
    binding.totalFeeTxt.setText(nf.format(itemTotal) + "đ");
    binding.deliveryTxt.setText(nf.format(delivery) + "đ");
    binding.totalTxt.setText(nf.format(total) + "đ");
}

```

3.4.4. Code xử lý chỉnh sửa thông tin người dùng

```

private void showEditProfileDialog() {
    Dialog dialog = new Dialog(requireContext());
    dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
    DialogEditProfileBinding dialogBinding = DialogEditProfileBinding.inflate(getLayoutInflater());
    dialog.setContentView(dialogBinding.getRoot());
    dialog.getWindow().setLayout(ViewGroup.LayoutParams.MATCH_PARENT,
    ViewGroup.LayoutParams.WRAP_CONTENT);

    // Lấy thông tin người dùng hiện tại
    UserModel currentUser = viewModel.getCurrentUser().getValue();
    if(currentUser != null) {
        // Điền thông tin hiện tại vào các trường nhập
        dialogBinding.nameEditText.setText(currentUser.getName());
        dialogBinding.emailEditText.setText(currentUser.getEmail());
        dialogBinding.phoneEditText.setText(currentUser.getPhoneNumber());
        // Không hiển thị mật khẩu vì lý do bảo mật
    }

    // Sự kiện click nút Hủy
    dialogBinding.btnCancel.setOnClickListener(v -> dialog.dismiss());

    // Sự kiện click nút Lưu

```

```

dialogBinding.btnSave.setOnClickListener(v -> {
    // Lấy dữ liệu từ các trường nhập liệu
    String name = dialogBinding.nameEditText.getText().toString().trim();
    String email = dialogBinding.emailEditText.getText().toString().trim();
    String phone = dialogBinding.phoneEditText.getText().toString().trim();
    String password = dialogBinding.passwordEditText.getText().toString().trim();

    // Kiểm tra dữ liệu nhập vào
    if(name.isEmpty()) {
        dialogBinding.nameInputLayout.setError("Vui lòng nhập họ tên");
        return;
    }

    if(email.isEmpty()) {
        dialogBinding.emailInputLayout.setError("Vui lòng nhập email");
        return;
    }

    // Cập nhật hồ sơ người dùng
    viewModel.updateUserProfile(name, email, phone, password.isEmpty() ? null : password);

    // Đóng dialog
    dialog.dismiss();

    // Hiển thị hộp thoại loading
    showLoadingDialog();
});

dialog.show();
}

private AlertDialog loadingDialog;

private void showLoadingDialog() {
    // Hiển thị dialog loading khi đang cập nhật thông tin
    AlertDialog.Builder builder = new AlertDialog.Builder(requireContext());
    builder.setView(R.layout.dialog_loading);
    builder.setCancelable(false);
    loadingDialog = builder.create();
    loadingDialog.show();

    // Tắt dialog khi cập nhật xong (dù thành công hay thất bại)
    viewModel.getUpdateSuccess().observe(getViewLifecycleOwner(), isSuccess -> {
        if(loadingDialog != null && loadingDialog.isShowing()) {
            loadingDialog.dismiss();
        }
    });

    viewModel.getErrorMessage().observe(getViewLifecycleOwner(), error -> {
        if(loadingDialog != null && loadingDialog.isShowing()) {
            loadingDialog.dismiss();
        }
    });
}

```

3.4.5. Code xử lý đăng xuất người dùng


```
private void showLogoutConfirmationDialog() {  
    // Hiển thị hộp thoại xác nhận đăng xuất  
    AlertDialog.Builder builder = new AlertDialog.Builder(requireContext());  
    builder.setTitle("Đăng xuất");  
    builder.setMessage("Bạn có chắc chắn muốn đăng xuất không?");  
    builder.setPositiveButton("Có", (dialog, which) -> {  
        // Đăng xuất khỏi Firebase  
        mAuth.signOut();  
  
        // Chuyển về màn hình đăng nhập và xóa stack hoạt động  
        Intent intent = new Intent(requireActivity(), LoginActivity.class);  
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);  
        startActivity(intent);  
        requireActivity().finish();  
    });  
    builder.setNegativeButton("Không", (dialog, which) -> {  
        dialog.dismiss();  
    });  
    builder.show();  
}
```

KẾT LUẬN

1. Kết quả đạt được

Sau quá trình nghiên cứu và phát triển, nhóm đã hoàn thành ứng dụng bán giày thể thao trên nền tảng Android với các chức năng cơ bản như:

- Giao diện thân thiện, dễ sử dụng, phù hợp với người dùng phổ thông.
- Hệ thống quản lý tài khoản cho phép người dùng đăng ký, đăng nhập, cập nhật thông tin cá nhân.
- Chức năng hiển thị danh sách sản phẩm chi tiết: tên giày, mô tả, hình ảnh, giá cả.
- Tích hợp giỏ hàng, cho phép thêm, xóa sản phẩm và tiến hành đặt hàng.
- Cơ sở dữ liệu được quản lý tập trung thông qua Firebase Realtime Database, giúp cập nhật dữ liệu nhanh và đồng bộ.
- Hệ thống quản trị cơ bản cho phép người quản lý cập nhật sản phẩm.

Những kết quả trên đã phần nào đáp ứng được mục tiêu đã đặt ra trong đề tài, hướng tới việc tạo ra một ứng dụng có tính thực tiễn cao, tiện lợi cho cả người dùng và người quản trị.

2. Nhược điểm

- Chức năng quản lý đơn hàng cho người quản trị chưa được triển khai, do hạn chế về thời gian và khối lượng công việc.
- Ứng dụng chưa tích hợp tính năng phân quyền người dùng, dẫn đến chưa phân biệt rõ ràng giữa khách hàng và người quản lý.
- Giao diện còn đơn giản, chưa có nhiều tùy chọn nâng cao hoặc hiệu ứng hiện đại.
- Chưa hỗ trợ các tính năng nâng cao như: thông báo đơn hàng, chương trình khuyến mãi tự động, hoặc hệ thống đánh giá sản phẩm.
- Chưa tích hợp thanh toán trực tuyến, quá trình đặt hàng mới chỉ dừng lại ở việc chọn sản phẩm và xác nhận đơn.

3. Hướng phát triển

- **Bổ sung chức năng quản lý đơn hàng:** cho phép người quản trị theo dõi đơn hàng của khách, xem trạng thái đơn hàng (chờ xử lý, đã xác nhận, đang giao, hoàn tất...), cập nhật và xử lý đơn hàng trực tiếp từ hệ thống.
- Tích hợp phân quyền người dùng để tách biệt vai trò khách hàng và quản trị viên.
- Thiết kế giao diện hiện đại hơn, nâng cao trải nghiệm người dùng, đồng thời tối ưu cho nhiều kích thước màn hình.
- Thêm tính năng thông báo đơn hàng, thông tin khuyến mãi qua Firebase Cloud Messaging.
- Tích hợp thanh toán trực tuyến qua ví điện tử hoặc thẻ ngân hàng.
- Xây dựng hệ thống đánh giá, phản hồi sản phẩm từ người dùng để tăng tương tác

và độ tin cậy của ứng dụng.

- Mở rộng ứng dụng lên các nền tảng khác như iOS hoặc phiên bản website trong tương lai.

TÀI LIỆU THAM KHẢO

- [1] David Flanagan, JavaScript: The Definitive Guide, 7th Edition, O'Reilly Media, 2020.
- [2] Adam Freeman, “Pro jQuery”, Apress, 2018.
- [3] Benjamin Jakobus, “Mastering Bootstrap 5”, Packt Publishing, 2018.