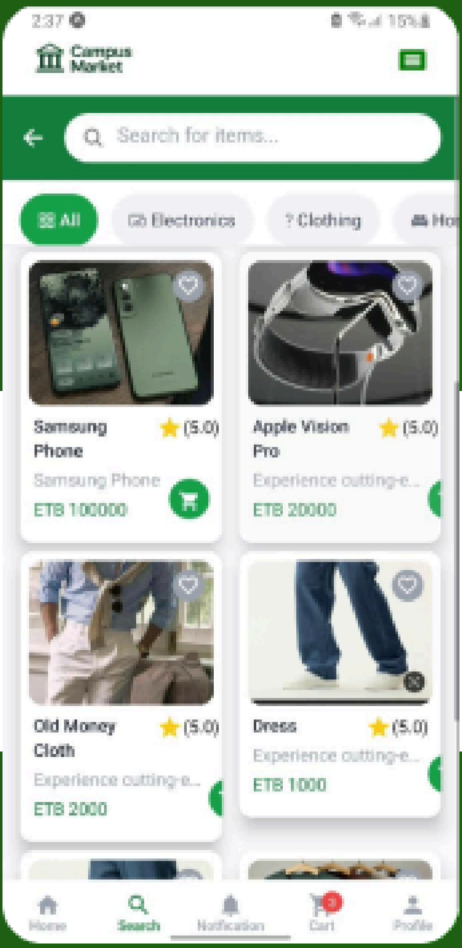
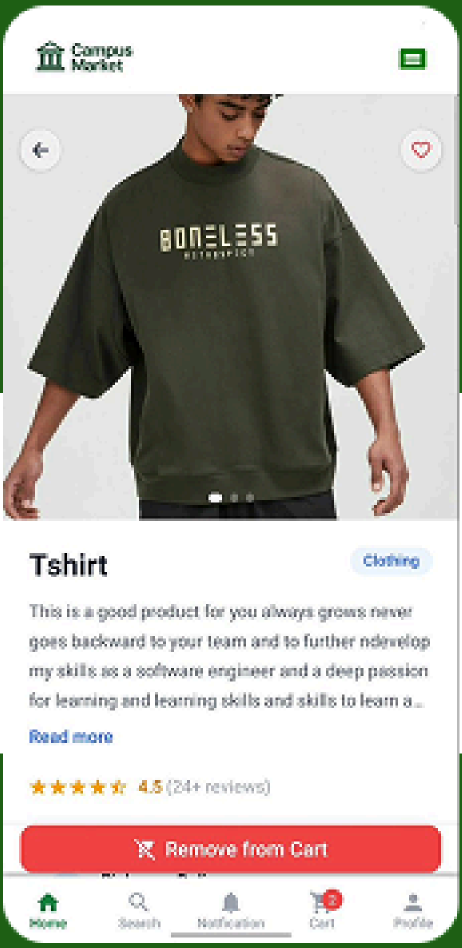
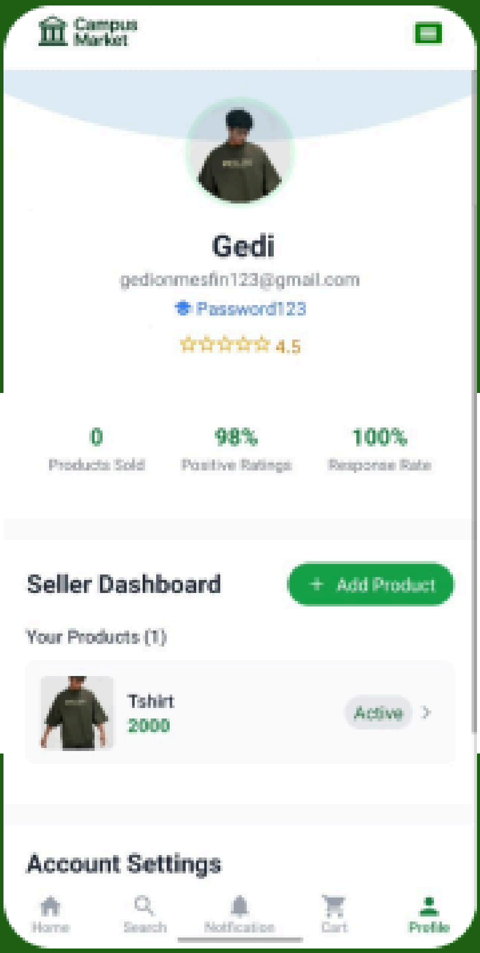
Mobile App Project

Campus Market place





This is a good product for you always grows never goes backward to your team and to further ndevelop my skills as a software engineer and a deep passion for learning and learning skills and skills to learn a...



Group Members	Id
1. Nahom Habtamu	Ugr/25347/14
2. Abduselam Tesfaye	Ugr/25696/14
3. Biruk Abebe	Ugr/25917/14
4. Ifnan Faysel	Ugr/26050/14
5. Sonte Adamu	Ugr/25780/14

SECTION 1

Chapter 1: Project Overview

CampusMarketplace is a mobile-first student-centric marketplace platform designed to bridge the gap between buyers and sellers within a university environment. It provides students with a smart, secure, and intuitive space to buy, sell, and communicate around products relevant to campus life — from textbooks and electronics to clothing, furniture, and more. The application is built with an emphasis on convenience, real-time interaction, and a polished user experience that resonates with the fast-paced student lifestyle.

This project started with a simple observation: students often struggle to find an efficient, trustworthy, and campus-specific way to exchange goods. While global platforms exist, they don't cater to the unique needs of students — such as hyperlocal communication, affordability, or academic-related listings. **CampusMarketplace** fills this gap with a tailored solution where the entire marketplace is restricted to campus users, making it safer and more relevant.

A key focus of the app is **seamless user authentication**. Students can create accounts, log in securely, and log out when needed. Personal information is protected through robust authentication systems like Firebase Auth or JWT-based APIs. This gives users confidence in the system's reliability and ensures their data remains safe.

Once inside the app, students can **create and manage product listings** with ease. Posting a product takes only a few steps: upload an image, enter a title, write a description, and set a price. If the product details need to change, users can easily edit or remove the listing at any time. The product feed is visually appealing and dynamically updates as new items are added or removed, making browsing enjoyable and efficient.

Searching for a product is also fast and flexible. With powerful **search and filter capabilities**, users can narrow down results based on product categories (like electronics, books, fashion), keywords, and pricing. This intelligent filtering saves time and helps students find what they need quickly.

One of the standout features is the **real-time chat functionality**, which enables direct communication between buyers and sellers. Whether a user wants to negotiate, ask questions, or simply confirm availability, they can do so instantly through an in-app messaging system. The chat interface is intuitive and optimized for mobile, mirroring the design and functionality students are already familiar with from apps like WhatsApp or Messenger.

The app also includes a **cart feature**, allowing users to add products they're interested in and revisit them later. This helps buyers manage their interest in multiple items without needing to make immediate decisions. The cart can also serve as a staging area for planned purchases or negotiations.

Students have complete control over their experience through the **profile and account management system**. Users can update their display name, contact details, and profile image. They can also manage their own listings, review chat history, and track their product interactions all in one centralized dashboard.

Another key element is the **help and support section**, where users can find FAQs, contact information, and troubleshooting guides. This ensures students can always find assistance when encountering issues or have questions about how to use the app’s features.

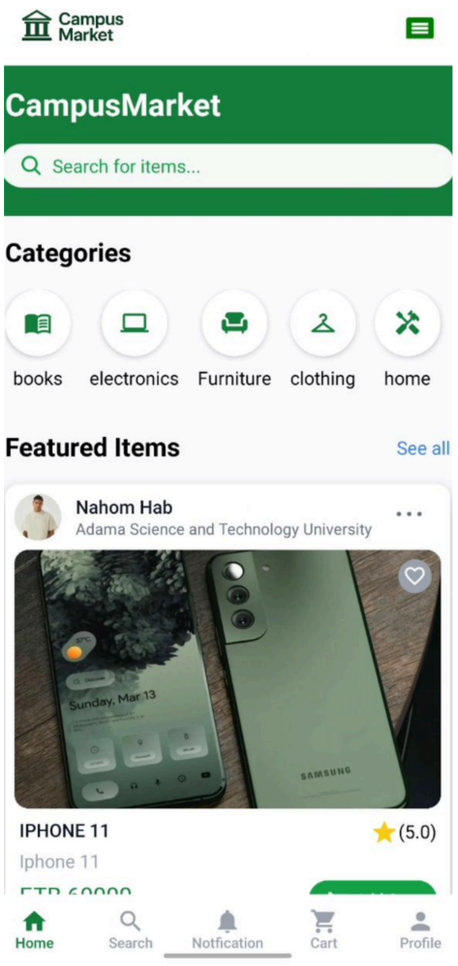
Core Features at a Glance

Product Listings

- Post items with images, descriptions, and prices
- View all listings in a clean, modern grid layout
- Filter listings by categories like: Books, Electronics, Fashion, Furniture

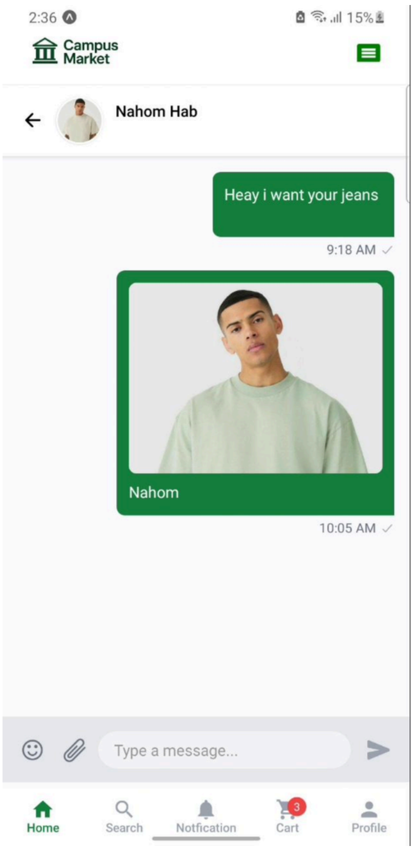
Smart Search & Filters

- Keyword-based product search
- Filter results by category, price, or popularity
- Live suggestions and results preview



Real-Time Chat

- Chat directly with the product owner
- Receive notifications instantly
- Built using Firebase or Socket.IO for live sync

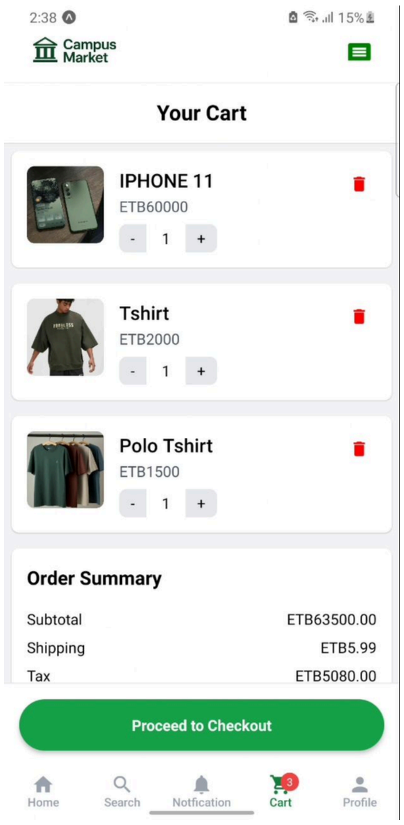


Heay i want your jeans
9:18 AM

Nahom
10:05 AM

Profile & Account Control

- Edit personal info (name, bio, picture)
- View your listings, favorite items, and chats
- Secure login/logout with Firebase Auth or JWT



Your Cart

IPHONE 11
ETB60000
- 1 +

Tshirt
ETB2000
- 1 +

Polo Tshirt
ETB1500
- 1 +

Order Summary

Subtotal	ETB63500.00
Shipping	ETB5.99
Tax	ETB5080.00

Cart

- Add items to your cart to purchase later
- Review and compare selected items
- Easy checkout interface

Product Management

- Edit posted products: images, details, price
- Delete items once sold
- See buyer interest and chat history

Help & Support

- In-app support center with FAQs
- Contact help service for assistance
- User-friendly feedback and report system

Chapter 2 : Requirement Gathering Document

2.1. Introduction

The Student Marketplace is a digital platform designed specifically for university students to buy and sell used items such as textbooks, gadgets, clothing, and furniture. The goal is to create a secure, easy-to-use, and engaging environment that supports student entrepreneurship, sustainability, and community interaction.

This document outlines the detailed requirements gathered for building the Student Marketplace system, including functional and non-functional needs, user personas, user stories, and the various roles involved.

2.2. Stakeholders

2.2.1 Students (Buyers & Sellers)

- Primary users of the platform.
- Can list products for sale.
- Can browse, search, and purchase products.
- Can communicate with other users through chat.
- Can manage their account and product listings.

2.2.2 University Community

- Supports the platform adoption and usage.
- Benefits from a reduced environmental footprint through reuse and resale.
- May integrate the system into university services and announcements.

2.3. User Personas

2.3.1 Sami – The Seller

- **Age**: 23
- **Background**: Electrical Engineering student, nearing graduation.
- **Goal**: Wants to sell used textbooks, headphones, and calculators he no longer needs.
- **Behavior**: Posts products occasionally, expects easy management of his listings.
- **Needs**: Simple posting interface, fast approval, chat with buyers.

2.3.2 Liya – The Buyer

- **Age**: 19

- **Background**: Freshman in Computer Science.
- **Goal**: Looking for affordable used laptop and course materials.
- **Behavior**: Frequently browses listings and compares prices.
- **Needs**: Fast search, category filters, chat to negotiate, a cart to save items.

2.4. User Stories

ID	As a...	I want to...	So that I can...
US1	Student	post products easily	sell my items quickly and without hassle
US2	User	search and filter products by category	find relevant items faster
US3	Buyer	chat with the product owner	ask questions and negotiate before buying
US4	User	edit my account and product listings	keep my information up-to-date
US5	Student	add items to my cart	review them later before making a decision

2.5 Functional Requirements

This section defines the core functionality the application must support to meet user and business needs. It outlines what the system **should do**, organized into feature-specific categories.

2.5.1 User Authentication

Description: Enables secure user identity management and access control.

Requirements:

- Users must be able to sign up using a valid email and password.
- Users must be able to log in using email and password credentials.
- Logout functionality must securely destroy user sessions or tokens.
- Session or token-based authentication (preferably JWT or Firebase Auth) must be implemented.
- Email verification should be required after signup before accessing full features.
- Users must be able to request and perform password resets via email links.

2.5.2 Product Listing Management

Description: Allows users to manage their product offerings.

Requirements:

- Users must be able to create a product listing with the following fields:
  - Product Title
  - Description
  - Price
  - Category (from predefined list)
  - Product Condition (New or Used)
  - One or more Photos
- Users must be able to edit any of their own product listings.
- Users must be able to delete any of their own listings.
- Validations should enforce required fields, file size/type limits, and field constraints (e.g., positive price).

2.5.3 Product Browsing

Description: Provides users with an interface to explore available products.

Requirements:

- All available listings must be visible on the homepage or browse page.
- Each product card must display:
  - Thumbnail Image
  - Product Title
  - Price
  - Short Description (limited characters)
  - Optional: Seller Name or Avatar
- Listings must be paginated or lazy-loaded for performance.

2.5.4 Product Search and Filtering

Description: Enables users to find relevant products through advanced discovery tools.

Requirements:

- Users must be able to search products using free-text keywords.
- Filtering must be supported based on:
  - Category
  - Price range (slider or min-max)
  - Product Condition (New / Used)
  - Date (Newest or Oldest first)
- Filters should be combinable and resettable.

2.5.5 Product Detail View

Description: Displays complete information about a product when selected.

Requirements:

- Upon clicking a product, a detail page must show:
  - Full Description
  - Additional Images (carousel or grid)
  - Seller Information
  - Contact or "Chat Now" button
  - “Add to Cart” button with product ID
- Suggested: Related products section below main details.

2.5.6 Add to Cart

Description: Provides basic cart functionality before checkout implementation.

Requirements:

- Users must be able to add multiple products to a temporary cart.
- Cart should persist through the session (or until logout/clear).
- Cart interface must support:
  - Viewing items with thumbnails, title, and price
  - Removing items from cart
  - Subtotal calculation

2.5.7 Real-Time Chat

Description: Allows direct messaging between buyer and seller for inquiries.

Requirements:

- Buyers must be able to initiate chat directly from the product detail page.

- Chat interface must support:
  - Text messages
  - Timestamps
  - Seen/unseen status indicators
- Chats must persist in the database (Firebase or MongoDB).
- Chat history must be viewable in the user profile (My Messages).

2.5.8 Account and Profile Management

Description: Allows users to manage their identity and view their marketplace activity.

Requirements:

- Users must be able to:
  - Edit profile picture, name, bio, and contact info
  - View their own posted listings
  - Delete their account (with confirmation)
- Profiles should show:
  - Profile image
  - Listings count
  - Contact options

2.5.9 Help & Support

Description: Provides users with access to assistance and issue reporting tools.

Requirements:

- FAQ section must be accessible with expandable question/answer format.
- Contact support form must allow users to send queries or issues.
- Each product listing must include a "Report" button for:
  - Spam
  - Inappropriate content
  - Scams

2.6 Non-Functional Requirements

2.6.1 Responsive UI

- The platform must support:
  - Mobile screens (360px and up)
  - Tablets
  - Desktop views
- Use adaptive layouts and CSS grid/flexbox for responsiveness.

2.6.2 Fast Performance

- Initial page loads should complete in **< 3 seconds**.
- Use pagination or lazy loading to improve performance.
- Compress images and optimize API calls.
- Cache product and user data where possible.

2.6.3 Secure Authentication

- Use HTTPS for all communication.
- All passwords must be hashed (e.g., bcrypt).
- Mitigate OWASP Top 10 risks:
  - Cross-Site Scripting (XSS)
  - Cross-Site Request Forgery (CSRF)
  - SQL/NoSQL Injection
  - Token theft

2.6.4 Real-Time Communication

- Real-time messaging must use WebSockets or Firebase.
- Notifications for new chat messages or purchases.
- Ensure real-time sync across user devices.

2.6.5 Scalability

- Platform must support:
  - Thousands of concurrent users
  - Real-time chats
  - Daily high-volume product uploads
- Modular architecture and microservices encouraged.

2.7 Visual & UX Requirements

- Minimalist and clean UI (Google Material Design or Tailwind CSS)
- Easy navigation with top or side menu
- Use FontAwesome icons for visual cues
- Light/Dark mode toggle (optional)
- Smooth UI transitions for:
  - Chat messages
  - Adding/removing cart items
  - Opening product detail

2.8 Technical Stack

Layer	Technology
Frontend	React.js + Tailwind CSS
Backend	Node.js + Express js
Database	MongoDB
Auth	JWT
Chat	Socket.IO
Hosting	Vercel / Netlify / Heroku / Cpanal

2.9 Future Enhancements

1. Integrated payments (Stripe, PayPal)
2. Product ratings and reviews
3. Delivery or pick-up scheduling features
4. AI-based product recommendation engine
5. Admin dashboard with analytics
6. Multi-language support with i18n

Chapter 3 : Design Document

3.1. Introduction

This document outlines the design strategy for the Student Marketplace platform, focusing on wireframes, mockups, UI/UX principles, and overall user journey. The goal is to create an intuitive, visually appealing, and responsive interface that enhances usability and engagement.

3.2. Wireframes & Mockups

3.2.1 Design Tool

All wireframes and high-fidelity mockups are designed using **Figma**, ensuring collaborative, scalable, and version-controlled design workflows.

3.2.2 Designed Screens

The following core screens have been designed and prototyped:

Screen	Description
Home	Displays product feed with filters, featured items, and categories.
Search	Search bar with live filtering, category selection, and price range slider.
Product Detail	Full product details, image gallery, seller info, and action buttons.
Chat	Real-time messaging interface with message history, seen status, and timestamps.
Profile	User info, profile editing, product management (posted items).
Cart	Shows saved products, subtotal, and checkout prompt.
Login/Signup	Secure authentication with form validation and error handling.

💡 **Note:** Components are modular and designed with reusability in mind to accelerate development and maintain consistency.

3.3. UI/UX Highlights

3.3.1 Visual Identity

- **Style**: Minimal, modern, student-friendly.
- **Color Scheme**: Bold campus-inspired colors (e.g., deep blue, energetic orange, leaf green).
- **Typography**: Clean, legible sans-serif fonts (Poppins).
- **Icons**: Font Awesome for consistent visual language.

3.3.2 Layout & Responsiveness

- Fully responsive across devices: desktop, tablet, mobile.

- Grid layout for product displays.
- Sticky navigation for key screens (Home, Chat, Cart).
- Adaptive components for screen size (e.g., modal for mobile chat view).

3.3.3 Interactions & Animations

- Smooth transitions between pages using fade and slide effects.
- Button and icon hover effects for engagement.
- Loader animations during data fetch.
- Real-time feedback for user actions (e.g., product added to cart, message sent).

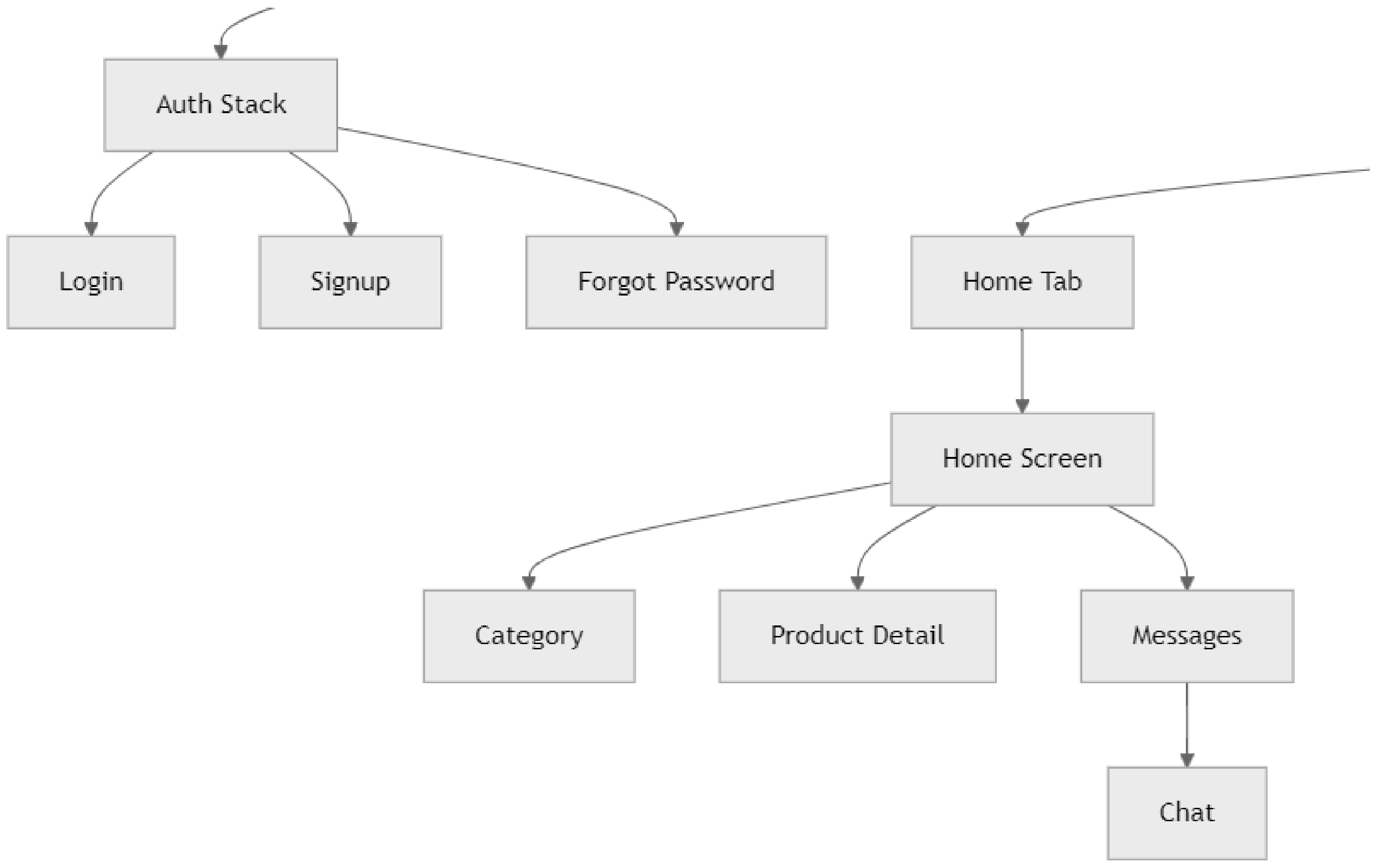
3.3.4 Accessibility

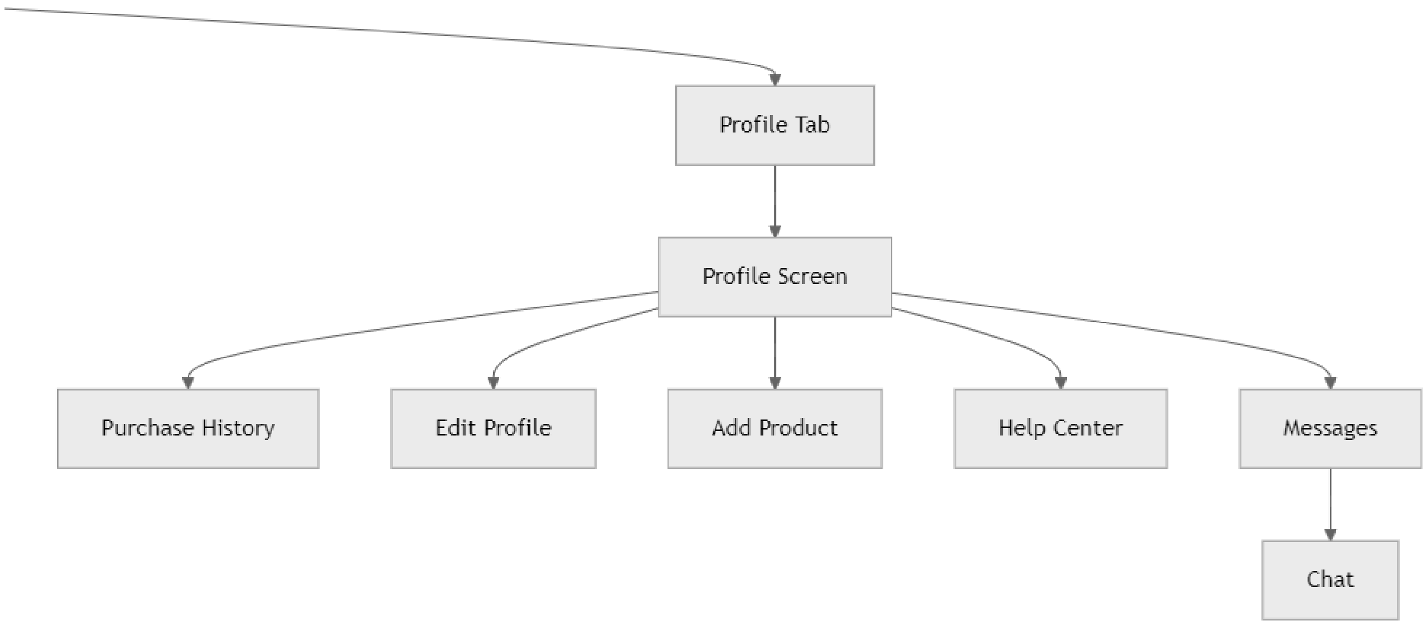
- Color contrast meets WCAG 2.1 AA standards.
- Keyboard navigable components.
- Alt text for images and labels for form fields.

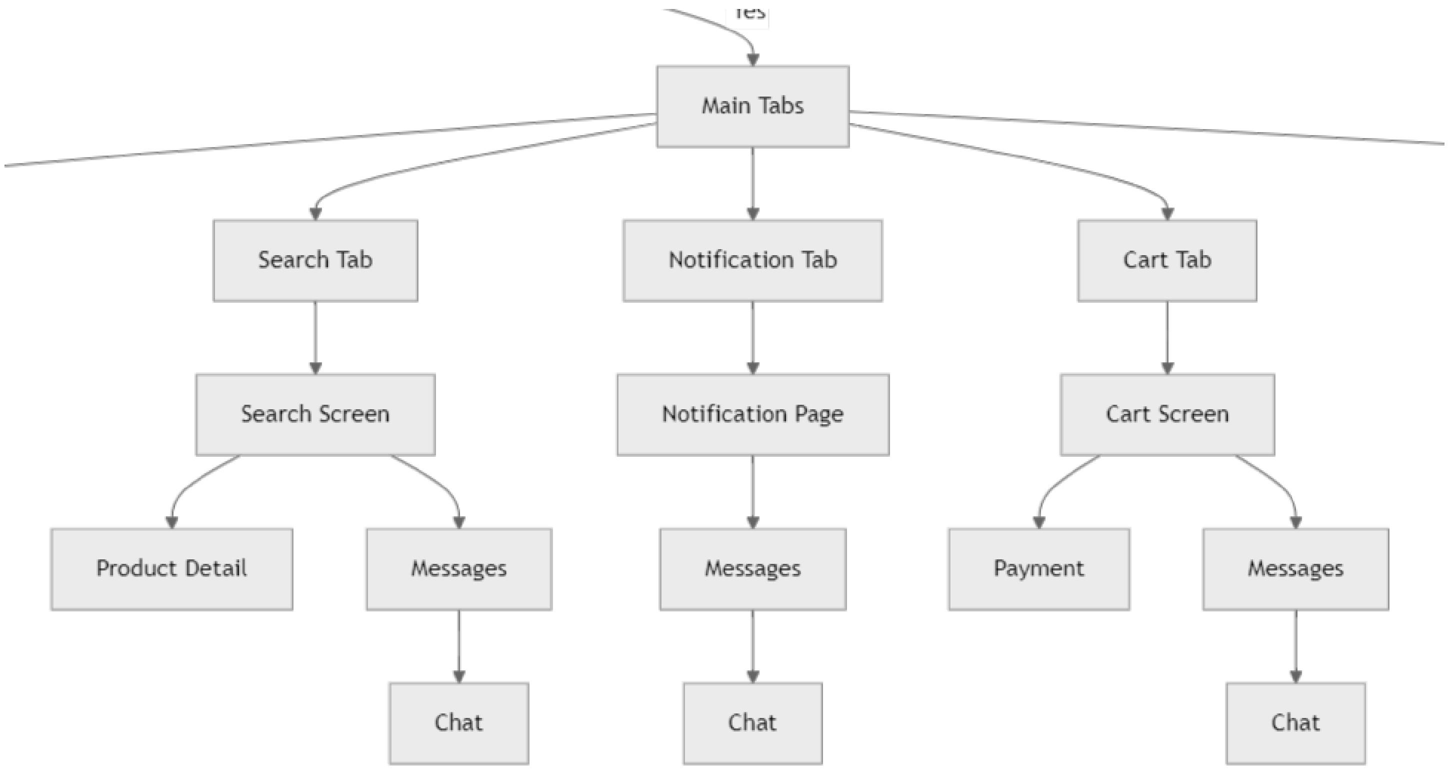
3.3.5 Dark Mode Support

- Toggle available in profile/settings.
- Inverts color scheme to reduce eye strain and battery usage.
- Applies to all pages including chat and product detail.

3.4. Navigation Flow







Explanation:

- **Authentication Screens**: Required before access to main features.

- **Home Screen**: Central hub displaying all product listings and navigation.
- **Search**: Quick access to filter and find items.
- **Product Detail**: Deeper look at a single listing; CTA buttons for chat/cart.
- **Chat**: Real-time communication between buyer and seller.
- **Profile**: User dashboard for managing personal data and listings.
- **Cart**: Stores selected products before purchase.

The user can return to **Home** or navigate directly to **Chat**, **Cart**, or **Profile** via bottom or side navigation.

3.5. Component Design (Optional Advanced Section)

Example Key Components:

Component	Description
ProductCard	Image, title, price, quick action icons
ChatBubble	Left/right aligned messages with timestamps
SearchBar	Live search with filter options
Modal	Used for product deletion confirmation, cart review
NavBar	Main navigation elements

3.6. Assets & Exports

- All design files are available in Figma under the project name: Student Marketplace UI/UX
- Export formats: .fig, .svg, .png, .pdf for developer handoff
- Design tokens (colors, spacing, typography) provided in style-guide frame

Chapter 4 : Architecture Document

4.1. Project Architecture Overview

The Student Marketplace is architected for clarity, scalability, and real-time interactivity. With a modular frontend powered by React and **Zustand** for global state management, it delivers smooth UX with modern UI patterns and RESTful API interactions. Real-time features like chat are supported using WebSocket or Firebase.

4.2. Folder Structure

```
/components     # ✅ Reusable UI components (e.g., ProductCard, ChatBubble)
/pages          # 📄 Route-based views (Home, ProductDetail, Profile, etc.)
/api            # 🔄 API interaction logic (e.g., productService, chatService)
/store          # 🧠 Zustand global state management files
/utils          # 🛠️ Helper utilities and functions (e.g., formatPrice)
/assets         # 🖼️ Static files: images, icons, logos
/navigation     # 🧭 Navigation configuration (React Router)
App.js          # 🚀 Main app entry point
```

Folder Breakdown

Folder	Description
components	Small, reusable building blocks for the UI
pages	Top-level page views tied to routes
API Backend	Backend communication (API calls)
store	Zustand stores for global state (e.g., userStore, cartStore)
utils	Formatting, validation, debounce, etc.
assets	Static content
navigation	Route definitions and guards

4.3. State Management

4.3.1 Global State – Zustand

Zustand is used to manage shared state across the application in a lightweight and intuitive way.

Examples of State Handled:

- **User authentication state**
- **Product listings and filters**
- **Cart contents**

4.4. Backend API Routes Documentation

This section describes the REST API routes implemented for the Student Marketplace application, including authentication, product management, orders, and messaging functionality.

4.4.1 Authentication Routes

HTTP Method	Endpoint	Description	Request Payload / Notes	Authentication Required
POST	/auth/google	Authenticate user via Google/GitHub OAuth	-	No
POST	/auth/login	Login user with email and password	{ email, password }	No
POST	/auth/signup	Register new user with optional profile image	Multipart form-data: user details + image file	No
PATCH	/auth/edit/:id	Update user profile, including image upload	Multipart form-data: updated fields + image file	No
GET	/auth/signout	Sign out current user	-	Yes
POST	/auth/sendOtp	Send OTP for user verification or password reset	{ email }	No
POST	/auth/VerifyOtp	Verify OTP submitted by user	{ email, otp }	No
POST	/auth/userExists	Check if user exists by email or username	{ email or username }	No
POST	/auth/forgot-password	Request password reset email	{ email }	No
POST	/auth/reset-password/:token	Reset password using token	{ newPassword }	No
GET	/auth/userInfo/:id	Get user profile information by user ID	-	No

GET	/auth/check-user-status	Check current user authentication status	-	Yes

Middleware:

- authenticateToken middleware protects sensitive routes like /check-user-status.

4.4.2 Order Routes

HTTP Method	Endpoint	Description	Request Payload / Notes	Authentication Required
POST	/order/create	Create a new order (with optional image)	Multipart form-data: order details + image file	Yes
GET	/order/user/:userId	Retrieve all orders made by a buyer	-	Yes
GET	/order/seller/:sellerId	Retrieve all orders for a seller	-	Yes
GET	/order/:orderId	Get details of a specific order by ID	-	Yes
PUT	/order/:orderId/status	Update the status of an order (e.g., shipped, delivered)	{ status }	Yes
PUT	/order/:orderId/cancel/:itemId	Cancel a specific item in an order	-	Yes

4.4.3 Product Routes

HTTP Method	Endpoint	Description	Request Payload / Notes	Authentication Required
POST	/product/	Create a new product with up to 5 images	Multipart form-data: product details + images (max 5 files)	Yes
GET	/product/	Get all products (supports filtering and pagination)	Query params: e.g., page, limit, category	No
GET	/product/search	Search products by keyword, category, etc.	Query params: q for query, other filters	No
GET	/product/:id	Get product details by product ID	-	No
GET	/product/seller/:id	Get all products posted by a specific seller	-	No
PUT	/product/:id	Update product details and images (up to 5)	Multipart form-data: updated details + images	Yes
DELETE	/product/:id	Delete a product by ID	-	Yes
POST	/product/:id/reviews	Add a review or comment to a product	{ comment, rating }	Yes

4.4.4 Messaging Routes

HTTP Method	Endpoint	Description	Request Payload / Notes	Authentication Required
POST	/message/send-message	Send a message (supports optional image)	Multipart form-data: { toUserId, messageText } + image	Yes
POST	/message/get-messages	Retrieve conversation messages between users	{ userId, partnerId }	Yes
PATCH	/message/mark-message/read/:id	Mark a message as read by message ID	-	Yes
GET	/message/conversation-partners/:id	Get a list of conversation partners for user	-	Yes

Notes on Middleware & Uploads

- **Authentication Middleware**: authenticateToken is applied to routes that require user login verification.
- **File Upload Handling**: multer is configured with memory storage and validates images with size limits (5MB max for product images).
- **Error Handling**: Image upload failures or validation errors are handled via custom middleware or controller logic.

Chapter 5. Development

5.1 Tools & Libraries

Tool / Library	Purpose
Expo / React Native	Framework for building cross-platform mobile app UI
Express.js API	Backend RESTful API server handling business logic
Firebase	Realtime database and authentication services
Socket.IO	Real-time bidirectional communication for chat features
Firestore	Cloud-hosted NoSQL database for syncing chat messages
Axios	HTTP client for API requests and responses
React Navigation	Routing and navigation management in React Native apps
FontAwesome Icons	Icon library for consistent, visually appealing UI icons

5.2 Key Features Implemented

1. Secure Login & Logout

- User authentication includes email/password and social login via Google/GitHub OAuth.
- JWT tokens secure user sessions, with token refresh and logout support.

2. Add/Edit/Delete Product Listings

- Users can create product listings with multiple images (up to 5), add detailed descriptions, set prices, and assign categories.
- Editing and deleting products is restricted to the product owner.

3. Smart Search & Category Filters

- Search functionality supports keyword queries and dynamic filtering by product categories.
- Efficient backend querying ensures fast, relevant results.

4. Stunning Product Detail Pages

- Detailed product views display images in a carousel, product description, price, seller info, and user reviews/comments.

5. Add to Cart Functionality

- Buyers can add products to their cart before purchasing.
- Cart supports item quantity adjustments and removal.

6. Real-Time Chat with Product Owner

- Integrated chat system using Socket.IO and Firestore for instant messaging.
- Supports sending text messages and optional image attachments.
- Chat history is persistently stored and synced in real-time.

7. Profile Editing

- Users can update their profile information including profile picture, contact details, and password.
- Profile changes are securely validated and updated on the backend.

8. Smooth UI Transitions

- Animations and transitions enhance user experience during navigation and interaction.
- Dark mode is supported with seamless toggling.

5.3 Development Highlights

- **Component-Based Architecture:** Reusable React Native components facilitate consistent design and faster feature development.
- **State Management:** Zustand is used for global state handling, simplifying app state synchronization across components.
- **API Integration:** Axios is utilized for making robust HTTP calls to the Express.js backend, ensuring reliable data exchange.
- **Error Handling:** Both frontend and backend include error detection and user-friendly feedback to improve reliability.
- **Responsive Design:** UI adapts smoothly to different device sizes and orientations, ensuring accessibility.

5.4 User Guide

1. Account Management

How to Sign Up

- Open the app and tap "Sign Up."
- Enter your email, create a password, and fill in your profile details.
- Optionally, upload a profile picture.
- Verify your email if prompted.
- You can also sign up using Google or GitHub OAuth.

How to Log In

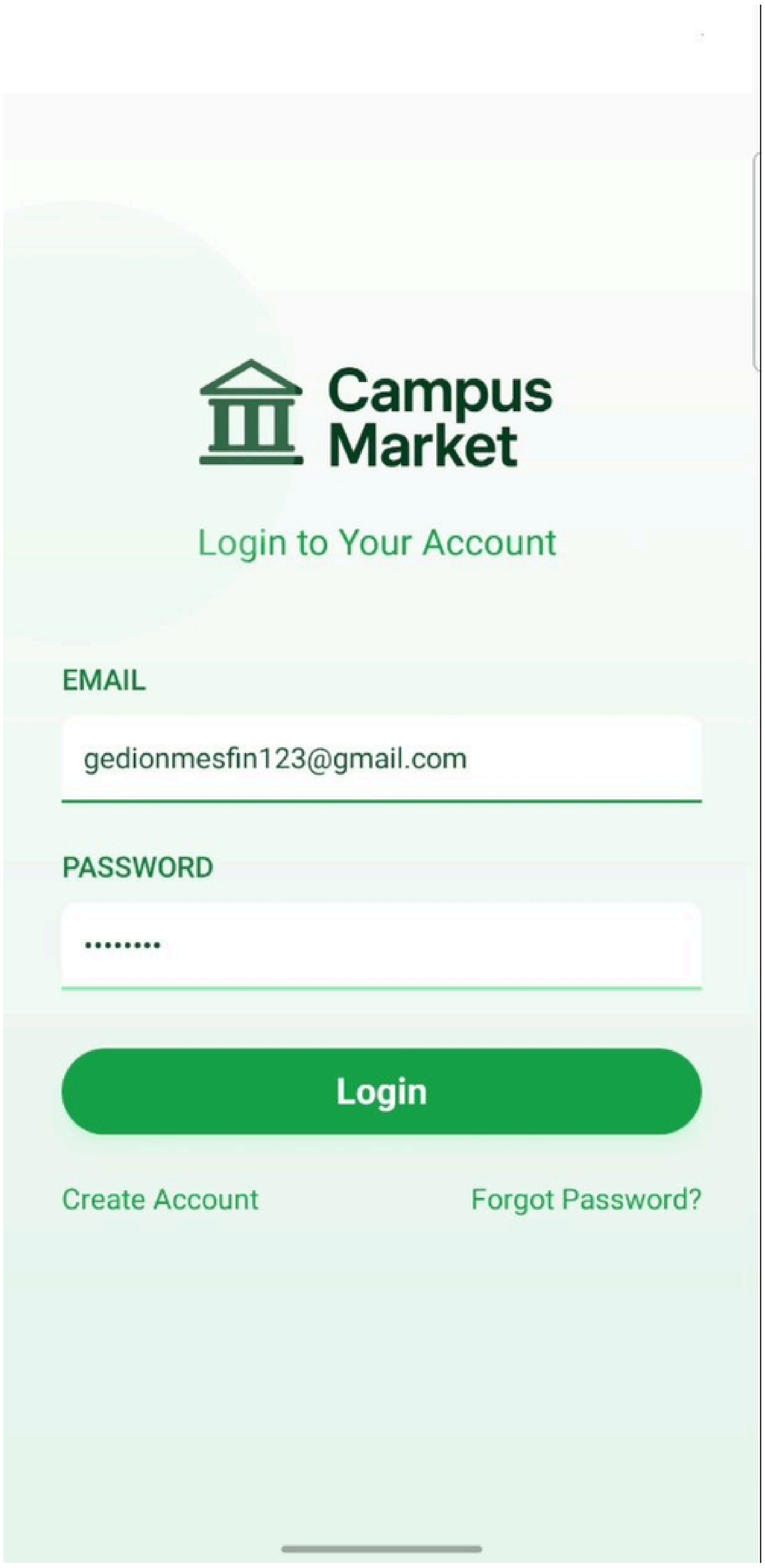
- Tap "Login" on the welcome screen.
- Enter your registered email and password, or use Google/GitHub login.
- If you forgot your password, tap "Forgot Password" to reset it.

How to Log Out

- Navigate to your Profile page.
- Tap the "Logout" button at the bottom.

How to Edit Your Profile

- Go to the Profile page.
- Tap "Edit Profile."
- Update your name, email, profile picture, or password.
- Save changes.



2. Posting Products

How to Create a New Product Listing

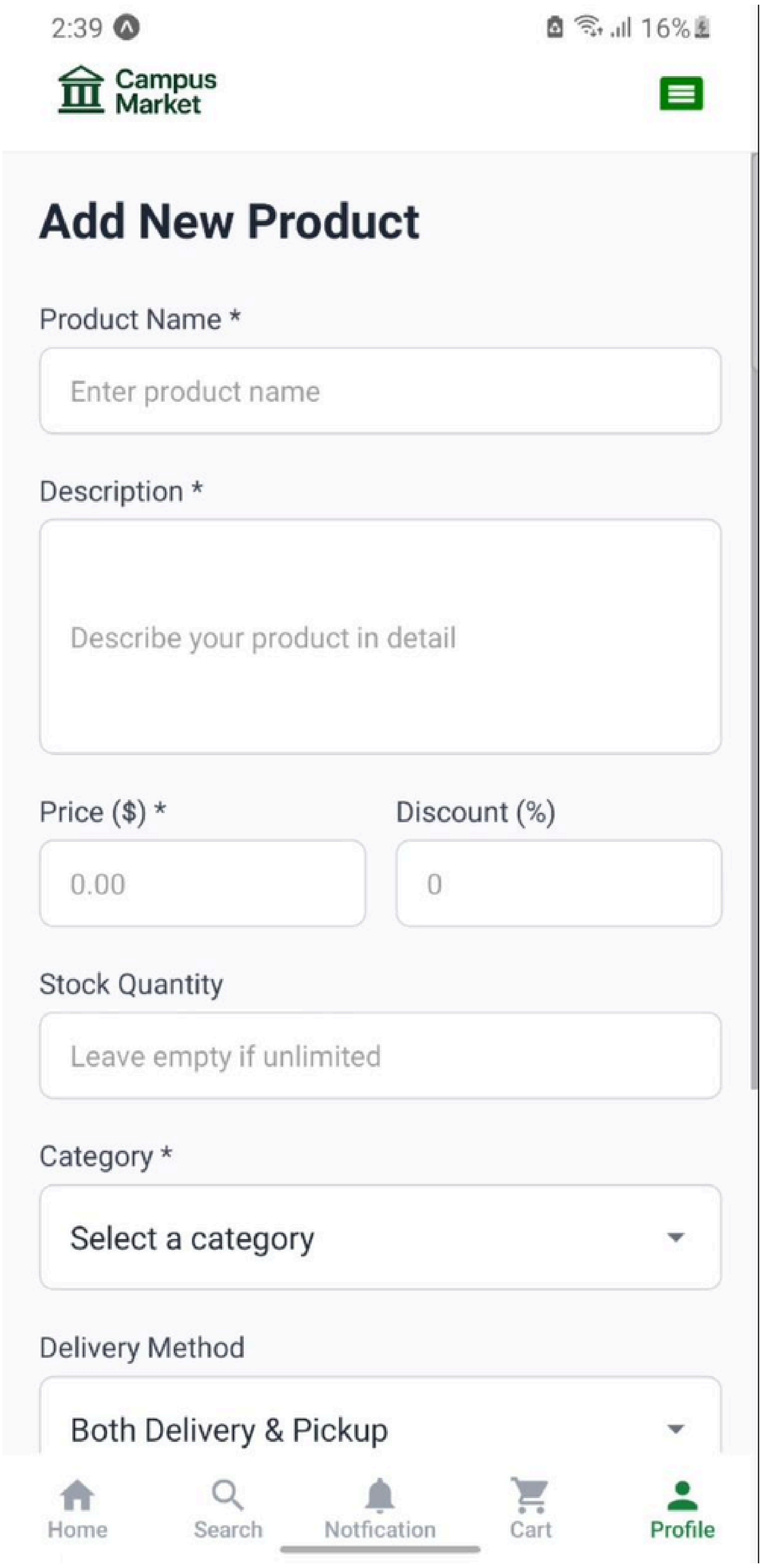
- From the Home or Profile screen, tap “Add Product.”
- Fill in product name, description, price, and category.
- Upload up to 5 product images.
- Review details and tap “Post.”
- Your product will be visible in the marketplace instantly.

How to Edit a Product Listing

- Navigate to your Profile and view your listed products.
- Select the product you want to edit.
- Tap “Edit.”
- Modify any details or update images.
- Save changes.

How to Delete a Product Listing

- On your product listing, tap “Delete.”
- Confirm to remove the product from the marketplace.



3. Browsing & Searching Products

How to Browse Products

- Open the Home screen to see latest and featured products.
- Scroll through products in various categories.

How to Search for Products

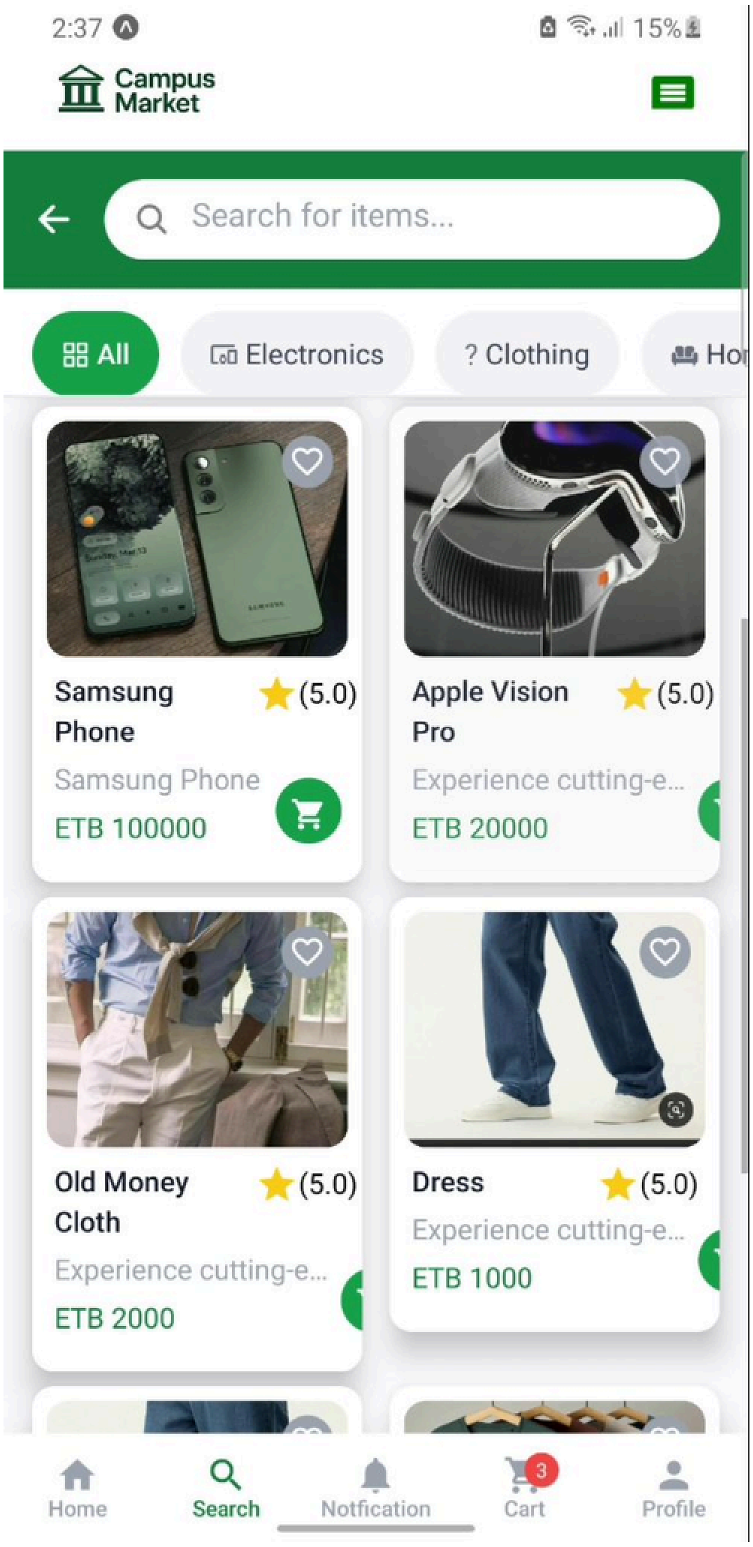
- Tap the Search icon in the navigation bar.
- Enter keywords related to the product you want.
- Results will update in real-time as you type.

How to Filter Search Results

- After searching, tap "Filter."
- Select category, price range, and condition filters.
- Apply filters to narrow results.

How to Sort Products

- Use sorting options like "Newest," "Price Low to High," or "Price High to Low."



Samsung Phone ★(5.0)
Samsung Phone
ETB 100000

Apple Vision Pro ★(5.0)
Experience cutting-e...
ETB 20000

Old Money Cloth ★(5.0)
Experience cutting-e...
ETB 2000

Dress ★(5.0)
Experience cutting-e...
ETB 1000

4. Viewing Product Details

How to View a Product

- Tap on any product card in the list or search results.
- You will see detailed info: description, price, seller info, and images.
- Swipe through product images in a gallery.

How to Contact the Seller

- On the product detail page, tap "Chat with Seller."
- Start a conversation instantly.

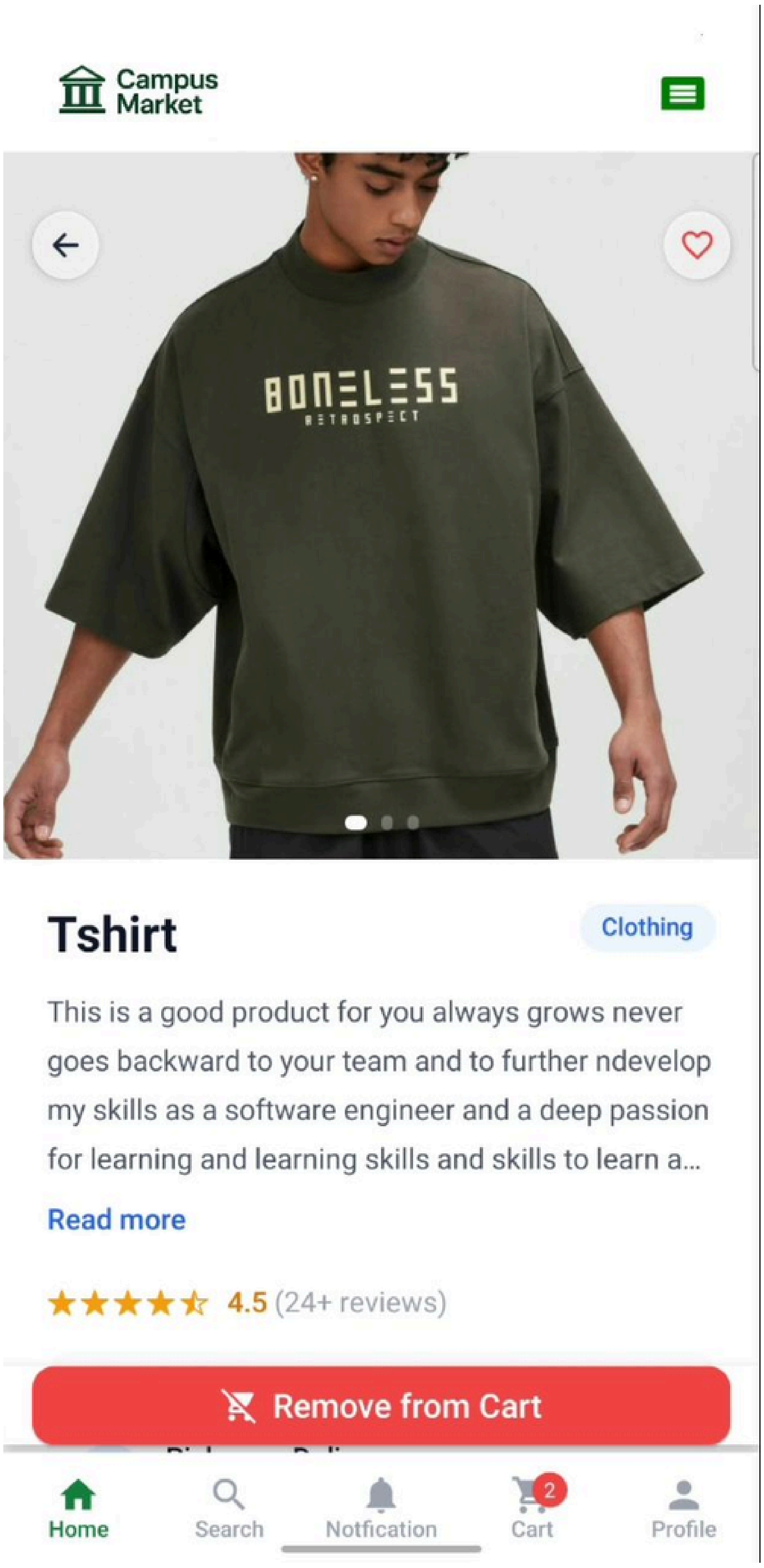
5. Notifications & Alerts

How to Receive Notifications

- Enable push notifications in your device settings.
- You'll get alerts for new messages, product updates, and orders.

How to Manage Notification Preferences

- In your Profile, go to Settings → Notifications.
- Toggle on/off specific notification types.



Tshirt

Clothing

This is a good product for you always grows never goes backward to your team and to further ndevelop my skills as a software engineer and a deep passion for learning and learning skills and skills to learn a...

Read more

★★★★☆ 4.5 (24+ reviews)

6. Shopping Cart & Purchasing

How to Add Products to Cart

- On any product detail page, tap “Add to Cart.”
- The product will be saved in your cart for later purchase.

How to View Your Cart

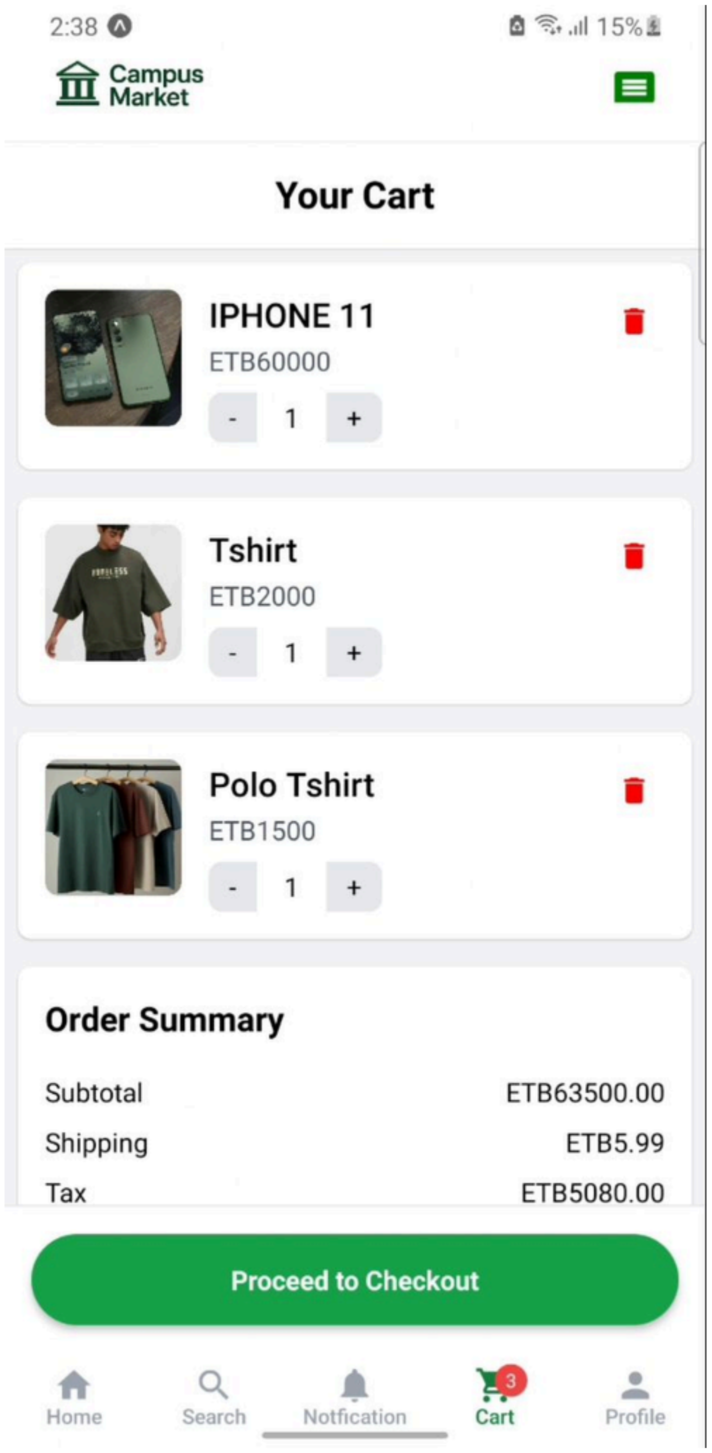
- Tap the Cart icon on the navigation bar.
- See all items you’ve added.

How to Remove Items from Cart

- In the Cart, swipe or tap “Remove” next to the item you don’t want.

How to Checkout

- Select items in the cart you want to buy.
- Proceed to payment (integration dependent).
- Confirm order and receive purchase confirmation.



2:38 15%

Campus Market

Your Cart

IPHONE 11
ETB60000
- 1 +

Tshirt
ETB2000
- 1 +

Polo Tshirt
ETB1500
- 1 +

Order Summary

Subtotal	ETB63500.00
Shipping	ETB5.99
Tax	ETB5080.00

Proceed to Checkout

Home Search Notfication Cart 3 Profile

7. Messaging & Communication

How to Start a Chat with a Seller

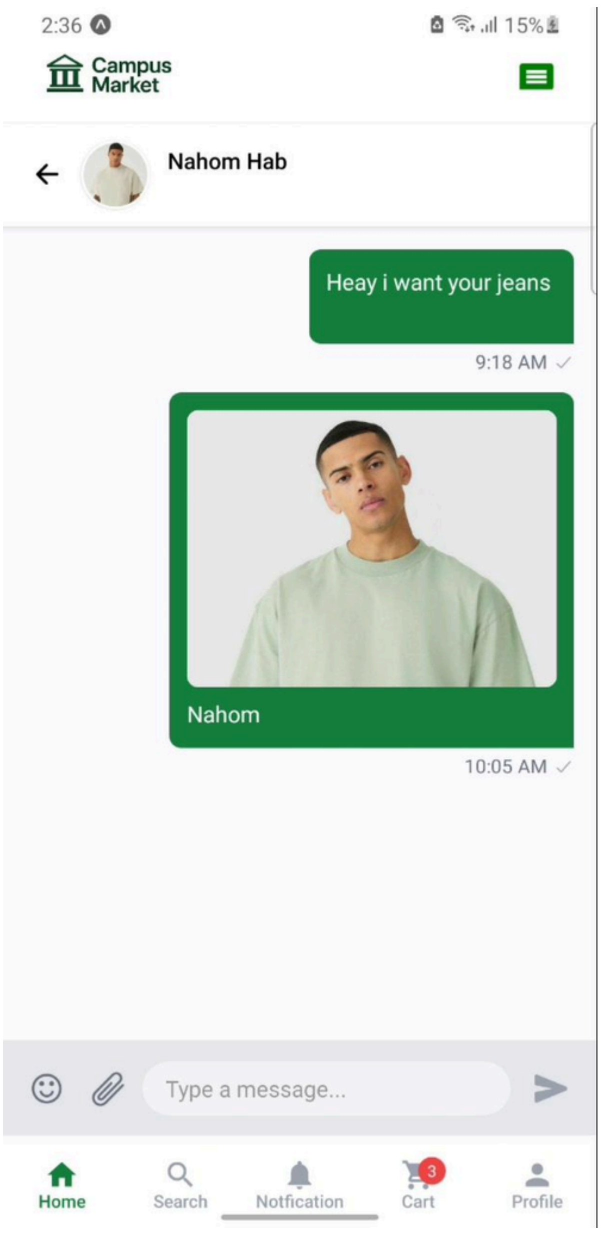
- Go to a product detail page.
- Tap “Chat with Seller.”
- Send messages or images.
- Notifications alert you to new messages.

How to View Existing Chats

- Access the “Messages” tab in the app.
- Tap any conversation to continue chatting.

How to Mark Messages as Read

- Messages automatically mark as read when you open the chat.



Heay i want your jeans

9:18 AM

Nahom

10:05 AM

8. Help & Support

How to Access Help

- Tap the “Help” or “Support” section in the menu.
- Browse FAQs or contact support via chat or email.

How to Report a Problem

- Use the “Report” button on product pages or chat screens.
- Describe the issue and submit.

9. Security & Privacy

How to Change Your Password

- Go to Profile → Security Settings → Change Password.
- Enter current and new passwords to update.

How to Enable Two-Factor Authentication (If Available)

- In Security Settings, enable 2FA via authenticator app or SMS.

10. Miscellaneous

How to View Your Purchase History

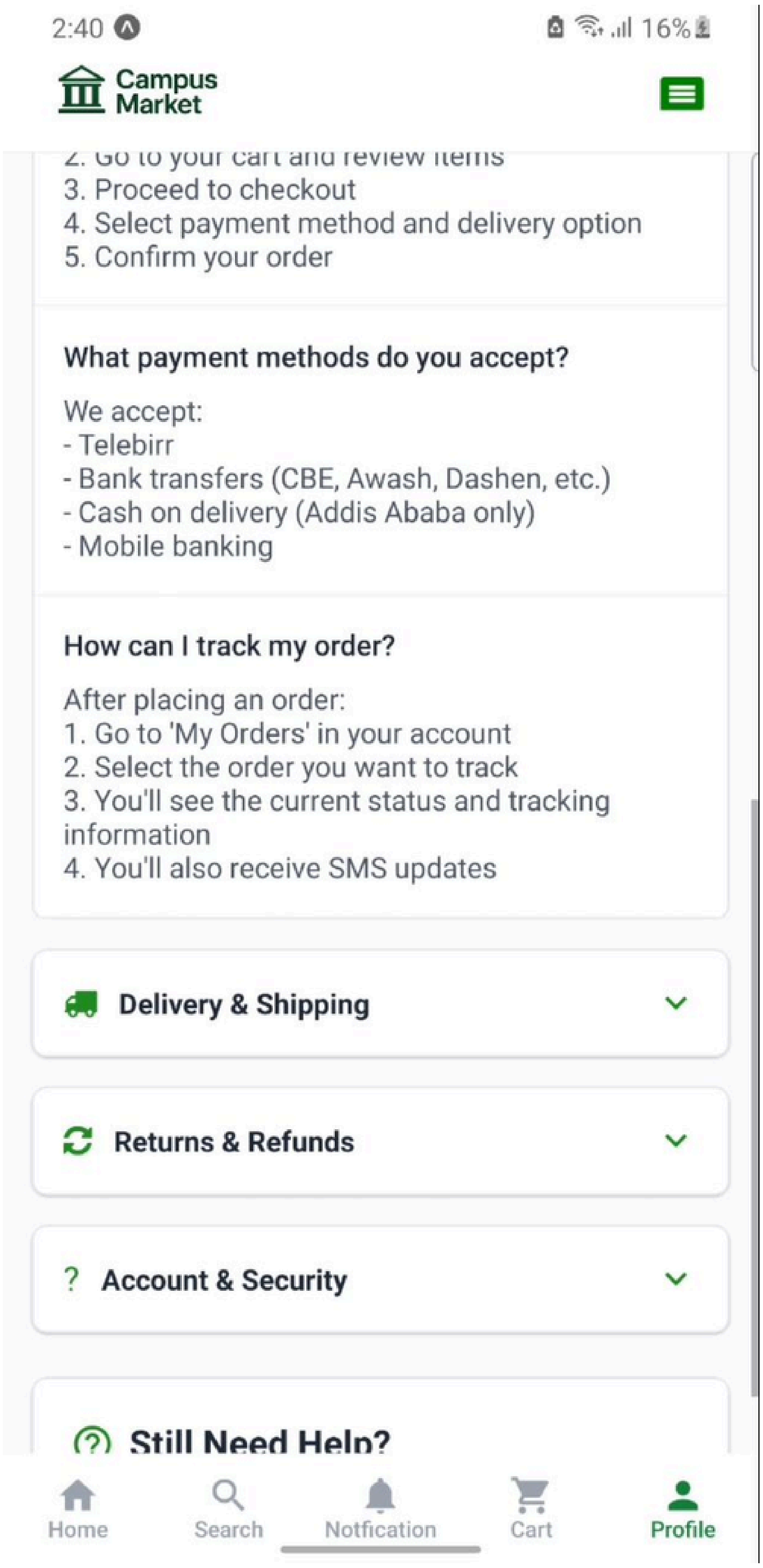
- Navigate to Profile → Orders.
- Review past orders with details.

How to View Your Sales History

- In Profile → Sales, see your sold products and earnings.

How to Log in Using Google or GitHub

- On the login screen, tap “Continue with Google” or “Continue with GitHub.”



2. Go to your cart and review items
3. Proceed to checkout
4. Select payment method and delivery option
5. Confirm your order

What payment methods do you accept?

We accept:
- Telebirr
- Bank transfers (CBE, Awash, Dashen, etc.)
- Cash on delivery (Addis Ababa only)
- Mobile banking

How can I track my order?

After placing an order:
1. Go to 'My Orders' in your account
2. Select the order you want to track
3. You'll see the current status and tracking information
4. You'll also receive SMS updates

- Follow the prompts to authorize.

Chapter 6 : Future Enhancements

As the student marketplace grows, the following features are planned to improve user experience, increase engagement, and streamline platform management:

1. Integrated Payments

Objective: Enable users to securely complete purchases directly within the app.

Features:

- Support for multiple payment methods (credit/debit cards, mobile money, digital wallets).
- Secure payment gateway integration with encryption and fraud detection.
- Order tracking and receipts generation.
- Refund and dispute management process.

Benefits:

- Simplifies the buying process, reducing reliance on external communication.
- Increases trust and convenience for buyers and sellers.

2. Ratings and Reviews for Sellers

Objective: Build trust and community credibility by allowing buyers to rate and review sellers and products.

Features:

- Star rating system and written feedback on completed transactions.
- Display average ratings on seller profiles and product listings.
- Review moderation to prevent abuse and spam.
- Seller response option to reviews.

Benefits:

- Helps buyers make informed decisions.
- Encourages sellers to maintain high-quality standards.

3. Notification System for Messages and Purchases

Objective: Keep users informed with real-time alerts about important activities.

Features:

- Push notifications for new messages, purchase updates, cart reminders, and promotions.
- In-app notification center for easy access to alerts history.
- User preferences for enabling or disabling specific notification types.

Benefits:

- Enhances engagement and responsiveness.
- Improves communication between buyers and sellers.

4. Admin Dashboard for Managing Listings

Objective: Provide platform administrators with tools to oversee and manage the marketplace effectively.

Features:

- View and moderate product listings, user accounts, and reports.
- Approve or reject products to maintain quality standards.
- Analytics dashboard showing platform usage, sales, and user activity.
- Tools to handle disputes, ban users, and manage categories.

Benefits:

- Ensures platform safety, reliability, and compliance.
- Provides insights for continuous improvement.

5. Multi-Language Support

Objective: Expand accessibility and usability for a diverse student community.

Features:

- Localization of UI text and notifications into multiple languages.
- Language selector in user settings or on the welcome screen.
- Support for RTL (Right-to-Left) languages where applicable.

Benefits:

- Increases inclusivity and user base growth.
- Enhances user experience for non-English speakers.