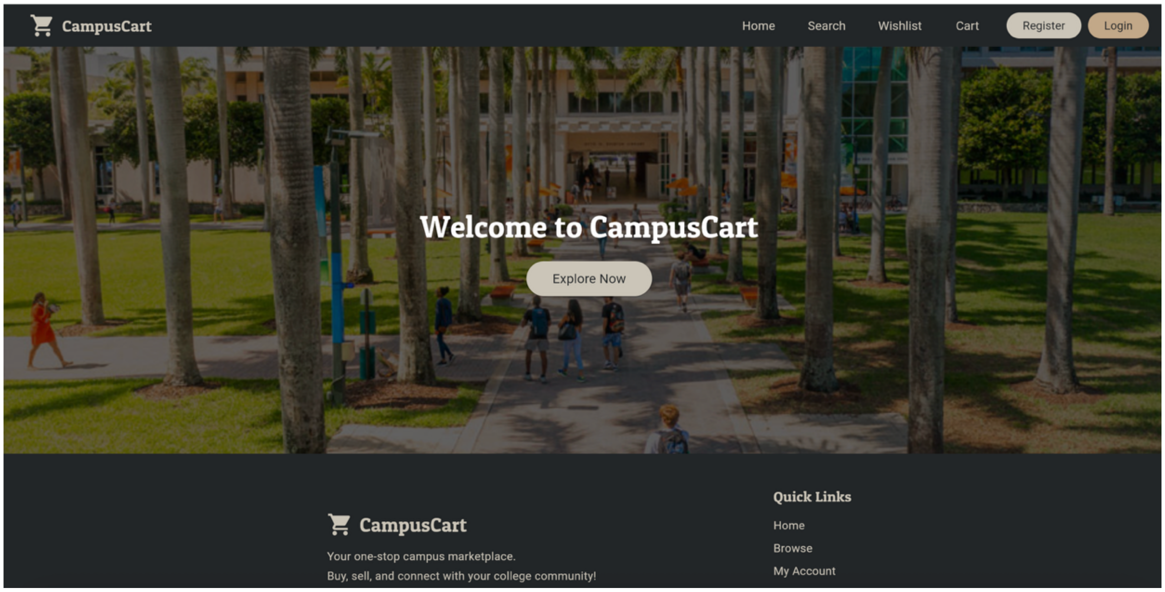
CampusCart – Project Documentation

Team: Jaime A. Perez Selman, Jessica Nunez Hernandez, Miguel Vega, Julio Mendez, Marcos Paiva

1. Project Overview



Your one-stop campus marketplace.
Buy, sell, and connect with your college community!

CampusCart is a web-based platform developed using **Flutter** and **Supabase**, designed to streamline the food ordering experience on university campuses. It serves as a bridge between students and campus food vendors by enabling mobile ordering, real-time inventory management, and digital storefronts for small food stalls.

This solution empowers **students** with a convenient way to browse, search, and order meals online, while providing **vendors** with tools to manage inventory, track orders, and increase efficiency.

2. Project Objectives

✅ Provide a **seamless digital ordering platform** for students.
✅ Enable **real-time inventory updates** for vendors.
✅ Reduce wait times at stalls by allowing **mobile ordering and pickups**.

✅ Offer a **role-based platform** with dedicated interfaces for students and vendors.
✅ Support **guest browsing** for students while enforcing vendor login for security.

3. Key Features

Student Features

- Browse stalls and products with a clean UI.
- **SearchView:** Full-text search for products across stalls.
- **Cart Management:** Add/remove products before checkout.
- **Wishlist:** Save products (requires login; unavailable to guests).
- Guest browsing available for students.

Vendor Features

- **Vendor Dashboard:** Manage inventory, add/edit products, and track stock.
- **Order Management:** View and fulfill real-time incoming orders.
- Vendors **must log in** to access their dashboard (no guest mode).

Authentication & Roles

- **Supabase Auth** for secure registration/login.
- app_users table with **role-based control** (student, vendor).
- Vendors require login; students can browse as guests.

4. Technology Stack

Component	**Technology**
Frontend	Flutter (Web support enabled)
Backend	Supabase (PostgreSQL-based)
Authentication	Supabase Auth (email/password)
Database	Supabase Database (relational schema)

Version Control	GitHub
Workflow	Agile (Scrum – 2-week sprints)

5. Database Schema

Tables

1. **app_users**
   - Stores user details (id, name, email, role)
2. **stalls**
   - Represents vendor stalls (id, vendor_id, stall_name)
3. **products**
   - Stores product info (id, stall_id, name, price, stock)
4. **orders**
   - Order metadata (id, student_id, total, status, created_at)
5. **order_items**
   - Items in an order (id, order_id, product_id, quantity, subtotal)

🔗 **Relationships:**

- stalls ↔ products → One-to-Many
- app_users ↔ orders → One-to-Many (student side)
- orders ↔ order_items → One-to-Many

6. System Architecture

- **Frontend:** Flutter web app with modular views (Home, Search, Cart, Wishlist, Vendor Dashboard).
- **Backend:** Supabase database with REST APIs for CRUD operations.
- **Authentication:** Email/password-based login and registration.
- **State Management:** Flutter's StatefulWidget and Provider where necessary.

7. UI/UX Design

- **Consistent Navigation Bar:**
    - Logo (left), Menu items (center), Rounded Register/Login buttons (right).
- **Views Implemented:**
    - **HomeView:** Landing page showing stalls.
    - **Student Home View:** Same as HomeView but with user account dropdown.
    - **SearchView:** Displays products with full-text search and stall info.
    - **WishlistView:** Displays saved products (logged-in students only).
    - **CartView:** Displays selected items for checkout.
    - **Vendor Dashboard:** Product and order management UI.

8. Agile Development Timeline

Sprint	**Deliverables**
Sprint 1	Supabase setup, database schema creation, authentication integration
Sprint 2	HomeView, Login/Register views, user role handling
Sprint 3	Vendor Dashboard layout and product CRUD
Sprint 4	SearchView, Cart, Wishlist, Navbar routing
Sprint 5	Final wiring, bug fixes, preparation for showcase deliverables

9. Setup Instructions

To Run the App Locally

1. Clone the repository from GitHub.
2. Install dependencies:

```
flutter pub get
```

3. Configure Supabase keys in `.env` or constants file.
4. Run the app:

```
flutter run -d chrome
```

Database Setup

- Import the provided SQL script into Supabase.
- Ensure that authentication triggers link Supabase Auth users to the app_users table.

10. Testing & Validation

✅ Authentication tested with valid and invalid credentials.
✅ Role-based routing verified (students vs. vendors).
✅ Product CRUD and inventory updates tested.
✅ Search, Cart, and Wishlist functionalities validated.

11. Final Deliverables

✅ **Source Code:** GitHub repository with final codebase.
✅ **Documentation:** This document.
✅ **Introduction Video:** Overview of CampusCart goals and benefits.
✅ **Demo Video:** Walkthrough of student and vendor workflows.
✅ **Final Deliverable Zip:** Containing all of the above.

12. Future Improvements

- Integration of payment gateway for online payments.
- Push notifications for order status updates.
- Analytics dashboard for vendors to track sales and trends.
- Multi-language support for international students.