Harvest Hub: Project Report

Date: May, 12, 2025 **Project:** Harvest Hub - Mobile Application

1. Introduction & Overview

This report outlines the current status, features, and technical architecture of the Harvest Hub mobile application. Our goal with Harvest Hub is to provide users with a seamless and enjoyable experience for browsing and purchasing fresh produce and other farm-to-table goods directly through their mobile devices. We've focused on building a user-friendly interface backed by a robust and scalable backend using Firebase.

2. Core Features Implemented

We're pleased with the range of functionalities currently live in the application:

- **User Authentication & Profile Management:**
  - Secure user registration and login via email and password, integrated with Firebase Authentication.
  - Input validation on signup and login forms to guide users.
  - A dedicated Profile screen where users can view their (mock) username and email, and log out. User data (username, email) is stored in Firestore.

- **Product Discovery & Browsing:**
  - **Home Screen:**
    The central hub featuring a dynamic layout with:
    - App branding (Home Icon).
    - A functional search bar (HomeSearch) to filter products by name.
    - Promotional banners (HomeBanner).
    - Categorized product carousels (e.g., "Exclusive Offer," "Best Selling") using the ProductCarousel component.

  - **Product Details Screen:**
    Tapping on a product reveals a comprehensive details screen showing:
    - Product image, name, price, weight, and pieces.
    - Quantity selection controls.
    - An "Add to Basket" feature.
    - Expandable sections (DropBox) for product details, nutrition, and reviews (currently using static data).

- **Shopping Cart & Checkout:**
  - **Robust Shopping Cart (CartScreen & CartContext):**
    - Users can add products from the Product Details screen or directly from carousels.
    - The cart displays all items with options to adjust quantity or remove items.

    - The cart total is dynamically calculated.
    - **Firestore Synchronization:** For logged-in users, the cart contents are persistently stored and synced with a currentCart document in their user-specific collection in Firestore. This allows for a consistent cart experience across sessions.
    - Option to clear the entire cart.

- **Checkout Process (CheckoutScreen):**
    - A multi-step (simulated on one screen for now) checkout process where users can:
        - Review their order summary.
        - Enter shipping details (name, phone, address, city, postal code) with basic validation.
    - The "Place Order" button triggers the order creation.

- **Order Placement & Confirmation:**
    - **Firestore Orders:** Upon successful checkout, an order document is created in the main orders collection in Firestore. This document includes user ID, items, total amount, shipping details, and a timestamp.
    - The user's Firestore cart is cleared after a successful order.
    - **Order Confirmation Screen (OrderConfirmationScreen):** Displays a success message with the order ID and total amount, offering options to continue shopping or view their profile.

- **Store Locator (MapScreen):**
    - An integrated map view (using react-native-maps) displaying predefined store locations with markers.
    - Includes a custom header for the screen.

- **User Interface & Experience:**
    - Consistent styling using a predefined color palette (MyColors.js) and custom fonts (MyFonts.jsx).
    - Responsive design elements used in components like HomeBanner and HomeIcon.
    - Custom, reusable components like CustomTextInput (with error handling and password visibility toggle) and CustomButton (with loading and disabled states).
    - Clear navigation structure using a bottom tab navigator for main sections (Home, Locations, Cart, Profile) and a stack navigator for deeper navigation (e.g., Details, Checkout).
    - User feedback through alerts (showAlert helper) for actions like adding to cart, login errors, etc.

3. Technical Overview

- **Framework:** React Native (with Expo)
- **Backend & Database:** Firebase (Authentication, Firestore)
- **Navigation:** React Navigation (Native Stack & Bottom Tabs)
- **State Management:**
    - Component-level state (useState).
    - Context API (CartContext) for global cart state management and Firestore sync.

- **Key Libraries:**
  - @expo/vector-icons for iconography.
  - react-native-maps for map functionalities.
  - prop-types for component prop validation.
  - react-native-responsive-dimensions for responsive UI.

- **Code Structure:**
  - Organized into Screens, Components, Utils, Context, and assets.
  - Helper functions, constants, and theme files are separated for maintainability.

4. Screen Interactions Flow

The application flow is designed to be intuitive:

1. **Splash Screen:** Initial loading and font/auth state check.
2. **Authentication:** Users are directed to Login or Signup if not authenticated, or to the Main App if already logged in.
3. **Main App (Tabs):**
   - **Home:** Browse products, search, navigate to Product Details.
   - **Locations:** View store map.
   - **Cart:** Manage cart items, proceed to Checkout.
   - **Profile:** View user info, access (future) settings/order history, logout.

4. **Product Details:** View detailed product information, add to cart.
5. **Checkout & Order Confirmation:** Complete the purchase process.

5. Potential Future Enhancements & Considerations

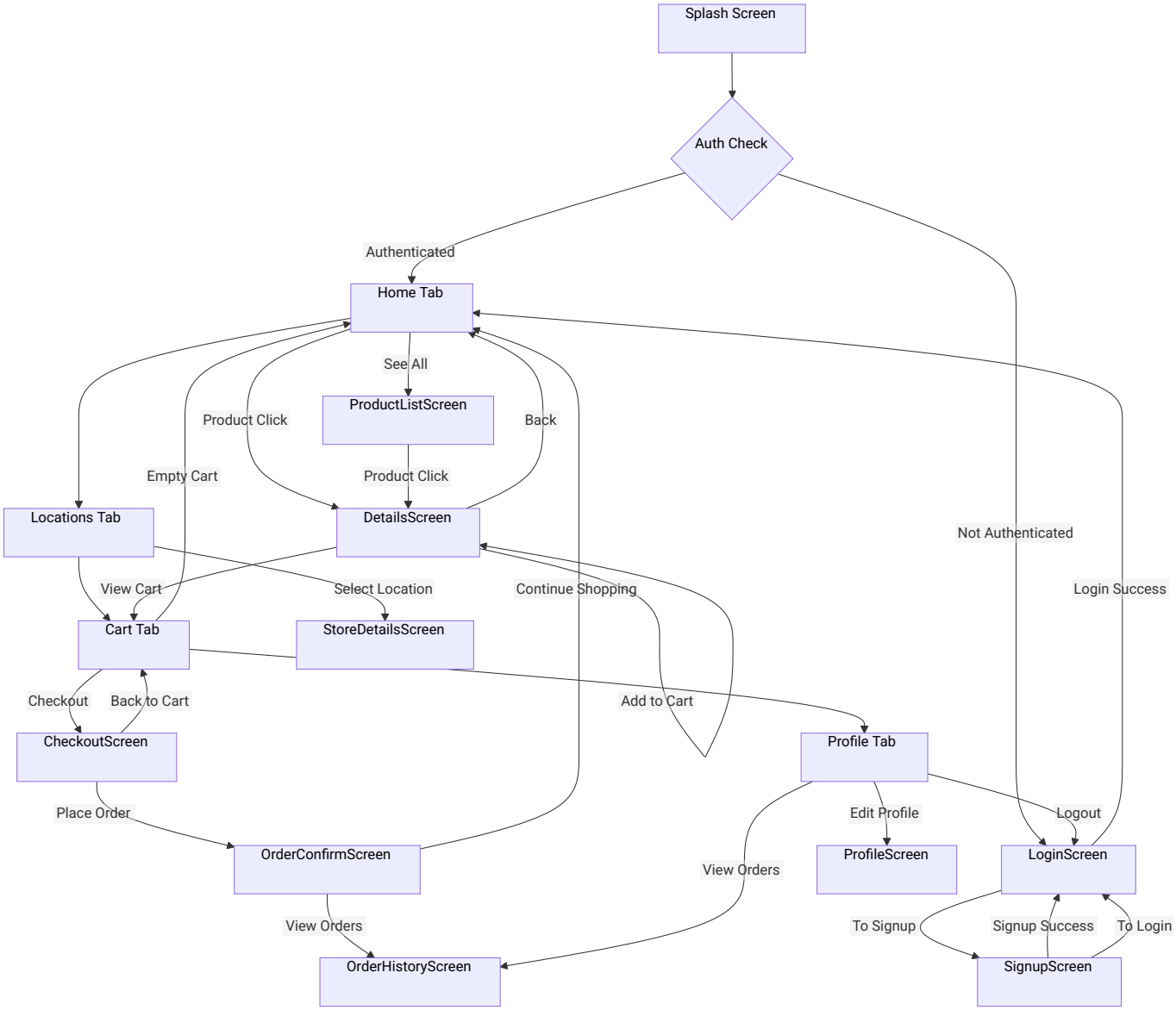
While the current version is functional, we've identified several areas for future development:

- **Full Order History:** Allow users to view past orders from their profile.
- **Payment Gateway Integration:** Implement real payment processing.
- **Push Notifications:** For order status updates, promotions, etc.
- **User Reviews & Ratings:** Allow users to submit and view product reviews.
- **Advanced Search & Filtering:** More sophisticated product discovery options.
- **User Address Book:** Save multiple shipping addresses.
- **Admin Panel:** For managing products, orders, and users.
- **Performance Optimizations:** Further image optimization and lazy loading for large lists.
- **Offline Support:** Basic browsing or cart functionality when offline.
- **Testing:** More comprehensive unit and integration testing.

6. Conclusion

The Harvest Hub application has made significant progress, establishing a solid foundation with

core e-commerce functionalities and a reliable backend. The integration of Firebase for authentication, Firestore for data persistence (especially for the user's cart and orders), and a well-structured React Native frontend positions the project well for future expansion and refinement. We're confident that Harvest Hub offers a valuable and user-friendly platform for its intended audience.


Splash Screen
Auth Check
Authenticated
Not Authenticated
Home Tab
See All
ProductListScreen
Product Click
Back
Empty Cart
Product Click
Locations Tab
DetailsScreen
View Cart
Select Location
Continue Shopping
Login Success
Cart Tab
StoreDetailsScreen
Checkout
Back to Cart
Add to Cart
CheckoutScreen
Profile Tab
Place Order
Edit Profile
Logout
OrderConfirmScreen
ProfileScreen
LoginScreen
View Orders
View Orders
To Signup
Signup Success
To Login
OrderHistoryScreen
SignupScreen