Haven - Collaborative Interior Design

Haven is a mobile AR application that helps users visualize furniture in their physical spaces before making purchasing decisions.

Setup for Testing

Prerequisites

Before you begin, ensure you have the following installed:

- Node.js
- npm
- Expo CLI
- iOS Development (for iOS builds):
    - macOS
    - Xcode
    - CocoaPods

Installation Steps

1. **Clone the repository**
   git clone https://github.com/reidmccaw/haven_app.git
   cd haven_app
2. **Install dependencies**
   npm install
3. **Install iOS dependencies (if building for iOS)**
   cd ios
   pod install
   cd ..

Running the App

Start the Expo development server

- npm start

Using Expo Go Does Not Work For AR Features

- Important: This app uses custom native modules (ViroReact for AR) that are not compatible with Expo Go. You must build a native app to use AR features.
- The app will not load if launched with Expo Go.

Building for Production

Local Build with Xcode

1. Generate native iOS project (if not already generated):
2. npx expo prebuild --platform ios
3. Open in Xcode:
4. open ios/Haven.xcworkspace
5. For physical device (iPhone):
    - Connect your iPhone to your Mac via USB cable
    - Trust the computer on your iPhone (if prompted)
    - In Xcode, select your connected device from the device dropdown
    - Select your development team
    - Click "Run" or press Cmd + R

Note: AR features require a physical device - the simulator does not have a camera.

App Purpose & Use Cases

Haven allows users to preview how furniture would look in their home before buying. The app is used when:

- Shopping for furniture online or in-store
- Redesigning or decorating a room or house
- Ensuring furniture fits and matches existing decor
- Needing real feedback on your designs from people you trust

3 Key Tasks Users Can Accomplish

1. Use AR to place furniture and view a layout
4. View feedback comments and make changes
5. Organize feedback on your project form multiple

Limitations

- AR Requirements: Requires a modern smartphone with AR capabilities
- Lighting Conditions: AR tracking works best in well-lit rooms with textured surfaces
- Surface Detection: May struggle to detect surfaces on very dark, reflective, or uniform surfaces
- Furniture Library: Limited to pre-loaded furniture models; users cannot upload custom furniture or search for new options
- Scale Accuracy: Virtual furniture sizing may not be perfectly accurate to real-world dimensions
- Performance: May experience lag on older devices or when moving the screen rapidly around

Accessibility Considerations & Limitations

Current Accessibility Features

- Large, high-contrast buttons for clear actions
- Clear visual feedback for interactions and haptics
- Simple, intuitive navigation flow

Accessibility Limitations

- No Screen Reader Support: AR visualization is inherently visual and not currently compatible with any voice application or readback
- Visual Dependency: App relies entirely on visual AR feedback with no audio or text alternatives
- Color Contrast: Some UI elements may not be clearly visible depending on lighting and the AR scene itself