Buy at Brown

Creating a platform for students to buy and sell within the Brown community

back home

overview

features

tech stack

challenges & solutions

demo video

takeaways

role

Front-End / Back-End Engineer & Product Designer

technologies

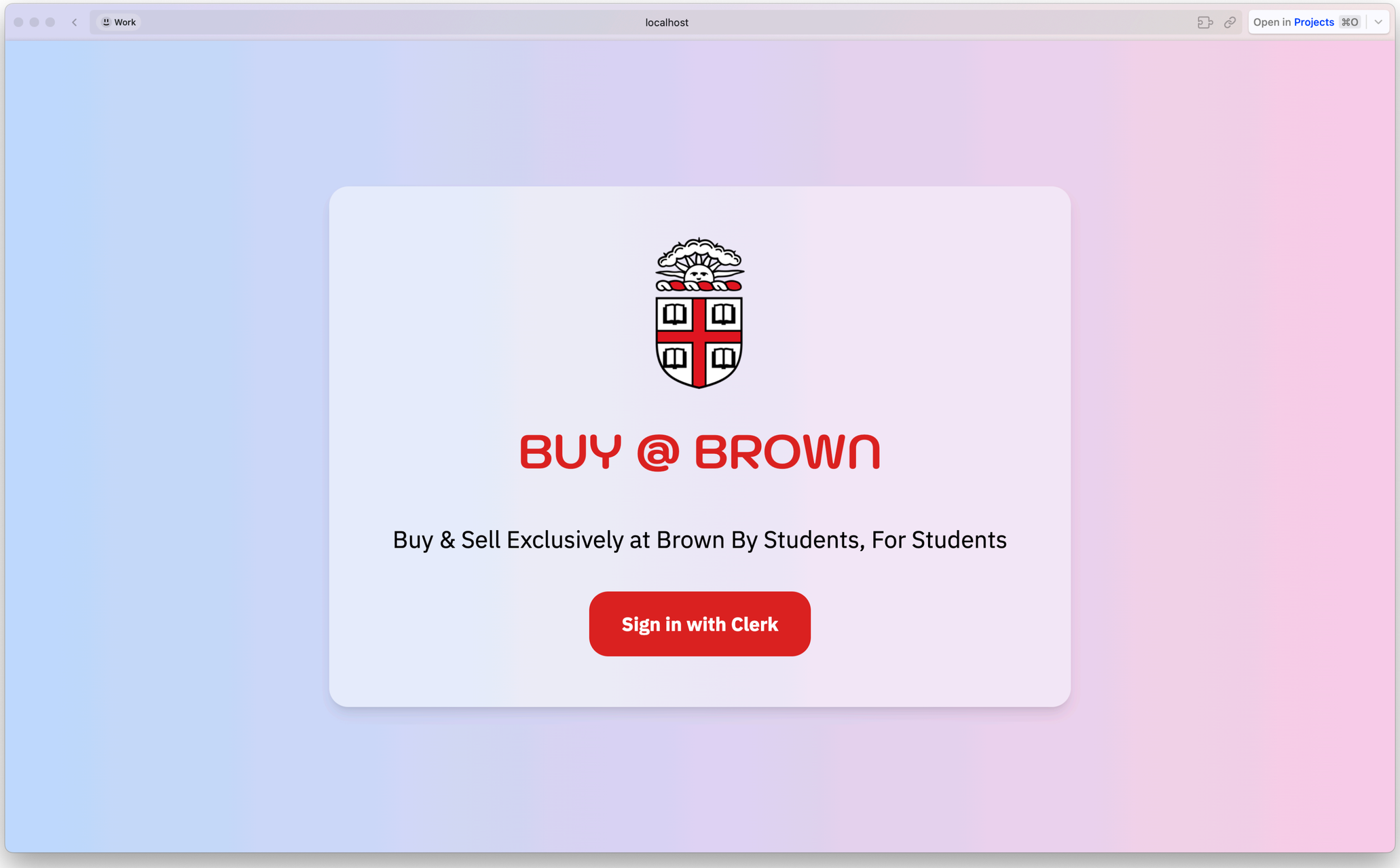
React Figma Firebase Java Typescript Clerk

timeline

November - December 2024

team

Jacob Solano
Claire Zhang
Bisola Folarin
Kayleen Cruz



overview

Buy at Brown: A Marketplace for Brown Students

Our team developed a full-stack web application using a tech stack combining a Java backend (Spark), a Firebase Firestore database, and a React + TypeScript frontend built with Vite. The app implements an architecture where the Java server acts as the middleware between the frontend and the database, delivering data access via RESTful APIs.

Buy at Brown is a web application that allows Brown students to buy and sell items within the Brown community. Users can create an account, post items for sale, and browse items for sale. Users can search for items by various categories and tags. Users have their own accounts with custom usernames, and they can favorite, list, and edit their own items. Users can contact sellers through the contact information provided within the item listing.

Visit the Github Repository

features

User Authentication

Secure login system that verifies Brown University email addresses, ensuring that only current students can access the platform.

Item Listings

Easy-to-use interface for creating, viewing, and managing product listings with image upload capabilities, descriptions, price setting, and category selection.

Search & Filter

Robust search functionality with filters for categories, price ranges, and condition to help students quickly find what they're looking for.

Messaging System

Built-in messaging that allows buyers and sellers to communicate directly through the platform to coordinate details and arrange meetups.

Saved Items

Ability to bookmark and save interesting listings for later, making it easier to track and compare options before making a purchase decision.

User Profiles

Detailed user profiles showing transaction history, active listings, and reviews to build trust within the community marketplace.

tech stack

Frontend

- React with TypeScript for type-safe development
- React Router for seamless navigation between pages
- CSS modules for component-scoped styling
- Responsive design for desktop and mobile usage

Backend

- Firebase Authentication for secure user management
- Cloud Firestore for real-time database functionality
- Firebase Storage for image and file storage
- Firebase Cloud Functions for server-side operations

DevOps & Tools

- Git/GitHub for version control and collaboration
- Firebase Hosting for deployment
- Jest for unit and integration testing
- Figma for UI/UX design and prototyping

challenges & solutions

Backend-to-Frontend Integration

Challenge: How do we connect a database controlled by a Java server in the backend to the React frontend user interface?

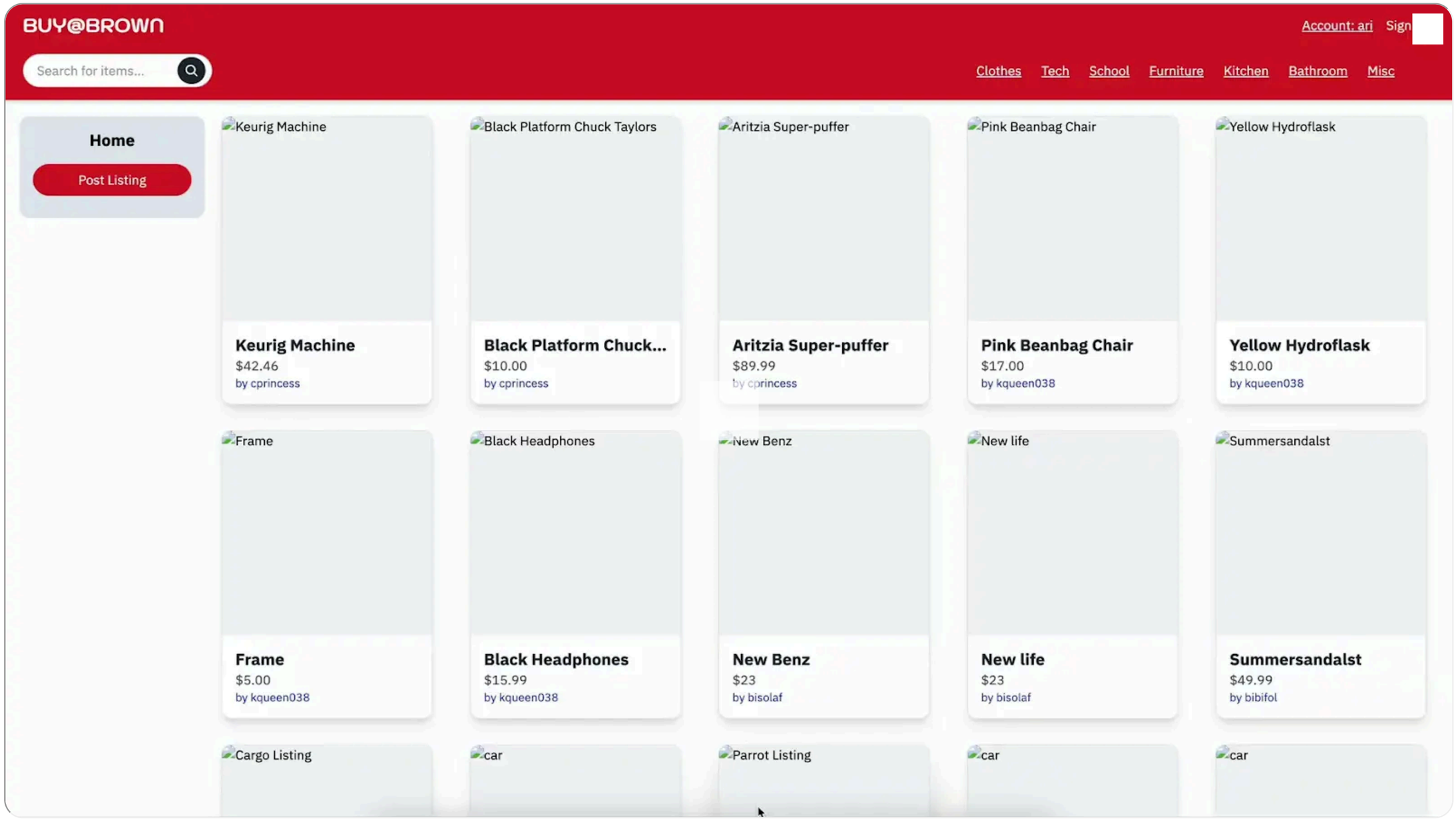
Solution: We implemented a RESTful architecture where the Java backend securely mediates all data interactions between the frontend and Firebase. The frontend uses api.ts to send HTTP requests to custom route handlers defined in Server.java (using Spark). These handlers call methods in FirebaseUtilities.java to read and write data in Firestore, returning JSON responses to the frontend. This approach allowed us to keep Firebase credentials hidden while maintaining control over data flow.

User-Friendly Navigation

Challenge: How do we create an interface that is intuitive and easy to navigate for new users?

Solution: We focused on clean UI design in React and implemented consistent, predictable navigation patterns. We also integrated Clerk for frontend user authentication to ensure secure login and a smooth user flow. Components were designed to clearly guide users through viewing, filtering, and interacting with listings.

demo video



takeaways

In the end, working across the full stack pushed me to think more critically about how each layer of an application communicates and depends on the others. Going forward, I'm excited to continue building systems that balance technical depth with great user experience.

Visit the Github Repository
