CONSUMER AND MOBILE

FrugaList

A consumer shopping app built on a game engine: a Unity mobile application with a Django web service behind it, finished and working.

2020	Unity	Django
Build began	Mobile front end	Web service behind it

WHY UNITY FOR A SHOPPING APP

A game engine will carry a consumer application when the interface wants motion and depth that a standard mobile toolkit makes expensive. It will also fight you, unless somebody on the team knows where the edges are.

The edges are specific: store review, application size, battery, and the fact that a game engine has opinions about the keyboard, text input and accessibility that a shopping flow cannot ignore.

WHAT WE BUILT

- **A Unity mobile client** for browsing and shopping, with the motion the founder wanted. FrugaList was a friend's shopping product, and we built all of it.
- **A Django web service** behind it for the catalog, the users and the transactions.
- **A finished, working build.**

The founder later shut the backend servers down over hosting costs, which is a product outcome and not an engineering one. We report it that way because a portfolio where nothing ever ends is not believable.

WHY IT MATTERS TO A BUYER

It is the reason Unity sits in the mobile column of our stack and not only in the training column. Unity is also the engine behind SPARC, our immersive training platform.

Unity · C# · Django · Python · iOS · Android

WHAT WE CAN START ON

A Unity build that has to behave like an application rather than a game. Or a consumer front end with real motion, backed by a service that holds up.

512-900-4871 · div42.com · roblesindustries.com