OpenUp Offers

Intro & Goal (Why build it?)

OpenUp Offers is an initiative to acquire users faster, cheaper, and for longer periods.

Users

Our mission is to return control of the online shopping experience to the consumer. People today are bombarded with irrelevant information about what they should buy, and they are tired of it. We know this because millions of people use products that help them choose what brands they are exposed to, as well as products that help them find the best deals on what they *actually* want to buy.

As users ourselves, we want to make better decisions about where and when to shop. We also want to control when and how we are marketed to. Thus, we are building a platform that combines these features, allowing us to track the prices of products *we* are interested in, while receiving opt-in recommendations of relevant products.

Product Strategy (How do we do it?)

Our success will be measured by our ability to **grow** and **retain our** user base.

To achieve this, we will focus on the user experience in three ways:

1. The value of the product must be clear.
2. The product must be easy to install and use.
3. The product must have multiple touch points.

What is it?

Landing page

1. To pique users' interest, a quick summary of what the value of the product is.
2. To establish users' trust, a section with bios of the team.
3. To sate users' curiosity, a more detailed outline of our business model, and a "contact us" section.
4. To satisfy legal requirements, a link to the privacy policy etc. at the bottom.

Frontend (browser extension)

Core functionality

1. To allow users to ==add items== on the page, the extension detects compatible URLs.
2. To allow users to ==specify the price== they would like to pay, there is a slider that they can use
3. To allow users to opt-in to ==receive recommendations==, there is a "yes", "no" option (optional)

Additional necessary functionality

4. To prevent user frustration and allow them to remove items, they can ==view existing items==, and remove them (pulled from price-check DB)
   a. To provide us with purchasing data, *and* provide users with a more satisfying experience, they are able to ==click "purchased"== to let them ==track how much they have saved==.
5. To engage the user and provide an additional touch point, users are able to ==view/receive push notifications==.

Backend

1. To pass data to the frontend/database we parse the current URL (price, product, etc.)
2. For future analysis, we record the current URL, tab, and session ID to the ==user-browsing table== [completed]
3. To let the user see his/her saved items, and to decide when to send users emails, if the user adds an item, we store the item in ==price-check table==
   a. This database has product_id, item, and current_price as columns
      i. This is a common database with product_id as the primary key, so that multiple users tracking the same product does not generate multiple identical items to be tracked
   b. In addition, there should be a ==user-product table==, to store which users are tracking which products
4. To tie all these data together, each user has an entry in a ==user-summary table==
   a. This has the following tables:
      i. user_id, age, ethnicity, latitude, longitude, total_amount_saved, number_of_items_saved, total_urls, date_installed, days_installed, active_days

Internal Dashboard

1. To monitor status of our systems, we have a report of current usage [completed]
2. For ad-hoc analysis, we can:

       a. Generate a raw .csv of current users [in progress]
       b. Download raw browsing data by specifying user IDs/emails
           i. We can download collapsed browsing summary data of all users
         ii. We can queue keywords to scan for
3. Candidate features for removal
    a. "Brands"
    b. "Offers"
    c. "Regional reports"

Planned Features

1. Generate a list of products to recommend to the user
2. Users can access the same functionality on mobile

Mockups/Wireframes

The latest set of wireframes can be found here: https://marvelapp.com/15i524c

How are we measuring success?

We aim to have 100,000 users by June 2017.

Who are our competitors?

Comparison Shopping (no alerts)

- Avast SafePrice (10,000,000+ users)
- Feng Money Saving Master (600+ users)
- ValueTag (1,800+ users)

Coupon Finder

- Honey (2,500,000+ users)

Price Alerts

- Personal Shopping Assistant (Beta; 2500+ users) – Microsoft
  - https://www.reddit.com/r/Windows10/comments/57tcpa/microsoft_personal_shopping_assistant_extension/
- Covvet
- Aliexpress Shopping Assistant (16,500+ users): only for Aliexpress
- Zen Shopping (2,500+ users)
- Explorate Shopping Bookmark (250+ users)