Home Search

GROCERIES ON A BUDGET

A website that provides a comparison of nearby grocery store item prices and recommends the optimal option based on a combination of price and accessibility.

Home Search

RIVER RATS

Rachel Li, Jihyun son, Ilana Strauss, Bella Davies, Tia Hannah

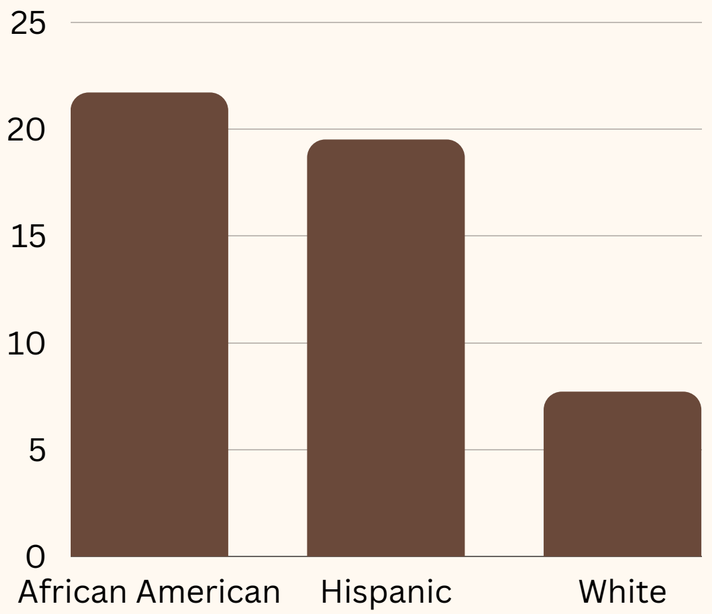
Home Search

PROBLEM STATEMENT

THE ISSUE BEING EXPLORED

- Inflation has broadened the gap in **economic disparity** among different ethnic groups.
- African American and Hispanic households were more likely to experience food insecurity than White households, with **21.7%** of Black households and **19.5%** of Hispanic households experience food insecurity, compared to **7.7%** of White households.

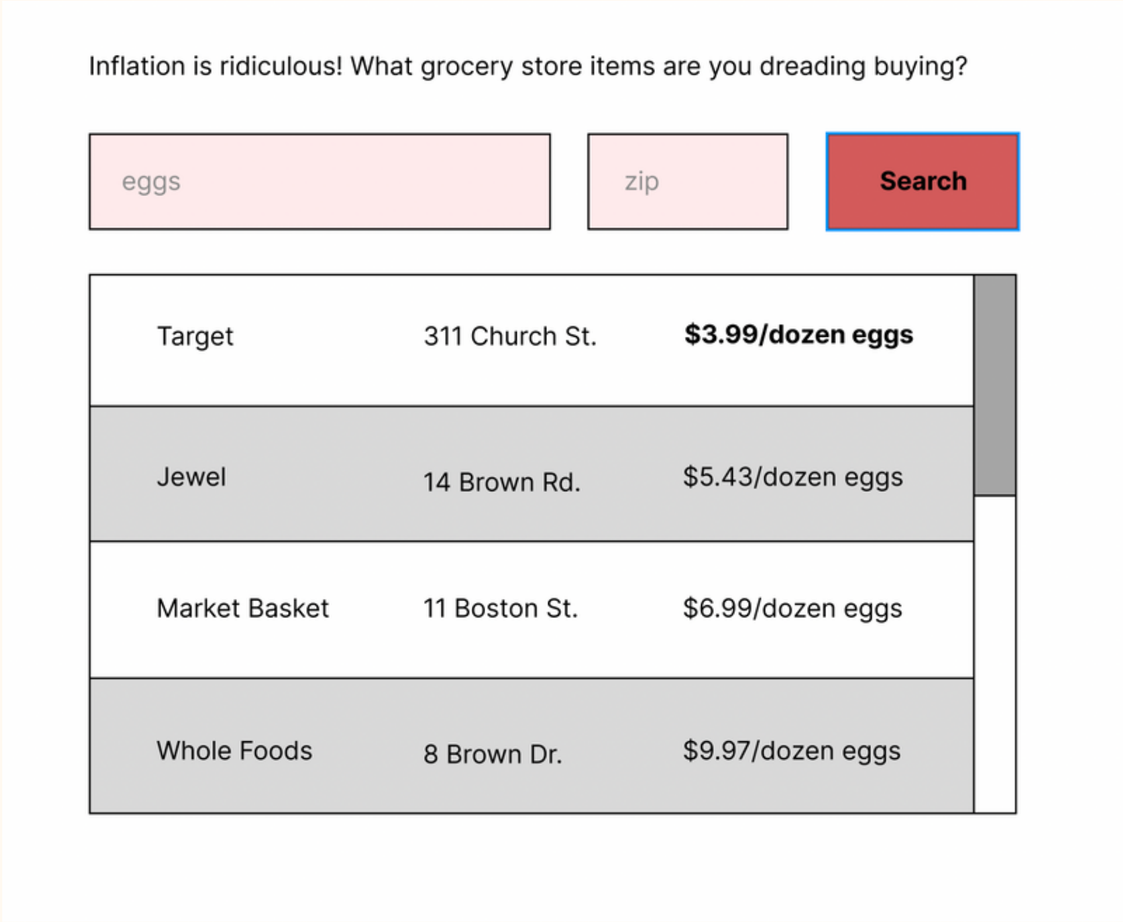
[1] USDA



PROJECT OVERVIEW

OUR SOLUTION

We created a framework for a grocery shopping website that generates the cheapest grocery item within 5 miles radius



HOW IT WORKS

TECHNOLOGY USED

We used Python, HTML, and CSS

```
Enter your address or ZIP code: 02215
Target Grocery (0.27 miles away): https://www.target.com/c/order-pickup/-/N-ng0a0?aisles
Target Grocery (0.63 miles away): https://www.target.com/c/order-pickup/-/N-ng0a0?aisles
Amazon Hub Locker+ (0.64 miles away): http://amazon.com/ulp
Wollaston's Market (0.80 miles away): http://wollastonsmarket.com/?utm_source=gmb&utm_me
Whole Foods Market (0.82 miles away): https://www.wholefoodsmarket.com/stores/symphony
Symphony Market / Halal / postmate (0.91 miles away): http://www.symphonymarket.net/
College Convenience (0.91 miles away): http://www.foodler.com/MA/Boston/Convenience/Conv
Trader Joe's (0.93 miles away): https://locations.traderjoes.com/ma/boston/510/?utm_sour
Trader Joe's (0.93 miles away): https://locations.traderjoes.com/ma/cambridge/502/?utm_s
Trader Joe's (1.03 miles away): https://locations.traderjoes.com/ma/brookline/501/?utm_s
Whole Foods Market (1.08 miles away): https://www.wholefoodsmarket.com/stores/riverstree
Target Grocery (1.14 miles away): https://www.target.com/c/order-pickup/-/N-ng0a0?aisles
International Halal Market (1.15 miles away): https://internationalhalalmarket.business.
H Mart Cambridge (1.19 miles away): https://www.hmart.com/
South End Food Emporium (1.26 miles away): https://thesouthendfoodemporium.com/
The Butcherie (1.31 miles away): http://www.butcherie.com/
Bazaar On Beacon Street (1.34 miles away): http://www.bazaarsupermarkets.com/
Whole Foods Market (1.38 miles away): https://www.wholefoodsmarket.com/stores/cambridge
```

closest_stores.ipynb

File Edit View Insert Runtime Tools Help Last saved at 8:43 AM

+ Code + Text

```python
import requests
from bs4 import BeautifulSoup
from geopy.geocoders import Nominatim
from geopy.distance import geodesic

# Set up the Google Maps API request
api_key = 'AIzaSyDYFAVWtbx70nJ2UFsqUFuQE_RA7S-uQ1A'
url = 'https://maps.googleapis.com/maps/api/place/nearbysearch/json'

# Get the user's location
geolocator = Nominatim(user_agent='my_application')
address = input('Enter your address or ZIP code: ')
location = geolocator.geocode(address)
user_lat, user_lon = location.latitude, location.longitude

# Make the Places API request
stores = []
params = {
    'key': api_key,
    'location': f'{user_lat},{user_lon}',
    'radius': 8046.72,  # 5 miles in meters
    'type': 'grocery_or_supermarket'
}
response = requests.get(url, params=params)
results = response.json()['results']
for result in results:
    lat, lon = result['geometry']['location']['lat'], result['geometry']['location']['lng']
    store_location = (lat, lon)
    distance = geodesic((user_lat, user_lon), store_location).miles
    if distance <= 5:
        place_id = result['place_id']
        details_url = 'https://maps.googleapis.com/maps/api/place/details/json'
        details_params = {
            'key': api_key,
            'place_id': place_id,
            'fields': 'name,website'
        }
        details_response = requests.get(details_url, params=details_params)
        details_result = details_response.json()['result']
        if 'website' in details_result:
            store_name = details_result['name']
            store_url = details_result['website']
            stores.append((store_name, distance, store_url))

# Sort the list of stores by distance
sorted_stores = sorted(stores, key=lambda x: x[1])

# Print the name, distance, and URL for each store within 5 miles, sorted by distance
for store in sorted_stores:
    print(f'{store[0]} ({store[1]:.2f} miles away): {store[2]}')
```

Home Search

DRAWBACKS TO HACK

THINGS TO CONSIDER

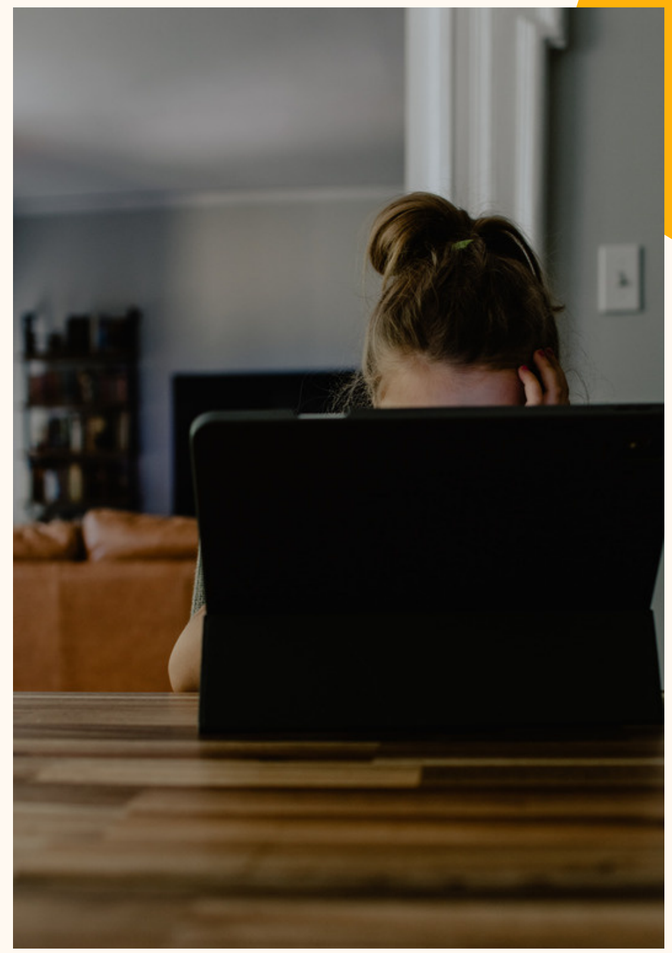
More comprehensive algorithm:

- Those who have been impacted the most by the rise in price heavily rely on public transportation. We could connect to Google Maps routes to provide in-app commuting times
- Factor in item quantities to show price per unit
- Some farmer markets don't have a website so they won't be able to participate

Home Search

CHALLENGES FACED

- Pull the information from the website and find the prices on the website.
- Websites have different URLs with different HTML code for each. The code we've developed is not universal
- Find the specific item on the website because the URL is different from page to page
- The page the specific item is on has to be on the URL that's in the code to print the item



GLOBAL IMPACT

OUR VISION

- The site can be used by everyone and the general public for finding more affordable groceries
- Would work for anyone, anywhere in the world since it uses a Google Maps API. Can search for any address or zip code.
- There are deeper rooted issues around the causes of food insecurity, but our project can help a lot of low-income families and individuals
- In the future, we could like to incorporate different dietary options to develop a more inclusive product
- We want to partner up with local farmers to help small businesses

Home Search

LEARNING LESSONS

- We learned to scrape data from websites and extract the information from the HTML to the code
- API key for Google Maps
- Beautiful Soup Python library for web scraping and extracting information from HTML files
- HTML & CSS for displaying output to the user
- Build a project that's feasible to execute within the given time constrain

FAVORITE MEMORIES