Arbitrachu

Pokemon Cards Arbitrage Identifier

601.466

Team members:
Ian Zheng (yzheng67@jhu.edu)
Shaopeng Zeng (szeng10@jhu.edu)
Ken He (yhe99@jhu.edu)

Arbitrachu extracts pricing data from online Japanese and US Pokemon card platforms as well as shipping costs from leading shipping agencies to help users to identify potential arbitrage opportunities.

Project Github

About The Project

We developed a Pokemon application designed to track and compare the price differences between the Japanese and US markets for Pokemon cards. This tool is especially useful for collectors and resellers looking to take advantage of price discrepancies across markets.

The major functionalities are:

- Track and compare Pokemon card prices between the Japanese and US markets.
- Identify profitable opportunities for buying and selling cards across markets.
- Fetch real-time card prices from trusted sources.
- Keep track of price trends and fluctuations over time.

Web Robot Ethics Guideline

For our semester project in computer science, we developed web crawlers to extract information on Pokemon card prices, currency exchange rates, and shipping information from sources outside of the JHU websites. To prioritize ethical use and ensure the safety of our code, we made the decision to refrain from unleashing our robot and instead only run the backend locally. This ensures that we do not constantly scrape the websites. Additionally, we implemented a policy to visit only one webpage for all scraping activities, and we do not navigate to other pages during runtime to ensure the safe usage of web agent development

Run Our Code

Run backend and frontend locally

Our code has a separate back-end and front-end. To run our code, navigate to each folder, initiate a virtual environment for Python, install all dependencies, and run the front-end and back-end separately.

1. Navigate to the backend folder and run pip install -r requirements.txt to install all dependencies.
2. Next, run the following code to set up the virtual environment for the backend
    - for window:
      python -m venv venv

    - for mac
      python3 -m venv venv
    - Then run
      python app.py
3. Navigate to the frontend folder and run npm i --force to install all dependencies.
4. Run the following code to start the frontend

   npm run start

5. You should be directed to http://localhost:3000/

Update Pokemon cards

Since we did not deploy our code to a third-party server, you will have to update Pokemon cards manually

- Update Japanese prices and info
    - Navigate to the backend folder and run python JP_pokemon_search.py
- Update US prices and info
    - Navigate to the backend folder and run python US_pokemon.py
- In the previous two features, we run the script and update to the mongodb database.
- Shipping info
    - Shipping info is scraped in runtime in our front end so no manual update is needed

Performance Benchmark

The project we built is an application so it is difficult to find a benchmark for performance evaluation. Here is some statistical information that may be helpful:

- The application we developed has been tested with a large dataset of 5,000 cards containing Japanese names and prices. Among these, we were able to identify and fetch valid US prices for 1,000 cards. This gave us a substantial dataset to work with in comparing price differentials between the two markets.

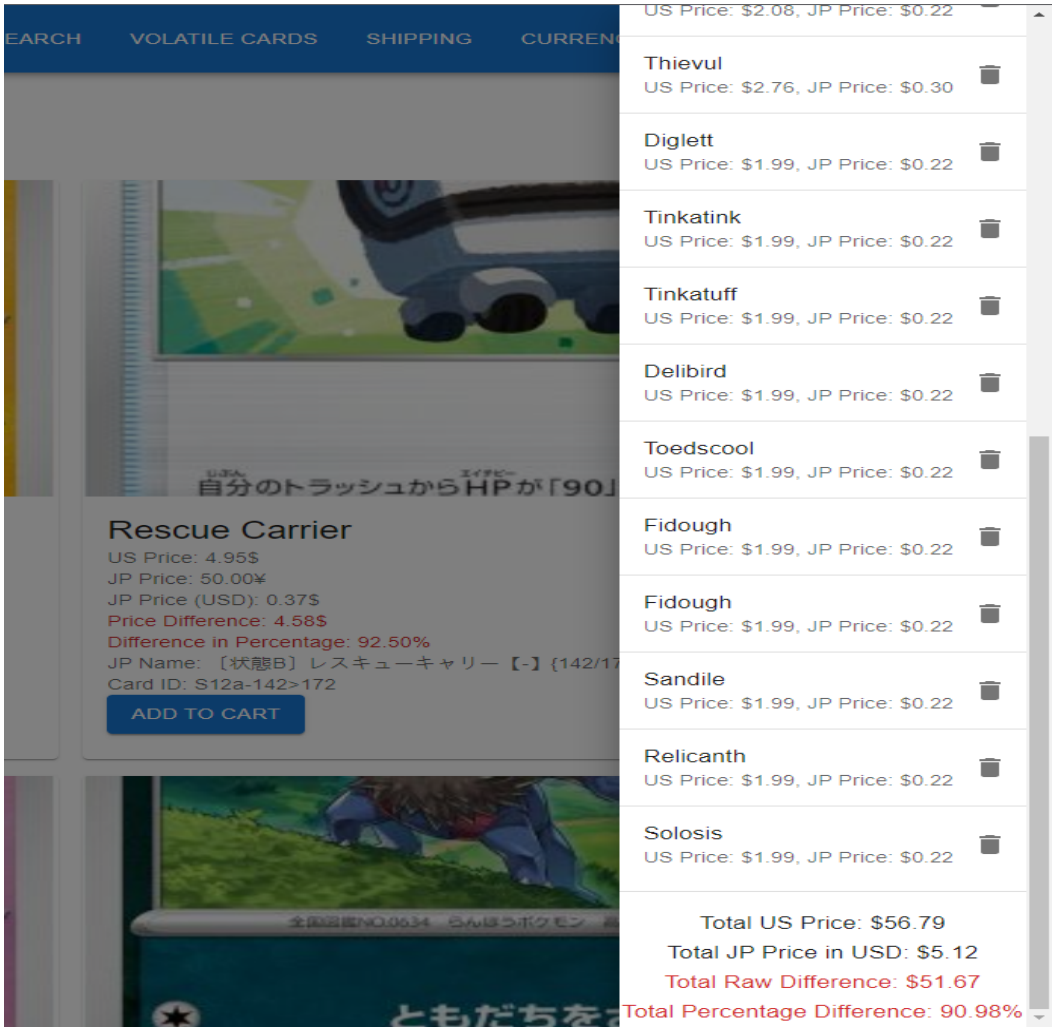
- The system was able to perform price matching on more than 1,500 cards, giving us detailed insights into the price variations between Japanese and US markets.

Through this process, we discovered a potential profit exceeding $5,000, averaging at over $3 per card.

- We then implemented a sorting algorithm that sorted the cards based on the price difference percentage. The system was able to quickly and efficiently sort this data, enabling users to view the cards with the highest price difference at the top.

- To benchmark this feature, we selected the top 20 cards based on the price difference percentage. The total percentage difference for these top 20 cards was an impressive 90.98% using only $5.12. This shows that the system is effective in identifying cards with the greatest potential for profit.

- This performance evaluation highlights the efficiency and effectiveness of our application in handling large datasets and providing valuable insights into price differences between markets.



Achievements

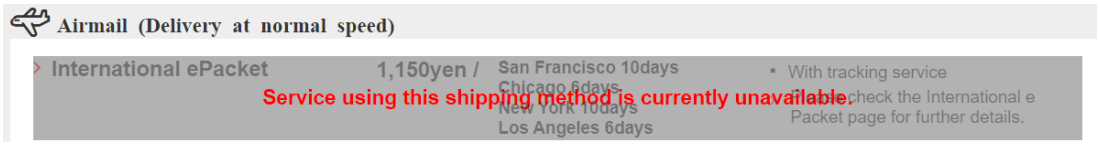
1. One of our key strengths is our ability to **consolidate information** from disparate sources and **derive meaningful insights**. In our code, we integrate information from multiple websites, across two different languages, and across various types of data including card pricing, exchange rates, and shipping information to identify arbitrage opportunities. Each section of the code has a moderate scope and complexity but combined together provides value for our users.
2. Our code is a **scalable** and **efficient** solution for collecting and monitoring pricing and card information for Pokemon cards from multiple pages of Japanese and US trading websites. We utilize Beautiful Soup and regular expressions to accurately extract relevant information, ensuring consistent and reliable data. The structure of our code is clear and we can easily scale it to more regions, languages, and currencies.
3. Our code allows users to compare shipping prices from three different shipping companies: **Japan Post, DHL, and FedEx.** By entering the required information, such as the number of cards, US zip code, and Japan zip code, users can easily view shipping prices and options from each company. Please note that due to the use of Selenium scripts, fetching the results may take approximately 15-30 seconds per shipping company. Link
4. Our code allows users to easily convert currencies between USD and JPY. You can always **click on the money icon** to access the currency conversion tool via a pop-up window. Link
5. Our code effectively extracts US card names and prices for Pokémon cards **using BeautifulSoup and Selenium**. This is a significant achievement, considering the complexities involved in dealing with different languages, text formats, and website structures. By using both Selenium and Beautiful Soup, we have ensured a robust and flexible approach to data extraction that can handle various challenges and adapt to future changes in website structures. Link Link
6. We successfully identified **1,000 cards** with valid US prices out of **5,000 cards** with JP names and JP prices. This high extraction rate speaks to the quality and effectiveness of our code, especially given the challenges of dealing with differences in text, languages, and cultures between the two countries. Link Link
7. Despite time constraints, we have **developed a fully functional frontend interface** that is both user-friendly and visually reasonable. By leveraging multiple libraries and technologies, we have created a decent experience for users to

navigate our websites and learn the price differences between JP and US cards. Link

8. We have made another significant leap in our project by introducing the ability to **sort Pokémon cards based on the price difference percentage**. This feature provides substantial value to the users by identifying the most profitable opportunities for buying and selling cards. By leveraging the power of our back-end data processing, we've turned raw data into actionable insights that can guide decision-making in the frontend.
9. We have incorporated a cart feature that allows users to compare the total price of their selected cards in both US dollars and Japanese yen. This feature allows users to make informed decisions about their purchases by offering a clear and straightforward way to assess potential cost savings or earnings.This includes not only the sorted list of cards but also visual indicators such as color-coded price differences and arrows indicating price trends. Link

Limitations and Possible Improvements

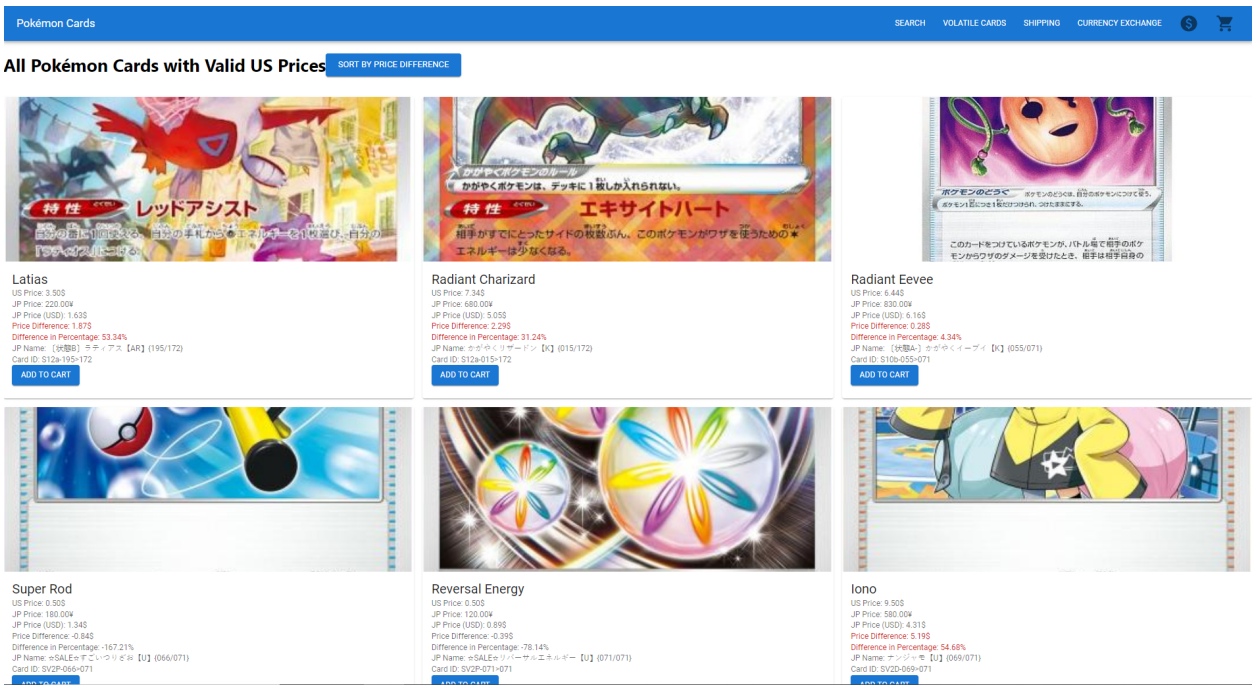
1. In the Pokémon card market, the condition of a card can greatly affect its price. In Japan, sellers often indicate the condition of the cards, and the prices we have extracted reflect cards in higher-end conditions. However, in the US market, buyers often do not specify the card conditions, and the US prices we have extracted are more reflective of used cards, which are usually **not in perfect** condition. Consequently, some Japanese prices **may appear higher than** US prices, not necessarily due to market differences, but due to differences in card conditions. Despite this limitation, it is important to note that many cards still show **significant profit potential** even when factoring in these condition discrepancies.
2. When scraping Japanese and US card prices, the solution is customized to the particular card trading websites. To extract data from more websites, we will need to build customizable scarping solutions for those websites as well and the marginal return is low. However, since most of the trading websites follow a similar structure in presenting information (name, id, and price), a possible improvement is to use machine learning techniques and automatically find the prices instead of customizing our code for each website. The complexity of the solution might be high since we need to make a universal solution for different websites but we could greatly increase the reliability of the prices since it comes from more sources.
3. When attempting to obtain shipping prices from FedEx, USPS, UPS, and DHL, we encountered difficulties using their APIs directly. To access the APIs for FedEx, USPS, and DHL, a business API key is required, which necessitates having an actual company and applying for it. As for UPS, their API only permits quoting shipping prices from the US to Japan, rather than from Japan to the US. Consequently, we decided to utilize Selenium scripts to perform GUI interactions directly on the respective websites. Link
    a. This approach results in longer delays, taking approximately 15-30 seconds per shipping company.
    b. The Selenium script may occasionally time out, requiring users to rerun the function on the webpage.
    c. When running Selenium, a pop-up window appears. Although I attempted to use "headless" mode to mitigate this issue, it increased the likelihood of encountering bugs.
    d. Some service may not be available at certain time and fail to scrape

i. 

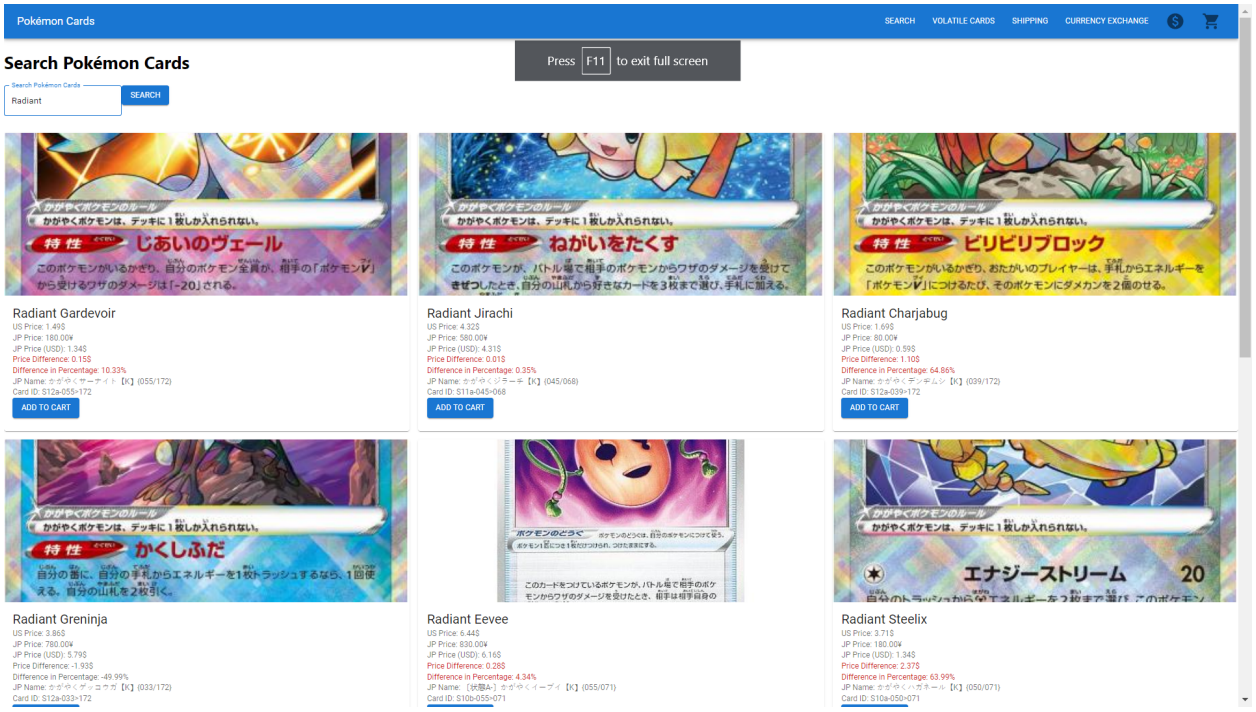
4. As our website is not yet deployed, we are unable to implement automated crawling functionality that would allow for regular updates, such as hourly or daily refreshes of card prices and information. This limitation impacts the real-time accuracy of our data. After the finals, we plan to deploy the website and enhance the crawling functionality for more up-to-date information.
5. Currently, our project focuses on a select number of websites that sell Pokémon cards. However, not all cards are available on these sites, which may limit the comprehensiveness of our platform. In the future, we plan to incorporate more websites into our data collection process to improve our market coverage and offer a more comprehensive service to our users.
6. Our current implementation relies on Beautiful Soup to scrape card information, which requires making an HTTP request for each card individually. This approach can be slow and resource-intensive, especially when dealing with a large number of cards. We are exploring alternative methods for data collection that can improve the efficiency of our scraping process without compromising the quality of our data.
7. Our project relies on the accuracy and availability of data from third-party websites, which may change their structure, policies, or availability at any time. This dependency presents a risk to the stability of our platform, and we will need to continually monitor and adapt to changes in these sources to maintain the accuracy and reliability of our data.

Sample and Screen Shots

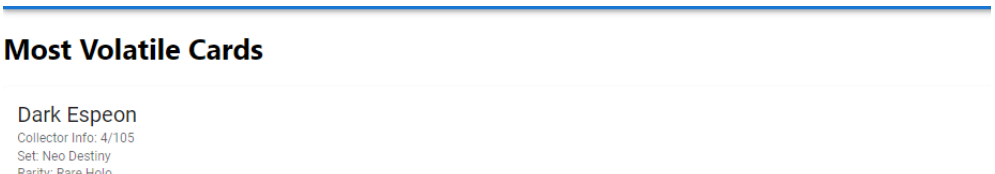
- Home Page:



- Card search:

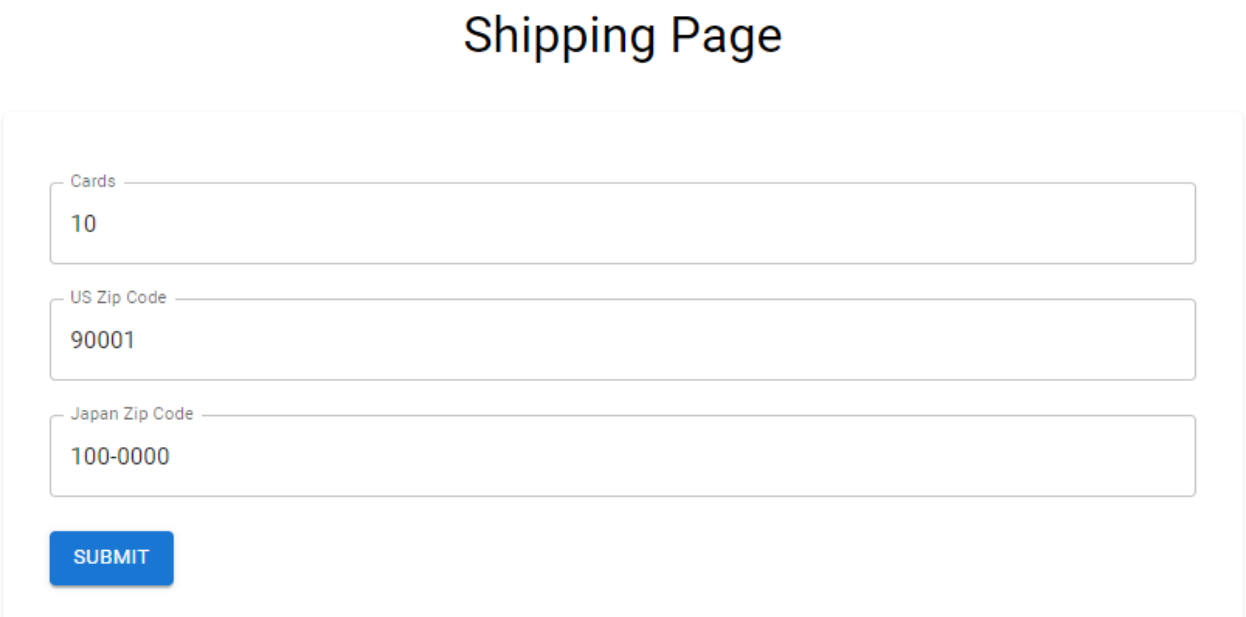


- Volatile cards tracker



- Shopping Cart:

- Shipping page:



Shipping Info

```
Region of origin : Tokyo → Destination :  (California) United States
Item to be sent : Package
Weight : 50grams
```

Fastest Way: EMS (Fastest delivery) 3,900yen ($28.95)/ 2 ~ 3days Click Here

Other Shipping Methods

- **Airmail (Delivery at normal speed):**
- **SAL (Delivery depending on the available transport):**
- **Surface mail (Delivery at the lowest price):**

Shipping Info

- **delivery_date:**

  Tue, May 16

- **price:**

  ¥15,984 *($118.64)

- **shipment_date:**

  13 May 2023

Shipping Info

- **Fri, May 19 Fri, May 19:**

  ¥10,091($74.9)

- **Tue, May 16 10:30 AM NEW:**

  ¥10,790($80.09)

- **Tue, May 16 4:30 PM:**

  ¥10,285($76.34)

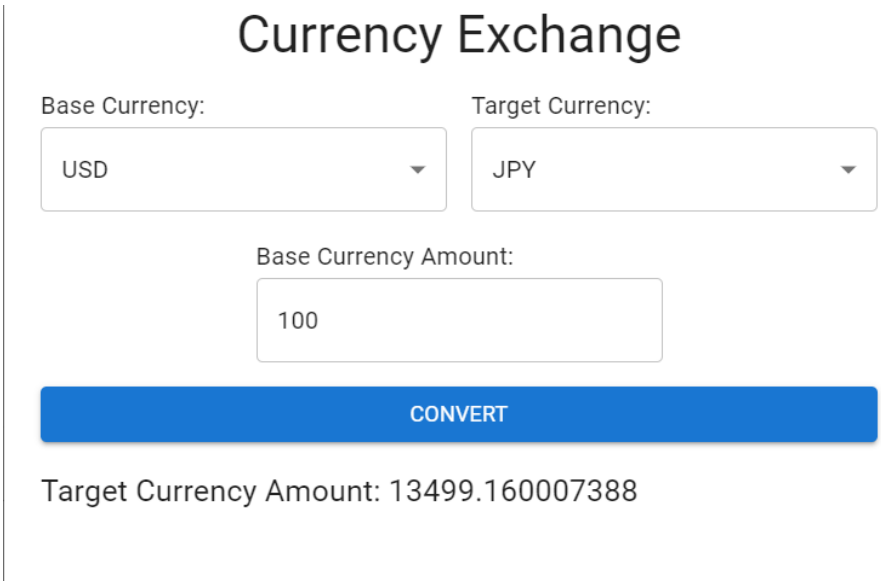
- **Tue, May 16 8:00 AM:**

  ¥21,278($157.94)

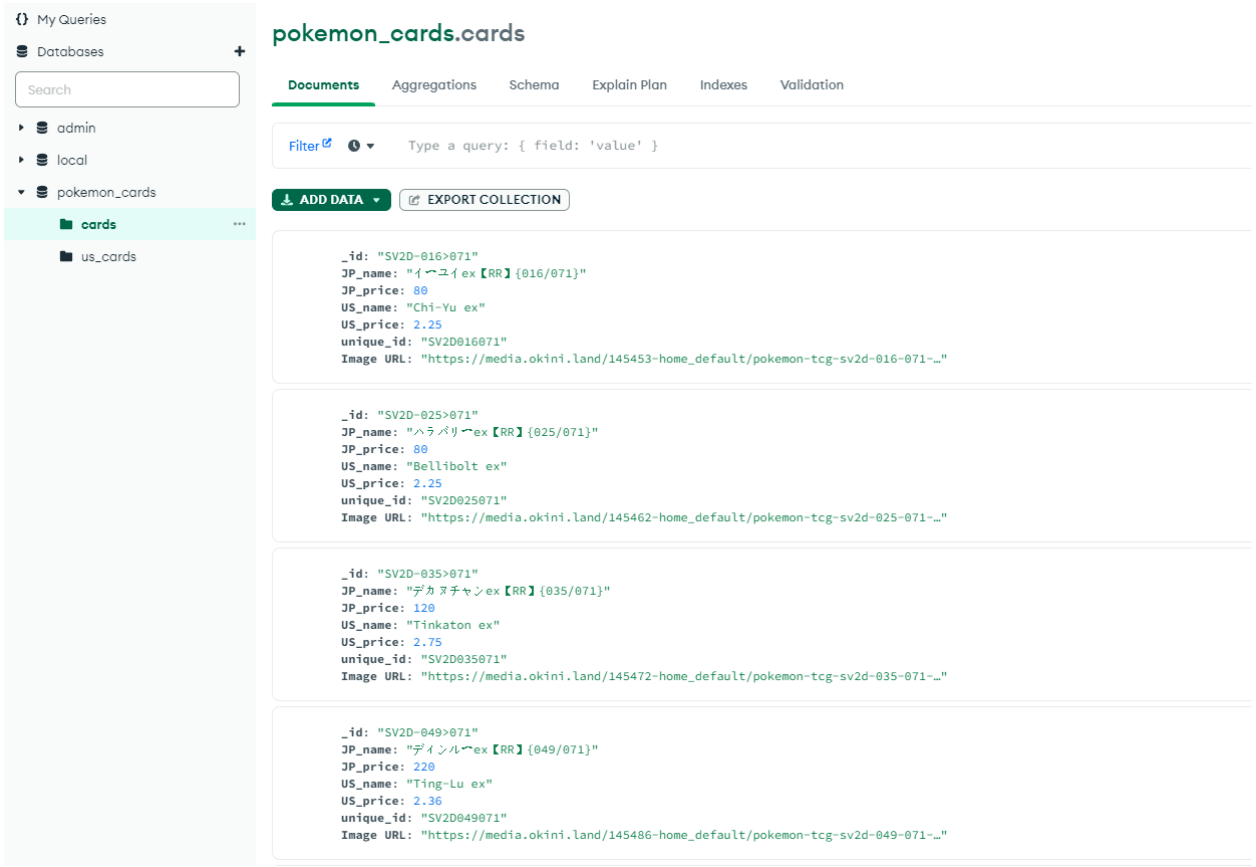
- **shipment_date:**

  Saturday, May 13, 2023

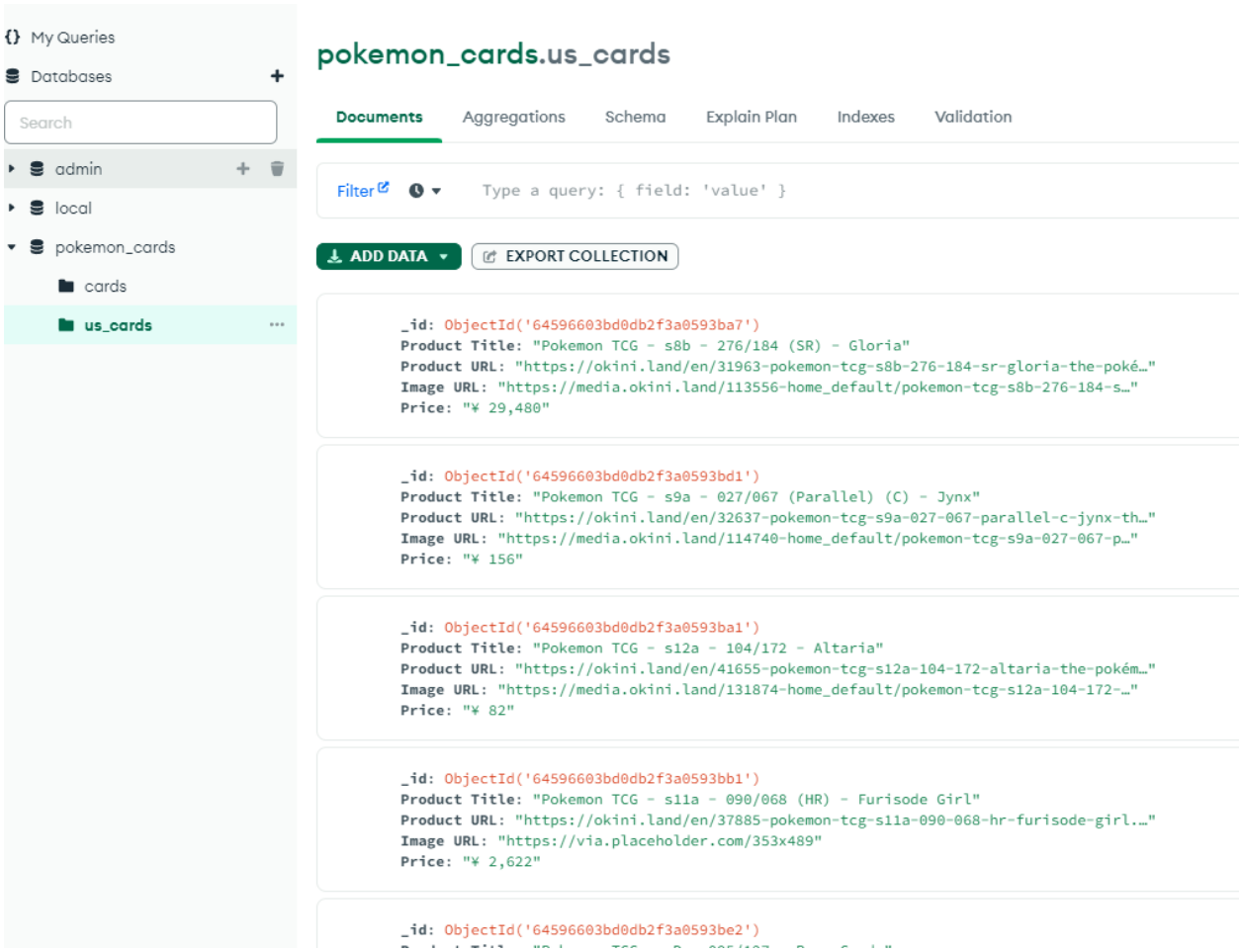
- Currency page



- Japanese Pokemon Card Database



- US Pokemon Card Database



Contributions

- Yujian (Ken) He
    - Shipping quote using selenium Link
    - Shipping quote using company API Link
    - Currency Exchange using currency API Link
    - Website deployment and Github Page Link
- Shaopeng Zeng
    - US Pokemon Card Name/Info Crawling Link
    - US Pokemon Card Price Crawling Link
    - US Pokemon Volatile Cards Crawling Link
    - Website deployment and Github Page Link
- Ian Zheng
    - Japanese Pokemon Card Name/Info/Price Crawling Link
    - MongoDB backend and card information API Link Link Link
    - Tests for backend API Link
    - Writeup Consolidaiton Link

Team Size Validation:

- The scale of the project involves tasks such as web scraping, data extraction, data manipulation, currency conversion, shipping cost calculation, and front-end development. We believe that the scope is difficult for a team of 2 to handle and with a team of 3, each team member can focus on specific aspects of the project, ensuring that all tasks are completed with expertise and attention to detail.
- With three people on the team, we can divide the workload and develop different components of the project simultaneously. This allows for quicker development and deployment of the final product, ensuring that we meet deadlines and deliver a high-quality solution to our users. Having multiple perspectives on the team enables us to approach challenges from different angles, leading to more innovative and effective solutions. By pooling our knowledge and experience, we can overcome obstacles more efficiently.