Project: The Pokeymen TCG

Members:

- Logan Fu
- Jonathan Yap
- Marcus Negron
- Calvin Pun

Group Name: The Card Lovers

Overview:

The purpose of this project is to present data about Pokemon cards on a website. The primary user interaction will use search queries to retrieve Pokemon card information from an API, storing said information in a database, and having a detailed page generated for each Pokemon card.

The project starts with a homepage, which contains an overview of each application and contains directories to each one.

The primary application we have working is the Pokedex that displays, retrieves, and stores information about Pokemon cards. Within this application, we have the CardList, which lists out all the Pokemon cards. Within CardList, users have the option to click on the name of the pokemon card they want to learn more about. They can add Pokemon cards by searching for Pokemon and using “Add to Pokedex”. We also have CardDetail, which gives more in depth information about each specific pokemon card (price, types, evolution, art, etc). Within CardDetail, users have the ability to delete the card.

The other application is CardData. This application will be primarily concerned with using matplotlib/pandas. The user can display information (in graphs), about the amount of cards based on a group by (ex: displaying a bar chart of the amount of cards grouped by type, rarity, artist, etc). It can also display price data based on rarity, compare prices between different cards, etc.

API Used:

- https://docs.pokemontcg.io/

What is a supertype?

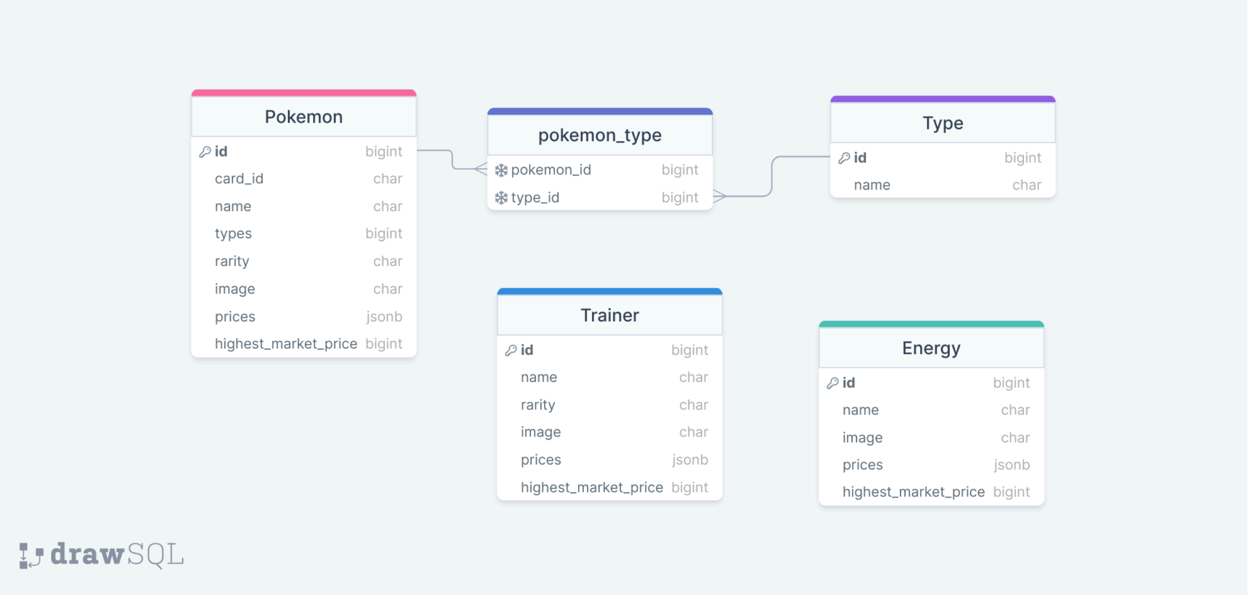
A supertype are the 3 types of Pokemon cards (Pokemon, Trainer, or Energy)

Applications:

- Home Page:
    - Basic homepage related to a single index view. Will describe each application and will link to each application.
    - The base template will also contain a basic navigation bar for each application.
- Pokedex:
    - Overview:
        - This is the primary application with the highest priority. It contains the model that will be used to interact with other applications, and the Pokedex application is the only application that will allow the user to interact with the database.
    - Views:
        - ListView: Display all Pokemon cards within the database. Also includes the search query from the API to get card data into the database, unless the search functionality needs its own view.
        - DetailView: Specifically show details for a Pokemon card in the ListView. Different for each supertype.
        - Delete views: Option for users to remove data in the database. An edit view is not planned, as the user should not be able to manually edit information from the database.
    - Models:
        - Pokemon: Model for containing pokemon card data, with fields such as name, attacks, types, etc
        - Trainer: Model for containing trainer card data, with fields such as name, image, etc
        - Energy: Model for containing energy card data
    - Actions:
        - Initial data button: Calls multiple API calls to retrieve data for all energy card types to fill the Energy model, and some to fill initial Pokemon/trainer data.

        - Detail: User clicks on the name of a specific card in the table to learn more about it in its detail view, or delete the card data from the database.
        - Add card: The user searches for a Pokemon card. The user then clicks on “Add To Pokedex” to add it.
- <u>CardData:</u>
    - Overview:
        - Display a lot of information about Pokemon cards based on the data in the database.
        - Amount of cards grouped by specific qualities
        - Price data based on rarity/price comparison between two cards
        - Generated graphs through pandas dataframe using data from database, and matplotlib
    - Views:
        - CardDataOverview: First page for the app. Generates aggregated graphs from all data within the database. Displays bar charts/pie charts for amount of cards grouped by specific qualities, price data based on rarity, etc. This view does not take user input, but rather displays data on the database at a glance.
        - ~~PriceComparison: User selects two cards from dropdowns and submits through a button. Page results in graphs comparing price data (high, low, market, etc) on bar charts.~~ (Incomplete)
    - Actions:
        - ~~Button to submit a form for PriceComparison~~

Database Design:



Pokemon	
id	bigint
card_id	char
name	char
types	bigint
rarity	char
image	char
prices	jsonb
highest_market_price	bigint

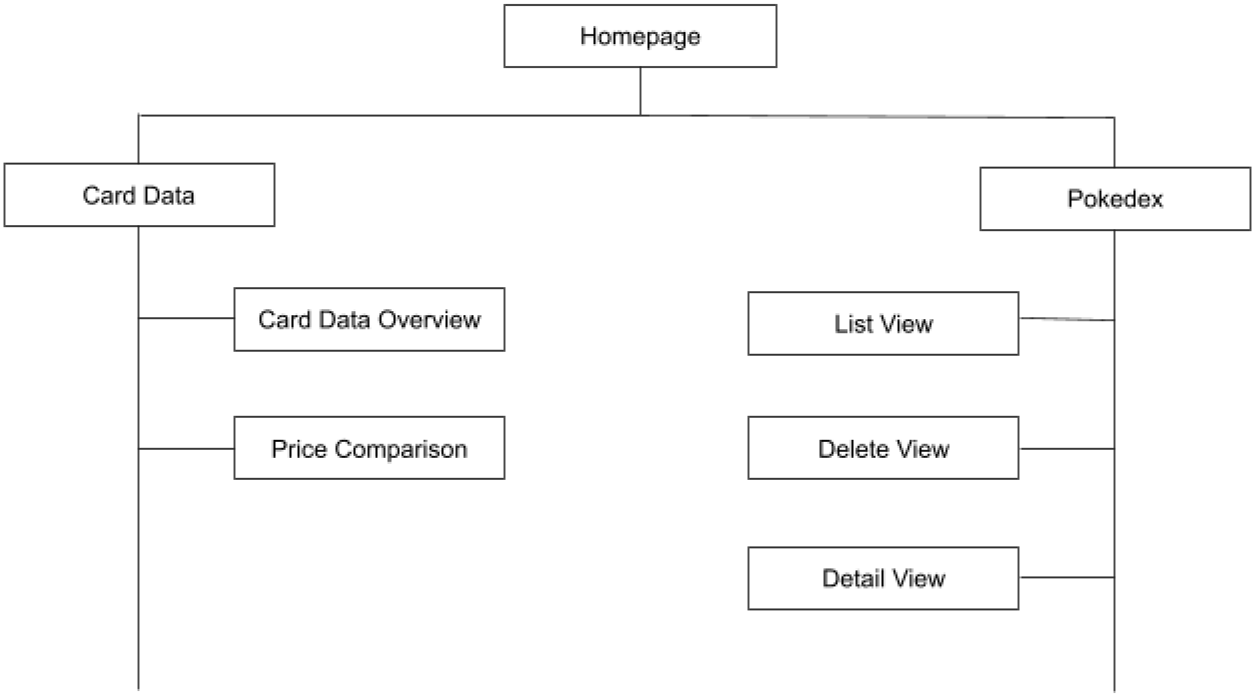
pokemon_type	
pokemon_id	bigint
type_id	bigint

Type	
id	bigint
name	char

Trainer	
id	bigint
name	char
rarity	char
image	char
prices	jsonb
highest_market_price	bigint

Energy	
id	bigint
name	char
image	char
prices	jsonb
highest_market_price	bigint

Site map:



Project task assignments:

Logan
Jonathan
Marcus
Calvin

- Home Page Application
    - User authentication
    - Navbar
    - Text/info about the site
- Pokedex Application
    - Add card functionality/view. Adds a card to the database based on supertype. Might need multiple views.
    - Search for card functionality (implement a system that returns results from a search query using a name keyword: https://docs.pokemontcg.io/api-reference/cards/search-cards). Might use a lot of VueJs. Results will have buttons that add the card to the database. Probably the most difficult/tricky task for Pokedex.
    - Display list of Pokemon cards based on supertype (table for Pokemon, Trainers, and Energy). Probably a generic list view.

  - Detail views for each supertype (detail for Pokemon, Trainers, and Energy)
  - Delete views
- CardData Application
  - CardDataOverview:
    - Pie and Bar graphs for amount of cards based on:
      - Supertype
      - Type
      - Rarity
    - Bar graphs comparing average price based on rarity
  - ~~Price Comparison function~~ (Not completed)