Tyson Bryant & Liam Armstrong

CS340 Introduction to Databases

12/10/2023

# Pokémon TCG Database

**Repository URL:**
https://github.com/tysonfromearth/Pokemon-Database/tree/liam-pokemon-database

**Executive Summary:**

Early feedback indicated that people might not understand details about the Pokemon Trading Card Game. We modified the “Project Outline” section of our report to include basic information about the Pokemon, the trading card game, decks, and cards. Here also, critiques of the relationship constraints convinced us to reverse the constraints on some relationships.

When we first started publishing work we did so via a public_html folder in our student accounts. Peers pointed out in review that while the permissions on the index.html file were correct, other files in the folder did not have the proper permissions, resulting in web pages which were not visible to other students. Adding the proper permissions remedied this problem. We also updated the schema so that sets do not require cards in reply to remarks that if sets did require cards then a set could be deleted if a card was deleted. Additionally, at this stage in project development we did not have an SQL insert in our DML file for creating an entry in the Decks_has_Cards table when a card is added to a deck so we wrote such a query.

When peers were able to get a better look at our html pages, it became clear from feedback that html pages for Pokemon and Sets would have to be added and that a home button would improve ease of navigation. The corresponding improvements were made.

During CRUD implementation, reviews noted that only read operations were working in terms of the CRUD paradigm, so CRUD operations were expanded for all entities except for Players, which was dropped for simplicity. However, as CRUD operations were integrated based off of the example starter code, peers observed that the operations were still showing the data from the Battlestar Galactica database. Accordingly, one of our final actions based on feedback has been to modify our application to display data specifically from our Pokemon Database. Furthermore, we yielded to suggestions to use one method to add data to the table. A navigation bar was added in this stage, as well.

**Project Outline**

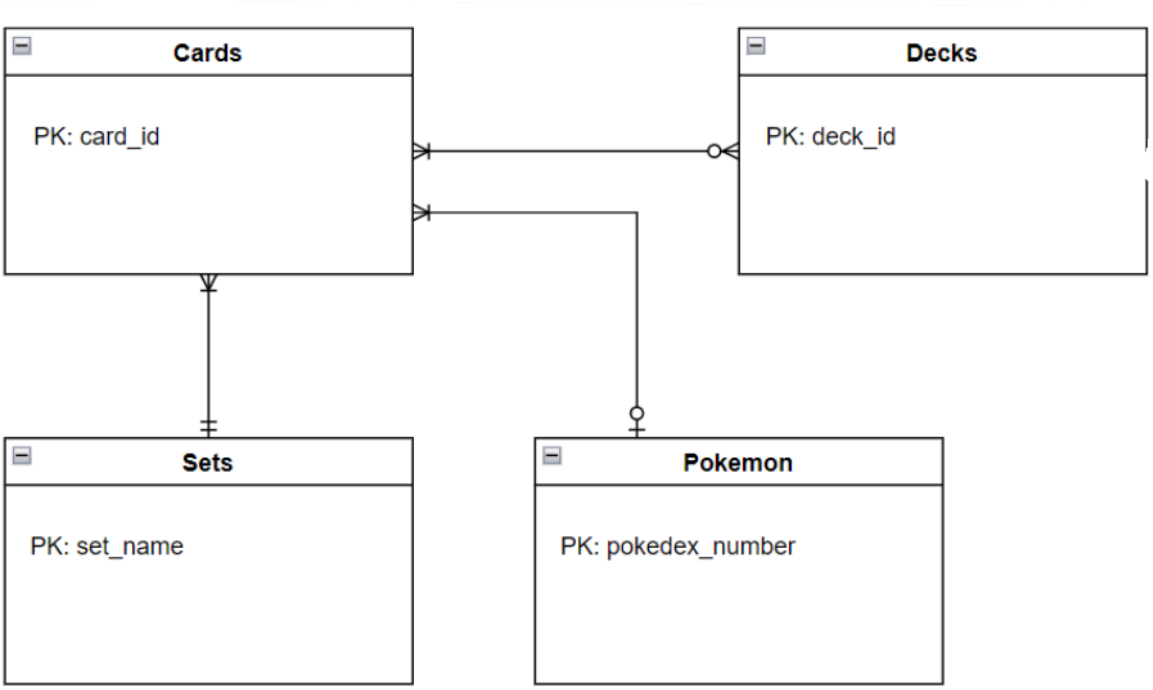
Collectors of the Pokémon Trading Card Game (TCG)  need an efficient system to manage their decklists, lists of cards contained in a deck used to play the game against opponents who have their own deck. Our database-driven website addresses this by cataloging a player’s decklists. The database also stores information on the cards that players can use in their decks which come from three legal sets. The legal sets are the Base Set and the Jungle and Fossil expansion sets, which have 102, 64, and 62 cards, respectively, for a total of 228 cards. Information about each of 150 unique Pokemon and which cards they appear on is also stored in the database. Users can design decks using Pokemon, Energy, and Trainer cards, ensuring they're prepared for battles.

**Database Outline**

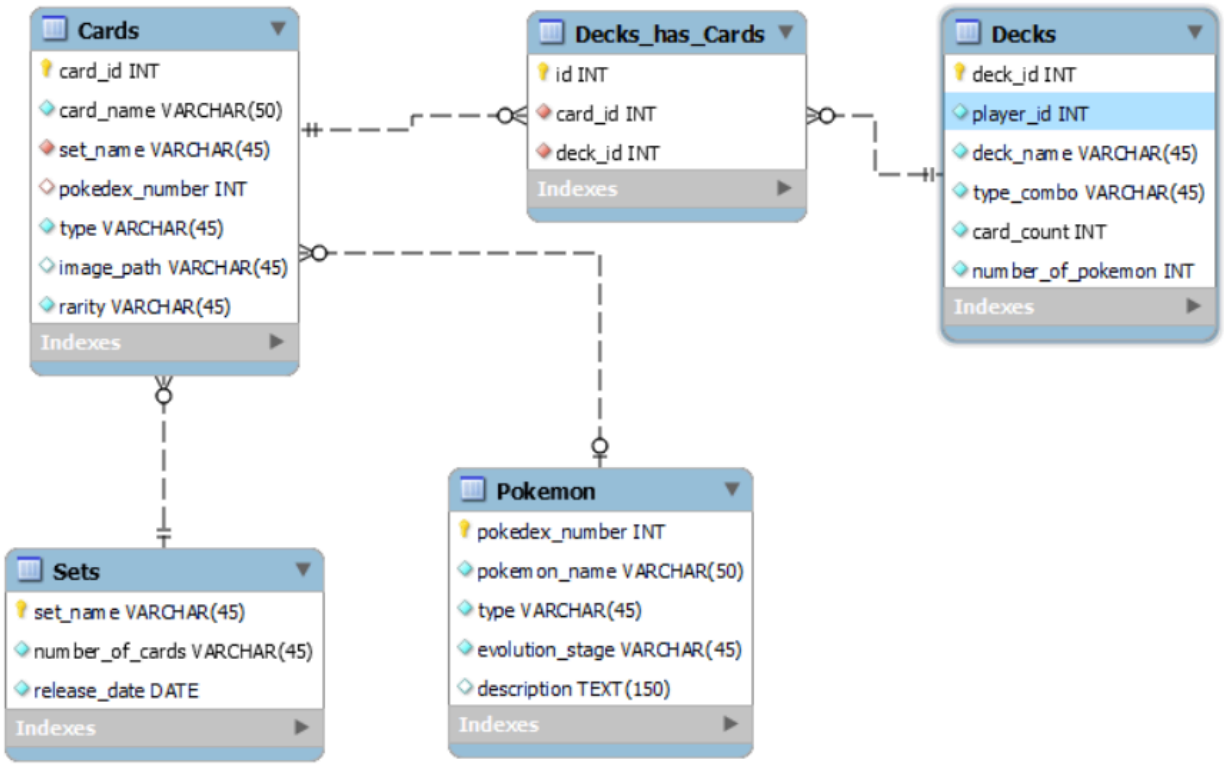
- **Cards**: Records the details of the cards found in Base Set and the Jungle and Fossil expansion sets.
    - **card_id**: INT (Primary Key, Auto-increment, Unique, Not Null)
    - **card_name**: VARCHAR (Not Null)
    - **set_name**: VARCHAR (Unique, Not Null, Foreign Key to Sets)
    - **pokedex_number**: INT (Nullable, Foreign Key to Pokémon) - A card may or may not be associated with a specific Pokémon.
    - **type**: VARCHAR (Not Null)
    - **image_path**: VARCHAR (Nullable)
    - **rarity**: VARCHAR (Not Null)
    - **relationship w/ Decks**: a M:N relationship between Cards and Decks is implemented with card_id and deck_id as FKs inside of the cards_decks intersection table. Each card can be part of multiple decks and each deck must have multiple cards.
    - **relationship w/ Sets**: a M:1 relationship between Cards and Sets is implemented with set_name as a FK inside of Cards. Each card belongs to one set, but each set has multiple cards.
    - **relationship w/ Pokemon**: a M:1 relationship between Cards and Pokemon is implemented with pokedex_id as a FK inside of Cards. Each card can represent one Pokémon, but each Pokémon must have at least one card.
- **Decks**: Records the lists of unique cards appearing in each of the decks the user has constructed. A standard Pokémon deck contains 60 cards. However, depending on the format or casual play, deck sizes can vary.
    - **deck_id:** INT (Auto-increment, Unique, Not Null, Primary Key) - The main identifier for a deck.
    - **deck_name:** VARCHAR(Not Null) - Represents the chosen name of the deck.
    - **type_combo**: VARCHAR(Not Null) - A list of the all the energy types of Pokemon contained in the deck.
    - **card_count**: INT (Not Null)
    - **number_of_pokemon**: INT, (Not Null)
    - **relationship w/ Cards**: a M:N relationship between Decks and Cards is implemented with card_name and deck_name as FKs inside of the Decks_has_Cards intersection table. Each card can be part of multiple decks and each deck must have multiple cards.
- **Sets**: Records details about each of the three relevant sets of cards.
    - **set_name**: VARCHAR(Unique, Not Null, Primary Key)
    - **number_of_cards**: INT (Not Null)
    - **release_date**: DATETIME, (Not Null)
    - **relationship w/ Cards**: a 1:M relationship between Sets and Cards is implemented with set_name as a FK inside of Cards. Each card belongs to one set, but each set has multiple cards.
- **Pokémon**: Records details about each of the 150 Pokémon; Catalogs the individual Pokémon characteristics and their evolutionary stages.
    - **pokedex_number**: INT (Primary Key, Unique, Not Null)
    - **pokemon_name**: VARCHAR (Unique, Not Null)
    - **type**: VARCHAR (Not Null)

    - **evolution_stage**: INT (Not Null)
    - **description**: TEXT
    - **relationship w/ Cards**: a 1:M relationship between Pokemon and Cards is implemented with pokedex_id as a FK inside of Cards. Each card can represent one Pokémon, but each Pokémon must have at least one card.
- **Decks_has_Cards**: Records the details of the cards found in Base Set and the Jungle and Fossil expansion sets.
    - **id**: INT (Primary Key, Auto-increment, Unique, Not Null)
    - **card_id**: INT (Foreign Key to Cards, Not Null)
    - **deck_id**: INT (Foreign Key to Decks, Not Null)
    - **relationship w/ Decks and Cards**: a M:N relationship between Cards and Decks is implemented with card_id and deck_id as FKs inside this entity

## ER Diagram




## Schema



**Sample Data**

**Cards Table:**

| card_name | card_id | set_name | pokedex_num | type | image_path | rarity |
|---|---|---|---|---|---|---|
| Pikachu | 1 | Base Set | 1 | Grass | /images/bulbasaur.jpg | Common |
| Charizard | 2 | Base Set | 2 | Fire | /images/squirtle.jpg | Common |
| Squirtle | 3 | Base Set | 3 | Water | /images/squirtle.jpg | Common |

**Pokemon Table:**

| pokedex_number | pokemon_name | type | evolution_stage | description |
|---|---|---|---|---|
| 1 | Bulbasaur | Grass/Poison | 1 | Bulbasaur is the… |
| 5 | Charmander | Fire | 1 | Charmander is the… |
| 7 | Squirtle | Water | 1 | Squirtle is the… |

**Decks Table:**

| deck_id | player_id | deck_name | type_combo | card_count | number_of_pokemon |
|---|---|---|---|---|---|
| 1 | 14567 | Pikachu Deck | Electric | 60 | 12 |
| 2 | 34566 | Water Deck | Water | 50 | 20 |
| 3 | 38506 | Fire Deck | Fire | 45 | 30 |

**Sets Table:**

| set_name | number_of_cards | release_date |
|---|---|---|
| Base Set | 102 | 1/9/91 |
| Jungle | 64 | 6/16/99 |
| Fossil | 62 | 10/10/99 |

**Players Table:**

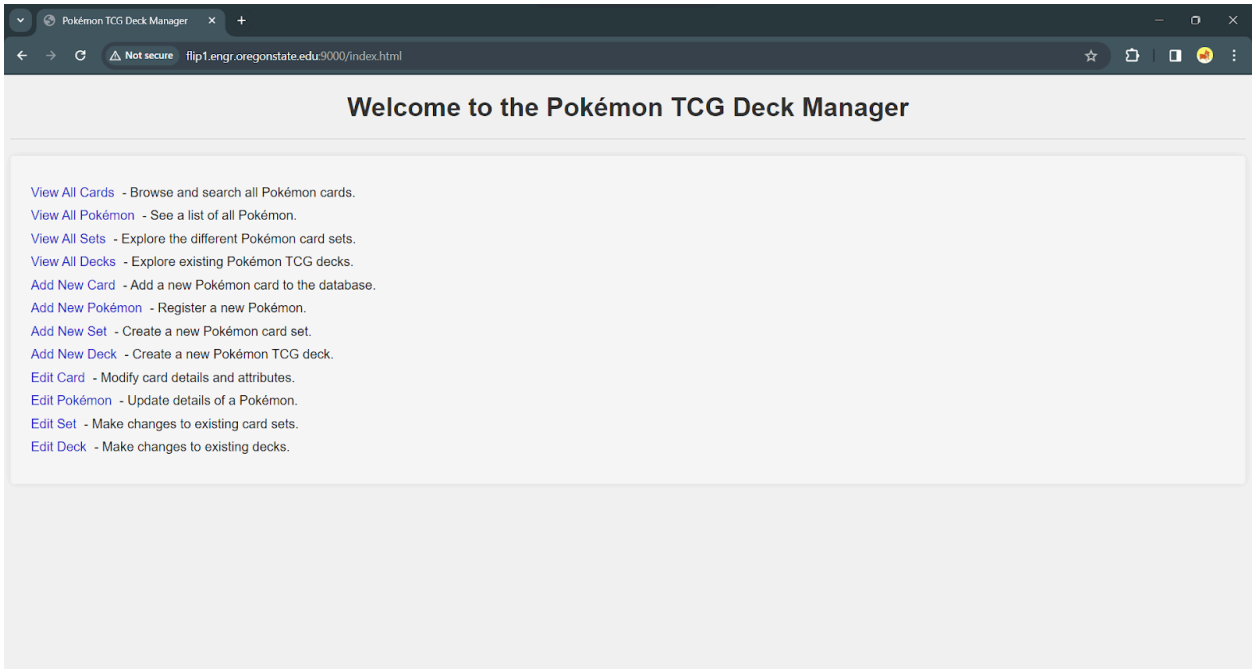
| player_id | name | country |
|---|---|---|
| 15670 | Ash | USA |
| 13450 | Misty | France |
| 14389 | Brooke | Belgium |

**Decks_has_Cards Table:**

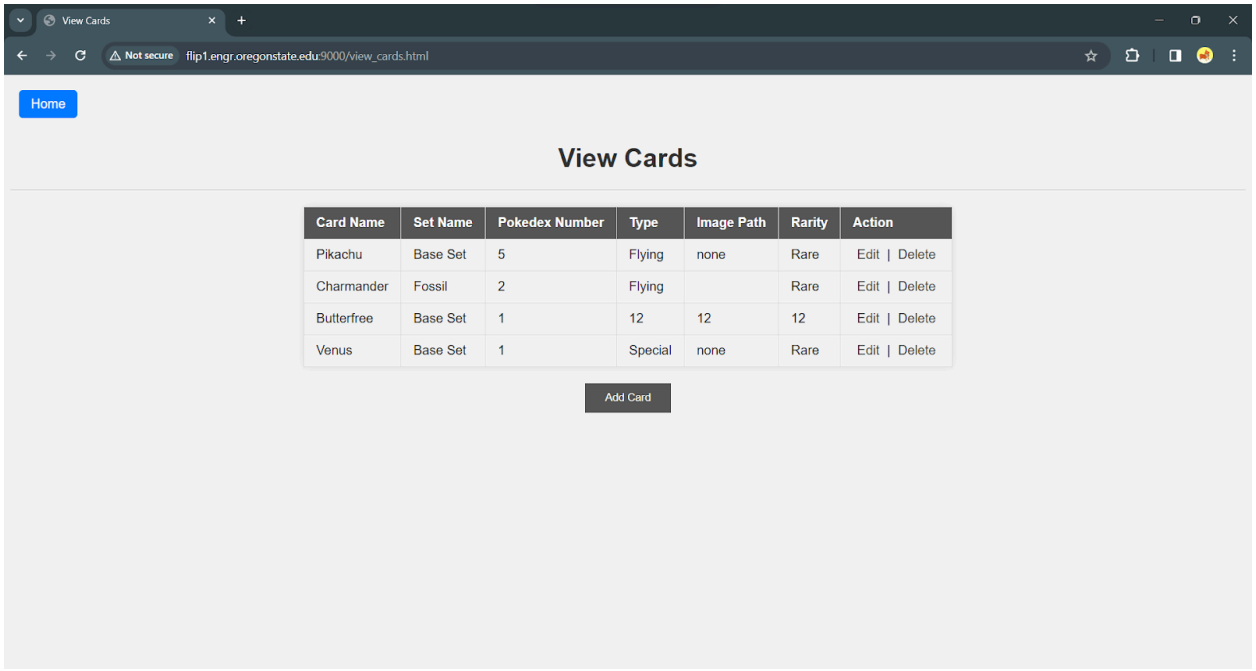
| id | card_id | deck_id |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 2 | 1 |
| 3 | 3 | 2 |

## UI Screen Shots

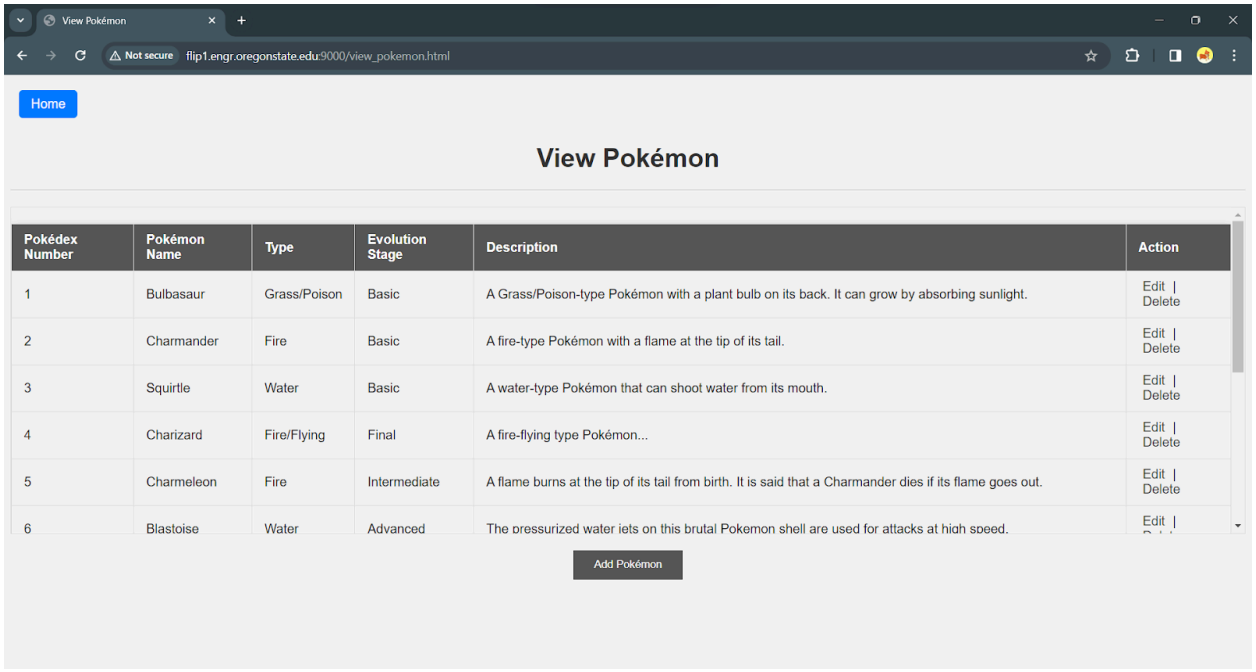
### Index Page



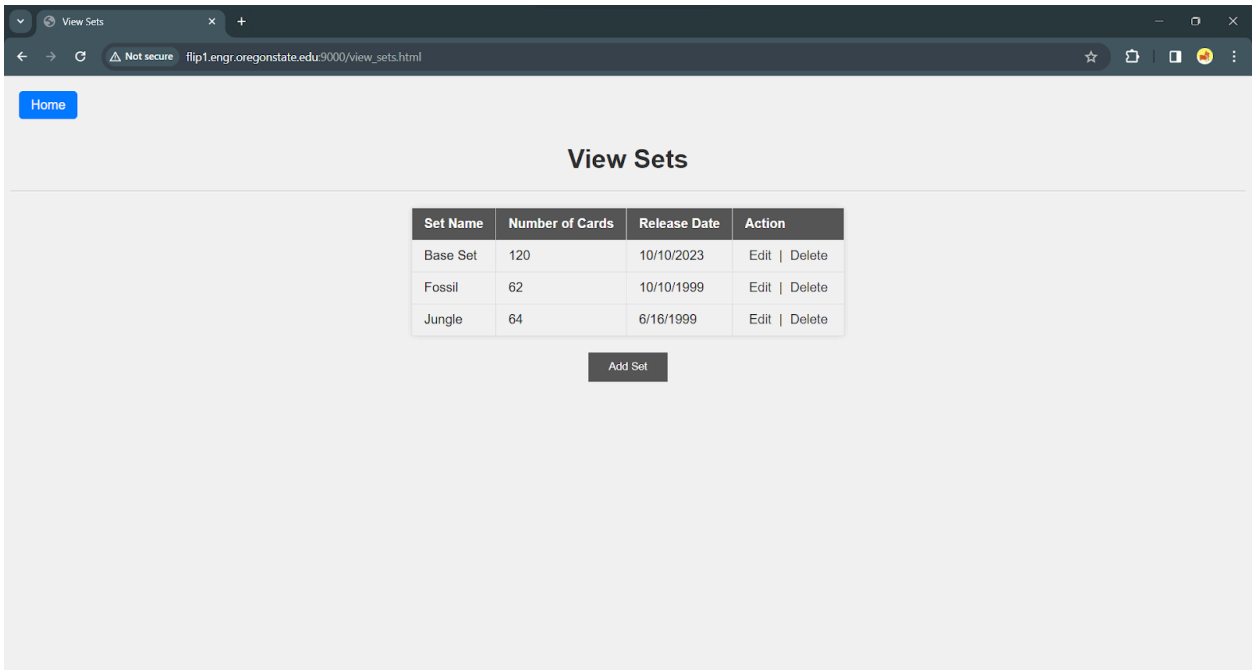

### Read Cards Page



## Read Pokemon Page

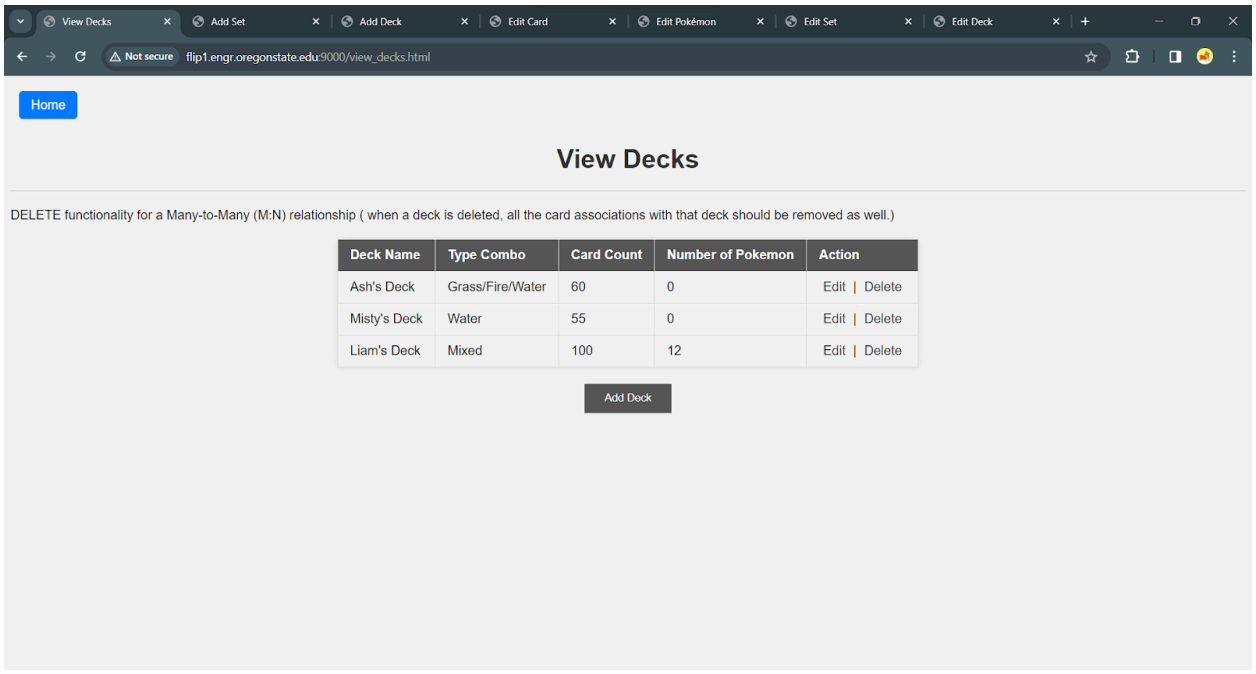



## Read Sets Page

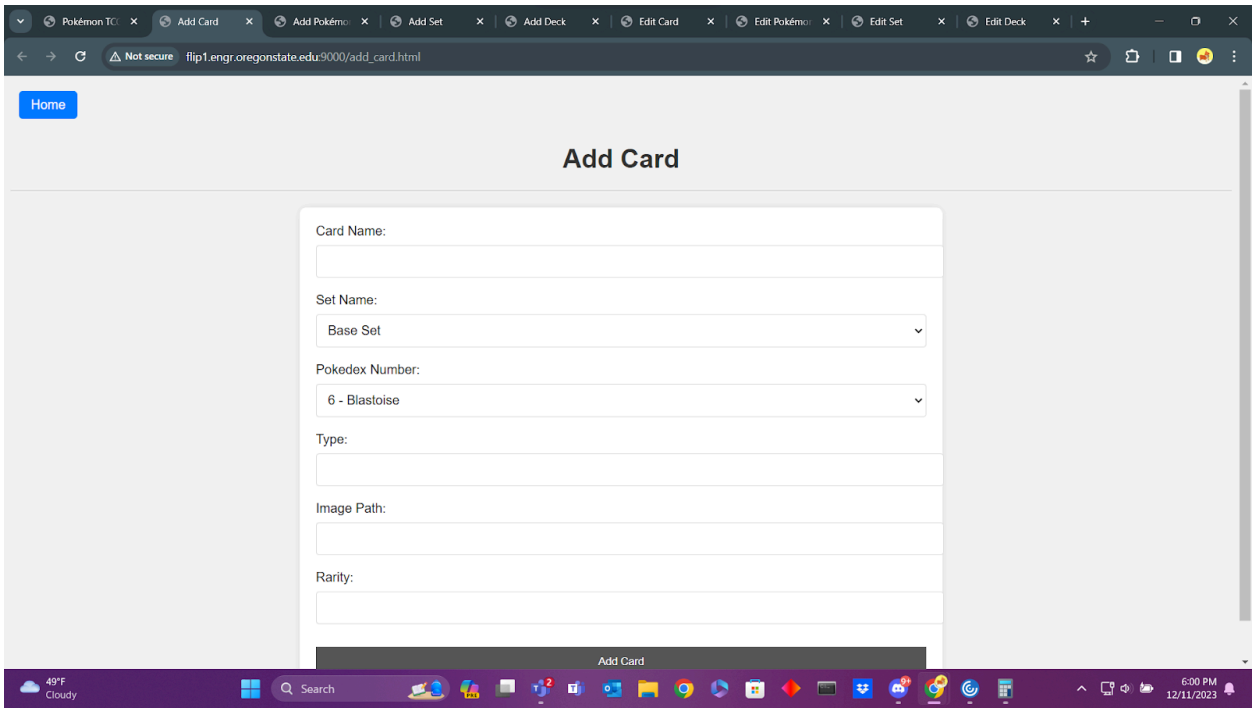


## Read Decks Page

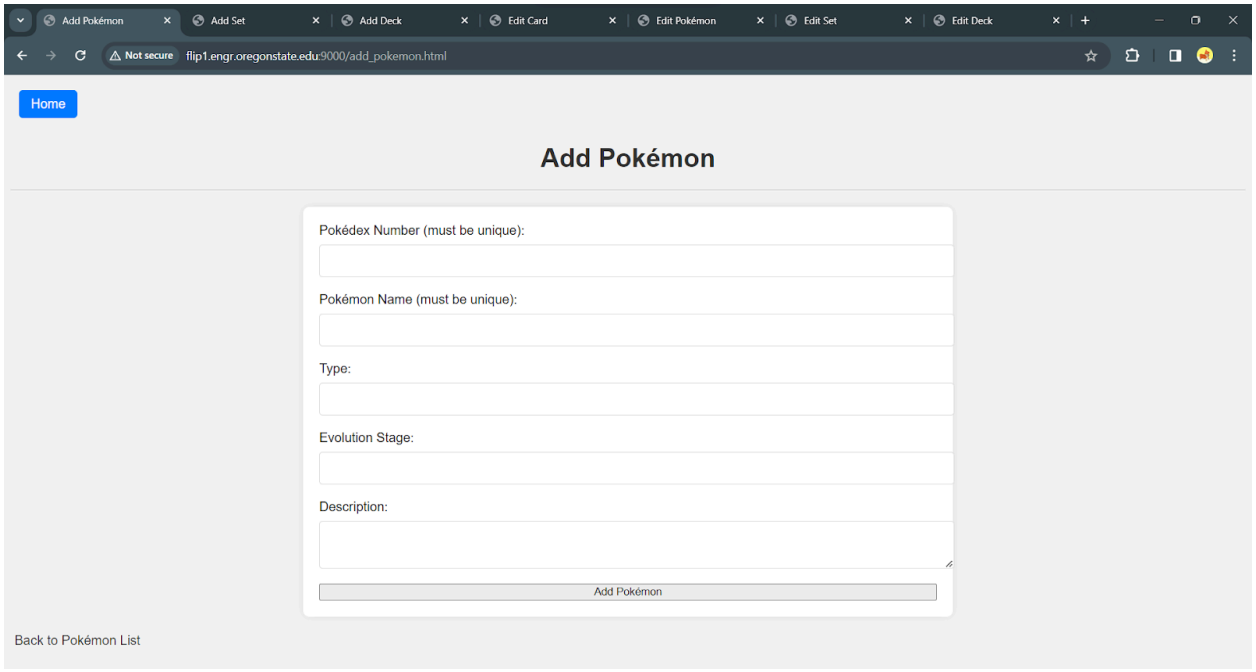
DELETE functionality for a Many-to-Many (M:N) relationship ( when a deck is deleted, all the card associations with that deck should be removed as well.)



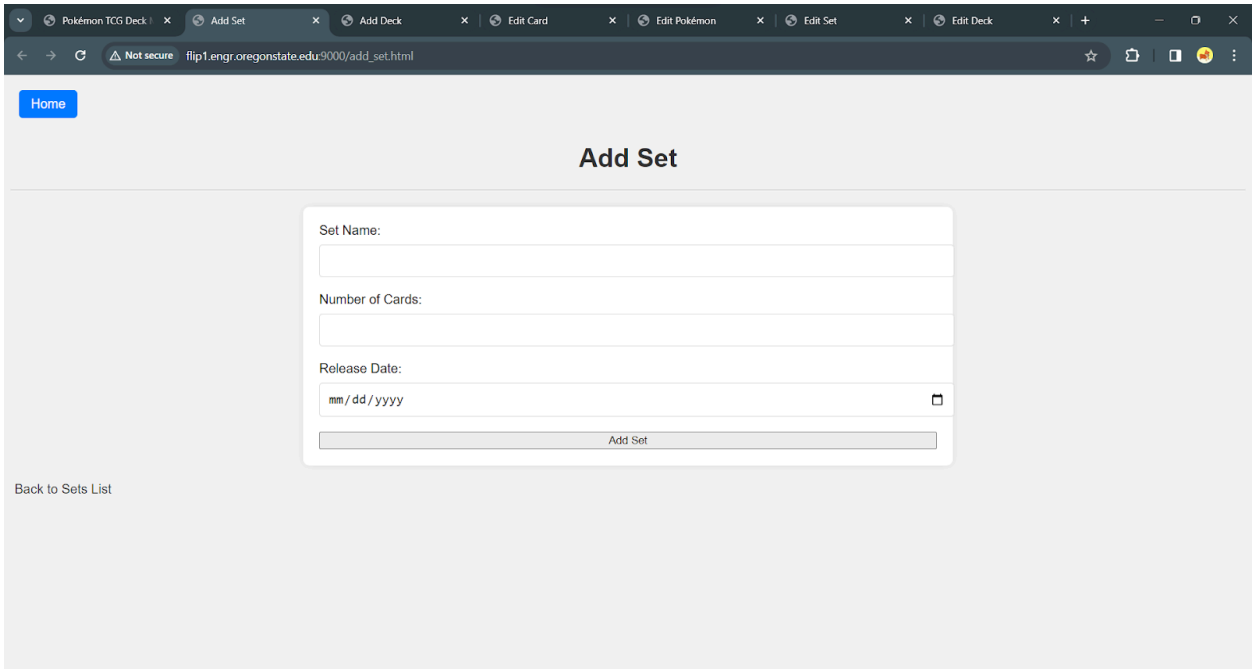

## Create New Card Page



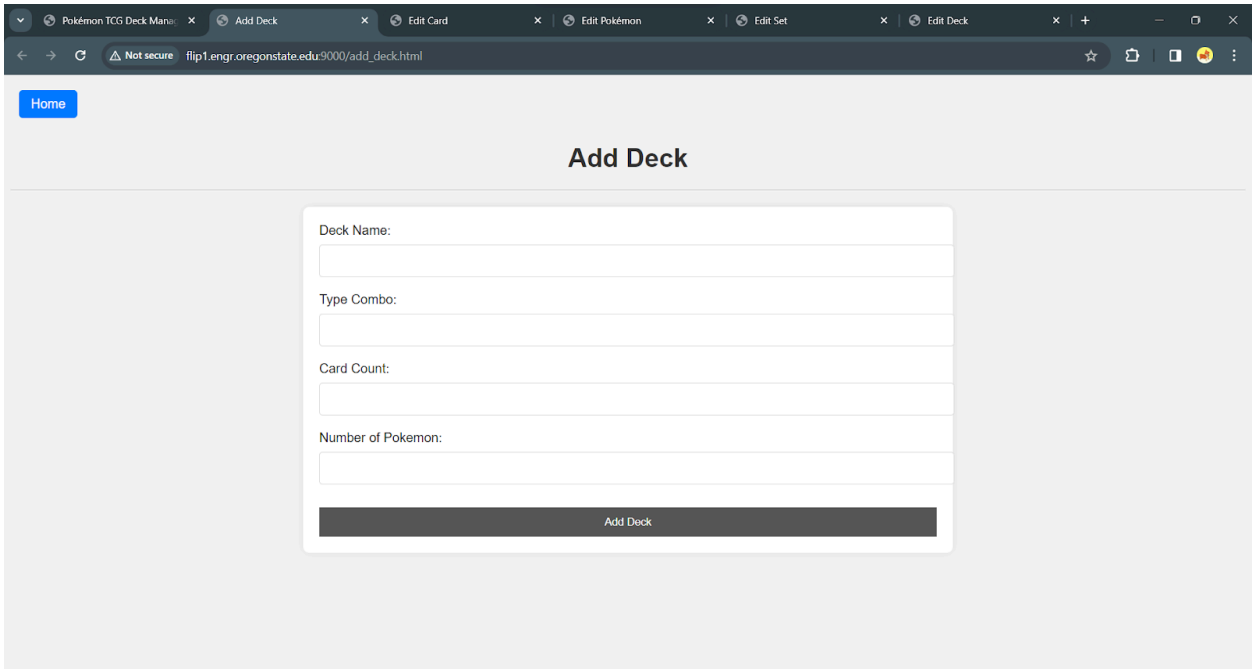
## Create New Pokemon Page



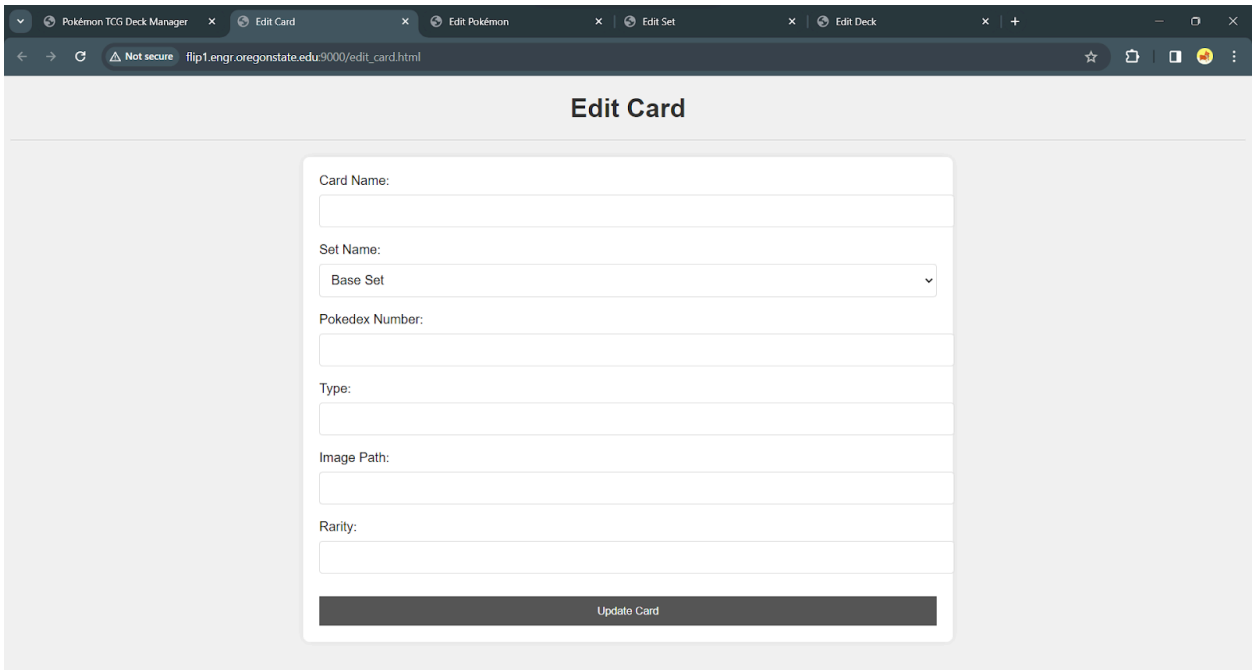

## Create New Set Page



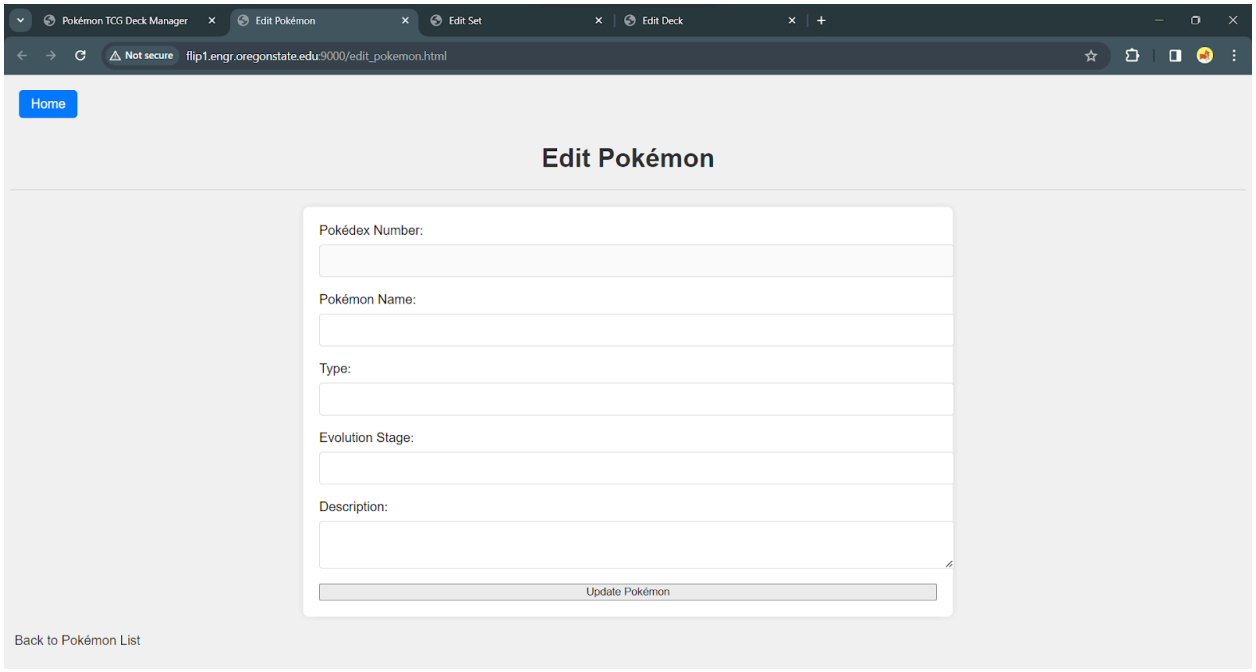
## Create New Deck Page




## Update Card Page

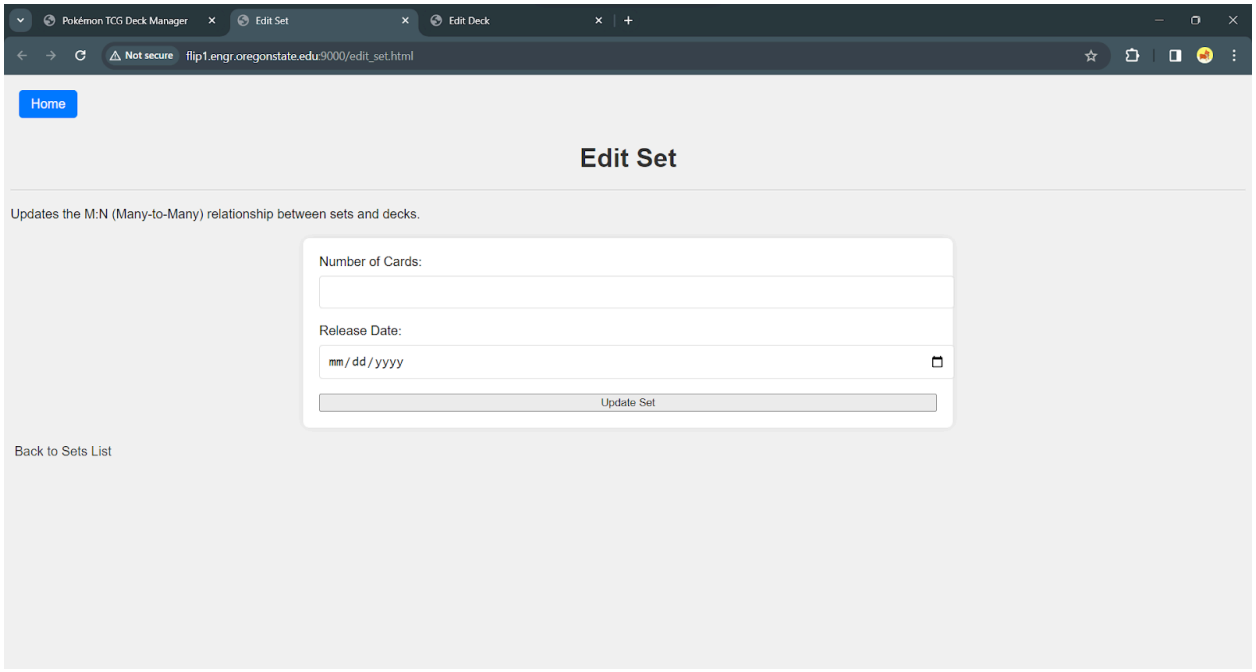
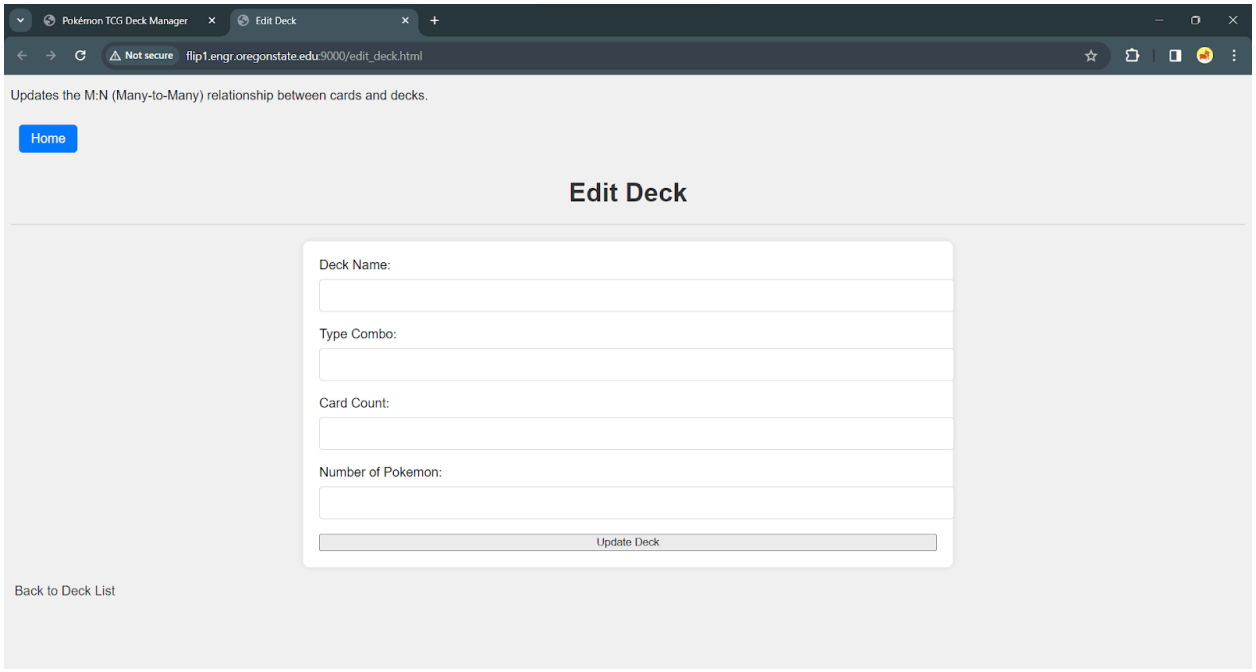


## Update Pokemon Page




## Update Set Page

Updates the M:N (Many-to-Many) relationship between sets and decks.



# Update Deck Page