



# G6 Pokedex

Justin Forkert

Spencer Mantis

Lucas Attias

# Introduction to the Database

---

- This database aims to support catching and battling Pokémon across all games by connecting players to Pokémon information including types, base stats, and generations
- Users can create an account to track the different Pokémon that they have caught
- Users are also able to query the database without an account and use the G6 Pokedex as a resource while they play the games
- The G6 Pokedex will be powered by SQL to query the database and will be a hybrid between existing Pokémon trackers and informational sites

# Agenda

- Business requirements
- Data Analysis of the Facts
- Conceptual Model
- Logical Model
- Functionality
- Application Screen Layouts
- Concluding Thoughts



# Business Requirements

- Provide a user-friendly platform to connect users to their Pokémons

- Provide a way to easily search for Pokémons by different attributes

- Provide a way for users to save Pokémons to a watchlist that can be easily accessed

- Provide a place to store Pokémons caught in games and keep track of user's progress

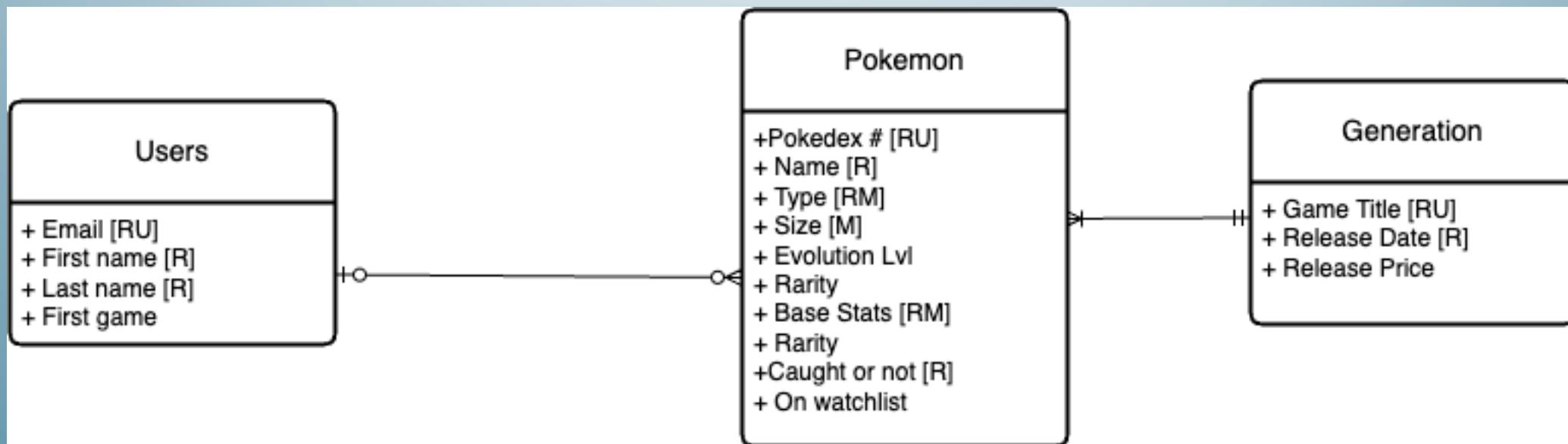
# Data Analysis of the Facts

A data analysis of the facts listing the entities, attributes, and relationships in the data model.

Entities & Attributes:				
Entity	Attribute	Props	Description	
<u>Pokemon</u>	Pokedex #	RU	Unique identification # for each Pokemon	
	Name	R	Name of Pokemon	
	Type	RM	All associated types for a pokemon	
	Size	M	Height and weight of the pokemon	
	Evolution Level		Level the Pokemon evolves at	
	Rarity		Classification of how rare a pokemon is	
	Base Stats	RM	Hp, speed, physical/special attack, physical/special defense of Pokemon	
	Caught	R	Has the user Caught It?	
	Watchlist		Has the user added it to their watchlist?	
<u>User</u>	First Name	R	First name of the user	
	Last Name	R	Last name of the user	
	Email	RU	Email of the user	
	First Game		First Pokemon game the user played	
<u>Generations</u>	Game Title	RU	Gametitle generartion was introduced in	
	Game Release Date	R	Date of generation release	
	Release Price		Price of game at release	
Relationships:				
Relationship	Entity	Rules	Min	Max
Pokemon introduced in generation	Pokemon	Is introduced in		1 1
	Generation	Introduces		1 M
User and Pokemon	User	Has		0 M
	Pokemon	Has		0 1

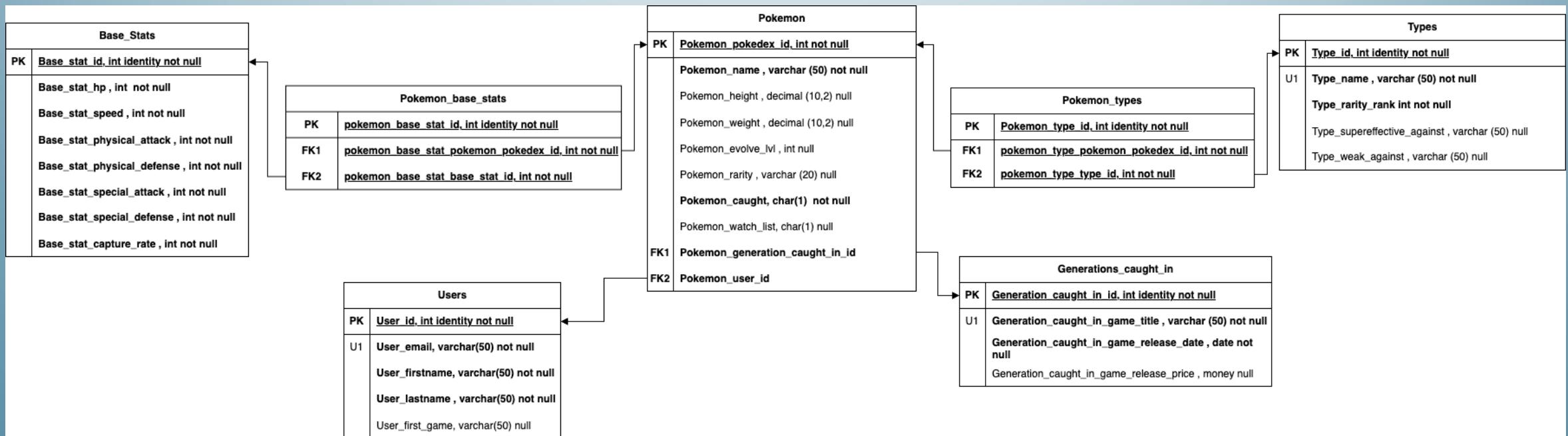
# Conceptual Model

The Pokémon entity has relationships to both the User entity and the Generation entity.

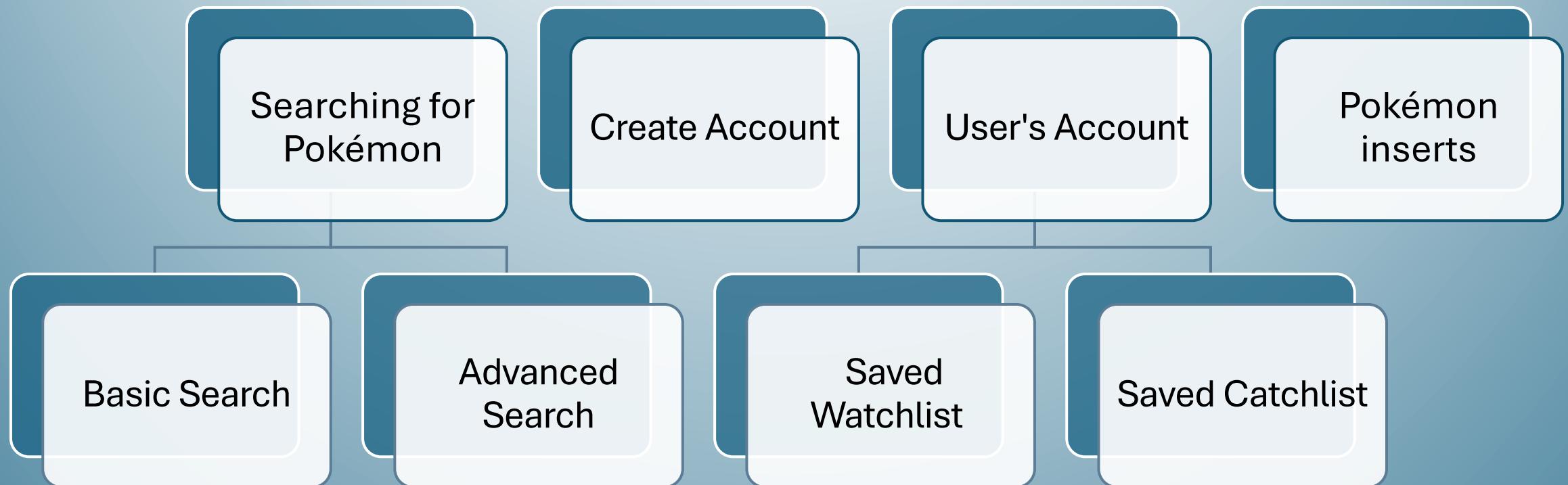


# Logical Model

Entities, attributes, and relationships are identified to display the needed data and business logic. Bridge tables were created, and Primary Keys/Foreign keys were used to ensure the integrity of the data.



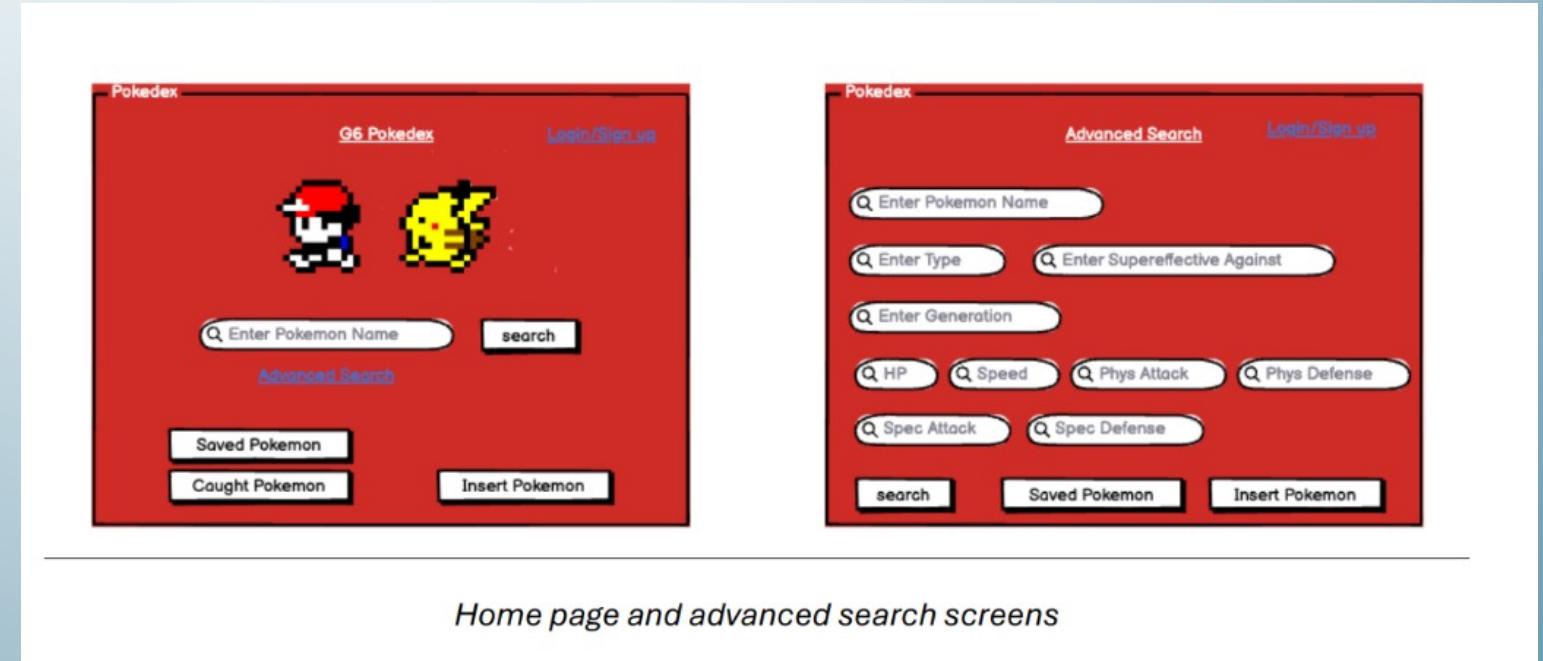
# Functionality Visualization



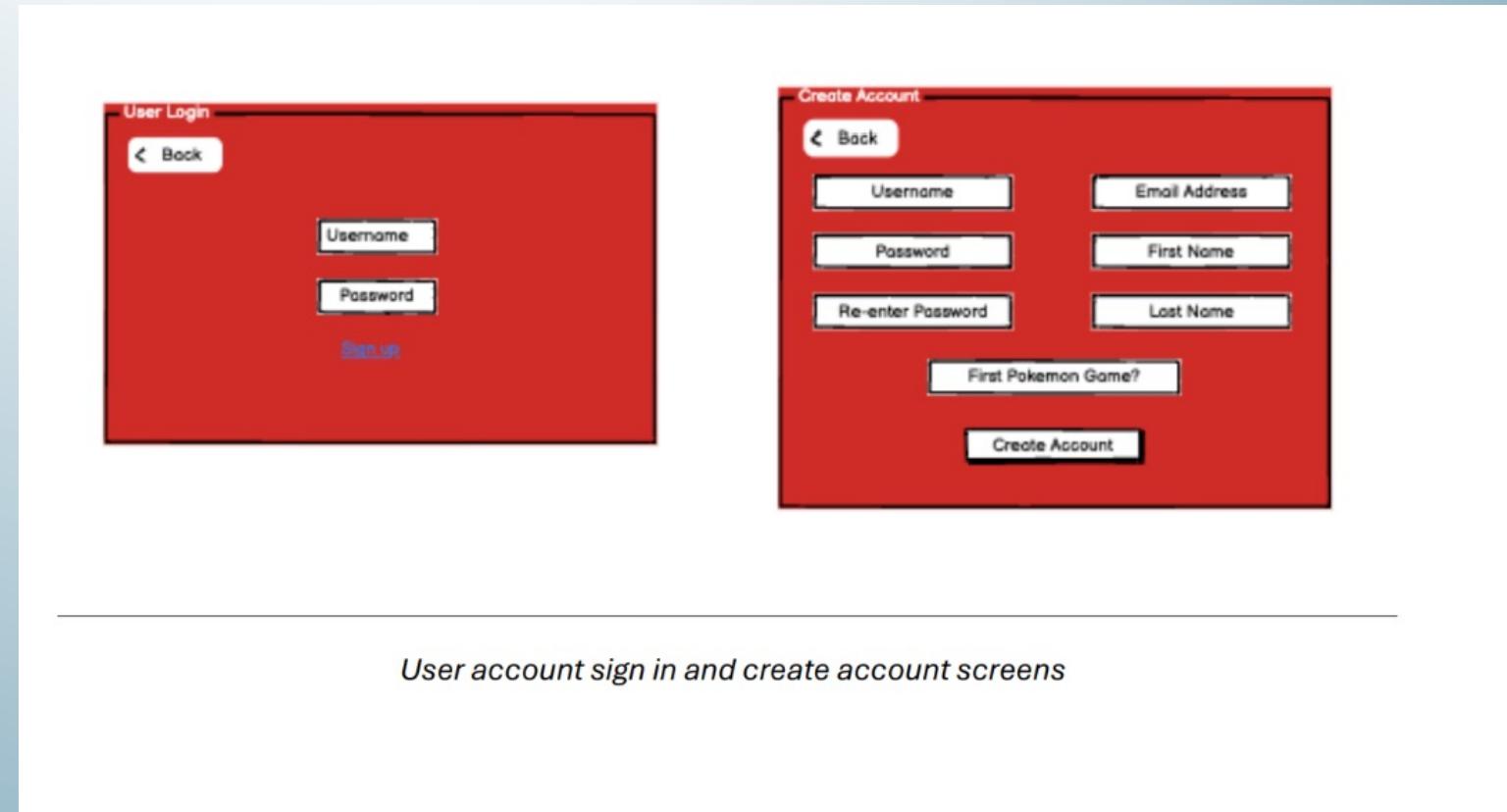
# Functionality (Questions)

1. Add transactions to ensure data integrity
2. A complex query of a specific Pokémon with a specific attribute (fastest Pokémon in gen 2)
3. Output all Pokémons that are super effective against a specific type
4. Show all Pokémons of a specific type and specific generation
5. Casual player view of the most commonly used basic info and stats
6. View of all Pokémons in gen 1, gen 2, and gen 3
7. View for super effective against every type
8. Show a user's catchlist
9. Show a user's watchlist
10. Show every Pokémon on both a watchlist and caught list to help understand how users interact with the database.
11. Show the average of a stat of all Pokémons
12. Show how many Pokémons have a base stat value greater than a specified integer

# Application Screen Layouts



# Application Screen Layouts



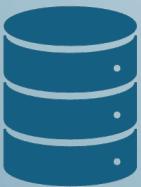
# Application Screen Layouts

The image displays six wireframe screens for a Pokémon application, arranged in two columns and three rows. Each screen shows a different table layout with various fields and navigation elements.

- Pokemon**: Shows a table with columns: Pokedex ID, Pokemon Name, Height (ft/in), Weight (lbs), Evolve Lvl, Caught?, and Generation Game Title. A "Save" button is present in the first row.
- Type**: Shows a table with columns: Type Name, Supereffective against, and Weak against. It includes entries for Grass and Poison.
- Generation**: Shows a table with columns: Generation Caught In, Game Title, Game Release Date (North America), and Game Release Price. It includes entries for 01 (Red/Blue) and 02 (Green/Yellow).
- Base Stats**: Shows a table with columns: Pokemon Name, HP, Speed, Physical Attack, Physical Defense, Special Attack, Special Defense, and Base Catch Rate. It includes entries for Bulbasaur.
- Saved Pokemon**: Shows a table with columns: Pokedex ID, Pokemon Name, Height (ft/in), Weight (lbs), Evolve Lvl, Caught?, and Generation Game Title. It includes entries for 002 (Ivysaur) and 003 (Venusaur).
- Enter Pokemon**: An input form with a header "Insert Pokemon" and a note "-If value unknown, leave blank-. It has fields for Pokedex ID, Pokemon Name, Height (ft/in), Weight (lbs), Evolve Lvl, Type Name, HP, Speed, Physical Attack, Physical Defense, Special Attack, and Special Defense. The Pokedex ID field contains "004" and the Pokemon Name field contains "Charmander".

Output of all tables, saved Pokémon watchlist, and insert Pokémon screens

# Limitations



Additional indexing needs to be added to increase the efficiency of the database



Populating the database can impact storage capabilities and overall performance



Scalability may require servers for running larger databases which can be problematic



Data can be further normalized to further secure data integrity

# Conclusion

With so many different Pokémons in existence having so many different attributes, it is currently very difficult to search for Pokémons by these attributes. It is very surprising that the most popular database currently for Pokémons, serebii.net, doesn't have this feature available to users already. Following the implementation of our database, Pokémon research becomes much more streamlined.

Team 6 created the G6 Pokedex database with the user in mind, and the G6 Pokedex gives users the edge they need to win more battles, catch more Pokémons, and be the very best that no one ever was.

