

Introduction:

Trainers are one of the essential components to the world of Pokemon. Pokemon trainers have the ability to create the perfect team combination of Pokemon as well as capture and collect the wide array of Pokemon that exist in the world. One of the main features of the game is the Pokemon Storage System, which allows any trainer to automatically store any newly caught Pokemon. Trainers are also able to access the Storage System in order to swap out current party members with Pokemon they have in storage, organize Pokemon into different boxes, and search their collection according to different search filters. In this project, we will be creating a database that will keep track of any registered Pokemon trainers as well as all the Pokemon that they have obtained on their journey.

Website Navigation:

1. Upon initialization of the website, a user is prompted to input their user information to access their storage box:

Pokemon Storage Box Login

<input type="text" value="username"/>	<input type="text" value="password"/>	<input type="button" value="Submit"/>
---------------------------------------	---------------------------------------	---------------------------------------

2. A successful login will allow the user to do a variety of search functions. The user can search between the overall pokemon database (Pokedex) as well as their own collection. The user can also logout which will allow another user to login and access the same functionality.

Pokemon Database Management System

Use the search functions below to filter Pokemon within the Pokedex.

Logout <input type="button" value="Logout"/>
Display Trainer Information <input type="button" value="Click to Display Information"/>
Pokedex Search Pokemon Name: <input type="text" value="Enter Pokemon Name"/> Type 1: <input type="text"/> Type 2: <input type="text"/> Egg Group 1: <input type="text"/> Egg Group 2: <input type="text"/> <input type="button" value="Search"/>
Display Trainer Collection <input type="button" value="Display Collection"/>
Trainer Collection Search Pokemon Name: <input type="text" value="Enter Pokemon Name"/> <input type="button" value="Search"/>
Search Pokedex Enter Pokedex Number: <input type="text" value="Enter Pokedex Number"/> <input type="button" value="Search"/>

Results of search queries will be displayed below on the webpage. Below are a few sample queries:

Display trainer information:

Trainer Name	Gender	Funds	Number of Badges
Red	M	1000000	8

Display all pokemon within trainer's collection:

Trainer ID	Capture ID	Nickname	Pokemon Name	Level	Nature	Type 1	Type 2	In Roster?
123456	1	Bulby	Bulbasaur	50	Bashful	Grass	Poison	1
123456	4	Catery	Caterpie	99	Calm	Bug	None	0
123456	7	Rattatattat	Rattata	28	Gentle	Normal	Dark	1
123456	10	Sandy	Sandslash	8	Impish	Ground	Ice	0
123456	13	Sandy	Sandslash	8	Impish	Ground	Ice	0
123456	16	Sandy	Sandslash	8	Impish	Ground	Ice	0
123456	20	Catery	Caterpie	99	Calm	Bug	None	0

Search query on Pokedex. Search for pokemon where Type 1 is 'grass'.

Pokemon Name	Pokedex Number	Type 1	Type 2	Egg Group 1	Egg Group 2
Bulbasaur	1	Grass	Poison	Monster	Grass
Ivysaur	2	Grass	Poison	Monster	Grass
Venusaur	3	Grass	Poison	Monster	Grass
Oddish	43	Grass	Poison	Grass	None
Gloom	44	Grass	Poison	Grass	None
Vileplume	45	Grass	Poison	Grass	None
Bellsprout	69	Grass	Poison	Grass	None
Weepinbell	70	Grass	Poison	Grass	None
Victreebel	71	Grass	Poison	Grass	None
Exeggcute	102	Grass	Psychic	Grass	None
Exeggutor	103	Grass	Psychic	Grass	None
Tangela	114	Grass	None	Grass	None

- Along with search queries, the user/trainer is able to register and add Pokemon to their collection, as well as update the 'in roster' attribute.

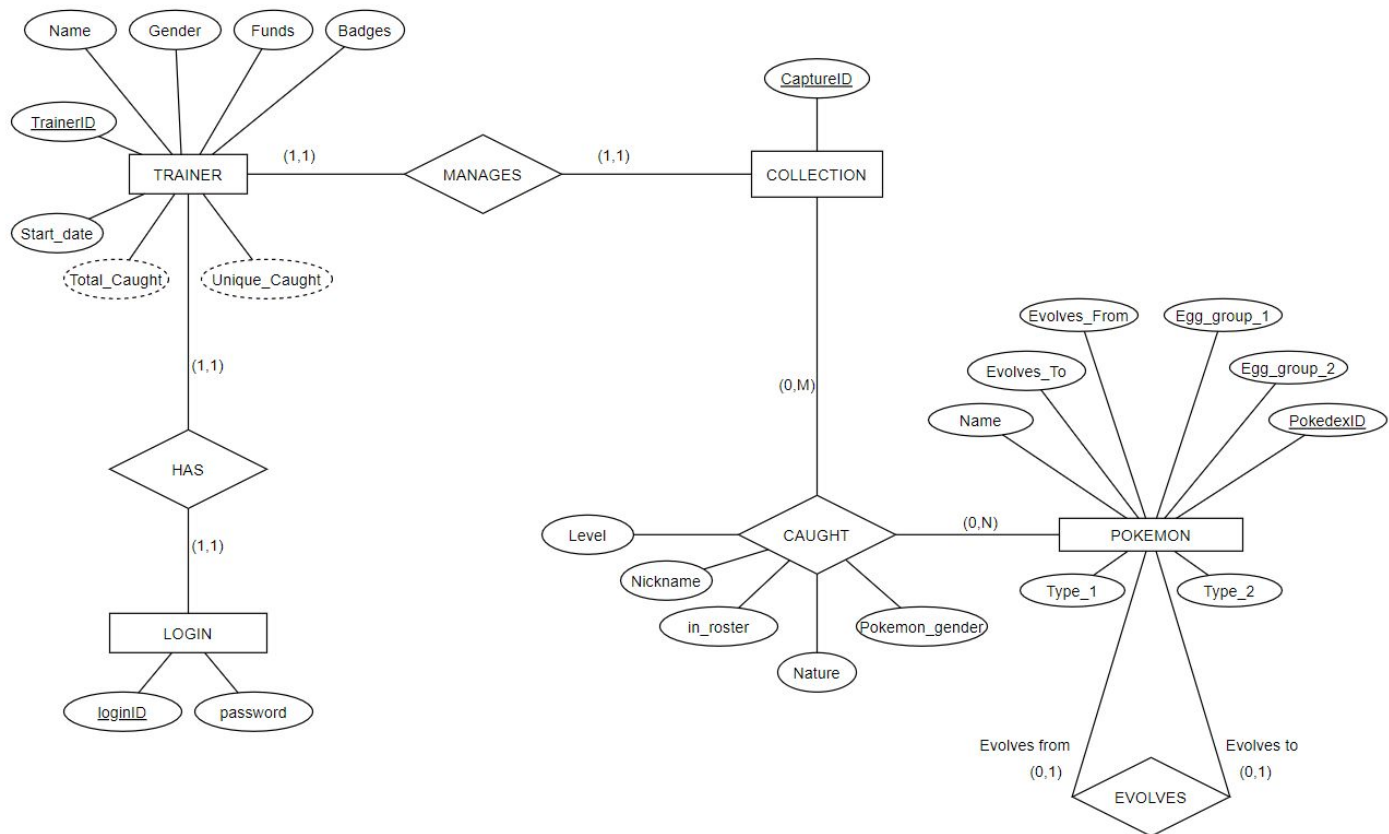
Add Pokemon	
Enter captureID	<input type="text"/>
Trainer ID	<input type="text"/>
Gender	<input type="text"/>
Pokedex Number	<input type="text"/>
Pokedex Level	<input type="text"/>
Nickname	<input type="text"/>
Nature	<input type="text"/>
<input type="button" value="Add"/>	

Switch Pokemon Lineup	
Enter captureID	<input type="text"/>
Trainer ID	<input type="text"/>
<input type="button" value="Switch Roster"/>	

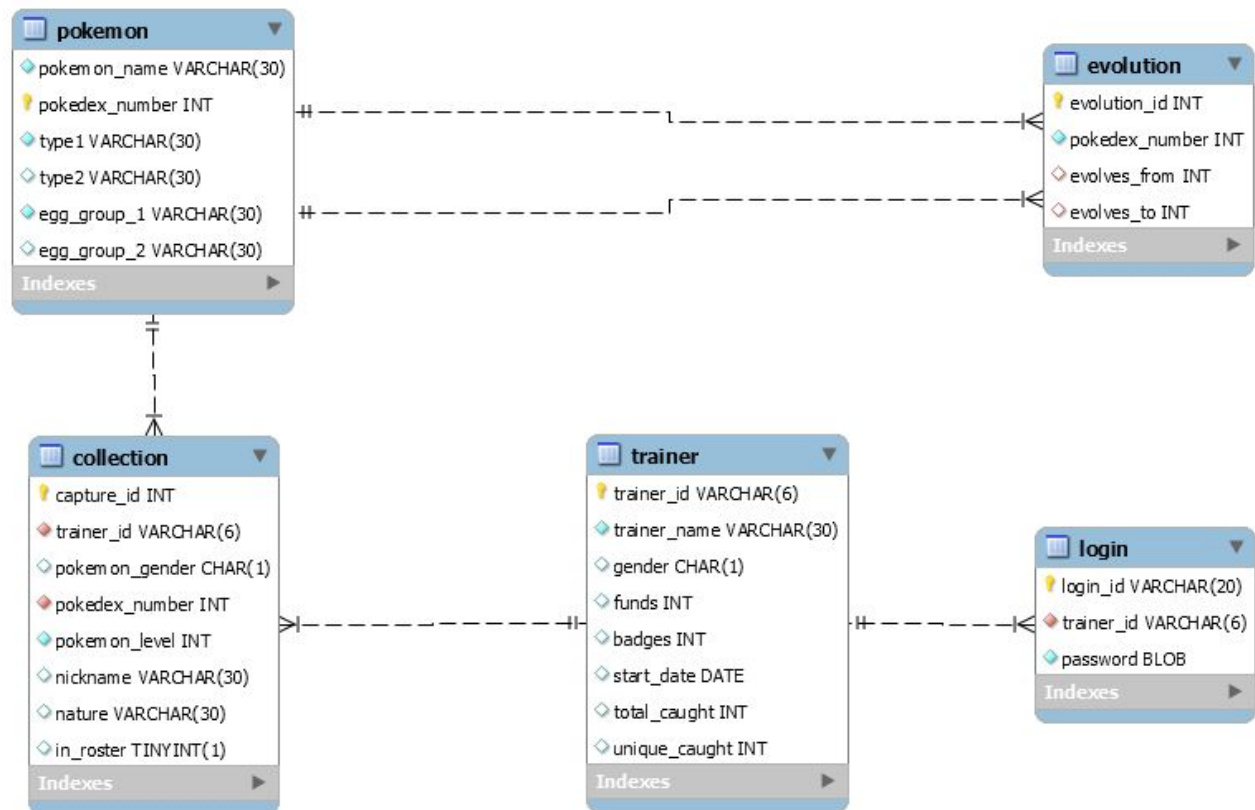
Implementation:

Originally we were going to create an MVC project using ASP.NET, C#, and Sql Server. We ran into problems trying to use ASP.NET and decided to go with what we kind of understood. So instead we used Python over C# and learned how to use a Flask extension. There still were issues in trying to download and install the correct libraries but we were able to work through those issues. Our project ended up being 2-tiered.

ER Diagram:



Relational Schema Diagram:



Requirements:

- We store the Pokémon Trainer's name, sex (gender), trainer ID number, current funds, the number of gym badges collected, the trainer's start date, and the number of unique Pokémon that they have caught. Trainers are unique to the trainer ID number.
- Each Pokémon has a unique name and a unique national Pokedex number. A Pokémon has two types (type 1 and type 2), where they must have one type, but can sometimes have a second type. Pokémon also has different egg groups (group 1 and group 2) where they must have an egg group type, but can also have a second type. Some Pokémon can have up to three stages of evolution, while others may have no evolution lines.
- A trainer's party (or active roster) consists of a number of Pokémon. A trainer must have at least one Pokémon in its party, and can only have a maximum of six.
- Trainer's are assigned a storage box where they can keep recently captured Pokémon whenever their party is full. Boxes are unique by box number as well as the trainer's ID.
- Any Pokémon that are captured and sent to either the trainer's party or storage box are assigned a nickname, level, gender (if applicable), and nature.

Future scope:

With the current build of the project, there are multiple features and updates that can be implemented in future iterations of the project:

- Addition of information to Pokemon table:
 - Pokedex description
 - Pokemon Images
 - Abilities
 - Entries of Pokemon from other game titles
- Update of styling to website
- Create navigational menu to separate website by types of function
- Build off of current database to add more tables:
 - Item Storage(Pokeballs, potions, rare candy, etc.)
 - Ability to keep track of items held by trainer's Pokemon
- Trading Pokemon between trainers
- Hosting database/website online on cloud based servers
- Implementation of administrative access to the database

Deficiencies:

Currently, there are multiple deficiencies within the project that can be improved in future iterations of the project. With issues with our initial database architecture (unable to connect to database using ASP.NET as planned in Milestone 1), the project had to be quickly migrated to using a two-tier architecture with python, flask, and MySQL connector. Below are a few of the deficiencies of the current project:

- Lack of styling with current build
- Separation of different functions within the web page. Current build has everything located within one web page
- Lack of implementation with evolutions in the pokemon table
- Abilities were taken off due to the complexity of what was needed at the time to make it work