Description

"CardQuest Master" is a web-based system designed to track and manage Pokémon card collections. This application serves as a repository for essential information regarding trainers, a wide array of Pokémon cards, and the collections amassed by each trainer.

Key Features of the Project:

- **Trainers Management:** Easily manage and maintain records of trainers, including their name, location, and email address. The project allows for seamless addition, editing, and deletion of trainer information.
- **Pokémon Cards Catalog:** Keep an organized database of Pokémon cards yet to be collected, enabling users to browse through available cards.
- **Collection Tracking:** Efficiently record and track the Pokémon cards collected by each trainer, providing a comprehensive overview of their amassed collection.

Entity Relationship Diagram (ERD)

Starting Django

Setting Up

Create a virtual environment

```
$ virtualenv [virtualenv_name]
```

Create a repository in Github

Activate the virtual environment (linux / macOS)

```
$ source bin/activate
```

Clone repository inside virtual environment

Install Django

```
$ pip install Django
```

Create project

```
$ django-admin startproject [project_name]
$ django-admin startproject projectsite
```

Open folder [project_name] in VS Code or preferred editor.
First [project_name] is the root directory of the project. The second [project_name] is the actual project directory.

Create application (you can have multiple app inside a project). Go to directory where you can find the manage.py.

```
$ python manage.py startapp [app name]
```

Register app in [project_name]/settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'cardquest', # <-- application name

]
```

Setup database by creating migrations and migrate

```
$ python manage.py makemigrations
$ python manage.py migrate
```

Run python server

```
$ python manage.py runserver
```

Open and inspect database using TablePlus
Create readme.md and .gitignore

Create models

Go to cardquest/models.py

Create a basemodel to inherited by all other class model

```
from django.db import models


class BaseModel(models.Model):
    created_at = models.DateTimeField(
        auto_now_add=True, db_index=True)
    updated_at = models.DateTimeField(auto_now=True)

    class Meta:
        abstract = True
```

```
class Trainer(BaseModel):
    name = models.CharField(max_length=100, null=True, blank=True)
    birthdate = models.DateField(null=True, blank=True)
    location = models.CharField(max_length=250, null=True, blank=True)
    email = models.EmailField(max_length=100, null=True, blank=True)

    def __str__(self):
        return self.name


class PokemonCard(BaseModel):
    RARITY_CHOICES = (
        ('Common', 'Common'),
        ('Uncommon', 'Uncommon'),
        ('Rare', 'Rare'),

    )
    CARDTYPE_CHOICES = (
        ('Fire', 'Fire'),
        ('Water', 'Water'),
        ('Grass', 'Grass'),
        ('Electric', 'Electric'),
        ('Psychic', 'Psychic'),
        ('Ice', 'Ice'),
        ('Dragon', 'Dragon'),
        ('Dark', 'Dark'),
        ('Normal', 'Normal'),
        ('Fighting', 'Fighting'),
        ('Flying', 'Flying'),
        ('Poison', 'Poison'),
        ('Ground', 'Ground'),
        ('Rock', 'Rock'),
        ('Bug', 'Bug'),
        ('Ghost', 'Ghost'),
        ('Steel', 'Steel'),
        ('Fairy', 'Fairy'),
    )
    name = models.CharField(max_length=100, null=True, blank=True)
    rarity = models.CharField(
        max_length=100, null=True, blank=True, choices=RARITY_CHOICES)
    hp = models.IntegerField(null=True, blank=True)
    card_type = models.CharField(
        max_length=100, null=True, blank=True, choices=CARDTYPE_CHOICES)
    attack = models.CharField(max_length=100, null=True, blank=True)
    description = models.CharField(max_length=250, null=True, blank=True)
    weakness = models.CharField(max_length=250, null=True, blank=True)
    card_number = models.IntegerField(null=True, blank=True)
    release_date = models.DateField(null=True, blank=True)
    evolution_stage = models.CharField(max_length=250, null=True, blank=True)
    abilities = models.CharField(max_length=250, null=True, blank=True)


class Collection(BaseModel):
    card = models.ForeignKey(PokemonCard, blank=True,
                             null=True, on_delete=models.CASCADE)
    trainer = models.ForeignKey(
        Trainer, blank=True, null=True, on_delete=models.CASCADE)
    collection_date = models.DateField()
```

Register models in admin.py

```
from django.contrib import admin
from .models import PokemonCard

admin.site.register(PokemonCard)
```

Create superuser account

```
$ python manage.py createsuperuser
```

Initial load data: Create app_name/management/commands/create_initial_data.py

```
# your_app/management/commands/create_initial_data.py

from django.core.management.base import BaseCommand
from cardquest.models import PokemonCard, Trainer


class Command(BaseCommand):
    help = 'Creates initial data for the application' #<-- description of the command

    def handle(self, *args, **kwargs):
        self.create_pokemon_cards() # <-- where logic is implemented
        # self.create_trainers()

    def create_pokemon_cards(self):
        # Create Pokemon Card instances
        card1 = PokemonCard(name="Pikachu", rarity="Rare", hp=60,
card_type="Electric", attack="Thunder Shock", description="A mouse-like pokemon that
can generate electricity.",
                            weakness="Ground", card_number=25, release_date="1999-01-
09", evolution_stage="Basic", abilities="Static")
        card1 = PokemonCard("Pikachu", "Rare", 60, [
                            "Electric"], "Thunder Shock", "A mouse-like pokemon that
can generate electricity.", ["Ground"], 25, "Basic", ["Static"])

        card1.save() #<-- save card1 to PokemonCard table
        self.stdout.write(self.style.SUCCESS(
            'Successfully created Pokemon cards.')) #<-- display success message

    def create_trainers(self):
        pass
```

Then, you can run this command using `python manage.py create_initial_data`.

Modify the admin.py for improvement

```
@admin.register(PokemonCard)
class PokemonAdmin(admin.ModelAdmin):
    list_display = ("name", "rarity")
    search_fields = ("name",)
```

First, make sure that your virtual environment is active. In the path where the manage.py is located, create requirements.txt. Using the terminal use the command below. It will automatically create a requirements.txt file that contains all the modules used in the virtual environment.

```
$ pip freeze > requirements.txt
```

Commit changes to git repository
Go to https://www.pythonanywhere.com/pricing/
click "Create a Beginner account"
username: [your-username]
email: your email add
password: your password
click "Register"
click "$Bash"
In Bash terminal, create virtual environment named venv

```
$ virtual venv
```

clone the project's repository

```
$ git clone [your-git-repo-url]
```

Activate virtual environment

```
$ source venv/bin/activate
```

change directory inside cardquest-master-app

```
$ cd cardquest-master-app
```

install requirements.txt

```
$ pip install -r requirements.txt
```

From Menu go to Web
Click "Add a new web app" > Next > Click "Manual Configuration"
Select the version of your python > Next
Configure WSGI file

```
  import os
  import sys

#path where you can find the manage.py
  path = '/home/cardquest/cardquest-master-app/projectsite'
  if path not in sys.path:
      sys.path.insert(0, path)
  os.environ['DJANGO_SETTINGS_MODULE'] = 'projectsite.settings'
  from django.core.wsgi import get_wsgi_application
  from django.contrib.staticfiles.handlers import StaticFilesHandler
  application = StaticFilesHandler(get_wsgi_application())
```

Set virtualenv path /home/cardquest/venv
Set static files path: /home/cardquest/cardquest-master-app/projectsite/static
Click Reload cardquest.pythonanywhere.com

Preparing static files

- inside cardquest folder,
- create static folder
- inside static folder create images folder
- copy bg.png
- file structure now should look like this static/images/bg.png
- inside static create css folder
- file structure now should look like this static/css/style.css
- copy style.css from https://github.com/NaldCapuno/pokemon
- replace in style.css

```
/* before */

      body {
            background-image: url('/img/bg.png');

/* after */
      body {
            background-image: url('../images/bg.png');
```

- inside cardquest folder
- create templates folder; inside that folder create includes folder
- create footer.html inside includes (file structure update: templates/includes/footer.html)
- copy footer snippet in footer.html

```
          <footer>
          <br />
          <br />
          <p>
            POKEMON<br />
            Gotta Catch 'Em All
          </p>
          <br>
          <div class="social-icons">
            <a href="https://github.com/developer" target="_blank" class="icon-
      link">
              <i class="fab fa-github"></i>
            </a>
            <a href="https://facebook.com/yourpage" target="_blank" class="icon-
      link">
              <i class="fab fa-facebook"></i>
            </a>
            <a href="https://instagram.com/youraccount" target="_blank"
      class="icon-link">
              <i class="fab fa-instagram"></i>
            </a>
            <!-- Add other social media icons similarly -->
          </div>
          <p>
            &copy; 2023 YourWebsiteName. All Rights Reserved.
          </p>
      </footer>
```

- templates/base.html
- add static before html code

```
        {% load static %}
```

- copy home.html from pokemon to base.html
- modify stylesheet link

```
        /* before */

            <link rel="stylesheet" href="css/style.css" />
```

```
      /* after */

      <link rel="stylesheet" href="{% static 'css/style.css' %}" />replace title
```

- replace title

```
/* before */
      <title>Pokemon | Home</title>
/* after */
      <title>{% block title %} CardQuest Master {% endblock %}</title>
```

- replace content convert to block

```
/* before */
<!--CONTENT-->
    <main>
      <section class="home-content"></section>
    </main>

/* after */
<!--CONTENT-->
    <main>
      {% block content %}

      {% endblock%}
    </main>
```

- include footer.html in base.html

```
/* before */
<footer>
    <br />
    <br />
    <p>
      POKEMON<br />
      Gotta Catch 'Em All
    </p>
    <br>
    <div class="social-icons">
      <a href="https://github.com/developer" target="_blank" class="icon-link">
        <i class="fab fa-github"></i>
      </a>
      <a href="https://facebook.com/yourpage" target="_blank" class="icon-link">
        <i class="fab fa-facebook"></i>
      </a>
      <a href="https://instagram.com/youraccount" target="_blank" class="icon-
link">
        <i class="fab fa-instagram"></i>
      </a>
      <!-- Add other social media icons similarly -->
    </div>
    <p>
      &copy; 2023 YourWebsiteName. All Rights Reserved.
    </p>
</footer>

/* after */
  {% include 'includes/footer.html' %}
```

cardquest/views.py
add to the existing code

```
from django.views.generic.list import ListView
from cardquest.models import PokemonCard


class HomePageView(ListView):
    model = PokemonCard
    context_object_name = 'home'
    template_name = "home.html"

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        return context
```

cardquest/url.py

```
from django.urls import path
from cardquest.views import HomePageView      <--
from cardquest import views                          <--

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.HomePageView.as_view(), name='home'), <--
]
```

Run server python3 manage.py runserver
Work on page Trainer
views.py

```
from cardquest.models import PokemonCard, Trainer

class TrainerList(ListView):
    model = Trainer
    context_object_name = 'trainer'
    template_name = 'trainers.html'
    paginate_by = 15
```

url.py

```
from cardquest.views import HomePageView, TrainerList <--

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.HomePageView.as_view(), name='home'),
    path('trainer_list', TrainerList.as_view(), name='trainer-list'), <--
```

trainers.html

```
{% extends 'base.html' %}
{% load static %}

{% block content %}
    <!-- section here -->
{% endblock %}
```

modify base.html

```
<!--NAVBAR-->
<a href="{% url 'trainer-list' %}" class="btn"> Trainer </a>
       Trainer
```