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",)
```