

Advanced Web Development Midterm

How to run the application?

Python version 3 is required to run this project. I tested it using Python version 3.9.6 so in case of any issue, please ensure Python version is up to date.

In order to run the application, please perform the following actions and commands in your OS terminal in the given order:

1. Create a Python virtual environment by running command `python -m venv <my_env>`.
2. Activate the virtual environment by running `my_env/Scripts/activate`.
3. Install the required packages by running `pip install -r /path/to/requirements.txt`
4. Navigate to directory `/midterm` and then run `python manage.py runserver`

How to import data?

All the migrations have been run and seed data preloaded in the included `db.sqlite3` file. However, if anything goes wrong, the data maybe reimported by taking the following steps:

1. Delete the file `db.sqlite3`.
2. Run `python manage.py makemigrations proteins`
3. Run `python manage.py migrate`
4. Run `cd scripts`
5. Run `python populate-proteins-db.py`

Models

I imported the provided data into the following models/tables:

1. Organism
2. Pfam
3. Domain
4. Protein
5. ProteinDomainMapping (exists solely to map domains to proteins)

In order to ensure the data is normalized and minimal redundant data is present in the tables, I separated out the organisms data into their own model: `Organism`. Since each `Protein` belongs to an organism, it uses a foreign key `taxonomy` to reference `Organism` model.

I also separated out the `Domain` and `ProteinDomainMapping` tables because this way, the mapping table didn't have extra columns which would be tricky to include with the relations. Keeping the mapping table separate from the Domain data made the process really simple with Django rest framework doing the include resolution for us.

I preferred to explicitly declare the primary keys for most of my tables because the IDs in Protein, Organism and Pfam table are already unique.

REST endpoints

I implemented the following REST endpoints

```
POST http://127.0.0.1:8000/api/protein/  
GET  http://127.0.0.1:8000/api/protein/[PROTEIN_ID]  
GET  http://127.0.0.1:8000/api/pfam/[PFAM_ID]  
GET  http://127.0.0.1:8000/api/proteins/[TAXA_ID]  
GET  http://127.0.0.1:8000/api/pfams/[TAXA_ID]  
GET  http://127.0.0.1:8000/api/coverage/[PROTEIN_ID]
```