Exercise: Models and migrations

Launch lab

Instructions

Goal

To create a model, learn to perform migrations and track changes in the model.

Objectives

The learner will create a model called Drinks

The learner will perform the steps required to apply migrations

Introduction

By now you have learned how to work with the view part of the Model View Template (MVT) pattern.

You can create view functions and map them to URLs.

The labs in this model will focus on the model part of the MVT pattern and in this lab, you will create and migrate a model called `Drinks`.

Scenario

The Little Lemon prototype web application is progressing well. Now that you have view function mapped to URLs, it's time to work with the data model.

You are assigned the task of creating a model that will be used as a database for storing the different menu items that are to be displayed on the Little Lemon website. In this scenario, you will design a model for a drinks menu that will list the beverages available at Little Lemon.

Learner Instructions

This lab will require you to modify the following files:

models.py

admin.py

Additionally, you are required to use the command line console inside the terminal of VS Code.

If not open already, go to **Terminal** on the Menu bar at the top of your screen and select **New Terminal**.

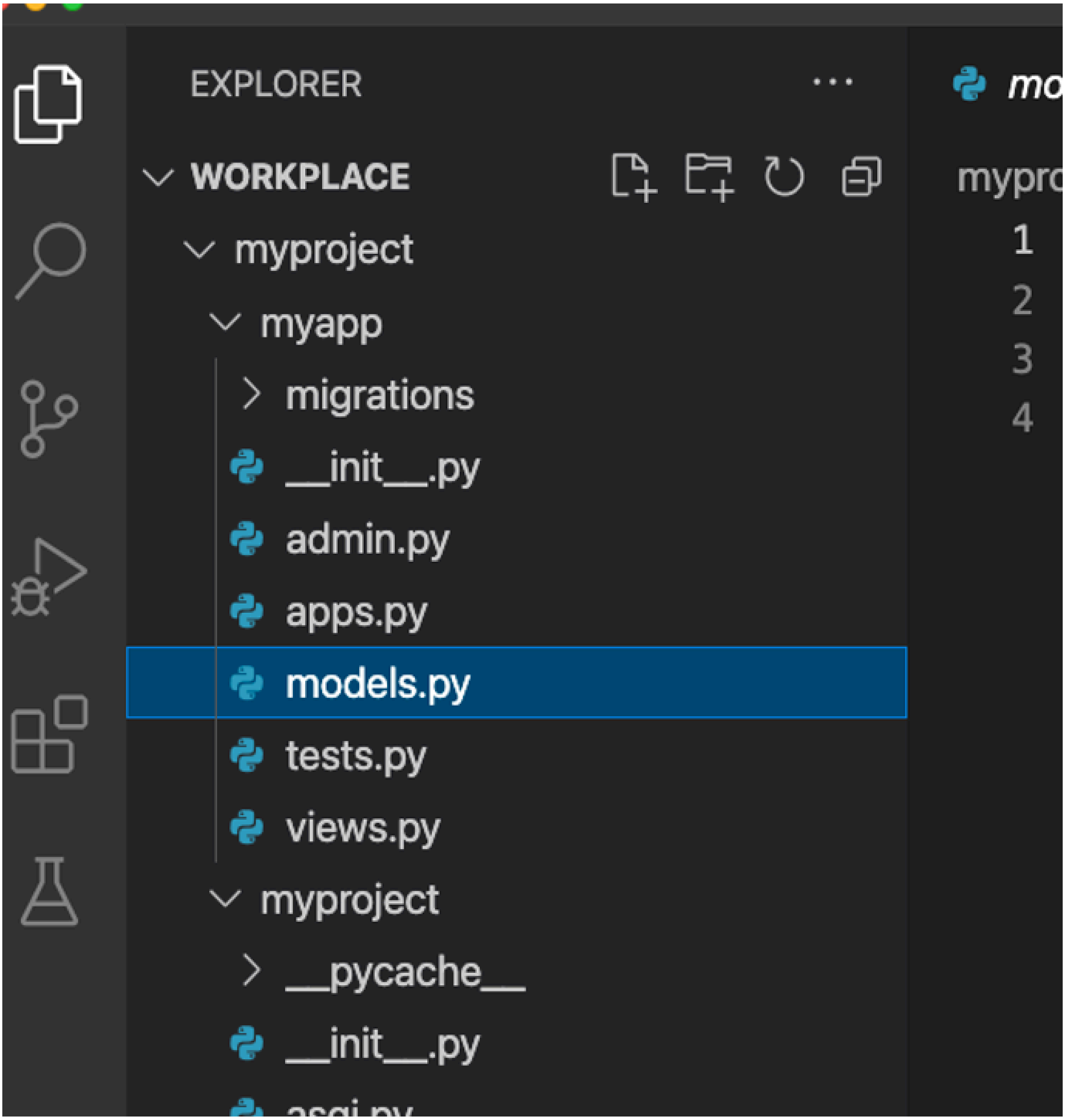
You have already built the project named **myproject** and added an app inside the project called **myapp**.

Follow the instructions below and ensure you check the output at every step and update the necessary files to create different views.

Steps

Step 1:

Expand the **myapp** folder in the left explorer panel and open the **models.py** file.



Step 2:
Create a class called `Drinks` and pass `models.Model` to it as an argument.
Step 3:
Inside the `Drinks` class, create the two attributes that you will require in the model and assign the respective form fields to them.
Additionally, pass the following specified arguments to those form fields:

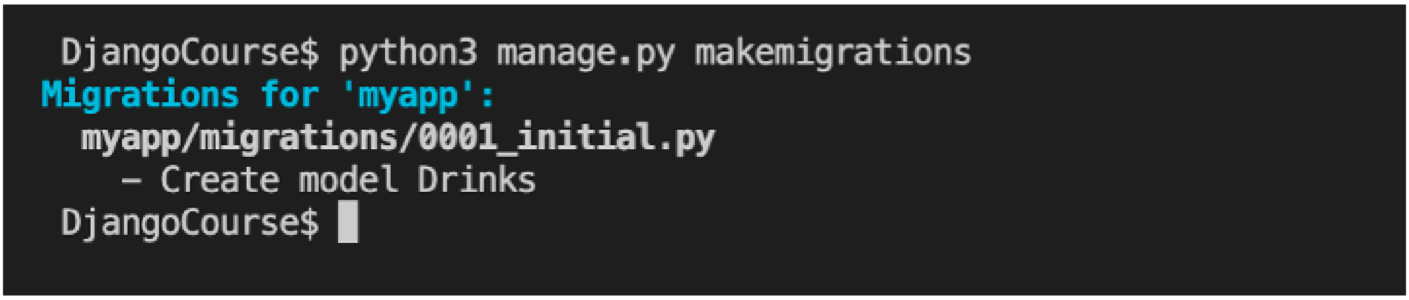
Attribute	Form field type	Arguments passed
drink	`CharField`	max_length = 200
price	`IntegerField`	

Tip: Remember that the form fields are present inside the `models` module imported by default.

Step 4:
Open the **admin.py** file and register the Model that you created by passing the model name to the function `admin.site.register()`.
Tip: Make sure you have imported the `Drinks` model from the **models.py** file.

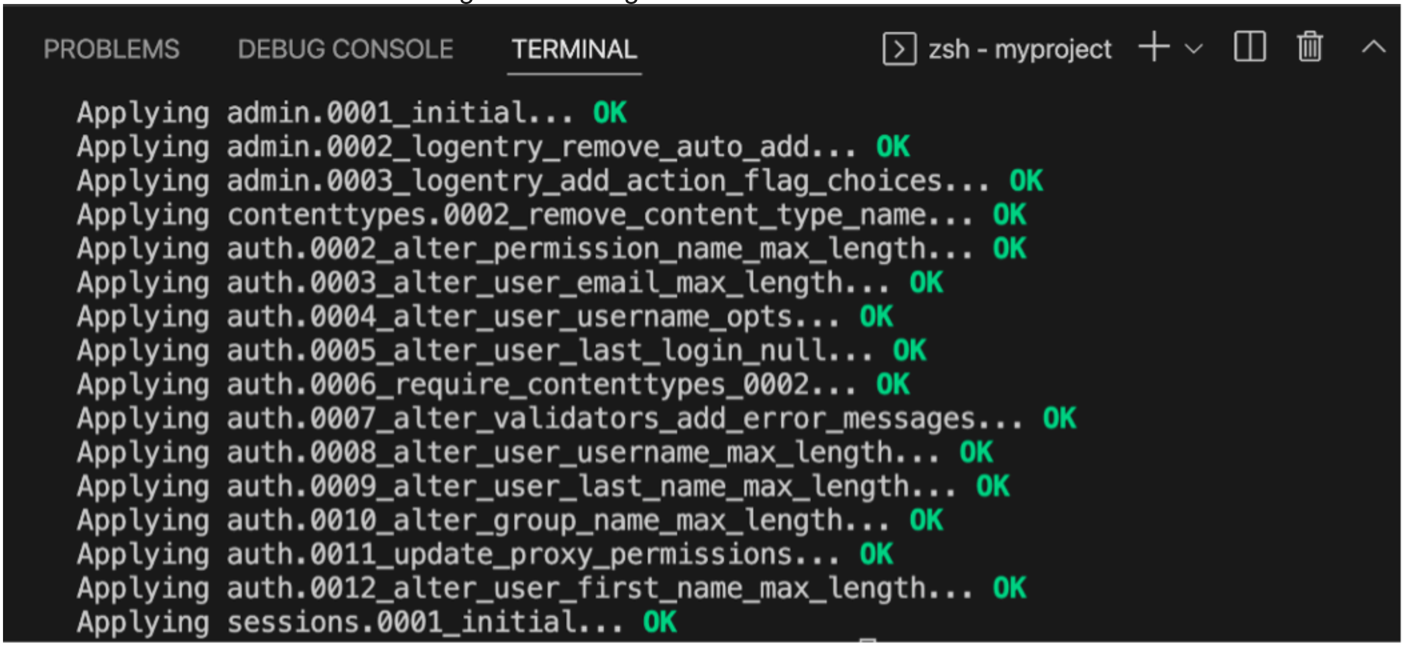
Step 5:
Make sure you are inside the **myproject** directory and run the development server.
Note: This generates the `db.sqlite3` database and is used by default in Django.

Step 6:
While you're still in the terminal, run the command to make migrations.
Tip: The output must look similar to:



```
 DjangoCourse$ python3 manage.py makemigrations
Migrations for 'myapp':
  myapp/migrations/0001_initial.py
    - Create model Drinks
 DjangoCourse$ 
```

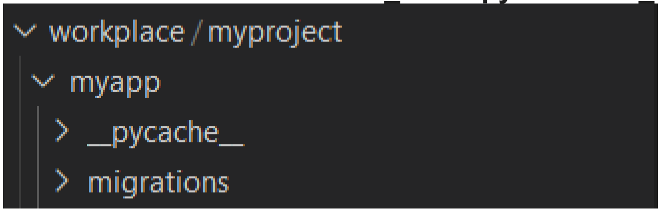
Step 7:
Run the command to perform the migration.
Tip: Check the information displayed in the command line to make sure that the migrations ran successfully without error. It should look something like the image below:



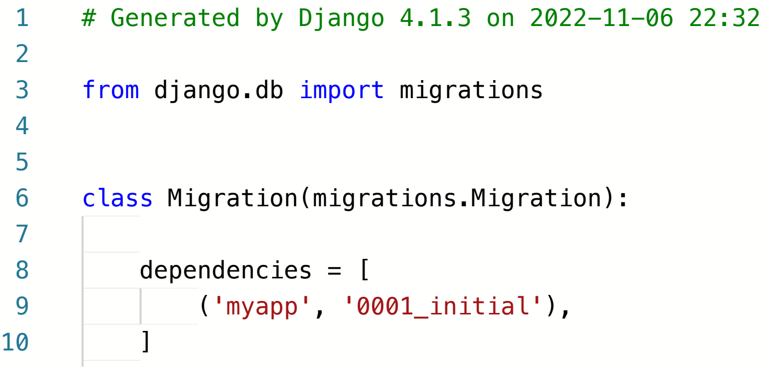
```
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

Step 8:
Once you are sure that the migrations ran successfully, return to the **models.py** file and modify the attribute name from **`drink`** to **`drink_name`**. Save the **models.py** file.
Step 9:
Run both the commands required to perform migrations one more time.
Note: A confirmation prompt will appear to confirm the changes. Type **`y`** to confirm the changes.

Step 10:
In the left explorer panel in VS Code, search for the **migrations** folder inside the **myapp** directory. Expand the folder and locate the files **0001_initial.py** and **0002_rename...py**. Open the files and explore their contents.



Notice that these files contain a list of operations and dependencies specific to the migration performed.

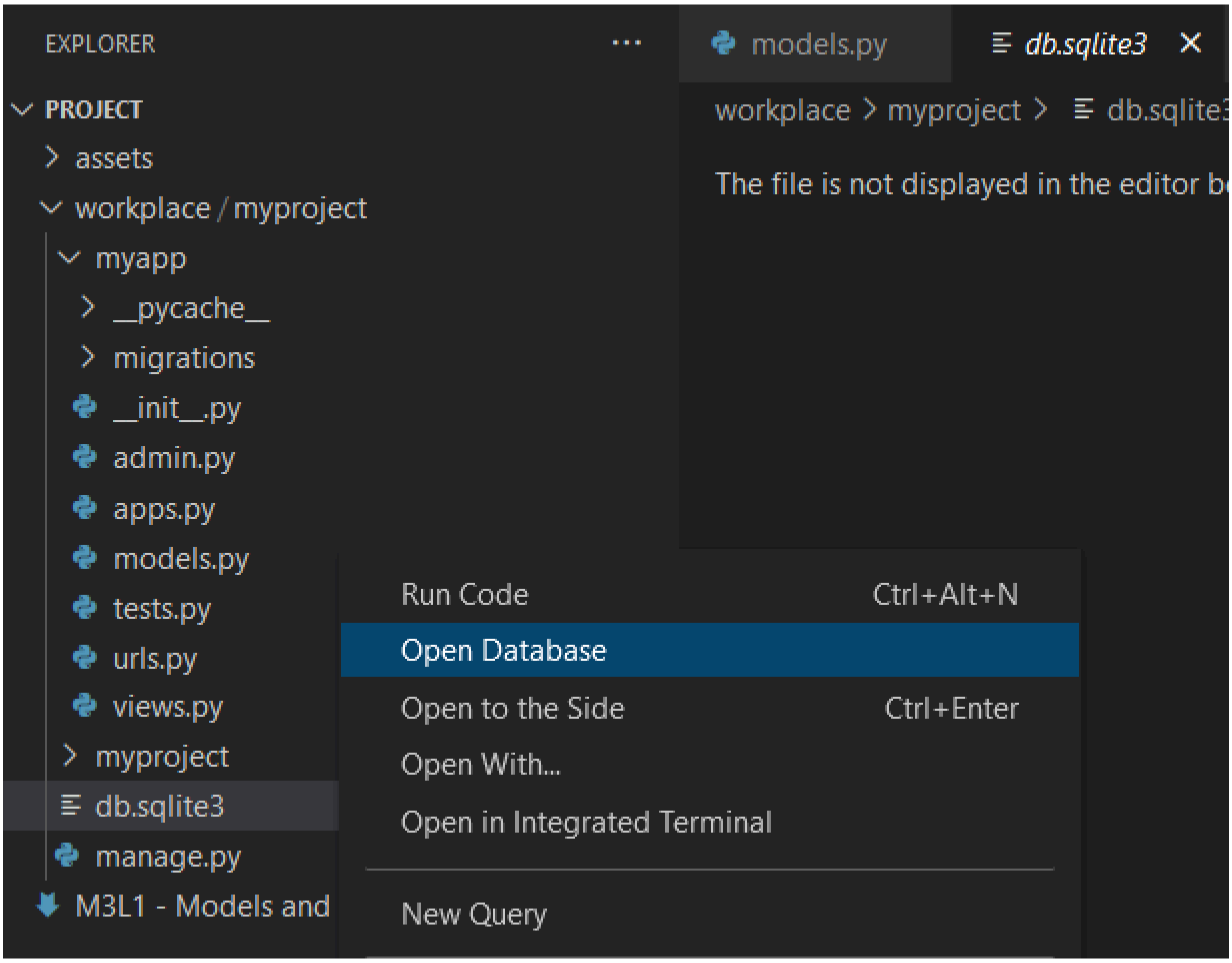


```
# Generated by Django 4.1.3 on 2022-11-06 22:32

from django.db import migrations


class Migration(migrations.Migration):

    dependencies = [
        ('myapp', '0001_initial'),
    ]

    operations = [
        migrations.RenameField(
            model_name='drinks',
            old_name='drink',
            new_name='drink_name',
        ),
    ]

```

Step 11:
Run the command to show the migration changes. The output should look similar to the following:

```
admin
 [X] 0001_initial
 [X] 0002_logentry_remove_auto_add
 [X] 0003_logentry_add_action_flag_choices
auth
 [X] 0001_initial
 [X] 0002_alter_permission_name_max_length
 [X] 0003_alter_user_email_max_length
 [X] 0004_alter_user_username_opts
 [X] 0005_alter_user_last_login_null
 [X] 0006_require_contenttypes_0002
 [X] 0007_alter_validators_add_error_messages
 [X] 0008_alter_user_username_max_length
 [X] 0009_alter_user_last_name_max_length
 [X] 0010_alter_group_name_max_length
 [X] 0011_update_proxy_permissions
 [X] 0012_alter_user_first_name_max_length
contenttypes
 [X] 0001_initial
 [X] 0002_remove_content_type_name
myapp
 [X] 0001_initial
 [X] 0002_rename_drink_drinks_drink_name
sessions
 [X] 0001_initial
```

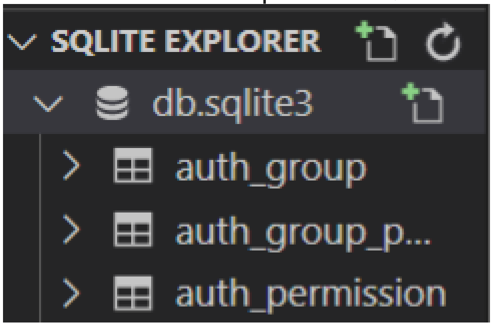
Step 12:
In the left explorer panel in VS Code, right-click on the database **`db.sqlite3`** that is generated and select **Open Database**.



The file is not displayed in the editor b

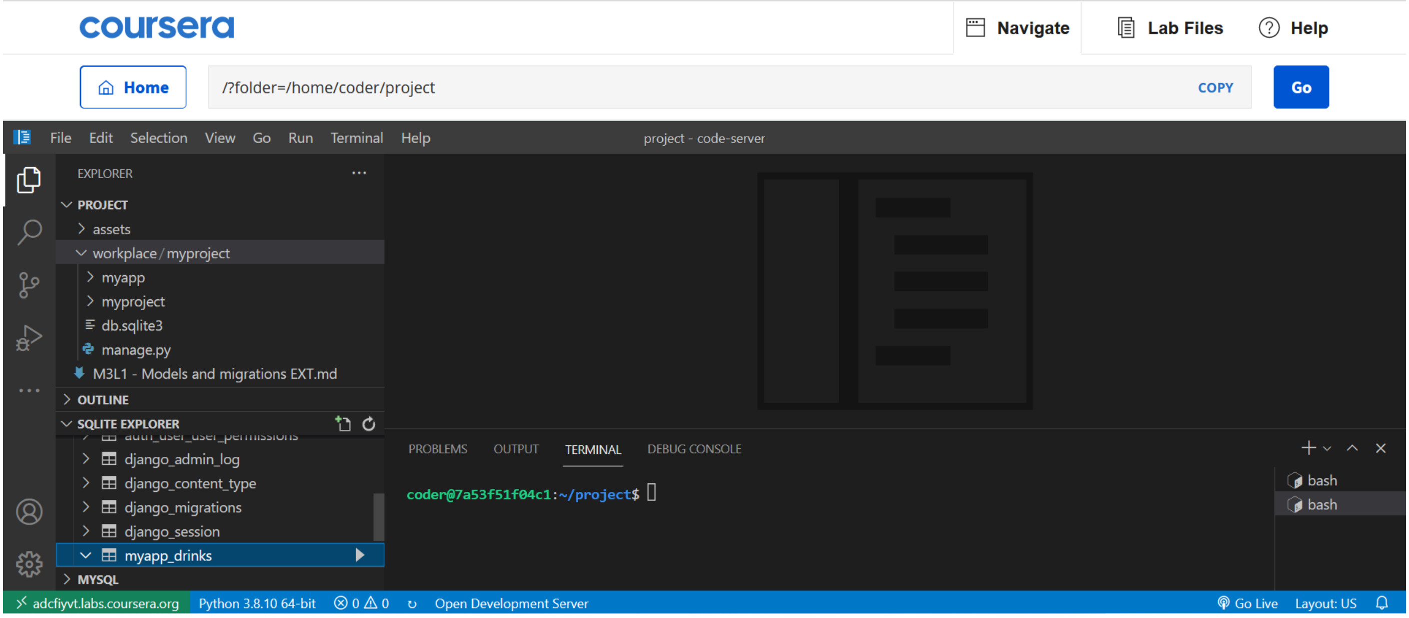
Step 13:

Notice that a menu open for **SQLITE EXPLORER** appears at the bottom of the left panel.



Now go to **SQLITE EXPLORER** and expand the option for **`db.sqlite3`**.

Scroll down to check if the **`myapp_drinks`** table is generated.



```
coder@7a53f51f04c1:~/project$
```

Note: You may not see any contents inside the **`myapp_drinks`** table as it is empty.

Concluding Thoughts

In this lab, you practiced creating a model (Drinks) and ran the migrations to create the associated database tables.