# 2.4 Models and Relationships

**General description:**

Within this template we explore creating a new one to many and one to one relationship models. Where one portion of our design has many different links that are all associated under one portion of the design (Portfolio has many Projects). Alongside this we focus on the one to one to relationship, where a student can have one Portfolio, and a project can be linked to only one portfolio.

## Creating Models and Relationships

* First, we should create our models. We will do this by setting up a class within the model’s portion labeled how it should appear. More about this can be seen within section 2.1A computer screen shot of a program

  Description automatically generated
* We will then create other models, such as the student and the portfolio portion. Whatever may be used in another model must be listed before this potion.A screen shot of a computer program

  Description automatically generated
  + Here a project and a student have a portfolio, so the portfolio must be listed first. The interpreter must know what a portfolio is before it is used further in the program.
  + Linking a portfolio to a student allows the student to have one portfolio.
  + Linking a project to a portfolio allows a portfolio to have many different projects.

## Linking the Admin Page to these Models

* To be able to use these changes within our project we must also link these to our website’s admin panel
  + Within the admin.py portion of our website we must register these tables withinA screen shot of a computer program

    Description automatically generated
  + From .models import \*
    - .models says we import from our source folder, the models portion
    - Import \* allows us to import everything within this file, if we had only one model, we could import specifically this model.

## Creating Migrations

* To make the changes official to our website we must migrate this information!
  + Within our terminal, and with our website offline we run the following command:
    - py manage.py makemigrations
      * This will allow our program to detect our needed changes and add them to a list to add to the website.
  + After we make the migrations, we have to push them as following:
    - Py manage.py migrate.
      * This will push our updates to both our SQL table and the website using the table!

## Issues Encountered

* A referenced model must be created first.
  + Solution: rearrange the models so that the referenced model is created first.

## Useful Resources

* [Django Documentation](https://docs.djangoproject.com/en/5.0/topics/db/models/)
  + The official documentation for Django models, containing more in-depth descriptions of how this data works!
* [W3 Schools – Django Models](https://www.w3schools.com/django/django_models.php)
  + Tutorial website explaining in detail how to create these models.
  + Has explanations and examples alongside each section!