**Models in Django:**

In modern Web applications, the arbitrary logic often involves interacting with a database. Behind the scenes, a *database-driven Web site* connects to a database server, retrieves some data out of it, and displays that data on a Web page.

Django is well suited for making database-driven Web sites, because it comes with easy yet powerful tools for performing database queries using Python

**Configuring the Database:**

**Settings.py**

# Database

#...

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),

}

}

* **ENGINE** tells Django which database engine to use. As we are using SQLite in the examples in this book, we will leave it to the default **django.db.backends.sqlite3.**
* **NAME** tells Django the name of your database. For example: **'NAME': 'mydb',**

Since we’re using SQLite, **startproject** created a full filesystem path to the database file for us.

what’s the difference between a *project*and an *app*? The difference is that of configuration vs. code:

* A project is an instance of a certain set of Django apps, plus the configuration for those apps. Technically, the only requirement of a project is that it supplies a settings file, which defines the database connection information, the list of installed apps, the **DIRS**, and so forth.
* An app is a portable set of Django functionality, usually including models and views that live together in a single Python package. For example, Django comes with a number of apps, such as the automatic admin interface. A key thing to note about these apps is that they’re portable and reusable across multiple projects.

If you’re using Django’s database layer (models), you must create a Django app. Models must live within apps. Thus, in order to start writing our models, we’ll need to create a new app.

Within the **mysite** project directory (this is the directory where your **manage.py**file is, **not** the **mysite**app directory), type this command to create a **books** app:

python manage.py startapp books

This command does not produce any output, but it does create a **books** directory within the **mysite**directory. Let’s look at the contents of that directory:

books/

/migrations

\_\_init\_\_.py

admin.py

models.py

tests.py

views.py

These files will contain the models and views for this app. Have a look at **models.py** and **views.py** in your favorite text editor. Both files are empty, except for comments and an import in **models.py**. This is the blank slate for your Django app.

**Defining Django Models in Python:**