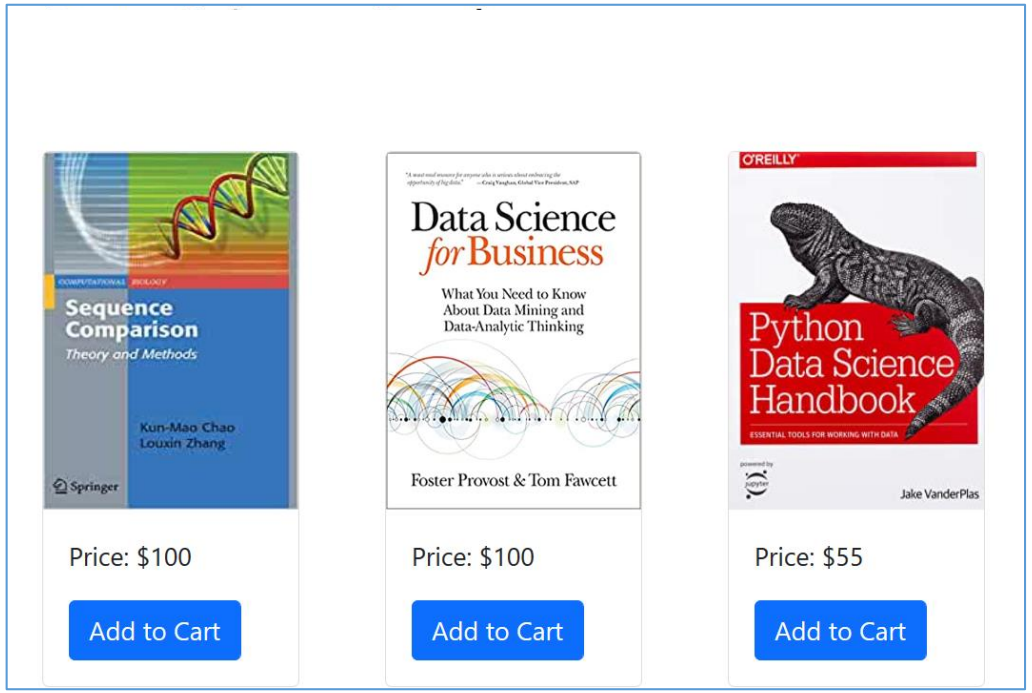
DSA5101
Introduction to Big Data for Industry

Django for Webpage design

LX Zhang
Department of Mathematics
National University of Singapore

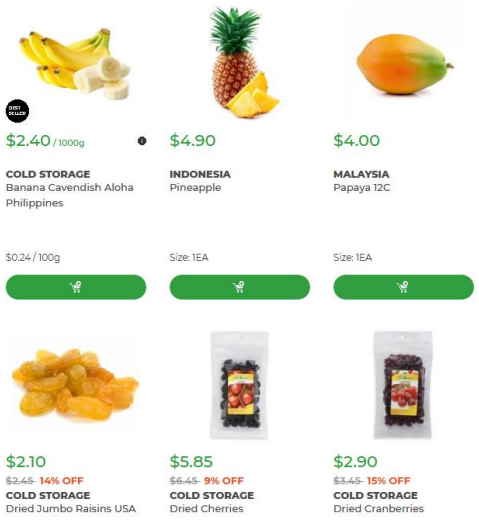
Programming for Web application

- Online shopping
- Online data collection and survey



Date	Title
10 Oct 2020	2 New Cases of Locally Transmitted COVID-19 Infection - Ministry of Health (MOH)
10 Oct 2020	2 New Cases of Locally Transmitted COVID-19 Infection - Ministry of Health (MOH)
9 Oct 2020	1 New Case of Locally Transmitted COVID-19 Infection - Ministry of Health (MOH)
9 Oct 2020	1 New Case of Locally Transmitted COVID-19 Infection - Ministry of Health (MOH)
8 Oct 2020	4 New Cases of Locally Transmitted COVID-19 Infection - Ministry of Health (MOH)
8 Oct 2020	4 New Cases of Locally Transmitted COVID-19 Infection - Ministry of Health (MOH)
7 Oct 2020	4 New Cases of Locally Transmitted COVID-19 Infection - Ministry of Health (MOH)

https://www.gov.sg/article/covid-19-updates-and-announcements

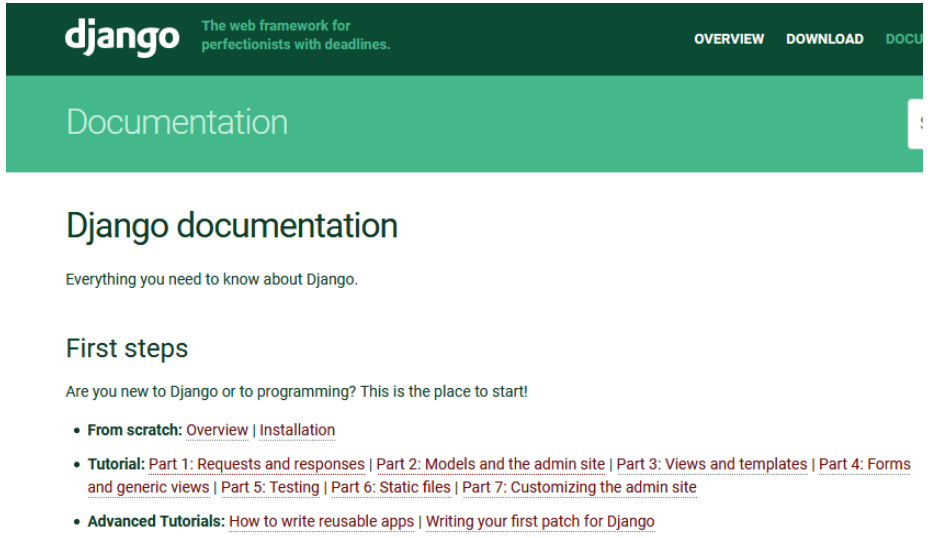


https://coldstorage.com.sg/search?q=fruits&fdept=fruits-vegetables

Django for Web development

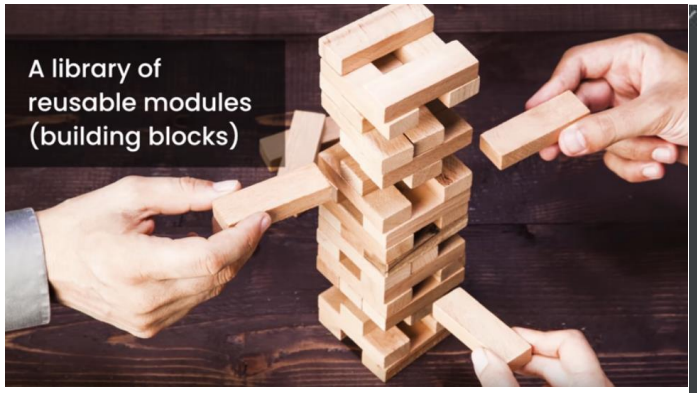
- It has been used to supports popular websites
  - -- Washington Post
  - -- New York Times
  - -- Youtube
  - -- Instagram

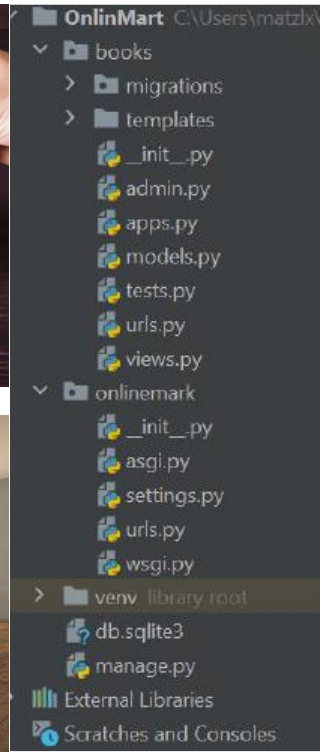
- It contains a lot of reusable libraries (modules)

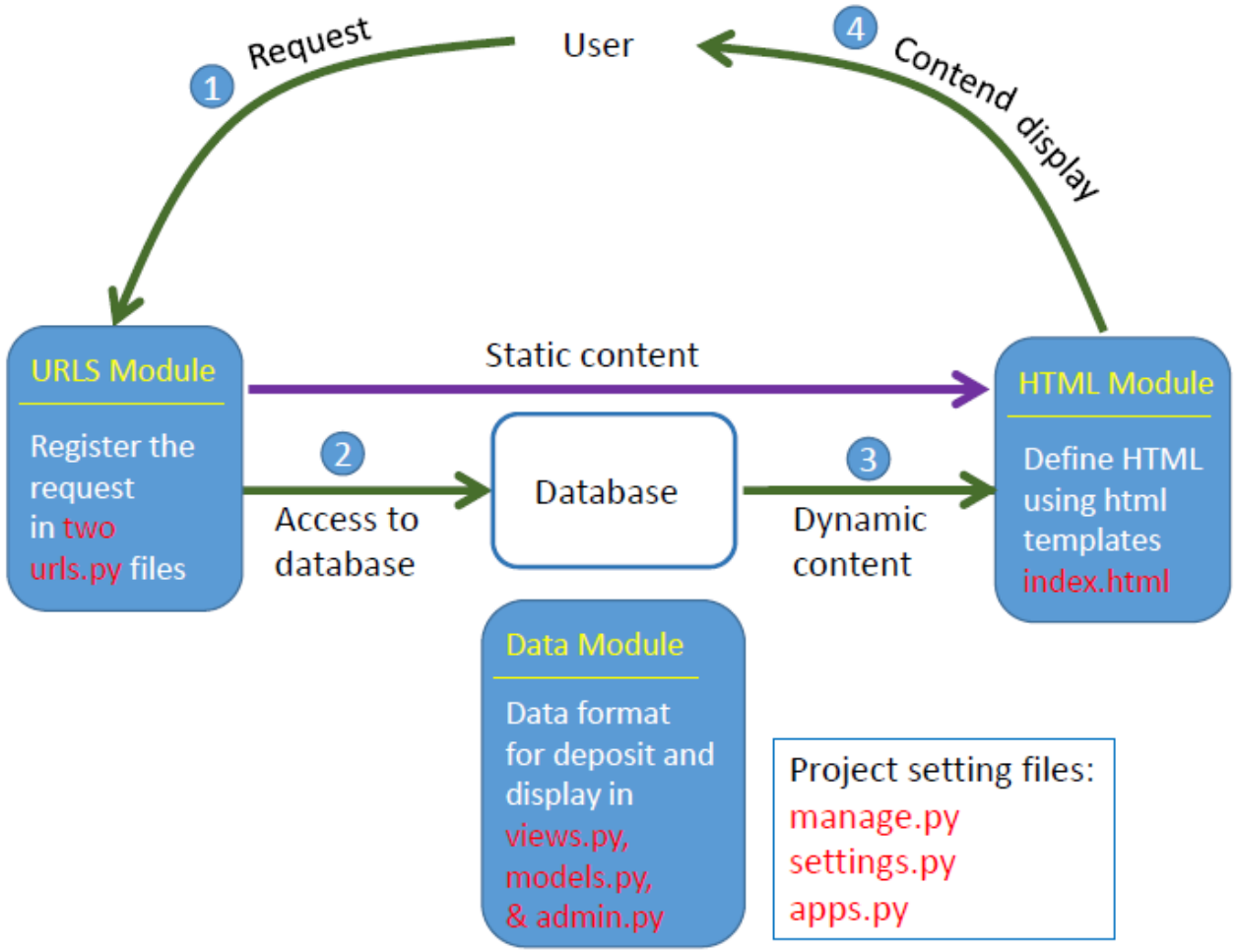


https://docs.djangoproject.com/

https://www.youtube.com/watch?v=_uQrJ0TkZlc
Start from 5h00m00s
https://www.youtube.com/watch?v=F5mRW0jo-U4


A library of
reusable modules
(building blocks)


OnlinMart C:\Users\matzlx\
books
migrations
templates
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py
onlinemark
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
venv library root
db.sqlite3
manage.py
External Libraries
Scratches and Consoles



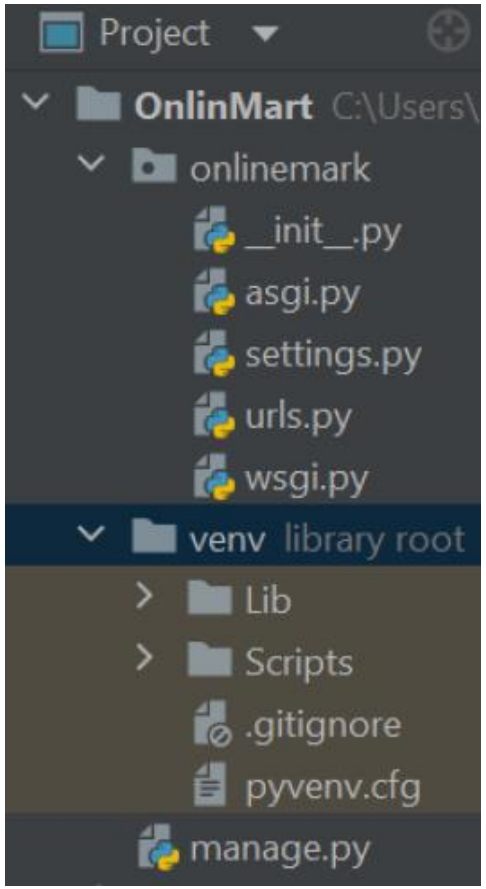
Front (user) End

Back (Server) End

I. Install Django on PyCharm and set a webserver

- Numbered lines with grey background are command line for creating a project
- For our project, no need to change asgi.py and wsgi.py

1. pip install django
2. django-admin startproject OnlineMart .

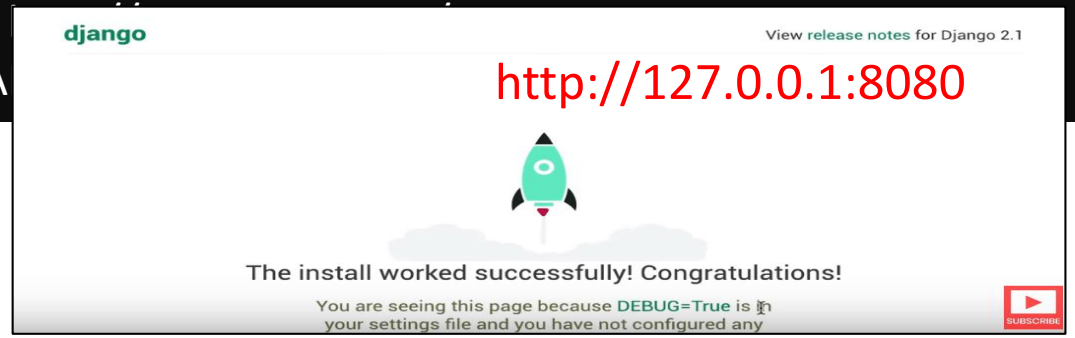


It creates a package folder "OnlineMart", containing:

__init__.py	It indicates that the current directory is a package. We can export different modules from this package
settings.py	Parameters setting for the project, e.g. registering applets
urls.py	The URL declarations for this Django project
asgi.py	Asgi: Asynchronous Server Gateway Interface
wsgi.py	Wsgi: webserver gateway interface. Both files provide interface between applications (built with Django) and the web-server.
manage.py	We need to run this program whenever each applet is added.

③ python manage.py runserver

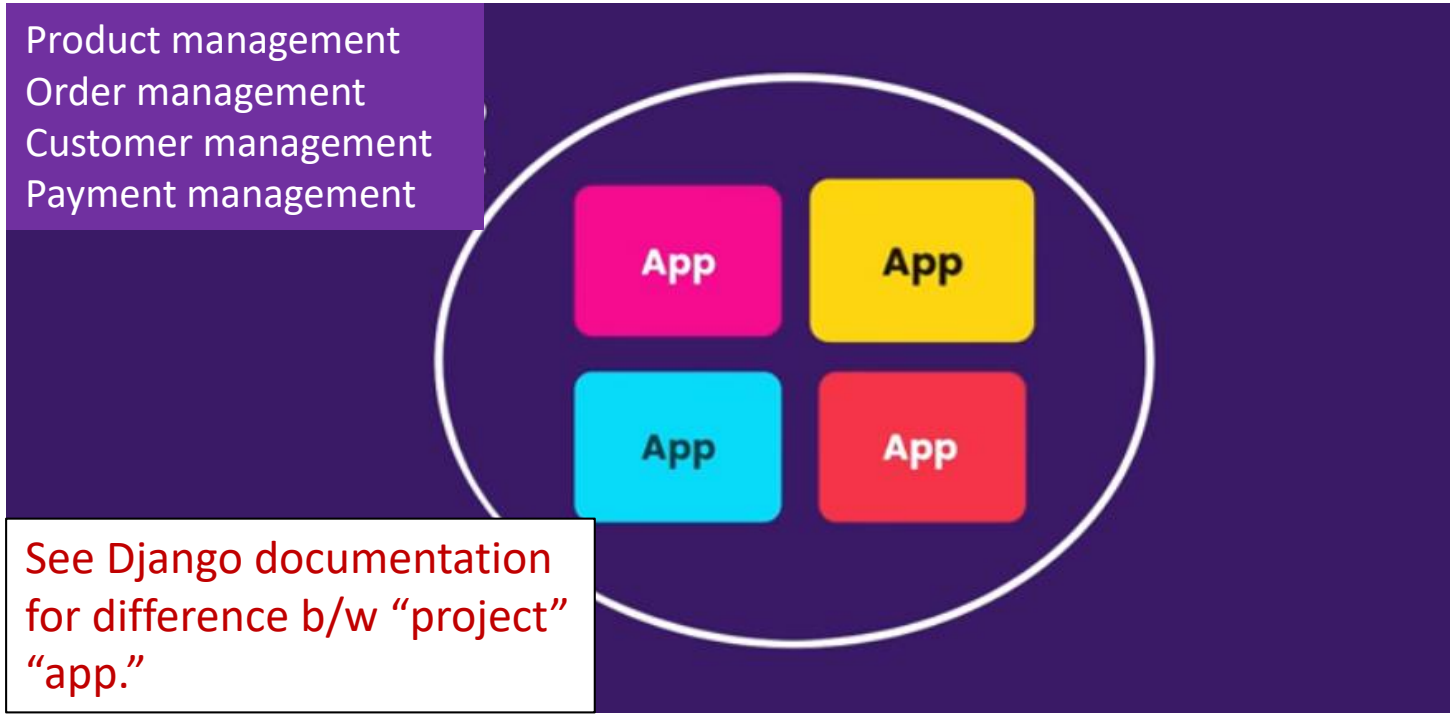
```
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
October 10, 2021 - 09:01:04
Django version 3.2.8, using settings 'onlinemark.settings'
Starting development server at
Quit the server with CTRL-BREA
```



Advanced How to bind the web server to a non-local address so someone can remotely view the development server?

Warning The sever must be on for webpage testing in the following slides

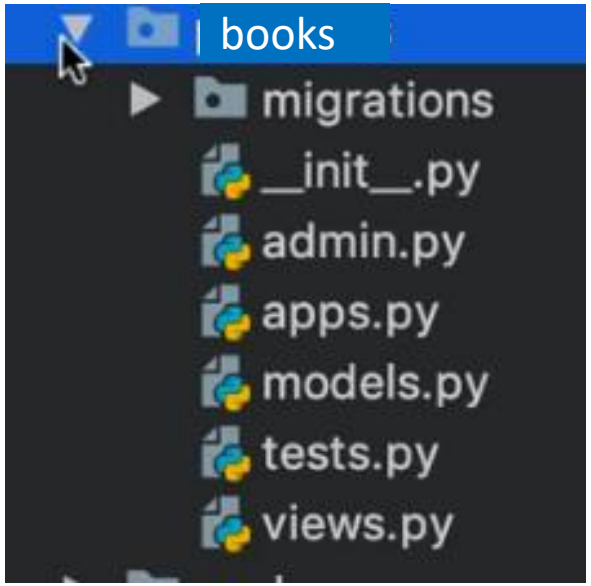
II. Create an app that displays static content in a webpage



See Django documentation for difference b/w “project” “app.”

- Use + to open another terminal window and start an app called “books”

4 python manage.py startapp books



For our project, no need to modify apps.py and test.py

__init__.py An empty file indicating that “books” is a package;

admin.py Register products in admin.py file.
It provide an interface with admin applet

models.py It defines data format, providing interface between database, admin and display modules

views.py It specifies how to responses to each request from users.

apps.py It contains a configuration class for the applet “books”

test.py It is for automatic test purposes.

In views.py, define a response function called “index”

```
from django.http import HttpResponse
from django.shortcuts import render

# Create your views here.

def index(request):
    return HttpResponse('My Book Store')
```

A view is a “type” of Web page in your Django application that generally serves a specific function and has a specific template.
The server will choose a view by decoding the URL (to be precise, the part of the URL after the domain name).

URL: http://127.0.0.1:8000/books/

Domain name

Questions: What is the part of the following URLs that is mapped to a view?

http://127.0.0.1:8000/books/offer

https://www.gov.sg/article/covid-19-updates-and-announcements

https://coldstorage.com.sg/search?q=fruits&fdept=fruits-vegetables

Each view is responsible for doing one of two things:

- Returning an HttpResponse object containing the content,
  -- a webpage
  -- a pdf (xml, zip)

  or
- Raising an exception such as Http404.

A view can read query results from a database and use a webpage template if necessary

http://127.0.0.1:8000/books/

urls.py for applet “books”

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('online_booksmart/', include('books.urls'))
]
```

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index),
]
```

urls.py for project

Caution There are two or more “urls.py”, one for project, others for apps.

Advanced Regular expression used in the first argument, see URL dispatcher section Section 3.2 about urls in the official documentation.

https://www.gov.sg/article/covid-19-updates-and-announcements

When the user clicks http://127.0.0.1:8000/books/ , the server response by:

- Find match “books/” in the file “onlinemart/urls.py”;
- Further match the rest (which is the empty in this case) in the file ./books/urls.py, which indicate the response is view.index(), which is HttpResponse(“My Book Store”) in our case.

Therefore, we the following when accessing the webpage.

III. Create a server-end database called “Books” that can be updated through the admin webpage.

To this, the program needs to implement:

-- to load a template,

-- access the database in SQLite to fill context,

-- and return an webpage

Advanced: How to connect Django with other database platforms, like SQL?

III.1 In the directory “books/models.py”, define a data class “Book”

```
from django.db import models

# Create your models here.
class Book(models.Model):
    name = models.CharField(max_length=255)
    price = models.IntegerField()
    image_url = models.CharField(max_length=2083)
```

III.2 In the project directory “.”, add a line for the configuration of the app “books” in the file “settings.py” as follows.

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'books.apps.BooksConfig'
]
```

5 python manage.py makemigrations

```
Migrations for 'books':
  books\migrations\0001_initial.py
    - Create model book
PS C:\Users\matzlx\PycharmProjects\OnlinMart>
```

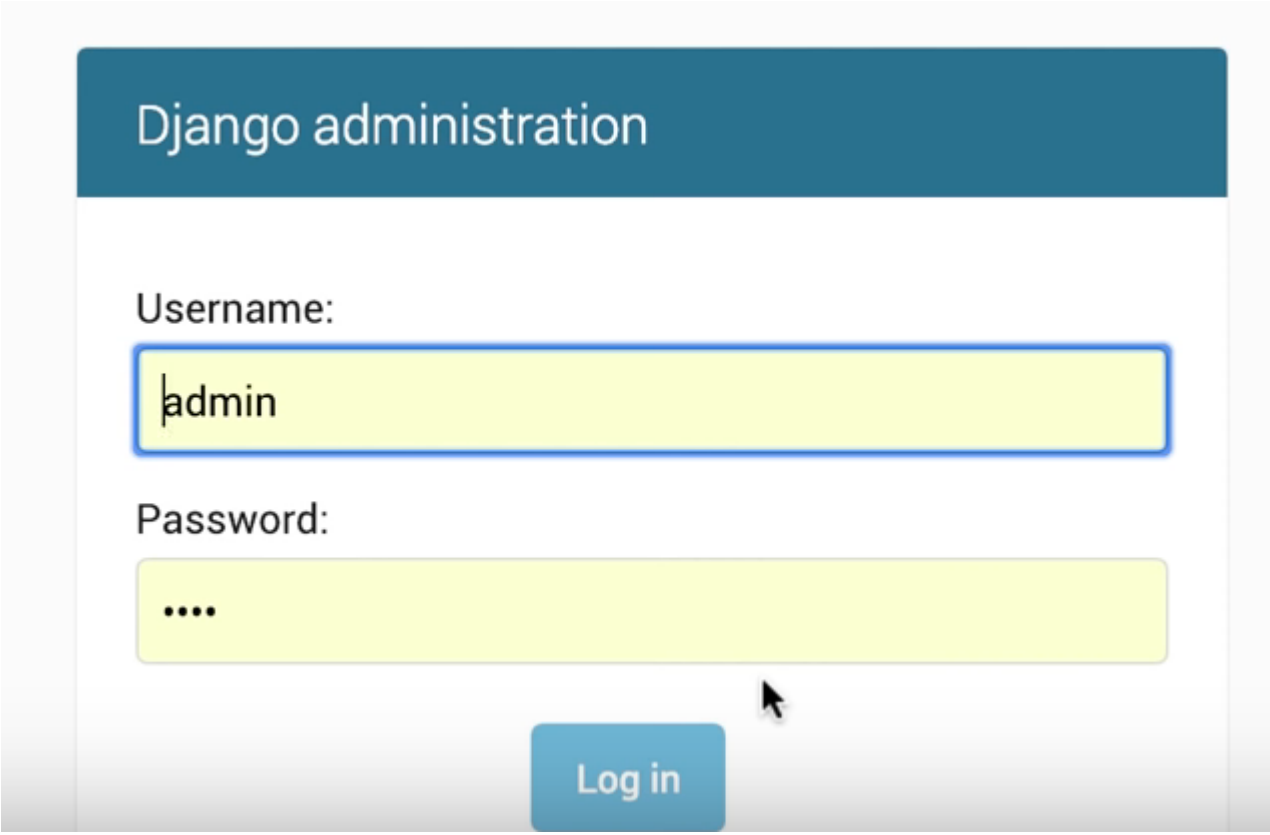
6 python manage.py migrate

```
manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, books, contenttypes,
sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
```

Now, a database of books has been established.

One can repeat these steps to create every database if there are more than one applets.

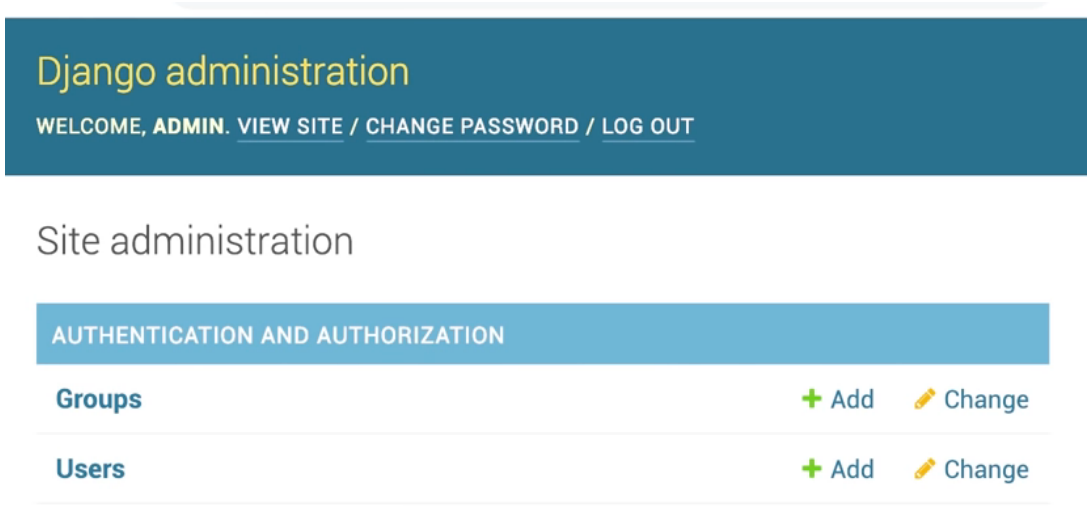
http://127.0.0.1:8000/admin



Run the following command to add an administrator

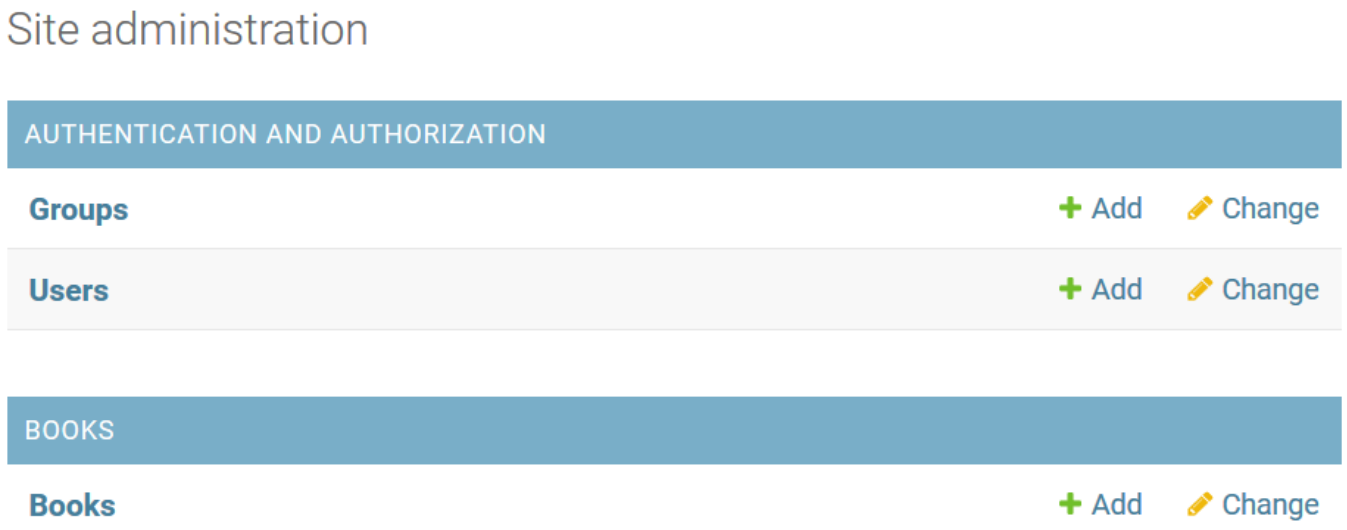
7
```
python3 manage.py createsuperuser
```

Click http://127.0.0.1:8000/admin/, you will see the following page



III.3 In the directory “.”, add a line to register the app book in “admin.py” as follows.

```
from django.contrib import admin
from .models import book

# Register your models here.

admin.site.register(book)
```



Add book

Name:

Price:

Image url:

Save and add another | Save and continue editing | SAVE

Image address taken from website

III.4 In the directory “.”, add a line to register the app “books” in “admin.py” as follows.

```
from django.contrib import admin
from .models import book

# Register your models here.
class BookAdmin(admin.ModelAdmin):
    list_display=('name', 'price')
```

Action: --------- Go 0 of 2 selected

	NAME	PRICE
☐	Data Science for Busineess	99
☐	Sequence Comparison	100

2 books

DB Browser for SQLite - C:\Users\matzlx\PycharmProjects\OnlinMart\db.sqlite3

File Edit View Tools Help

New Database | Open Database | Write Changes | Revert Changes | Open Project | Save Project | Attach Database | Close Database

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Create Table | Create Index | Print

Name	Type	Schema
Tables (12)		
auth_group		CREATE TABLE "auth_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(150) NOT NULL UNIQUE)
auth_group_permissions		CREATE TABLE "auth_group_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "group_id" integer NOT NULL REFEREN
auth_permission		CREATE TABLE "auth_permission" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "content_type_id" integer NOT NULL REFEREN
auth_user		CREATE TABLE "auth_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "password" varchar(128) NOT NULL, "last_login" dat
auth_user_groups		CREATE TABLE "auth_user_groups" ("id" int
auth_user_user_permissions		CREATE TABLE "auth_user_user_permissior
books_book		CREATE TABLE "books_book" ("id" integer N
django_admin_log		CREATE TABLE "django_admin_log" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "action_time" datetime NOT NULL, "object_
django_content_type		CREATE TABLE "django_content_type" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app_label" varchar(100) NOT NULL, "mo
django_migrations		CREATE TABLE "django_migrations" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app" varchar(255) NOT NULL, "name" varc
django_session		CREATE TABLE "django_session" ("session_key" varchar(40) NOT NULL PRIMARY KEY, "session_data" text NOT NULL, "expire_date" datet
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
Indices (15)		
auth_group_permissions_group_id_b1...		CREATE INDEX "auth_group_permissions_group_id_b120cbf9" ON "auth_group_permissions" ("group_id")
auth_group_permissions_group_id_pe...		CREATE UNIQUE INDEX "auth_group_permissions_group_id_permission_id_0cd325b0_uniq" ON "auth_group_permissions" ("group_id", "p
auth_group_permissions_permission_i...		CREATE INDEX "auth_group_permissions_permission_id_84c5c92e" ON "auth_group_permissions" ("permission_id")
auth_permission_content_type_id_2f4...		CREATE INDEX "auth_permission_content_type_id_2f476e4b" ON "auth_permission" ("content_type_id")
auth_permission_content_type_id_cod...		CREATE UNIQUE INDEX "auth_permission_content_type_id_codename_01ab375a_uniq" ON "auth_permission" ("content_type_id", "codena
auth_user_groups_group_id_97559544		CREATE INDEX "auth_user_groups_group_id_97559544" ON "auth_user_groups" ("group_id")
auth_user_groups_user_id_6a12ed8b		CREATE INDEX "auth_user_groups_user_id_6a12ed8b" ON "auth_user_groups" ("user_id")
auth_user_groups_user_id_group_id_...		CREATE UNIQUE INDEX "auth_user_groups_user_id_group_id_94350c0c_uniq" ON "auth_user_groups" ("user_id", "group_id")
auth_user_user_permissions_permissi...		CREATE INDEX "auth_user_user_permissions_permission_id_1fbb5f2c" ON "auth_user_user_permissions" ("permission_id")
auth_user_user_permissions_user_id_...		CREATE INDEX "auth_user_user_permissions_user_id_a95ead1b" ON "auth_user_user_permissions" ("user_id")
auth_user_user_permissions_user_id_...		CREATE UNIQUE INDEX "auth_user_user_permissions_user_id_permission_id_14a6b632_uniq" ON "auth_user_user_permissions" ("user_
django_admin_log_content_type_id_c...		CREATE INDEX "django_admin_log_content_type_id_c4bce8eb" ON "django_admin_log" ("content_type_id")
django_admin_log_user_id_c564eba6		CREATE INDEX "django_admin_log_user_id_c564eba6" ON "django_admin_log" ("user_id")

CREATE TABLE "auth_permission" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "content_type_id" integer NOT NULL REFERENCES "django_content_type" ("id") DEFERRABLE INITIALLY DEFERRED, "codename" varchar(100) NOT NULL, "name" varchar(255) NOT NULL)

Edit Database Cell

Mode: Text

1 1

Type of data currently in cell: Text / Numeric
1 character(s)

Apply

Remote

Identity Select an identity to connect

DBHub.io | Local | Current Database

Name	Last modifie

SQL Log | Plot | DB Schema | Remote

UTF-8

53

54

55

Slide 53 of 66 English (Singapore)

Notes Comments

Type here to search

29°C Mostly cloudy ENG 7:01 PM 10/19/2021

DB Browser for SQLite - C:\Users\matzlx\PycharmProjects\OnlinMart\db.sqlite3

File Edit View Tools Help

New Database | Open Database | Write Changes | Revert Changes | Open Project | Save Project

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: books_book

Filter in any column

	id	name	price	image_url
	Filter	Filter	Filter	Filter
1	1	Sequence Comparison	100	https://images-na.ssl-images-amazon.com/...
2	2	Data Science for Busineess	100	https://images-na.ssl-images-amazon.com/...
3	3	Python Data Sciecne Handbook	55	https://m.media-amazon.com/images/I/...

IV. Create a template to display dynamic content

IV.1. Redefine the response function in “views.py”

```
from django.http import HttpResponse
from django.shortcuts import render

# Create your views here.

def index(request):
    return HttpResponse('My Book Store')
```

The first “views.py”

```
from django.shortcuts import render
from django.http import HttpResponse
from .models import Book

# Create your views here.

def index(request):
    book_all = Book.objects.all()
    return render(request, 'index.html', {'books': book_all})
```

After refining

Advanced: What functions are available in product.objects class?

IV. Create a template to display dynamic content

IV. 2. Create a subdirectory called “templates” under “/book” and a file called “index.html” that contains the following contents.

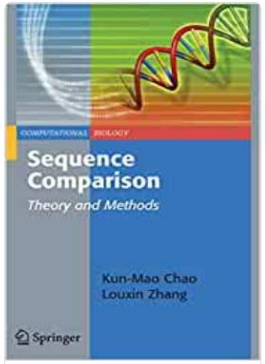
Remark: {% %}
markers for page templates
{{ }} surrounding variables

```
<body>
<h1>Books</h1>
<ul>
   {% for book in books %}
   <li>- {{ book.name }}:  ${{ book.price }} </li>
   {% endfor %}
</ul>
</body>
```

Returned webpage

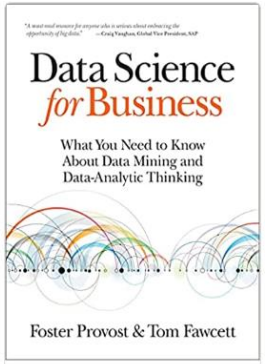
V. How to create the following card format?

Data Science Books



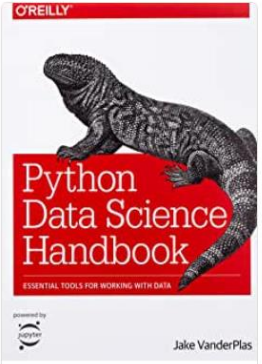
Price: $100

Add to Cart



Price: $100

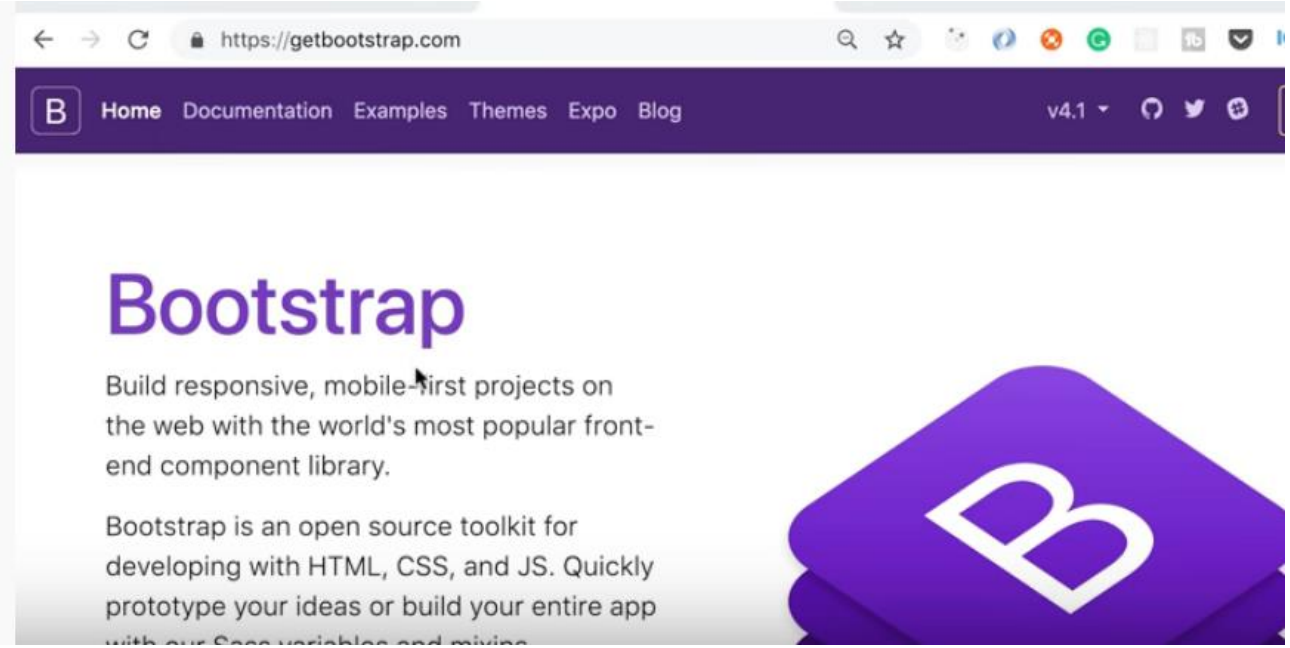
Add to Cart



Price: $55

Add to Cart

https://getbootstrap.com



V.1 Go to the website to download the following starter template
copy it into a file called “base.html” and modify it by adding a content block

Starter template

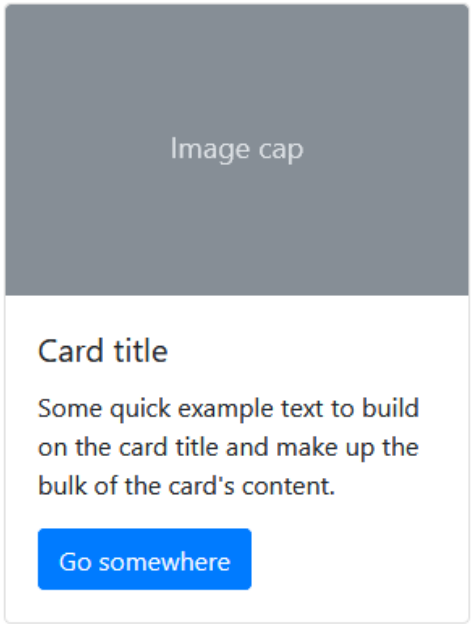
Be sure to have your pages set up with the latest design and development standards. That me
HTML5 doctype and including a viewport meta tag for proper responsive behaviors. Put it all t
your pages should look like this:

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-t

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.3/css/bootstrap.min.css" integr

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: jQuery and Bootstrap Bundle (includes Popper) -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4(
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.3/js/bootstrap.bundle.min.js" integrity="sha384

    <!-- Option 2: jQuery, Popper.js, and Bootstrap JS
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4(
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTG
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.3/js/bootstrap.min.js" integrity="sha384-w1Q4or
    -->
  </body>
</html>
```

delete

```
<body>
 <!-- As a link -->
 <nav class="navbar navbar-light bg-light">
   <a class="navbar-brand" href="#">Product Bar</a>
</nav>
 <div class = "container">
   {% block content %}
   {% endblock %}
 </div>
```

V.2 Download the following “card” template copy it into a file called “index.html” and modify it by adding a content block, shown in the next slide



Copy

```
<div class="card" style="width: 18rem;">
  <img src="..." class="card-img-top" alt="...">
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's cc
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```

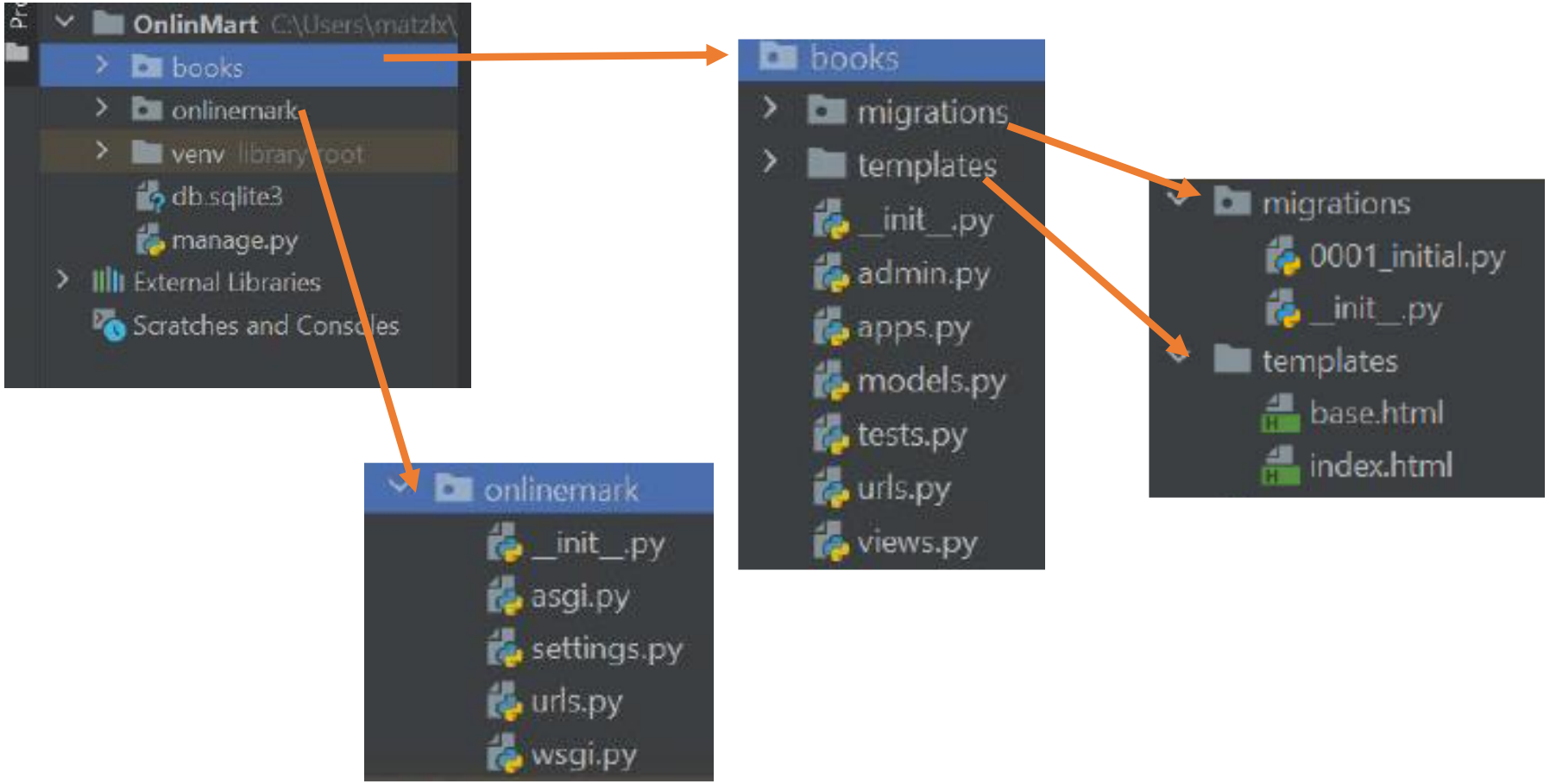
“index.html” after modification

```
{% extends 'base.html' %}
{% block content %}
<h1> Data Science Books</h1>
<br>
<br>
<div class="row">
   {% for book in books %}
   <div class="col">
      <div class="card" style="width: 10rem;">
         <img src="{{ book.image_url }}" class="card-img-top" alt="no available"
width="200"
    height="250">
         <div class="card-body">
            <!-- <h5 class="card-title">{{ book.name }}</h5> -->
            <p class="card-text">Price: ${{ book.price }}</p>
            <a href="#" class="btn btn-primary">Add to Cart</a>
         </div>
      </div>
   </div>
   {% endfor %}
</div>
{% endblock %}
```

- Python commands are inside {% %}
- Python variables are inside {{ }}

```
<div class="card" style="width: 18rem;">
  <img src="..." class="card-img-top" alt="...">
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```

Done!


OnlinMart C:\Users\matzlx\
books
onlinemark
venv library root
db.sqlite3
manage.py
External Libraries
Scratches and Consoles
books
migrations
templates
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py
migrations
0001_initial.py
__init__.py
templates
base.html
index.html
onlinemark
__init__.py
asgi.py
settings.py
urls.py
wsgi.py

Summary

- Django is a popular framework for designing webpages that are widely used in online shopping, and for survey
- Deploy the program package to another server machine is relatively easy.
- Doing a few times worth more than watching a thousand times