# **PART B – Create Django Web Application for Classic Books**

# Django – ENVIRONMENT SETUP prior to Books Application **project-books**

Setting up a virtual environment allows you to customize the environment for each individual project you work on – for example, if Project#1 requires Python with the **django** package, then you can keep that environment separate from your Project#2 environment that requires **numpy, pandas**, etc. This is also important if you are “packaging” your project because you only want to include packages needed for your specific project.

### **SETUP - Create virtual environment: books\_env**

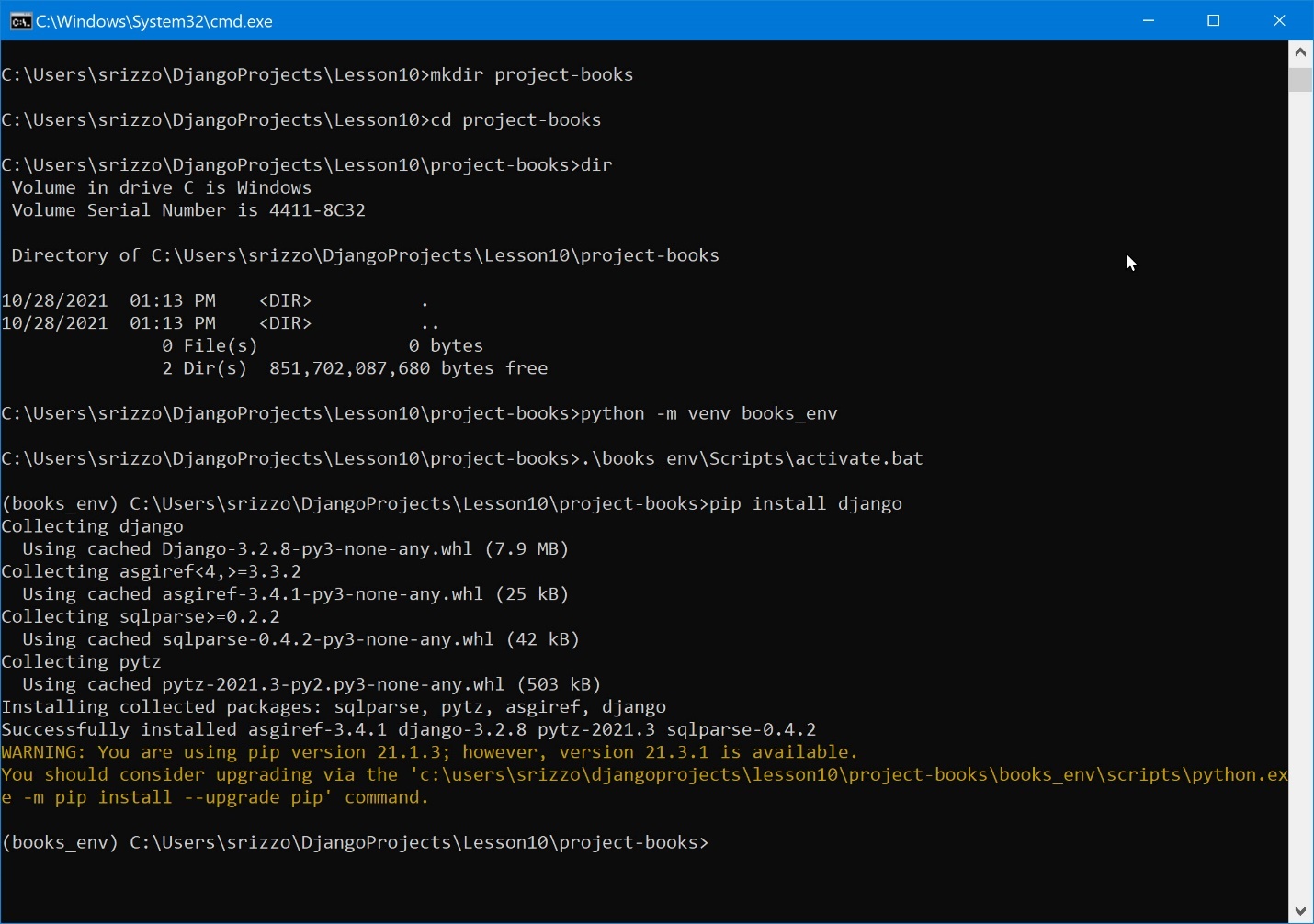
* Execute all the commands shown below to create your **books\_env** virtual environment
* Install Django in your virtual environment: **pip install django**

**IMPORTANT**: My screen captures were made on a Windows machine.

**MAC**: If you are on a Mac, the command to activate the **books\_env** virtual environment will be: …/project-books>**books\_env/bin/activate**

…/project-books>**books\_env/bin/deactivate** 🡨 deactivates virtual environment

**Note**: After you activate the virtual environment **(books\_env)** will prefix your command prompt.



### **Windows PowerShell users …**

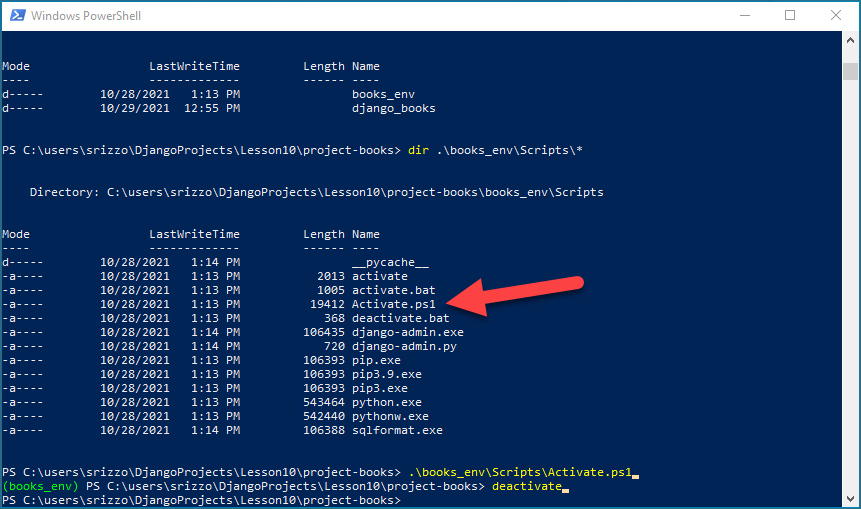
If you choose to use Windows PowerShell, be aware that the commands to use to activate/deactivate your virtual environment are different. Here’s

There is a **.ps1** version of the Activate command specifically for PowerShell.

To **activate** you enter (I always prefix the current directory with .\ out of habit 😊):

…/project-books>**.\books\_env\Scripts\Activate.ps1**

And, when you are ready to deactivate, you simply enter **deactivate** on the command line.



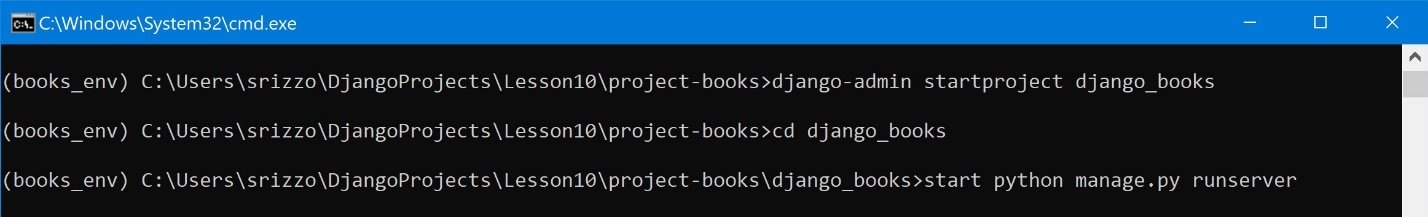
## Create Django Project – **django\_books**

CD into the **project-books** directory

* **cd project-books**

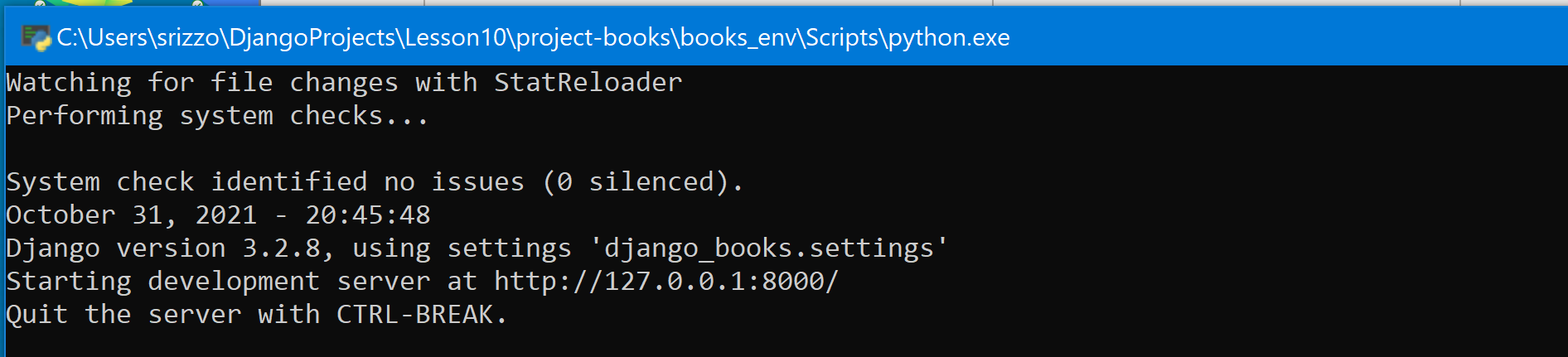
### Execute the command to start the Django server (**this will open a separate command window**)

* **start python manage.py runserver** 🡨 ‘start’ is a Windows-only command
* **python manage.py runserver** 🡨 if running on MacOS

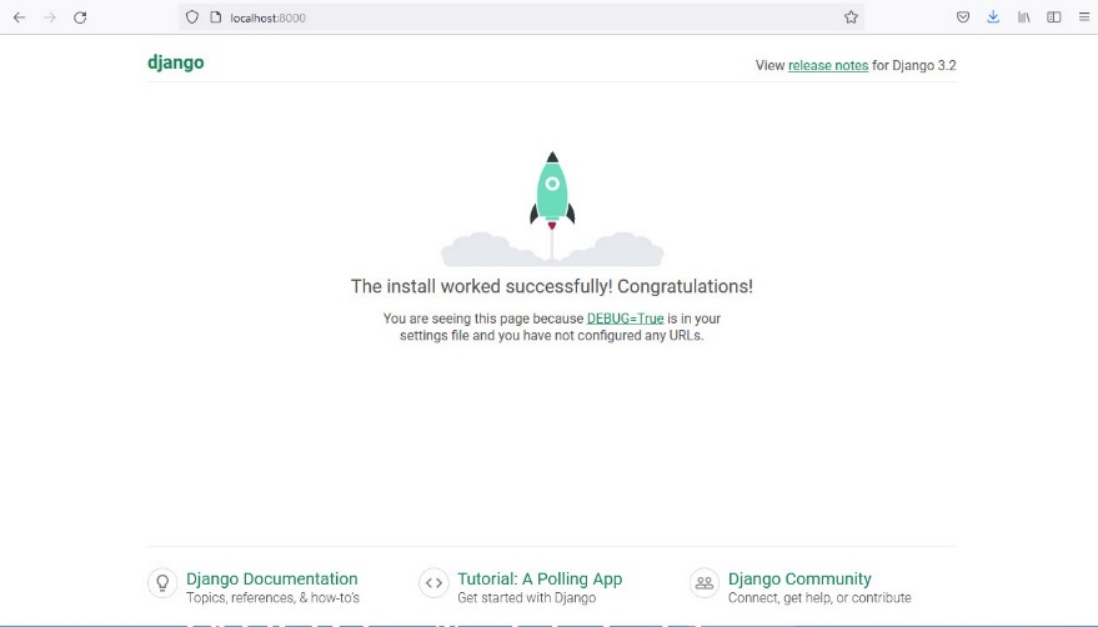
****

## **SCREEN CAPTURE # 1**

Command window as a result of the **runserver** command



Open a browser and go to <http://localhost:8000>. You should see the default Django webpage



## Create Django Application – **books**

### Execute the command to create your **books** application

* **python manage.py startapp books**



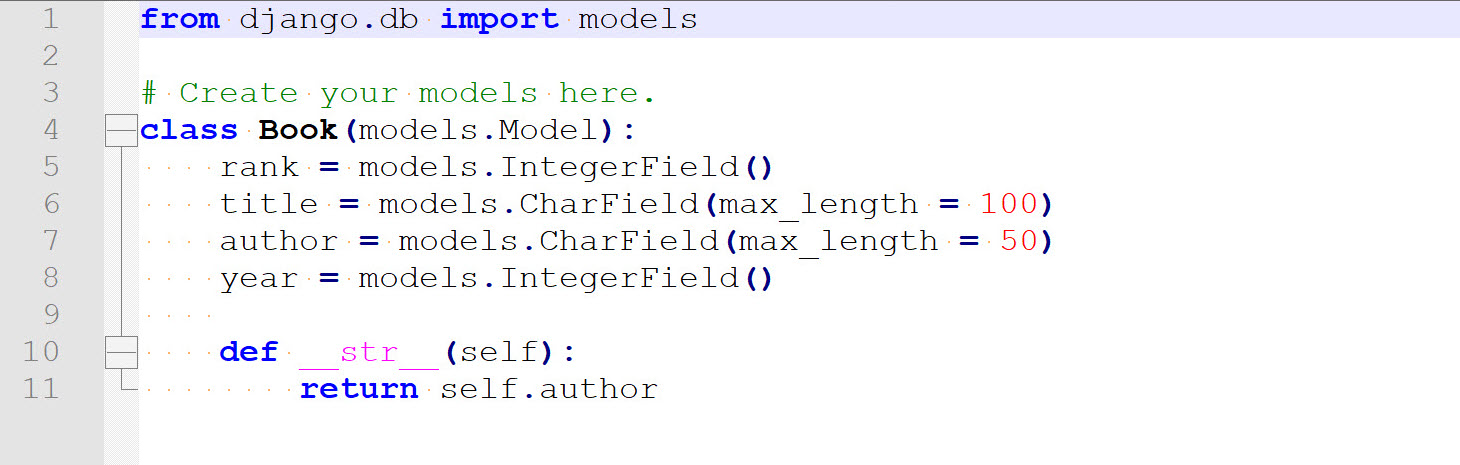
* Add your app to the INSTALLED\_APPS list in **django\_books\django\_books\settings.py**

****

## Create a Model for your Classic Books data

### Create a model definition to represent the data fields in our Classic Books file

* Add the following model definition to **django\_books\books\models.py**



## Make a Migration and migrate your data to the database

### Execute the **makemigrations** command from the **django\_books** directory

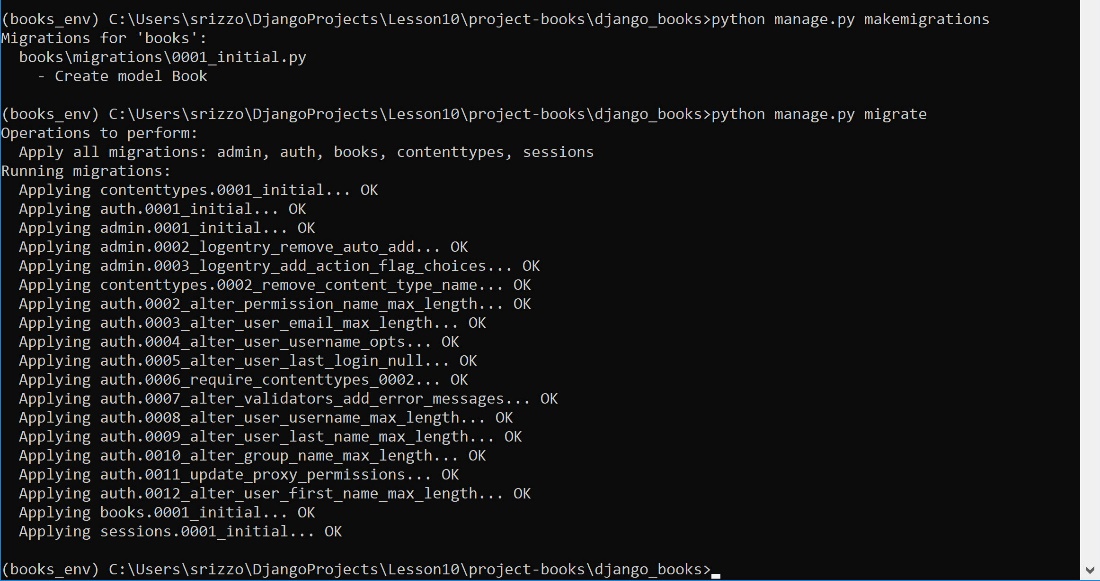
* **python manage.py makemigrations**

This creates a **migrations** directory (under **\books** ) and an **0001\_initial.py** file using the **models.py** file.



### Execute the **migrations** command to perform any outstanding migrations

* **python manage.py migrate**

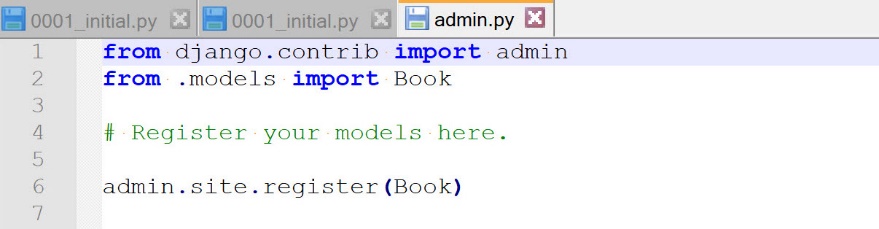


**SUMMARY:** At this point you have created a project, application, a model of your data, a database and the model of your data has been migrated to the database.

## Register the model to admin

### Register your models to the admin…

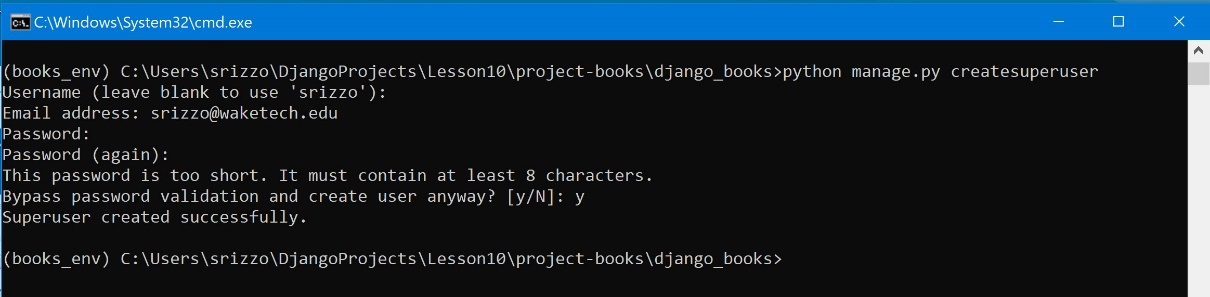
* Edit **django\_books\books\admin.py**



## Create SuperUser …and Viewing in the Admin panel

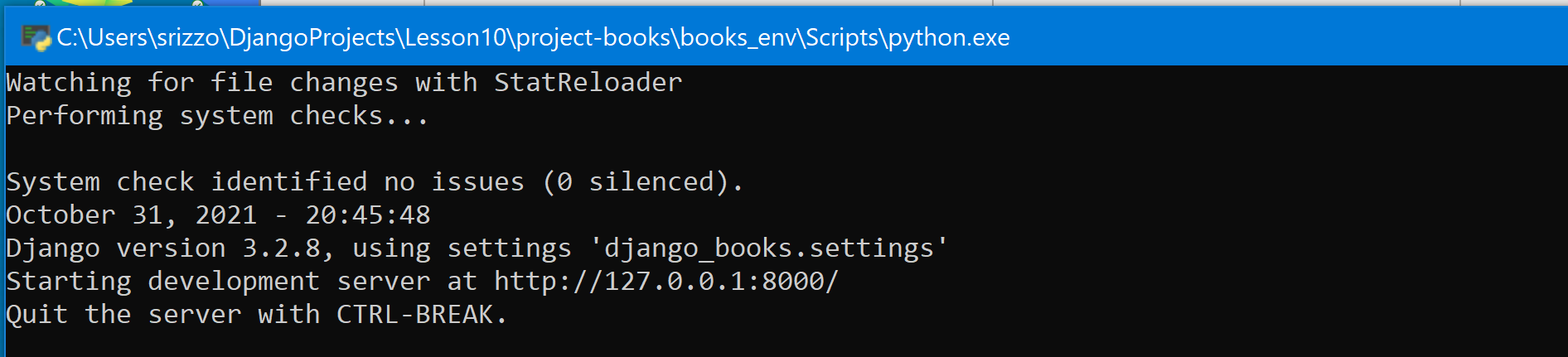
### Execute the **createsuperuser** command from the **django\_books** directory

* **python manage.py createsuperuser**
* Use your **WTCC email address**
* Use Password **csc221**



**If you server is not still running, start the server** …

* Execute the command to start the Django server (this will open a separate command window)
  + **start python manage.py runserver**

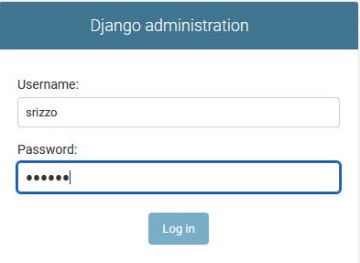


## Login & View in the Admin panel

### Login as administrator …

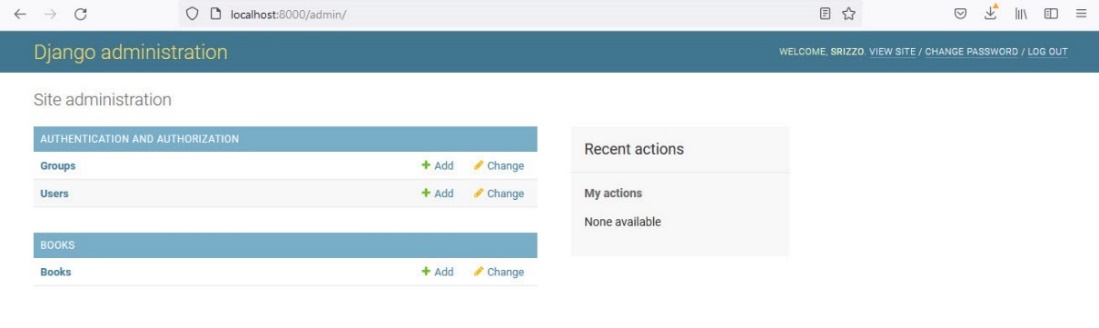
* Open a browser and go to the Admin login: <http://localhost:8000/admin>

## **SCREEN CAPTURE # 2**



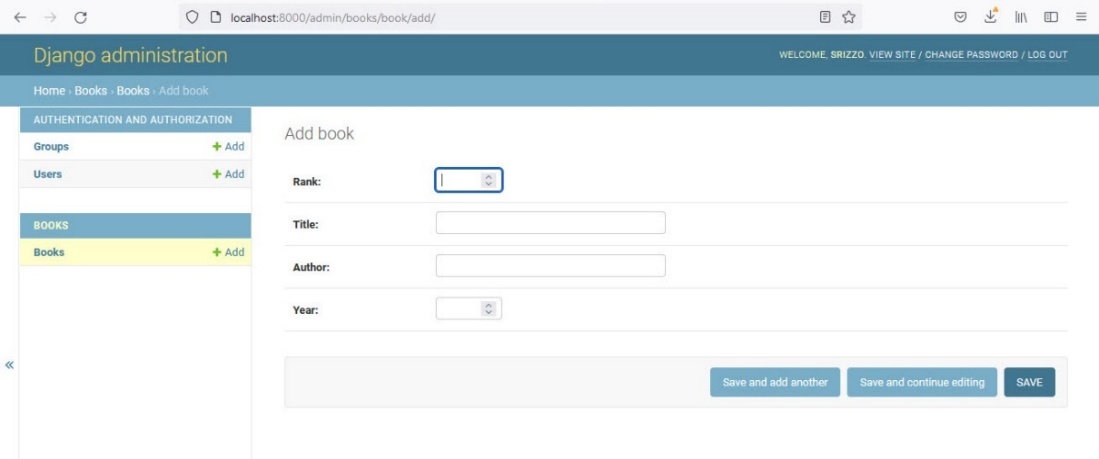
* Log in on Django administration page

You should see the following view …



If you click on **+ Add** (to the right of **Books**) you should see data entry fields for each of the fields defined by your model.

## **SCREEN CAPTURE # 3**



**SUMMARY:** At this point you have created an admin user and verified the model migration.

**DO NOT ENTER data via the Admin page. This is where the file you created in PART A will be used to populate the database with information for the 10 Classic Books.**

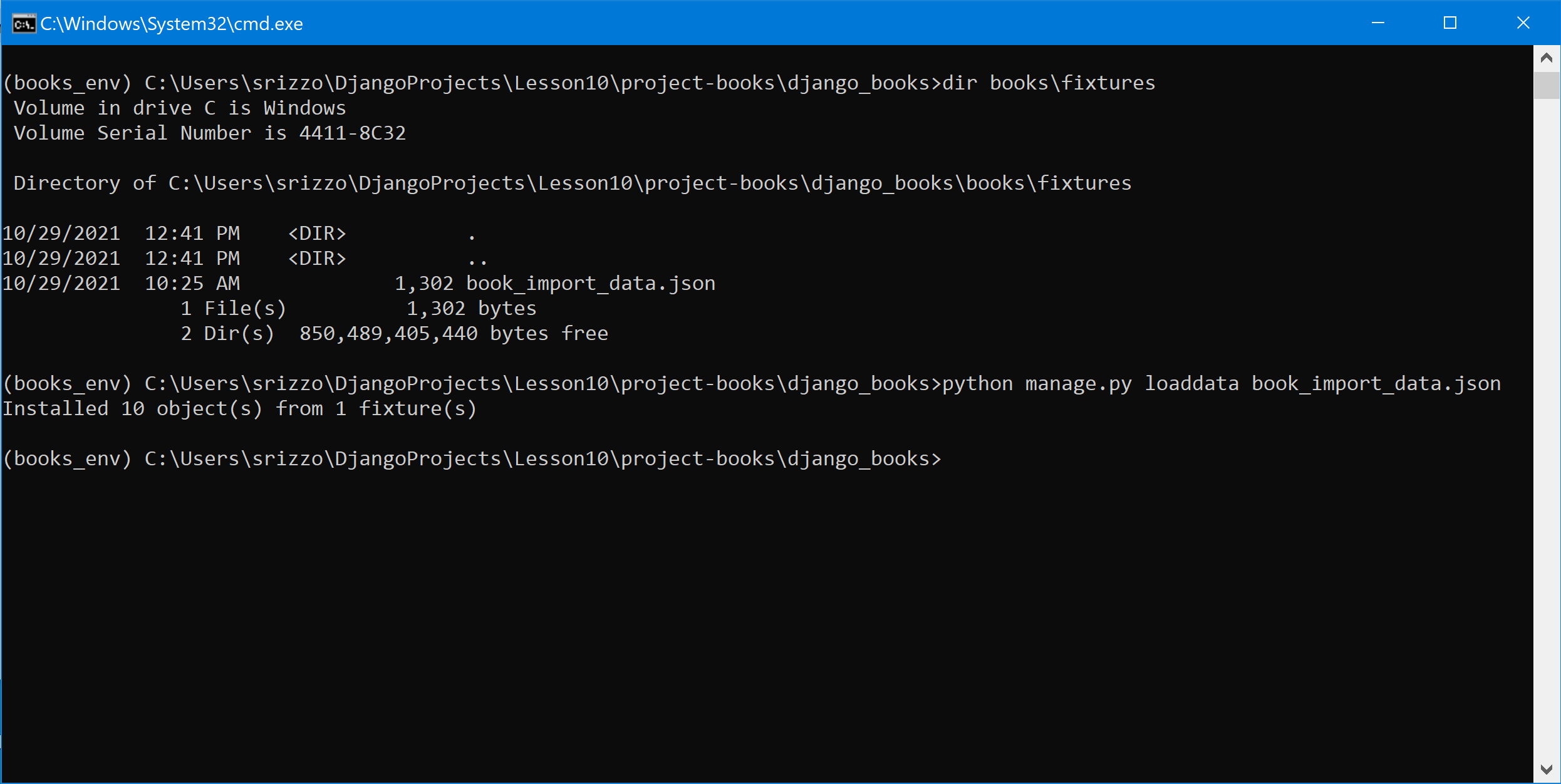
## Load JSON data into DATABASE (db.sqlite3)

**PREREQ** – You need the **book\_import\_data.json** file from PART A

### Run manage.py loaddata <fixturename>,

* Create **fixtures** directory as a subdirectory inside your **books** app directory
* Copy file **book\_import\_data.json** to the **fixtures** directory
* Run **python manage.py loaddata <*fixturename*>,** where ***<fixturename>*** is the JSON data file
  + You should see “Installed 10 object(s) from 1 fixture(s) message if successful

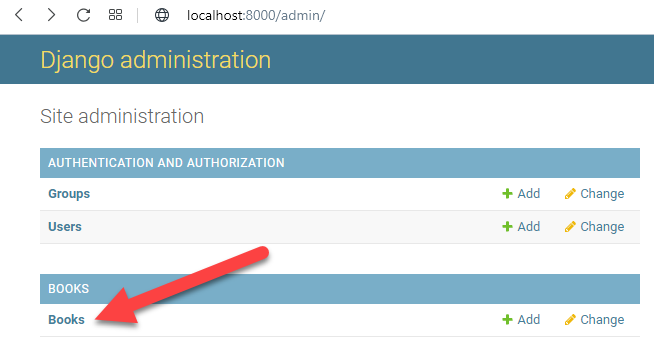
## **SCREEN CAPTURE # 4**



## Verify load JSON data into DATABASE (db.sqlite3)

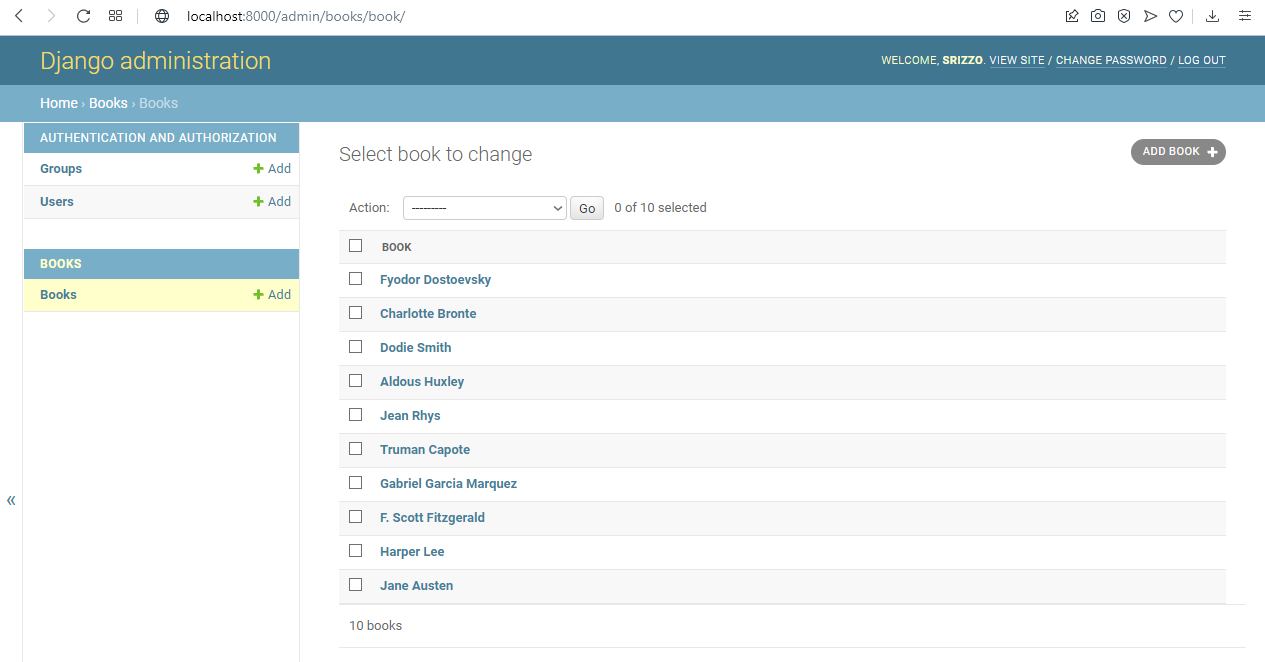
### Login as administrator …

* Open a browser and go to <http://localhost:8000/admin>
  + your userid
  + PWD = **csc221**
* Click on the **Books** link to display the loaded books



* Verify all 10 books display …

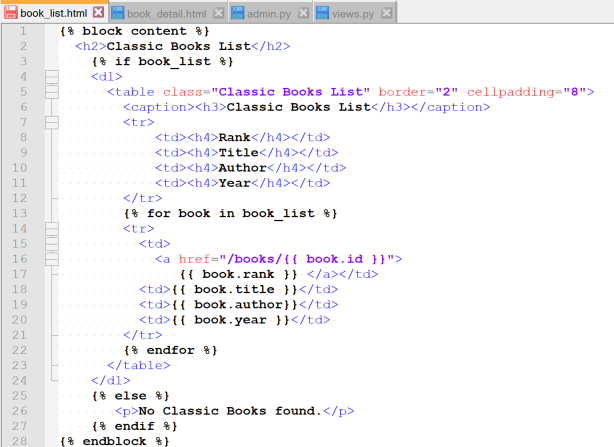
## **SCREEN CAPTURE # 5**



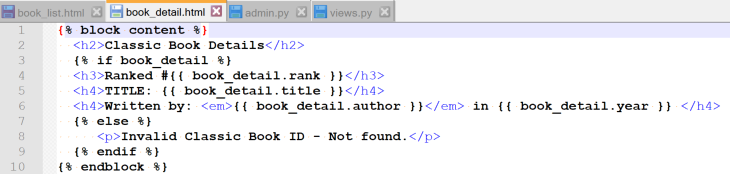
## Create List and Detail views (for non-Admin users)

### Create a new List and Detail view in your books app

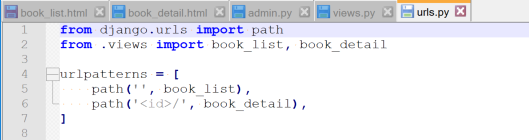
* Create new directories under the books app: **templates\books**
* Create file **book\_list.html** in this new directory
  + **django\_books\books\templates\books\book\_list.html**



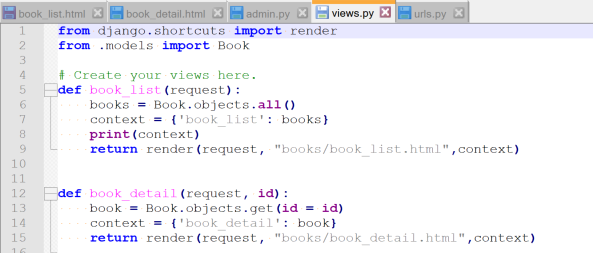
* Create file **book\_detail.html** in this new directory
  + **django\_books\books\templates\books\book\_detail.html**



* Create a new file under the books app: **urls.py**
  + **django\_books\books\urls.py**
    - **path(‘ ’, book\_list)** ~ is for the List View
    - **path(‘<id>/ ’, book\_detail)** ~ is for the Detail View



* Update file **views.py** under the books app
  + **django\_books\books\views.py**
    - Create the definition for **book\_list**
    - Create the definition for **book\_detail**

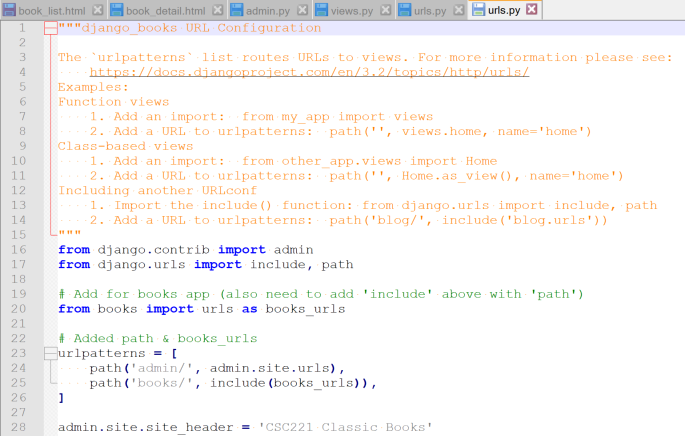


Notice, the **book\_detail** function definition takes an additional parameter (**id**). The **id** identifies the specific book to view on the detail page.

## Update the django\_books/urls.py to import books/urls.py

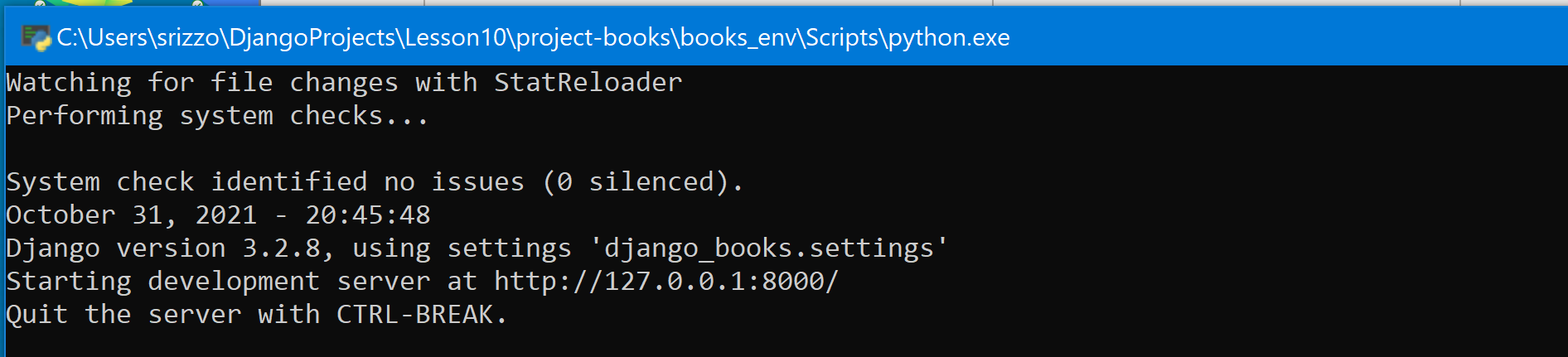
### Update the urls.py file for your django\_books project

* Update file **urls.py** to import our new books **urls.py** file
  + **django\_books\urls.py**
  + Adding admin.site.site\_header creates a web page title



If you server is not still running, start the server …

* Execute the command to start the Django server (this will open a separate command window)
  + **start python manage.py runserver**

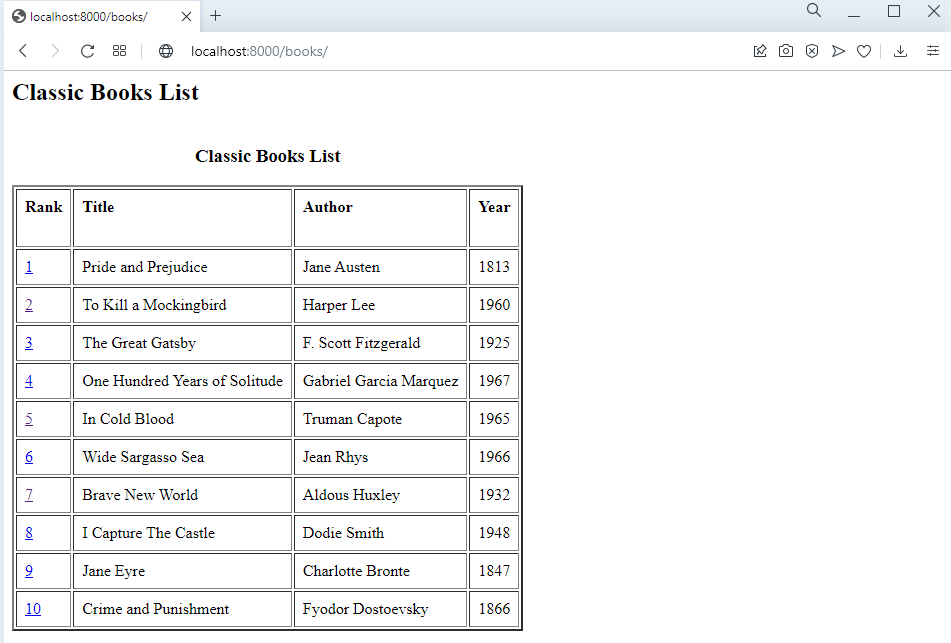


## Verify your List & Detail Views

### Verify you can view the loaded book data using List & Detail Views …

* Open a browser and go to <http://localhost:8000/books/>

## **SCREEN CAPTURE # 6**



* Click on any of the book **Rank** value links (I selected ‘1’ – Pride and Prejudice, select ANY book)

## **SCREEN CAPTURE # 7**

