Library Inventory Project

Made by Hafsa Hussain and Vincent PICOT

Installation guide:

- Be sure to have ***MySql 5.0*** or higher installed and running on your platform
- Create a database for the project in MySql, if you don't have another user than root, please consider creating a new user with access to only this project database
- Copy the ***.env.example*** file and rename it ***.env***, then fill the file - ***DB_NAME*** is for the database name you just created, ***DB_USER*** and ***DB_PASSWORD*** is for the user that is going to access the database
- Install Python 3.6 or higher
- Install python package ***Virtualenv*** https://virtualenv.pypa.io/en/stable/
- Create a new virtualenv and activate it (Follow the documentation linked above) **Everything bellow need to be done in the virtualenv**
- Install ***Django 2.1*** or higher https://docs.djangoproject.com/en/2.2/topics/install/ be sure to follow all the steps
- If you didn't through the Django installation guide, install ***mysqlclient*** https://pypi.org/project/mysqlclient/
- Install ***django-bootstrap-customizer*** 0.1.4 or higher https://johnfraney.github.io/django-bootstrap-customizer/
- Be sure to be in the root folder of the project before doing the following commands
- run ***python3 manage.py migrate***
- run ***python3 manage.py createsuperuser*** You'll need this user to connect to the admin site, be sure to know the access of this user
- run ***python3 manage.py runserver***
- Go to: http://127.0.0.1:8000/admin/ for the admin site
- Before going to the user site, go to the admin site and create a bootstrap theme and link it to the user site url (Examples in the django-bootstrap-customizer documentation)
- Check the id in the table ***django-site*** of the user site you linked above. In ***settings.py*** change ***SITE_ID*** with this id.
- Go to: http://127.0.0.1:8000 for the user site
- You're all set, have fun!
- If you are installing on a server instead of local, go to the documentation of ***Django***, there is a tutorial on how to deploy a django project.

User manual:

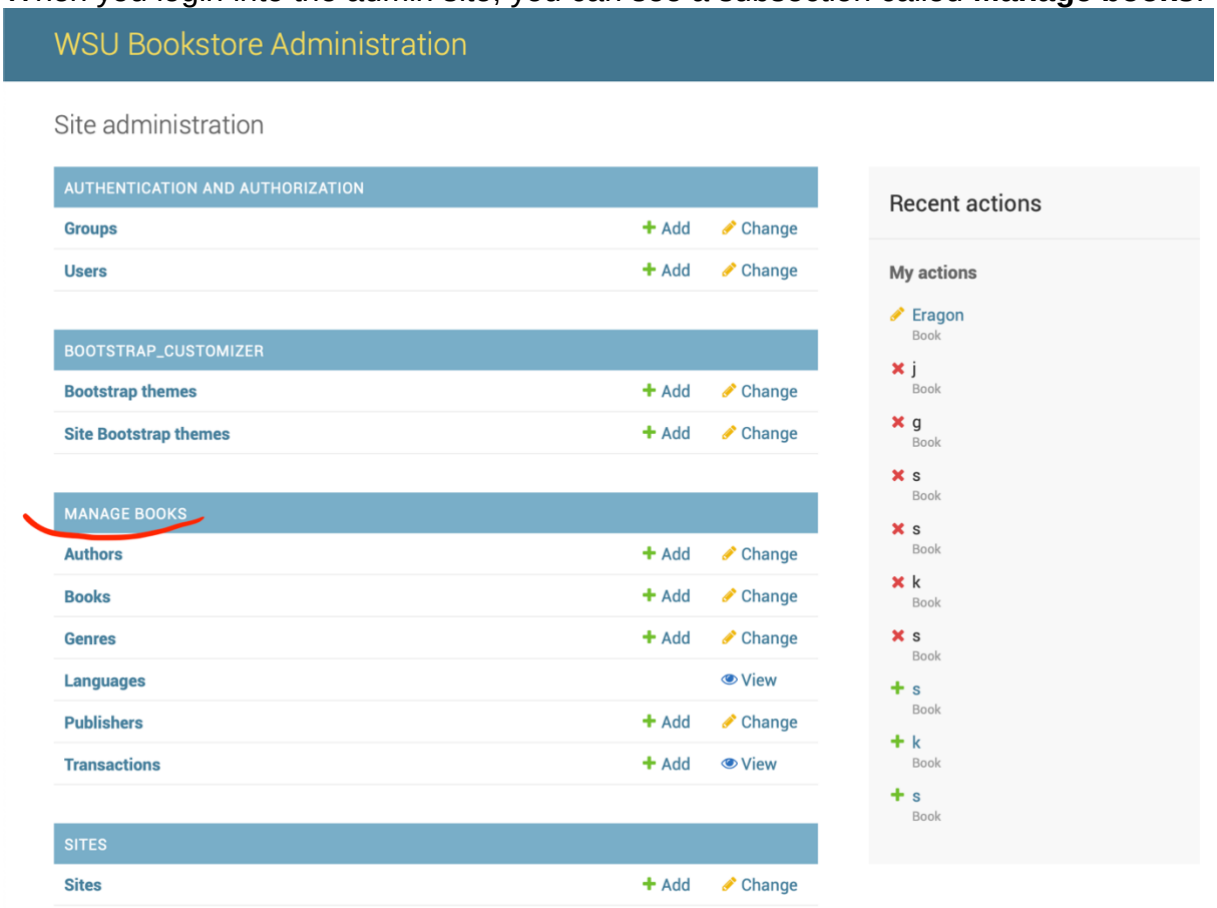
This project is separated into two parts, the admin site and the user site. We will cover extensively the first one, as its where most of the interaction happens, and then describe briefly the user site.

Admin site:

The admin site is a site that only the Bookstore/Library staff can use.
The role of this site is to organize books (and everything that revolves around it), the users that can access the admin site (And what kind of content they have access to), and finally, the colors of the user site (Primary color, secondary color, warning color, …).

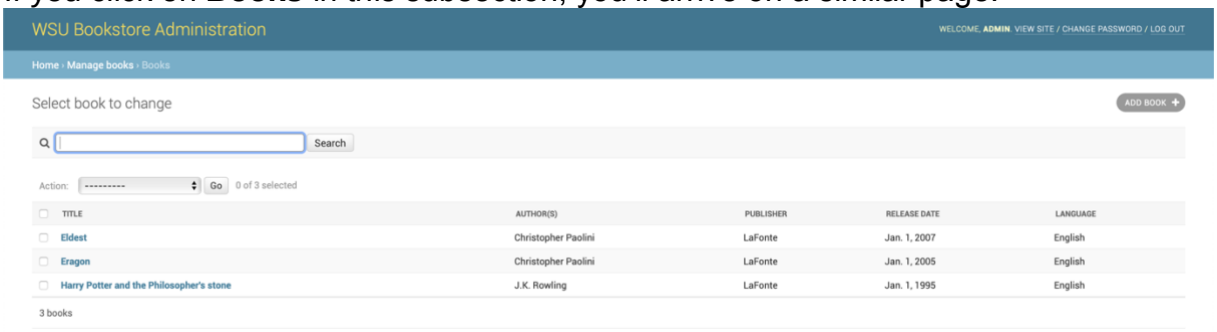
Organize books:

When you login into the admin site, you can see a subsection called **Manage books**.

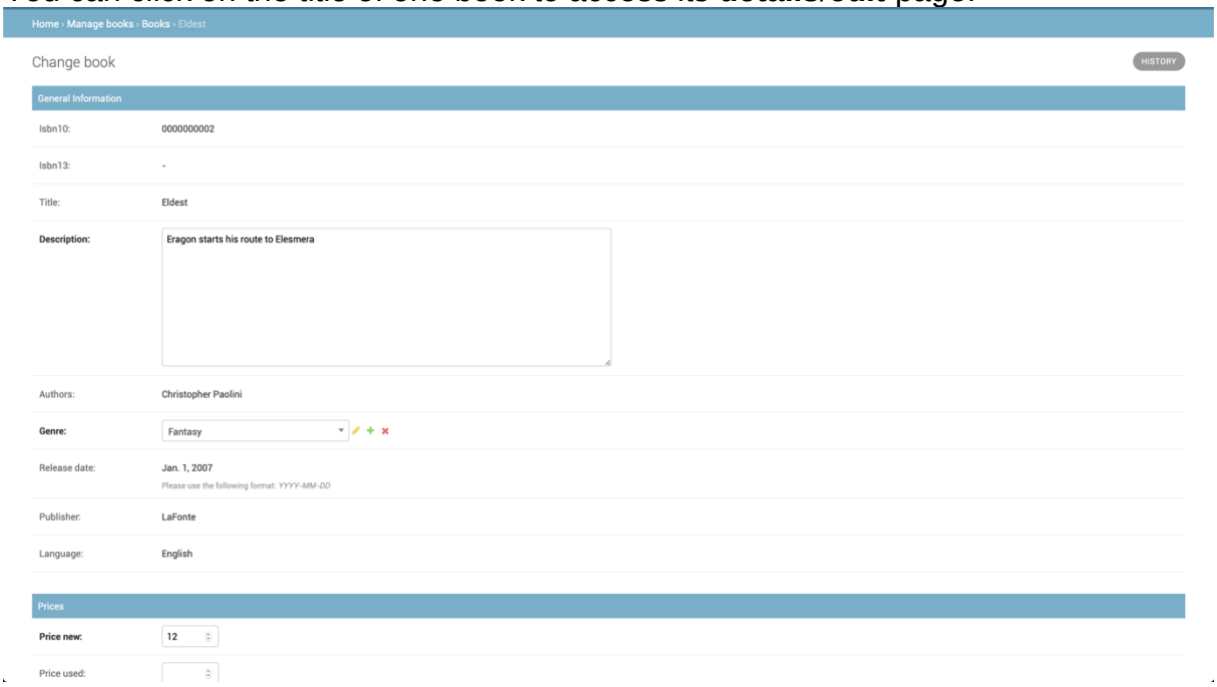


In this subsection, you can see everything that revolves around a book. If you are the admin user, or an all-access staff, you can add/edit/delete/view everything except for Languages – that you can view only – and transactions – that you can't edit.

If you click on **Books** in this subsection, you'll arrive on a similar page:



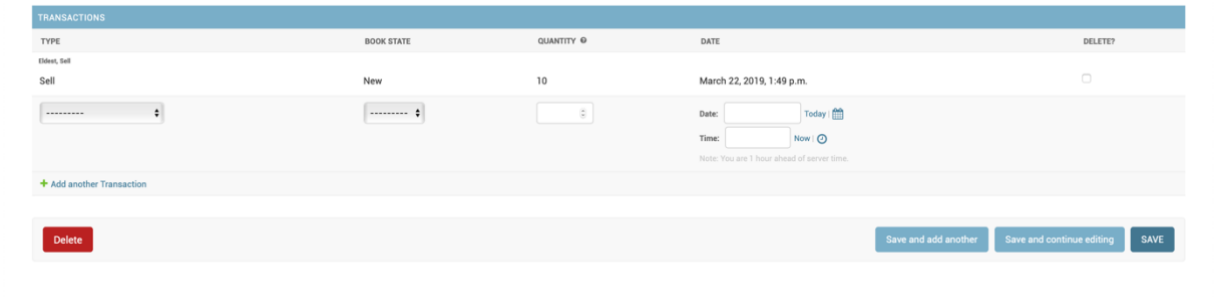
You can click on the title of one book to access its **details/edit** page.



You'll see that you are not able to edit every attribute (This is perfectly normal, a book's title isn't supposed to change, neither its ISBN, and so on).
You can edit whatever is editable, and then save the book.

You can also create a book by clicking on the **Add book** button. You'll arrive on the same page layout as the **Edit** page, except that every field will be empty. (Notice that some attributes names are in bold, and other in normal font. Bold name means that the field is mandatory, and you won't be able to save a book without this field.)

At the end of the book **edit/add** page, you have a **Transactions** subsection.



In this subsection, you can add several transactions for the book in question (Shipment of books, sell books, …). It will give you a warning if you try to sell a book that is not in stock, or that not enough books are available.

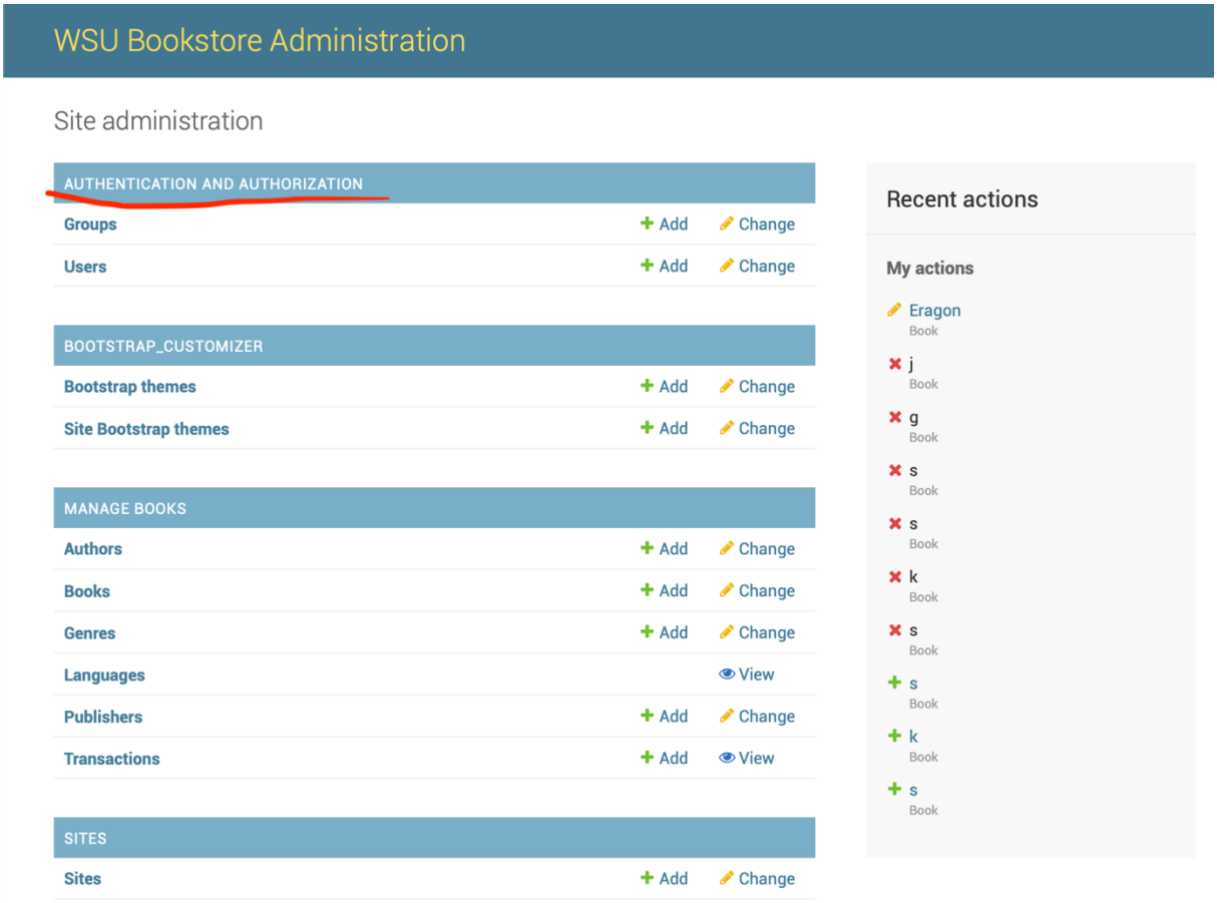
You can also add transaction in the **Transaction** entry in the **Manage books** subsection. However, it's a bit more complex as you need to manually select the book in which you want to assign the transaction when adding a new one.

To delete a book, you can select the book on the book listing, and then in the dropdown on the top of the list named **Action** select **Delete selected books**, otherwise, you can also go to the edit page of a book and click delete at the bottom of the page.

All the other subsection of **Manage books** – Authors, genres, publishers, … - works the same way as described for books when trying to add/edit/delete/view them.

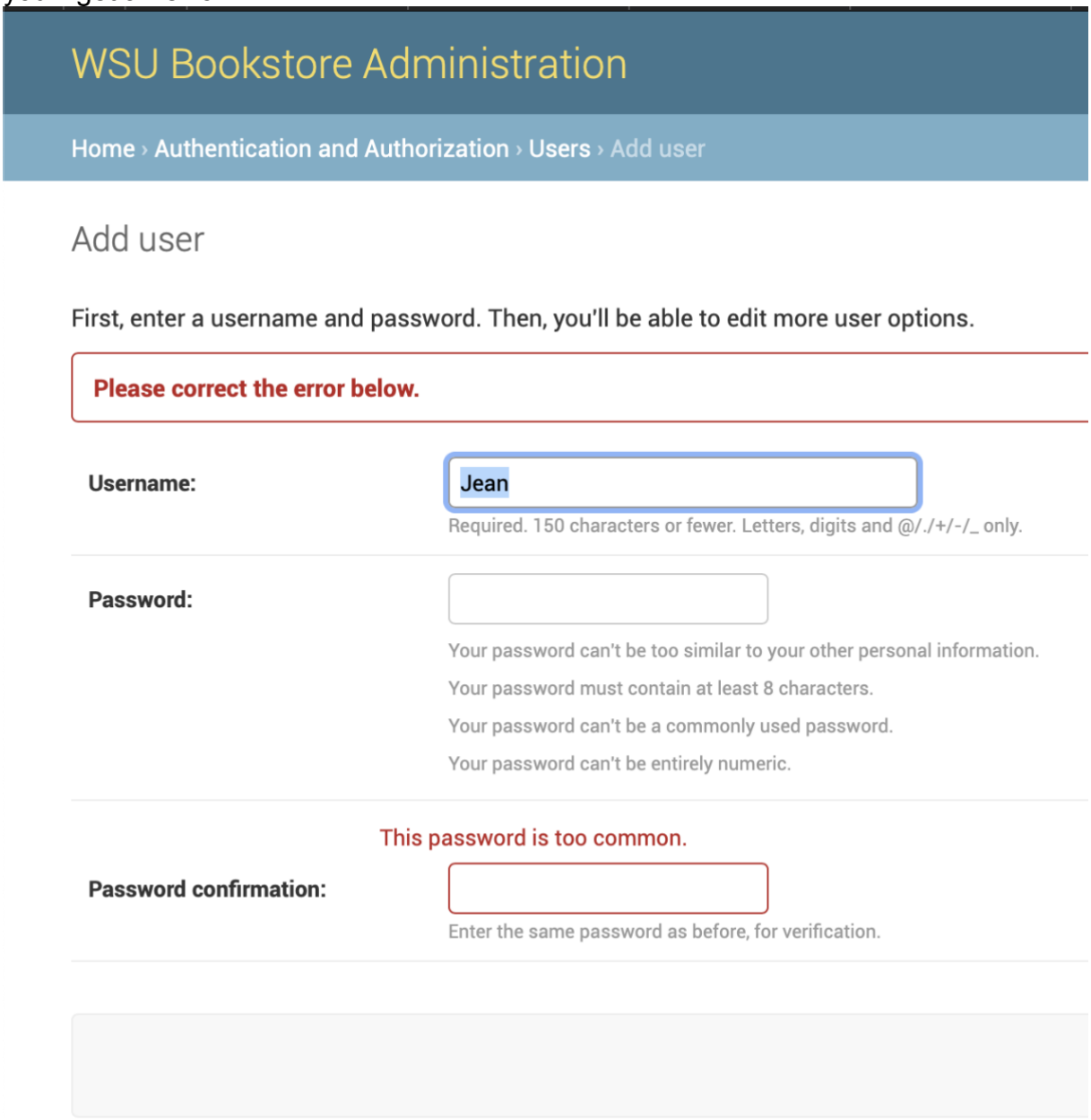
Manage staff members and access rights:

You can manage staff that can access the admin site, and also their rights in the **AUTHENTICATION AND AUTHORIZATION** subsection of the admin site.

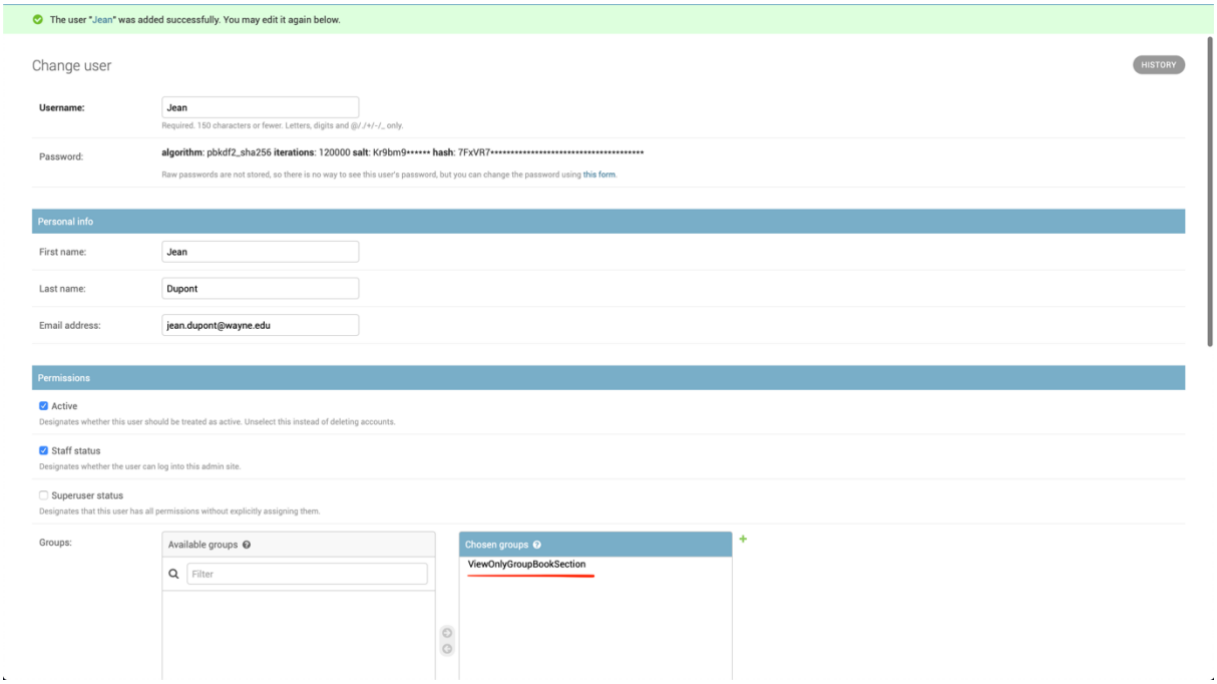


The **Users** entry allows you to create new staff members (Username, password) and to assign them to a group.

The **Groups** entry allows you to create a group, and all the members of this group will have the accesses defined in this group. (For example, can view everything, but cannot edit/add/delete anything).
When you attempt to create a user with a too common password (e.g azertyuiop), you'll get an error.

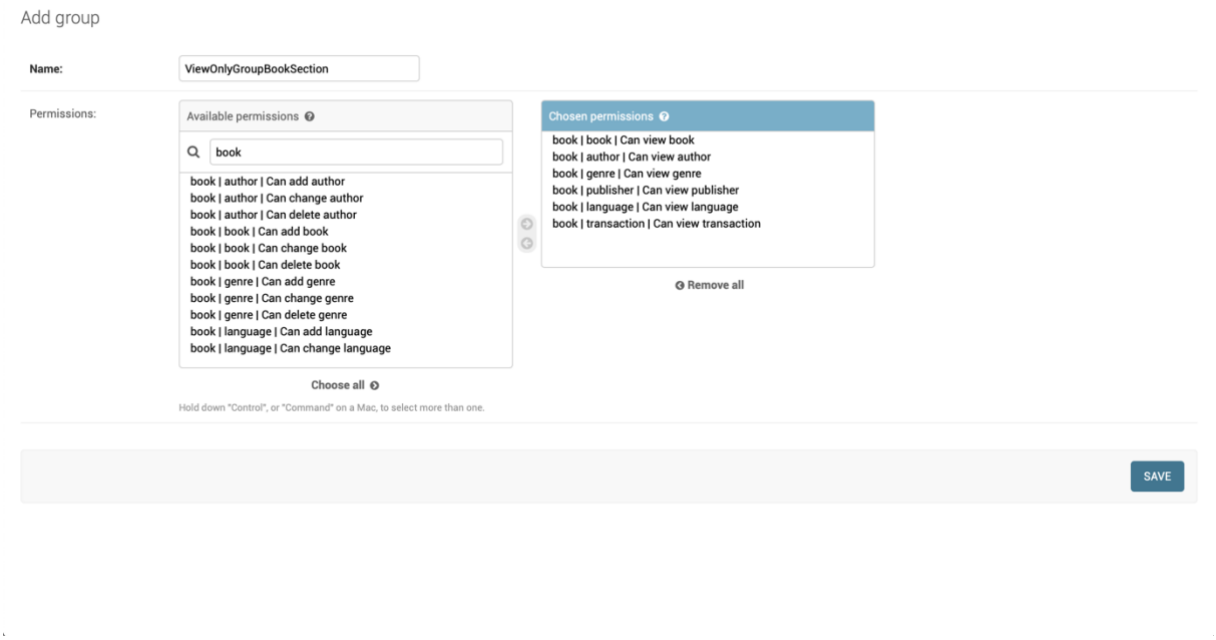


Once you created the user (username and password), you'll arrive on a more detailed page asking for a lot more information, like assigning the user to groups, its firstname, lastname, …



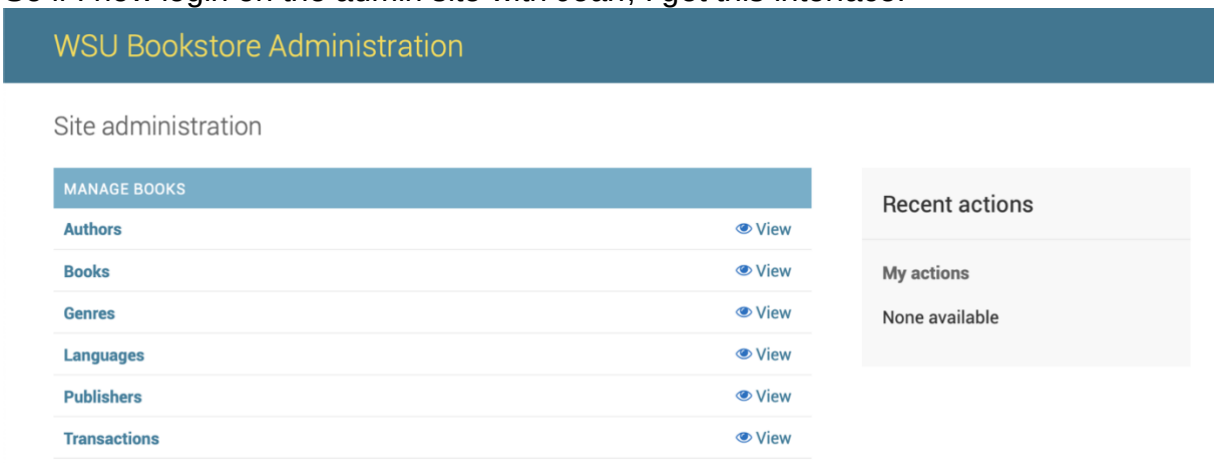
In this, we can see that I assigned the user to the group **ViewOnlyGroupBookSection** described just below.

You can create a book this way:



Assigning a name and permissions to the group. In this example, I give to this group members only the permission to view information beneath **Manage books**. It means they won't be able to add/edit/delete anything, neither view the website templates or the staff users.

So if I now login on the admin site with *Jean*, I get this interface.



As we can see, *Jean* cannot edit/add/delete anything, neither can he view something else than the **Manage books** subsection.

Change the user site color:

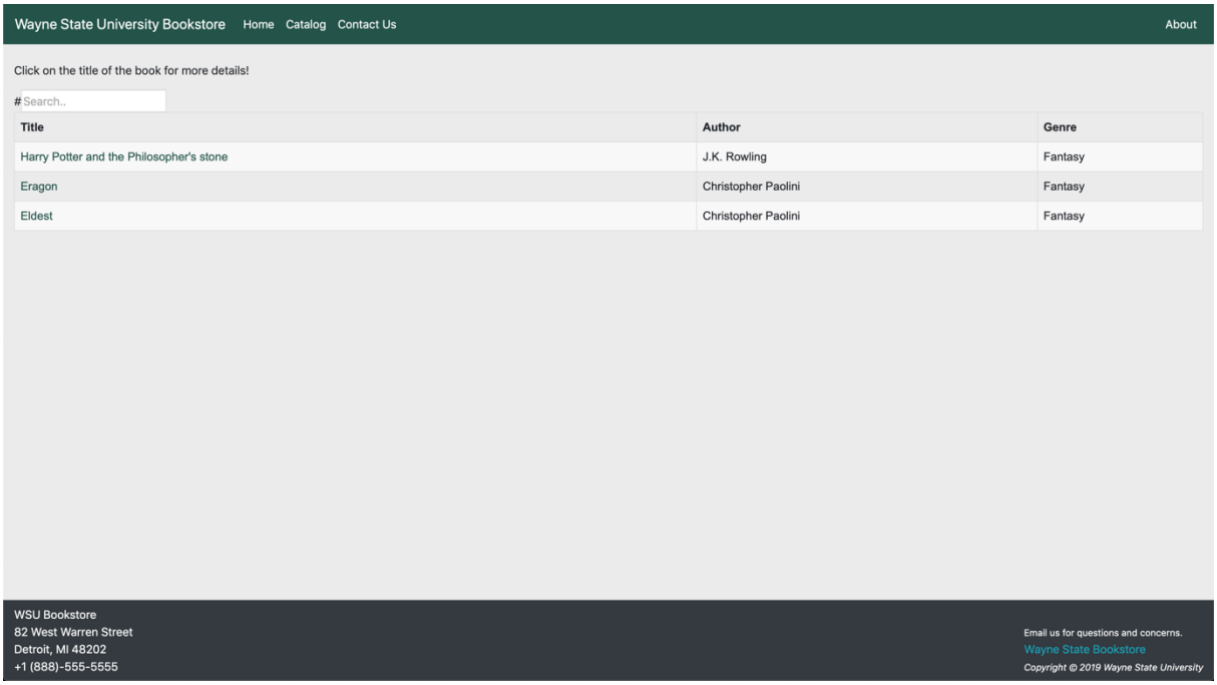
If you go under **Bootstrap themes**, you can add new themes, that will correspond to your user site colors (Once you've linked the template to the URL in **Site bootstrap themes**). In the theme, you'll see several color inputs. You can edit all of them (The two main ones being **Primary** and **Secondary**). If you change those colors, save the theme (The saving can take some time), and go to the user site, you'll see the color changed to what you've set.

User site:

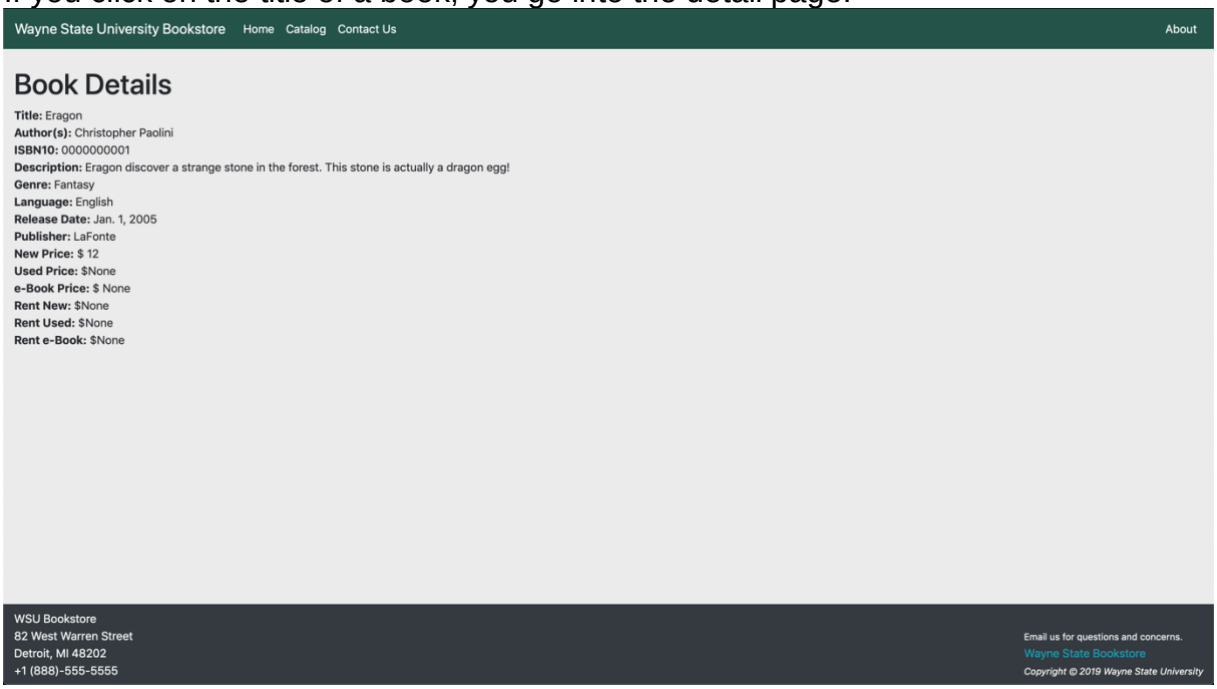
There are three main pages on the user site, the home page, the catalog page with book detail page, and the contact us page.

On the home page, you just have an introduction text, not much interaction possible instead going to another page through the navbar.

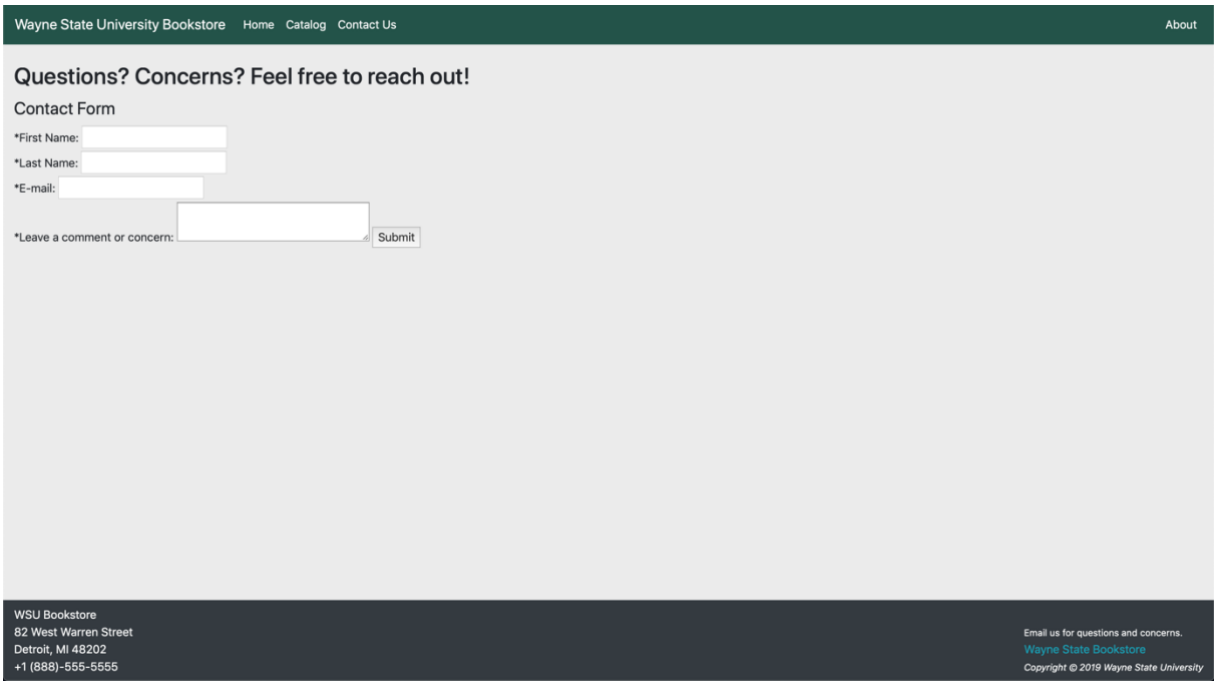
On the catalog page, you have the listing of all the books available in the bookstore, it looks like this:



If you click on the title of a book, you go into the detail page.



The last important page of the user site is the contact us page. It a basic form to contact the bookstore.



Pre-project Team Review:

Team Name: LibraryInventory

	Names:	Access IDs:
1)	Hafsa Hussain	gh7070
~~2)~~	~~Haris Rovcanin~~	~~fs7356~~ **Update:** Left the group
3)	Vincent PICOT	gr9185

Github accounts:
Vincent-picot
Hafsa-hussain

What is one goal you want to accomplish with this project?
One goal we wanted to accomplish with this project was to discover how to create a website in Python along with interacting with a database through Python.

List the names of your team members that you have met and spoke personally.
Since we're just two in the groups, we met each other several times.

What is one goal your team want to accomplish with this project?
Same answer as the first question.

Do you and your teammates have decided a project?
Yes.

If you have decided on a project, describe the project here with your own words
A CRUD application in the form of a Website. This application would simulate a university bookstore.

URL of GitLab or Github project. The repository should contain the team contract agreement on a folder named "management"
https://github.com/vincent-picot/LibraryInventory

Updated project proposal:

Team Name : LibraryInventory

	Names:	Access IDs:
1)	Hafsa Hussain	gh7070
~~2)~~	~~Haris Rovcanin~~	~~fs7356~~ **Update:** Left the group
3)	Vincent PICOT	gr9185

Project Proposal Form :

1. Describe your project in one paragraph.
   We want to create a web application to keep track of a library's inventory (The university bookstore for example).

2. Did you include a prototype presentation file into your 'Design' folder in your project repository?
   No, we did not as we don't know yet how our application will looks like or be organized.

3. What type of project are you doing (CRUD – Game – Data science – etc)?
   Our application will be a CRUD project (Add, delete, edit, access a book).
   Update: We can add/edit/delete/access books, but also everything that revolves around it (Staff user, authors, publisher, book transaction, …)

4. What is the URL of your repository?
   https://github.com/vincent-picot/LibraryInventory

5. Did you include your instructor and TA in your repository?
   Yes.

6. Did you include the following files in a folder named 'Management' in your project repository?
   - Team contract
   - Planning document
   - Gantt Chart (basic and advanced features clearly annotated) (**Update:** We gave up the Gantt Chart as we were only two, and meeting each other often was better from our point of view)

7. Did you add your tasks to your project management tool? Which tool are you using? E.g. Project in Github:

Update: Like with the Gantt Chart above, we didn't really used any management tool as it was easier for us to just meet.

8. How familiar are your team members with OOP and UML diagrams?

Hafsa and Haris are familiar with the concept of OOP but never really went into it.
Vincent knows both OOP concept and UML diagrams. He worked a lot with OOP.

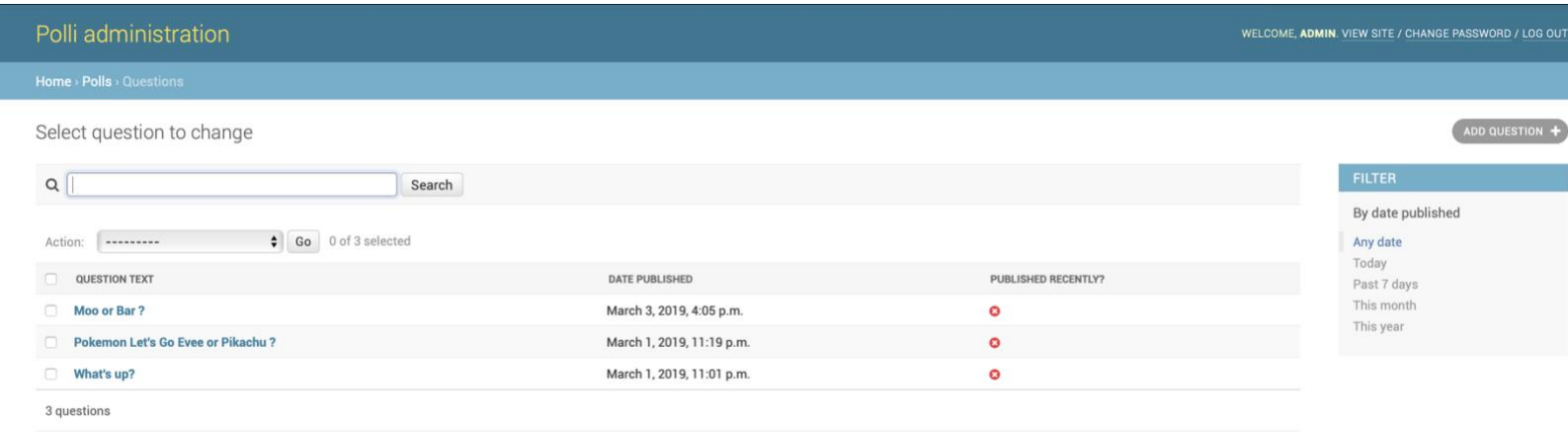
Update: After this project, Hasfa knows better how OOP worked as she had to use it at several places.

Updated project outline:

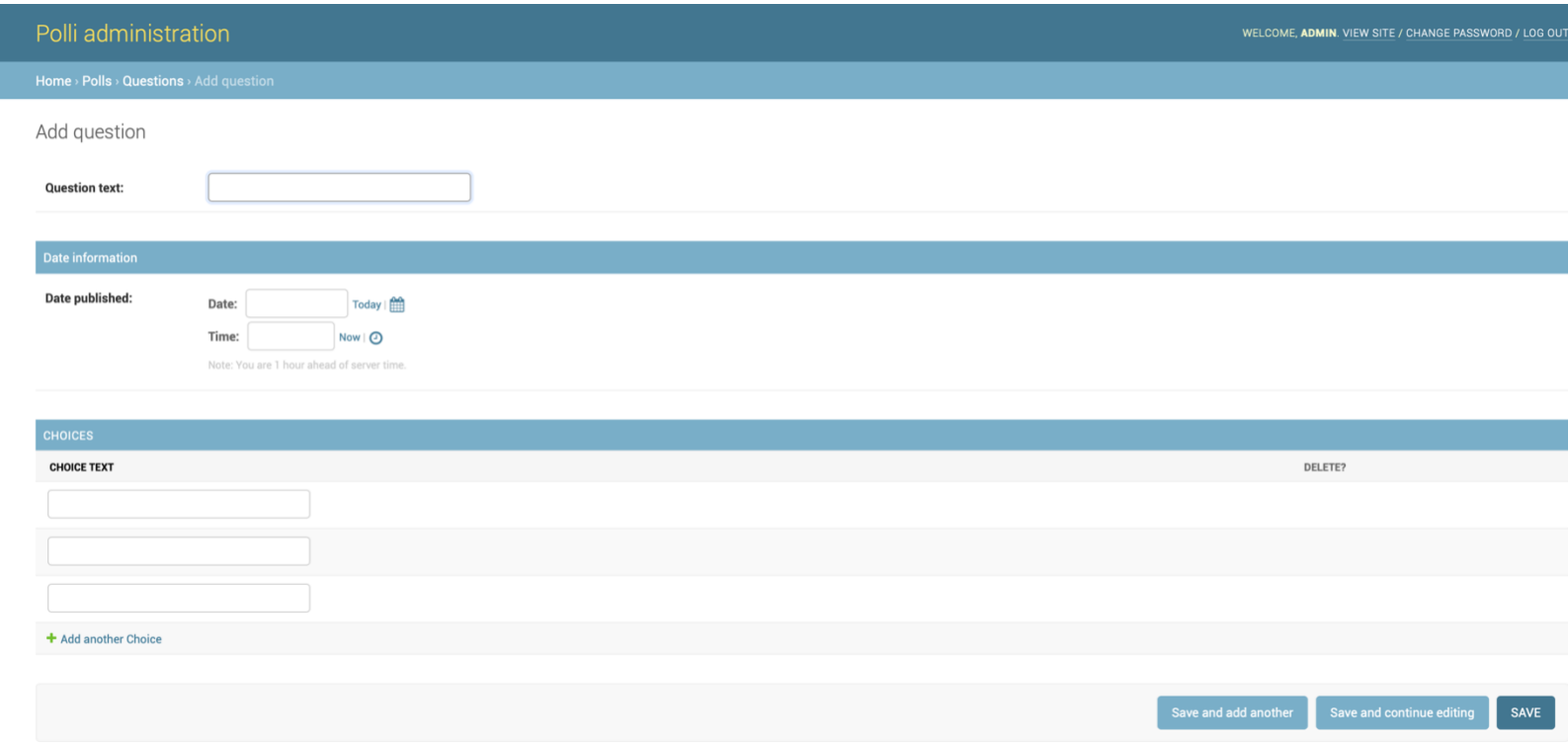
1) Vincent Picot gr9185
   Hafsa Hussain gh7070

2)
   - We all found multiple open source projects related to the aspect of a library operating system but decided we wanted to implement our own project from scratch.
   - Yes, we tried to use the Django framework and decided to use it. Below are screenshots of the tutorial projects that we followed; the full code is on the GitHub repository under the TestDjango folder.

Administration page listing the questions.



Administration page to add a new question and choices



index.html page

- Moo or Bar ?
- Pokemon Let's Go Evee or Pikachu ?
- What's up?

vote.html

page

Pokemon Let's Go Evee or Pikachu ?

- ◯ Evee
- ◯ Pikachu

Vote

results.html page

Pokemon Let's Go Evee or Pikachu ?

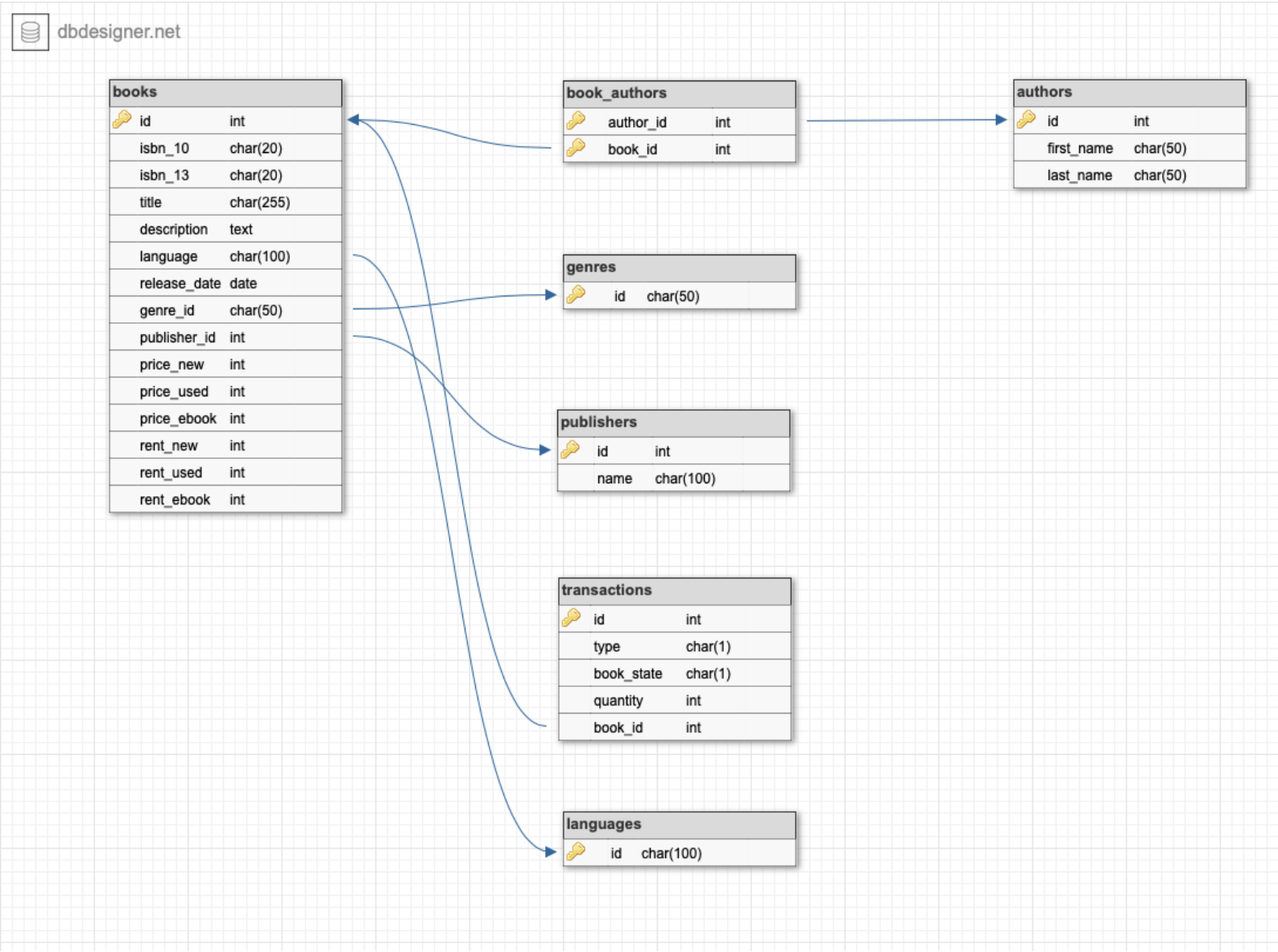
- Evee -- 5 votes
- Pikachu -- 4 votes

Vote again?

- No, we did not get stuck while trying demos.

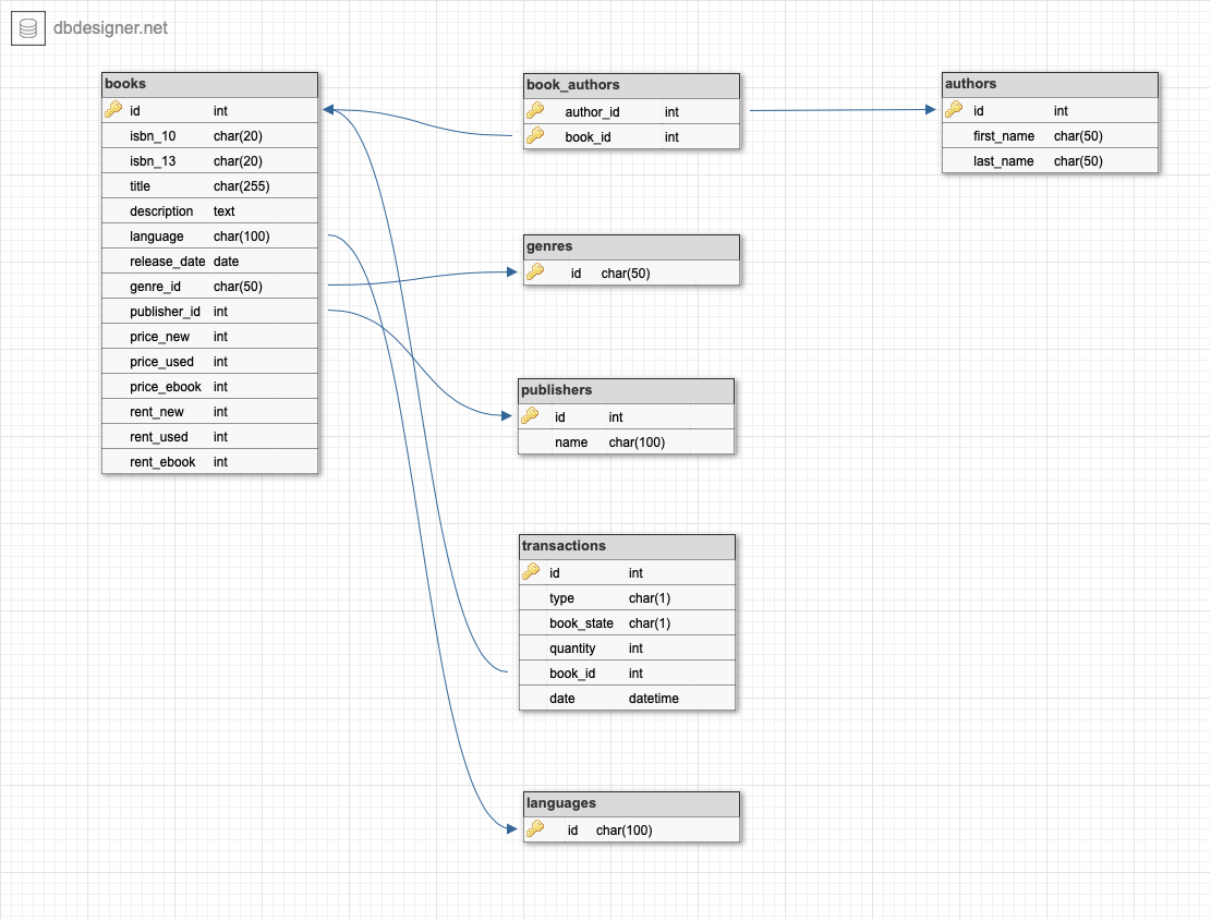
- Yes, we have started to implement the front(Hafsa) and back end(Vincent) separately.
  The process was first to define a database schema as we are doing a CRUD application. For that we made some reflexing on what our application should be able to do. We ended up with this schema:

Update: This test project is still in the repository and submitted in the code ZIP file. It helped us a lot when developing our actual project. That is because this example project covers a lot of the different aspects of Django.



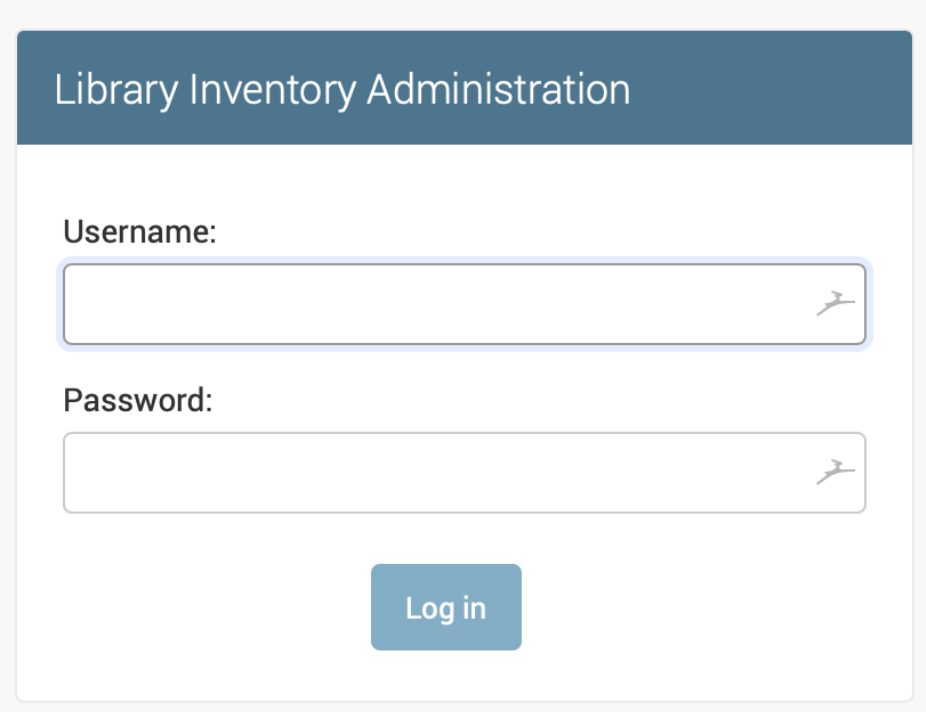
Vincent then created the project structure that Django need and recommend and started to implement this database model into Python Class to use with the Django framework.
In the meantime, Hafsa started to work on the html and css for the index page of our project and linking this html file with the Python code and Django to create the user interface.

Update: The database structure changed a little bit, especially in transaction where we added the datetime of the transaction, below is a screenshot of the current database structure.

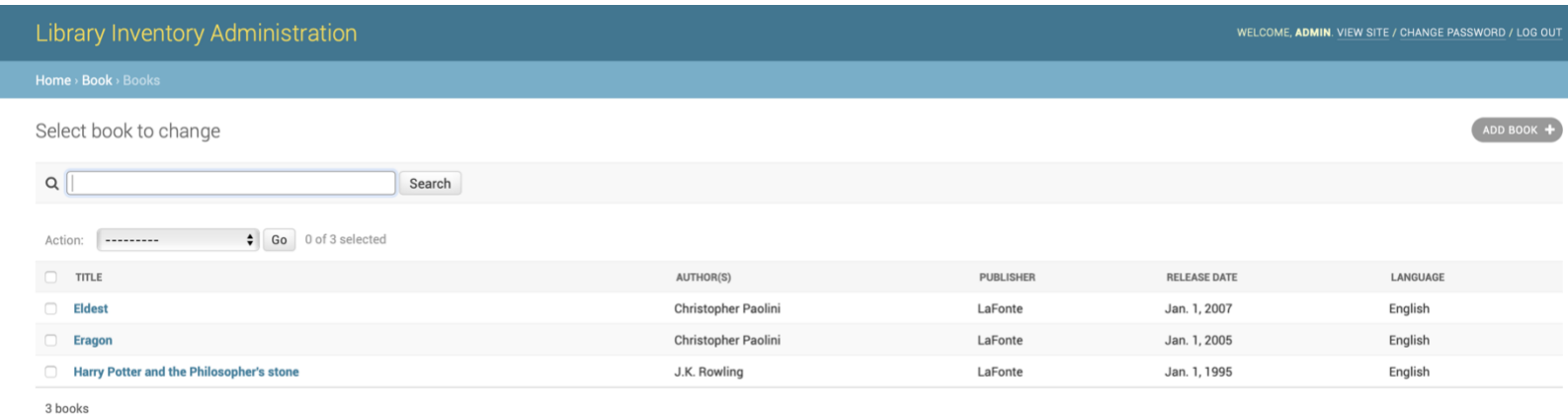


Here are some screenshots of what we did already:

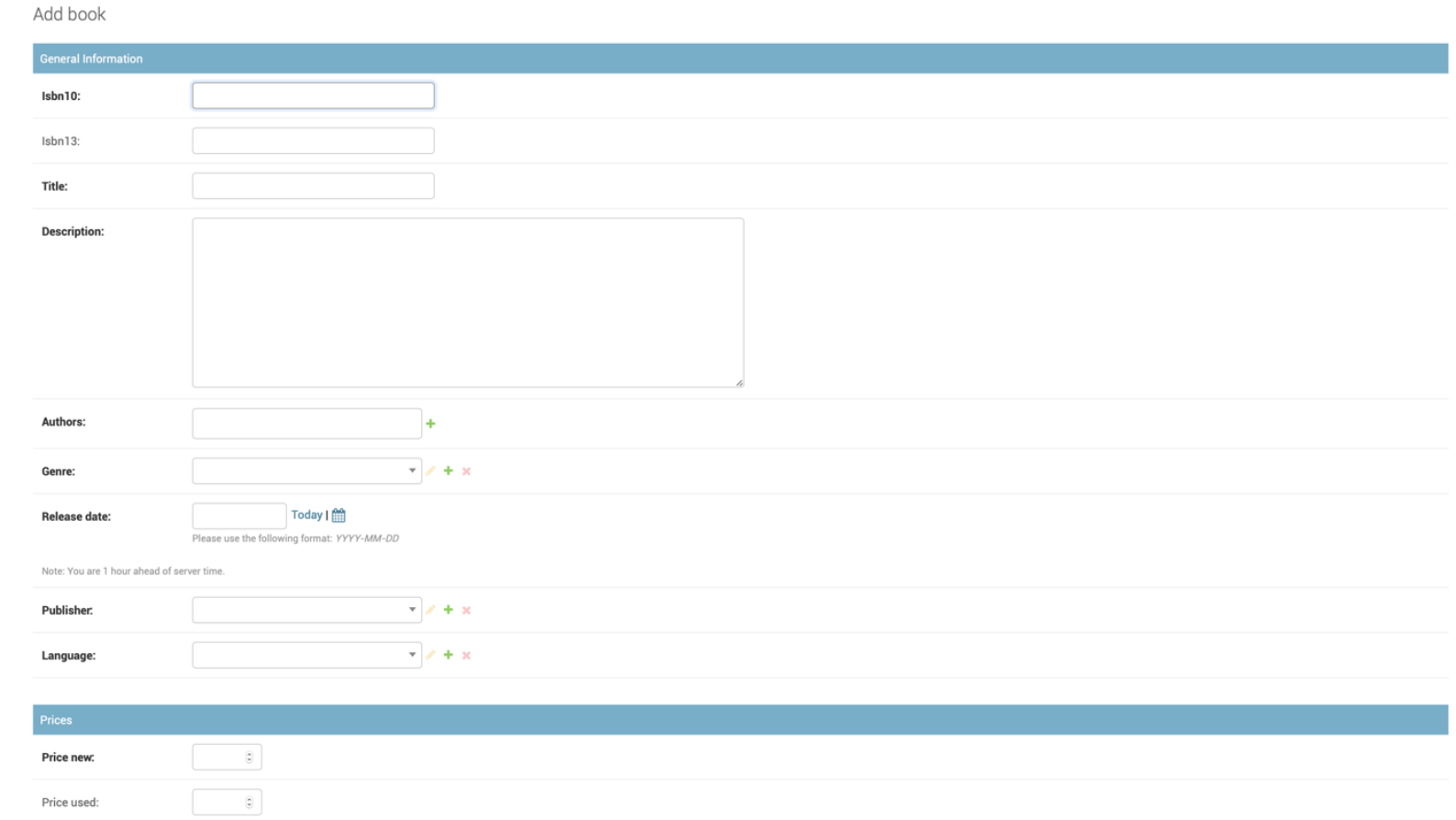
Login page of the administration site



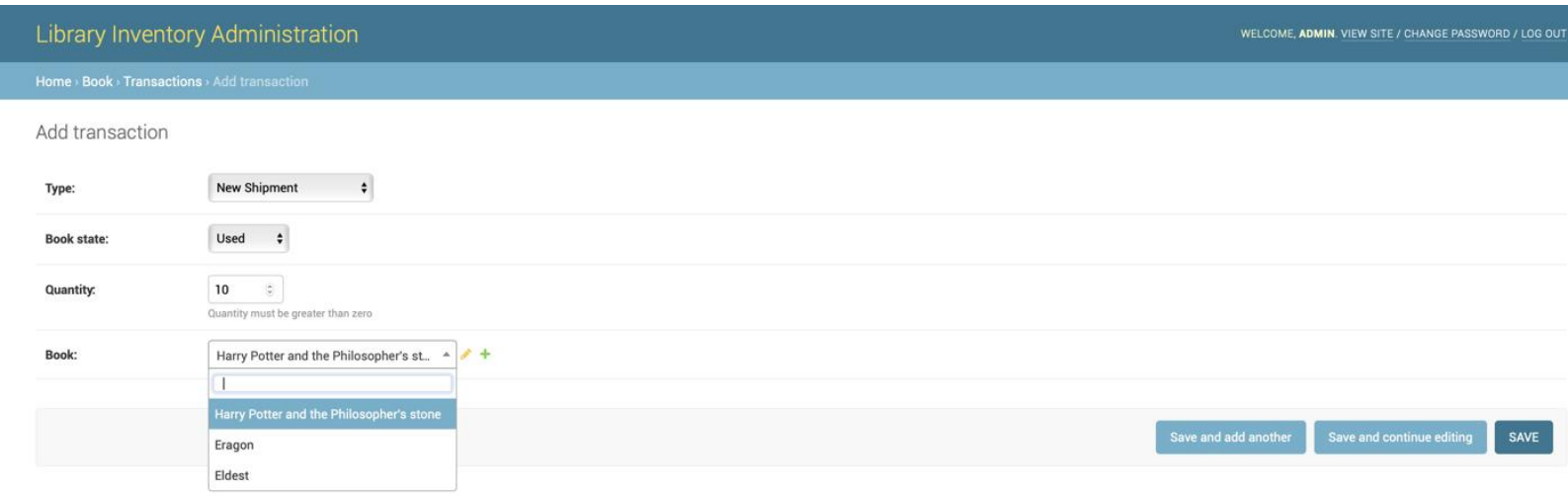
List of the book page in the administration site



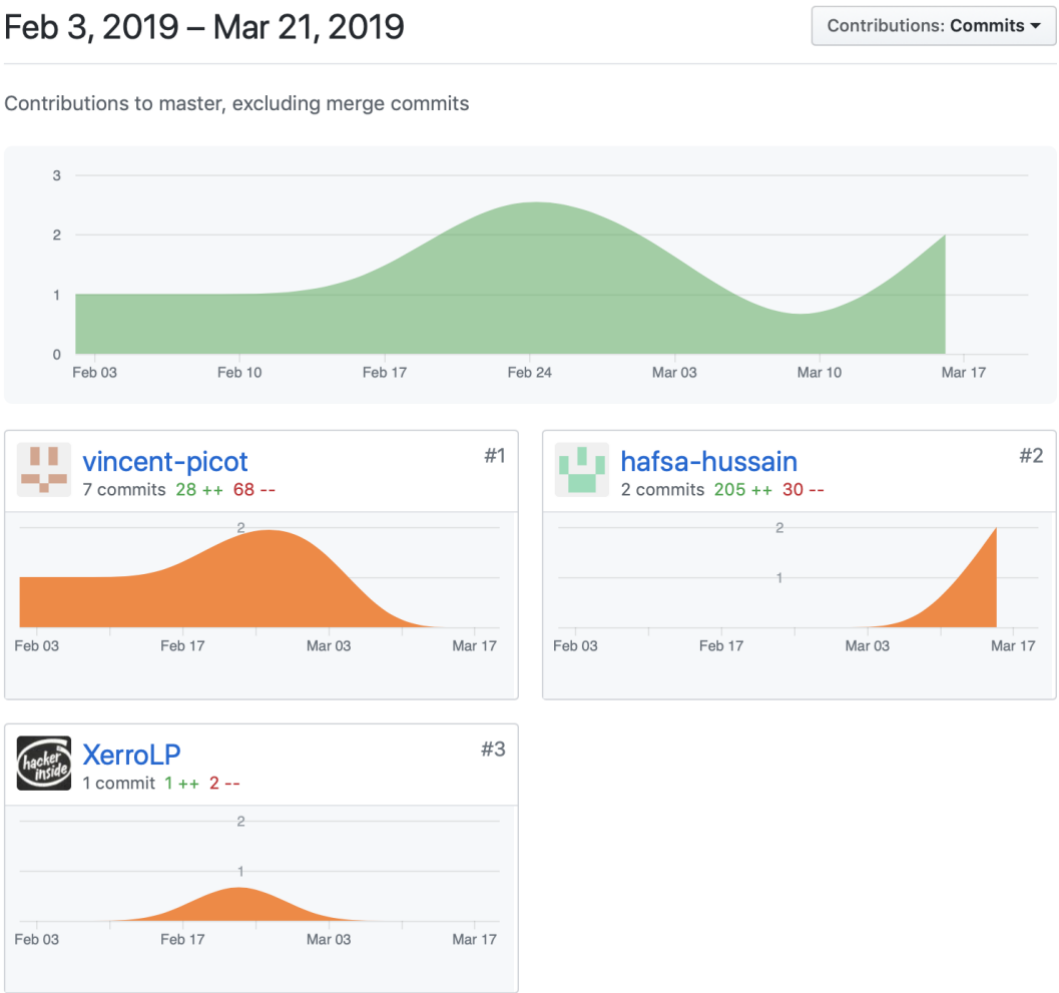
Page to add a new book in the administration site

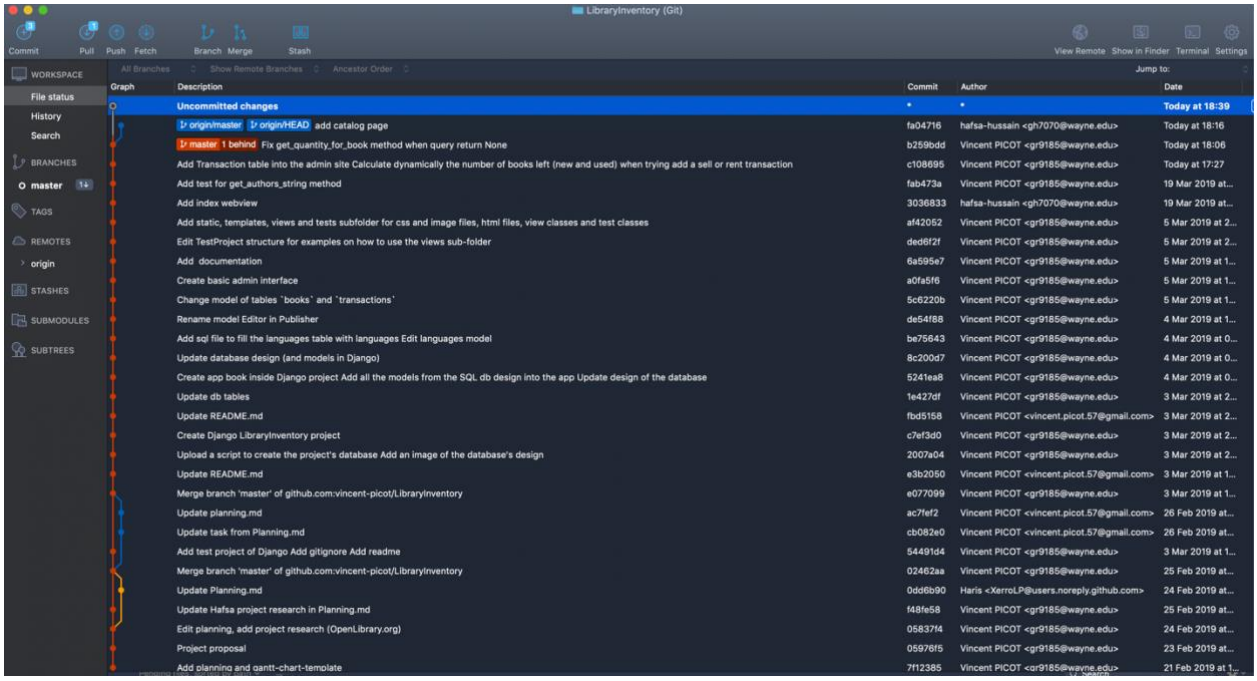


Page to add a new book transaction in the administration site

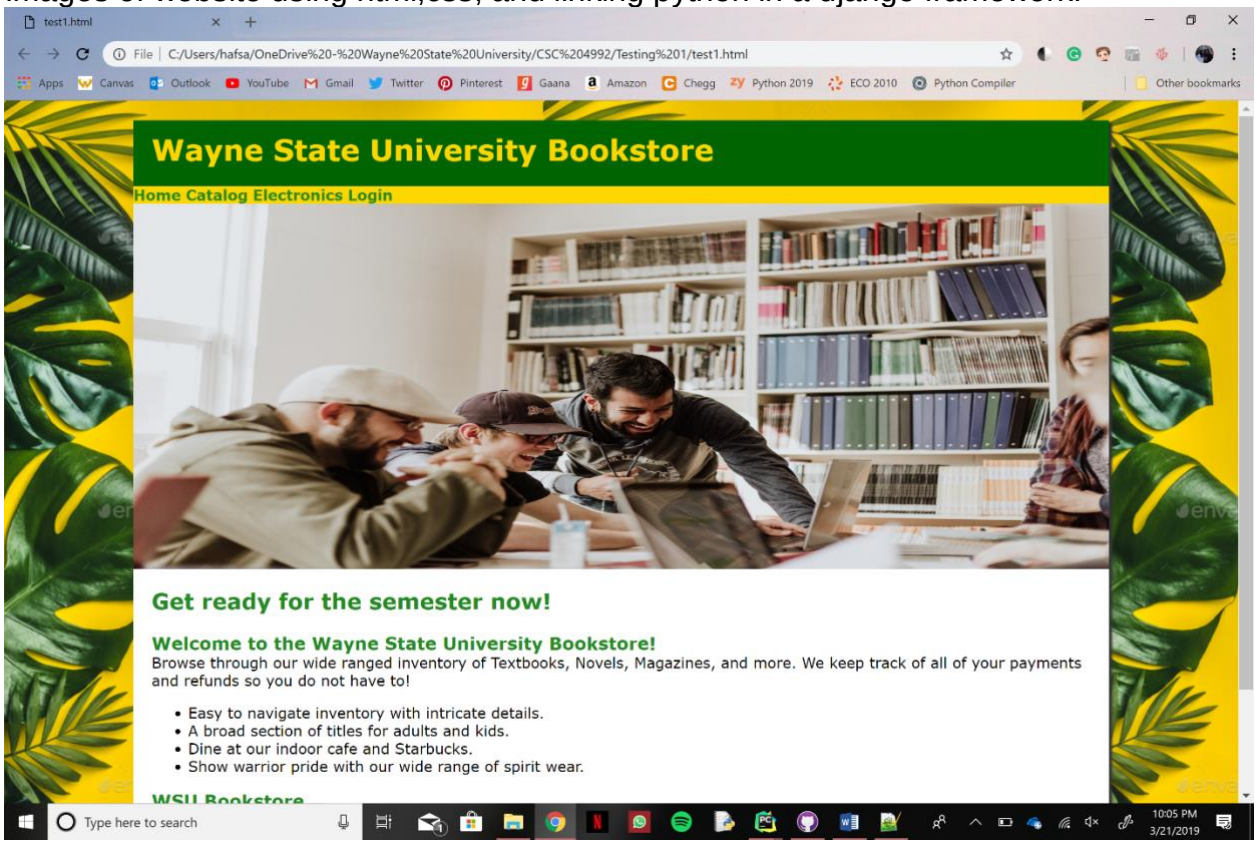


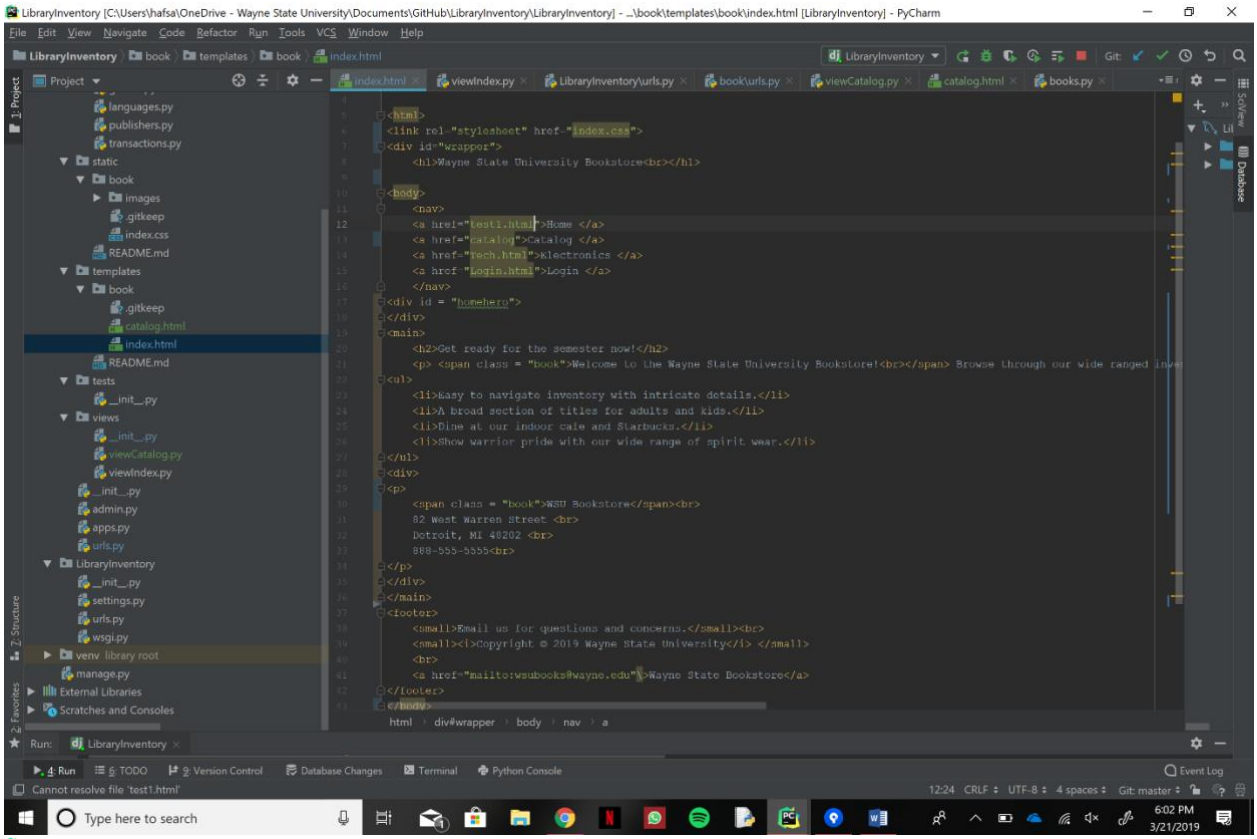
3) Commit screenshot: (Has not been updated for some reason)





Images of website using html,css, and linking python in a django framework.




LibraryInventory [C:\Users\hafsa\OneDrive - Wayne State University\Documents\GitHub\LibraryInventory\LibraryInventory] - ...\book\templates\book\index.html [LibraryInventory] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LibraryInventory book templates book index.html
Project
languages.py
publishers.py
transactions.py
static
book
images
.gitkeep
index.css
README.md
templates
book
.gitkeep
catalog.html
index.html
README.md
tests
__init__.py
views
__init__.py
viewCatalog.py
viewIndex.py
__init__.py
admin.py
apps.py
urls.py
LibraryInventory
__init__.py
settings.py
urls.py
wsgi.py
venv library root
manage.py
External Libraries
Scratches and Consoles
index.html
viewIndex.py
LibraryInventory\urls.py
book\urls.py
viewCatalog.py
catalog.html
books.py
<html>
<link rel="stylesheet" href="index.css">

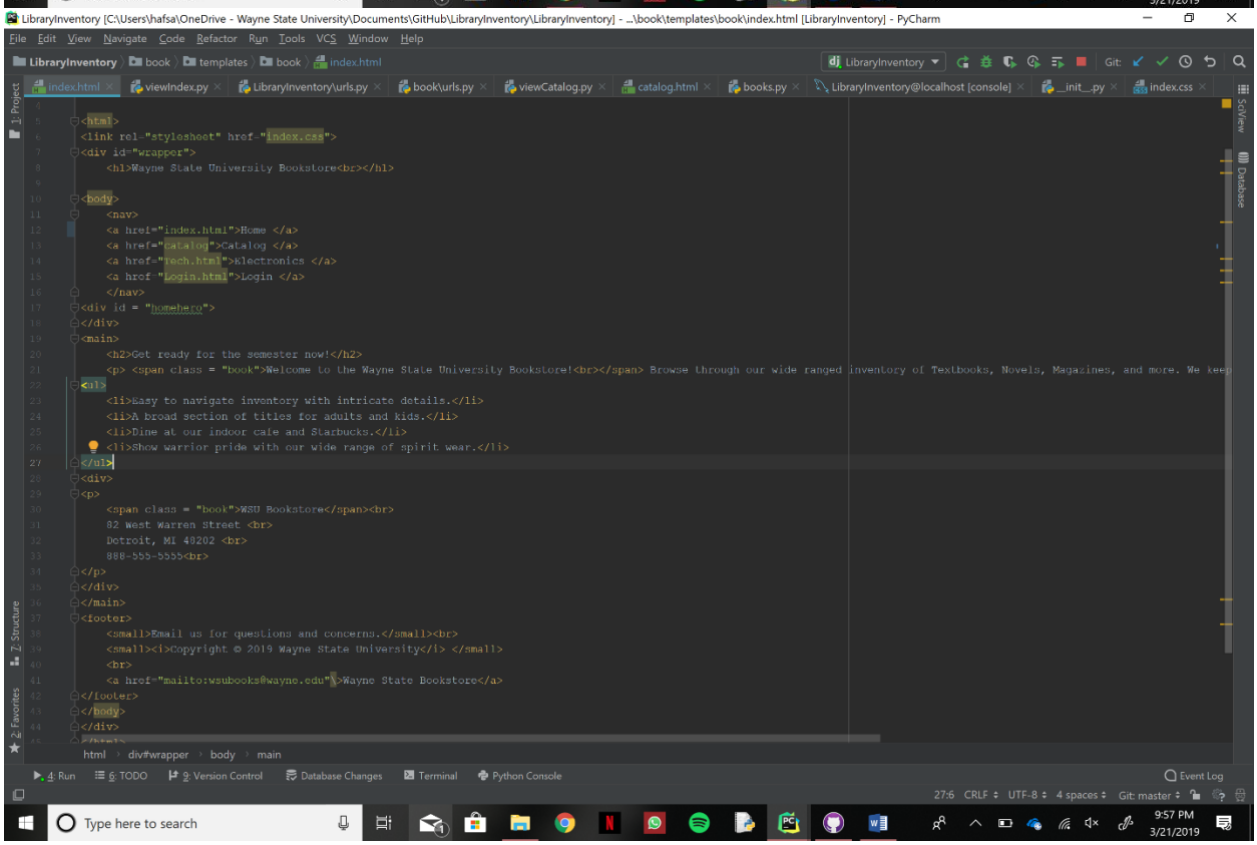
<h1>Wayne State University Bookstore
</h1>
<body>
<nav>
<a href="test1.html">Home </a>
<a href="catalog">Catalog </a>
<a href="tech.html">Electronics </a>
<a href "Login.html">Login </a>
</nav>

<main>
<h2>Get ready for the semester now!</h2>
 Welcome to the Wayne State University Bookstore!
 Browse through our wide ranged inv
<ul>
<li>Easy to navigate inventory with intricate details.</li>
<li>A broad section of titles for adults and kids.</li>
<li>Dine at our indoor cafe and Starbucks.</li>
<li>Show warrior pride with our wide range of spirit wear.</li>
</ul>

WSU Bookstore
82 West Warren Street
Detroit, MI 48202
888-555-5555

</main>
<footer>
<small>Email us for questions and concerns.</small>
<small>Copyright © 2019 Wayne State University </small>

<a href="mailto:wsubooks@wayne.edu">Wayne State Bookstore</a>
</footer>
</body>
html div#wrapper body nav a
Run: LibraryInventory
4: Run 6: TODO 9: Version Control Database Changes Terminal Python Console
Event Log
Cannot resolve file 'test1.html'
12:24 CRLF UTF-8 4 spaces Git: master
Type here to search
6:02 PM
3/21/2019


LibraryInventory [C:\Users\hafsa\OneDrive - Wayne State University\Documents\GitHub\LibraryInventory\LibraryInventory] - ...\book\templates\book\index.html [LibraryInventory] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LibraryInventory book templates book index.html
index.html
viewIndex.py
LibraryInventory\urls.py
book\urls.py
viewCatalog.py
catalog.html
books.py
LibraryInventory@localhost [console]
__init__.py
index.css
<html>
<link rel="stylesheet" href="index.css">

<h1>Wayne State University Bookstore
</h1>
<body>
<nav>
<a href="index.html">Home </a>
<a href="catalog">Catalog </a>
<a href="tech.html">Electronics </a>
<a href "Login.html">Login </a>
</nav>

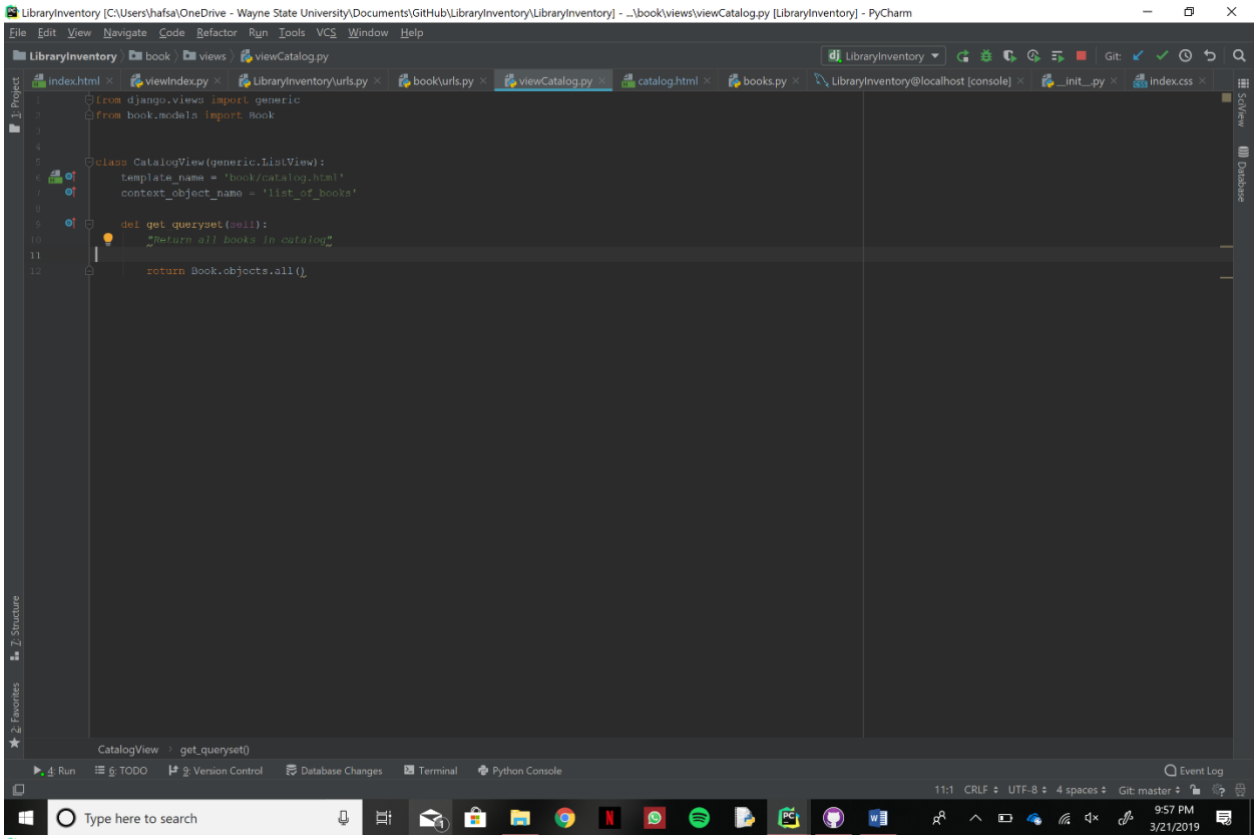
<main>
<h2>Get ready for the semester now!</h2>
 Welcome to the Wayne State University Bookstore!
 Browse through our wide ranged inventory of Textbooks, Novels, Magazines, and more. We keep
<ul>
<li>Easy to navigate inventory with intricate details.</li>
<li>A broad section of titles for adults and kids.</li>
<li>Dine at our indoor cafe and Starbucks.</li>
<li>Show warrior pride with our wide range of spirit wear.</li>
</ul>

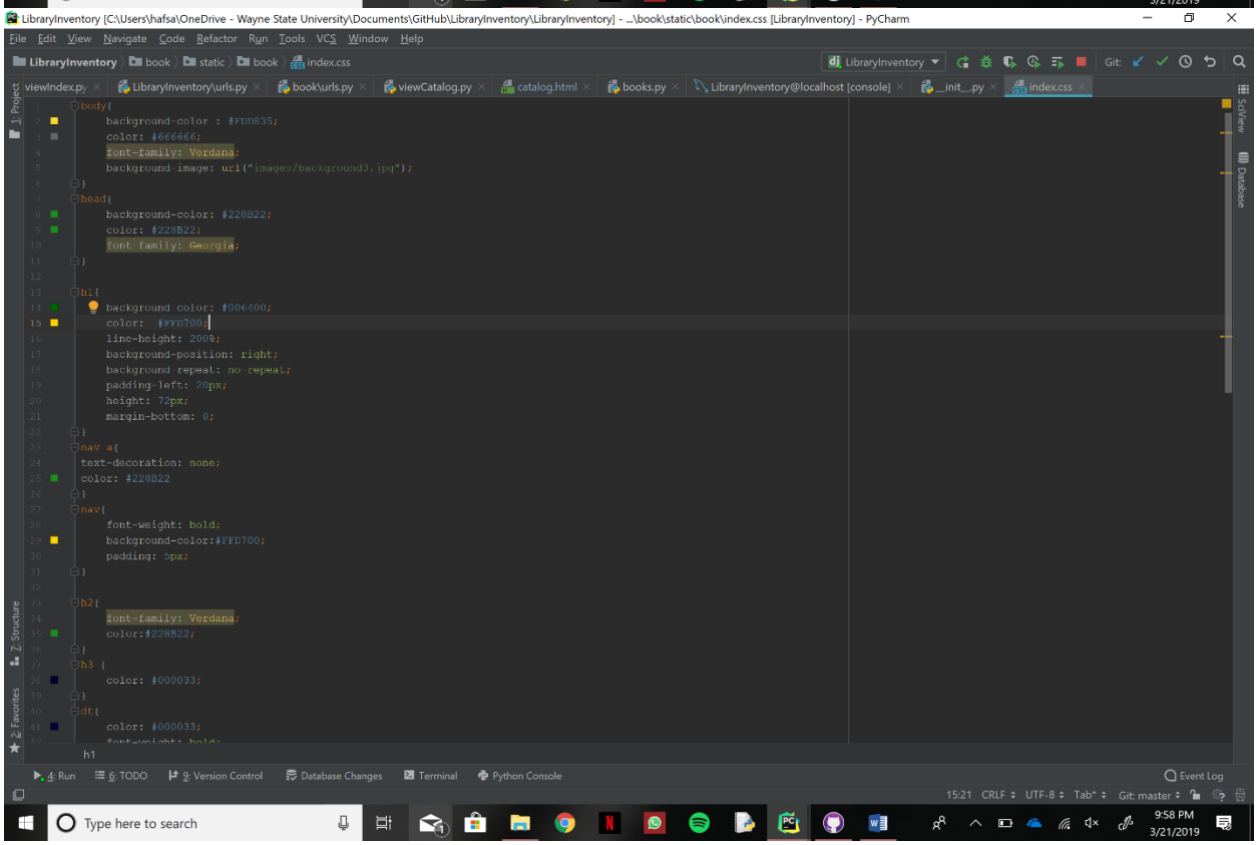
WSU Bookstore
82 West Warren Street
Detroit, MI 48202
888-555-5555

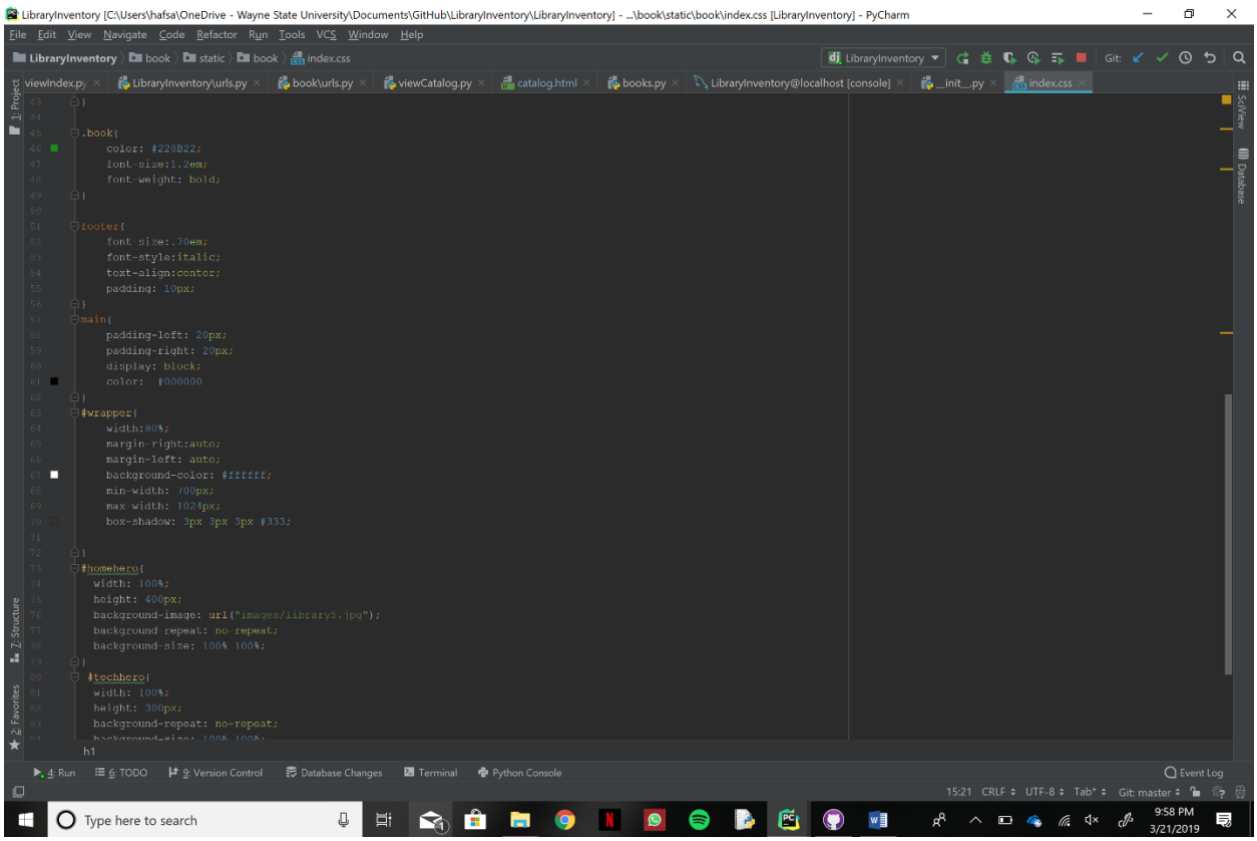
</main>
<footer>
<small>Email us for questions and concerns.</small>
<small>Copyright © 2019 Wayne State University </small>

<a href="mailto:wsubooks@wayne.edu">Wayne State Bookstore</a>
</footer>
</body>

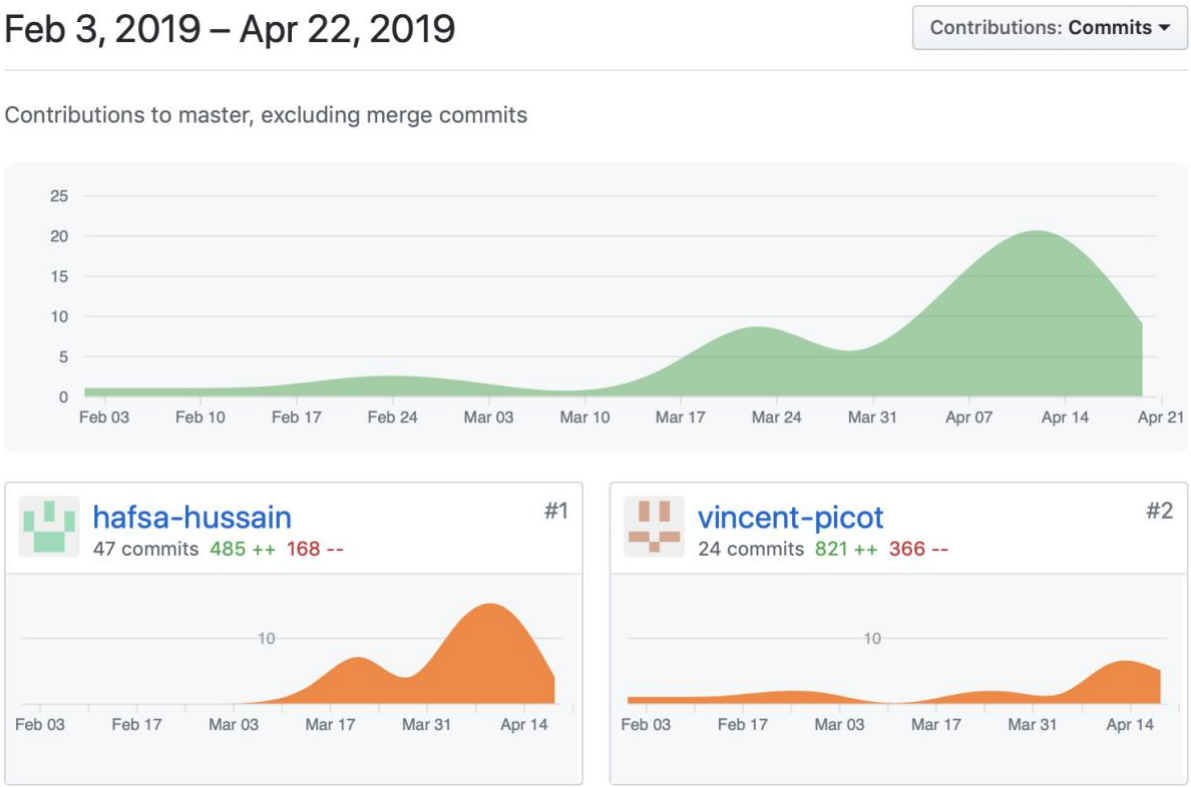
html div#wrapper body main
4: Run 6: TODO 9: Version Control Database Changes Terminal Python Console
Event Log
27:6 CRLF UTF-8 4 spaces Git: master
Type here to search
9:57 PM
3/21/2019

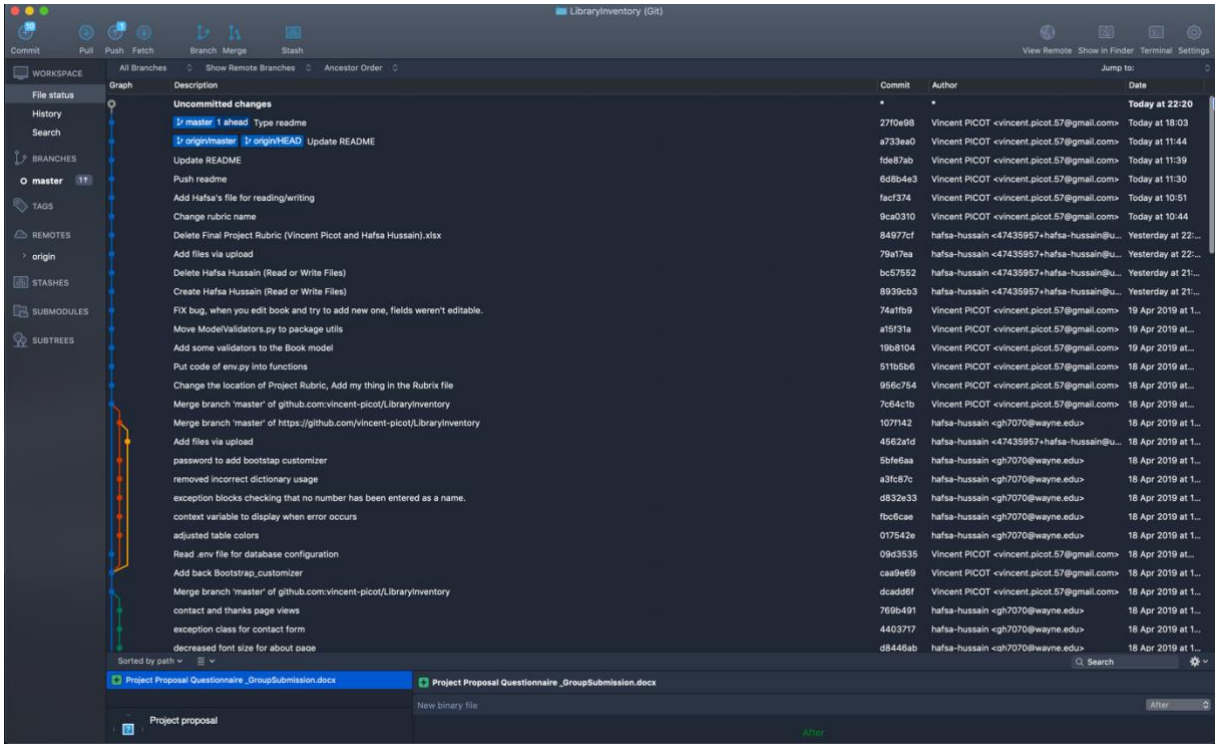

LibraryInventory [C:\Users\hafsa\OneDrive - Wayne State University\Documents\GitHub\LibraryInventory\LibraryInventory] - ...\book\views\viewCatalog.py [LibraryInventory] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LibraryInventory book views viewCatalog.py
index.html viewIndex.py LibraryInventory\urls.py book\urls.py viewCatalog.py catalog.html books.py LibraryInventory@localhost [console] __init__.py index.css
from django.views import generic
from book.models import Book
class CatalogView(generic.ListView):
template_name = 'book/catalog.html'
context_object_name = 'list_of_books'
def get_queryset(self):
"Return all books in catalog"
return Book.objects.all()
CatalogView get_queryset()
4: Run 6: TODO 9: Version Control Database Changes Terminal Python Console Event Log
11:1 CRLF UTF-8 4 spaces Git: master
Type here to search
9:57 PM
3/21/2019

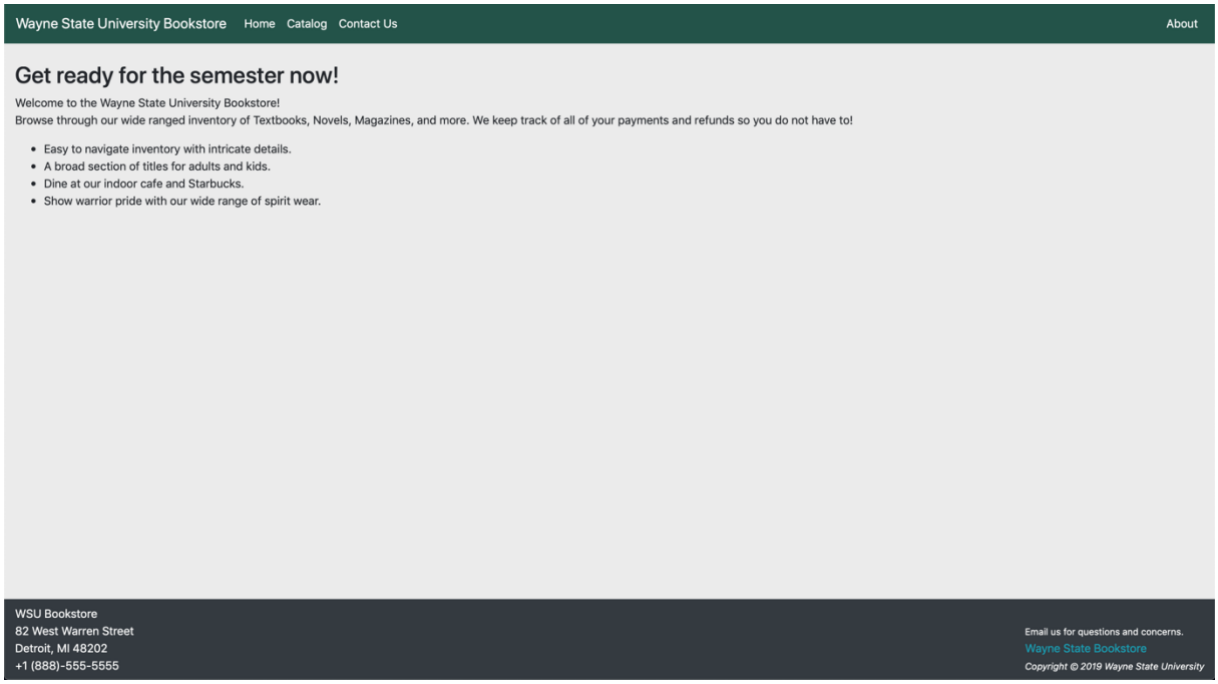

LibraryInventory [C:\Users\hafsa\OneDrive - Wayne State University\Documents\GitHub\LibraryInventory\LibraryInventory] - ...\book\static\book\index.css [LibraryInventory] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
LibraryInventory book static book index.css
viewIndex.py LibraryInventory\urls.py book\urls.py viewCatalog.py catalog.html books.py LibraryInventory@localhost [console] __init__.py index.css
body{
background-color : #FDD835;
color: #666666;
font-family: Verdana;
background-image: url("images/background3.jpg");
}
head{
background-color: #228B22;
color: #228B22;
font family: Georgia;
}
h1{
background-color: #006400;
color: #FFD700;
line-height: 200%;
background-position: right;
background-repeat: no-repeat;
padding-left: 20px;
height: 72px;
margin-bottom: 0;
}
nav a{
text-decoration: none;
color: #228B22
}
nav{
font-weight: bold;
background-color:#FFD700;
padding: 5px;
}
h2{
font-family: Verdana;
color:#228B22;
}
h3 {
color: #000033;
}
dt{
color: #000033;
h1
4: Run 6: TODO 9: Version Control Database Changes Terminal Python Console Event Log
15:21 CRLF UTF-8 Tab* Git: master
Type here to search
9:58 PM
3/21/2019



Update: Bellow are screenshot of the updated part from the above screenshots.




LibraryInventory (Git)
Commit
Pull
Push
Fetch
Branch
Merge
Stash
View Remote
Show in Finder
Terminal
Settings
WORKSPACE
File status
History
Search
BRANCHES
master
TAGS
REMOTES
origin
STASHES
SUBMODULES
SUBTREES
All Branches
Show Remote Branches
Ancestor Order
Jump to:
Graph
Description
Commit
Author
Date
Uncommitted changes
master 1 ahead Type readme
origin/master origin/HEAD Update README
Update README
Push readme
Add Hafsa's file for reading/writing
Change rubric name
Delete Final Project Rubric (Vincent Picot and Hafsa Hussain).xlsx
Add files via upload
Delete Hafsa Hussain (Read or Write Files)
Create Hafsa Hussain (Read or Write Files)
FIX bug, when you edit book and try to add new one, fields weren't editable.
Move ModelValidators.py to package utils
Add some validators to the Book model
Put code of env.py into functions
Change the location of Project Rubric, Add my thing in the Rubrix file
Merge branch 'master' of github.com:vincent-picot/LibraryInventory
Merge branch 'master' of https://github.com/vincent-picot/LibraryInventory
Add files via upload
password to add bootstap customizer
removed incorrect dictionary usage
exception blocks checking that no number has been entered as a name.
context variable to display when error occurs
adjusted table colors
Read .env file for database configuration
Add back Bootstrap_customizer
Merge branch 'master' of github.com:vincent-picot/LibraryInventory
contact and thanks page views
exception class for contact form
decreased font size for about page
Sorted by path
Search
Project Proposal Questionnaire _GroupSubmission.docx
Project Proposal Questionnaire _GroupSubmission.docx
Project proposal
New binary file
After
After


Wayne State University Bookstore
Home
Catalog
Contact Us
About
Get ready for the semester now!
Welcome to the Wayne State University Bookstore!
Browse through our wide ranged inventory of Textbooks, Novels, Magazines, and more. We keep track of all of your payments and refunds so you do not have to!
Easy to navigate inventory with intricate details.
A broad section of titles for adults and kids.
Dine at our indoor cafe and Starbucks.
Show warrior pride with our wide range of spirit wear.
WSU Bookstore
82 West Warren Street
Detroit, MI 48202
+1 (888)-555-5555
Email us for questions and concerns.
Wayne State Bookstore
Copyright © 2019 Wayne State University

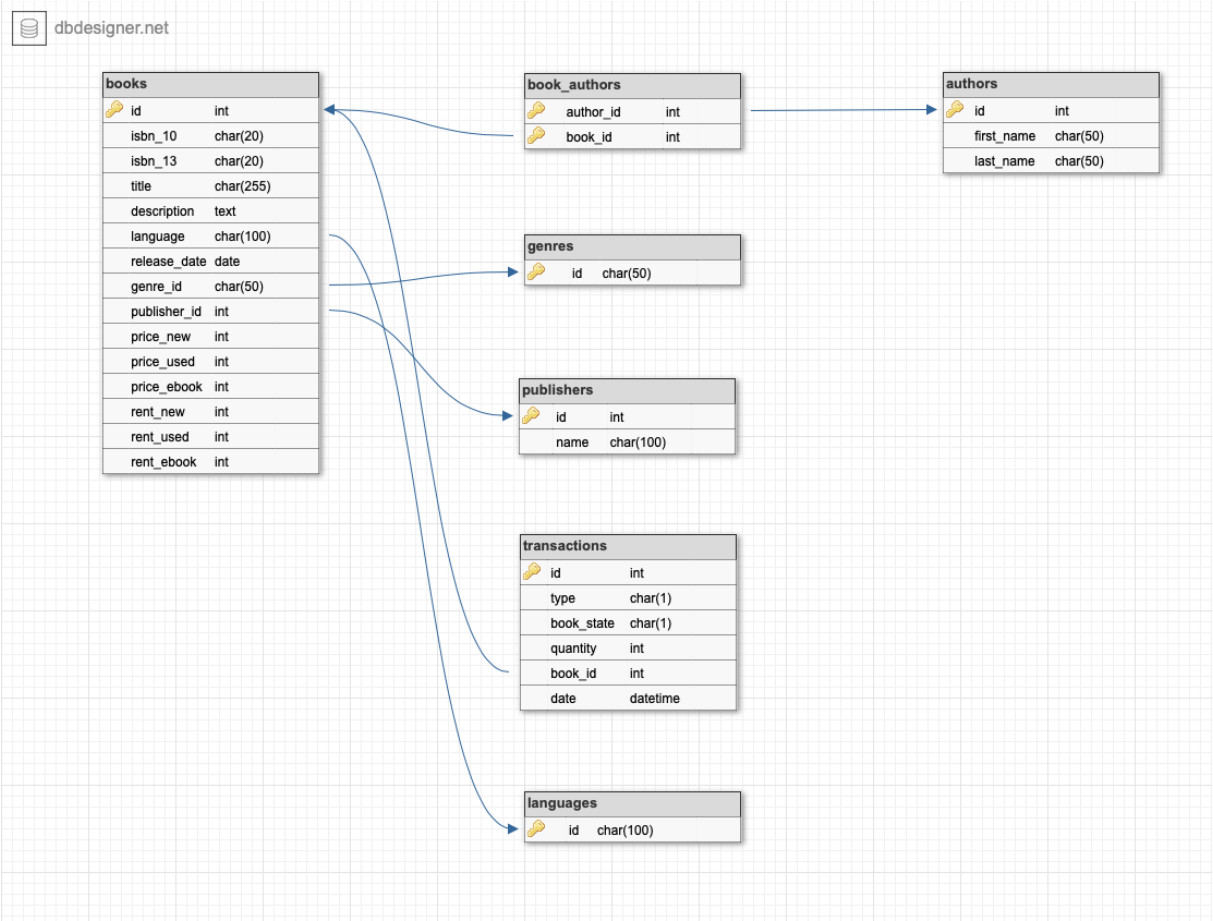
4)

- Because there are only two of us now, we do not make much use of the Gantt chart or the Kanban board, and rather meet frequently to keep each other updated as well as work together.

- We both work together to brainstorm ideas on how to better our project as well as do the actual coding portion. Originally, the work was split between 3 people but has now changed to 2 which is something we are in the process of adjusting to. Fortunately, we work well together and assist each other when needed.

Final architecture documentation:

Below is a screenshot of our database architecture:



Below is a schema of our project architecture:

LibraryInventory *(root of the project)*
|
|
| ---------- book *(Django app, our main project, also a python package)*
| | |
| | | ---------- admin *(Python package, use by Django, contains all the classes describing the admin site edit/add/delete/view pages of the models)*
| | |
| | | ---------- migration *(Python package auto generated by Django when a database modification is necessary)*
| | |
| | | ---------- models *(Python package used by Django to map the classes to tables in the database)*
| | |
| | | ---------- static *(Basic folder used by Django, contains all the static files for the user site (e.g. images, stylesheets, …))*
| | |
| | | ---------- templates *(Basic folder used by Django, contains all the HTML files)*
| | |
| | | ---------- tests *(Python package used by Django, contains all the unit tests of the book app)*
| | |
| | | ---------- utils *(Python package, contains all the utility functions/classes that the book app needs)*
| | |
| | | ---------- views *(Python package used by Django, contains all the user site views/page)*
| | |
| | | ---------- apps.py *(File auto-generated by Django)*
| | |
| | | ---------- url.py *(File generated by Django, map every url of the user site to a view in the views package)*
|
|
|
| ---------- LibraryInventory *(Django app auto-generated with the project, it is the admin site and the root handling all the other Django apps in the project)*
|
|
| ---------- templates *(Basic folder, contains the HTML files to override the basic Django admin site layout)*
|
|
| ---------- utils *(Python package, contains all the utility functions/classes that the whole project apps and admin site would need)*

If you need more information about a specific package or module inside a package, you can consult directly the specific package/module you are interested in. Every module/class/method/attribute is commented to explain its purpose, where and how it’s used.