REPORT
DJANGO PROJECT
E-LIBRARY

GITHUB-LINK: https://github.com/paolakaz-dev/library

PAULINA KAZMIERCZAK

3rd SEMESTER

WEB DEVELOPMENT

TABLE OF CONTENT

1. INTRODUCTION

The task is to create the e-library fully functioning system. The objective will include the admin/ user platform where they can manage their account and make actions assigned to the role.

The concept is to create platform created only for students. Everyone can sign up, but only the user who will provide the student information such as (branch, contact, picture) will have a possibility to check out the items. Registration of Student is made by the Admin.

2. PROCESS

a. Define the user / admin views

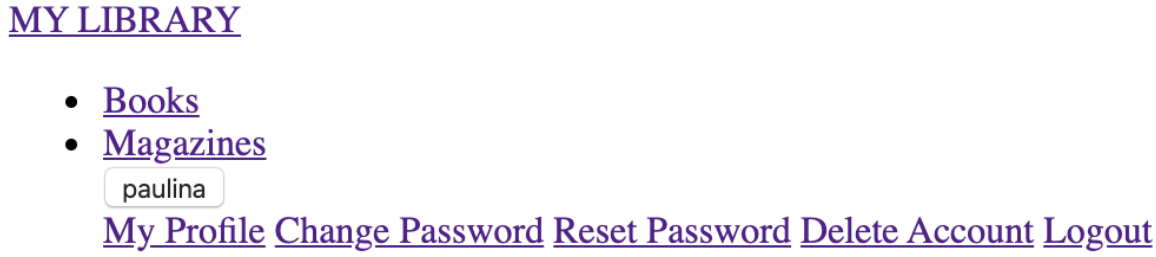
There are two kinds of users of the system. Staff and customers, who can check out books and magazines. In order to distinguish the role which is defined by the possibility of making specific action by creating two dashboards. One for the stuff = superuser, who has power to update/insert/delete and manipulate entire data and the user, who is able to manage its own account and check out/in items in e-library.
The dashboards got defined in view.ps by creating the if statements as well as in the html files by diving the actions by the roles.

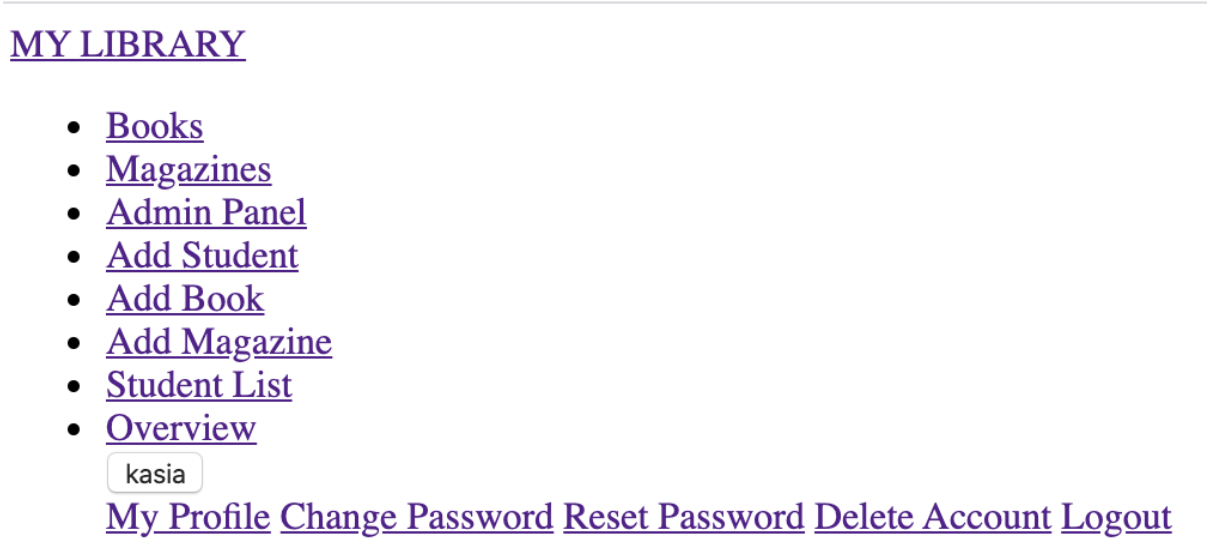
```
@login_required
def MagazineCreate(request):
    if not request.user.is_superuser:
        return redirect('index')
```

```
{% if user.is_superuser %}
    <li class="nav-item">
        <a class="nav-link"
    </li>
```

b. Overview of admin / user dashboards

Here is the created systems for admin as well as for user based on the a requirement specification for each role.





c. Student creation

As I mentioned in the introduction, the concept is that e-library is for students which had to get confirmed by the stuff.

```
# STUDENT CREATION
class Student(models.Model):
    roll_no = models.CharField(max_length=10,unique=True)
    name = models.CharField(max_length=10)
    branch = models.CharField(max_length=3)
    contact_no = models.CharField(max_length=10)
    total_books_due=models.IntegerField(default=0)
    email=models.EmailField(unique=True)
    pic=models.ImageField(blank=True, upload_to='profile_image')
    def __str__(self):
        return str(self.roll_no)
```

Here I have specified the views – only for students. If the name of the student = with the surname of the username. The user get the access to the all functions which are assigned for the student.

```
student=Student.objects.get(name=request.User)
```

d. Borrower creation

Of course, that's possible to be a student and not check out/in items, so there is another model called *borrower* – which has a relation with student and wanted book.

```python
# BOOK BORROWER CREATION
class Borrower(models.Model):
    student = models.ForeignKey('Student', on_delete=models.CASCADE)
    book = models.ForeignKey('Book', on_delete=models.CASCADE, help_text= "Yo
    issue_date = models.DateTimeField( null=True,blank=True)
    return_date = models.DateTimeField( null=True,blank=True)
    def __str__(self):
        return self.student.name+" borrowed "+self.book.title
```

3. IMPROVMENTS

The *to do* list, which is an overview of future work tasks in order to accomplish the goal, which is to build fully functioning e-library system meeting the given requirements.

- check out magazines – process,
- add the form with rules to the borrow-book action,
- create the overview list of outstanding checkouts instead of normal overview,
- user view / admin personal view – ability to check the current overview / check out / return functions
- pleasant user interface – adding style

4. CONCLUSION

The project allowed me to practice the knowledge gained in the class as well as forced me to make a deeper research about Django and the way how the app should be constructed. Anyway, is not always easy to handle and find the mistakes, but the good documentation as well as bunch of another developers who made the same mistake and documented that on the forums, helps to solve it minutes by digging in the Internet.

APPENDIX

LOGIN INFORMATION

superuser login

username: paulina

password: kazmierczak

user login

username: kasia

password: kazmierczak