Tecnologias e Programação Web

Helder Zagalo

Game store
Web Application

Rui Fernandes, 92952
Leandro Silva, 93446

DETI
Universidade de Aveiro

January 20, 2021

1 Introduction

This report will succinctly describe our implementation of the second project of TPW. Namely it will cover the structure of the implementation, main features available in our app and information on how to access it both remotely and locally.

2 Technologies Used

As required, our project has a Components-Based Web Development approach, the frontend being implemented using the Angular framework and the backend that handles Web services was done with Django REST framework.

We made good use of what we learned in the classes and, with some research, implemented other more advanced technologies, the most noteworthy being Authentication on both ends of the app.

3 Our App

Our objective with our App was to make a small and simple online shop for resale of games and consoles either new or used. Any registered user could log in and put a game up for sale as well as buy other user's items, in contrast a non registered user can still browse the shop but cannot buy or sell anything. It's important to note that a user puts an Article for sale which can in turn contain one or more Items (Games or consoles).

To better summarize what is possible to do in the App we'll list the relevant implemented functionalities.

3.1 Functionalities

- Fully implemented Sign up and login
- The landing page features a simplified version of the store, displaying the most popular games by number of views and some basic filters.
- The **store** tab in the navbar will lead us to the the full store, one can also click on the **games** or **consoles** sub tabs to search only for a certain type of item.
- At the store page one can see all the articles and experiment around with the various filters that work all in conjunction
- By clicking on an article, the **article's details** page opens and we can see various information on the article. By clicking on the User's Reviews button we can see the reviews left by various users to the user selling the article, as well as leave our own review. Two other important buttons are the **Add to Cart** and **Add to Saved** that do exactly as said, as well as redirect to the relevant page.
- The **Shop Cart** tab in the navbar will lead us to our shopping cart where we can see all the items on our cart as well as the total cost and fees of the order. We remove any of them by clicking on the X next to them, and of course place our order to checkout.
- The **Saved** tab in the navbar will display all the articles we have saved to see later, here we can also apply some filters to help search.
- The **My Articles** tab in the navbar will in turn display all the articles we have currently on sale, the ones we already sold, and the ones that we bought.
- On the **Account** tab (or the sub tab **Profile**) we can see our profile and all our information, and by clicking edit profile change our picture and biography.
- On the **Sell Article** sub tab we can create an article for sale through a multi-part form where we can add/edit/delete games or consoles, and then give the article's details to post it on the store. Various verifications are in place, such as not allowing for an article to have no items.
- Finally we can log out of our account with the Log Out sub tab.

4 How to Run our App

In any case, it is heavily recommended that our app is run using mozilla firefox browser, and for ease of exploration you can use the already registered user: username:Manel123, pass=adorojogos.

We deployed our app as suggested to heroku for the front end, and to pythonanywhere for the backend. Sadly however we had some bugs in deployment and it isn't possible to add Articles/Items because of an error with image uploading we could not resolve, in addition the drop down options (for example LogOut) seem to not always appear even in Firefox. So for a thorough exploration it needs to be run locally, we apologize for this inconvenience. To access it remotely use the following links:

- `https://gooodgames.herokuapp.com/`
- `http://leand12.pythonanywhere.com/`

For the local project some requirements are needed for both DRF and Angular.

DRF requirements, some are base from the work in classes but in any case:

- Django==3.1.2
- django-cors-headers==3.6.0
- django-filter==2.4.0
- djangorestframework==3.12.2
- djangorestframework-jwt==1.11.0
- django-allauth==0.44.0
- django-rest-auth==0.9.5

As for angular, the following additional libraries were used:

- $ npm install -s moment
- $ npm install -s jwt-decode
- $ npm install -s @types/jwt-decode