WEEKLY REPORT #2

Project
WEB scraper for online stores
Team Members
Nikita Ruchkin (n.ruchkin@innopolis.university)
GitHUB
https://github.com/reterman/test_project.git

PART 1: BRIEF DESCRIPTION OF THE PROJECT IDEA

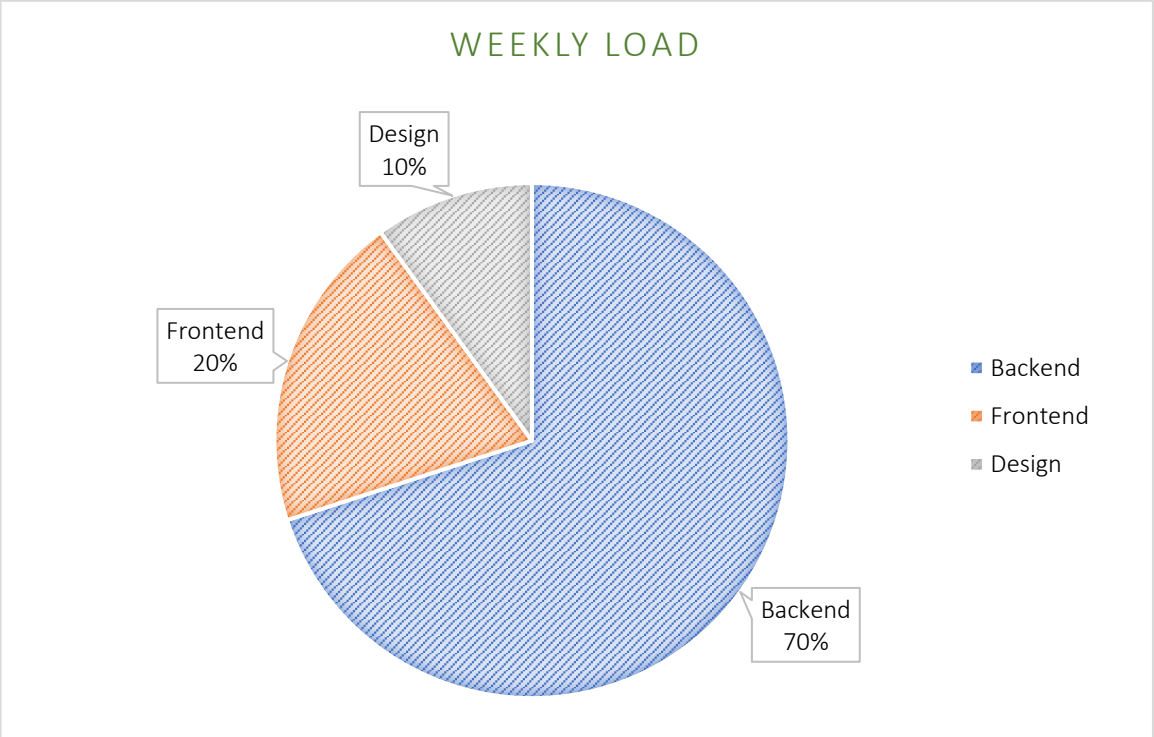
A price comparison tool that uses web scraping to gather pricing information from various online retailers for a specific product, and provides the user with a side-by-side comparison of the prices, allowing them to save money and make informed purchasing decisions. The tool can also provide notifications and alerts when the price of an item drops below a certain threshold.

PART 2: WEEKLY SPRINT

2.1. Workload distribution among team members

Since I am the only participant in my team, I performed every part of the project development on my own.

During the week, I spent a total of 5 hours on the second sprint.



2.2. Weekly tasks

#	Brief description	Hours spent
#1	Added a second marketplace from which the scraper searches for product data (Ozon.ru)	1

#2	Updated the method of transmitting product data to the site	0,5
#3	Added the ability to run both scrapers at the same time	1
#4	Made a check for the correctness of entering data into the search bar	0,5
#5	Improved site performance by saving the previous query	0,5
#6	Finalized the website design	1,5

2.3. Detailed description of the tasks

#1 Added a second marketplace from which the scraper searches for product data (Ozon.ru)

Through the study and analysis of dynamic css-classes, I discovered a constant pattern in their name. Thanks to this vulnerability, I was able to compose an expression by which I managed to get the necessary data about each product with Ozon.ru

#2 Updated the method of transmitting product data to the site

The early implementation of the data collection system by writing them in a notebook did not justify itself with its complexity. As a result, for a more convenient way of combining data and further manipulations with them, the idea was proposed to use lists.

#3 Added the ability to run both scrapers at the same time

For a more efficient and user-friendly way, it was decided to allocate an additional file in which the ability to allocate multiple threads to run all scrapers simultaneously was implemented. This method has improved performance and will allow you to spend less time on processing.

#4 Made a check for the correctness of entering data into the search bar

Using javascript syntax, I added a function to check the correctness of entering data into the search bar.

#5 Improved site performance by saving the previous query

In the course of the work, a vulnerability was discovered related to the re-launch of all search functions in the event of a page refresh or a return to the previous one (in cases of the user's desire to continue viewing the list of found products). Using flask-caching and session, I added a check for event data and, if any are detected, instead of running the scraper, I use cached data.

#6 Finalized the website design

Using the Bulma framework and my own css classes, I updated the design of the site, made it more user-friendly.

#7 Cleaning the code and adding comments

The usual cleaning of the code and providing clear comments for further convenience orientation in the code

PART 3: AN OBSTACLE THAT OCCURRED DURING THE SPRINT

#1 It was discovered that some popular marketplaces use captcha verification, which may become a serious problem in the future.

#2 There are quite a few effective ways to improve the speed of Seleniuma.

PART 4: THE MAIN TASKS FOR THE FOLLOWING SPRINTS

#1 To increase the number of used marketplaces.

#2 Improve the visual component of the site content.

#3 Optimize the work of selenium, to increase the speed of issuing product cards.

#4 The ability to search for products using sorting by price, rating, comments, etc.