MINISTRY OF EDUCATION AND SCIENCE OF THE KYRGYZ REPUBLIC

KYRGYZ-GERMAN INSTITUTE OF APPLIED INFORMATICS

COURSEWORK

Programming Languages 1

On the subject «Atlas shop» online market

Supervisor: R. Narmukhamedov (Senior Lecturer) \_\_\_\_\_\_\_\_

English Language Advisor: G. Jumalieva (Senior Lecturer) \_\_\_\_\_\_\_\_

Completed by: AIN-1-22 B. Asanbaev \_\_\_\_\_\_\_\_

A. Jusubalieva \_\_\_\_\_\_\_\_

A. Maksootova \_\_\_\_\_\_\_\_

Bishkek 2022

**Faculty of Informatics**

**Course Project**

**Subject’s information**

|  |  |
| --- | --- |
| Session | January 2023 |
| Program | First Semester |
| Lecturer | Radomir Narmukhamedov |
| Coursework Type | Project |
| Subject | Programming Language 1 (Python) |

**Information about students**

|  |  |  |
| --- | --- | --- |
| Full name |  | Group |
| Baitemir Asanbaev |  | AIN-1-22 |
| Altynai Jusubalieva |  | AIN-1-22 |
| Azima Maksutova |  | AIN-1-22 |

**Content**

[Introduction 4](#_Toc121462231)

[1 Requirements for the website 5](#_Toc121462232)

[2 Team Roles 6](#_Toc121462233)

[3 Architecture representation 7](#_Toc121462234)

[4 User’s guide 8](#_Toc121462235)

[4.1 Main page 8](#_Toc121462236)

[4.2 Catalog page 9](#_Toc121462237)

[4.3 Cart page 10](#_Toc121462238)

[4.4 About page 11](#_Toc121462239)

[4.5 Authentication page 10](#_Toc121462238)

[4.6 Profile page 11](#_Toc121462239)

[5 Special requirements 12](#_Toc121462240)

[6. Functionality 12](#_Toc121462241)

[7. Code Review 13](#_Toc121462242)

[7.1 Code Review in general 13](#_Toc121462243)

[7.2 Django Models 13](#_Toc121462244)

[7.3 Django Urls 14](#_Toc121462245)

[7.4 Django Views 15](#_Toc121462246)

[Conclusion 16](#_Toc121462247)

[References 16](#_Toc121462248)

# Introduction

The relevance of the topic. An online store is an application built using the technology of a modern stack. Like a regular store, an online store implements the following main functions: presentation of goods (services) to the buyer, order processing, sale and delivery of goods.

As a part of this project, requirements and limitations was analyzed. As a result, the Django framework was selected as a base solution of this project. The frontend part for this project is based on HTML, CSS and JavaScript. The database solution selected is SQLite.

The internal schema of the website and database was designed, according to the requirements. As for the front part, the website UI was designed according to the modern UI/UX design principles.

The result of the project is a fully working website with the frontend interface part and backend logic. Codes and schemas for both backend and frontend parts are included in the project documentation.

# 1 Requirements for the website

Atlas shop is internet market developed to let people order daily products from any place. All products are displayed on the catalog page and there are few filters functionality. Before client can order products, he need to sign in or sign up. Admins accounts have permission to create, update and delete products.

The website should contain following sections:

1. Main page
2. Catalog of products page
3. Cart page
4. Create, update, delete product (admin user only)
5. Feedback section
6. Sign in, sign up forms
7. About us page

The website data should be permanently saved in the database.

Interface should be intuitive, understandable and easy to use.

All components should be in united style and color palette.

User’s data should be safe and secure from intruders

# 2 Team Roles

The team for the project consists of 3 team members:

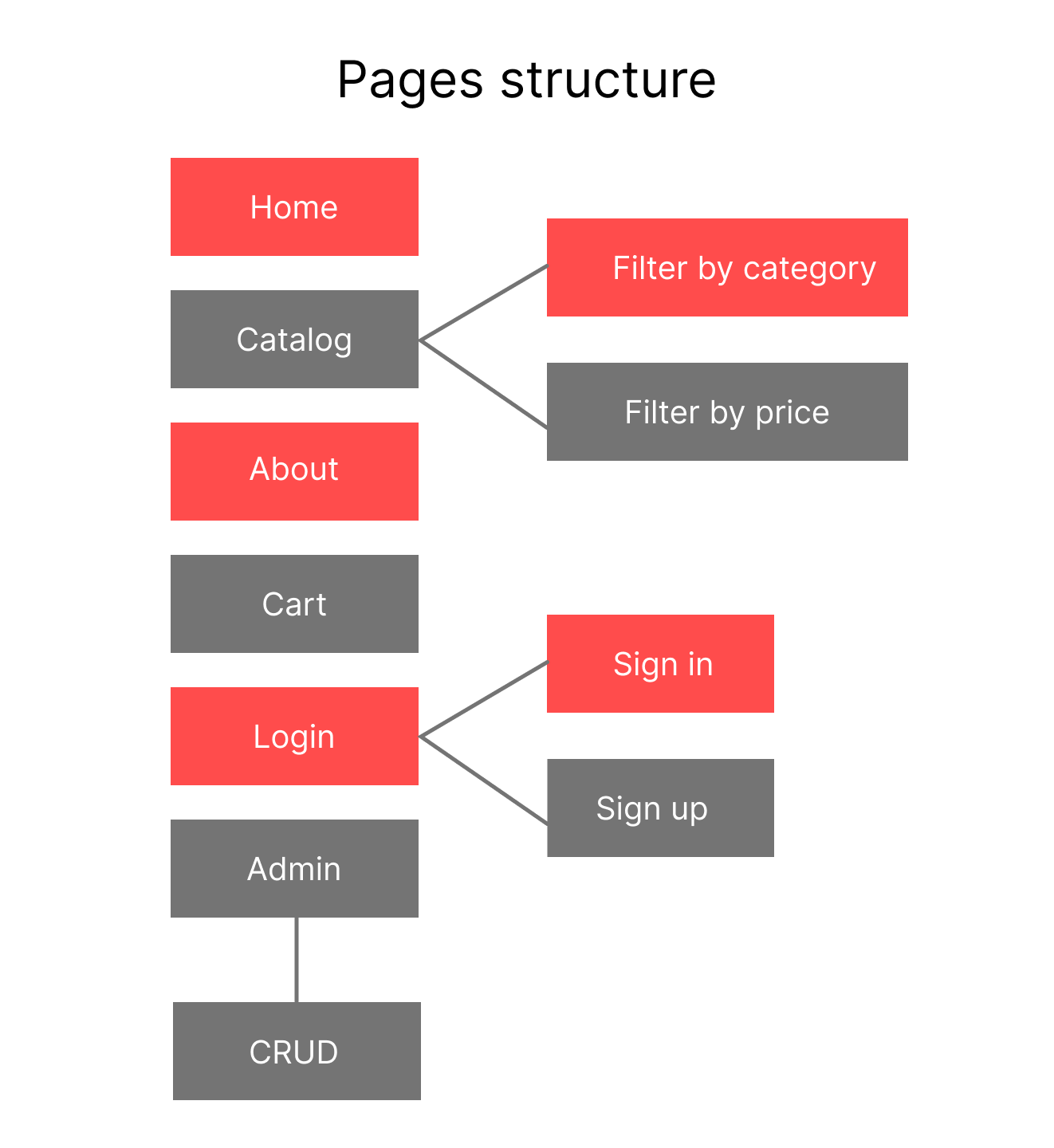
1. Baitemir Asanbaev
2. Altynai Jusubalieva
3. Azima Maksutova

The roles were distributed according to the table below:

|  |  |
| --- | --- |
| **Team member** | **Role** |
| Baitemir Asanbaev | Coach, Team lead, Full stack developer |
| Altynai Jusubalieva | Backend development |
| Vladislav Kan | Frontend development |

# 3 Architecture representation

The architecture of the pages is described on following picture:



# 4 User’s guide.

## Home page

## 

It contains following elements:

1. Navigation bar, with access to all other pages
2. Welcome header
3. Cart navigation section
4. Slider of clients feedback

## 

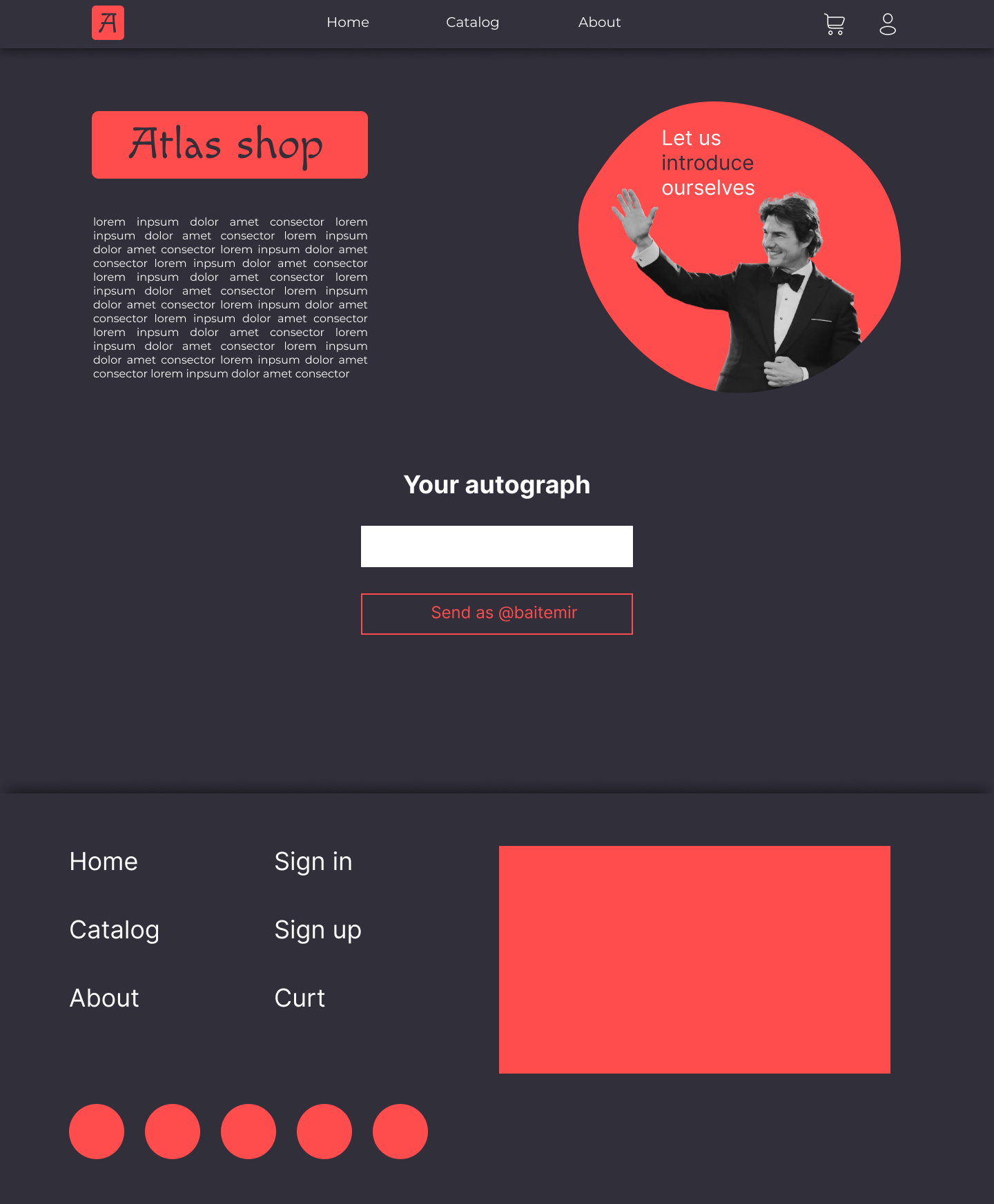
## Catalog page



It contains following elements:

1. Navigation bar, with access to all other pages
2. Greeting header
3. Filter by category and price inputs
4. A list of products with ability to add them to cart

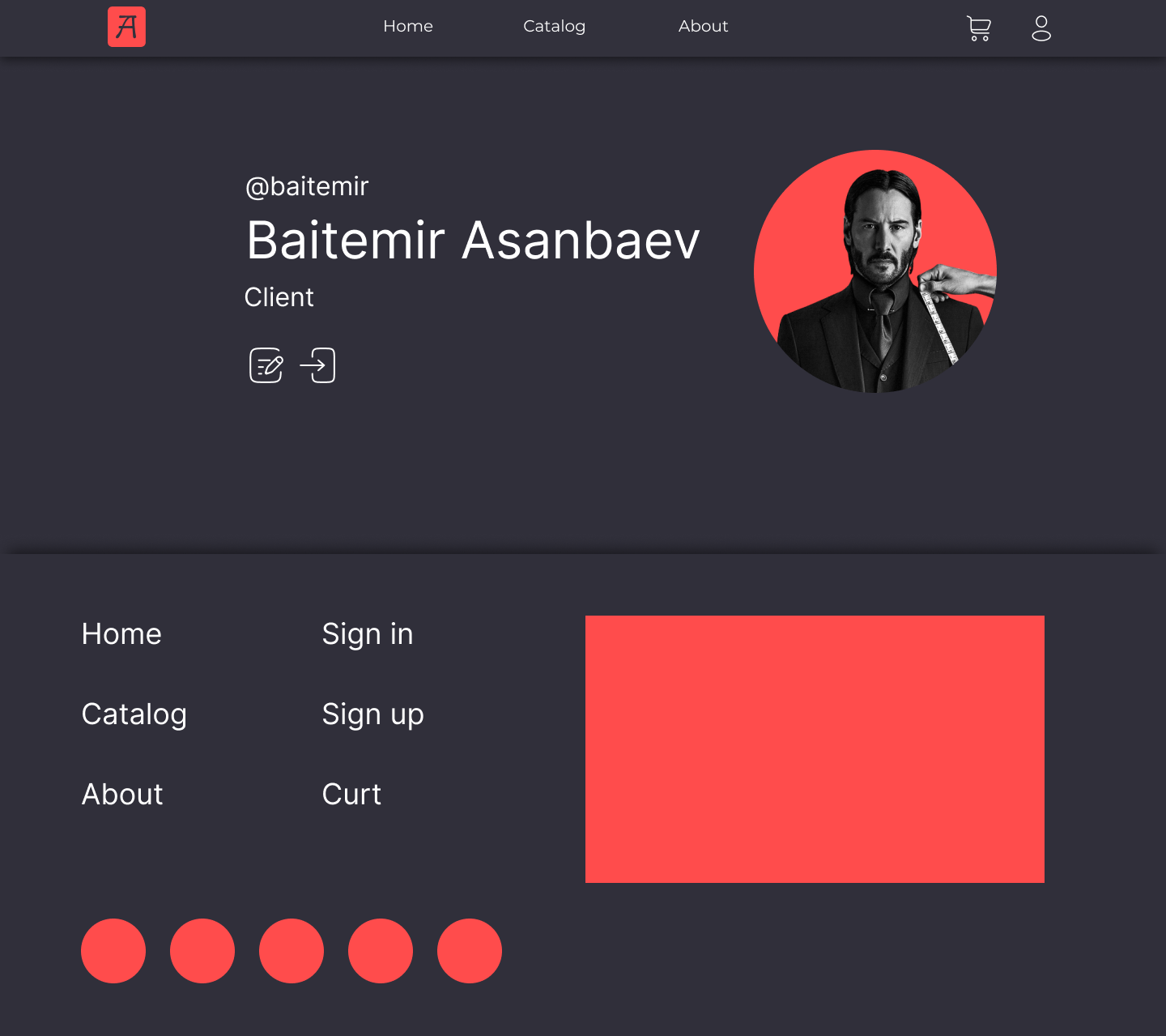
## About page



It contains following elements:

1. Navigation bar, with access to all other pages
2. Information about company
3. Form to leave comment which will be showed in home page (login required)

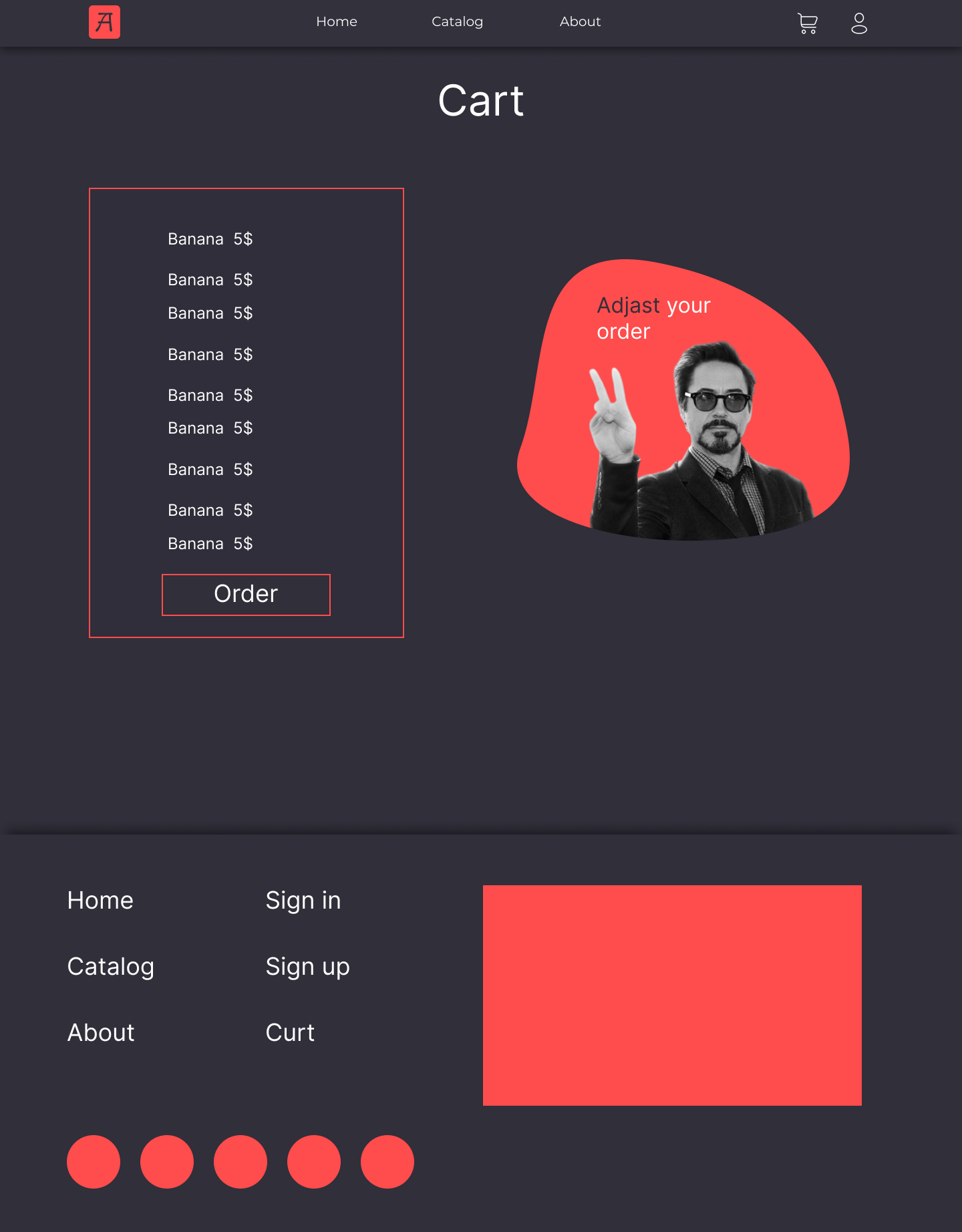
## Profile page



It contains following elements:

1. Navigation bar, with access to all other page
2. All user’s info, username, name, role.
3. Two buttons that represents edit profile and logout

## Cart page



It contains following elements:

1. Navigation bar, with access to all other page
2. Products added to cart
3. A button which finishes the order.

# 5 Special requirements

The website should run on Django, and it should use database.

The database should support SQL and the data should remain there for long time. The data should not erase when the application is stopped.

# 6. Functionality

The website should provide following pages:

* Main page
* Catalog page
* Profile page
* About page
* Log in page
* Cart page
* Add product page (for admin users only)

# Code Review

Source code is posted on GitHub, link is in References section. There will be reviewed essential part of project.

Backend part was written with Django, so code is in python programming language.

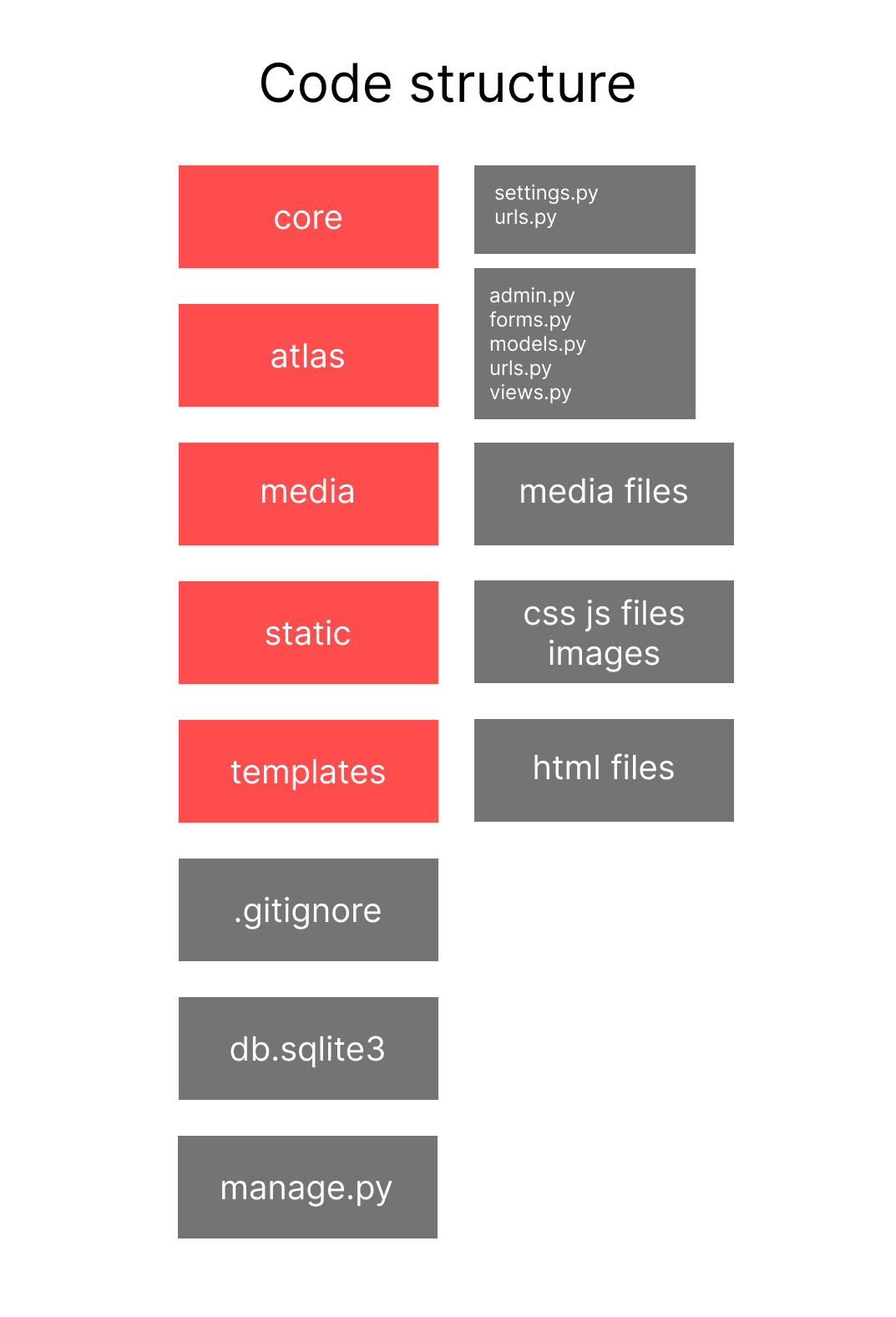
Frontend part was written with HTML CSS and JS languages. HTML files are stored in templates folder and CSS and JS files are stored in static folder.

## Stack

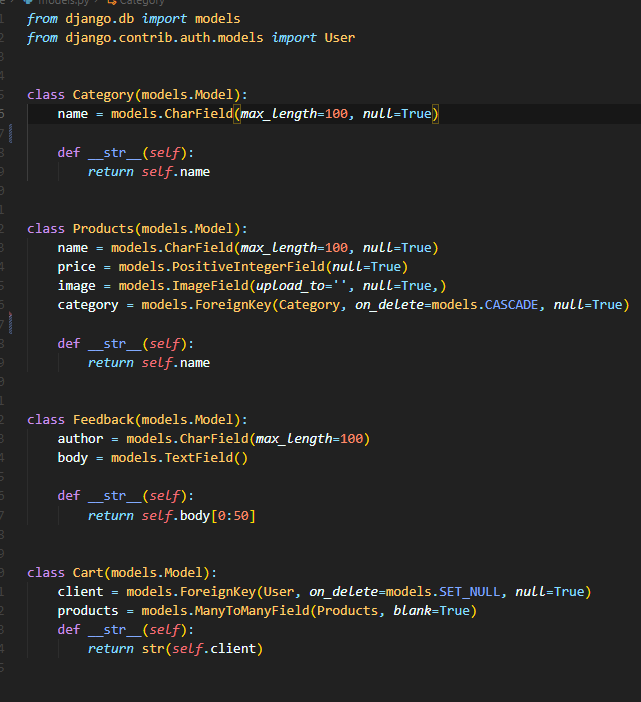


* GitHub
* VSCode
* PyCharm
* HTML5
* CSS3
* JavaScript
* Python
* Django
* SQLite3

## Code structure

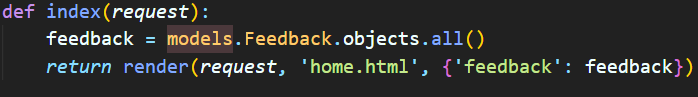


## Models

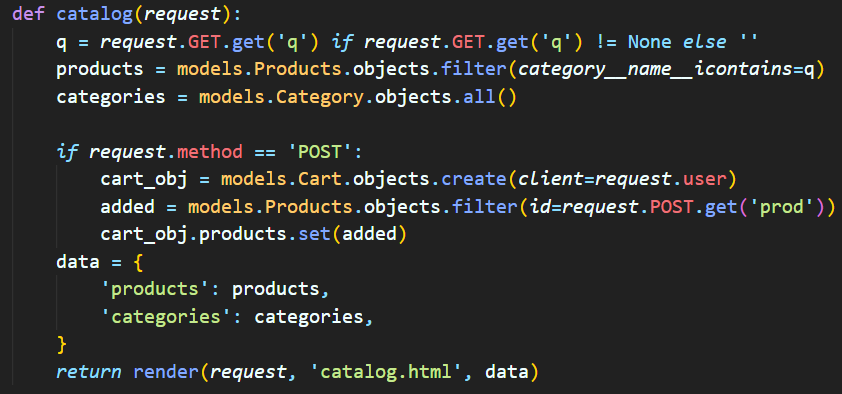


There are 4 custom models and 1 default model (User). Each product connected with Category via foreign key. Each product added to cart also connected with user via foreign key, so only current user could see what’s in his cart. Feedback model represents comments client leave.

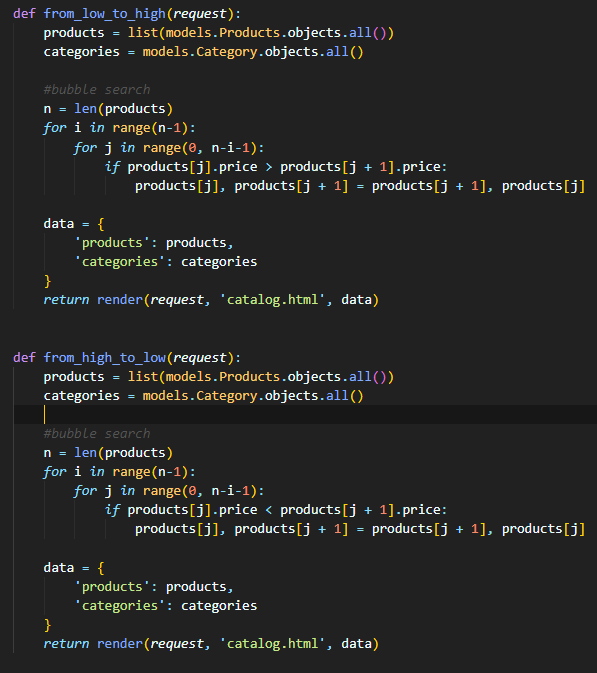
## Views



Index view gets data from models and renders home page via html



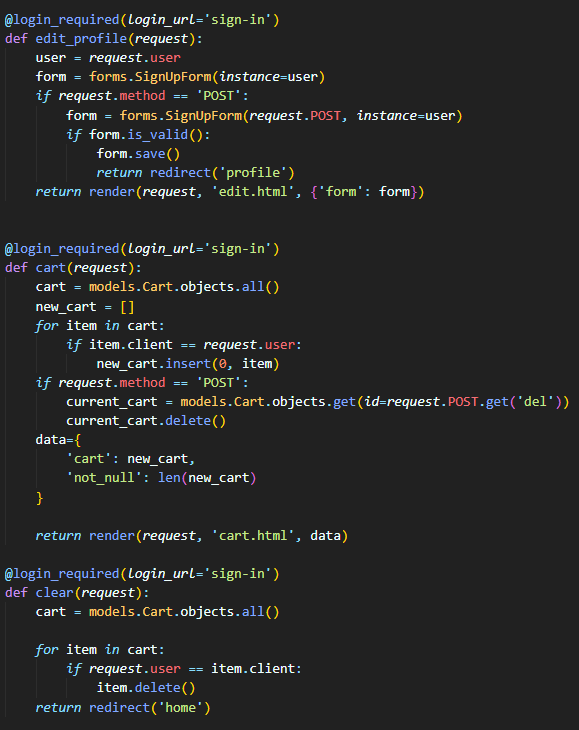
Catalog view get query parameter from address bar and implements search by category. And when client wants to add product to cart, POST method triggers and an object added to Cart.



Search views: from high to low, from low to high. When one of them triggers, this function sorts list of products via bubble sort algorithm and returns sorted list.



Authorization views. If user already has an account, he can sign in with username and password. Else he has to create a new one (sign up).



Profile and cart manipulation views are not available if user is not authenticated.

# Conclusion

The aim of this project was to find an ultimate solution for daily product shopping problem. And it turned out to be online shop web site. It has simple design, intuitive interface, working functionality and optimized code. With this site people’s live become a bit easier.

# References

1. Source code <https://github.com/BaitemirAsanbaev/kursovaya>
2. Figma layout [https://www.figma.com](https://www.figma.com/file/3NPKt4PGa3LhqbshYfINOa/Untitled?node-id=0%3A1&t=f7tnTnprovrOUOqn-1)
3. Django documentation <https://docs.djangoproject.com/en/4.1/>
4. YouTube Django tutorial <https://www.youtube.com/watch?v=PtQiiknWUcI>