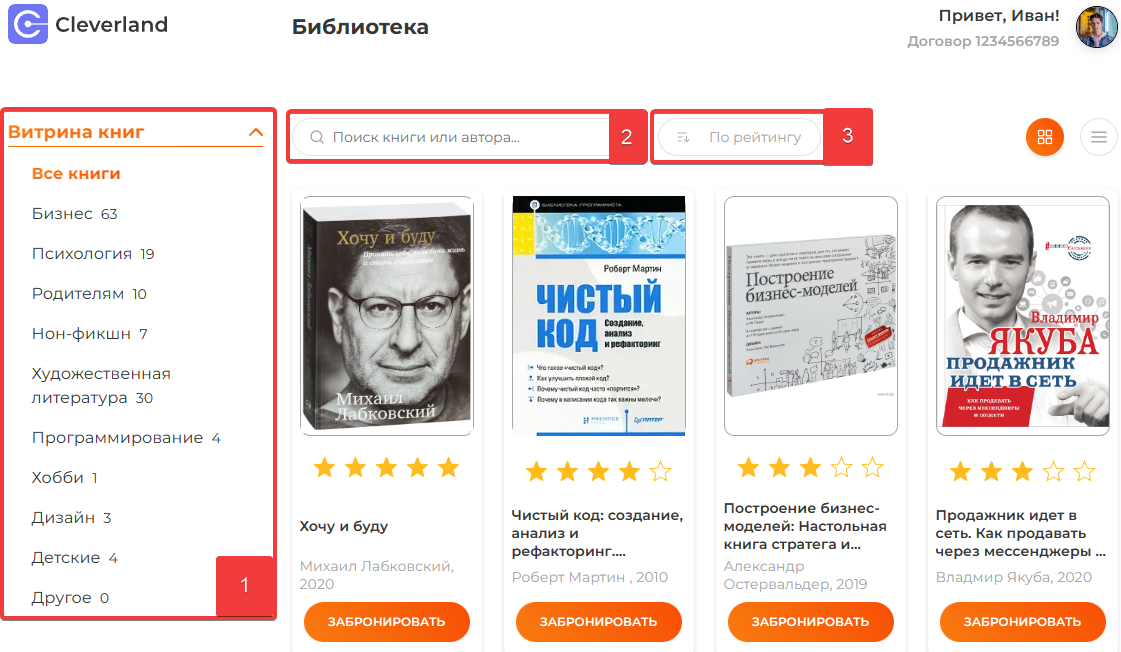
**Поиск и сортировка данных**

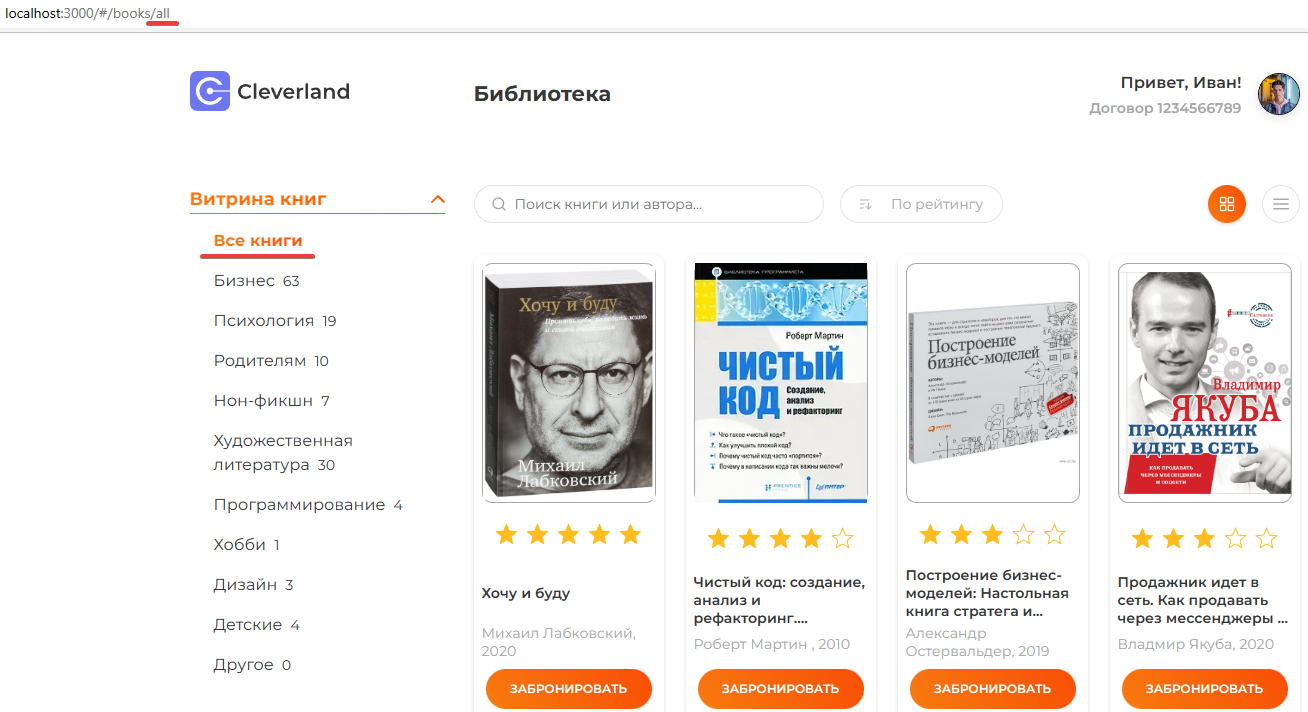
На странице имеется три способа отфильтровать данные:

1. Сортировка по категориям
2. Поисковая панель
3. Кнопка сортировки по рейтингу по возрастанию и убыванию

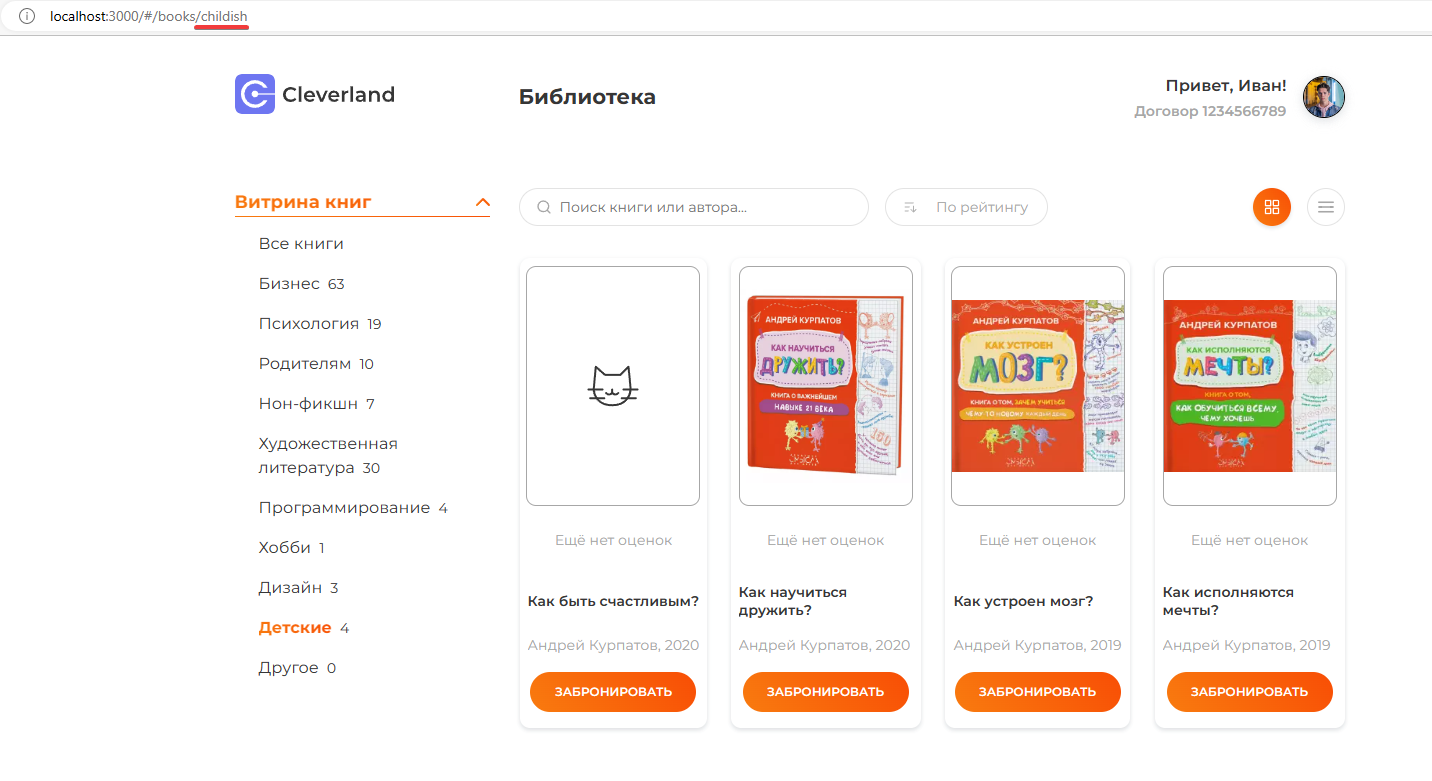


**Сортировка по категориям**

По умолчания должны отображаться все категории (пункт **«Все книги»**).



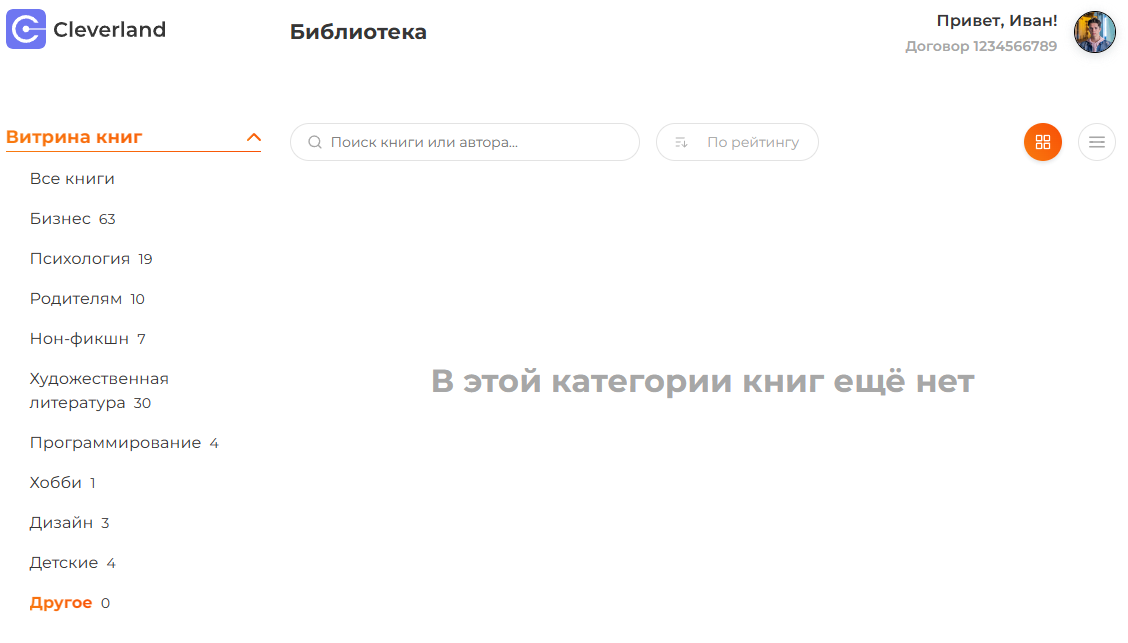
Если мы, например, выбираем категорию **Детские**, то на странице должны отобразиться только те книги, которые относятся к данной категории. При этом url должен содержать название выбранной категории, в данном случае **childish.**



Наличие выбранной категории в url является обязательным и проверяется автотестами. Ещё для примера, если выбраны **Все книги**, то в url должен содержать **all** и так по аналогии для каждой категории.

Также обратите внимание на то, что цифра рядом с названием категории должна быть равна количеству книг в данной категории. Как видно на скриншоте выше в категории **Детские** 4 книги и соответственно на экране у вас должно быть отображено 4 книги. Исключением является пункт **Все книги**, там отображать количество книг не нужно.

Если категория не содержит ни одной книги, то вы должны отображать текст “**В этой категории книг ещё нет**”.

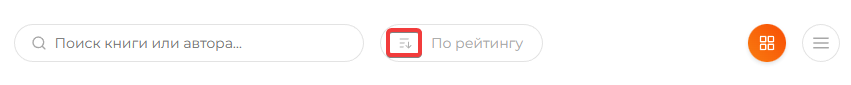


**Сортировка по рейтингу**

По умолчанию книги должны быть отсортированы по убывающему рейтингу. Т.е. сначала будут показаны книги с самым высоким рейтингом, а затем книги с более низким рейтингом. Книги без рейтинга будут находиться в самом конце списка.



Смена способа сортировки происходит по нажатию на кнопку сортировки по рейтингу, внутри которой имеется иконка, которая будет меняться в зависимости от выбранного способа сортировки. По умолчания кнопка выглядит так:

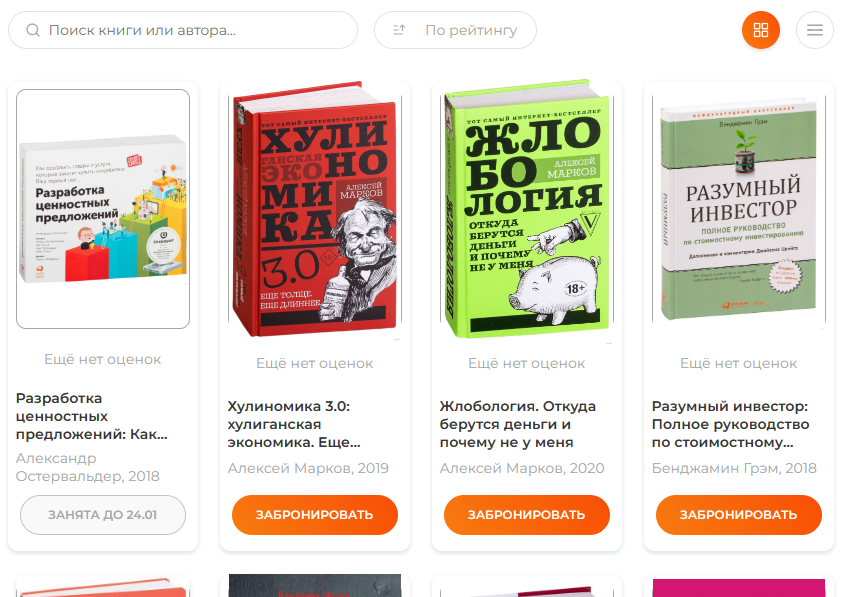


И указывается на то, что сейчас выбрана сортировка по убыванию, от большего к меньшему.

При нажатии на кнопку должна смениться иконка и список книг должен быть отсортирован в обратном порядке, то есть по возрастанию.



Книги с низким рейтингом или вообще без рейтинга теперь будут отображены в начале списка.



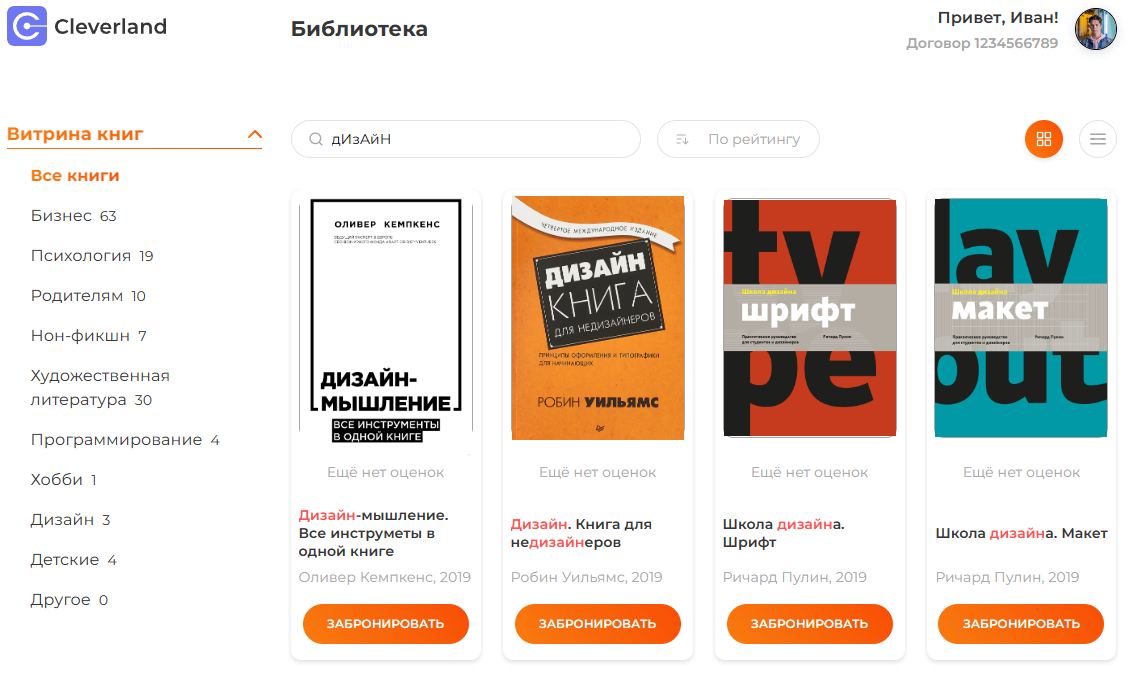
При повторном нажатии на кнопку сортировка снова становиться по убыванию по принципу переключателя.

Порядок сортировки сохраняется при смене категории и при поиске через поисковую панель.

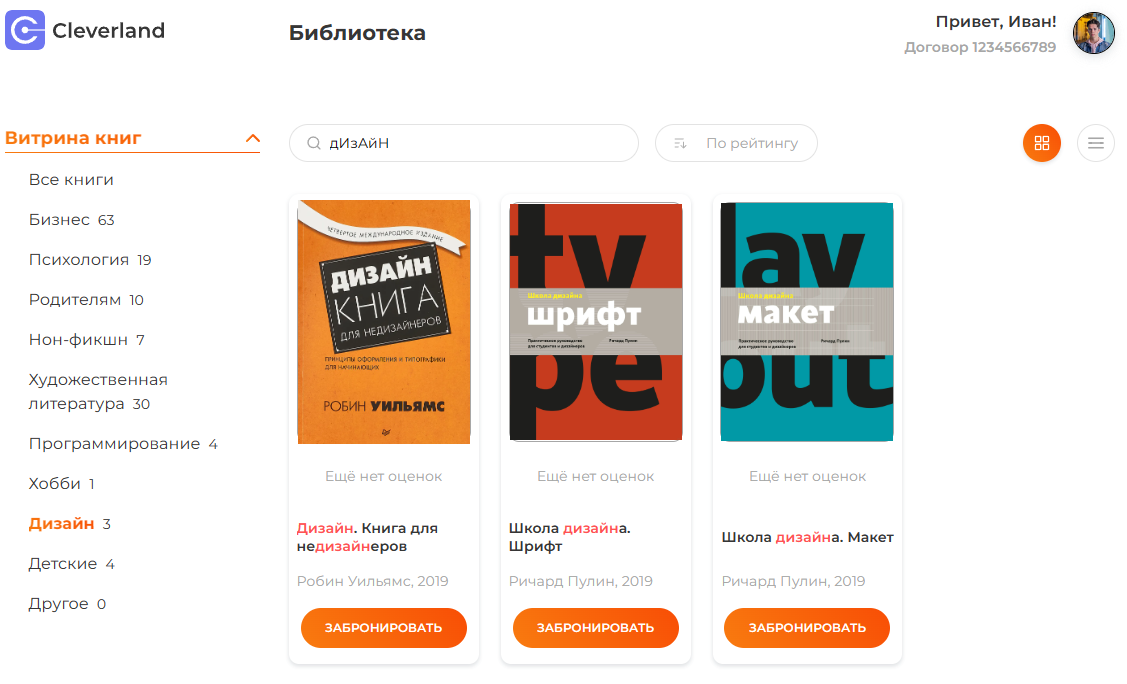
**Поисковая панель**

При вводе значений в поисковую панель происходит поиск по названию книг в выбранной категории, при этом совпадения поиска должны быть подсвечены. Поиск должен быть независим от регистра в котором происходит ввод в поисковую панель. При смене категории значение введенное в поисковую панель не сбрасывается и так же применяется к выбранной категории.

Например, если мы введем в поисковую панель “дИзАйН”, то по нашему запросу будут найдены все книги в названии которых содержится значение, которое мы вводили и будут подсвечены все совпадения в независимости от регистра в котором мы вводили запрос.

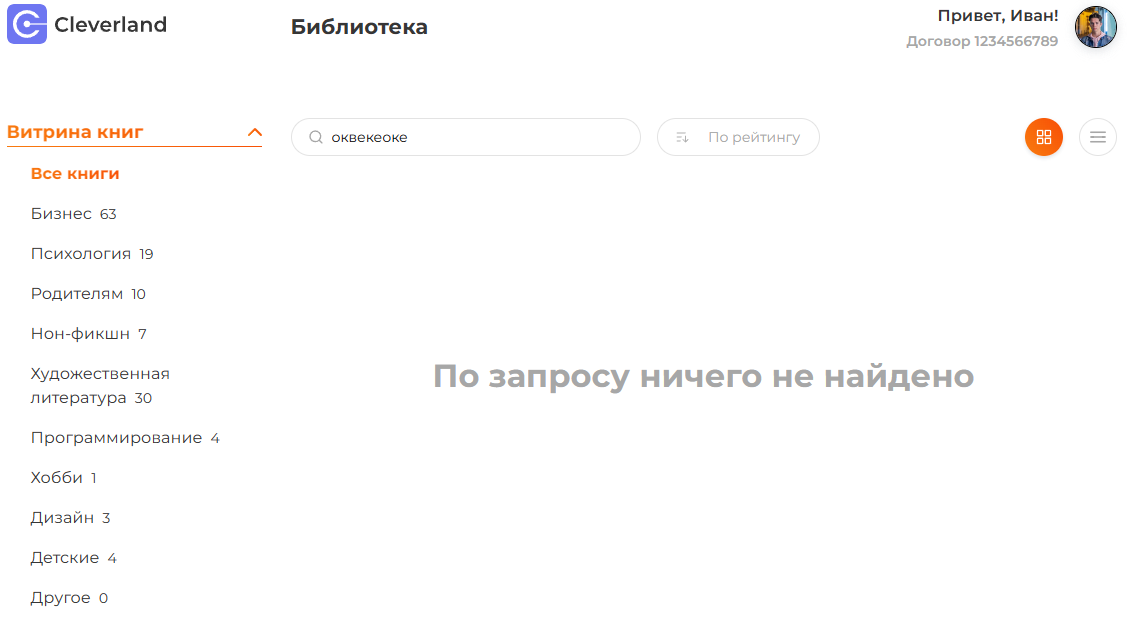


Также при смене категории введенное нами значение не должно сбрасываться и должно примениться к выбранной категории.

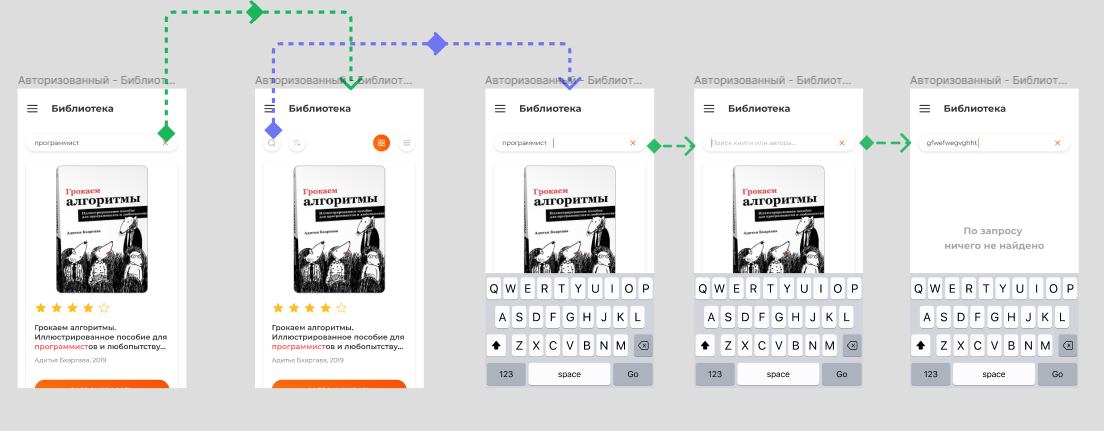


При очистке поисковой панели должны быть отображены все книги соответствующие выбранной категории, как при простой сортировке по категориям.

Если при вводе запроса нет подходящих данных, то вам необходимо отображать текст “**По запросу ничего не найдено**”.



В мобильной версии при нажатии на кнопку открытия поисковой панели должна открываться поисковая панель, при этом если мы введем какое-то значение в инпут и нажмем на крестик, тем самым свернем поисковую панель, то значение, которое мы вводили, должно сохраниться и при повторном открытии поисковой панели мы должны там увидеть значение, которое мы вводили. Наглядный пример как это должно работать вы можете найти в figma, ссылка на которую будет внизу инструкции.



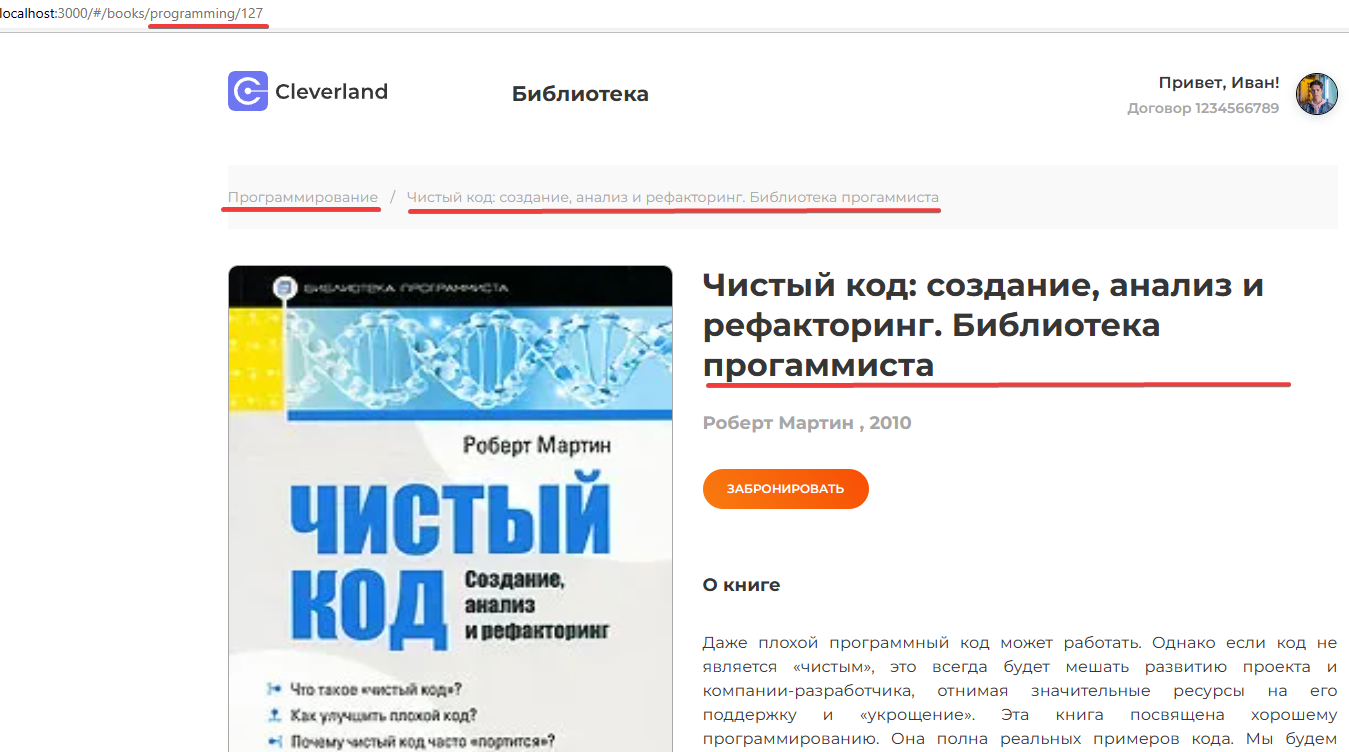
Будьте внимательны при работе с макетом, автотесты будут проверять даже такие мелочи как цвет каретки в поисковой панели 🙂

**Хлебные крошки**

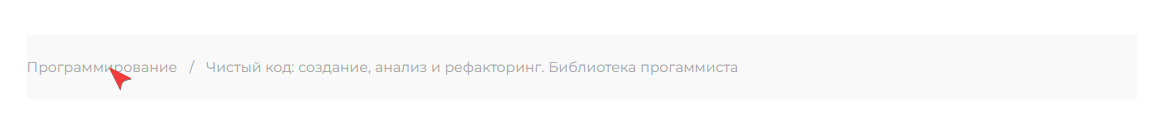
Также в данном спринте необходимо реализовать возможность возврата на главную страницу из хлебных крошек на странице с книгой. Разберем по шагам.

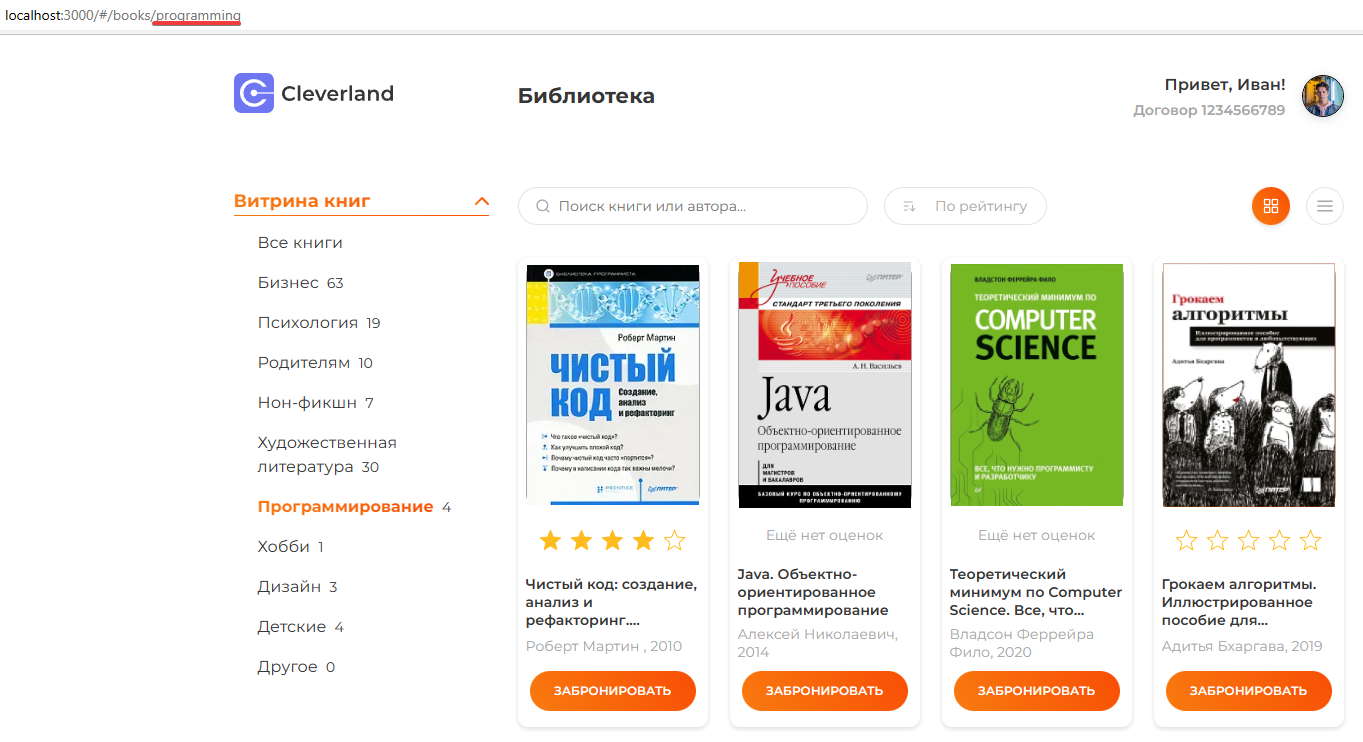
Для примера выберем категорию **Программирование** и выберем там первую книгу. На что следует обратить внимание:

1. Категория, с которой мы перешли на страницу конкретной книги должна отображаться в хлебных крошках на русском языке!
2. Название книги должно совпадать с текстом в хлебных крошках после слэша.
3. Url страницы должен включать в себя название категории на английском и id выбранной книги через слэш. В данном примере **programming/127**.

****

При клике на категорию в хлебных крошка должен происходить возврат на главную страницу с выбранной категорией **Программирование** в нашем случае. Если категория была **Дизайн**, то должна быть активна категория **Дизайн**. И так по аналогии для каждой категории.





При возврате на главную страницу из url страницы убирается **id** книги, но категория остается и в нашем примере url страницы должен содержать **programming**.Все это будет проверяться автотестами, так что будьте внимательны.

Еще один **важный момент**, что при возврате на главную страницу вам необходимо повторно отправлять запрос за книгами(не категориями!), чтобы поддерживать актуальность данных на странице.

**Один из путей решения этой задачи**

Для решения данной задачи вам понадобиться пару хуков: **useState**, в котором будет храниться наш отсортированный массив данных и **useEffect**, который будет срабатывать при изменении категории, метода сортировки или при вводе значений в поисковую панель.

Для фильтрации данных отлично подойдет метод массивов **filter**, а для сортировки по рейтингу метод **sort**.

При реализации подсветки совпадений в названиях книг использование сторонних библиотек не запрещается, но можно обойтись и без них. В интернете вы сможете найти массу вариантов решения этой задачи.

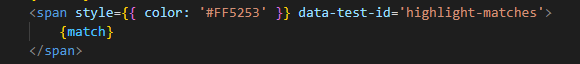
**Для прохождения автотестов вам понадобиться**

Поисковая панель должна иметь атрибут ***data-test-id=’input-search’***



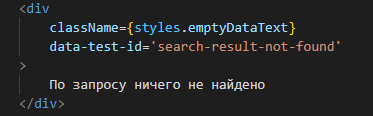
Тег, которым вы будете оборачивать текст для подсветки совпадений поиска должен иметь атрибут ***data-test-id=’highlight-matches’***

В нашем примере это span.



Тег, в котором будет отображен текст “По запросу ничего не найдено” должен иметь атрибут***data-test-id=’search-result-not-found’***

В нашем примере это div.

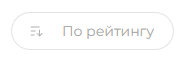


Тег, в котором будет отображен текст “В этой категории книг ещё нет” должен иметь атрибут ***data-test-id=’empty-category’***

В нашем примере это div.

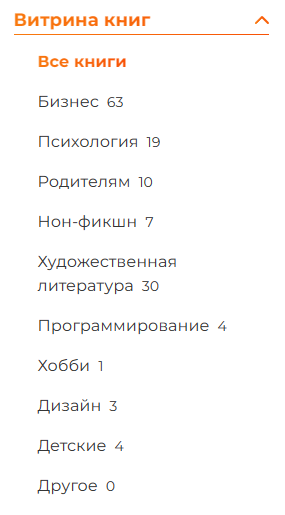


Кнопка сортировки по рейтингу должна иметь ***data-test-id=’sort-rating-button’***



Категории в меню навигации должны иметь следующие data-test-id:

Для десктопного варианта меню

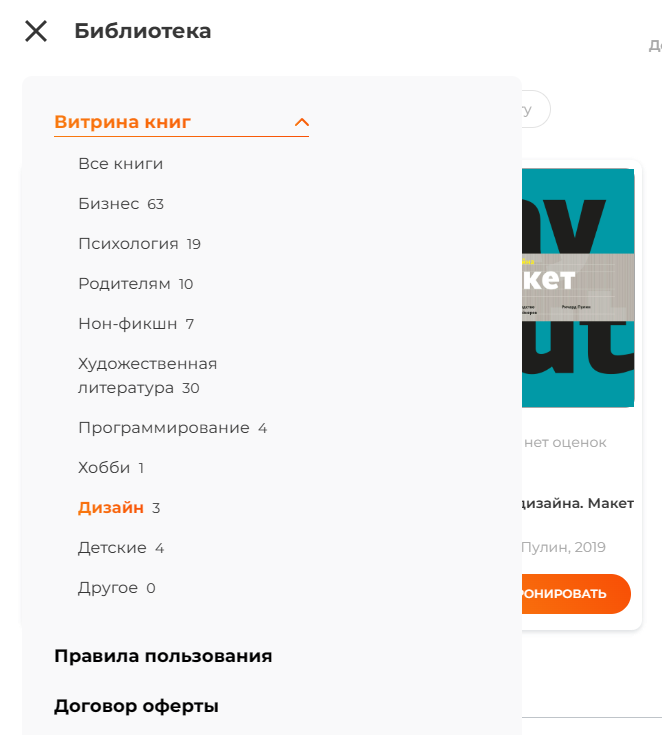


Каждая категория должна иметь свой data-test-id, который начинается с **navigation**. Например, для категории программирование ***data-test-id=’navigation-programming’***,

для категории дизайн ***data-test-id=’navigation-design’***.

Для этого можно воспользоваться следующей конструкцией: ***data-test-id=`navigation-${category}`*** и подставлять категорию, которую мы получили с сервера, а для категории **Все книги** ***data-test-id=’navigation-books’***.

Для бургер меню принцип с категориями остается такой же, только вместо **navigation** будет **burger.**



Например, ***data-test-id=’burger-programming’***, ***data-test-id=’burger-design’***. И подкидывать значение в зависимости от категории ***data-test-id=`burger-${category}`***. Для категории **Все книги** ***data-test-id=’burger-books’***.

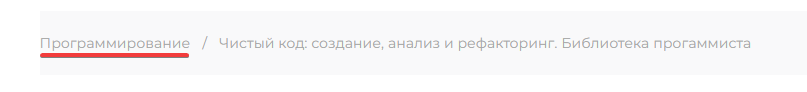
Еще раз заострю внимание на том, что если у нас выбрана категория **Все книги**, то url страницы должен содержать **all**, но data-test-id должен иметь ***data-test-id=’navigation-books’*** для десктопной версии меню и ***data-test-id=’burger-books’*** для бургер меню.

Тег, в котором вы будете отображать количество книг в категории должен иметь атрибут для десктопной версии меню ***data-test-id=`navigation-book-count-for-${category}`***. Например, для категории Программирование ***data-test-id=’navigation-book-count-for-programming’***.

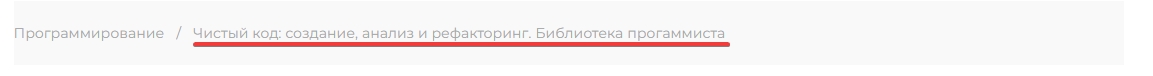
И для бургер меню ***data-test-id=`burger-book-count-for-${category}`***. Например, ***data-test-id=’burger-book-count-for-programming’***.



Ссылка возврата на главную страницу должна иметь ***data-test-id=’breadcrumbs-link’***.

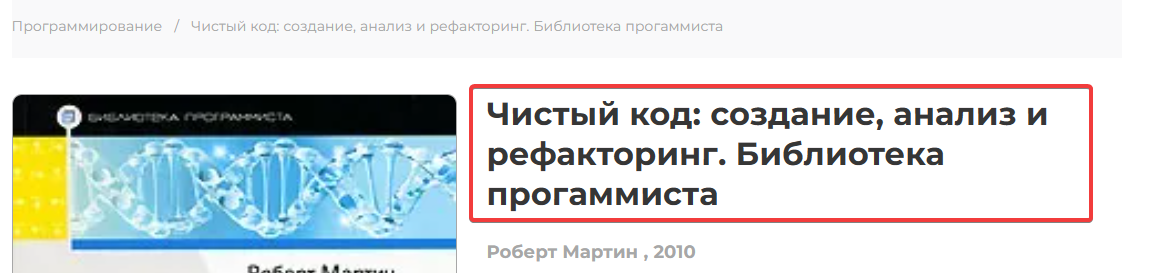


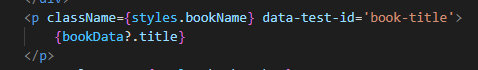
Тег, в котором вы будете отображать текст с названием книги должен иметь ***data-test-id=’book-name’***.





Тег, в котором вы будете отображать название книги должен иметь ***data-test-id=’book-title’***.





Возможно, какие-то из этих data-test-id у вас уже имеются. И не удаляйте data-test-id прошлых спринтов, они используются в новых спринтах.

Ссылка на макет в figma: [макет спринт 4](https://www.figma.com/file/d8LhhLjMkaTfPvAcYQULNv/Library---students-file?node-id=3601%3A32185&t=WHYHy5dLhHhBpShp-0)

Всем удачи! По вопросам или за разъяснениями обращайтесь в телеграм-канал.