

Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 21.Б07-мм

Улучшение фронтенд-части веб-приложения проекта Desbordante

СОЛОВЬЁВА Лиана-Юлия Викторовна

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
ассистент кафедры информационно-аналитических систем Чернышев Г. А.

Санкт-Петербург
2024

Оглавление

Введение	3
1. Определения	4
2. Постановка задачи	6
3. Описание проблем	7
4. Используемые инструменты	9
5. Описание решения	11
5.1. Динамический заголовок веб-страниц	11
5.2. Редактирование анимации хедера	11
5.3. Улучшение подсказок при регистрации и авторизации . .	12
5.4. Переработка плавающих элементов	12
5.5. Переработка представления иконок	14
6. Тестирование	16
Заключение	17
Список литературы	18

Введение

Desbordante¹ [3] — наукоёмкий профайлер, созданный для работы с данными. Благодаря реализованным алгоритмам, которым задаются определенные параметры, можно выявлять и валидировать различную полезную информацию, такую как:

- точные функциональные зависимости [4];
- приблизительные функциональные зависимости [5];
- условные функциональные зависимости [6];
- ассоциативные правила [1];
- алгебраические ограничения [2] и др.

Проект представлен в виде трех интерфейсов для взаимодействия с пользователями: консольное и веб-приложения², а также `pip`-пакет³. Проект продолжает развиваться, добавляются новые возможности, находят и исправляются не замеченные ранее ошибки.

Отдельно стоит обратить внимание на веб-приложение, которое состоит из фронтенд и бэкенд-частей. С ростом реализованных алгоритмов для работы с данными растет и кодовая база фронтенд-части. В результате в ней накапливается большое количество неиспользуемого кода, дубликатов и ошибок. Все эти факторы усложняют процесс разработки. Также существует ряд визуальных недостатков, которые оставляют негативные впечатления у пользователей при взаимодействии с сайтом.

Данная работа направлена на выявление, анализ и исправление визуальных ошибок, добавление новых компонентов и внесение изменений для повышения качества кодовой базы фронтенд-части проекта Desbordante.

¹<https://github.com/Desbordante/desbordante-core>

²<https://desbordante.unidata-platform.ru/>

³<https://pypi.org/project/desbordante/>

1. Определения

ReactDOM — тип элементов в TypeScript, объединяющий всё, что способен отобразить React. В частности, строки, числа, null и д.р.

SEO (Search Engine Optimization) — оптимизация веб-сайтов для увеличения попаданий в результаты поисковых запросов.

SVG (Scalable Vector Graphics) — масштабируемая векторная графика, изображения которой хранятся в виде математического описания геометрических простых составляющих (кривые, точки, круги и т.д.) на языке разметки.

SVGProps (от англ. **properties**) — тип элементов в TypeScript, который представляет собой набор параметров, которые могут принимать svg-компоненты (обработчики событий, класс, цвет, размер и д.р.)

Портал (portal) — компонент, дающий возможность позиционирования элемента относительно не только родительского, а любого на странице.

Хук (hook) — функция, позволяющая использовать возможности React без создания классов, например, для управления состоянием компонентов.

Хедер (header) — компонент, на всех страницах веб-приложения расположенный в самом верху. Преимущественно используется для размещения главной информации (название, логотип), навигации (меню, ссылки на другие страницы), кнопок регистрации и авторизации.

Под **плавающими элементами** будем понимать модальные окна, тултипы и выпадающие списки.

Модальные окна (modal windows) — это окна, которые появляются в ответ на срабатывание определенного события (нажатие на кнопку, успешная авторизация и д.р.) поверх основного интерфейса, блокирующие взаимодействие с ним.

Тултипы (tooltip) — это окна, всплывающие при наведении курсора и исчезающие после его увода, часто небольшие и содержащие подсказки.

Выпадающие списки — это элементы, состоящие из двух частей:

поля, показывающего одну или несколько выбранных опций, и самого выпадающего списка, появляющегося при фокусировке (наведение курсора или клик) на этом поле и содержащего список опций для выбора.

2. Постановка задачи

Целью работы является улучшение фронтенд-части веб-приложения проекта Desbordante. Для её выполнения были поставлены следующие задачи:

1. провести обзор визуальных ошибок и проблем кодовой базы;
2. выявить и проанализировать причины рассмотренных проблем и найти способ их устранения;
3. устранить имеющиеся проблемы и ошибки;
4. протестировать результат работы.

3. Описание проблем

1. Веб-приложение Desbordante состоит не из одной страницы, но у всех них единый заголовок: «Desbordante | Open-source Data Profiling Tools». Он отражает лишь название и краткое описание веб-приложения, однако ничего не говорит о назначении открытой страницы, что может затруднять понимание содержания выбранной вкладки пользователем.
2. Задумка стиля хедера заключается в том, что при прокрутке до определенного момента его фон является прозрачным, а потом приобретает цвет и рамку, тем самым выделяется его расположение поверх основного наполнения страницы. Однако изначально получилось так, хедер наезжает на содержимое, но его стиль не меняется.
3. Когда пользователь только регистрируется, ему нужно указать данные о себе, почту и пароль. При этом, на пароль накладываются определенные требования, однако пользователь не видит этой информации. Если указывается не удовлетворяющая правилам последовательность символов, то система сообщит: «Weak password» («Слабый пароль»). Так пользователь не поймет, какой набор символов нужно использовать в пароле, чтобы зарегистрироваться.
4. Контент, который располагается внутри тултипа, передается строкой, однако иногда требуется вставить, например, список или ссылку.
5. Модальные окна и тултипы реализованы самостоятельно (без специальных библиотек), из-за чего вызывают ошибки. Кроме того, они всегда находятся на странице, хотя, в зависимости от ситуации, могут быть скрыты для пользователя. Это негативно сказывается на производительности, ведь браузер их всё равно «видит», а значит должен загрузить, а также может повлиять на SEO в худшую сторону.

6. Модальные окна «мигают», то есть появляется визуальная ошибка, в которой окно на мгновение пропадает при изменении данных, расположенных внутри.
7. В консоли иногда появляется ошибка⁴, связанная с изменением параметров в содержимом модальных окон, а именно с невозможностью прочитать параметр у переменной со значением `undefined`.
8. Всплывающие компоненты, находящиеся внутри модального окна, не могли отображаться поверх него, то есть не могли выходить за его границы. Это приводит к тому, что открывающуюся информацию неудобно смотреть.
9. В веб-приложении Desbordante используется множество svg-файлов («иконок»), встречающихся в различных компонентах (фоны страниц, кнопки, логотип, поля ввода и т. д.). Причем одна и та же иконка может использоваться разных размеров и цветов, а реализация разных случаев могла происходить также по-разному: создание нового файла с похожим svg-кодом и нужными параметрами, использование классов стиля, меняющих цвет и размер по умолчанию. При добавлении на страницу одной и той же иконки разные разработчики могли назвать её по-разному или наоборот, разные иконки на разных страницах имеют одно имя. Всё это приводит к путанице и затрудняет чтение и разработку кода.

⁴<https://github.com/Desbordante/desbordante-web/issues/76>

4. Используемые инструменты

TypeScript [16] — это расширение JavaScript — языка программирования, популярного для разработки веб-приложений. Главным отличием является наличие статической типизации, что упрощает разработку кода и рефакторинг.

React [12] — фреймворк для создания пользовательских интерфейсов, которая позволяет строить приложения из независимых и переиспользуемых элементов.

Next.js [11] — это фреймворк для разработки приложений на React, который поддерживает как серверный рендеринг, так и статическую генерацию страниц. Он упрощает маршрутизацию и оптимизацию производительности, что делает его идеальным выбором для создания современных веб-приложений с высокой производительностью.

Next SEO [10] — инструмент для упрощения управления метаданными и SEO в приложениях на Next.js. С его помощью разработчики могут настраивать заголовки, описания и другие важные атрибуты, что способствует повышению видимости сайта в поисковых системах.

React Select [13] — библиотека для создания многофункциональных выпадающих списков в приложениях на React. Она предлагает богатый набор возможностей, в том числе настройку собственных стилей.

React Spring [14] — библиотека для анимации в приложениях на React, основанная на принципах физики. Она предлагает интуитивно понятные API для создания плавных и естественных анимаций, позволяя разработчикам добавлять разнообразные динамические эффекты.

Floating UI [7] — библиотека, предназначенная для работы с всплывающими элементами. Она предоставляет понятную документацию и полезные инструменты для создания, позиционирования и анимации. Примеры простых вариантов использования можно найти на их сайте.

Jotai [9] — библиотека для управления состоянием в React-приложениях с атомарным подходом. Каждый атом представляет собой отдельную единицу состояния, что позволяет легко управлять данными

и минимизировать сложность архитектуры приложения.

5. Описание решения

5.1. Динамический заголовок веб-страниц

Для большего удобства пользователей было решено выбрать такие заголовки для различных страниц, чтобы они отражали их назначение.

Заголовки, отражающие назначение конкретных страниц, не только пользователям помогут определять, какая вкладка открыта, но и поисковым роботам даст бóльшую информацию о назначении страниц, то есть повысит вероятность попадания в релевантные ответы на запросы людей.

Данное изменение было реализовано с помощью библиотеки Next SEO.

Новыми заголовками стали, например:

- «Desbordante | Papers» для страницы с публикациями,
- «Desbordante | Team» для страницы с участниками проекта,
- «Desbordante | Step 1: select a pattern to discover or verify»,
- «Desbordante | Step 2: select a dataset»,
- «Desbordante | Step 3: configure algorithm» для соответствующих страниц конфигурации задачи.

5.2. Редактирование анимации хедера

Момент перехода определяется расстоянием блока контента до верхней границы страницы. Первоначально изменение стиля хедера происходило, когда этот блок находился в менее чем 48 пикселях, однако данное значение оказалось неудачным. После рассмотрения вариантов 64, 80 и 100 пикселей, было решено остановиться на 64, когда верхняя граница основной части касается хедера, то есть с его нижней видимой границей (фиолетовой рамкой).

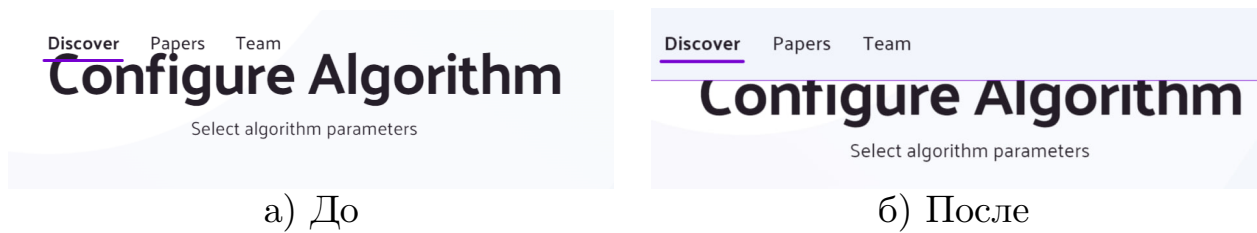


Рис. 1: Появление хедера

5.3. Улучшение подсказок при регистрации и авторизации

Для лучшего понимания требований системы пользователем было решено добавить всплывающую подсказку, в которой описаны все требования, а также в случае, если пароль не удовлетворяет этим правилам, система уведомит об этом и посоветует обратить внимание на тултип.

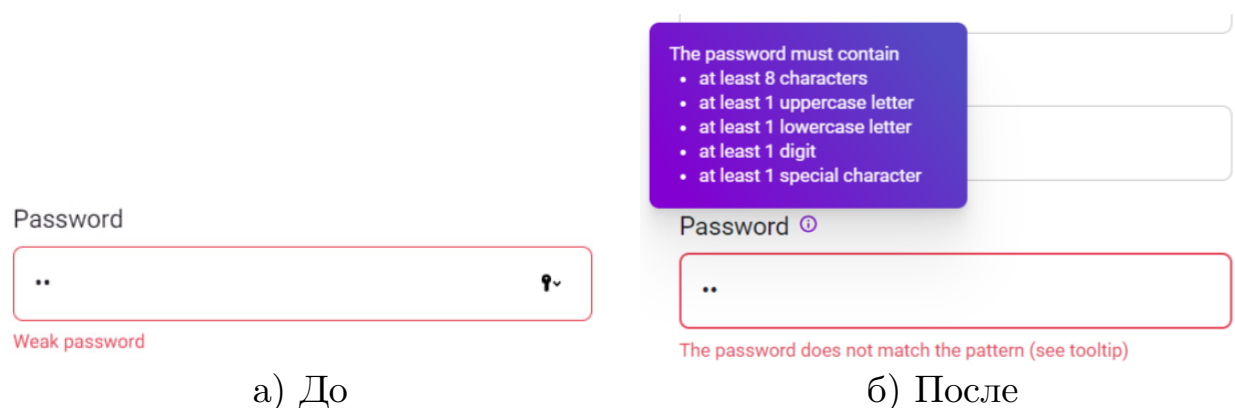


Рис. 2: Регистрация. Ввод пароля

5.4. Переработка плавающих элементов

Для переработки модальных окон и тултипов использовалась библиотека Floating UI. Её применение позволило избежать добавления компонентов на страницу, которых в данный момент не должно быть, что снижает время загрузки и повышает эффективность SEO.

Переработка тултипов заключалась в двух отдельных изменениях. Во-первых, их надо было переписать с помощью новой библиотеки. Во-

вторых, тип передаваемого для отображения внутри всплывающего окна компонента изменен со строки на `ReactNode`. Такие изменения были полезны при улучшении подсказок при регистрации.

Переписывание модальных окон с использованием библиотеки могло исправить все связанные с ними ошибки.

Мигали они из-за визуальных эффектов, реализованных с помощью библиотеки `React Spring`. Однако `Floating UI` предоставляет собственную анимацию, с которой проблем не возникло.

Ошибка в консоли появлялась из-за самостоятельно разработанного командой ранее хука `useModal` и хранения информации для отображения о всех открытых модальных окнах в атоме `Jotai`, который мог быть пустым. Изначально предполагалось, что данная реализация поможет избежать дублирования кода, однако позже стало ясно, что она приводит только к ошибкам. Решением послужил отказ от данной технологии: теперь модальные окна вызываются непосредственно в ответ на соответствующие события, так как в один момент может быть открыто не более одного модального окна.

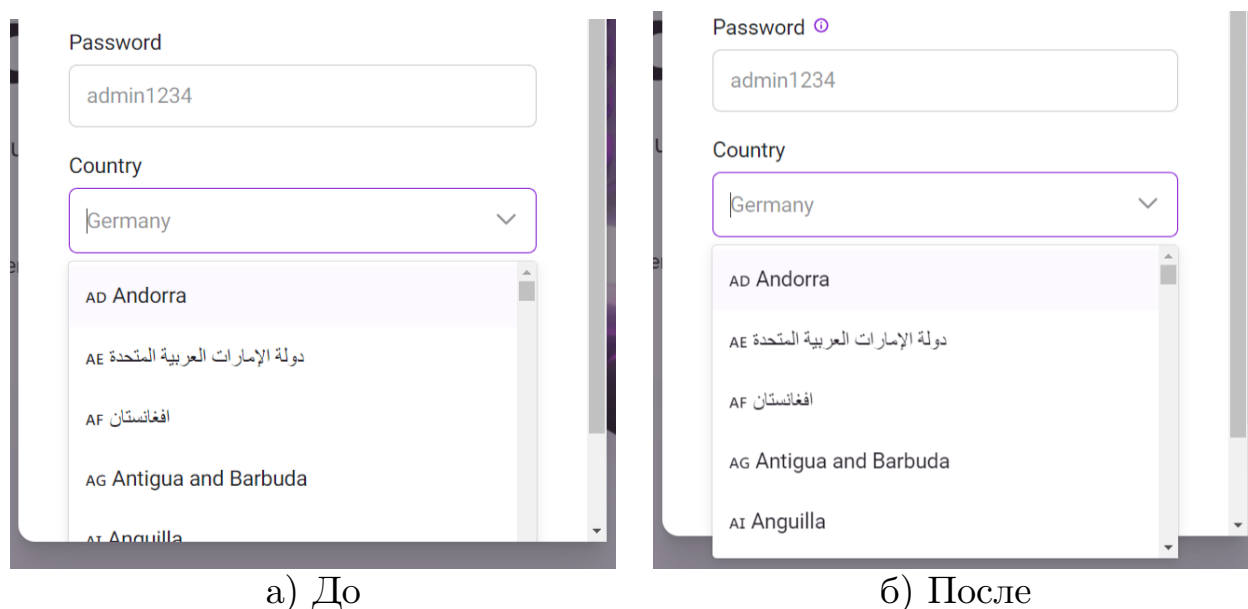


Рис. 3: Выпадающий список

Проблема с вложенными плавающими элементами была решена добавлением порталов. Дополнительно для них был создан специальный

компонент-хранилище. Floating UI и React Select предлагают собственные реализации порталов, поэтому дополнительных библиотек не понадобилось.

5.5. Переработка представления иконок

Целью данной работы в контексте переработки представления иконок являлось привести их к единому виду (стандартный цвет и размер), удалить дубликаты разных и реализовать единый способ представления.

На момент начала данной работы в файлах проекта находилось 33 иконки и 4 разных фона. Ввиду малочисленности последних они были различными и использовались на каждой странице одинаковым образом, чего нельзя сказать про первые. На рисунке 4 можно видеть все иконки, как они хранились в svg-файлах до изменения.

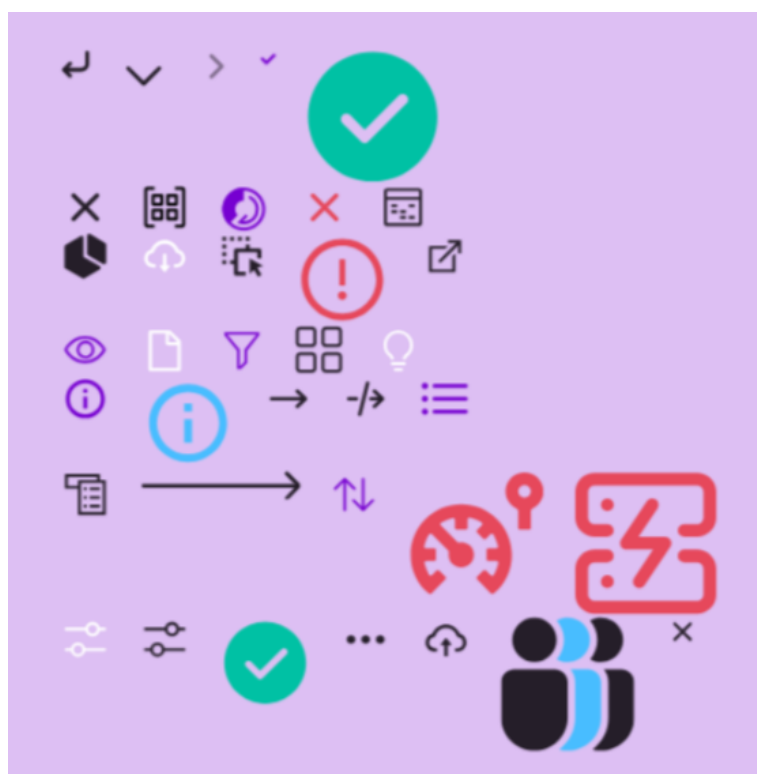


Рис. 4: Множество иконок до изменений

В отдельные случаи можно выделить иконки «галочка» и «очередь». Первая используется в проекте повернутая в разные стороны, однако

представлена в виде двух отдельных файлов, а вторая имеет в себе два цвета, каждый из которых мы хотим определять самостоятельно и независимо друг от друга, однако если менять не сам svg-код, мы сможем определить лишь один цвет. Это также указывает на то, что нужно создать отдельный компонент, в котором в зависимости от типа иконки будут указываться не только стандартные SVGProps, но и направление, несколько независимых цветов и другие параметры, которые могут понадобиться в будущем.

Решением стало создание общего хранилища svg-кодов для иконок и фонов, а также был создан компонент, в аргументах которого обязательно передается предопределенное имя иконки, которую мы хотим добавить на страницу, а так же, опционально, её параметры. В результате все svg-компоненты в проекте будут добавляться на страницу единым образом, что облегчит чтение и разработку кода.

6. Тестирование

Для тестирования в проекте настроены и используются два популярных инструмента.

Jest [8] — это инструмент для тестирования JavaScript-приложений. Он предлагает простую в использовании структуру для написания модульных тестов, поддерживает изоляцию тестов, обеспечивает быструю обратную связь благодаря параллельному запуску тестов и имеет возможность создания отчетов о покрытии кода, что позволяет разработчикам легко выявлять недостатки в тестировании.

React Testing Library [15] — это библиотека для тестирования компонентов React, ориентированная на удобство использования. Благодаря API и методам, которые имитируют взаимодействие пользователя, она дает возможность тестировать компоненты так, как их использует пользователь, а не сосредотачиваться на внутренней реализации.

В проекте уже были написаны некоторые тесты, однако, из-за изменения реализации, не все новые компоненты проходили их. Такое, например, случилось с тултипом: из-за того, что он теперь не существует на странице до наведения курсора на соответствующую иконку информации, а когда появляется, то оказывается в специальном контейнере, тестирующая система не могла его найти.

Также надо было написать тесты, которых ранее не было. Тестированию подверглись модальные окна (появляются при нажатии на кнопку открытия, закрываются при нажатии на кнопку закрытия и клика в любое место вне), выпадающие списки (открываются по нажатию на поле ввода, закрываются при выборе опции и по клику вне), в том числе с множественным выбором, а также проверялся ответ системы на различные, правильные и неправильные, варианты потенциальных паролей.

Итого было написано 20 тестов.

Заключение

Результатом данной работы стало улучшение фронтенд-части веб-приложения проекта Desbordante. Были выполнены следующие задачи:

1. проведен обзор визуальных ошибок и проблем кодовой базы;
2. выявлены и проанализированы причины рассмотренных проблем и найден способ их устранения;
3. устранены имеющиеся проблемы и ошибки;
4. протестирован результат работы.

Код работы находится в GitHub репозитории веб-приложения Desbordante⁵.

⁵<https://github.com/Desbordante/desbordante-web/pull/133>

Список литературы

- [1] Agrawal Rakesh, Srikant Ramakrishnan. Fast Algorithms for Mining Association Rules in Large Databases // VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile / Ed. by Jorge B. Bocca, Matthias Jarke, Carlo Zaniolo. — Morgan Kaufmann, 1994. — P. 487–499. — URL: <http://www.vldb.org/conf/1994/P487.PDF>.
- [2] Brown Paul G., Hass Peter J. BHUNT: Automatic Discovery of Fuzzy Algebraic Constraints in Relational Data // Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29. — VLDB '03. — VLDB Endowment, 2003. — P. 668–679. — URL: <https://www.vldb.org/conf/2003/papers/S20P03.pdf>.
- [3] Desbordante: from benchmarking suite to high-performance science-intensive data profiler (preprint) / George A. Chernishev, Michael Polyntsov, Anton Chizhov et al. // CoRR. — 2023. — Vol. abs/2301.05965. — arXiv : [2301.05965](https://arxiv.org/abs/2301.05965).
- [4] Functional Dependency Discovery: An Experimental Evaluation of Seven Algorithms / Thorsten Papenbrock, Jens Ehrlich, Janik Marten et al. // Proc. VLDB Endow. — 2015. — Vol. 8, no. 10. — P. 1082–1093. — URL: <http://www.vldb.org/pvldb/vol8/p1082-papenbrock.pdf>.
- [5] Kruse Sebastian, Naumann Felix. Efficient Discovery of Approximate Dependencies // Proc. VLDB Endow. — 2018. — mar. — Vol. 11, no. 7. — P. 759–772. — URL: <https://www.vldb.org/pvldb/vol11/p759-kruse.pdf>.
- [6] Rammelaere Joeri, Geerts Floris. Revisiting Conditional Functional Dependency Discovery: Splitting the “C” from the “FD” // Machine Learning and Knowledge Discovery in Databases / Ed. by Michele Berlingerio, Francesco Bonchi, Thomas Gärtner et al. — Cham : Springer International Publishing, 2019. — P. 552–568.

- [7] Документация Floating UI. — URL: <https://floating-ui.com/docs/getting-started> (дата обращения: 21 сентября 2024 г.).
- [8] Документация Jest. — URL: <https://jestjs.io/docs/getting-started> (дата обращения: 3 октября 2024 г.).
- [9] Документация Jotai. — URL: <https://jotai.org/docs/introduction> (дата обращения: 21 сентября 2024 г.).
- [10] Документация Next SEO. — URL: <https://www.npmjs.com/package/next-seo> (дата обращения: 21 сентября 2024 г.).
- [11] Документация Next.js. — URL: <https://nextjs.org/docs> (дата обращения: 21 сентября 2024 г.).
- [12] Документация React. — URL: <https://react.dev/> (дата обращения: 21 сентября 2024 г.).
- [13] Документация React Select. — URL: <https://react-select.com/home> (дата обращения: 21 сентября 2024 г.).
- [14] Документация React Spring. — URL: <https://react-spring.dev/docs> (дата обращения: 21 сентября 2024 г.).
- [15] Документация React Testing Library. — URL: <https://testing-library.com/docs/react-testing-library/intro> (дата обращения: 3 октября 2024 г.).
- [16] Документация TypeScript. — URL: <https://www.typescriptlang.org/docs/> (дата обращения: 21 сентября 2024 г.).