

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОЙ ЧАСТИ ДЛЯ САЙТА С
ОБРАЗОВАТЕЛЬНЫМИ ИГРАМИ ДЛЯ ДЕТЕЙ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Наумова Ильи Геннадьевича

Научный руководитель

зав. каф. техн. пр.,

к. ф.-м. н., доцент

И. А. Батраева

Заведующий кафедрой

к. ф.-м. н., доцент

С. В. Миронов

Саратов 2024

СОДЕРЖАНИЕ

| | |
|--|----|
| ВВЕДЕНИЕ | 3 |
| 1 Основное содержание работы | 5 |
| 1.1 Веб-приложение SPA (Single Page Application) | 5 |
| 1.2 Библиотека React | 5 |
| 1.3 Библиотека Recoil | 6 |
| 1.4 Контейнеризация приложения (Docker) | 6 |
| 1.5 Стартап как диплом | 7 |
| 1.6 Структура приложения | 7 |
| 1.7 Сервис для авторизации | 8 |
| 1.8 Сервис для пользователей и их взаимодействия | 8 |
| 1.9 Сервис для игр | 9 |
| 1.10 Управление состоянием через Recoil | 9 |
| 1.11 Компонент «Header» | 10 |
| 1.12 Страница со списком игр | 10 |
| 1.13 Страница игры | 11 |
| 1.14 Страница профиля | 11 |
| 1.15 Контейнеризация и деплой приложения | 12 |
| ЗАКЛЮЧЕНИЕ | 13 |

ВВЕДЕНИЕ

Актуальность темы. В современном мире развитие информационных технологий оказывает значительное влияние на образовательный процесс, где цифровые образовательные ресурсы становятся неотъемлемой частью учебного процесса. В последние годы, с ростом доступности интернета и вычислительных технологий, веб-приложения для образования привлекли внимание как образовательных учреждений, так и разработчиков программного обеспечения. Эти приложения предлагают интерактивные возможности для обучения и саморазвития, и поэтому их интеграция в учебные программы видится перспективным направлением развития современного образования.

Образовательные игры для детей занимают особую нишу в этом направлении, так как сочетают в себе образовательный процесс с игровыми элементами, что делает обучение более захватывающим и эффективным. Несмотря на большое количество исследований и разработок в данной области, постоянное обновление технологий и подходов требует новых решений, что делает тему актуальной на сегодняшний день. Поэтому разработка удобных и функциональных интерфейсов становится важной задачей, влияющей на качество и доступность современного образования.

Цель бакалаврской работы — разработка пользовательской части для сайта с образовательными играми для детей, которая будет отвечать современным требованиям удобства использования, интерактивности и визуальной привлекательности.

Поставленная цель определила **следующие задачи**:

1. изучить современные тенденции в разработке образовательных приложений;
2. проанализировать технологический стек;
3. спроектировать архитектуру пользовательского интерфейса;
4. реализовать интерактивные компоненты с применением современных библиотек;
5. провести интеграцию с серверной частью для обработки данных;
6. опубликовать приложение на сервере.

Структура и объём работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 7 приложений. Общий объём работы — 54 страницы, из них 40 страниц — основное содержание, вклю-

чая 10 рисунков, съёмный носитель в качестве приложения, список использованных источников информации — 21 наименование.

1 Основное содержание работы

1.1 Веб-приложение SPA (Single Page Application)

Single Page Application (SPA) — это современный подход к разработке веб-приложений, при котором всё приложение загружается в одном HTML-документе. Вместо загрузки новых страниц с сервера при каждой навигации, SPA динамически обновляет содержимое текущей страницы, что обеспечивает более плавный и быстрый пользовательский опыт (UX).

Основные характеристики SPA:

- **Единый HTML-документ:** всё приложение загружается в одном HTML-документе, а содержимое страницы обновляется динамически.
- **Обогащенный клиентский функционал:** большая часть логики выполнения находится на стороне клиента, что снижает нагрузку на сервер.
- **Асинхронные запросы:** взаимодействие с сервером осуществляется через AJAX-запросы или другие методы асинхронной загрузки данных.

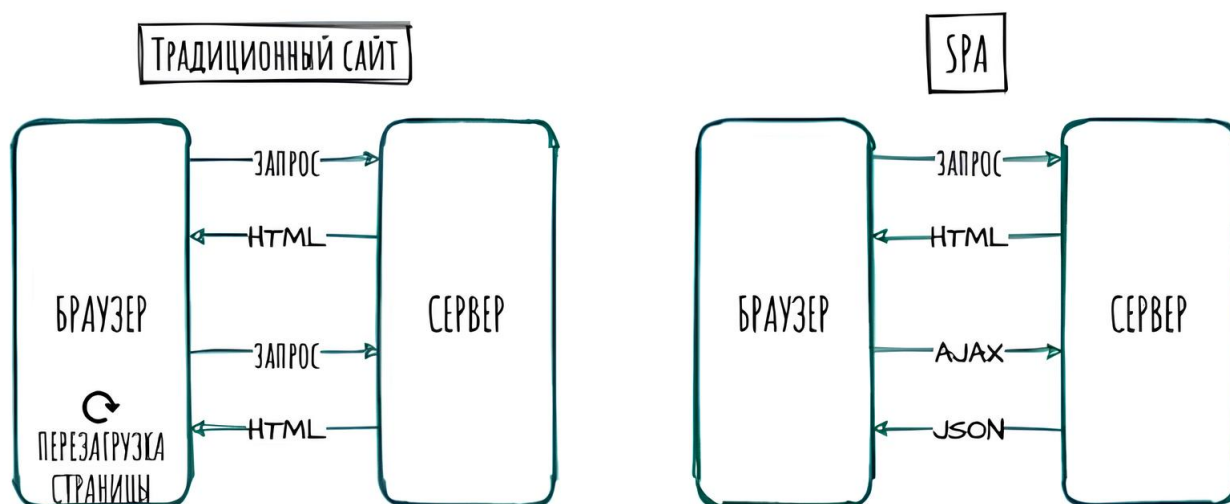


Рисунок 1 – Сравнение традиционного сайта с SPA

1.2 Библиотека React

React — это популярная библиотека для создания пользовательских интерфейсов, разработанная и поддерживаемая Facebook. Она предназначена для упрощения разработки сложных веб-приложений за счет использования компонентного подхода и виртуального DOM. React широко используется в индустрии благодаря своей производительности, гибкости и активному сообществу разработчиков.

Основой React является концепция компонентов. Компоненты представляют собой независимые, повторно используемые блоки кода, которые описывают часть пользовательского интерфейса. Каждый компонент может содержать в себе логику, структуру и стиль, что делает их модульными и легко поддерживаемыми. Компоненты могут быть как функциональными (основанными на функциях), так и классовыми (основанными на классах).

1.3 Библиотека Recoil

Recoil — это библиотека для управления состоянием, разработанная командой Facebook, которая интегрируется с React и решает некоторые проблемы, связанные с масштабированием состояния в больших приложениях. Recoil предлагает простой и гибкий способ управления глобальным состоянием, сохраняя при этом производительность и удобство работы с локальным состоянием. Основные концепции Recoil включают атомы, селекторы и асинхронное состояние, каждая из которых играет важную роль в управлении состоянием приложения.

Атомы представляют собой минимальные единицы состояния в Recoil. Они могут быть использованы любым количеством компонентов, которые могут подписываться на изменения этих атомов. Когда атом изменяется, все подписанные компоненты автоматически обновляются, что обеспечивает синхронность состояния между различными частями приложения. Атомы могут хранить любые данные — от простых чисел и строк до сложных объектов и массивов. Это делает их универсальными для различных задач управления состоянием.

Селекторы позволяют создавать производные состояния на основе атомов или других селекторов. Они вычисляют значения динамически, что делает управление зависимыми данными простым и эффективным. Например, селектор может вычислять итоговую сумму корзины покупок на основе атомов, представляющих отдельные элементы корзины. Селекторы также автоматически кэшируют свои значения, что улучшает производительность за счет избегания ненужных повторных вычислений. Это особенно полезно для сложных вычислений, которые могут требовать значительных ресурсов.

1.4 Контейнеризация приложения (Docker)

Контейнеризация стала ключевой технологией в современной разработке и развертывании приложений. Одним из самых популярных инструментов для контейнеризации является Docker. Docker позволяет создавать, разворачивать и

управлять контейнерами, которые изолируют приложение и все его зависимости в единую среду, обеспечивая консистентность и переносимость.

1.5 Стартап как диплом

Пользовательская часть сайта, разработанная в рамках данной бакалаврской работы, предназначалась для университетского проекта «**EduPlay**», который участвует в программе «**Стартап как диплом**».

«**Стартап как диплом**» — это программа, которая направлена на вовлечение талантливых студентов в развитие экосистемы технологического предпринимательства, а также на поддержку бизнеса, находящегося на начальной стадии. Разработка и реализация программы обучения абитуриентов и преподавателей университетов в подготовке стартапов в качестве ВКР предусмотрены программой «Цифровая экономика».

EduPlay — это веб-портал с образовательными играми для учеников младших классов и детей дошкольного возраста, который разрабатывается командой студентов.

1.6 Структура приложения

Приложение разработано с использованием современных технологий и лучших практик веб-разработки, что обеспечивает его гибкость, масштабируемость и удобство использования.

Оно состоит из нескольких ключевых модулей, каждый из которых отвечает за определенные функции:

1. **React:** используются функциональные компоненты для построения пользовательского интерфейса. Каждый компонент отвечает за определенный участок страницы.
2. **Управление состоянием с помощью Recoil:** позволяет легко управлять глобальным состоянием приложения и обеспечивать реактивность.
3. **Работа с API при помощи Axios:** обеспечивает получение и отправку данных на сервер. Включают в себя авторизацию, работу с пользователями и игровыми данными.
4. **Роутинг с использованием React Router:** позволяет создавать динамичные и многостраничные приложения в рамках SPA.
5. **Инструменты сборки и разработки:**

- **Vite:** используется для быстрой и эффективной сборки приложения, обеспечивая минимальное время отклика и упрощенную настройку проекта.
- **Docker:** позволяет контейнеризировать приложение для его деплоя и масштабирования.

1.7 Сервис для авторизации

Сервис для авторизации предназначен для выполнения операций входа, регистрации и получения данных текущего пользователя.

Авторизация

Авторизация осуществляется отправкой POST-запроса на путь `/Auth/Login`, передавая имя пользователя и пароль. В случае успешного запроса возвращается объект с токеном.

Регистрация

Регистрация работает точно таким же образом, только в запрос на путь `/Auth/Register`, помимо имени пользователя и пароля, передётся ещё и `email`.

1.8 Сервис для пользователей и их взаимодействия

Сервис для пользователей и их взаимодействия охватывает широкий спектр операций, включая получение данных пользователей, управление друзьями и обновление информации профиля.

Основные функции сервиса

1. Получение данных пользователя

Функции для получения данных пользователя по идентификатору или имени пользователя отправляют GET-запросы на соответствующие конечные точки API и возвращают информацию о пользователе, включая его профильные данные и текущий статус.

2. Управление друзьями

Функции для управления друзьями позволяют выполнять операции по добавлению, удалению, принятию заявок в друзья и получению списка друзей. Эти функции используют различные HTTP-методы (POST, DELETE, PATCH) для взаимодействия с API.

3. Получение списка компаньонов

Функция для получения списка компаньонов пользователя отправляет

GET-запрос на соответствующую конечную точку API и возвращает данные о всех компаньонах, связанных с пользователем.

4. Обновление профиля

Функции для обновления профиля пользователя позволяют изменять различные данные профиля, такие как имя, фамилия, дата рождения, имя пользователя и пароль. Эти функции отправляют PUT-запросы на соответствующие конечные точки API с новыми значениями.

5. Получение и обновление прогресса и результатов

Функции для получения лучших результатов пользователя и его прогресса отправляют GET-запросы и возвращают соответствующие данные. Эти функции позволяют отслеживать достижения и прогресс пользователя в образовательных играх.

6. Управление статусом дружбы

Функции для получения и управления статусом дружбы между пользователями отправляют GET-запросы для получения текущего статуса дружбы и позволяют управлять заявками в друзья.

7. Загрузка и обновление аватара

Функция для загрузки и обновления аватара пользователя отправляет PUT-запрос с данными формы, содержащими изображение профиля пользователя.

1.9 Сервис для игр

Сервис для игр работает аналогично предыдущим двум сервисам, предоставляя функции для получения списка игр, тем, информации об игре и обновления результатов игры.

1.10 Управление состоянием через Recoil

Управление состоянием в приложении реализовано с использованием библиотеки Recoil. Используются различные атомы для управления состоянием, охватывающим различные аспекты, такие как данные пользователя, списки игр и тем, состояния загрузки и параметры поиска. Также используется селектор `filteredGameState`, который отвечает за фильтрацию списка игр на основе критериев поиска и выбранной темы. Это позволяет компонентам получать доступ к уже отфильтрованному списку игр без необходимости реализовывать логику фильтрации непосредственно в компонентах.

1.11 Компонент «Header»

Компонент Header отвечает за отображение верхней части интерфейса приложения, включая логотип, меню навигации и кнопки для авторизации/регистрации или профиль пользователя, если он авторизован.

Меню навигации в заголовке предоставляет пользователю ссылки для перехода между основными страницами приложения. Меню может содержать как общедоступные элементы, так и элементы, доступные только авторизованным пользователям.



Рисунок 2 – Header

1.12 Страница со списком игр

Компонент поиска на странице с играми предоставляет пользователям возможность быстро находить нужные игры по их названиям.

Компонент сортировки на странице с играми позволяет пользователям фильтровать игры по различным темам.

Если список игр загружен, но при выборе темы или поисковом запросе массив пустой, отображается блок «Не найдено».

Итоговый вид страницы со списком игр представлен на рис. 3.

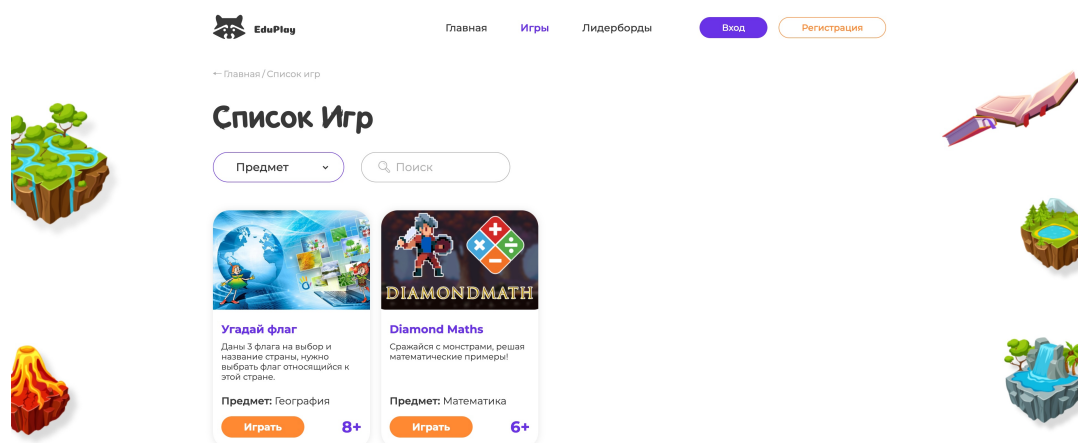


Рисунок 3 – Страница со списком игр

1.13 Страница игры

На странице игры происходит интеграция с играми, которые хранятся на другом хостинге. Эти игры загружаются в приложение через элемент `iframe`, что позволяет встраивать их содержимое на страницу.

Механизм взаимодействия

Для обеспечения взаимодействия между играми в `iframe` и React-приложением используется механизм `postMessage`. Этот метод позволяет передавать сообщения между окном, содержащим `iframe`, и окном самого `iframe`. В результате React-приложение может отправлять команды игре и получать от нее различные данные, такие как результаты, прогресс и другие события.



Рисунок 4 – Страница с игрой

1.14 Страница профиля

На странице профиля каждого пользователя отображается:

— Общая информация

- **Никнейм:** уникальное имя пользователя, используемое для идентификации в системе
- **Имя**
- **Фамилия**
- **Возраст**
- **Лучший результат в игре**

— Прогресс по предметам

Прогресс пользователя в различных предметах, которые он изучает через игры.

— Выбранный компаньон

Пользователь может выбрать компаньона, который будет отображаться на его странице профиля.

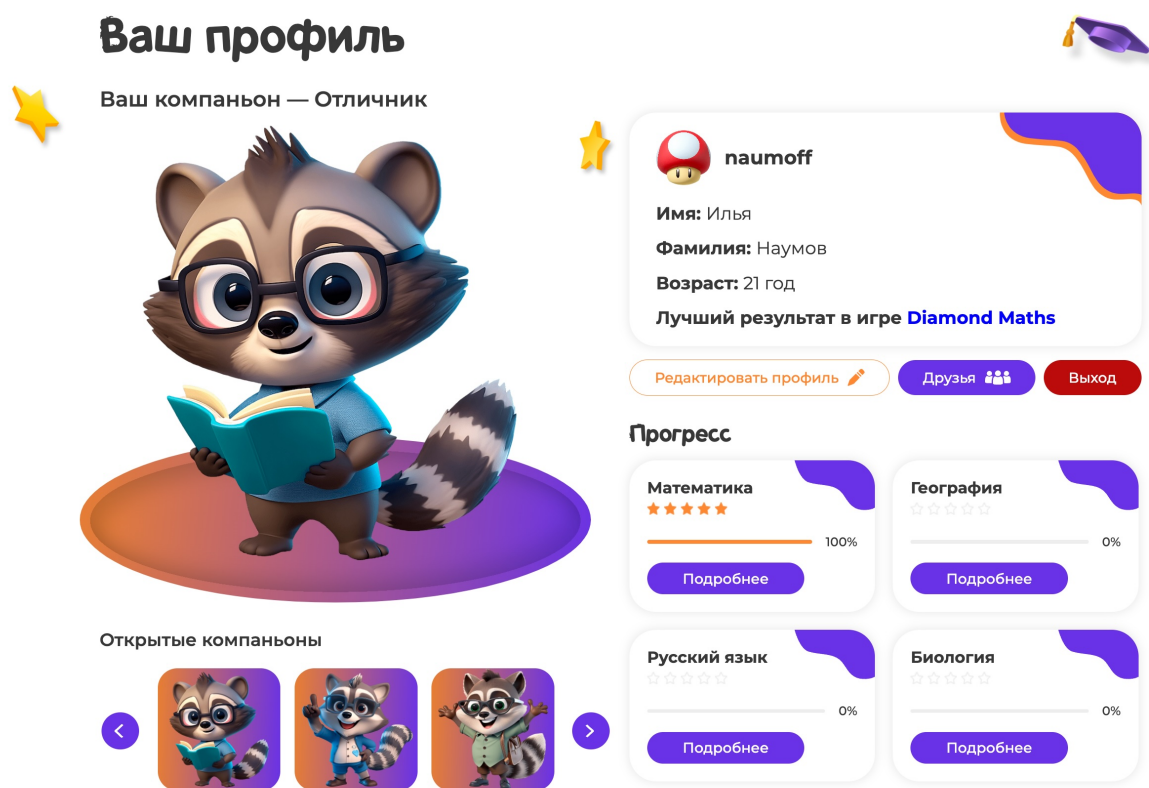


Рисунок 5 – Профиль пользователя

1.15 Контейнеризация и деплой приложения

Для контейнеризации и деплоя приложения используется `Dockerfile` для сборки и запуска приложения, а также `docker-compose.yml` для настройки Nginx в контейнере.

ЗАКЛЮЧЕНИЕ

В ходе данной работы была разработана пользовательская часть для сайта с образовательными играми для детей.

Для достижения поставленной цели в рамках работы:

1. были изучены современные тенденции в разработке образовательных приложений;
2. проанализирован технологический стек;
3. спроектирована архитектура пользовательского интерфейса;
4. реализованы интерактивные компоненты с применением современных библиотек;
5. проведена интеграция с серверной частью для обработки данных;
6. опубликовано приложение на сервере.

Таким образом, все поставленные в рамках работы задачи были выполнены.

В ходе анализа современных образовательных приложений были выявлены ключевые концепции и тенденции, а также важность интеграции игровых элементов в учебный процесс. Разработанное веб-приложение представляет собой интуитивно понятный и привлекательный интерфейс, который способствует эффективному взаимодействию пользователей с образовательным контентом.

Разработка удобных и функциональных интерфейсов для образовательных приложений играет важную роль в современном образовании, делая процесс обучения более доступным и захватывающим.