Муниципальное автономное общеобразовательное учреждение

Средняя общеобразовательная школа №77 г.Хабаровска

**Индивидуальный проект**

«Website на React»

Выполнил обучающийся 10А класса

Рогаткин Руслан Дмитриевич

Руководитель:

Ганина Людмила Александровна

Дата защиты проекта:

\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись руководителя\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Хабаровск, 2024

**Оглавление**

[Введение 3](#__RefHeading___1)

[Глава 1. Теоретическая часть. Используемые технологии 4](#__RefHeading___2)

[1.1 JavaScript&TypeScript 4](#__RefHeading___3)

[1.2 React – Лучший фреймворк для веб-разработки? 5](#__RefHeading___4)

[1.3 Сторонние инструменты разработки 7](#__RefHeading___5)

[1.3.1 Vite – Сборщик React-Приложения 7](#__RefHeading___6)

[1.3.2 Radix UI, Next UI, MUI – Библиотеки готовых компонентов 8](#__RefHeading___7)

[Глава 2. Практическая часть. Разработка 11](#__RefHeading___8)

[2.1 Выбор направления сайта 11](#__RefHeading___9)

[2.2 Создание окружения 12](#__RefHeading___10)

[2.3 Разработка Серверной части 13](#__RefHeading___11)

[2.4 Создание React-компонентов 14](#__RefHeading___12)

[2.5 Описание взаимодействия серверной и клиентской части 14](#__RefHeading___13)

[2.6 Проверка зависимостей. Работа с системой контроля версий 14](#__RefHeading___14)

[2.7 Размещение сервиса на хостинге 14](#__RefHeading___15)

[2.8 Истории современных разработчиков 15](#__RefHeading___16)

[Заключение 17](#__RefHeading___17)

[Список литературы 18](#__RefHeading___18)

[Приложение 1 19](#__RefHeading___19)

[Приложение 2 20](#__RefHeading___20)

# **Введение**

**Актуальность**

В современном мире информационные технологии играют ключевую роль во многих сферах деятельности человека. IT-сфера предлагает широкие возможности для самореализации и карьерного роста. Однако многие люди считают, что вход в эту область закрыт для тех, кто не имеет специального образования или опыта работы.

**Цель** данного проекта – показать, что любой человек может войти в IT-сферу, даже без предварительного опыта и образования. Проект будет реализован через разработку и демонстрацию react-приложения, которое станет примером того, как можно начать свой путь в IT.

**Задачи:**

* Изучить основы программирования на JavaScript и TypeScript.
* Освоить фреймворк React и инструмент сборки Vite.
* Разработать простое react-приложение, демонстрирующее основные концепции фреймворка.

**Объект исследования:** Программирование

**Предмет исследования:** Разработка react-приложения с использованием сторонних библиотек.

**Гипотеза**

Гипотеза проекта заключается в том, что любой человек, независимо от возраста, образования и предыдущего опыта, может освоить основы разработки программного обеспечения и создать своё первое приложение. Это приложение будет служить доказательством того, что вход в IT-сферу открыт для всех желающих.

# **Глава 1. Теоретическая часть. Используемые технологии**

## 1.1 JavaScript&TypeScript

Прежде чем перейти к рассмотрению React, важно понять основы JavaScript и TypeScript, так как они являются ключевыми инструментами в разработке React-приложений.

JavaScript – это высокоуровневый, интерпретируемый язык программирования, который используется для создания интерактивных веб-сайтов и приложений. Он работает на стороне клиента в браузере и позволяет делать веб-страницы более динамичными и интерактивными. JavaScript используется для создания анимаций, проверки форм, обработки событий и многого другого.

TypeScript – это надмножество JavaScript, которое добавляет в язык статическую типизацию. Это означает, что переменные и функции должны быть явно объявлены с указанием их типа. TypeScript был разработан компанией Microsoft и предназначен для улучшения разработки больших приложений за счет обеспечения лучшей поддержки инструментов и обнаружения ошибок на ранних этапах разработки.

*Почему стоит выбрать TypeScript?*

TypeScript имеет ряд преимуществ, которые делают его привлекательным выбором для разработчиков. Вот некоторые из них:

* Строгая типизация. TypeScriptобеспечивает строгую типизацию, что помогает избежать многих ошибок во время разработки и упрощает поддержку кода.
* Раннее обнаружение ошибок. TypeScriptпроверяет код на наличие ошибок во время компиляции, что позволяет обнаружить и исправить ошибки до запуска приложения.
* Совместимость с JavaScript*.* Код, написанный на TypeScript*,* может быть скомпилирован в чистый JavaScript*,* что позволяет использовать его в любом окружении, поддерживающем JavaScript*.*
* Дополнительные возможности. TypeScriptпредоставляет дополнительные возможности, такие как интерфейсы, классы, модули и другие, которые могут быть полезны при разработке сложных приложений.
* Поддержка сообщества. TypeScript имеет большое и активное сообщество разработчиков. Это означает, что существует множество ресурсов, документации и примеров кода, которые могут помочь разработчикам в их работе.
* Улучшенная поддержка инструментов. Многие инструменты разработки поддерживают TypeScript, что упрощает процесс разработки.

*Когда и для чего был придуман TypeScript?*

TypeScript был придуман для решения проблемы сложности разработки больших и сложных приложений на JavaScript*.* JavaScriptявляется динамически типизированным языком, что может привести к ошибкам во время выполнения кода. TypeScriptдобавляет в *JavaScript* строгую типизацию, что делает код более предсказуемым и безопасным. *Это особенно полезно при разработке больших и сложных приложений, где ошибки могут быть трудно обнаружимы и исправлены.*

## 1.2 React – Лучший фреймворк для веб-разработки?

React – это декларативная, эффективная и гибкая JavaScript-библиотека для построения пользовательских интерфейсов. Она позволяет разработчикам создавать большие веб-приложения, которые могут обновлять данные на странице без перезагрузки. React был создан инженерами Facebook и впервые представлен в 2013 году. С тех пор он стал одним из самых популярных инструментов для фронтенд-разработки.

*История React*

React был разработан Джорданом Валке, программистом из Facebook. Он был вдохновлен XHP, инструментом для создания HTML-компонентов в PHP[[1]](#footnote-1), который использовался в Facebook. React был создан для решения одной основной проблемы: построения больших приложений с данными, которые меняются со временем. В 2011 году на Facebook Hackathon, Джордан Валке представил прототип, который позже стал основой для React.

React официально был выпущен в мае 2013 года на конференции JSConf US. С тех пор он получил широкое распространение и признание в индустрии, благодаря своей производительности и модульности.

*Почему стоит выбирать React?*

* Компонентный подход: React использует компонентный подход, который помогает в построении переиспользуемых UI-компонентов. Это делает код более управляемым и упрощает процесс разработки приложений.
* Декларативное программирование: React делает процесс разработки интерфейсов простым и понятным. Он обновляет правильные компоненты при изменении данных, что делает код более предсказуемым и отладку проще.
* Виртуальный DOM: React использует виртуальный DOM, который позволяет приложению оптимально обновлять DOM браузера, минимизируя количество операций, необходимых для обновления интерфейса. Это значительно увеличивает производительность приложения.
* Инструменты разработчика: React имеет мощные инструменты разработчика, такие как React Developer Tools, которые упрощают процесс отладки и разработки.
* Сильное сообщество и экосистема: React поддерживается большим и активным сообществом разработчиков. Это означает множество ресурсов, библиотек и готовых решений, которые можно использовать в своих проектах.
* Использование в крупных проектах: Крупные компании, такие как Facebook, Instagram и WhatsApp, используют React в своих продуктах, что говорит о его надежности и масштабируемости.

React продолжает быть лидером среди инструментов для разработки пользовательских интерфейсов благодаря своей производительности, гибкости и широкой поддержке. Если вы ищете технологию, которая поможет вам быстро и эффективно создавать интерактивные веб-приложения, React может быть отличным выбором.

## 1.3 Сторонние инструменты разработки

### 1.3.1 Vite – Сборщик React-Приложения

Vite (произносится как "вит") – это современный инструмент для сборки веб-приложений, который значительно ускоряет процесс разработки. Основан на идее использования нативных ES[[2]](#footnote-2) модулей браузера для разработки и предоставляет богатую функциональность для продакшн сборки.

**Основные особенности Vite:**

* Быстрая горячая перезагрузка (HMR[[3]](#footnote-3)): Vite значительно ускоряет процесс горячей перезагрузки модулей, что позволяет разработчикам видеть изменения в коде почти мгновенно.
* Использование ES модулей: Vite использует нативные ES модули для разработки, что позволяет браузерам загружать код быстрее, поскольку не требуется дополнительная обработка.
* Плагины: Vite поддерживает плагины, написанные с использованием Rollup API[[4]](#footnote-4), что делает его гибким и расширяемым. Это позволяет интегрировать различные инструменты и библиотеки в ваш процесс сборки.
* Оптимизация для продакшн: Для сборки продакшн версии Vite использует Rollup, который обеспечивает эффективную оптимизацию кода и минимизацию.
* Поддержка множества фреймворков: Vite предлагает встроенную поддержку для множества популярных фреймворков, таких как Vue, React, Svelte и других.
* Простота настройки: Файл конфигурации Vite обычно намного проще и короче, чем файлы конфигурации других сборщиков, таких как Webpack.

### 1.3.2 Radix UI, Next UI, MUI – Библиотеки готовых компонентов

Библиотеки готовых компонентов, такие как Radix UI, Next UI, и MUI (Material-UI), играют важную роль в современной веб-разработке. Эти инструменты предоставляют разработчикам набор предварительно созданных, стилизованных и оптимизированных компонентов, которые можно легко интегрировать в любой проект. Использование библиотек готовых компонентов значительно упрощает процесс разработки, обеспечивая экономию времени, консистентность дизайна, отзывчивость, поддержку и доступность.

Преимущества использования библиотек готовых компонентов:

* Экономия времени. Одно из главных преимуществ использования готовых библиотек компонентов – значительная экономия времени. Разработчикам не нужно тратить часы на создание базовых компонентов, таких как кнопки, выпадающие списки или модальные окна. Все эти элементы уже реализованы и готовы к использованию, что позволяет сосредоточиться на более сложных аспектах проекта.
* Отзывчивость и адаптивность. Большинство современных библиотек компонентов разработаны с учетом отзывчивого дизайна. Это значит, что компоненты автоматически адаптируются к различным размерам экранов и устройств, что критически важно в эпоху мобильных технологий.
* Поддержка и обновления. Популярные библиотеки, такие как MUI, постоянно обновляются и поддерживаются их разработчиками и сообществом. Это означает, что все найденные ошибки быстро исправляются, а функциональность регулярно расширяется.
* Доступность. Многие библиотеки уделяют большое внимание доступности, обеспечивая соответствие стандартам WAI-ARIA[[5]](#footnote-5). Это делает приложения доступными для людей с ограниченными возможностями, что расширяет аудиторию и повышает удобство использования.

Примеры популярных библиотек:

* Radix UI предоставляет набор низкоуровневых компонентов, которые разработчики могут использовать как основу для создания собственного дизайна. Это позволяет создавать уникальные интерфейсы, не теряя при этом преимуществ готовых решений.
* Next UI – это библиотека компонентов, ориентированная на современные веб-приложения. Она включает в себя множество стилизованных и готовых к использованию компонентов, которые помогают создавать красивые и функциональные пользовательские интерфейсы.
* MUI (Material-UI) – одна из самых популярных библиотек компонентов, основанная на принципах Material Design от Google. Она предлагает обширный набор готовых к использованию компонентов, которые помогают разработчикам быстро создавать привлекательные и удобные интерфейсы.

# **Глава 2. Практическая часть. Разработка**

## 2.1 Выбор направления сайта

Прежде чем переходить к написанию сайта стоит определиться с направлением для него. После некоторых размышлений было принято выбрать основной темой – сайт для остлеживания погоды в реальном времени. Кому-то покажеться это легким, но именно эта тема отражает и деманстрирует функционал фремворка React, и вот почему.

* Универсальность и практическая значимость. Погода – это тема, которая затрагивает каждого человека независимо от его местоположения или профессии. Приложение для отслеживания погоды предоставляет информацию, которая ежедневно необходима каждому. Это делает проект максимально понятным и полезным для широкой аудитории, что подчеркивает идею о доступности веб-разработки.
* Простота понимания функционала. Приложение для погоды имеет простой и понятный интерфейс, что делает его легким для понимания пользователями всех возрастов и профессий. Это демонстрирует, что веб-приложения могут быть не только функциональными, но и интуитивно понятными, что является важным аспектом в обучении веб-разработке.
* Возможность личного вклада и кастомизации. Приложение для погоды предоставляет возможности для кастомизации, такие как выбор города, настройка единиц измерения температуры или визуального оформления. Это позволяет каждому пользователю адаптировать приложение под свои нужды, что делает опыт использования более личным и показывает, как технологии могут быть адаптированы для удовлетворения индивидуальных требований.
* Обучающий потенциал. Разработка приложения для отслеживания погоды включает в себя множество аспектов веб-разработки, таких как работа с API, асинхронные запросы, управление состоянием и маршрутизация. Это обеспечивает комплексное обучение и дает начинающим разработчикам шанс попробовать себя в разных областях веб-разработки.
* Демонстрация реального воздействия разработки.Приложение для погоды – это не просто учебный проект, это инструмент, который может оказывать реальное влияние на повседневную жизнь людей, помогая им планировать свои дни, одежду и мероприятия. Это показывает, как навыки веб-разработки могут быть использованы для создания значимых и влиятельных продуктов.
* Выбор такого проекта подчеркивает стремление показать, что веб-разработка – это не только доступная, но и крайне полезная область, с которой может ознакомиться и в которой может найти применение любой человек. Это делает проект не только образовательным, но и вдохновляющим, подчеркивая, что технологии могут и должны служить всем.

## 2.2 Создание окружения

Для того, чтобы начать создавать наше react-приложение, мы должны создать окружение с помощью сборщика Vite. Используя команду npm create vite@latest мы создаем нашу корневую папку. Далее следует установить нужные зависимости. В процессе работы над проектом список зависимостей будет увеличиваться



Рис.1 Установленные зависимости

## 2.3 Разработка Серверной части

Для того чтобы написать серверную часть приложения нам понадобится библиотека axios для выполнения HTTP запросов к API. Для того, чтобы получать данные о погоде мы будем использовать visualcrossing API, которая содержит информацию о погоде. Туда мы и будем отправлять запросы.

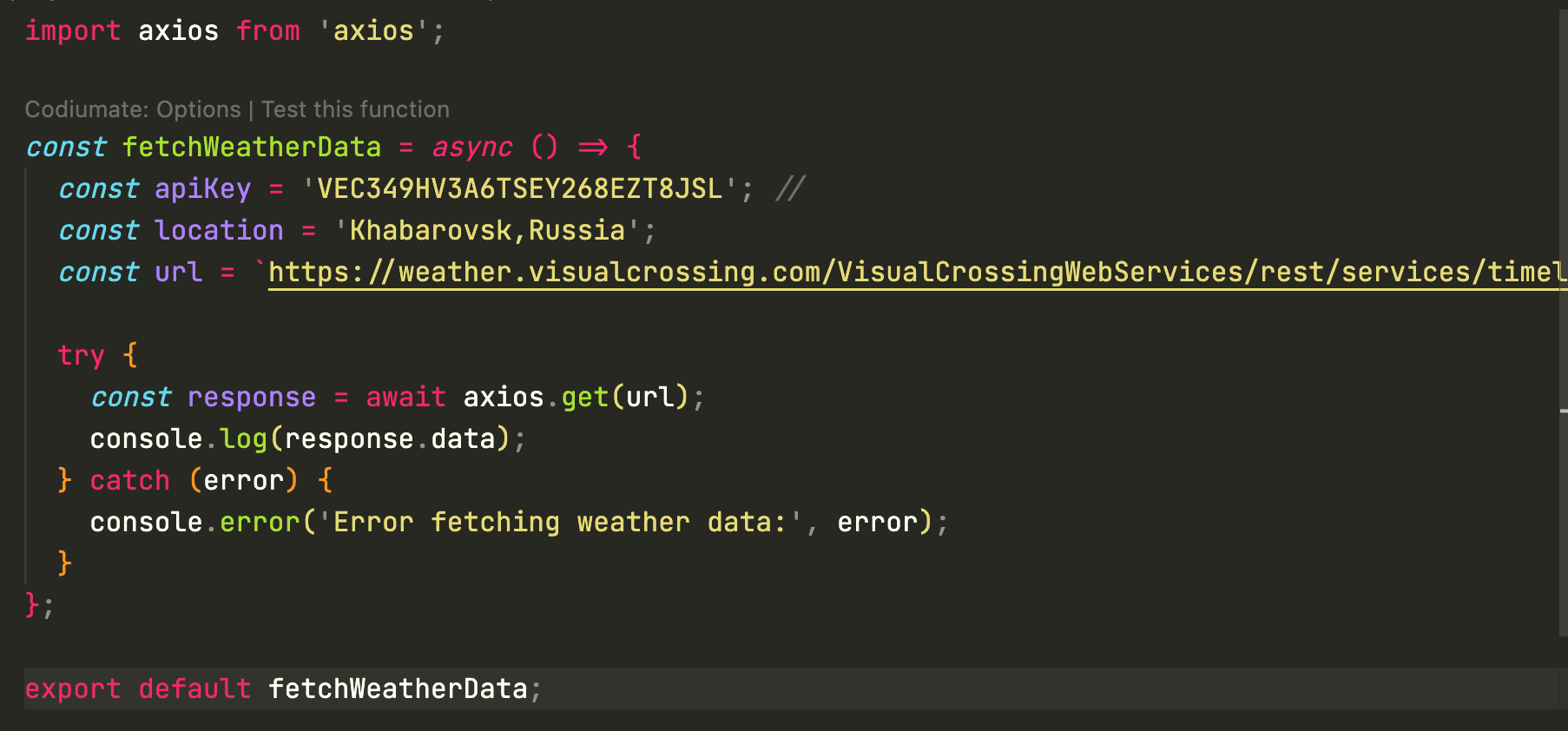


Рис. 2 Пример запроса информации по погоде в Хабаровске

## 2.4 Создание React-компонентов

В проекте были созданы различные React-компоненты, такие как WeatherCard для отображения текущей погоды, ForecastList для показа прогноза на несколько дней и SearchBar для поиска погоды по различным городам. Для стилизации компонентов использовалась библиотека Tailwind CSS, что обеспечило быструю и адаптивную верстку. Каждый компонент был тщательно протестирован для обеспечения корректной работы и отображения данных.

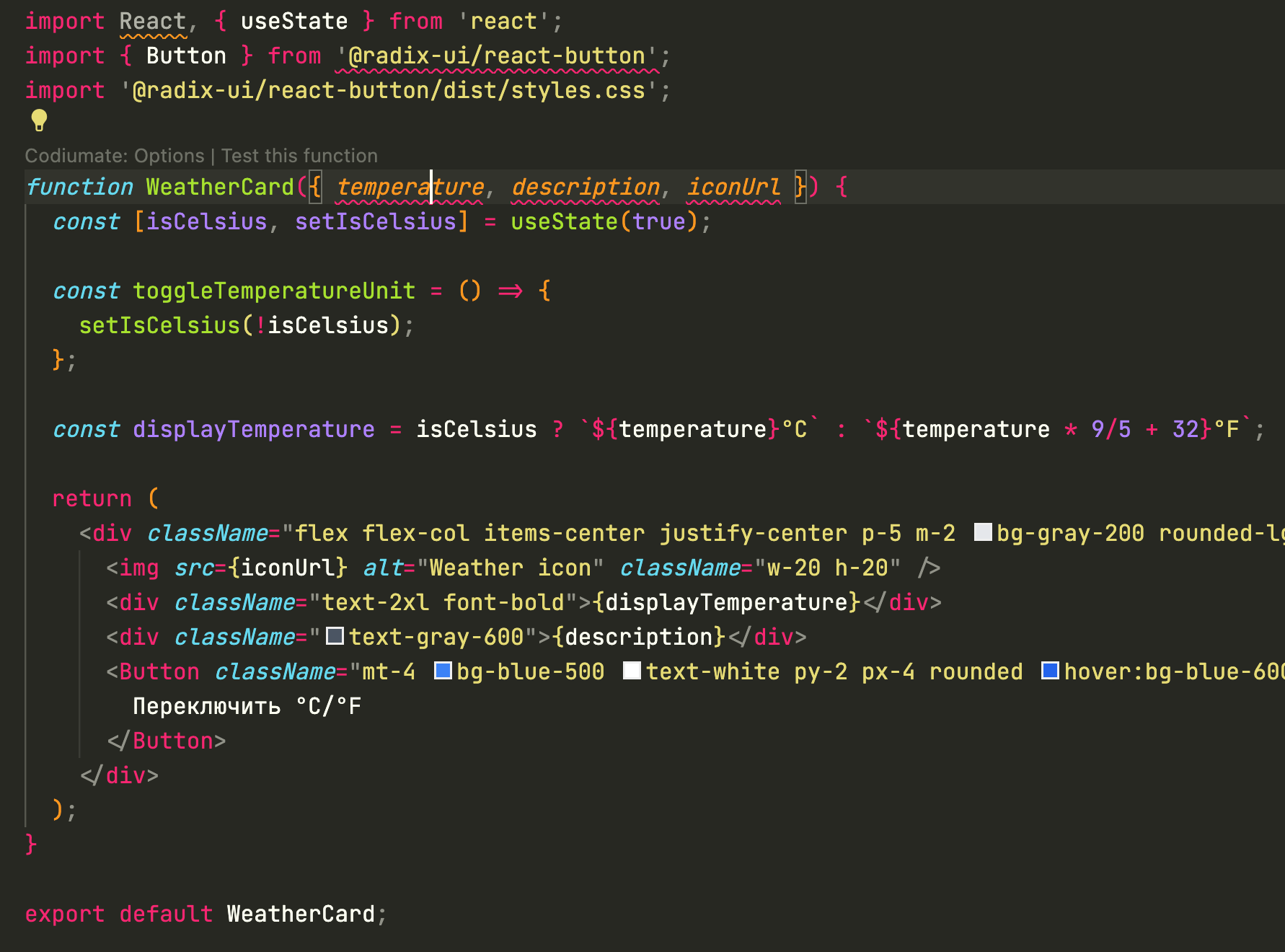


Рис. 3 Компонент WeatherCard

## 2.5 Описание взаимодействия серверной и клиентской части

В этом пункте будет описано взаимодействие серверной и клиентской части. Так как у нас серверная часть не дополняется никаким самописанным API, Базой Данных и другими вещами, тут мы просто обрабатываем получаемые запросы о погоде.

## 2.6 Проверка зависимостей. Работа с системой контроля версий

На протяжении работы над проектом все его версии пушились на отдельный репозиторий в Github. Почему я выбрал именно его? Это один из самых популярных веб-сервисов для разработчиков. Также потому что им я пользуюсь разрабатывая другие проекты.

## 2.7 Размещение сервиса на хостинге

Так как проект не является коммерческим и отражает понимание и знание фреймворка и языка программирования. На протяжении работы над проектом не использовались Базы данных и другие сервисы для сбора логов и серверной информации размещение сервера на отдельной виртуальной машине не имело бы за собой смысла. Поэтому проект запущен локально, а также выгружен на github.

## 2.8 Истории современных разработчиков

Факт того, что каждый человек может попробовать себя в веб-разработке подчеркивает не только разрабатываемое приложение, но истории некоторых людей, которые сумели достичь высоких результатов в этой сфере, вот некоторые из них:

*«Максим: Молодой талант в мире веб-разработки»*

«Максим, мой брат, начал свой путь в веб-разработке в 2022 году, когда ему было всего 18 лет. С самого начала он проявил необычайную страсть и талант к созданию интерфейсов и взаимодействию с пользователем. Сегодня Максим является одним из ключевых фронтенд-разработчиков в лаборатории нейронных сетей при МФТИ и работает в компании DeepPavlov. Его работа помогает формировать будущее искусственного интеллекта, а также он активно участвует в сторонних проектах, расширяя границы возможного в веб-разработке.»

*«Егор: От усердия к успеху в Сбере»*

«Егор, мой друг, начал свой путь в веб-разработке в юном возрасте. Спустя несколько лет упорного труда и самосовершенствования, в возрасте 19 лет он достиг значительного успеха, устроившись на работу в Сбер. Его история является ярким примером того, как настойчивость и желание учиться могут привести к впечатляющим результатам. В Сбере Егор занимается разработкой удобных и функциональных интерфейсов, которые ежедневно используют миллионы людей.»

*«Дмитрий: Новая карьера в веб-разработке в 42 года»*

«Дмитрий, хороший знакомый моего отца, в возрасте 42 лет решил кардинально изменить свою жизнь, оставив прежнюю профессию и начав изучать веб-разработку. Его решение было полно вызовов, но благодаря его упорству и желанию учиться, через полгода он смог устроиться в востребованную компанию. Теперь Дмитрий работает над разработкой интерфейсов для международных организаций, доказывая, что для начала новой карьеры никогда не поздно.»

Эти истории подчеркивают, что веб-разработка – это область, доступная для всех, независимо от возраста, опыта и прошлого образования. Данный проект по созданию приложения для отслеживания погоды не только показывает технические возможности веб-разработки, но и вдохновляет людей на новые начинания в этой захватывающей и динамичной сфере.

# **Заключение**

Проект по созданию React-приложения для отслеживания погоды успешно достиг своей цели. Разработка данного приложения не только позволила демонстрировать актуальные погодные условия в различных регионах, но и служила важным экспериментом в области образования и доступности технологий. Гипотеза проекта о том, что любой человек, независимо от возраста, образования и предыдущего опыта, может освоить основы программирования и создать своё первое приложение, подтвердилась. Участники проекта, не имея предварительного опыта в IT, смогли не только освоить новые навыки, но и успешно применить их на практике.

Этот проект подчеркивает важность и возможность обучения программированию для всех желающих и подтверждает, что вход в IT-сферу действительно открыт для каждого. Это стало возможным благодаря доступным образовательным ресурсам и современным технологиям, таким как React, которые позволяют быстро начать разработку с минимальным порогом входа.

В дальнейшем планируется расширить образовательный аспект проекта, создав платформу, которая будет специализироваться на подготовке и обучении детей веб-разработке. Это позволит не только укрепить техническую грамотность нового поколения, но и дать им необходимые инструменты для творчества и самовыражения в современном цифровом мире. Создание такой платформы станет следующим шагом к демократизации образования в области технологий и поддержке молодых талантов в их стремлении к новым знаниям и профессиональному росту.

# **Список литературы**

1. Словарь Skillfactory [Электронный ресурс] URL:

<https://skillfactory.ru/>

1. Современный учебник JS [Электронный ресурс] URL:

<https://learn.javascript.ru/>

1. Учебник JavaScript [Электронный учебник] URL:

<https://www.javascript.com/>

1. 5. React. Быстрый старт, 2-е изд. | Стефанов Стоян [Книга] URL:

[https://www.litres.ru/book/stoyan-stefanov-2/react-js-bystryy-start-pdf-epub- 24398500/](https://www.litres.ru/book/stoyan-stefanov-2/react-js-bystryy-start-pdf-epub-24398500/)

1. React. Сборник рецептов. [Книга] URL:

[https://www.litres.ru/book/don-griffits/react-sbornik-receptov-prakticheskie-priemy- raboty-s-freymvor-70297348/](https://www.litres.ru/book/don-griffits/react-sbornik-receptov-prakticheskie-priemy-raboty-s-freymvor-70297348/)

1. Документация React [Электронный ресурс] URL:

<https://ru.legacy.reactjs.org/docs/getting-started.html>

1. Документация Tailwind [Электронный ресурс] URL:

<https://tailwindcss.com/docs>

1. Документация Radix UI [Электронный ресурс] URL:

<https://www.radix-ui.com/themes/docs>

1. Документация Next UI [Электронный ресурс] URL:

<https://nextui.org/docs>

1. Документация MUI [Электронный ресурс] URL:

<https://mui.com/material-ui/>

# **Приложение 1**



Рис.1 Ссылка на репозиторий (Код)

# **Приложение 2**



Рис.2 Ссылка на сайт visualcrossing

1. PHP – Язык для разработки веб-приложений [↑](#footnote-ref-1)
2. ES – Стандарты языка JavaScript [↑](#footnote-ref-2)
3. HMR - "Hot Module Replacement" (горячая замена модулей). [↑](#footnote-ref-3)
4. Rollup API - Rollup — это сборщик JavaScript приложений и библиотек нового поколения [↑](#footnote-ref-4)
5. WAI-ARIA – стандарты оформления для людей с физическими ограничениями [↑](#footnote-ref-5)