**ВВЕДЕНИЕ**

Современный мир невозможно представить без онлайн-игр. Они стали неотъемлемой частью нашей жизни, предлагая развлечение, отдых и даже возможность заработка. С развитием технологий игры становятся все более доступными и разнообразными, привлекая к себе все больше пользователей.

Одним из популярных жанров онлайн-игр являются игры на совпадение (Match-Match). Простые правила, увлекательный геймплей и яркое оформление делают их привлекательными для людей всех возрастов.

В данной дипломной работе будет рассмотрена разработка Match-Match игры с использованием React и Redux.

React - это библиотека JavaScript для создания пользовательских интерфейсов, разработанная компанией Facebook (ныне Meta). Она была создана Джорданом Уолкером в 2011 году и впервые использовалась в ленте новостей Facebook.

React был разработан, чтобы упростить создание сложных пользовательских интерфейсов. Вместо того, чтобы манипулировать DOM напрямую, React использует виртуальный DOM, который представляет собой абстрактное представление пользовательского интерфейса. Когда состояние приложения изменяется, React сравнивает виртуальный DOM с реальным DOM и обновляет только те элементы, которые изменились. Это делает React очень быстрым и эффективным.

React был выпущен как проект с открытым исходным кодом в 2013 году и с тех пор стал одной из самых популярных библиотек JavaScript для создания пользовательских интерфейсов.

Redux - это библиотека JavaScript для управления состоянием приложения. Она была создана Дэном Абрамовым и Эндрю Кларком в 2015 году.

Redux вдохновлен архитектурой Flux, которая была разработана Facebook для управления состоянием приложений React. Redux упрощает Flux, делая его более предсказуемым и лаконичным.

Redux хранит состояние приложения в одном объекте, называемом хранилищем. Хранилище - это единственный источник правды для состояния приложения. Чтобы изменить состояние приложения, необходимо отправить действие. Действие - это объект, который описывает, что нужно изменить.

Сочетание React и Redux позволяет создавать Match-Match игры с динамичным и отзывчивым интерфейсом, а также эффективно управлять состоянием игры.

Актуальность темы обусловлена растущей популярностью онлайн-игр, в том числе Match-Match игр. Разработка таких игр с использованием современных технологий, таких как React и Redux, позволяет создавать привлекательные и удобные для пользователей игры, которые могут конкурировать с существующими аналогами.

Цель и задачи работы

Целью дипломной работы является разработка Match-Match игры с использованием React и Redux.

Для достижения этой цели необходимо решить следующие задачи:

* Изучить особенности Match-Match игр. В рамках этой задачи будет проведен анализ существующих Match-Match игр, выявлены их основные характеристики и особенности.
* Ознакомиться с библиотеками React и Redux. Будет изучена документация React и Redux, а также рассмотрены примеры их использования в разработке игр.
* Спроектировать архитектуру игры. На этом этапе будет определена структура игры, взаимодействие ее компонентов и потоки данных.
* Разработать и реализовать игровой процесс. Будут разработаны алгоритмы и механики игры, а также реализована логика игрового процесса.
* Создать пользовательский интерфейс. Будет разработан пользовательский интерфейс игры, который будет удобным и интуитивно понятным для пользователей.
* Протестировать и отладить игру. На этом этапе будет проведено тестирование игры с целью выявления и исправления ошибок.

В рамках дипломной работы будет проведено более глубокое изучение жанра Match-3 игр, а также рассмотрены особенности использования React и Redux для разработки игр.

React и Redux хорошо подходят для разработки Match-3 игр, поскольку они позволяют создавать динамичные и интерактивные интерфейсы, а также эффективно управлять состоянием игры. React позволяет легко создавать и отображать элементы игры, а Redux помогает отслеживать состояние игрового поля, счет и другие данные.

Использование React и Redux для разработки Match-Match позволяет создать современную, динамичную и увлекательную игру, которая будет интересна широкой аудитории.Ожидаемые результаты

В результате выполнения дипломной работы будет разработана Match-Match игра с использованием React и Redux. Игра будет обладать динамичным и отзывчивым интерфейсом, а также эффективным управлением состоянием игры.

Практическая значимость работы заключается в том, что разработанная игра может быть использована как самостоятельный продукт, так и в качестве основы для создания более сложных игр.

# ГЛАВА 1 АНАЛИЗ И ИССЛЕДОВАНИЕ TELEGRAM-BOT

# Исследование предметной области

# Match-3 игры, к которым относится и Match-Match, представляют собой популярный жанр казуальных видеоигр. Основная механика игры заключается в сопоставлении трех или более элементов одного цвета или типа, чтобы они исчезли с игрового поля. Match-3 игры привлекают игроков всех возрастов своей простотой, увлекательностью и доступностью.

# Популярность Match-3 игр обусловлена несколькими факторами:

# Легко освоить, но сложно стать мастером: Правила игры просты и интуитивно понятны, что делает их доступными для игроков любого уровня опыта. В то же время, достижение высоких результатов и прохождение сложных уровней требует стратегического мышления и планирования.

# Увлекательный игровой процесс: Match-3 игры вызывают привыкание, поскольку успешное сопоставление элементов приносит удовлетворение и побуждает продолжать игру. Разнообразие игровых механик, бонусов и уровней поддерживает интерес игроков.

# Доступность: Match-3 игры доступны на различных платформах, включая мобильные устройства, компьютеры и игровые консоли. Это позволяет игрокам наслаждаться игрой в любое время и в любом месте.

# Match-3 игры постоянно развиваются, в них появляются новые механики, визуальные эффекты и сюжетные линии. Использование React и Redux для создания Match-Match позволяет создать современную, динамичную и увлекательную игру, которая будет интересна широкой аудитории.

# В рамках дипломной работы будет проведено более глубокое изучение жанра Match-3 игр, а также рассмотрены особенности использования React и Redux для разработки игр.

# История Match-3 игр

# Первой Match-3 игрой считается Tetris, выпущенная в 1984 году. Хотя Tetris не является классической Match-3 игрой, он заложил основу для жанра, продемонстрировав увлекательность и потенциал игр, основанных на сопоставлении элементов.

# В 1994 году была выпущена игра Shariki, которая более точно соответствует современным представлениям о Match-3 играх. Shariki предлагала игрокам собирать шарики одного цвета в линии, чтобы они исчезали с игрового поля. Игра была простой, но увлекательной, и быстро приобрела популярность.

# В 2000-х годах жанр Match-3 игр получил дальнейшее развитие благодаря появлению новых технологий и платформ. Игры стали более сложными, с улучшенной графикой, звуком и геймплеем. Появились новые механики, такие как бонусы, специальные элементы и ограничения по времени.

# Одним из самых ярких примеров Match-3 игр является Candy Crush Saga, выпущенная в 2012 году. Candy Crush Saga предлагает игрокам проходить уровни, сопоставляя конфеты разных цветов. Игра отличается яркой графикой, увлекательным геймплеем и социальными элементами, что сделало ее одной из самых популярных мобильных игр всех времен.

# React и Redux для разработки Match-3 игр

# React и Redux — это популярные инструменты для разработки веб-приложений. React — это библиотека JavaScript для создания пользовательских интерфейсов, а Redux — это библиотека для управления состоянием приложения.

# React и Redux хорошо подходят для разработки Match-3 игр, поскольку они позволяют создавать динамичные и интерактивные интерфейсы, а также эффективно управлять состоянием игры. React позволяет легко создавать и отображать элементы игры, а Redux помогает отслеживать состояние игрового поля, счет и другие данные.

# Использование React и Redux для разработки Match-Match позволяет создать современную, динамичную и увлекательную игру, которая будет интересна широкой аудитории.

# 1.2. Классификация Match-Match игр

Match-Match игры, также известные как игры на запоминание или игры с парными карточками - это жанр казуальных видеоигр, в которых игрок должен найти пары одинаковых объектов на игровом поле. Обычно объекты представлены в виде карточек, которые перевернуты лицевой стороной вниз. Игрок должен переворачивать карточки по две, чтобы найти совпадающие пары.

Match-Match игры популярны благодаря своей простой, но увлекательной механике, которая подходит для игроков всех возрастов. Они развивают память, внимание и концентрацию, а также предлагают расслабляющий и увлекательный игровой процесс.

Существует множество различных типов Match-Match игр, которые можно классифицировать по нескольким параметрам:

1. По типу объектов:

* Карточки с изображениями: это классический тип Match-Match игры, где на карточках изображены различные картинки.
* Карточки с символами: на карточках изображены символы, такие как буквы, цифры или геометрические фигуры.
* Карточки со словами: на карточках написаны слова, и игрок должен найти пары слов, которые связаны между собой каким-либо образом.
* Карточки с аудио: на карточках воспроизводятся звуки, и игрок должен найти пары карточек с одинаковыми звуками.
* Трехмерные объекты: вместо карточек используются трехмерные объекты, которые игрок должен вращать, чтобы найти совпадающие пары.

2. По количеству игроков:

* Однопользовательские игры: игрок играет против компьютера.
* Многопользовательские игры: игрок играет против других игроков.

3. По цели:

* Найти все пары: цель игры - найти все пары карточек на игровом поле.
* Набрать очки: игрок получает очки за каждую найденную пару, и цель игры - набрать как можно больше очков.
* Ограничение по времени: игрок должен найти все пары карточек за ограниченное время.
* Выполнить задание: игрок должен выполнить определенное задание, например, найти все пары карточек определенного цвета или найти пары карточек в определенном порядке.

4. По стилю:

* Казуальный: Игры с простой графикой и расслабляющим геймплеем.
* Сюжетный: Игры с развивающимся сюжетом, который раскрывается по мере прохождения уровней.
* Головоломка: Игры, которые требуют от игрока стратегического мышления и планирования.

5. По платформе:

* Мобильные игры: Match-Match игры, разработанные для мобильных устройств.
* Игры для ПК: Match-Match игры, разработанные для персональных компьютеров.
* Онлайн-игры: Match-Match игры, в которые можно играть онлайн в браузере.

Match-Match игра, разрабатываемая в рамках данной дипломной работы, будет иметь следующую классификацию:

* По типу объектов: карточки с изображениями.
* По количеству игроков: однопользовательская игра.
* По цели: найти все пары и набрать очки.
* По стилю: казуальная игра с элементами головоломки.
* По платформе: веб-приложение, доступное через браузер.

Match-Match игры - это разнообразный и увлекательный жанр, который продолжает развиваться и предлагать новые игровые возможности.

Влияние Match-Match игр на развитие игровой индустрии:

* Match-Match игры оказали значительное влияние на развитие игровой индустрии, особенно в области мобильных игр.
* Доступность: Простая механика и увлекательный геймплей Match-Match игр делают их доступными для широкой аудитории, включая игроков, которые не являются заядлыми геймерами.
* Инновации: Разработчики Match-Match игр постоянно придумывают новые механики и функции, чтобы сделать свои игры более увлекательными и захватывающими.

Match-Match игры - это важная часть игровой индустрии, и они, вероятно, останутся популярными еще долгие годы.

Примеры Match-Match игр:

* Memory: Классическая Match-Match игра с карточками, на которых изображены картинки.
* Mahjong: Match-Match игра с использованием китайских иероглифов.
* Concentration: Match-Match игра, в которой игрок должен найти пары карточек с одинаковыми словами.
* Animal Pairs: Match-Match игра, в которой игрок должен найти пары карточек с изображениями животных.
* Matching Madness: Match-Match игра с различными уровнями сложности и бонусами.

Будущее Match-Match игр:

* Новые механики: Разработчики постоянно придумывают новые механики, чтобы сделать Match-Match игры более интересными. Например, некоторые Match-Match игры теперь включают в себя элементы головоломки, такие как ограничение по времени или количеству ходов.
* Новые технологии: Новые технологии, такие как виртуальная реальность и дополненная реальность, могут быть использованы для создания более захватывающих Match-Match игр.
* Интеграция с другими жанрами: Match-Match игры могут быть интегрированы с другими жанрами, такими как RPG, стратегии и головоломки, чтобы создать новые и уникальные игровые возможности.

Match-Match игры - это увлекательный и доступный жанр, который нравится игрокам всех возрастов. С постоянным развитием и инновациями Match-Match игры, вероятно, останутся популярными еще долгие годы.

# 1.3 Анализ предметной области

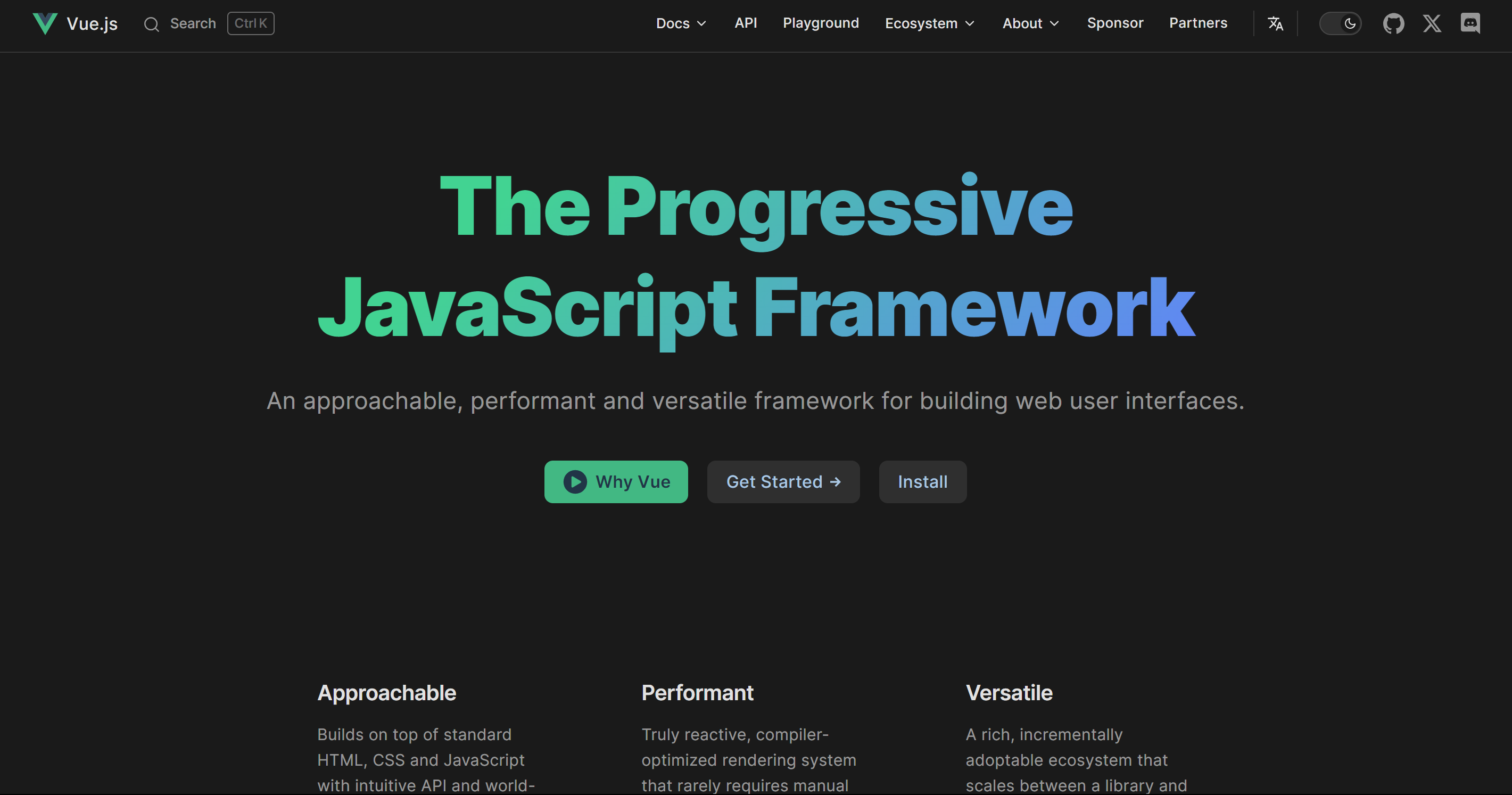
В зависимости от поставленных целей и задач, пользователь выбирает наиболее подходящий для него вариант. Игра "Match-Match" - это интерактивное веб-приложение, которое создано для развлечения пользователей и тренировки их памяти.

На данный момент игры с использованием библиотек становятся все более популярными и востребованными среди пользователей в онлайн-сфере.

Существует несколько способов разработки веб-приложений с использованием. Первый способ - написать приложение самостоятельно, используя знания библиотек. Это позволяет создать приложение на любом языке программирования, который вы знаете. Второй способ - использовать готовые компоненты и библиотеки для упрощения процесса разработки. Это может быть полезно, если у вас ограничены ресурсы или опыт в программировании.

Для создания игры "Match-Match" потребуется выбрать подходящие библиотеки и компоненты. Существует множество библиотек, которые могут быть использованы для управления состоянием приложения, визуализации данных и взаимодействия с пользователем. Рассмотрим наиболее популярные инструменты и технологии для создания игрового приложения:

Vue (или Vue.js) - это прогрессивный JavaScript фреймворк, позволяющий создавать динамические веб-приложения с помощью компонентной архитектуры и реактивной системы обновления данных при изменении состояния, обладая простотой в изучении и использовании, а также широкими возможностями и плагинами для расширения функциональности. Официальный сайт данной библиотеки представлен на рисунки 1.3.



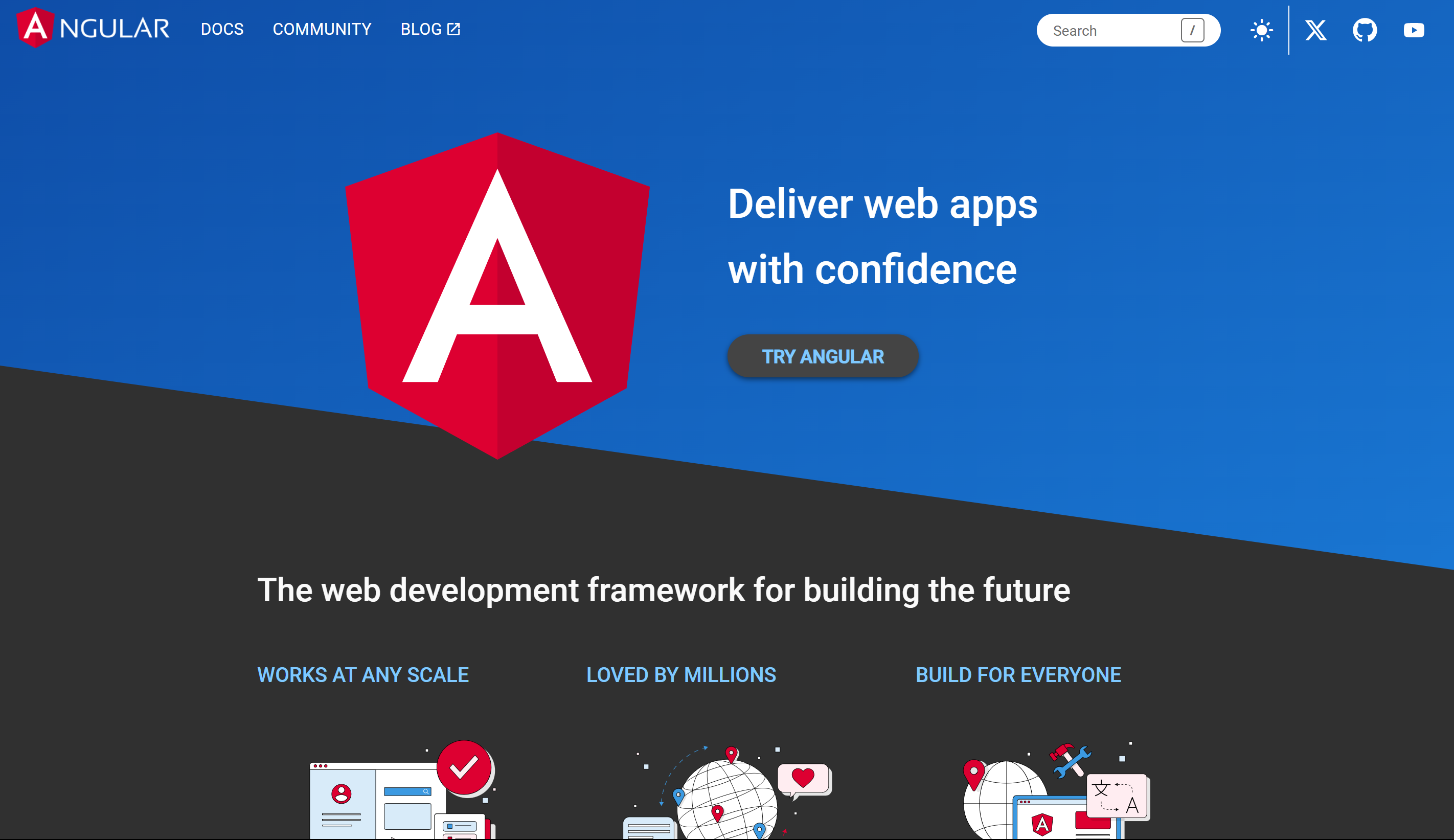
[Источник: Vue.js (Электронный ресурс). –

URL: [https://vuejs.org/](https://https://vuejs.org/)

(дата обращения 02.02.2024). – Текст: электронный]

Рисунок 1.3. Страница сайта Vue.js

Angular - это фреймворк разработки одностраничных приложений на JavaScript, предоставляющий комплексный набор инструментов для создания масштабируемых веб-приложений. Angular обладает реактивным подходом к программированию, а также мощными возможностями для управления данными и состоянием приложения. Он предлагает широкий спектр функциональности, включая модульность, инъекцию зависимостей, директивы, пайпы и многое другое, что делает его привлекательным выбором для разработки сложных веб-приложений. Представлен на рисунке 1.4.



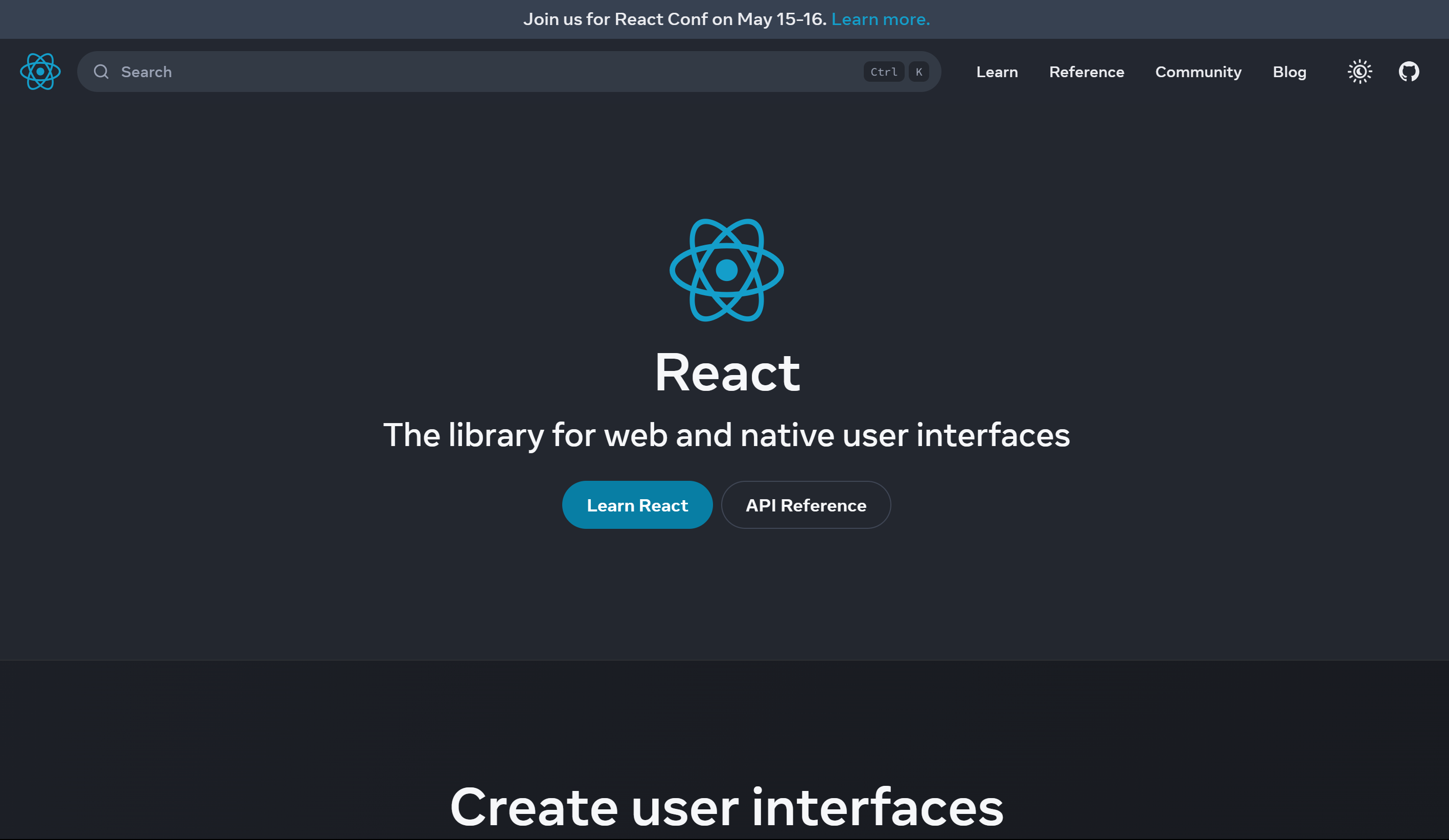
[Источник: Angular (Электронный ресурс). –

URL: https://angular.io/

(дата обращения 02.02.2024). – Текст: электронный]

Рисунок 1.4. Страница сайта Angular

React - это библиотека JavaScript для создания пользовательских интерфейсов, которая позволяет разрабатывать эффективные одностраничные приложения с использованием компонентной модели. Основным принципом React является создание множества независимых компонентов, каждый из которых отвечает за свое состояние и визуальное представление. React обладает виртуальным DOM (Document Object Model), который улучшает производительность при обновлении интерфейса, обеспечивая быструю отрисовку без лишних затрат ресурсов. Одной из ключевых особенностей React является JSX - расширение JavaScript, которое позволяет описывать структуру компонентов с помощью HTML-подобного синтаксиса, делая код более понятным и легким для чтения. Представлен на рисунке 1.5.



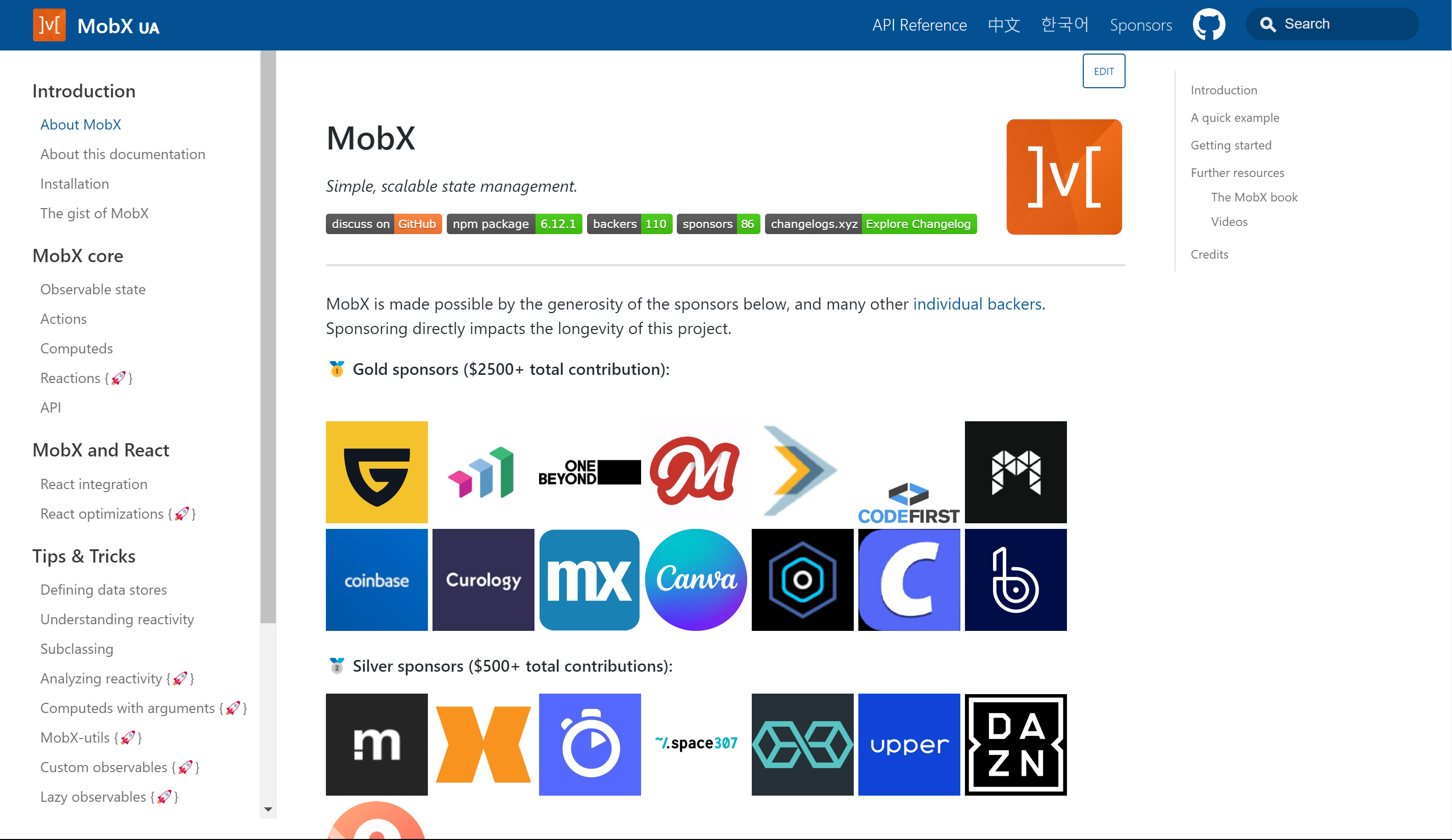
[Источник: React (Электронный ресурс). –

URL: https://react.dev/

(дата обращения 02.02.2024). – Текст: электронный]

Рисунок 1.5. Страница сайта React

MobX - это библиотека управления состоянием, предназначенная для создания реактивных приложений на основе наблюдаемых объектов. Основная идея MobX заключается в том, что любые изменения данных автоматически отслеживаются, что приводит к автоматическому обновлению всех зависимых частей приложения. MobX обеспечивает простоту использования за счет минимального количества необходимого шаблонного кода. Она также позволяет организовать легко читаемый и поддерживаемый код за счет использования принципа "наблюдаемых" данных и реактивных вычислений. MobX широко используется в React-приложениях, где позволяет эффективно управлять состоянием компонентов и обеспечивать быструю перерисовку интерфейса при изменении данных. В целом, MobX представляет собой удобное и эффективное средство управления состоянием для разработки современных веб-приложений. Представлен на рисунке 1.6.



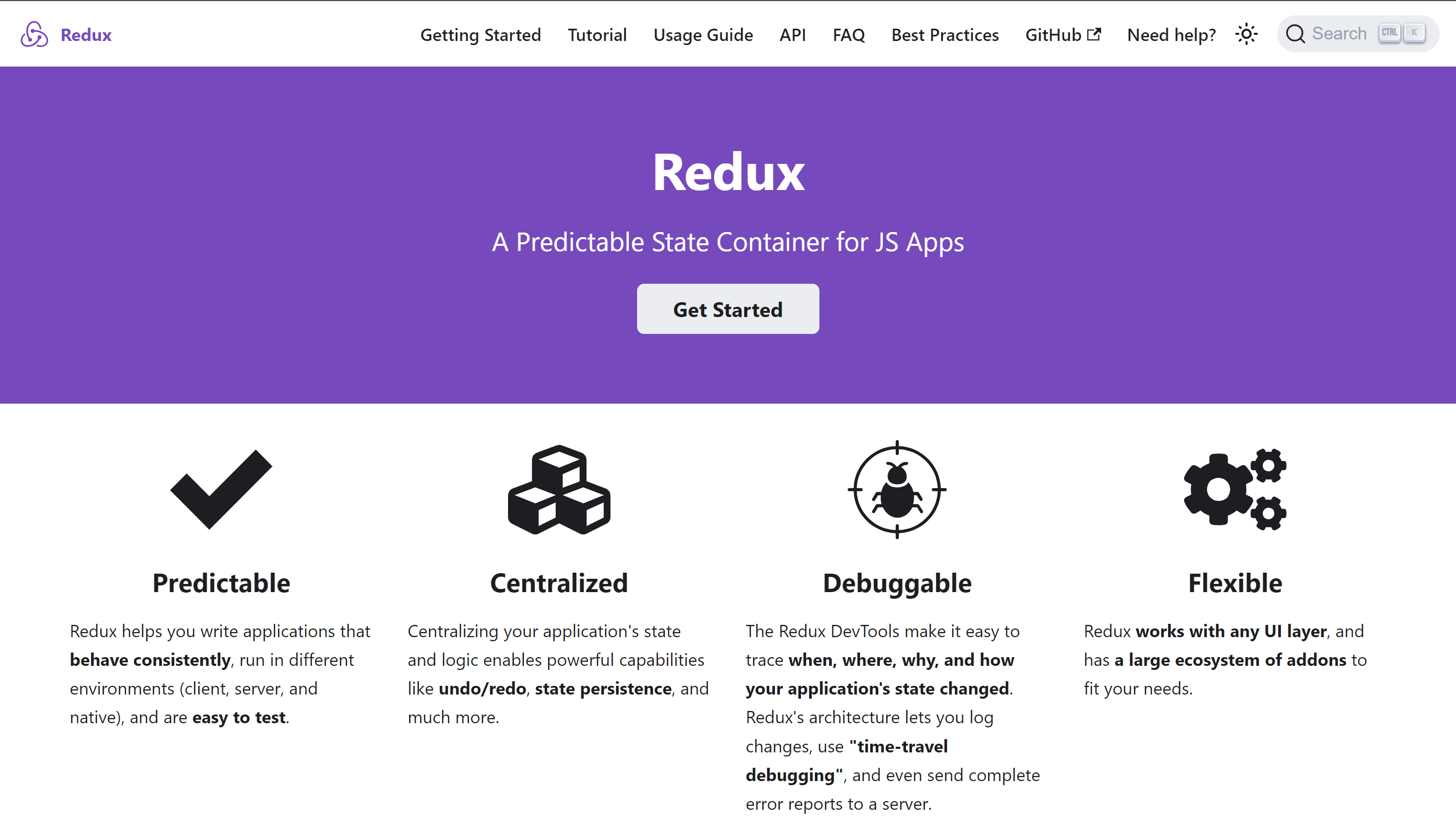
[Источник: MobX (Электронный ресурс). –

URL: https://mobx.js.org/README.html

(дата обращения 02.02.2024). – Текст: электронный]

Рисунок 1.6. Страница сайта MobX

Redux - это библиотека управления состоянием, предназначенная для разработки приложений на основе однонаправленного потока данных. Основная концепция Redux состоит в том, чтобы хранить все состояние приложения в единственном объекте-хранилище (store), который обновляется только с помощью чистых функций-редюсеров. Это обеспечивает предсказуемость и отслеживаемость изменений состояния, делая код проще в поддержке и отладке. Redux также включает в себя возможности для создания действий (actions) и использования middleware для обработки асинхронных операций. В связи с этим, Redux позволяет эффективно управлять сложным состоянием приложений и облегчает передачу данных между компонентами. Хотя использование Redux может потребовать больше шаблонного кода по сравнению с другими инструментами управления состоянием, его однонаправленный поток данных и строгость подхода делают его инструментом выбора для крупных и масштабируемых приложений, где предсказуемость и централизованное управление состоянием играют важную роль. Представлен на рисунке 1.7.



[Источник: Redux (Электронный ресурс). –

URL: https://redux.js.org/

(дата обращения 02.02.2024). – Текст: электронный]

Рисунок 1.7. Страница сайта Redux

Из вышеперечисленных библиотек, для разработки бота былb выбраны React и Redux. Они подходит из-за возможности создания сложных компонент и эффективного управления состоянием.

В программировании за сохранение кода в контрольных точках отвечает система контроля версий — специальная технология, которую можно подключить к любому проекту. Система контроля версий страхует от ошибок и возвращает код в то состояние, когда всё работало.

Git – это система для управления версиями исходного кода программ. Официальный логотип Рисунок 1.8



[Источник: Git (Электронный ресурс). –

URL: [Git (git-scm.com)](https://git-scm.com/)

(дата обращения 03.02.2024). – Текст: электронный]

Рисунок 1.8 – Страница сайта Git

Основное преимущество Git — скорость работы, простота и работа с большими проектами. В отличие от других систем контроля версий, Git не записывает изменения к каждому файлу, а как бы фотографирует весь проект целиком.

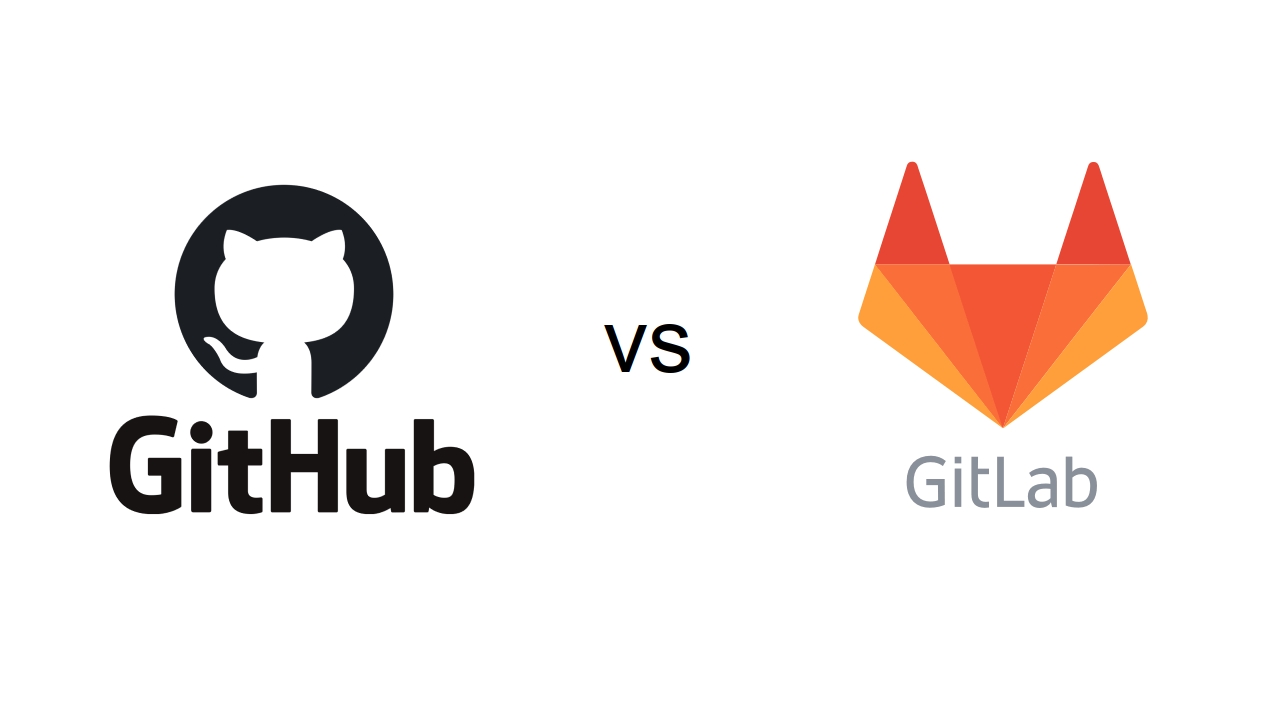
Это лишь некоторые из ключевых возможностей Git, которые делают его одним из наиболее популярных инструментов для управления версиями исходного кода. Git обеспечивает высокую гибкость, быстродействие и надежность в разработке программного обеспечения.

Не путать с GitHub — это онлайн-сервис, который основан на технологии Git. Он хранит репозитории в интернете, автоматически синхронизирует их с репозиториями у разработчиков, следит за обновлениями кода, позволяет редактировать код прямо в репозитории и копировать себе чужие репозитории.

Основные возможности Git:

1. Система коммитов: позволяет сохранять код в контрольных точках и возвращаться к предыдущему состоянию проекта.
2. Комплекс связанных веток: позволяет создавать ответвления от основной версии проекта и экспериментировать с кодом, не мешая другим участникам команды.
3. Инструмент совместного создания кода: разрешает всем участникам команды свободно перемещаться между ветками других разработчиков для копирования нужных фрагментов кода.
4. Распределённая система версий: обеспечивает удалённую работу над проектами и позволяет восстанавливать их из копий, хранящихся у каждого участника команды.

При разработке программного кода выбор удаленного репозитория для системы Git выбран GitHub (Рисунок 1.9)



[Источник: GitHub (Электронный ресурс). –

URL: [GitHub: Let’s build from here · GitHub](https://github.com/?ysclid=ltmyq8csjo535630964)

(дата обращения 03.02.2024). – Текст: электронный]

Рисунок 1.9 – Страница сайта GitHub