**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

**Журнал по практике**

Студент: Благушин Никита Александрович

Институт №3 “Системы управления, информатика и электроэнергетика”

Кафедра №304 “Вычислительные машины, системы и сети”

Учебная группа М30-107СВ-24

Направление подготовки (специальность) 09.04.01 “Информатика и вычислительная техника”

*(шифр) (название направления, специальности)*

Вид практики Ознакомительная

*(учебная, производственная, преддипломная или другой вид практики)*

Руководитель практики от МАИ

Ткачев Олег Алексеевич /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

*(фамилия, имя, отчество) (подпись) (дата)*

Студент

Благушин Никита Александрович /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

*(фамилия, имя, отчество) (подпись) (дата)*

Москва 2024 г.

**1.Место и сроки проведения практики**

*Наименование предприятия:* Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский авиационный институт (национальный исследовательский университет)»

*Название структурного подразделения:* Кафедра 304

*Сроки проведения практики:*

*-дата начала практики: 01.09.24*

*-дата окончания практики: 24.12.24*

1. **Инструктаж по технике безопасности**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

*(подпись проводившего)* *(дата проведения)*

1. **Индивидуальное задание студенту**

Поиск, анализ и подготовка материалов для написания диссертации.

Формируемые в ходе выполнения задания компетенции:

|  |  |  |
| --- | --- | --- |
| № пп | Шифр | Компетенция |
| 1 | ОК- 1 | Демонстрация возможности обобщения, анализа, критического осмысления информации, систематизации, прогнозирования, постановка исследовательских задач. |
| 2 | ОК-11 | Умение оформлять отчеты о проведенной научно-исследовательской работе и подготавливать публикации по результатам исследования. |
| 3 | ОК-8 | Способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности. |
| 4 | ОПК-6 | Владение методами и средствами получения, хранения, переработки и трансляция информации посредством современных компьютерных технологий. |
| 5 | ОПК-7 | Способность анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров. |

Достигаемые в ходе выполнения задания результаты освоения:

|  |  |  |
| --- | --- | --- |
| № пп | Шифр | Результат освоения |
|  | В-1 (ОК-1) | Владеть методами обобщения, анализа, критического осмысления информации |
|  | В-1 (ОК-11) | Владеть способами оформления отчетов о проведенной научно-исследовательской работе и подготовки публикации по результатам исследования |
|  | В-1 (ОК-8) | Владеть способностью к самостоятельному обучению новым методам исследования при изменении научного и научно-производственного профиля своей профессиональной деятельности |
|  | В-1 (ОК-9) | Владеть методами проведения научных исследований |
|  | В-1 (ОПК-6) | Владеть современными методами и средствами получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий |
|  | В-1 (ОПК-7) | Владеть методами анализа профессиональной информации, выделения в ней главного, структурирования, оформления и представления в виде аналитических обзоров |
|  | В-2 (ОК-1) | Владеть методами анализа диссертации, получение необходимых навыков для начала работы с диссертациями и научными статьями |
|  | З-1 (ОК-1) | Знать методы обобщения, анализа, критического осмысления информации |
|  | З-1 (ОК-11) | Знать требования оформления отчетов о проведенной научно-исследовательской работе и подготовки публикации по результатам исследования |
|  | З-1 (ОК-8) | Знать особенности самостоятельного обучения новым методам исследования при изменении научного и научно-производственного профиля своей профессионально |
|  | З-1 (ОК-9) | Знать методы проведения научных исследований |
|  | З-1 (ОПК-6) | Знать современные методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий |
|  | З-1 (ОПК-7) | Знать методы анализа профессиональной информации, выделения в ней главного, структурирования, оформления и представления в виде аналитических обзоров |
|  | З-2 (ОК-1) | Знать методы систематизации, прогнозирования, постановки исследовательских задач и выбора путей их решения на основании принципов научного познания |
|  | У-1 (ОК-1) | Уметь обобщать, анализировать, критически осмыслять информацию |
|  | У-1 (ОК-11) | Уметь оформлять отчеты о проведенной научно-исследовательской работе и подготавливать публикации по результатам исследования |
|  | У-1 (ОК-8) | Уметь самостоятельно обучаться новым методам исследования при изменении научного и научно-производственного профиля своей профессиональной деятельности |
|  | У-1 (ОК-9) | Уметь проводить научные исследования |
|  | У-1 (ОПК-6) | Уметь применять современные методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий |
|  | У-1 (ОПК-7) | Уметь анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров |

**4.План выполнения индивидуального задания**

**План работ практиканта по месту прохождения практики**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Дата | Содержание или наименование проделанной работы | Подразделение | Продолжительность, часы | Компетенция | Подпись консультантапрактики от предприятия |
| 18.11 | **Постановка задачи и описание предметной области** | Кафедра 304 | 20 | ОПК-7 |  |
| 28.11 | Обоснование актуальности и необходимости решения задачи | Кафедра 304 | 20 | ОПК-7 |  |
| 11.12 | Описание предметной области | Кафедра 304 | 5 |  |  |

Благушин Н.А. /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

(*подпись студента-практиканта) (*дата)

Ткачев О.А. /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

(*подпись руководителя практики от МАИ) (*дата)

**План самостоятельной работы студента**

|  |  |  |  |
| --- | --- | --- | --- |
| № пп | Содержание или наименование проделанной работы | Продолжительность, часы | Компетенция |
| 1 | Постановка задачи и описание предметной области |  | ОК- 1 |
| 2 | Изучение текущего положения методов решения данной задачи |  | ОК-2 |
| 3 | Анализ научных статей на смежные темы |  | ОК-8 |
| 4 | Поиск и подготовка материалов для последующего написания диссертации |  | ОК-8 |
| 5 | Изучение методов написания научных статей и диссертаций |  | ОПК-6 |
| Итого | |  |  |

Благушин Н.А. /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

(*подпись студента-практиканта) (*дата составления)

Ткачев О.А. /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

(*подпись руководителя практики от МАИ) (*дата составления)

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

**Отчет о прохождении практики**

Студент Благушин Никита Александрович

Институт №3 “Системы управления, информатика и электроэнергетика”

Кафедра№304 “Вычислительные машины, системы и сети”

Учебная группа №М3О-107СВ-24

Направление подготовки (специальность) 09.04.01 “Информатика и вычислительная техника”

*(шифр) (название направления, специальности)*

Вид практики: Ознакомительная

*(учебная, производственная, преддипломная или другой вид практики)*

Руководитель практики от МАИ

Ткачев Олег Алексеевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(фамилия, имя, отчество) (подпись)*

*Наименование предприятия:* Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский авиационный институт (национальный исследовательский университет)»

*Название структурного подразделения (отдел, лаборатория):* “Кафедра 304”

Студент

Благушин Н.А./\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

*(фамилия, имя, отчество) (подпись) (дата)*

Москва 2024 г

Содержание

[ВВЕДЕНИЕ 8](#_Toc185893148)

[1. Анализ и сравнение языков программирования для веб-приложений 9](#_Toc185893149)

[2. Анализ и сравнение фреймворков JavaScript 13](#_Toc185893150)

[3. Анализ и сравнение инструментов разработки 17](#_Toc185893151)

[4. Выбор технологий и инструментария для Backend разработки 21](#_Toc185893152)

[ЗАКЛЮЧЕНИЕ 23](#_Toc185893153)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 24](#_Toc185893154)

# ВВЕДЕНИЕ

Когда речь заходит о разработке ПО или цифрового продукта - стратегически важным является выбор оптимальных технологий, инструментов и методологий разработки. Под технологиями в профессиональной среде подразумеваются – языки программирования, библиотеки и фреймворки; под инструментами – IDE (интегрированная среда разработки от англ. Integrated Development Environment) и ПО для проектирования архитектурных связей, макета интерфейса, создания дизайна; под методологиями – принципы, парадигмы, стили написания кода, методы организации и поддержания архитектуры проекта и кодовой базы.

Цель данной ознакомительной практики – провести обзор и анализ технологий и инструментария, с последующим выбором конкретных средств для разработки цифровой платформы.

Анализ и сравнение средств будут происходить с учётом выбранной темы будущей выпускной квалификационной работы, тема которой: “Разработка цифровой платформы для поиска команды в проект-стартап”.

## Анализ и сравнение языков программирования для веб-приложений

При выборе языка программирования для создания цифровой платформы важно учитывать ряд факторов, таких как производительность, масштабируемость, поддержка сообщества, наличие библиотек и фреймворков, а также требования к проекту. Ниже представлено сравнение нескольких популярных языков программирования, которые часто используются в разработке цифровых платформ.

1. JavaScript

Преимущества:

• Фронтенд и бэкенд: используется как для клиентской, так и для серверной разработки (с помощью Node.js).

• Асинхронное программирование: поддержка асинхронного программирования, что улучшает производительность веб-приложений.

• Богатая экосистема: широкий выбор библиотек и фреймворков, таких как React, Angular, Vue.js.

• Браузерная поддержка: поддерживается всеми современными браузерами.

Недостатки:

• Безопасность: код JavaScript виден пользователю, что может привести к уязвимостям.

• Отладка: меньшая эффективность отладки по сравнению с другими языками.

• Ограниченная поддержка наследования: поддержка только одиночного наследования.

2. Python

Преимущества:

• Простота и читаемость: легкий для изучения и чтения синтаксис.

• Богатая библиотека: широкий выбор библиотек для веб-разработки (Django, Flask), анализа данных, машинного обучения.

• Сообщество: большое и активное сообщество, множество ресурсов и документации.

• Кроссплатформенность: работает на всех основных операционных системах.

Недостатки:

• Производительность: медленнее по сравнению с компилируемыми языками.

• Память: высокое потребление памяти.

• Мобильные приложения: ограниченная поддержка для клиентских и мобильных приложений.

3. Java

Преимущества:

• Платформонезависимость: "Write once, run anywhere" благодаря JVM.

• Производительность: высокая производительность и хорошее управление памятью.

• Безопасность: встроенные механизмы безопасности.

• Сообщество: широкое сообщество и множество библиотек.

Недостатки:

• Сложность: более сложный синтаксис и кривая обучения по сравнению с другими языками.

• Ресурсоемкость: высокое потребление памяти и ресурсов.

• Медленное развитие: медленное внедрение новых функций.

4. C#

Преимущества:

• Интеграция с .NET: отличная интеграция с .NET Framework, что упрощает разработку приложений для Windows.

• Производительность: высокая производительность и хорошее управление памятью.

• Инструменты: мощные инструменты разработки, такие как Visual Studio.

• Кроссплатформенность: поддержка кроссплатформенной разработки с помощью .NET Core.

Недостатки:

• Платформозависимость: исторически зависим от Windows, хотя ситуация улучшается с .NET Core.

• Кривая обучения: более сложный синтаксис и кривая обучения по сравнению с другими языками.

• Цена: некоторые инструменты и библиотеки могут быть платными.

5. PHP

Преимущества:

• Простота: легкий для изучения и использования.

• Широкое использование: широко используется для веб-разработки, особенно для серверной части.

• Сообщество: большое сообщество и множество ресурсов.

• Интеграция: хорошая интеграция с различными базами данных и веб-серверами.

Недостатки:

• Производительность: медленнее по сравнению с компилируемыми языками.

• Безопасность: часто подвергается критике за уязвимости безопасности.

• Современность: некоторые считают его устаревшим по сравнению с более новыми языками.

Ниже приведена сравнительная таблица (таблица 1) языков программирования по ключевым параметрам для веб-разработки.

Таблица 1 - Сравнение языков программирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр | JavaScript | Python | Java | C# | PHP |
| Фронтенд | Да | Нет | Нет | Нет | Нет |
| Бэкенд | Да (Node.js) | Да | Да | Да | Да |
| Простота изучения | Средняя | Высокая | Низкая | Средняя | Высокая |
| Производительность | Средняя | Низкая | Высокая | Высокая | Средняя |
| Безопасность | Низкая | Средняя | Высокая | Высокая | Низкая |
| Кроссплатформенность | Высокая | Высокая | Высокая | Высокая | Высокая |
| Сообщество | Большое | Большое | Большое | Среднее | Большое |
| Продолжение таблицы 1 | | | | | |
| Библиотеки и фреймворки | Широкий выбор | Широкий выбор | Широкий выбор | Широкий выбор | Широкий выбор |
| Инструменты разработки | Средние | Средние | Высокие | Высокие | Средние |

По итогам проведенного анализа различных языков программирования для веб-приложений было принято решение выбрать в качестве основного языка – JavaScript. Причинами выбора можно отметить четыре фактора:

• Фронтенд и бэкенд: возможность использования как для клиентской, так и для серверной разработки.

• Асинхронное программирование: поддержка асинхронного программирования, что улучшает производительность веб-приложений.

• Богатая экосистема: широкий выбор библиотек и фреймворков, таких как React, Angular, Vue.js.

• Браузерная поддержка: поддерживается всеми современными браузерами.

JavaScript является универсальным языком, который позволяет создавать как клиентские, так и серверные приложения, что делает его идеальным выбором для веб-разработки.

# Анализ и сравнение фреймворков JavaScript

Для анализа и сравнения фреймворков JavaScript, включая React, Vue, Angular и Next.js, мы можем применить схему сравнения, аналогичную той, что была использована для языков программирования. Основные параметры для сравнения включают простоту изучения, производительность, безопасность, кроссплатформенность, поддержку сообщества, наличие библиотек и фреймворков, а также инструменты разработки.

* 1. React

Преимущества:

* Простота изучения: React известен своей простотой изучения благодаря компонентному подходу.
* Производительность: React использует виртуальный DOM для оптимизации производительности.
* Сообщество: React имеет одно из самых больших сообществ разработчиков среди фреймворков JavaScript.

Недостатки:

* Изучение: хотя React относительно прост в изучении, глубокое понимание его концепций может потребовать времени.
* Производительность: в некоторых случаях, особенно при неправильном использовании, React может привести к ухудшению производительности.
  1. Vue

Преимущества:

* Простота изучения: Vue предлагает простой и понятный синтаксис, что делает его доступным даже для новичков.
* Производительность: Vue использует компиляцию для оптимизации производительности и предлагает плагины для улучшения работы с DOM.
* Сообщество: Vue имеет активное сообщество разработчиков и множество ресурсов для изучения.

Недостатки:

* Размер сообщества: по сравнению с React и Angular, сообщество Vue меньше, хотя оно активно растет.
  1. Angular

Преимущества:

* Структура и организация: Angular предлагает строгую структуру и организацию кода, что упрощает работу над крупными проектами.
* Производительность: Angular использует систему модулей и компонентов для оптимизации производительности.
* Сообщество: Angular имеет большое и активное сообщество разработчиков.

Недостатки:

* Сложность изучения: Angular имеет более сложный синтаксис и большое количество концепций, что может затруднить начало работы.
  1. Next.js

Преимущества:

* Серверный рендеринг: Next.js позволяет улучшить SEO и время загрузки страниц за счет серверного рендеринга.
* API маршруты: встроенная поддержка создания API маршрутов для упрощения разработки бэкенда.
* Оптимизация изображений: инструменты для оптимизации изображений, улучшающие производительность.
* Поддержка TypeScript: официальная поддержка TypeScript для повышения надежности кода.
* Большое сообщество: активное сообщество разработчиков с множеством ресурсов и примеров.

Недостатки:

* Изучение: для полноценной эксплуатации Next.js может потребоваться время на изучение, особенно если вы новичок в этой области.
* Зависимость от Vercel: тесная интеграция с Vercel может ограничить гибкость в выборе хостинга.

Для проведения сравнительного анализа фреймворков JavaScript (React, Vue, Angular, Next.js), составлена Таблица 2, которая включает в себя ключевые параметры, такие как: простота изучения, производительность, безопасность, кроссплатформенность, поддержку сообщества, наличие библиотек и фреймворков, а также инструменты разработки.

Таблица 2 – Сравнение фреймворков JS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | React | Vue | Angular | Next.js |
| Простота изучения | Средняя | Высокая | Средняя | Средняя |
| Производительность | Высокая | Высокая | Высокая | Высокая |
| Безопасность | Высокая | Высокая | Высокая | Высокая |
| Кроссплатформенность | Высокая | Высокая | Высокая | Высокая |
| Сообщество | Большое | Среднее | Большое | Большое |
| Библиотеки и фреймворки | Широкий выбор | Множество плагинов | Множество библиотек | Официальные и сторонние |
| Инструменты разработки | Средние | Средние | Высокие | Средние |
| Простота изучения | Средняя | Высокая | Средняя | Средняя |

На основе вышеуказанного анализа, Next.js представляет собой выгодный выбор для разработки цифровой платформы. Он предлагает множество преимуществ, включая серверный рендеринг, статическую генерацию сайтов, API маршруты, оптимизацию изображений, поддержку TypeScript и большое сообщество. Эти характеристики делают Next.js идеальным выбором для проектов, требующих высокой производительности, SEO и удобства разработки.

## Анализ и сравнение инструментов разработки

При разработке цифровой платформы важно выбрать подходящие инструменты, которые будут оптимизировать процесс разработки, обеспечивать качество кода и упрощать совместную работу в команде. Ниже представлено сравнение различных IDE, инструментов для создания дизайна, проектирования архитектуры приложения, структур баз данных и связей, а также инструментов для контроля версий и хостинга.

Подробный анализ и сравнение IDE для веб-разработки

1. WebStorm

Преимущества:

• Интеграция с VCS: поддержка Git, SVN, Mercurial и других систем контроля версий.

• Поддержка фреймворков: отличная поддержка популярных фреймворков, таких как React, Angular, Vue.js.

• Умные подсказки: интеллектуальные подсказки и автозаполнение кода.

• Рефакторинг: мощные инструменты для рефакторинга кода.

• Дебаггер: встроенный дебаггер для JavaScript и Node.js.

Недостатки:

• Цена: платная лицензия, что может быть дорого для некоторых пользователей. Однако студенты могут получить бесплатную образовательную лицензию

• Ресурсоемкость: требует значительных системных ресурсов.

2. Visual Studio Code (VSCode)

Преимущества:

• Бесплатность: бесплатная и с открытым исходным кодом.

• Расширяемость: огромное количество расширений для поддержки различных языков и фреймворков.

• Интеграция с VCS: поддержка Git внутри IDE.

• Легкость: легкий и быстрый.

• Интеграция с терминалом: встроенный терминал.

Недостатки:

• Настройка: требует времени на настройку и установку необходимых расширений.

• Производительность: может замедляться при большом количестве установленных расширений.

3. Vim

Преимущества:

• Легкость: очень легкий и быстрый.

• Возможности редактирования: мощные возможности редактирования текста с использованием команд.

• Расширяемость: поддержка плагинов для расширения функциональности.

• Кроссплатформенность: работает на всех основных операционных системах.

Недостатки:

• Крутая кривая обучения: требует значительного времени на изучение.

• Отсутствие GUI: отсутствие графического интерфейса может быть неудобным для некоторых пользователей.

4. Sublime Text

Преимущества:

• Легкость: быстрый и легкий.

• Расширяемость: поддержка плагинов и пакетов для расширения функциональности.

• Многоплатформенность: работает на Windows, macOS и Linux.

• Интерфейс: простой и интуитивно понятный интерфейс.

Недостатки:

• Цена: платная лицензия, хотя есть бесплатная версия с периодическими напоминаниями о покупке.

• Ограниченная поддержка VCS: требует установки дополнительных плагинов для полноценной работы с системами контроля версий.

5. Atom

Преимущества:

• Бесплатность: бесплатный и с открытым исходным кодом.

• Расширяемость: большое количество доступных пакетов и тем.

• Интеграция с Git: встроенная поддержка Git и GitHub.

• Интерфейс: простой и настраиваемый интерфейс.

Недостатки:

• Производительность: может быть медленным при работе с большими проектами.

• Ресурсоемкость: требует значительных системных ресурсов.

Ниже приведена таблица (таблица 3) сравнения выбранных IDE:

Таблица 3 - Сравнение IDE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр | WebStorm | VSCode | Vim | Sublime Text | Atom |
| Цена | Платная | Бесплатная | Бесплатная | Платная | Бесплатная |
| Ресурсоемкость | Высокая | Средняя | Низкая | Низкая | Высокая |
| Поддержка фреймворков | Отличная | Отличная | Ограниченная | Хорошая | Хорошая |
| Интеграция с VCS | Отличная | Отличная | Ограниченная | Ограниченная | Отличная |
| Параметр | WebStorm | VSCode | Vim | Sublime Text | Atom |
| Расширяемость | Высокая | Высокая | Высокая | Высокая | Высокая |
| Интерфейс | Интуитивный | Интуитивный | CLI | Интуитивный | Интуитивный |
| Кривая обучения | Низкая | Низкая | Высокая | Низкая | Низкая |
| Производительность | Высокая | Средняя | Высокая | Высокая | Средняя |
| Дебаггер | Встроенный | Расширяемый | Ограниченный | Ограниченный | Ограниченный |

После анализа различных IDE для веб-разработки было принято решение выбрать WebStorm. Основные причины выбора:

• Интеграция с VCS: поддержка различных систем контроля версий.

• Поддержка фреймворков: отличная поддержка популярных фреймворков, таких как React, Angular, Vue.js.

• Умные подсказки и рефакторинг: интеллектуальные подсказки, автозаполнение кода и мощные инструменты для рефакторинга.

• Встроенный дебаггер: поддержка дебаггера для JavaScript и Node.js.

Хотя WebStorm является платным продуктом, студенты могут получить бесплатную образовательную лицензию, что решает проблему с ценой. Это делает WebStorm доступным и мощным инструментом для студентов, занимающихся веб-разработкой.

## Выбор технологий и инструментария для Backend разработки

В процессе разработки цифровой платформы особое внимание уделяется выбору технологий и инструментария для реализации backend части системы. Это важный этап, который напрямую влияет на архитектуру, производительность, безопасность и масштабируемость будущего продукта. В данном разделе рассматривается выбор следующих технологий и инструментов: Strapi и Insomnia.

* 1. Strapi

Strapi — это открытый источник, headless CMS, построенный на Node.js. Он предоставляет разработчикам возможность быстро создавать и управлять контентом без необходимости написания кода. Одним из ключевых преимуществ Strapi является его гибкость и способность адаптироваться под различные требования проекта. Также Strapi предлагает удобный интерфейс для управления контентом, что значительно упрощает процесс разработки и обновления контента. Выбор Strapi обусловлен его возможностью быстро запускать проекты, обеспечивать удобство управления контентом и поддерживать высокую скорость разработки.

* 1. Insomnia

Insomnia — это мощный инструмент для тестирования API, который позволяет разработчикам отправлять HTTP-запросы к своим сервисам и просматривать ответы. Insomnia предлагает удобный графический интерфейс пользователя, поддержку различных форматов данных, автоматическое создание документации OpenAPI и многое другое. Выбор Insomnia обусловлен его способностью упростить процесс тестирования API, обеспечивая быстрый и эффективный способ взаимодействия с backend сервисами во время разработки.

Выбор Strapi, PostgreSQL и Insomnia для backend части платформы обусловлен их совместимостью, надежностью, производительностью и удобством использования. Эти технологии и инструменты обеспечивают разработчикам необходимые инструменты

# ЗАКЛЮЧЕНИЕ

Данная работа была посвящена анализу и сравнению различных инструментов разработки для веб-приложений. В результате исследования и оценки по ключевым критериям были выбраны следующие инструменты:

Язык программирования: JavaScript, благодаря его универсальности и поддержке как фронтенд, так и бэкенд разработки.

Инструмент разработки (IDE): WebStorm, за его мощные функции и интеграцию с системами контроля версий, с решением проблемы стоимости через бесплатную студенческую лицензию.

CMS/ORM: Strapi, за его гибкость и удобство управления контентом без необходимости написания кода, что позволяет быстро запускать проекты и обеспечивает высокую скорость разработки.

Система управления базами данных: PostgreSQL, за свою надежность, производительность и масштабируемость, обеспечивающую высокую производительность, надежность и гибкость в управлении данными.

Инструмент для тестирования API: Insomnia, за его способность упростить процесс тестирования API, обеспечивая быстрый и эффективный способ взаимодействия с backend сервисами во время разработки.

Эти инструменты обеспечат эффективную и качественную разработку веб-приложений, удовлетворяя потребности проекта и способствуя достижению поставленных целей.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Документация Vercel : сайт. – URL: https://vercel.com/docs (дата обращения: 20.12.2024)
2. Python, JavaScript или C++? // Habr : сайт. – URL: https://habr.com/ru/companies/selectel/articles/801765/ (дата обращения: 20.12.2024)
3. Документация Next.js : сайт. – URL: https://nextjs.org/docs (дата обращения: 20.12.2024)
4. MDN - документация JavaScript и CSS : сайт. – URL: https://developer.mozilla.org/ru/ (дата обращения: 10.04.2024)
5. Документация Strapi : сайт. – URL: https://docs.strapi.io/ (дата обращения: 20.12.2024)