**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 7](#_Toc194309834)

[1. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ 8](#_Toc194309835)

[1.1 Обоснование выбора средств реализации компонента(ов) архитектурного решения 8](#_Toc194309836)

[1.1.1 Сравнительный обзор современных средств разработки клиентской части 9](#_Toc194309837)

[1.1.2. Сравнительный обзор современных СУБД 11](#_Toc194309838)

[1.2. Описание реализации компонента(ов) архитектурного решения 13](#_Toc194309839)

[1.3. Обеспечение информационной безопасности архитектурного решения 13](#_Toc194309840)

[1.4. Описание эксплуатации компонентов архитектурного решения 13](#_Toc194309841)

[ЗАКЛЮЧЕНИЕ 14](#_Toc194309842)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc194309843)

# ВВЕДЕНИЕ

# 1. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

## 1.1 Обоснование выбора средств реализации компонента(ов) архитектурного решения

При разработке веб-приложения важно учитывать надежность, производительность, удобство разработки и поддержки кода. В данном проекте были выбраны следующие ключевые инструменты:

1. Язык программирования — Python.
2. Фреймворк — Django.
3. Среда разработки — PyCharm.
4. Система управления базами данных (СУБД) — PostgreSQL.

Эти технологии обеспечивают гибкость, удобство разработки, поддержку масштабируемости и безопасности, что делает их оптимальным решением для создания веб-приложения. Рассмотрим каждую из этих технологий подробнее.

#### Таблица 1.1 — Сравнительная характеристика среды разработки

|  |  |  |
| --- | --- | --- |
| Среда разработки | Преимущества | Недостатки |
| PyCharm | Интеллектуальная поддержка Python, встроенные инструменты для Django и работы с базами данных, удобный отладчик | Высокое потребление ресурсов, платная профессиональная версия |
| Visual Studio Code | Легкость, множество расширений, бесплатность | Ограниченный встроенный функционал для Python |
| Eclipse (с PyDev) | Мощный инструмент, поддержка различных языков | Сложная настройка, перегруженный интерфейс |
| Atom | Простота, кроссплатформенность | Медленная работа с большими проектами |

Для реализации проекта выбрана среда разработки **PyCharm**, так как она предоставляет удобные инструменты для работы с кодом, отладкой и тестированием. PyCharm поддерживает интеграцию с Django, PostgreSQL и другими необходимыми инструментами, что делает разработку более эффективной. Кроме того, среда обладает интеллектуальной подсветкой синтаксиса, автодополнением кода и мощными инструментами анализа, что значительно упрощает процесс написания программного кода.

### 1.1.1 Сравнительный обзор современных средств разработки клиентской части

Для разработки клиентской части веб-приложений используется множество различных инструментов и технологий. В качестве средств разработки клиентской части могут быть рассмотрены:

1. Python с фреймворком Django:

Преимуществами комбинации являются:

* простота синтаксиса, что делает Python доступным для новичков;
* высокая производительность благодаря интеграции с PostgreSQL;
* сильное сообщество и поддержка документации.

Недостатки:

* потребность в серверной части, так как Python в основном используется для серверных решений.

Таким образом, Django предоставляет встроенные возможности для разработки интерфейсов, маршрутизации и работы с базами данных. Отличается удобством и возможностями расширения с помощью сторонних библиотек.

1. JavaScript с фреймворком React:

Преимуществами комбинации являются:

* очень популярная технология для создания динамических и интерактивных пользовательских интерфейсов;
* быстрая разработка и множество готовых библиотек для использования.

Недостатки:

* требует серверной части для обработки данных;
* высокий порог вхождения для новичков в фронтенд-разработке.

Таким образом, React активно используется для создания клиентских приложений, которые требуют высокой интерактивности и динамичности.

1. TypeScript с фреймворком Angular:

Преимуществами комбинации являются:

* Статическая типизация позволяет легко масштабировать проекты и повышает надежность кода;
* Большое сообщество и документация.

Недостатки:

* Высокая сложность в освоении по сравнению с JavaScript;
* Потребность в настройке компиляции и дополнительных инструментах.

Таким образом, Angular является фреймворком с большим количеством встроенных инструментов для создания сложных клиентских приложений.

Выбор **Python** в качестве основного языка программирования обусловлен его простым и читаемым синтаксисом, что ускоряет процесс разработки и снижает вероятность ошибок. Python обладает обширной экосистемой библиотек и фреймворков, что делает его универсальным инструментом для создания веб-приложений. Кроме того, активное сообщество разработчиков и широкая документация позволяют оперативно находить решения возникающих проблем.

Использование **Django** в качестве фреймворка для разработки веб-приложения оправдано его встроенными инструментами, обеспечивающими быструю и удобную разработку. Django предлагает мощную систему аутентификации, автоматическое управление базой данных, гибкую систему маршрутизации и высокий уровень безопасности. Это делает его отличным выбором для создания масштабируемых и надежных веб-приложений.

### 1**.1.2. Сравнительный обзор современных СУБД**

Современные системы управления базами данных (СУБД) предоставляют разнообразные возможности для хранения и обработки данных. Рассмотрим несколько популярных СУБД.

1. PostgreSQL:

Преимущества:

* высокая производительность и масштабируемость;
* поддержка сложных запросов и транзакций;
* полная совместимость с SQL;
* бесплатность и открытый исходный код.

Недостатки:

* может быть сложнее в настройке и обслуживании по сравнению с другими СУБД.

Таким образом, PostgreSQL идеально подходит для сложных систем, где необходимо хранение больших объемов данных, высокая производительность и отказоустойчивость.

1. MySQL

Преимущества:

* простота использования и большая популярность;
* подходит для приложений с менее сложной структурой данных;
* хорошая производительность при высоких нагрузках.

Недостатки:

* ограниченные возможности при работе с транзакциями и сложными запросами.

Таким образом, MySQL можно выбрать для менее сложных проектов, однако она уступает PostgreSQL по возможностям работы с транзакциями и сложными запросами.

1. SQLite:

Преимущества:

* легковесность и простота использования;
* подходит для небольших проектов и приложений с низким уровнем нагрузки;
* быстрое встраивание в приложение.

Недостатки:

* не предназначена для работы с большими объемами данных и высокими нагрузками.

Таким образом, SQLite идеален для небольших приложений и прототипов, но для больших и масштабируемых проектов лучше выбрать PostgreSQL.

1. Oracle:

Преимущества:

* высокая производительность и надежность;
* большие возможности для работы с бизнес-данными.

Недостатки:

* высокая стоимость лицензий;
* сложность в настройке и обслуживании.

Таким образом, Oracle является отличным решением для крупных корпоративных приложений, однако из-за высокой стоимости лицензий и сложности администрирования она не является лучшим выбором для малых и средних проектов.

В качестве системы управления базами данных выбрана **PostgreSQL**, так как она обеспечивает высокую производительность, надежность и поддержку сложных SQL-запросов. PostgreSQL отличается хорошей масштабируемостью, что позволяет эффективно работать с большими объемами данных. Открытый исходный код и активное сообщество обеспечивают постоянное развитие и улучшение данной СУБД, что делает её лучшим выбором для данного проекта.

## **1.2. Описание реализации компонента(ов) архитектурного решения**

## 1.3. Обеспечение информационной безопасности архитектурного решения

## 1.4. Описание эксплуатации компонентов архитектурного решения

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1.