Лабораторная работа №3

Задание №1

Для проектирования бэкенда необходимо воспользоваться программным обеспечением, с помощью которого можно создавать различные диаграммы и схемы. Такие инструменты позволяют визуализировать архитектуру системы, продемонстрировать взаимодействие между компонентами и обеспечить ясное представление о структуре и логике работы бэкенда.

Для анализа инструментальных средств для построения диаграмм и схем были выбраны такие популярные решения как Draw.io, Visio, Lucidchart. В таблице 1 представлено сравнение этого программного обеспечения.

Таблица 1 – Сравнение программного обеспечения для построения диаграмм и схем

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Draw.io | Visio | Lucidchart |
| Бесплатное использование | Полностью бесплатный | Требует покупки лицензии | Бесплатная версия с ограниченными возможностями |
| Офлайн режим | Поддерживается через десктопное приложение | Полностью поддерживается | Не поддерживается |
| Поддержка шаблонов и символов | Обширная библиотека шаблонов | Очень широкий выбор профессиональных шаблонов | Хорошая поддержка с современными шаблонами |
| Совместная работа | Поддержка реального времени через веб | Ограниченная, через SharePoint или OneDrive | Отличная поддержка совместной работы в реальном времени |

В результате сравнения программного обеспечения был выбран Draw.io, так как он удобен в использовании, полностью бесплатный и имеет офлайн режим.

Telegram предоставляет удобный API для взаимодействия с ботами, поэтому для реализации бэкенда для телеграмм-бота были выбраны следующие популярные языки программирования: Python, Java, C. В таблице 2 представлено сравнение языков программирования.

Таблица 2 – Сравнение языков программирования для реализации бэкенда

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Python | Java | C |
| Простота разработки | Минимальный шаблонный код, динамическая типизация | Много шаблонного кода, статическая типизация | Ручное управление памятью, сложный синтаксис |
| Библиотеки для Telegram | Python-telegram-bot, aiogram, pyTelegramBotAPI | TelegramBots, java-telegram-bot-api | Ограниченный выбор, низкоуровневые реализации |
| Производительность | Для бота оптимальная | Высокая за счёт JIT компиляции | Очень высокая за счёт нативной компиляции |
| Кроссплатформенность | Работает везде, где есть интерпретатор | Хорошая | Ограниченная, требуется перекомпиляция |
| Документация и сообщество | Огромное сообщество, много туториалов | Большое сообщество | Много материалов, но сложно для новичков |
| Интеграция с веб-приложением | Django, Flask, FastAPI | Spring Boot | Требуется дополнительный фреймворк |

В результате сравнения языков программирования был выбран Python, так как для задачи телеграмм-бота, где важнее скорость разработки и простота поддержки, чем максимальная производительность, Python превосходит Java и C. Производительность Python достаточна для обработки запросов бота, при этом сильно экономит на разработке.

Задание №2

Технологический стек состоит из:

– Программное обеспечение для проектирования

* Draw.io – сервис для создания схем, диаграмм.
* Figma – сервис для создания макетов интерфейса, логотипа, иконок.

– Программное обеспечение для разработки

Фронтэнд

* Aiogram
* python-telegram-bot
* pyTelegramBotAPI

Бэкенд

* python-telegram-bot
* requests

Базы данных

* SQLite

Аналитика

* Grafana
* Prometheus

Инфраструктура

* Docker

Тестирование

* Pytest
* Unittest
* Pytest-asyncio