Таблица сравнения фреймворков для бэкенда

|  |  |  |
| --- | --- | --- |
| Фреймворки | Плюсы | Минусы |
| Kivy | * Библиотека мультитач-виджетов * Гибкий графический движок для создания интерактивных интерфейсов * Множество предопределенных GUI-виджетов * Пользовательские виджеты (не использует нативные виджеты) * Кроссплатформенность * Подходит для Android и iOS * Поддержка работы с мультимедиа и сетью | * Недостаточно гибкий для сложных приложений * Может столкнуться с несовместимостью с некоторыми технологиями * Неродной пользовательский интерфейс |
| BeeWare | * Кросплатформенность * Подходит для Android и iOS * Подходит для более сложных проектов * Поддерживает собственный пользовательский интерфейс | * Использует нативные интерфейсы * Обновления могут привносить ошибки, исправление которых требует немало времени * Не реализованы востребованные функции |
| PyQt | * Интеграция с другими библиотеками и фреймворками * Поддержка интернационализации и локализации * Доступ к большому количеству функциональных возможностей из библиотеки Qt * Поддержка мультимедиа, интеграция баз данных | * Плохая и неполная документация * Сложный набор функций * Некоторые инструменты отладки недоступны из-за особенностей работы * Занимает большой объем памяти |
| Tkinter | * Поддержка распространенных виджетов, используемых в приложениях с графическим интерфейсом * Событийно-ориентированное программирование для обработки взаимодействия с пользователем * Встроенная поддержка изображений и цветов * Кроссплатформенность * Гибок и стабилен | * Не эффективен в работе над масштабными программами * Ошибки переносимости * Ограниченный функционал * Устаревший интерфейс * Не включает расширенные виджеты |
| PySide | * Поддержка отображения изображений, видео и других типов мультимедиа * Позволяет создавать пользовательский интерфейс * Позволяет использовать основные ресурсы пользовательского интерфейса на основе виджетов | * Не имеет специального инструмента развертывания * Отсутствие надлежащей документации * Не работают некоторые внешние библиотеки |
| PySimpleGUI | * Возможность использовать для создания сложных пользовательских интерфейсов * Прост в освоении * Мультиплатформенная поддержка * Можно использовать для создания сложных пользовательских интерфейсов | * Создание мобильного приложения осуществимо только с использованием PyDroid3, Termux * Никаких зависимостей от пакетов, кроме tkinter |
| Flutter | * Красивый, удобный и гибкий интерфейс * Нативность приложений для iOS и Android * Собственный графический движок * Интерфейс легко разбивается на отдельные модули | * Требует много памяти и мощности компьютера * Интерфейс создается с помощью кода, из-за чего грань между логикой и дизайном гораздо тоньше |

Таблица сравнения СУБД

|  |  |  |
| --- | --- | --- |
| CУБД | Плюсы | Минусы |
| MySQL | * Высокая гибкость имеющейся системы * Практичность * Поддержка графических интерфейсов для упрощения управления манипуляциями * Поддержка ODBC драйвера MyOBC * Осуществляет частое чтение данных * Высокая производительность | * Ограниченный функционал * Затраты на преобразование * Серьезные последствия при обнаружении критических ошибок |
| SQLite | * высокая скорость работы * хранение электронных материалов в пределах одного файла * повышенный уровень надежности * нулевые конфигурации * небольшой размер исходного файла * автономное функционирование * Кроссплатформенность | * ограниченность поддержки типов данных * отсутствие хранимых процедур * ограниченность в применение |
| PostgreSQL | * Осуществляет частое обновление данных * Имеет пользовательский тип данных * Поддержка множества типов данных * Работа с большими объемами * Поддержка сложных запросов * Одновременная модификация базы * Высокая мощность и широкая функциональность * Кроссплатформенность | * Повышенное потребление ресурсов * Отсутствие некоторых функций * Низкая скорость работы * Низкая производительность |
| SQAlchemy | * Безопасность. Параметры запросов экранируются * Высокая производительность * Переносимость * Большая гибкость | * Не выполняет большие операции с данными, связанные с объединением нескольких таблиц и разделением базы данных между процессами * Неэффективность |
| MongoDB | * Гибкая система хранения информации * Высокая производительность при выполнении простых запросов * Простая масштабируемость * Высокая скорость и более высокая доступность | * Отсутствие хранимых процедур и функций (нельзя полностью автоматизировать работу) * Не поддерживает объединения в качестве реляционной базы данных * Если индексация реализована неправильно или имеет какие-либо неточности, MongoDB будет работать с очень низкой скоростью * Ограниченный размер данных и вложенности * Дублирование данных затрудняет обработку наборов данных * Требует большого объема хранилища |
| DynamoDB | * Не теряет производительность при масштабировании * Простой API на основе REST * Обеспечивает обновления, резервное копирование и выполнение других административных задач * Бесшовная репликация данных * Не требует базовой инфраструктуры * Быстрое время работы | * Ограниченные возможности запроса * Невозможно использовать соединения таблиц * Трудно предсказать затраты * Ограниченные возможности хранения * Отсутствие модели развертывания |

### Мобильное приложение “Виселица” не многофункционально, а из базы данных имеет только словарь, поэтому лучше всего использовать фреймворк Kivy и базу данных SQLite.