

Таблица сравнения фреймворков

Фреймворки	Плюсы	Минусы
Kivy	<ul style="list-style-type: none"> <li>• Библиотека мультитач-виджетов</li> <li>• Гибкий графический движок для создания интерактивных интерфейсов</li> <li>• Множество предопределенных GUI-виджетов</li> <li>• Пользовательские виджеты (не использует нативные виджеты)</li> <li>• Кроссплатформенность</li> <li>• Подходит для Android и iOS</li> <li>• Поддержка работы с мультимедиа и сетью</li> </ul>	<ul style="list-style-type: none"> <li>• Недостаточно гибкий для сложных приложений</li> <li>• Может столкнуться с несовместимостью с некоторыми технологиями</li> <li>• Неродной пользовательский интерфейс</li> </ul>
BeeWare	<ul style="list-style-type: none"> <li>• Кроссплатформенность</li> <li>• Подходит для Android и iOS</li> <li>• Подходит для более сложных проектов</li> <li>• Поддерживает собственный пользовательский интерфейс</li> </ul>	<ul style="list-style-type: none"> <li>• Использует нативные интерфейсы</li> <li>• Обновления могут привносить ошибки, исправление которых требует немало времени</li> <li>• Не реализованы востребованные функции</li> </ul>
PyQt	<ul style="list-style-type: none"> <li>• Интеграция с другими библиотеками и фреймворками</li> <li>• Поддержка интернационализации и локализации</li> <li>• Доступ к большому количеству функциональных возможностей из библиотеки Qt</li> <li>• Поддержка мультимедиа, интеграция баз данных</li> </ul>	<ul style="list-style-type: none"> <li>• Плохая и неполная документация</li> <li>• Сложный набор функций</li> <li>• Некоторые инструменты отладки недоступны из-за особенностей работы</li> <li>• Занимает большой объем памяти</li> </ul>

Tkinter	<ul style="list-style-type: none"> <li>• Поддержка распространенных виджетов, используемых в приложениях с графическим интерфейсом</li> <li>• Событийно-ориентированное программирование для обработки взаимодействия с пользователем</li> <li>• Встроенная поддержка изображений и цветов</li> <li>• Кроссплатформенность</li> <li>• Гибок и стабилен</li> </ul>	<ul style="list-style-type: none"> <li>• Не эффективен в работе над масштабными программами</li> <li>• Ошибки переносимости</li> <li>• Ограниченный функционал</li> <li>• Устаревший интерфейс</li> <li>• Не включает расширенные виджеты</li> </ul>
PySide	<ul style="list-style-type: none"> <li>• Поддержка отображения изображений, видео и других типов мультимедиа</li> <li>• Позволяет создавать пользовательский интерфейс</li> <li>• Позволяет использовать основные ресурсы пользовательского интерфейса на основе виджетов</li> </ul>	<ul style="list-style-type: none"> <li>• Не имеет специального инструмента развертывания</li> <li>• Отсутствие надлежащей документации</li> <li>• Не работают некоторые внешние библиотеки</li> </ul>
PySimpleGUI	<ul style="list-style-type: none"> <li>• Возможность использовать для создания сложных пользовательских интерфейсов</li> <li>• Прост в освоении</li> <li>• Мультиплатформенная поддержка</li> <li>• Можно использовать для создания сложных пользовательских интерфейсов</li> </ul>	<ul style="list-style-type: none"> <li>• Создание мобильного приложения осуществимо только с использованием PyDroid3, Termux</li> <li>• Никаких зависимостей от пакетов, кроме tkinter</li> </ul>
Flutter	<ul style="list-style-type: none"> <li>• Красивый, удобный и гибкий интерфейс</li> <li>• Нативность приложений для iOS и Android</li> <li>• Собственный графический движок</li> <li>• Интерфейс легко разбивается на отдельные модули</li> </ul>	<ul style="list-style-type: none"> <li>• Требуется много памяти и мощности компьютера</li> <li>• Интерфейс создается с помощью кода, из-за чего грань между логикой и дизайном гораздо тоньше</li> </ul>

Таблица сравнения СУБД

СУБД	Плюсы	Минусы
MySQL	<ul style="list-style-type: none"> <li>• Высокая гибкость имеющейся системы</li> <li>• Практичность</li> <li>• Поддержка графических интерфейсов для упрощения управления манипуляциями</li> <li>• Поддержка ODBC драйвера MyODBC</li> <li>• Осуществляет частое чтение данных</li> <li>• Высокая производительность</li> </ul>	<ul style="list-style-type: none"> <li>• Ограниченный функционал</li> <li>• Затраты на преобразование</li> <li>• Серьезные последствия при обнаружении критических ошибок</li> </ul>
SQLite	<ul style="list-style-type: none"> <li>• Высокая скорость работы</li> <li>• Хранение электронных материалов в пределах одного файла</li> <li>• Повышенный уровень надежности</li> <li>• Нулевые конфигурации</li> <li>• Небольшой размер исходного файла</li> <li>• Автономное функционирование</li> <li>• Кроссплатформенность</li> </ul>	<ul style="list-style-type: none"> <li>• Ограниченность поддержки типов данных</li> <li>• Отсутствие хранимых процедур</li> <li>• Ограниченность в применении</li> </ul>
PostgreSQL	<ul style="list-style-type: none"> <li>• Осуществляет частое обновление данных</li> <li>• Имеет пользовательский тип данных</li> <li>• Поддержка множества типов данных</li> <li>• Работа с большими объемами</li> <li>• Поддержка сложных запросов</li> <li>• Одновременная модификация базы</li> <li>• Высокая мощность и широкая функциональность</li> <li>• Кроссплатформенность</li> </ul>	<ul style="list-style-type: none"> <li>• Повышенное потребление ресурсов</li> <li>• Отсутствие некоторых функций</li> <li>• Низкая скорость работы</li> <li>• Низкая производительность</li> </ul>
SQLAlchemy	<ul style="list-style-type: none"> <li>• Безопасность. Параметры запросов экранируются</li> <li>• Высокая производительность</li> <li>• Переносимость</li> <li>• Большая гибкость</li> </ul>	<ul style="list-style-type: none"> <li>• Не выполняет большие операции с данными, связанные с объединением нескольких таблиц и разделением базы данных между процессами</li> <li>• Неэффективность</li> </ul>

MongoDB	<ul style="list-style-type: none"> <li>• Гибкая система хранения информации</li> <li>• Высокая производительность при выполнении простых запросов</li> <li>• Простая масштабируемость</li> <li>• Высокая скорость и более высокая доступность</li> </ul>	<ul style="list-style-type: none"> <li>• Отсутствие хранимых процедур и функций (нельзя полностью автоматизировать работу)</li> <li>• Не поддерживает объединения в качестве реляционной базы данных</li> <li>• Если индексация реализована неправильно или имеет какие-либо неточности, MongoDB будет работать с очень низкой скоростью</li> <li>• Ограниченный размер данных и вложенности</li> <li>• Дублирование данных затрудняет обработку наборов данных</li> <li>• Требуется большого объема хранилища</li> </ul>
DynamoDB	<ul style="list-style-type: none"> <li>• Не теряет производительность при масштабировании</li> <li>• Простой API на основе REST</li> <li>• Обеспечивает обновления, резервное копирование и выполнение других административных задач</li> <li>• Бесшовная репликация данных</li> <li>• Не требует базовой инфраструктуры</li> <li>• Быстрое время работы</li> </ul>	<ul style="list-style-type: none"> <li>• Ограниченные возможности запроса</li> <li>• Невозможно использовать соединения таблиц</li> <li>• Трудно предсказать затраты</li> <li>• Ограниченные возможности хранения</li> <li>• Отсутствие модели развертывания</li> </ul>

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