

СРАВНЕНИЕ ТЕХНИЧЕСКИХ СРЕДСТВ ДЛЯ СОЗДАНИЯ НАСТОЛЬНЫХ ПРИЛОЖЕНИЙ

П. В. Черепанов, студент гр. ПИМ-241

Научный руководитель – А. А. Тайлакова, к.т.н., доцент

Кузбасский государственный технический университет имени Т. Ф. Горбачева,
г. Кемерово

В современном мире, где очень сильно развиты облачные технологии, сайты, веб-приложения, пользователи редко отдают предпочтения полноценным компьютерным программам. В силу понятных причин. Пользователю обычно не хочется устанавливать программы, использовать место на своем жестком диске. Поэтому, в основном, разработчики при решении разных прикладных задач, при автоматизации разных бизнес-процессов выбирают путь создания веб-приложения. Однако порой бывают ситуации, когда облачное решение будет излишним. Например, веб-приложение нужно hostить на удаленном сервере. При отсутствии заинтересованного лица или заказчика, готового платить за хостинг, это становится проблемой.

Для создания компьютерных программ существует большое количество языков программирования: C++, C#, Java, JavaScript, Python. Для каждого из этих языков найдутся библиотеки и фреймворки для создания десктопных приложений. У Java JavaFX, у JavaScript Electron - это фреймворк с открытым исходным кодом, который позволяет разработчикам создавать кроссплатформенные настольные приложения.[1]

Для Python разработано большое количество библиотек. PyQt, Tkinter, Kivy – это лишь самые популярные библиотеки.[1] Проведем сравнительный анализ по следующим критериям: простота установки, кроссплатформенность, сообщество и документация, внешний вид, дополнительные функции с помощью таблицы.

Характеристика/Библиотека	PyQT	Tkinter	Kivy
Простота установки	Требуется установка	Входит в стандартную библиотеку Python	Требуется установка
Кроссплатформенность	Windows, macOS, Linux, iOS, Android	Windows, macOS, Linux	Windows, macOS, Linux, iOS, Android
Сообщество и документация	Обширная документация, большое сообщество	Обширная документация, большое сообщество	Обширная документация, большое сообщество
Внешний вид	Современный и привлекательный внешний вид	Устаревший внешний вид	Современный и привлекательный внешний вид
Дополнительные функции	Поддержка многопоточности, интеграция с OpenGL, база данных и т.п.[2]	Базовые функции GUI, ограниченная поддержка дополнительных функций.[4]	Поддержка мультитач экранов, мультимедийные возможности.[5]

PyQT – хороший выбор для создания проекта средней величины. Например, для программного продукта для обработки результатов инструментальных наблюдений устойчивости бортов карьеров. Обеспечение устойчивости бортов карьеров является критически важной задачей в горнодобывающей промышленности. Эта программа нужна специалисту-маркшейдеру для повседневной работы. Программа должна выполнять следующие функции: сохранение данных, выполнение необходимых математических операций, ведение статистики, анализ данных, визуализация. Также нужен экспорт данных в Excel и AutoCAD. PyQT может помочь в реализации этих функций. В компьютерной программе нужно реализовать запись координат реперов. С помощью библиотеки можно быстро создать поля для ввода X, Y, Z и кнопку для записи. С помощью библиотеки sqlite3 можно полученные от пользователя данные сохранять в СУБД SQLite. А с помощью PyQtGraph в приложении можно отображать графики по данным из базы данных.

Программа, созданная на основе PyQT отлично будет выполнять поставленные задачи. Находясь локально на компьютере у маркшейдера, компьютерное приложение облегчит повседневную работу специалиста. Позволит быстрее и эффективнее выполнять свою работу.

Список литературы

1. Electron: офиц. сайт. URL: <https://www.electronjs.org/> (дата обращения: 07.10.2024).
2. PyQT: офиц. сайт. URL: <https://doc.qt.io/qtforpython-6/> (дата обращения: 07.10.2024).
3. Kivy: офиц. сайт. URL: <https://kivy.org/> (дата обращения: 07.10.2024).
4. Tkinter: офиц. сайт. URL: <https://docs.python.org/3/library/tkinter.html> (дата обращения: 07.10.2024).