**Доклад**

**Тема:** Анализ оценок и Визуализация данных из Сетевого Города при помощи программирования

1. **Актуальность темы:** Хотелось бы начать с актуальности темы моего проекта. IT технологии важны для модернизации системы образования, так как они предоставляют доступ к образовательным ресурсам и инструментам независимо от местоположения, обогащают учебный процесс мультимедийными материалами и обеспечивают коммуникацию и сотрудничество между учениками и преподавателями. В пример можно привести WhatsApp, различные видеоуроки, а также компьютерные программы вроде Word.
2. **Цель проекта:** Целью проекта является автоматизация и упрощения какой-либо задачи в образовательном процессе. За такую задачу выбран контроль ученика за оценками.
3. **Задачи:** На данном слайде представлены задачи, а именно этапы создания программы.
4. **ЯП Python:** Язык программирования Python - универсальный язык программирования, использующийся для решения почти любых задач, используется в основном для web-приложений, машинном обучении, а также для многих других. Он используется в моём проекте.
5. **ООП:** Объектно-ориентированное программирование – это подход к программированию, при котором программа рассматривается как набор объектов, взаимодействующих друг с другом. Простой пример. Допустим, что природа – это программа, объектами в ней будут являются камни, палки, животные и т.д. Все они напрямую или косвенно взаимодействуют друг с другом.
6. **Структуры проекта:** Немного о типах архитектур IT проектов.

* Монолитная структура: Программа как одно целое, в начале проекта хороша, потом трудно масштабируется и тестируется.
* Многоуровневая структура: Программа разделена на несколько уровней (слоёв), каждый из которых отвечает за некоторый функционал.
* Микросервисная структура: Программа разделена на множество независимых сервисов, каждый из которых выполняет свой функционал. Обеспечивает высокую гибкость, но требует сложной инфраструктуры и управления.

1. **Выбор многоуровневой:** Её выбор обоснован следующими преимуществами, которые показаны на текущем слайде.

Для данного проекта, эта архитектура подходила больше всего.

1. **Визуализация структуры:** На данном слайде представлена блок-схема работы программы со всеми уровнями. Краткая суть работы программы на схеме.
2. **Установка Python:** Установка языка программирования Python, очень лёгкая и делается в 2 шага. Нужно зайти на официальный сайт, который вы видите на слайде, скачать установщик, а после просто установить.
3. **Установка библиотек:** Библиотеки в Python – это несколько файлов кода, объединённых для получения некоторого функционала. Устанавливаются они тоже довольно просто. Первое, нужно вызвать командную строку, сочетанием клавиш Win+R, прописать в открывшейся строке cmd, после откроется командная строка, которую вы можете видеть на слайде. Далее потребуется ввести следующую команду.
4. **Создание GitHub репозитория:** Для проекта требовался мощный инструмент контроля версий кода, а также их хранения, для этого был использован портал разработчиков GitHub которые предоставляет такие услуги. Нужно было зарегистрироваться на официальном сайте и создать новый репозиторий (зелёная кнопочка) и по мере написания программы добавлять в репозиторий файлы.
5. **Вход в сетевой город и парсинг таблицы:** Программа при её запуске будет автономно, используя логин и пароль, полученный от пользователя, входить в портал сетевого города, и после парсить (копировать) себе таблицу.
6. **Фильтрация и вывод графиков:** Далее из полученной таблицы требовалось удалить такие данные как УП, Н и других значений, которые невозможно отобразить на графике.
7. **БД:** Программе требовалось отдельное место для хранения данных пользователя, было решено организовать базу данных, а часть программы сделать Системой управления этой Базой Данных, у которой будут функции, записи, обновления, получения, удаления данных.
8. **UI:** Понятие пользовательское интерфейса- это интерфейс взаимодействия программы и пользователя. Графическая составляющая программы. В UI входят такие элементы дизайна как: кнопочки, поля ввода, менюшки, и т.д.
9. **Telegram:** В качестве пользовательского интерфейса был выбран популярный мессенджер Telegram по следующим причинам. (Представлены преимущества использования телеграмма в качестве UI требуется выделить третий пункт: предоставляет инструмент для работы мессенджера и ЯП Python.)
10. **Демонстрация**