**Описание архитектуры системы**

Данный документ описывает необходимые сервисы, приложения и технологии для создания курсового проекта.

1. **Дизайн на Figma, получение готовой вёрстки**

**Figma** — это приложение, которое специализируется на предоставлении целого пакета сервисных услуг и функций для разработки интерфейсов различной сложности. Само приложение кроссплатформенное и может использоваться как десктопное приложение, так и браузерное. По своей сути представляет из себя векторный графический редактор, поддерживает большую базу плагинов. С самого начала поддерживал генерацию CSS стилей и кода для мобильных устройств. Большим плюсом является хранение макетов в облаке, и возможность работы одновременно над одним проектом целой командой дизайнеров. Для разработчиков есть удобный экспорт ресурсов (например векторных иконок) в форматы PNG/SVG/JPG, а так же экспорта страниц в PDF файл.

Плагины в Figma служат для того, чтобы облегчить задачу дизайнера и разработчика - расширяя функционал базового приложения Figma при помощи пользовательских расширений (плагинов).

При помощи одного из таких плагинов, FigmaToCode, была сгенерирована html-разметка и наборы css-стилей для будущего сайта.

1. **Запуск веб-приложения при помощи связки Python и Flask**

**Flask** — фреймворк для создания [веб-](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5)п[риложений](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2. Относится к категории так называемых микрофреймворков — минималистичных каркасов веб-приложений, сознательно предоставляющих лишь самые базовые возможности.

Поддерживается установка посредством пакетного менеджера PyPI, версия 1.0 совместима с Python 2.7, Python 3.3 и выше.

При помощи данного инструментария, а именно локального сервера Flask и шаблонизатора Jinja2, будет осуществлено взаимодействие пользователя с веб-приложением (переход по страницам, возможность оставлять комментарии).

1. **Хранение отзывов пользователей в базе данных SQLite**

**SQLite** — компактная встраиваемая СУБД. Слово «встраиваемый» означает, что SQLite не использует парадигмы клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором исполняется программа.

При помощи базы данных и возможностей Flask и его шаблонизатора мы будем сохранять отзывы пользователей в базу данных, а также подгружать отзывы из неё на страницы сайта.

Схема таблицы с комментариями в базе данных:

