**Дипломная работа по курсу «Python – разработчик».**

**Тема дипломной работы: «Анализ и сравнение написания web-приложений с использованием разных фреймворков»**

Содержание

1. Обзор проекта......................................................................................................2
2. Сравнение написания web-приложений с использованием фреймворков Django, FastAPI и Flask.......................................................................................3
3. Заключение..........................................................................................................4

**Обзор проекта**

Предлагается написать простые web-приложения на базе фреймворков Django, FastAPI и Flask, с целью сравнения написания приложений на указанных фреймворках. Приложения (различные сайты) позволяющие подключать базы данных SQLlite3, создавать взаимосвязь между таблицами базы данных, сохранение информации вводимой через приложения в базы данных и вывод информации с использованием HTML-шаблонов.

Предусматриваются механизмы создания персонала и авторизация на страницах сайтов.

**Сравнение написания web-приложений с использованием фреймворков Django, FastAPI и Flask**

1. **Пакеты**

Среди Django, Flask и FastAPI у Django больше всего пакетов, позволяющих повторно использовать код. Это полнотекстовый фреймворк веб-разработки, в отличие от Flask и FastAPI, которые являются минималистичными фреймворками, используемыми для создания быстрых веб-сайтов.

1. **Сообщество**

У Django самое значительное сообщество из-за его широкого использования и популярности после Flask, у которого также процветающее сообщество. У FastAPI, с другой стороны, небольшое сообщество, потому что он относительно новый.

1. **Производительность**

По производительности FastAPI является лидером, потому что ориентирован на скорость, затем следует Flask и, наконец, Django, который не очень быстр.

1. **Гибкость**

Гибкость - это то, что разработчики очень ценят, а Flask более гибкий, чем Django. Fast API, с другой стороны, гибок с точки зрения кода и не ограничивает компоновку кода. Итак, мы можем сказать, что Flask является наиболее гибким из всех трех.

**Заключение**

Эти приложения представляют полнофункциональные платформы для пользователей, различных сайтов по размещению информации о сотрудниках организаций в частности о руководстве. Данные приложения могут служить базой для расширения их функционала в части добавления структурных подразделений с последующим добавлением сведений о персонале, их фильтрацией и выводом на сайте. Ограничение доступа к определенным сведениям без авторизации на сайте.