Дипломная работа по теме: **Анализ и сравнение написания web-приложений с использованием разных фреймворков.**

**Автор:** Макаров Константин Вячеславович.

Содержание:

1. Введение.
2. Основные понятия и определения.
3. Методы и подходы к разработке.

**Введение.**

Веб-приложение — это программа, которую можно открыть через интернет с помощью браузера на любом устройстве. Оно дает пользователям возможность взаимодействовать и может использоваться для разнообразных целей, таких как ведение бизнеса, общение, обучение и многое другое. Использование веб-приложений имеет ряд преимуществ, включая удобство использования, быструю загрузку, доступ к информации из любой точки мира и работу с мобильных устройств.

**Обоснование выбора темы:**

1. Перспективы развития веб-приложений выглядят многообещающе благодаря развитию технологий, которые делают их все более функциональными и интерактивными.
2. Потребности рынка: существует растущая потребность в специалистах, обладающих навыками создания веб-приложений для разных задач.
3. Практическая значимость: Работа в области разработки веб приложения позволит применить полученные знания на практике и получить опыт в данном направлении.
4. Личный интерес и потенциальные перспективы: Использование различных инструментов и создание продуктов. Опыт создания интересных и разносторонних проектов с использованием фреймворков Django, FastAPI, Flask. Кроме того за основу взят небольшой бизнес члена семьи в сфере услуг – салон красоты.

Выбор темы «**Анализ и сравнение написания web-приложений с использованием разных фреймворков»** обусловлен ее актуальностью, потребностями рынка, практической значимостью и личным интересом, что делает эту тему подходящей для проведения дипломной работы.

Существует огромное количество видов веб-приложений: от простейших инструментов для создания текстов до сложных корпоративных систем. Выбор подходящего варианта для вашего бизнеса должен основываться на ваших индивидуальных потребностях и целях.

Перспективы развития веб-приложений выглядят многообещающе благодаря развитию технологий, которые делают их все более функциональными и интерактивными.

**Определение цели и задач исследования:**  
  
Цель исследования: Разработать простые веб-приложения с использованием Django, Flask и FastAPI, провести их сравнение.  
  
Задачи исследования:  
  
1. Обзор фреймворков для написания веб приложений.

2. Сравнение приложений.

3. Написание дипломной работы: Составить дипломную работу, включающую в себя введение, обзор литературы, методологию и результаты сравнения приложений с использованием Django, Flask и FastAPI.

**2.Основные понятия и определения.**

**Веб-приложение** — [клиент-серверное](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_%E2%80%94_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) приложение, в котором [клиент](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) взаимодействует с [веб-сервером](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) при помощи [браузера](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80). Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется преимущественно на сервере, обмен информацией происходит по [сети](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C). Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной [операционной системы](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) пользователя, поэтому веб-приложения являются межплатформенными службами. Веб-приложения стали широко использоваться в конце [1990-х](https://ru.wikipedia.org/wiki/1990-%D0%B5_%D0%B3%D0%BE%D0%B4%D1%8B) — начале [2000-х годов](https://ru.wikipedia.org/wiki/2000-%D0%B5_%D0%B3%D0%BE%D0%B4%D1%8B).

Веб-приложение состоит из клиентской и серверной частей, тем самым реализуя [технологию «клиент-сервер»](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80). Тут HTML-страницы генерируются на лету в зависимости от запроса пользователя. Почтовые клиенты, соцсети, поисковики, интернет-магазины, онлайн-программы для бизнеса, это все веб-приложения.

Клиентская часть реализует [пользовательский интерфейс](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81), формирует запросы к серверу и обрабатывает ответы от него.

Серверная часть получает запрос от клиента, выполняет вычисления, после этого формирует [веб-страницу](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) и отправляет её клиенту по сети с использованием [протокола](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) [HTTP](https://ru.wikipedia.org/wiki/HTTP).

Веб-приложения можно разделить на несколько типов, в зависимости от разных сочетаний его основных составляющих:

1. **Backend** (бэкенд или серверная часть приложения) работает на удаленном компьютере, который может находиться где угодно. Она может быть написана на разных языках программирования: PHP, Python, Ruby, C# и других. Если создавать приложение используя только серверную часть, то в результате любых переходов между разделами, отправок форм, обновления данных, сервером будет генерироваться новый HTML-файл и страница в браузере будет перезагружаться.
2. **Frontend** (фронтенд или клиентская часть приложения) выполняется в браузере пользователя. Приложение может состоять только из клиентской части, если не требуется хранить данные пользователя дольше одной сессии. Это могут быть, например, фоторедакторы или простые игрушки.

Веб-приложения обычно используют базы данных для хранения данных. Это позволяет хранить информацию даже после того, как вы закроете свой браузер. Эти базы данных могут находиться на сервере разработчика или в облаке. Интернет-приложения включают в себя важную составляющую в виде API (программный интерфейс веб-приложения), который позволяет им взаимодействовать со множеством сторонних систем, таких как платежные системы, CMS, социальные сети и множество других.