**Автономная некоммерческая общеобразовательная организация "Физтех-лицей"
(АНОО «Физтех-лицей» им. П.Л. Капицы)**

XX научно-практическая конференция

«Старт в инновации»

«Собственное веб-приложение на Flask как бытовой инструмент»

Выполнили:
Томилов Михаил Федорович
Литвинов Владимир Юрьевич
Руководитель:
Литвинов Юрий Львович

Московская область, г. Долгопрудный

2021 г.

Оглавление

Введение

Наши родители покупают продукты крупными партиями, ведь так получается дешевле и быстрее. Однако, необходим список продуктов, чтобы не забыть купить необходимое. Мы не раз пробовали составлять списки в бумажном виде, но выяснили, что электронными пользоваться гораздо удобнее. Вскоре стало понятно, что пользоваться таблицами неудобно, поэтому отец одного из участников создал приложение-ассистент для покупок в локальном версии, и через какое-то время выявились недостатки локального расположения приложения, такие как очень сложная реализация единого списка на нескольких пользователей, необходимость адаптации приложения под каждое устройства по отдельности и тому подобное. Тогда возникла задача переноса этого приложения на веб-платформу и последующего развития уже в таком виде.

Сейчас человек использует в быту все больше инструментов, основанных на информационных технологиях, часть из которых переходит на веб-платформу, ведь в большинстве случаев это оказывается практичнее. Однако, не всегда существует инструмент, отвечающий всем индивидуальным требованиям.

У нас возникла именно такая проблема, и мы решили сами создать веб-приложение и выяснить, эффективен ли такой подход.

Цель: подтвердить или опровергнуть целесообразность самостоятельного создания веб-приложений.

Задачи:

1. Изучить разработку веб-приложений
2. Создать удобное в личном использовании веб-приложение на Flask, используя базу данных SQLite и Bootstrap
3. Оценить затраченное время и усилия, оценить полученный результат.

Объект исследования – веб-приложения

Предмет исследования – Python, Flask, SQLite, Bootstrap

Гипотеза:

Самостоятельное создание веб-приложений целесообразно, но в большинстве случаев есть готовое решение, удовлетворяющее всем потребностям и не требующее трудозатрат.

Методы:

- изучение и анализ литературы;
- программирование на Python
- изучение обратной связи от использования приложения
- оценка затраченного времени и усилий

Практическое применение работы: Созданное нами приложение будет использоваться нашими родителями при закупках.

Глава 1. Как устроены веб-приложения

Работа веб-приложений осуществляется на трех уровнях - Presentation Layer, Application Layer, Data Layer. На Presentation Layer происходит представление пользователю в удобном для него виде результатов работы приложения - то есть ответов на его запросы, также отправленные с этого уровня через различные формы. Application Layer осуществляет обработку запросов пользователя и отправку ответов, отправляя запросы и получая ответы от Data Layer, где хранятся данные пользователей и приложения.

Уровень Presentation обычно представляет из себя какое-то количество простых страниц, которые составляются на основе шаблонов и ответов от уровня Application и на которых размещены веб-формы как способ передачи уровню Application запросов. То есть,

происходит такое “общение”: пользователь отправляет через форму запрос серверу и получает от него ответ в сформированной странице.

Application Layer может выглядеть очень по-разному в зависимости от приложения, но выполняет одинаковые функции. Очень часто базируется на выбранном нами языке Python и фреймворке Flask, так как относительно просты в изучении и надежны в работе. Работа этого уровня обычно налаживается через хостинги, предоставляющие услуги по аренде удаленных серверов. Сервера сильно напоминают обычные домашние компьютеры, но работают на специально адаптированных операционных системах.

Data Layer почти всегда является централизованной базой данных с ее системой управления. В первую очередь, базы данных можно разделить на большие группы по моделям данных, на которых они основаны, например, реляционные, иерархические и сетевые. Наиболее распространенный тип - реляционные.

Глава 2. Реализация нашего приложения

2.1. Общее устройство

В нашем веб-приложении для реализации Presentation Layer мы использовали HTML5 - язык разметки страниц и Bootstrap - набор HTML - шаблонов оформления, упрощающих разметку.

Для Application Layer мы использовали язык Python и Flask- фреймворк для Python, чьим расширением и является Bootstrap. Фреймворки - это платформы, задающие структуру приложения.

Data Layer в нашем случае представляет из себя базу данных и её систему управления SQLite3.

Поскольку этот проект требует множество дополнений для Python, мы посчитали разумным использовать виртуальную среду venv (virtual environment). Когда устанавливаются дополнения для Python, меняется конфигурация для интерпретатора, представляющего код Python компьютеру в двоичном виде. Для разных проектов необходимы разные конфигурации, которые могут существовать только в разных средах, поэтому и создаются виртуальные среды.

Наше приложение существует в виде пакета, которым считается папка app, содержащая файл __init__.py, определяющий, как этот пакет вызывать для использования. В папке проекта Shopster находится сам файл приложения shopster.py, config.py - файл, в котором хранятся значения конфигурирующих приложение параметров, также файл базы данных app.db и связанная с ним папка migrations, служебная папка виртуальной среды venv, ну и сам пакет app, в котором содержаться основные компоненты приложения. Теперь подробнее об этих файлах и папках:

-shopster.py практически пуст, так как большая часть приложения разделена на файлы в пакете app, поэтому всё, что выполняется в основном файле приложения - это импорт этого модуля

-app содержит в себе следующее:

- папку templates со всеми HTML-шаблонами приложения, относится к уровню Presentation
- __init__.py, сообщающий, как вызывать приложение
- forms.py - файл, в котором задаётся вид форм и отправка данных из этих форм, является связующим звеном между уровнями Presentation и Application
- models.py, связывающий уровни Data и Application, о нем и о работе базы данных будет рассказано в отдельном разделе
- routes.py - файл, который определяет ответы на запросы от Presentation Layer, по которым и формируются веб-страницы на основе шаблонов, а также отвечает за получение данных из форм

-папке venv в рамках этой работы неинтересна, так как при создании большинства приложений любительского уровня в ней ничего не меняют

-о config.py всё было расказано раньше, он не представляет особого интереса

-об app.db будет рассказано позже в разделе “Базы данных”

2.2. Базы данных

Мы использовали реляционную базу данных. Структуру данных в реляционных базах можно представить в виде некоторого количества таблиц, какие-то ячейки из которых связаны с другими таблицами - relationship(связь), благодаря чему тип данных, на котором основаны эти базы данных, и получил своё название. В файле models.py и формируются эти таблицы и связи, а сама база находится в файле app.db. В нашей базе существуют таблицы User - для пользователей, Product - для продуктов в списках и Category - для категорий продуктов.

В процессе работы над приложением структуру данных в базе приходится изменять, а ввиду устройства реляционных баз данные напрямую перенести в новую структуру невозможно. Для этого используются миграции баз данных, и их работа - одна из сложнейших для изучения тем в создании веб-приложений. Папка migrations хранит историю этих миграций.

2.3. Внешний вид сайта

В этом разделе пойдет речь о том, как устроен Presentation Layer в нашем приложении. Опыт использования локальной версии показал, что требуется две различные страницы списка для составления списка и просмотра при совершении покупок. Поэтому у нашего приложения есть следующие HTML-шаблоны и страницы:

-base.html единственный шаблон из тех, что будут перечислены, которому не соответствует отдельная страница, он включается во все страницы и отвечает за отображение верхнего меню, которое обеспечивает навигацию по всем остальным страницам и отображает имя текущего пользователя

-in_store.html отвечает за список продуктов для просмотра в магазине, является облегченной версией страницы для составления списка, в отличие от основной версии отображает только продукты с количеством, больше нуля и не позволяет удалять продукты

-index.html отвечает за страницу составления списка, отображает список продуктов по категориям, позволяет менять их количество, название и закупщика, имеет фильтр по закупщикам и возможность полного удаления продукта

-login.html отображает форму входа, однако, формы регистрации нет, хоть и ее можно легко добавить и наладить обработку, так как в ближайшее время планируется только ручное добавление в базу данных, а иначе при использовании хостинга будет опасность появления лишних пользователей, которые могут превысить лимит выделенных мощностей, который невысок ввиду использования бесплатного хостинга.

-product_form.html отображает форму добавления и изменения продуктов

Заключение

В результате научной деятельности нам удалось создать веб-приложение “Shopster”, отвечающее требованиям наших семей. На создания этого приложения ушло немного времени - около пяти месяцев неактивной работы около трёх часов в каждый выходной, что показало эффективность создания веб-приложения самому.

Проанализировав результаты научной работы, мы выяснили, что у создания веб-приложения есть плюсы и минусы.

Плюсы:

1. Нам удалось создать удобное в использовании приложение для членов наших семей.
2. Мы освоили материал, который может пригодиться в будущем.

3. В дальнейшем будет возможность усовершенствовать приложение и монетизировать его.

Минусы:

1. Для создания своего приложения необходимо время.
2. На освоение темы, необходимой для приложения, было потрачено большое количество сил.

По итогам нашей научной работы мы подтвердили целесообразность создания в определенных случаях своего приложения, облегчающего быт. Однако, мы потратили время и силы на его создание. В большинстве ситуаций, отличных от нашей, существует большое количество готовых решений, поэтому в таких случаях создавать приложения бессмысленно, если не учитывать пользу от изучения этой темы, знания по которой могут впоследствии сыграть роль в выборе специальности.

Источники:

1. **https://habr.com/ru/post/346306/**
2. **https://stepik.org/course/67/info**