Пояснительная записка к проекту Webserver: «Продажа мотоциклов».

* Автор: Софья Чекмаева, ученица 9 класса «А», школы № 1080
* Актуальность проекта:

В настоящее время многие люди, как активные пользователи сети Интернет, сталкиваются с необходимостью выбора того или иного товара или же услуги через компьютер. Однако удобство некоторых сайтов (из опыта автора проекта и его знакомых) оставляет желать лучшего. Поэтому задумка данной работы заключается в изучении способов разработки веб-приложений, а также в создании собственного сайта (в упрощённой форме) по продаже товаров.

* Цель:

Создать модель веб-сервера для работы с сервисом продажи мотоциклов

* Задачи:

1. Изучить источники информации, документацию, необходимую для реализации задумки, проанализировать аналогичные продукты
2. Продумать и построить архитектуру проекта: определить количество файлов, их содержание, найти подходящие шаблоны
3. Продумать структуру основного файла, через который будет осуществляться связь с остальными частями программы
4. Провести тестирование программы, выявить ошибки и их причины, поработать над устранением возникших проблем
5. Проверить работоспособность программы после исправления ошибок
6. Оформить отчёт по итогам проектной работы: предоставить записку, поясняющую суть и способы реализации задумки, и презентацию, отражающую ход работы и ключевые моменты в нём
7. Представить результаты работы научной общественности, защитить проект

* Методы и описание реализации программы:

В конечном варианте работы «Webserver: Продажа мотоциклов» автор попытался использовать все технологии, которые были упомянуты в курсе по данной теме. Проект представлен набором файлов разного содержания и формата. Вот информация о некоторых из них:

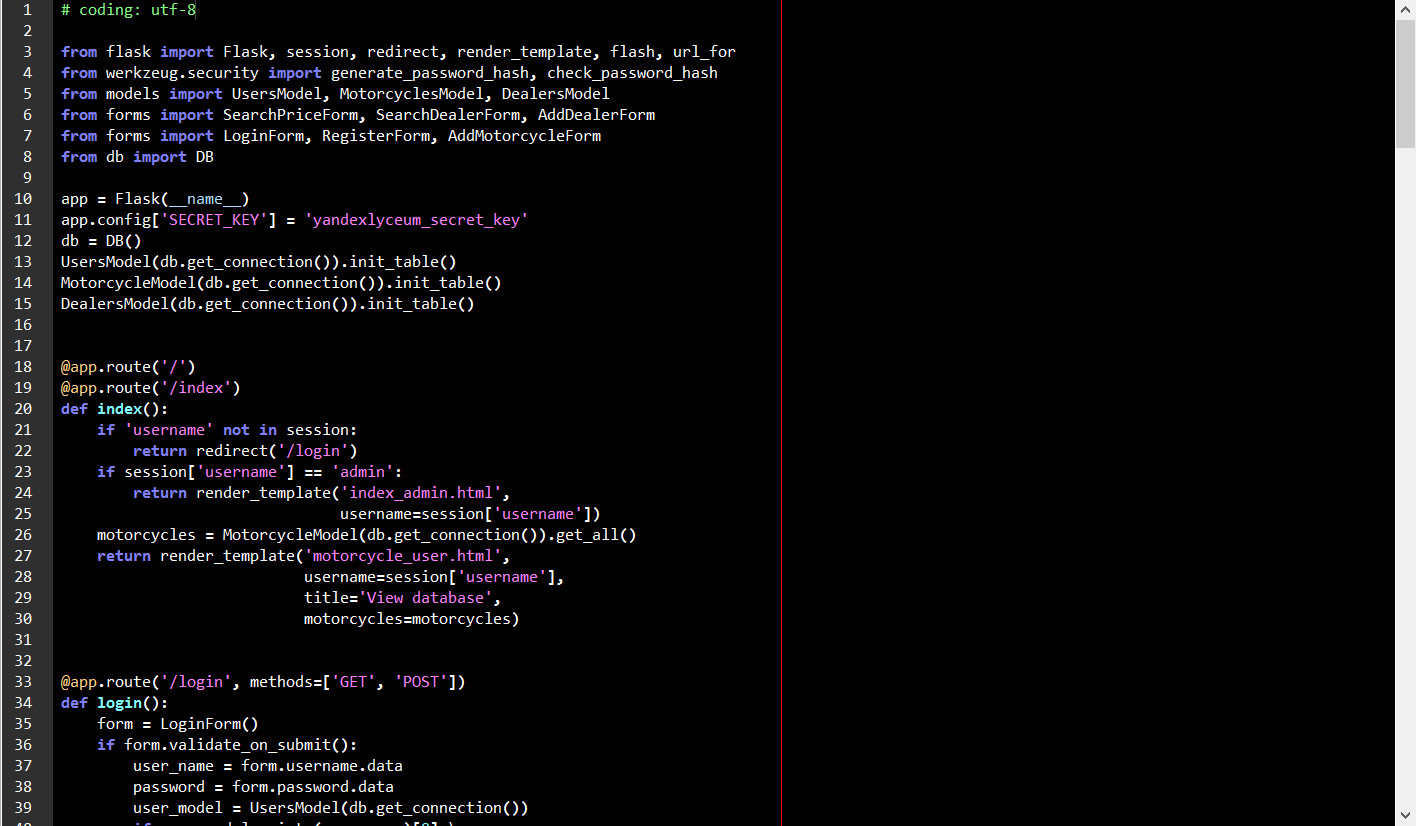
1. index.py - Самый главный файл веб-сервера Flask, где прописаны все пути, по которым пользователь или авторизованный дилер сможет "перемещаться" по странице.В нём использованы модули flask (и его составляющие: Flask, session, redirect, render\_template, flash, url\_for), werkzeug.security и осуществляется работа с содержимым других файлов (таких как models.py, forms.py, но о них чуть позже)
2. models.py – в данном файле реализованы классы UsersModel, DealersModel и MotorcycleModel, которые принимают модели данных, то есть показывают, каким образом хранится информация в базе данных sqlite и как её оттуда извлечь
3. motorcycles.db - файл формата db с таблицами, в которых содержится информация о мотоциклах
4. db.py - данный файл создан для чтения информации из файла "motorcycles.db" на Python, при этом используется модуль sqlite3
5. forms.py – здесь собраны все классы FlaskForm, созданные для авторизации, регистрации, внесения данных о пользователе, мотоцикле, диллерах

Кроме того, перейдя по ссылке (см. в конце пояснительной записки) в репозиторий автора, можно увидеть папку templates (в ней в формате HTML хранятся все шаблоны, необходимые для работы программы) и папку static (содержит весь контент для работы программы, а именно скрипты и изображения).

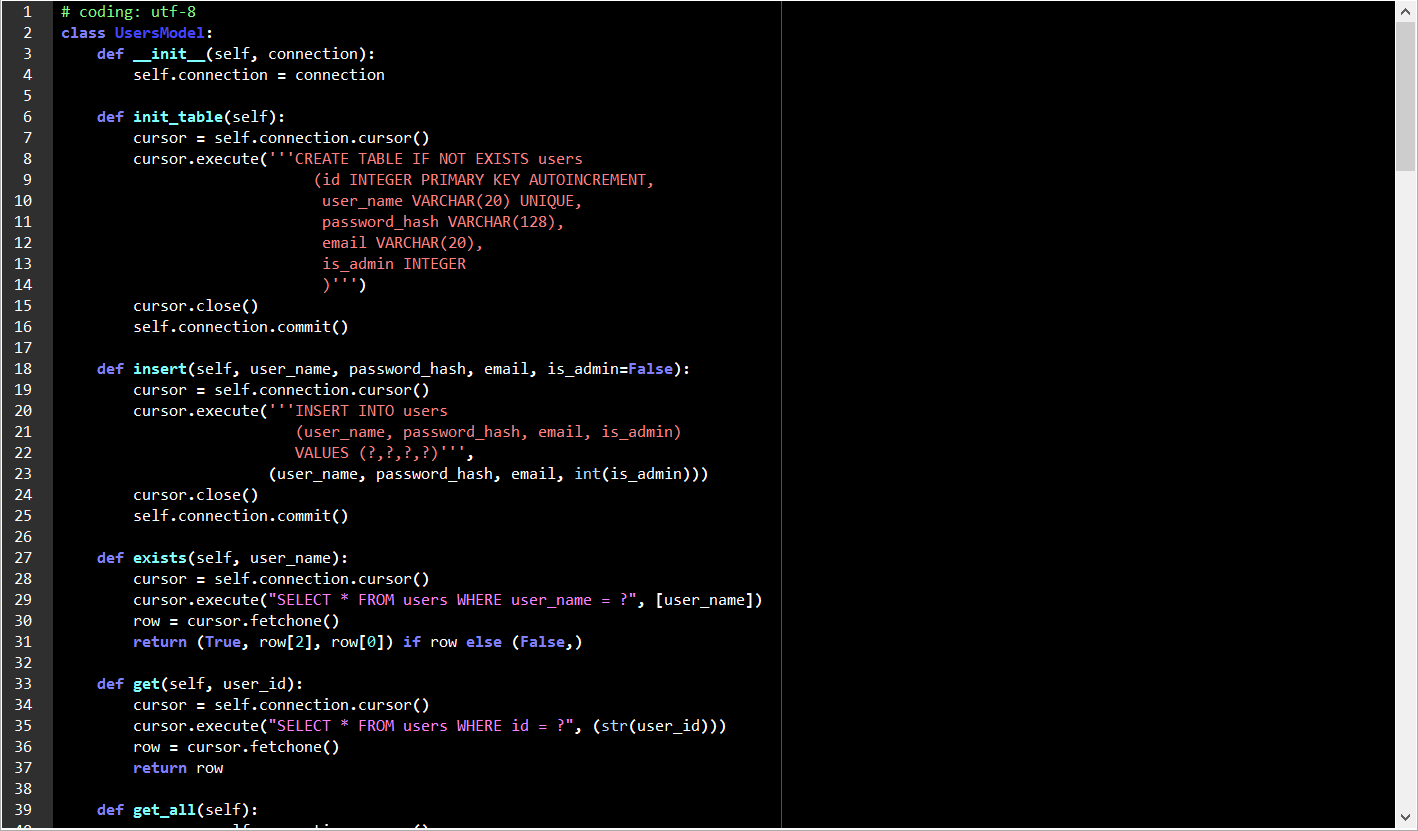
* Ход работы:

Здесь представлена лишь часть кода – некоторые используемые методы и конструкции:

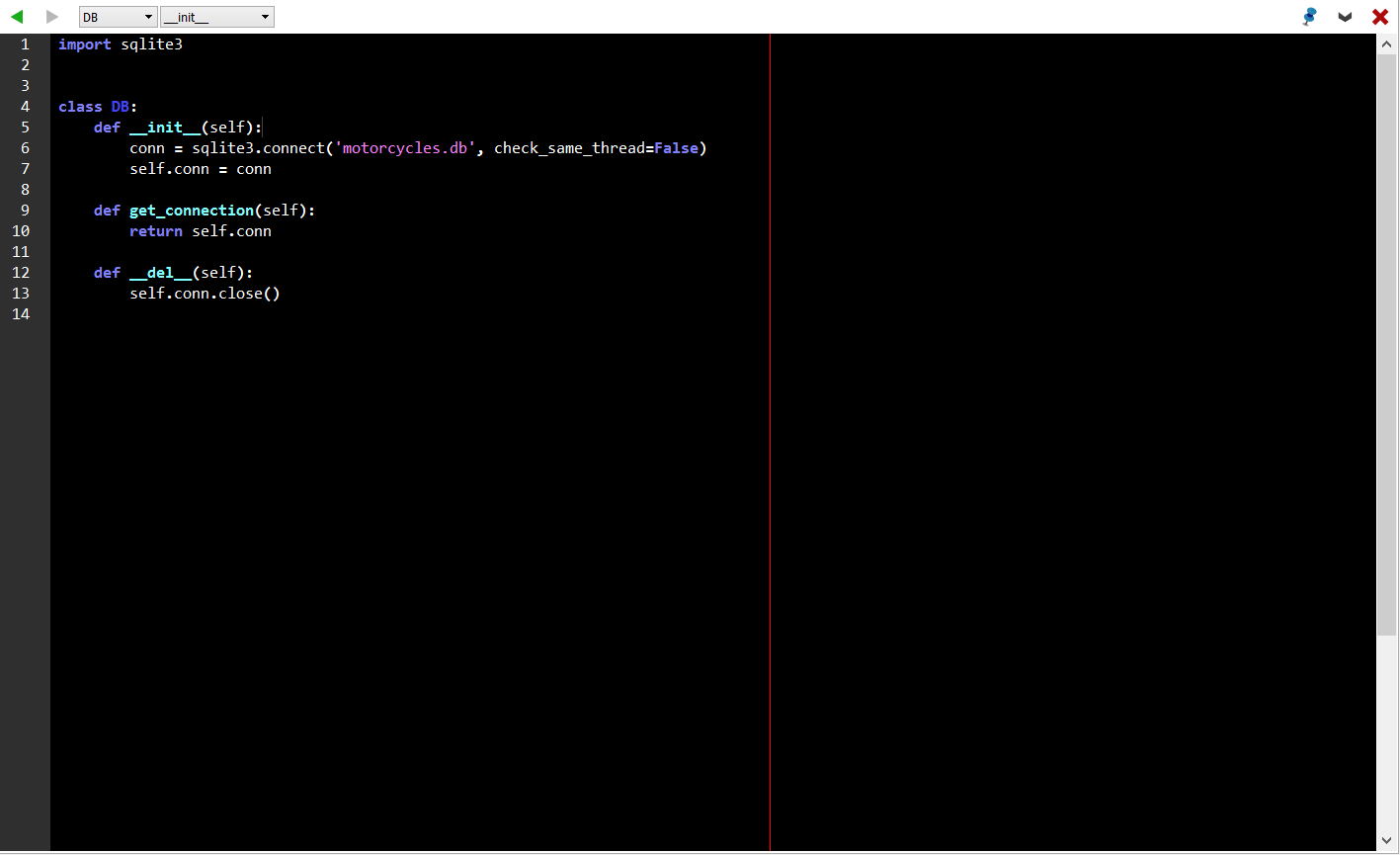
Основной код (из файла index.py):



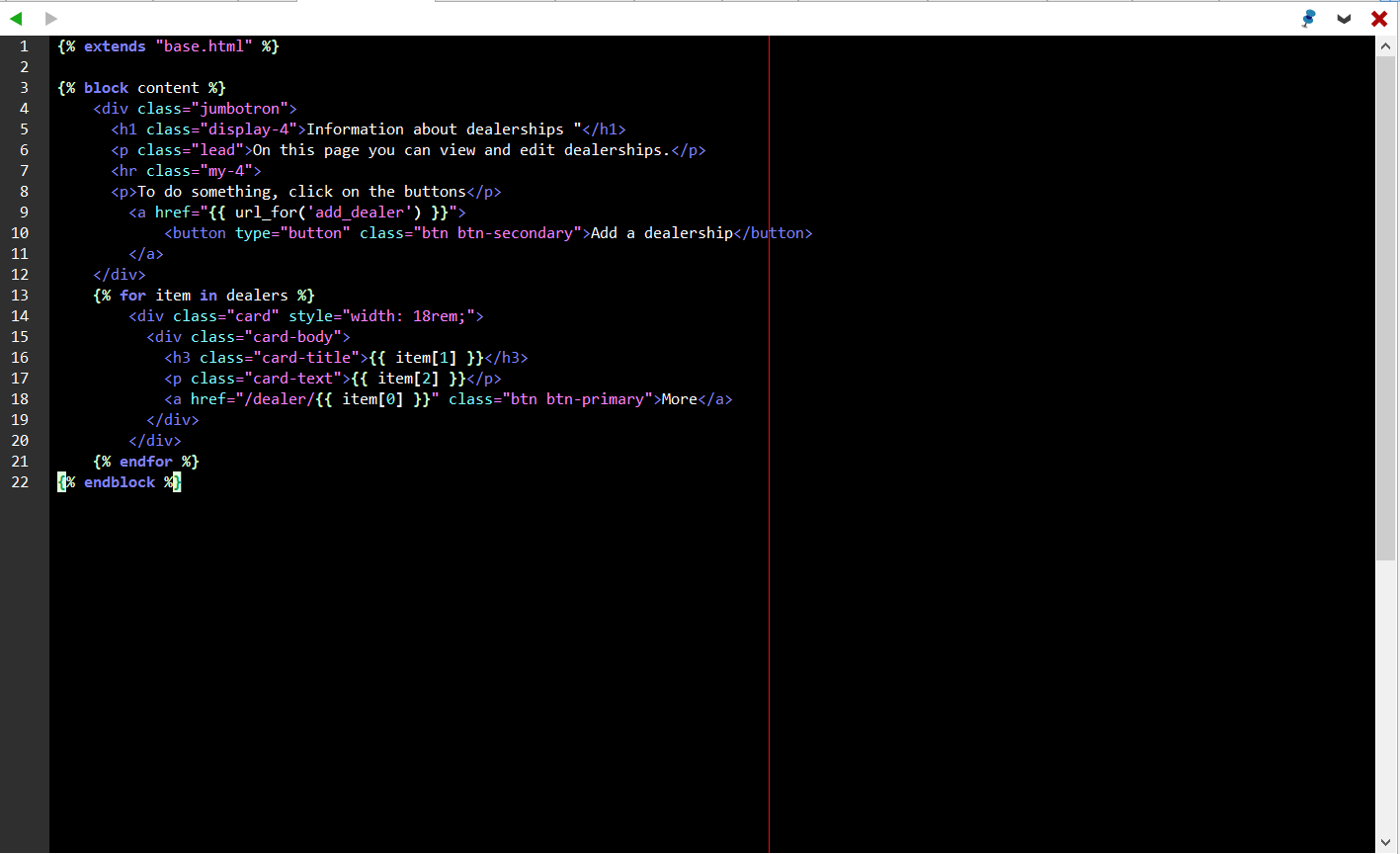
Классы – модели данных из файла models.py:

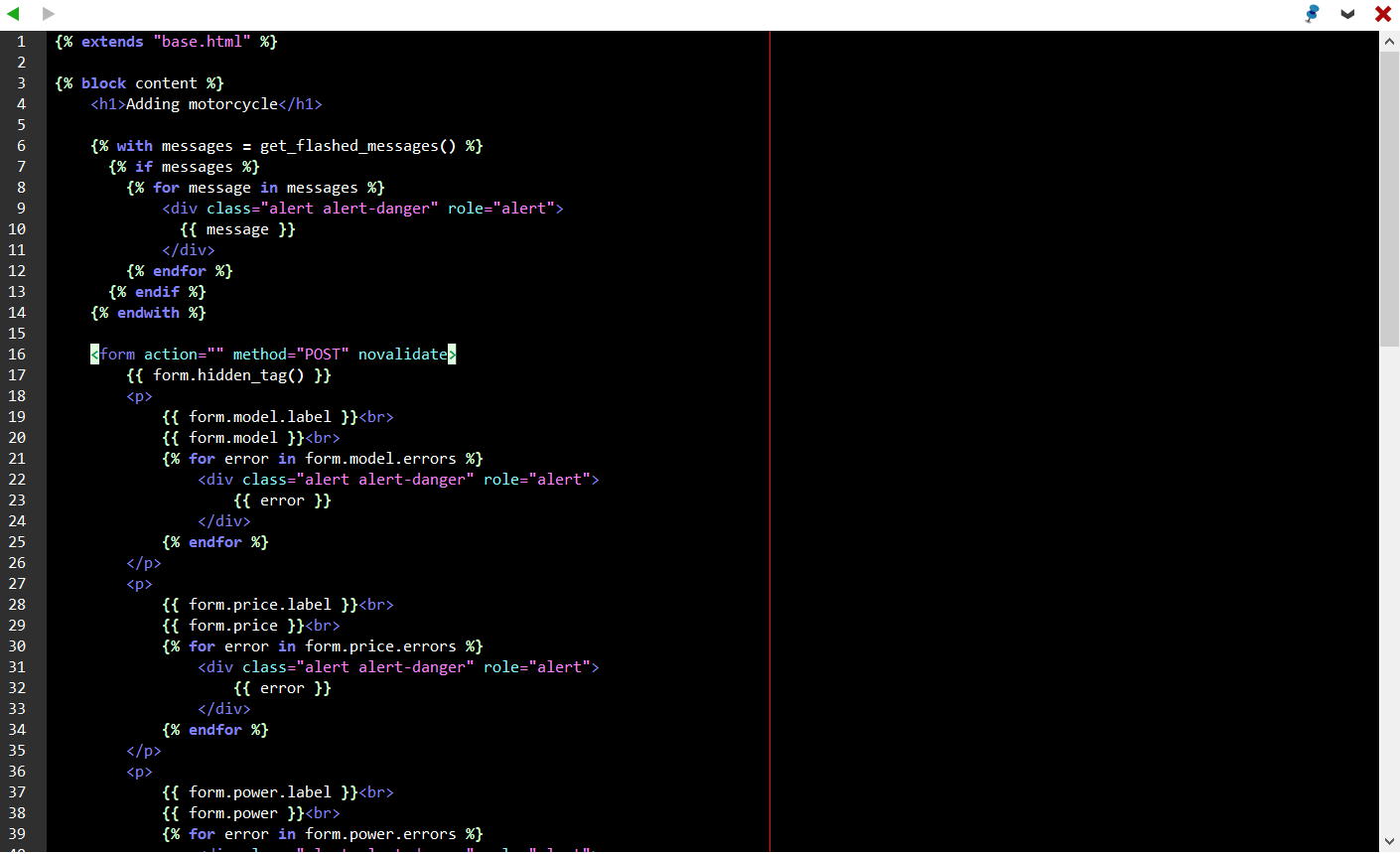


Простая реализация чтения файла формата db с помощью sqlite3:



И несколько шаблонов:





* Ссылка на репозиторий в GitHub:

https://github.com/Chekmaeva/Project-WebServer