Пояснительная записка к проекту «Novito (WEB SERVER + API)»

Авторы проекта: Подзизей Александр, Васильев Георгий

**Идея проекта:**

Вебсайт «Novito» - простой и понятный сервис для размещения объявлений и поиска необходимых товаров

На сайте вас встретит:

* Главная страница со всеми опубликованными объявлениями
* Возможность регистрации и входа в уже существующий аккаунт по логину и паролю
* Страница создания объявлений
* Страница редактирования опубликованного вами объявления

**Описание технологий:**

* Bootstrap для различных элементов сайта (в т.ч. дизайн домашней страницы)
* Шаблоны: base.html для главной оболочки сайта, flask-wtf для логина и регистрации
* ORM-модели: создание БД novito.db, таблиц markets, users и categories в ней
* Регистрация и авторизация: занесение данных пользователя в таблицу users БД, запоминание его пароля и следовательно возможность входа на страницах регистрации и авторизации соответственно
* Загрузка и использование файлов: загрузка картинок при создании объявления, их просмотр другими пользователями
* API-REST: люди, не причастные к созданию проекта, могут взаимодействовать с БД путем отправки/получения json-файлов
* Хранение данных: novito.db как главная БД сайта в целом. Хранение данных пользователя, конкретных объявлений и их категорий. Таблицы: users, markets, category
* Хостинг: сайт хостится на pythonanywhere по адресу <https://flairy.pythonanywhere.com>

**В работе использованы следующие библиотеки:**

1. Flask
2. Flask-Login
3. Flask-RESTful
4. Flask-WTF
5. Wtforms
6. Data
7. Forms
8. Sqlalchemy
9. sqlalchemy-serializer

**Описание реализации**

**Обработчики flask\_app:**

* Register – регистрация
* Login – авторизация
* Logout – выход из аккаунта
* Add\_market – создание нового объявления
* Edit\_market – редактирование пользовательских объявлений
* Delete\_market – удаление объяления
* Ad – страница конкретного объявления
* Confirm – подтвердить действие на сайте
* Index – домашняя(главная) страница сайта

**API-REST:** Шесть классов для обращения к трём таблицам: по два класса на каждую таблицу. Один позволяет работать со списком объектов из таблицы, а второй - с одним объектом

**Внешние функции:**

**Main.py**

* Main – инициализация БД, запуск работы сервера

**Create\_db.py**

* Main – подключение к БД, заполнение ее данными
* Add\_user, add\_market – вспомогательные функции для заполнения таблиц БД

**Db\_session.py**

* Global\_init – инициализация БД