**Пояснительная записка**

**Проблема**

Отсутствие ресурса, где жители люди могли бы обмениваться и получать информацию разного вида.

**Цель**

Создать WEB-приложение, где посетители могут читать новости, делиться своими новостями, задавать вопросы и получать ответы, а также продавать свои товары

**Описание**

Проект представляет собой Flask-приложение, где используются базы данных SQL при помощи Flask-sqlalchemy и стороннее API. Классы представлены моделями для работы со следующими таблицами базы данных **website.db**:

* “users” – таблица пользователей – модель User, поля: id (Уникальный идентификатор пользователя), name (Имя или логин), hashed\_password (Хешированый пароль), created\_date (Дата создания пользователя)
* “news” – таблица пользовательских записей – модель News, поля: id (Уникальный идентификатор записи), user\_id (Id автора), title (Заголовок), text (Содержание), created\_date (Дата создания), image (Имя изображения в директории), likes, dislikes (Кол-во лайков и дизлайков соотв.), who\_liked (Id пользователей, поставивших лайк), who\_disliked (Id пользователей, поставивших дизлайк)
* “threads” – записи на форуме – модель Threads, поля: id (уникальный Идентификатор), user\_id (Id автора), theme (Тема для новой ветки на форуме), text (Содержание), answer\_to (Id записи, на которую отвечают), ancestor\_thread\_id (Id родительской темы на форуме), created\_date (Дата создания), image (Имя изображения в директории)
* “sells” – таблица товаров на маркете – модель Sells, поля: id, user\_id, name (Название товара), description (Описание), created\_date, money (стоимость), address (Адрес продавца), contact (Номер телефона или email продавца), category (Категория товара)

Классами также являются формы, находящиеся в каталоге **forms/,** сделанные при помощи flask-wtf:

* AnswerForm – для ответа на запись в форуме
* BlogCreationForm – для создания записей в блоге
* CreateThreadForm – для создания новой ветки на форуме
* LoginForm – вход пользователя в аккаунт
* Market – создание объявления на маркете
* RegisterForm – регистрация пользователей
* Filter и ThemeChoose – для фильтра объявлений на маркете или записей на форуме

Был создан класс **NewsFromInternet** для парсинга данных с сайта newsvl.ru.

Само приложение находится в файле **main.py** и является обработчиком следующих страниц:

* “/” – главная, шаблон base.html
* “/register” – регистрация, шаблон register.html
* “/login” – вход в аккаунт, шаблон login.html
* “/create\_blog” – создание записи в блоге, шаблон create\_blog.html
* “/blogs” – показ всех записей в блогах пользователей, шаблон blogs.html
* “/blogs/<blog\_id>” – страница записи в блоге, шаблон blog\_page.html
* /forum – форум, шаблон forum.html
* /forum/<branch\_id> - страница ветки в форуме, шаблон branch\_page.html
* /aswer\_to/ - ответ на сообщение в форуме, шаблон answer.html
* /create\_forum\_branch – создание новой ветки в форуме, шаблон create\_thread.html
* /userprofile/<user\_id> - центр активности пользователя, шаблон userprofile.html
* /news – новости с сайта vl.ru, шаблон news.html
* /news/<news\_id> – страница новости с сайта, шаблон news\_page.html
* /market – маркет, шаблон market.html
* /sell – создание объявления о продаже, шаблон sell.html
* /market/<sell\_id> - страница объявления о продаже, шаблон sell\_page.html

Обработчики действий:

* /like /dislike – поставить лайк/дизлайк записи в блоге
* /logout – выйти из аккаунта
* /delete\_from\_market – удалить объявление с маркета

HTML-шаблоны находятся в директории **templates/,** загруженные изображения – в **static/uploads/,** стилевые файлы – в **static/style/.**

**Выводы**

После завершения работы, цели были достигнуты, проблема решена. Перспективы развития заключаются в улучшении работоспособности приложения, а также в усилении безопасности хранимых данных.