

Документация проекта WL

Введение

Данный проект представляет собой веб приложение, с использованием микросервиса python flask. Данное приложение состоит из нескольких модулей, обладающих собственным уникальным функционалом, шаблонами и формами, для отображения, создания новых объектов класса, либо их редактирования. Все шаблоны расширяют базовый шаблон base.html

Каждый модуль представляет из себя папку в которой файл blueprint_”название модуля”, шаблоны находятся в папке templates/”название модуля”, и файл forms.py в нем описаны объекты наших html страниц, через которые мы передаем запросы к базе данных. Присутствует система миграций для удобного обновления базы данных, настройки которой находятся в файле manage.py

Модели

Все классы описаны в одном файле models.py, для удобного взаимодействия между ними

Метод

```
def load_user(user_id)
```

Данный метод на вход получает id пользователя, а возвращает объект с данным id

Класс-пользователь

```
class User(db.Model, UserMixin)
```

наследуется от модели базы данных и защищенного протоколом flask_login Usermixin, представляет из себя объект пользователя нашего веб приложения

Метод класса пользователь-новое сообщение

```
def new_messages(self)
```

метод который возвращает последнее сообщение написанное пользователю

Метод класса пользователь который запрашивает токен сброса

```
def get_reset_token(self, expires_sec=1800):
```

Статический метод класса пользователь

```
def verify_reset_token(token):
```

Метод проверяет запрошенный токен и возвращает объект пользователя в случае удачи

Метод класса пользователь, для получения строкового представления объекта

```
def __repr__(self):
```

Класс пост

```
class Post(db.Model):
```

Класс постов, наследуется от базы данных, представляет из себя объявление пользователя, вместе с YouTube ссылкой творчества пользователя

Класс сообщений

```
class Message(db.Model):
```

Класс сообщений, наследуется от базы данных представляет из себя приватное сообщение между двумя пользователями

Конфигурация

Конфигурация описана в файле condig.py

Класс конфигурации

```
class Configuration(object):
```

Класс в котором идет подключение базы данных через SQLALCHEMY
И прописаны ключи для безопасной авторизации

Представление

Общая визуализация описана в файле view

Отвечает за отображение главной страницы

```
@app.route('/')
```

```
def index():
```

Отвечает за отображение ошибки 404

```
@app.errorhandler(404)
```

```
def error_404(error):
```

Отвечает за отображение ошибки 403

```
@app.errorhandler(403)
```

```
def error_403(error):
```

Отвечает за отображение ошибки 500
`@app.errorhandler(500)`
`def error_500(error):`

Запуск приложения

Приложение запускается благодаря файлу `start.py`
Подключаются все модули приложения

Модуль пользователи `users`

Данный модуль отвечает за операции над пользователями

Метод

`def register():`
отвечает за переход на страницу регистрации
и создание нового пользователя

Метод

`def login():`
отвечает за переход на страницу авторизации
и авторизации пользователя

Метод

`def logout():`
отвечает за выход авторизованного пользователя

Метод

`def account():`
отвечает за переход на страницу профиля

Метод

`def user_posts(username):`
отвечает за переход на страницу постов выбранного пользователя

Метод

`def user_messages(username):`
отвечает за переход на страницу сообщений выбранного пользователя

Метод

```
def reset_request():
```

отвечает за сброс пароля

Метод

```
def reset_token(token):
```

отвечает за создание нового пароля

Метод находится utils.py модуля users

```
def save_picture(form_picture):
```

отвечает за загрузку аватарки пользователя

Метод находится utils.py модуля users

```
def send_reset_email(user):
```

отвечает за сброс email адреса

Модуль постов posts

Метод

```
def new_post():
```

отвечает за создание нового поста

Метод

```
def post(post_id):
```

отвечает за отображение поста

Метод

```
def update_post(post_id):
```

отвечает за изменение поста

Метод

```
def delete_post(post_id):
```

отвечает за удаление поста

Модуль сообщений messages

Метод

```
def send_message(username):
```

отвечает за отправку сообщений

Метод

```
def message(message_id):  
    отвечает за отображение сообщений
```

Метод

```
def index():
```


Страница со всеми сообщениями

Модуль сообщений main

Метод

```
def home():
```


отвечает за отображение домашней страницы

Метод

```
def about():
```


отвечает за отображение страницы о проекте