Пояснительная записка

Проект: “Мой фотоальбом”

Введение

Проект представляет собой Web сайт, на котором пользователь может просматривать, выкладывать и фильтровать фотографии других пользователей по тегам. Также предоставляется возможность регистрации и входа в учетную запись.

Реализация

Библиотеки

Проект реализован с помощью библиотек:

- Flask
- Flask-Login - для аутентификации и авторизации пользователя
- Flask-WTF - для создания HTML форм
- SQLAlchemy - для работы с БД
- werkzeug - для хэширования пароля
- bootstrap - для оформления пользовательского интерфейса

Структура проекта

В проекте были реализованы следующие модули:

- main - запуск Web сервера
- db_session - функции для создания БД и выполнения запроса к ней
- models - описание моделей БД
- forms - описание HTML форм
- app - описание flask приложения и обработчиков адресов (URL)

Классы для HTML-форм

```
class RegisterForm(FlaskForm):
    """
    Форма регистрации нового пользователя
    """
    username = StringField('Имя', validators=[DataRequired()])
    login = StringField('Логин', validators=[DataRequired()])
    password = PasswordField('Пароль', validators=[DataRequired()])
    password2 = PasswordField('Пароль (подтверждение)',
validators=[DataRequired(), EqualTo('password')])
    submit = SubmitField('Зарегистрироваться')
```

```
class LoginForm(FlaskForm):
    """
    Форма логина
    """
    login = StringField('Логин', validators=[DataRequired()])
    password = PasswordField('Пароль', validators=[DataRequired()])
    remember_me = BooleanField('Запомнить меня')
    submit = SubmitField('Войти')
```

```
class NewPhotoForm(FlaskForm):
    """
    Форма новой фотографии
    """
    title = StringField('Название', validators=[DataRequired()])
    content = FileField('Путь к фотографии',
validators=[DataRequired()])
    is_private = BooleanField('Приватное')
    tags = StringField('Тэги (через пробел)')
    submit = SubmitField('Добавить')
```

Классы модели БД

```
class User(UserMixin, SqlAlchemyBase):
    """
    Пользователь приложения
    """
    __tablename__ = 'users'

    id = sqlalchemy.Column(sqlalchemy.Integer, primary_key=True,
autoincrement=True)
    username = sqlalchemy.Column(sqlalchemy.String, nullable=True)
    login = sqlalchemy.Column(sqlalchemy.String, index=True,
unique=True)
    password = sqlalchemy.Column(sqlalchemy.String, nullable=True)
    created_dt = sqlalchemy.Column(sqlalchemy.DateTime,
default=datetime.datetime.now)
    photos = orm.relationship("Photo", back_populates='user')

    def set_password(self, password):
        self.password = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password, password)
```

```
class Photo(SqlAlchemyBase):
    """
    Фотография пользователя
    """
    __tablename__ = 'photos'

    id = sqlalchemy.Column(sqlalchemy.Integer, primary_key=True,
autoincrement=True)
    content = sqlalchemy.Column(sqlalchemy.BLOB)
    title = sqlalchemy.Column(sqlalchemy.String, nullable=True)
    created_dt = sqlalchemy.Column(sqlalchemy.DateTime,
default=datetime.datetime.now)
    is_private = sqlalchemy.Column(sqlalchemy.Boolean,
default=True)
    user_id = sqlalchemy.Column(sqlalchemy.Integer,
sqlalchemy.ForeignKey("users.id"))
    user = orm.relationship('User')
    tags = orm.relationship('Tag', secondary="photos_tags",
backref="photos")
```

```
class Tag(SqlAlchemyBase):
    """
    Метка в фотографии
```

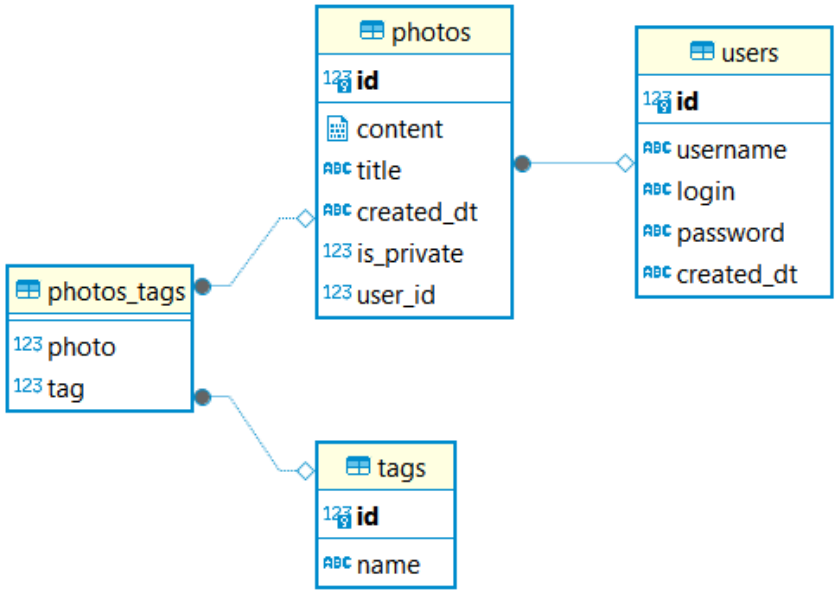
```
    """
    __tablename__ = 'tags'

    id = sqlalchemy.Column(sqlalchemy.Integer, primary_key=True,
autoincrement=True)
    name = sqlalchemy.Column(sqlalchemy.String, index=True)
```

```
"""
Связь много-ко-многим между фотографией и меткой.
"""
PhotoTags = sqlalchemy.Table(
    'photos_tags',
    SqlAlchemyBase.metadata,
    sqlalchemy.Column('photo', sqlalchemy.Integer,
sqlalchemy.ForeignKey('photos.id')),
    sqlalchemy.Column('tag', sqlalchemy.Integer,
sqlalchemy.ForeignKey('tags.id'))
)
```

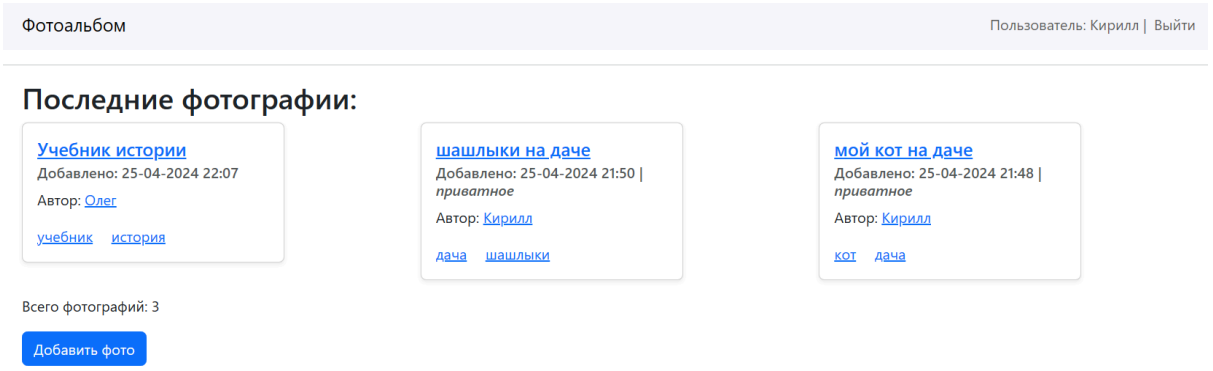
Структура БД

Реализована база данных следующей структуры:

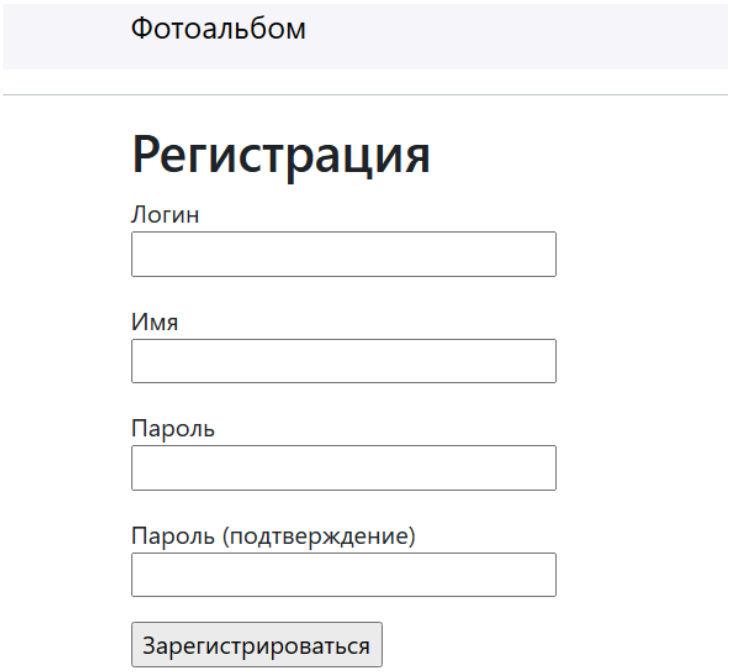


Пользовательский интерфейс

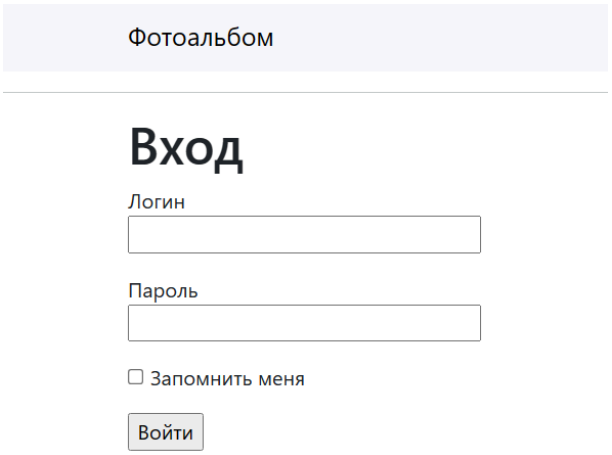
Главная страница



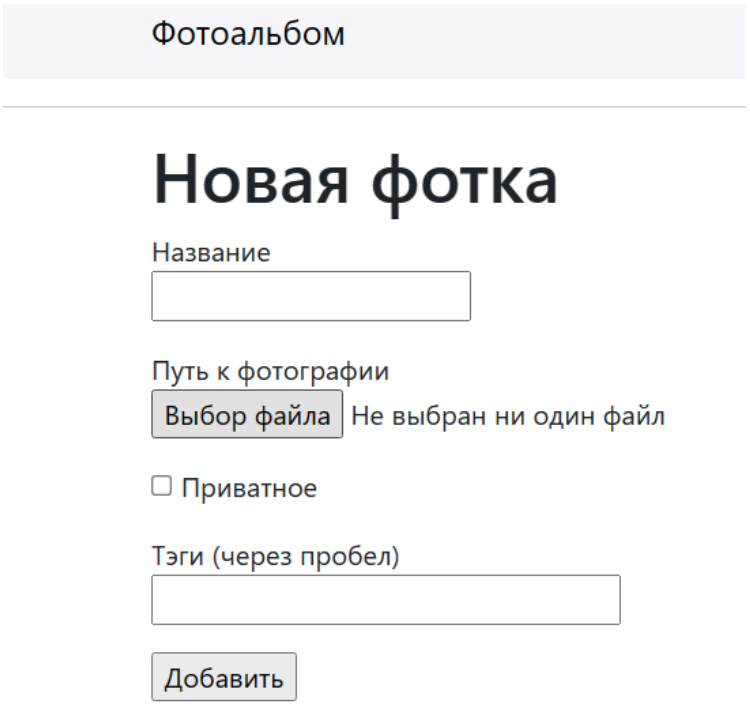
Страница регистрации пользователя



Страница входа в учетную запись



Страница добавления новых фотографий



Страница просмотра фотографии

мой кот на даче

Автор: Кирилл / 25-04-2024 21:48

Тэги: кот дача

Удалить

Страница фильтрации по тегам

Фотоальбом

Фотографии с тэгом: дача

мой кот на даче
Добавлено: 25-04-2024 21:48
Автор: Кирилл
кот дача

шашлыки на даче
Добавлено: 25-04-2024 21:50
Автор: Кирилл
дача шашлыки

Всего фотографий: 2

Установка зависимостей и запуск

Для запуска программы необходимо установить выше описанные библиотеки:

```
$ pip install -f .\requirements.txt
```

Запуск сервера выполняется следующим образом:

```
$ python main.py
Подключение к базе данных по адресу
sqlite:///photo.db?check_same_thread=False
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production
deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:8080
Press CTRL+C to quit
```