МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Донецкий национальный технический университет»

Факультет ИСП

Кафедра ПИ им Л.П.Фельдмана

Лабораторная работа № 4

на тему: «Создание самодокументирующегося кода»

по курсу: «Профессиональная практика программной инженерии»

Проверил:

асс. каф. ПИ им. Л.П.Фельдмана Филипишин Д.А.

Выполнил:

ст. гр. ПИ-21б

Голосиенко А.И.

Донецк-2024

Цель работы – научиться добавлять в программный код специальным образом оформление докблок-комментарии, для последующей автоматической генерации API reference, а также познакомиться с форматом оформления документации DocBook.

Вариант 4. Форум (с ветками-обсуждениями с обязательной реализацией групповых ролей: гость, посетитель, модератор, администратор);

1. Регистрация на GitHub (рис. 1). Аккаунт: [youngtosha](https://github.com/youngtosha). Репозиторий: [DiscussionBoard](https://github.com/youngtosha/DiscussionBoard).

from sqlalchemy import Column, Integer, String  
from app.database import Base  
  
  
class User(Base):  
 *"""  
 Модель пользователя для базы данных.  
  
 Атрибуты:  
 id (int): Уникальный идентификатор пользователя.  
 username (str): Уникальное имя пользователя.  
 email (str): Уникальный email пользователя.  
 password\_hash (str): Хеш пароля пользователя.  
 role (str): Роль пользователя (по умолчанию "visitor").  
 """* \_\_tablename\_\_ = "users"  
  
 id = Column(Integer, primary\_key=True, index=True)  
 username = Column(String, unique=True, nullable=False)  
 email = Column(String, unique=True, nullable=False)  
 password\_hash = Column(String, nullable=False)  
 role = Column(String, default="visitor") # guest, visitor, moderator, admin  
  
 def \_\_repr\_\_(self):  
 *"""Возвращает строковое представление объекта User."""* return f"<User(id={self.id}, username={self.username}, email={self.email}, role={self.role})>"  
  
 def set\_password(self, password: str):  
 *"""  
 Устанавливает хеш пароля для пользователя.  
  
 :param password: Строка с паролем пользователя.  
 """* from hashlib import sha256  
 self.password\_hash = sha256(password.encode()).hexdigest()  
  
 def check\_password(self, password: str) -> bool:  
 *"""  
 Проверяет, совпадает ли переданный пароль с сохраненным хешем.  
  
 :param password: Строка с паролем для проверки.  
 :return: True, если пароли совпадают, иначе False.  
 """* from hashlib import sha256  
 return self.password\_hash == sha256(password.encode()).hexdigest()  
  
 def is\_admin(self) -> bool:  
 *"""  
 Проверяет, является ли пользователь администратором.  
  
 :return: True, если роль пользователя - "admin".  
 """* return self.role == "admin"  
  
 def to\_dict(self) -> dict:  
 *"""  
 Преобразует объект пользователя в словарь.  
  
 :return: Словарь с данными пользователя (без пароля).  
 """* return {  
 "id": self.id,  
 "username": self.username,  
 "email": self.email,  
 "role": self.role  
 }

Рисунок 1 – Пример файл с ДокБлоками user.py

**Sphinx**: Это одна из наиболее популярных библиотек для генерации документации. Она поддерживает различные форматы вывода, включая HTML, PDF, ePub и другие. Документация Sphinx может быть написана на языке разметки reStructuredText.

Структура полученного файла представлена на рисунке 5.

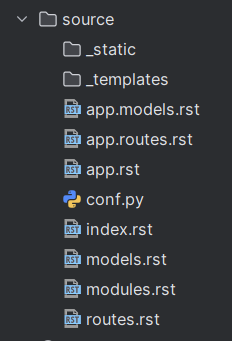


Рисунок 2 – Структура папки source

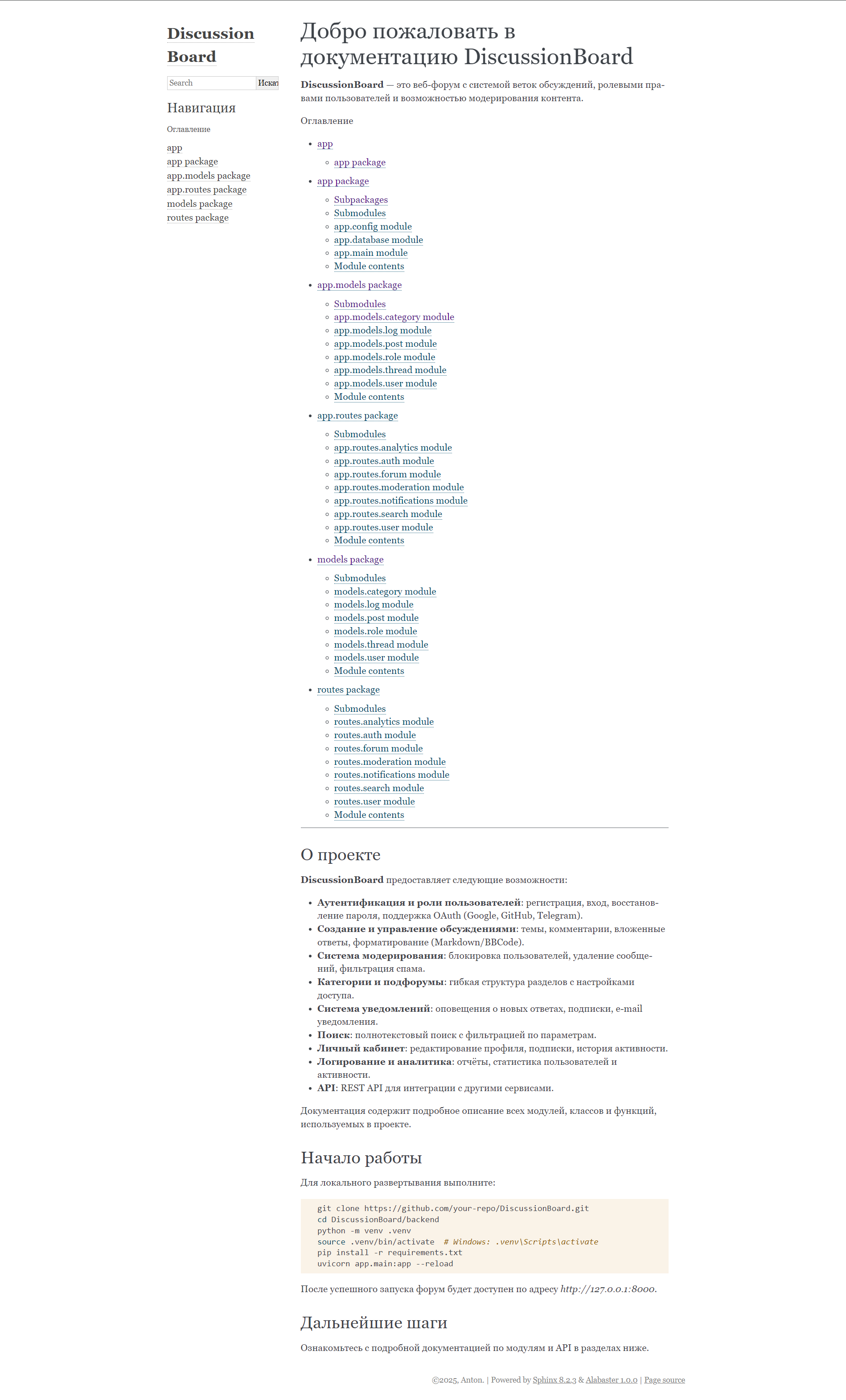


Рисунок 3 – Главная страница полученного сайта с документацией

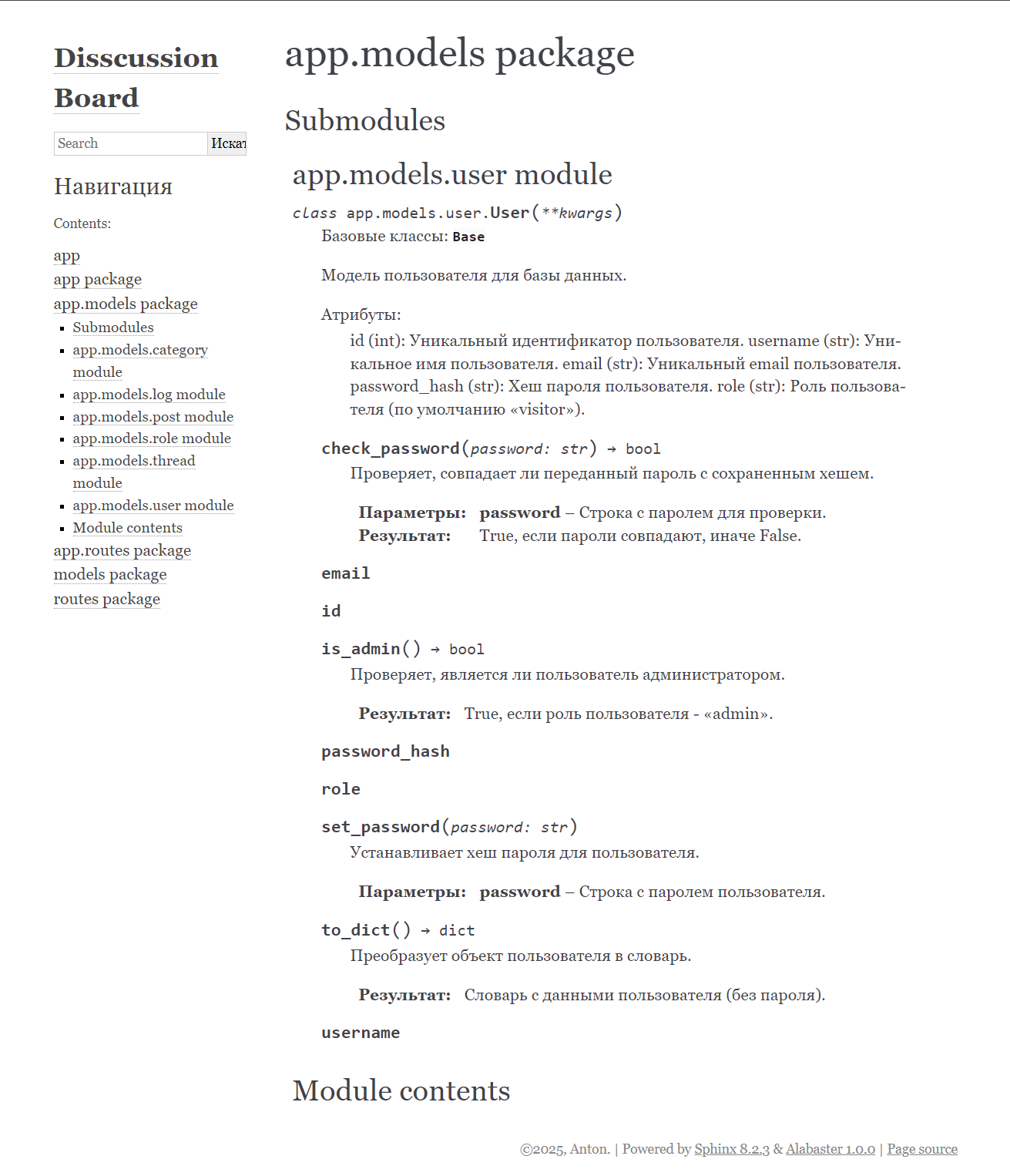


Рисунок 4 – Пример документации файла models