**Цель работы:**

Составление тестовой документации для проекта на Django с использованием React, JWT авторизацией, Mongoose и Redis.

Проект предназначен для отслеживания расписания сети кинотеатров в реальном времени.

**Описание тестируемого приложения:**

Приложение представляет собой бэкенд-сервер, разработанный на Puthon с использованием Django. В приложении реализована аутентификация и авторизация пользователей с помощью JWT токенов. Для хранения пользовательских данных используется MongoDB с помощью Mongoose. Также в приложении используется Redis для кэширования данных.

**Тестовая документация:**

**Тест-сьюты:**

**Таблица 1. Список тест-сьютов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Автор | Приоритет | Заголовок | Список тест-кейсов |
| 1 | user | 1 | Аутентификация | 1.1 Авторизация |
|  |  |  |  | 1.2 Регистрация |
| 2 | user | 1 | Сервис авторизации | 2.1 Авторизация пользователя |
|  |  |  |  | 2.2 Ошибка авторизации |
| 3 | user | 1 | Получение расписания | 3.1 Позитивный сценарий |
|  |  |  |  | 3.2 Негативный сценарий |
| 4 | user | 1 | Отображение расписания | 4.1 Позитивный сценарий |
|  |  |  |  | 4.2 Негативный сценарий |
| 5 | user | 1 | Фильтрация расписания | 5.1 Позитивный сценарий |
|  |  |  |  | 5.2 Негативный сценарий |
| 6 | user | 1 | Поиск пользователя | 6.1 Поиск по ID |
|  |  |  |  | 6.2 Поиск по имени |
|  |  |  |  | 6.3 Поиск всех пользователей |
| 7 | user | 1 | Управление пользователями | 7.1 Регистрация нового пользователя |
|  |  |  |  | 7.2 Изменение данных пользователя |
|  |  |  |  | 7.3 Удаление пользователя |

**Тест-кейсы:**

**Таблица 2. Список тест-кейсов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Описание | Предусловия | Шаги | Ожидаемый результат |
| 1.1 | Авторизация (позитивный) | Пользователь не авторизован в системе | 1. Пользователь вводит свой логин и пароль | Пользователь успешно авторизован |
| 1.2 | Регистрация (негативный) | Пользователь не зарегистрирован в системе | 1. Пользователь вводит существующий в системе email | Выводится сообщение о существующем пользователе |
| 2.1 | Авторизация пользователя (позитивный) | Пользователь зарегистрирован в системе | 1. Пользователь вводит свой логин и пароль | Пользователь успешно авторизован |
| 2.2 | Ошибка авторизации (негативынй) | Пользователь вводит неверные учетные данные | 1. Пользователь вводит неправильный логин и/или пароль | Выводится сообщение об ошибке аутентификации |
| 3.1 | Получение расписания (позитивный) | Сервер пытается связаться с базой кинотеатра и получить актуальную информацию | 1. Связь с бд установлена | Данные выведены |
| 3.2 | Получение расписания (негативный) | Сервер пытается связаться с базой кинотеатра и получить актуальную информацию | 1. Связь с бд не установлена | Возвращается сообщение об ошибке |
| 4.1 | Отображение расписания (позитивный) | Расписание существует в системе | 1. Пользователь запрашивает отображения расписания | Возвращается расписание |
| 4.2 | Отображение расписания (негативный) | Расписание не существует в системе | 1. Пользователь запрашивает отображения расписания | Возвращается сообщение об отсутствии расписания |
| 5.1 | Фильтрация расписания (позитивный) | Пользователь запрашивает фильтрацию данных | 1.Расписание существует в системе | Возвращается список коротких ссылок пользователя |
| 5.2 | Фильтрация расписания(негативный) | Пользователь запрашивает фильтрацию данных | 1. Расписание не существует в системе | Возвращается сообщение о отсутствии ссылок у пользователя |
| 6.1 | Поиск пользователя по ID | Пользователь хочет найти пользователя по ID | 1. Пользователь предоставляет ID пользователя | Возвращается пользователь с указанным ID |
| 6.2 | Поиск пользователя по имени | Пользователь хочет найти пользователя по username | 1. Пользователь предоставляет имя пользователя | Возвращается пользователь с указанным именем |
| 6.3 | Поиск всех пользователей | Пользователь хочет найти всех пользователей | 1. Пользователь запрашивает список всех пользователей | Возвращается список всех пользователей |
| 7.1 | Регистрация нового пользователя | Пользователь хочет зарегистрироваться | 1. Пользователь предоставляет данные для регистрации | Создается новый пользователь |
| 7.2 | Изменение данных пользователя | Пользователь хочет изменить данные профиля | 1. Пользователь предоставляет новые данные пользователя | Данные пользователя успешно изменены |
| 7.3 | Удаление пользователя | Пользователь хочет удалить пользователя | 1. Пользователь предоставляет ID пользователя | Пользователь успешно удален |

**Выводы по работе:**

При разработке тестовой документации были составлены подробные тест-кейсы, охватывающие различные сценарии использования функциональности. Эти кейсы были структурированы в соответствии с требованиями к документации, обеспечивая полное покрытие основных функций и возможных вариантов использования. Была проведена проверка корректности авторизации и аутентификации пользователей, управление пользователями, включая регистрацию, изменение данных и удаление. Все функции были протестированы на позитивные и негативные сценарии, чтобы обеспечить надежную работу системы и защитить ее от возможных уязвимостей.

### Использованные источники:

1. Официальная документация Django.
2. Документация по тестированию Django.
3. Материалы по web-разработке.