

Содержание тестовой документации

Содержание тестовой документации.....	1
Условия, ограничения и предположения для тестовой документации.....	2
1. Общие характеристики приложения.....	2
2. Функциональные характеристики.....	3
Регистрация.....	3
Вход.....	3
Создание задачи.....	3
Редактирование задачи.....	4
Удаление задачи.....	4
Фильтрация по статусу.....	4
3. Технические характеристики.....	4
API.....	4
Безопасность.....	5
База данных.....	5
Дополнительные ограничения.....	5
4. Ограничения и условия тестирования.....	6
5. Дополнительные замечания.....	7
Чек-лист для ручного тестирования веб-приложения «Список задач».....	7
Баг-репорты для веб-приложения «Список задач».....	20
Баг-репорт 1: Ошибка 500 при создании задачи с пустым заголовком через API..	20
Баг-репорт 2: Отображается неверный статус задачи на мобильном устройстве после редактирования.....	21
Баг-репорт 3: Некорректное сообщение об ошибке при регистрации с некорректным email.....	21
Баг-репорт 4: Некорректное отображение длинного заголовка задачи на мобильном устройстве.....	22
Тесты API для веб-приложения «Список задач».....	22
Эндпоинт: POST /login.....	22
Позитивные сценарии.....	22
Негативные сценарии.....	23
Эндпоинт: GET /tasks, POST /tasks.....	25
Позитивные сценарии.....	25
Негативные сценарии.....	26
Эндпоинт: GET /tasks/{id}, PUT /tasks/{id}, DELETE /tasks/{id}.....	28
Позитивные сценарии.....	28
Негативные сценарии.....	29
Пример запросов/ответов в Postman.....	30
Python-скрипты для автоматизации API(Примеры).....	33
JavaScript скрипты для полуавтоматизации (для упрощенной работы с коллекциями в Postman, автоматический запрос и получение JWT токена).....	35
Риски и приоритеты для веб-приложения “Список задач”	38
Таблица рисков и приоритетов для API.....	38
Таблица рисков и приоритетов для Frontend.....	41

Условия, ограничения и предположения для тестовой документации

1. Общие характеристики приложения

- **Тип приложения:** Одностраничное веб-приложение (Single Page Application, SPA), что соответствует современным трендам веб-разработки и упрощает тестирование UI/UX.
- **Технологический стек:**
 - **Frontend:** React.
 - **Backend:** Node.js с фреймворком Express.
 - **База данных:** PostgreSQL.
- **Стадия разработки:** Бета-версия (v1.0), функциональность реализована, но возможны баги, что требует тщательного тестирования.
- **Среда развертывания:** Облачная инфраструктура, обеспечивающая масштабируемость и доступность.
- **Локализация:** Только русская. Используются:
 - Кириллические символы для всех текстовых полей.
 - Формат даты: ДД.ММ.ГГГГ (например, 14.05.2025).
 - Формат времени: 24-часовой (например, 19:45).
- **Целевая аудитория:** Офисные сотрудники и фрилансер, использующие приложение для управления личными и рабочими задачами. Они ожидают:
 - Интуитивно понятный интерфейс.
 - Быструю загрузку и отклик.
 - Надежность ключевых функций (регистрация, вход, управление задачами).
- **Поддерживаемые браузеры:**
 - Google Chrome, Mozilla Firefox, Safari, Microsoft Edge.
- **Адаптивность:**
 - Полная поддержка десктопов, планшетов и мобильных устройств.
 - Breakpoints: 320px (мобильные), 768px (планшеты), 1024px (десктопы).
- **Производительность:**
 - Время загрузки страниц/компонентов: менее 2 секунд при стандартном соединении.
 - Поддержка до 1000 одновременных пользователей без значительного ухудшения производительности.
 - Обработка до 1 миллиона задач в системе.
- **Масштабируемость:**
 - Поддержка до 10 000 зарегистрированных пользователей.
- **Регуляторные требования:**
 - Соответствие GDPR для обработки персональных данных (email пользователей):
 - Хранение email только с согласия.
 - Возможность удаления аккаунта и связанных данных.
 - Защита данных при передаче (HTTPS).
- **Ограничения функциональности:**
 - Нет оффлайн-режима.
 - Нет аналитики или инструментов мониторинга.
 - Нет массовых действий.
 - Нет зависимостей между задачами или истории изменений.
 - Нет уведомлений или напоминаний о задачах.

2. Функциональные характеристики

Регистрация

- **Поля формы:**
 - **Email:** Обязательное, валидный формат (например, user@example.com).
 - **Пароль:** Обязательное, минимум 8 символов, включая буквы (A–Z, a–z), цифры (0–9) и один специальный символ (например, !@#\$%).
 - **Имя:** Необязательное, до 50 символов (кириллица, пробелы, дефисы).
- **Процесс:**
 - После заполнения формы пользователь получает письмо с уникальной ссылкой для подтверждения email.
 - После подтверждения перенаправляется на страницу входа.
- **Ограничения:**
 - Email уникален в системе.
 - Нет поддержки входа через соцсети.
- **Сообщения об ошибках:**
 - «Email уже зарегистрирован».
 - «Некорректный формат email».
 - «Слабый пароль» (не соответствует требованиям).
 - «Поле email не может быть пустым».

Вход

- **Поля формы:**
 - **Email:** Обязательное.
 - **Пароль:** Обязательное.
- **Функциональность:**
 - Успешный вход возвращает JWT-токен для API.
 - Сессия длится 24 часа (или 30 дней с опцией «Запомнить меня»).
 - Поддерживается вход с нескольких устройств без конфликтов.
- **Безопасность:**
 - После 5 неудачных попыток входа — блокировка на 15 минут.
 - Ошибка: «Неверный email или пароль».
- **Функция «Забыли пароль»:**
 - Пользователь вводит email, получает письмо со ссылкой для сброса пароля.
 - Ссылка активна 1 час.
 - Ошибка: «Email не найден», если аккаунт не существует.

Создание задачи

- **Поля:**
 - **Заголовок:** Обязательное, до 255 символов (кириллица, латиница, цифры, спецсимволы).
 - **Описание:** Необязательное, до 1000 символов.
 - **Дата выполнения:** Необязательная, не ранее текущей даты (в формате ДД.ММ.ГГГГ).
 - **Приоритет:** Необязательный, выбор из: низкий, средний, высокий (по умолчанию — средний).
- **Ограничения:**
 - Максимум 1000 задач на пользователя.
 - Валидация: заголовок не пустой, дата не в прошлом.
- **Процесс:**
 - После создания пользователь возвращается к списку задач.
 - Отображается уведомление: «Задача успешно создана».

- **Ошибки:**
 - «Заголовок не может быть пустым».
 - «Дата выполнения не может быть в прошлом».
 - «Достигнут лимит задач».

Редактирование задачи

- **Доступные поля:** Все (заголовок, описание, дата, приоритет), кроме даты создания.
- **Процесс:**
 - Пользователь выбирает задачу, редактирует поля, сохраняет.
 - Уведомление: «Задача обновлена».
- **Ограничения:** Те же, что при создании (валидация заголовка, даты).

Удаление задачи

- **Процесс:**
 - Пользователь выбирает задачу, подтверждает удаление («Удалить задачу?»).
 - Уведомление: «Задача удалена».
- **Ограничения:**
 - Удаление необратимо.
 - Только владелец задачи может удалить.

Фильтрация по статусу

- **Статусы:**
 - **Все:** Все задачи пользователя.
 - **Активные:** Невыполненные задачи (без привязки к дате).
 - **Выполненные:** Завершенные задачи.
- **Процесс:**
 - Фильтрация серверная (запрос к API).
 - При отсутствии задач отображается: «Нет задач, соответствующих фильтру».
- **Ограничения:**
 - Фильтры сбрасываются при выходе из системы или обновлении страницы.
 - Нет комбинированных фильтров или сортировки (например, по дате или приоритету).
 - Нет пользовательских фильтров.

3. Технические характеристики

API

- **Аутентификация:**
 - JWT-токены, передаются в заголовке Authorization: Bearer <token>.
 - Токен выдается при входе, срок действия — 24 часа.
- **Эндпоинты:**
 - **POST /login:**
 - Запрос: { "email": "string", "password": "string" }.
 - Ответ: { "token": "string" } (200) или ошибка (401, 400).
 - **GET /tasks:**
 - Параметры: ?status=all|active|completed (фильтрация).
 - Ответ: [{ "id": number, "title": string, "description": string, "dueDate": string, "priority": string, "status": string }] (200).

- **POST /tasks:**
 - Запрос: { "title": string, "description": string, "dueDate": string, "priority": string }.
 - Ответ: { "id": number, ... } (201) или ошибка (400).
- **GET /tasks/{id}:**
 - Ответ: { "id": number, ... } (200) или ошибка (404).
- **PUT /tasks/{id}:**
 - Запрос: { "title": string, ... }.
 - Ответ: { "id": number, ... } (200) или ошибка (400, 404).
- **DELETE /tasks/{id}:**
 - Ответ: {} (204) или ошибка (404).
- **Формат данных:** JSON.
- **Коды ответа:**
 - 200 (OK), 201 (Created), 204 (No Content).
 - 400 (Bad Request), 401 (Unauthorized), 404 (Not Found), 429 (Too Many Requests).
- **Ограничения:**
 - 100 запросов в минуту на пользователя.
 - Максимальный размер тела запроса: 10 КБ.
- **Ошибки:**
 - /login: «Неверные учетные данные» (401), «Некорректный формат данных» (400).
 - /tasks: «Недостаточно прав» (401), «Лимит задач превышен» (400).
 - /tasks/{id}: «Задача не найдена» (404).

Безопасность

- **Протокол:** HTTPS для всех запросов.
- **Защита:**
 - Валидация входных данных против SQL-инъекций, XSS, CSRF.
 - Уникальные email и ID задач в БД.
- **Тестируемые уязвимости:**
 - Неправильная авторизация в API.
 - Утечка данных (например, доступ к чужим задачам).
 - Ввод вредоносных данных (скрипты, SQL).

База данных

- **Ограничения:**
 - Уникальность email пользователей.
 - Уникальность ID задач.
 - Поля задач: id (автоинкремент), userId (связь с пользователем), title, description, dueDate, priority, status, createdAt.
- **Хранение:**
 - Данные хранятся до удаления аккаунта.
 - Логи не ведутся.

Дополнительные ограничения

- **Кэширование:** Отсутствует (все запросы напрямую к БД).
- **Офлайн-режим:** Не поддерживается.
- **Интеграции:** Только SMTP-сервер для отправки email (регистрация, сброс пароля).
- **Доступность:**
 - Минимальная поддержка: ARIA-атрибуты для фильтров, навигация с клавиатуры.
 - Не тестируется полное соответствие WCAG.

- **Часовые пояса:**
 - Даты хранятся в UTC, отображаются в локальном времени пользователя (Москва, UTC+3).
- **Конкурентность:**
 - Не тестируется, предполагается одиночное использование.

4. Ограничения и условия тестирования

- **Типы тестирования:**
 - **Функциональное:** Проверка всех функций (регистрация, вход, задачи, фильтрация).
 - **Usability:** Удобство интерфейса для целевой аудитории (интуитивность, сообщения).
 - **Безопасность:** Базовые тесты на SQL-инъекции, XSS, CSRF, утечки данных.
 - **Производительность:** Проверка времени ответа API (< 500 мс) и загрузки UI (< 2 сек).
- **Исключения из scope:**
 - Нагрузочное тестирование.
 - Тестирование сложных интеграций.
 - Миграция данных из других систем.
 - Полное тестирование доступности.
 - Тестирование на устаревших браузерах (например, IE).
- **Объем данных:**
 - Тестирование с 1–1000 задач на пользователя.
 - До 100 пользователей для симуляции.
- **Чек-лист:**
 - Формат: Список названий тест-кейсов.
 - Объем: 70 кейсов (35 позитивных, 35 негативных).
 - Покрытие: Регистрация, вход, создание/редактирование/удаление задач, фильтрация, безопасность.
 - Включает граничные случаи (длинные строки, некорректные email, даты).
- **Баг-репорты:**
 - Количество: 4.
 - Поля: Заголовок, описание, шаги воспроизведения, фактический результат, ожидаемый результат, критичность (критический/высокий/средний/низкий).
- **Тестирование API:**
 - Инструменты: Postman для ручного тестирования, примеры кода на Python (requests) для автоматизации, JavaScript (скрипт в Postman) для полуавтоматизации.
 - Эндпоинты: /login, /tasks, /tasks/{id}.
 - Тесты:
 - Позитивные: Валидные запросы, корректные ответы.
 - Негативные: Неверные данные, отсутствующие ресурсы, ограничения (429).
 - Безопасность: Неправильные токены, доступ к чужим данным.
 - Производительность: Время ответа < 500 мс.
- **Оценка рисков и приоритетов:**
 - Критерии: Влияние на пользователей, частота использования, сложность исправления.
 - Высокий приоритет: Регистрация, вход, создание/удаление задач (критичны для UX).
 - Средний приоритет: Редактирование, фильтрация.
 - Низкий приоритет: UI-оформление, редкие граничные случаи.
 - Формат: Таблица (функция, риск, приоритет, обоснование).

- Включает план смягчения (регрессионное тестирование, мониторинг).

5. Дополнительные замечания

- **Предполагаемые баги:**
 - Функциональные: Ошибки валидации (пустые поля, некорректные даты).
 - UI: Некорректное отображение на мобильных устройствах.
 - API: Ошибки авторизации (401), неверные ответы (500).
 - Безопасность: Возможность доступа к чужим задачам.
- **Приоритет тестирования:**
 - Критические пути: Регистрация → вход → создание/удаление задач.
 - Второстепенные: Фильтрация, редактирование.
- **Объем тестирования:**
 - Учитывается ограниченное время (2 дня), поэтому фокус на ключевых функциях и типичных багах.
- **Допущения по UX:**
 - Пользователи ожидают быструю реакцию (например, создание задачи < 1 сек).
 - Ошибки отображаются на русском языке, четко и понятно.
- **Расшифровка ID:**
 - TT_FS_TL_TS001_CL001_PS/NS
 - TT - Test Task
 - FS - FoodSoul
 - TL - Task List
 - TS001 - Test Suit 001
 - CL001 - Check List 001
 - PS/NS - Positive/Negative Scenario
 - TT_FS_TL_F/B_BR001
 - TT - Test Task
 - FS - FoodSoul
 - TL - Task List
 - F/B - Frontend/Backend
 - FaB - Frontend and Backend
 - BR001 - Bug Report 001
 - TT_FS_TL_TS002_API_P/I_PS/NS
 - TT - Test Task
 - FS - FoodSoul
 - TL - Task List
 - TS002 - Test Suit 002
 - P/I - POST /login
 - G/t, P/t - GET /tasks, POST /tasks
 - G/t/i, P/t/i - GET /tasks/{id}, PUT /tasks/{id}
 - D/t/i - DELETE /tasks/{id}
 - CL001 - Check List 001
 - PS/NS - Positive/Negative Scenario

Чек-лист для ручного тестирования веб-приложения «Список задач»

Позитивные проверки

1. TT_FS_TL_TS001_CL001_PS. Регистрация нового пользователя в форме регистрации при корректных данных

Вводимые данные:

Email: test@example.com
Пароль: Pass123!
Имя: Иван

Ожидаемый результат: Пользователь получает письмо с ссылкой для подтверждения. После клика перенаправляется на страницу входа.

2. TT_FS_TL_TS001_CL002_PS. Подтверждение email в письме после регистрации при клике по ссылке

Ожидаемый результат: Пользователь перенаправляется на страницу входа, отображается сообщение: «Email подтвержден».

3. TT_FS_TL_TS001_CL003_PS. Вход пользователя в форме входа при валидных данных

Вводимые данные:

Email: test@example.com
Пароль: Pass123!

Ожидаемый результат: Пользователь авторизуется, перенаправляется на страницу со списком задач.

4. TT_FS_TL_TS001_CL004_PS. Вход с опцией «Запомнить меня» в форме входа при авторизации

Вводимые данные:

Email: test@example.com
Пароль: Pass123!
Чекбокс «Запомнить меня»: включен

Ожидаемый результат: Пользователь авторизуется, сессия сохраняется 30 дней. При повторном входе не требуется авторизация.

5. TT_FS_TL_TS001_CL005_PS. Сброс пароля в форме «Забыли пароль» при существующем email

Вводимые данные:

Email: test@example.com

Ожидаемый результат: Пользователь получает письмо со ссылкой для сброса пароля, активной 1 час.

6. TT_FS_TL_TS001_CL006_PS. Установка нового пароля в форме сброса пароля при клике по ссылке

Вводимые данные:

Новый пароль: NewPass123!

Ожидаемый результат: Пароль обновлен, пользователь перенаправляется на страницу входа.

7. TT_FS_TL_TS001_CL007_PS. Создание задачи в форме создания задачи при заполнении всех полей

Вводимые данные:

Заголовок: Купить продукты
Описание: Молоко, хлеб, яйца
Дата: 15.05.2025
Приоритет: Высокий

Ожидаемый результат: Задача создана, пользователь возвращается к списку задач, отображается уведомление: «Задача успешно создана».

8. TT_FS_TL_TS001_CL008_PS. Создание задачи в форме создания задачи при минимальном заполнении

Вводимые данные:

Заголовок: Позвонить клиенту

Ожидаемый результат: Задача создана, приоритет по умолчанию — средний, пользователь видит задачу в списке.

9. TT_FS_TL_TS001_CL009_PS. Редактирование задачи в форме редактирования при изменении всех полей

Вводимые данные:

Заголовок: Позвонить менеджеру
Описание: Обсудить проект
Дата: 16.05.2025
Приоритет: Низкий

Ожидаемый результат: Задача обновлена, отображается уведомление: «Задача обновлена», новые данные видны в списке.

10. TT_FS_TL_TS001_CL010_PS. Пометка задачи как выполненной в списке задач при клике на чекбокс

Ожидаемый результат: Задача перемещается в статус «Выполненные», отображается в соответствующем фильтре.

11. TT_FS_TL_TS001_CL011_PS. Удаление задачи в списке задач при подтверждении

Вводимые данные:

Кнопка «Удалить» → Подтверждение «Удалить задачу?»

Ожидаемый результат: Задача удалена, отображается уведомление: «Задача удалена», задача исчезает из списка.

12. TT_FS_TL_TS001_CL012_PS. Фильтрация задач в списке задач при выборе «Все»

Ожидаемый результат: Отображаются все задачи пользователя (активные и выполненные).

13. TT_FS_TL_TS001_CL013_PS. Фильтрация задач в списке задач при выборе «Активные»

Ожидаемый результат: Отображаются только невыполненные задачи.

14. TT_FS_TL_TS001_CL014_PS. Фильтрация задач в списке задач при выборе «Выполненные»

Ожидаемый результат: Отображаются только выполненные задачи.

15. TT_FS_TL_TS001_CL015_PS. Отображение списка задач на десктопе при разрешении 1920x1080

Ожидаемый результат: Список задач отображается корректно, все элементы (заголовок, фильтры, кнопки) видимы и доступны.

16. TT_FS_TL_TS001_CL016_PS. Отображение списка задач на мобильном устройстве при разрешении 320x568

Ожидаемый результат: Интерфейс адаптируется, элементы не перекрываются, прокрутка работает, фильтры доступны.

17. TT_FS_TL_TS001_CL017_PS. Создание задачи в форме создания задачи на планшете при разрешении 768x1024

Вводимые данные:

Заголовок: Подготовить отчет

Ожидаемый результат: Задача создана, интерфейс адаптивный, форма отображается корректно.

18. TT_FS_TL_TS001_CL018_PS. Навигация по интерфейсу с клавиатуры на странице задач при использовании Tab

Ожидаемый результат: Все элементы (фильтры, кнопки, чекбоксы) доступны, фокус перемещается логично.

19. TT_FS_TL_TS001_CL019_PS. Отображение сообщения при пустом списке задач на странице задач

Ожидаемый результат: Отображается текст: «Нет задач, соответствующих фильтру» или «Создайте первую задачу».

20. TT_FS_TL_TS001_CL020_PS. Обновление страницы задач на странице задач при нажатии F5

Ожидаемый результат: Список задач и фильтры сохраняются, пользователь остается авторизованным.

21. TT_FS_TL_TS001_CL021_PS. Создание задачи с максимальной длиной заголовка в форме создания задачи

Вводимые данные:

Заголовок: *строка из 255 кириллических символов*

Ожидаемый результат: Задача создана, заголовок полностью отображается в списке.

22. TT_FS_TL_TS001_CL022_PS. Создание задачи с максимальной длиной описания в форме создания задачи

Вводимые данные:

Описание: *строка из 1000 символов*

Ожидаемый результат: Задача создана, описание полностью сохраняется и отображается.

23. TT_FS_TL_TS001_CL023_PS. Редактирование задачи с изменением только приоритета в форме редактирования

Вводимые данные:

Приоритет: Высокий

Ожидаемый результат: Задача обновлена, новый приоритет отображается в списке.

24. TT_FS_TL_TS001_CL024_PS. Фильтрация задач при отсутствии выполненных задач в списке задач

Вводимые данные:

Фильтр: «Выполненные»

Ожидаемый результат: Отображается сообщение: «Нет задач, соответствующих фильтру».

25. TT_FS_TL_TS001_CL025_PS. Создание задачи с текущей датой в форме создания задачи

Вводимые данные:

Дата: 14.05.2025

Ожидаемый результат: Задача создана, дата корректно отображается в формате ДД.ММ.ГГГГ.

26. TT_FS_TL_TS001_CL026_PS. Регистрация пользователя в форме регистрации с кириллическим именем

Вводимые данные:

Email: new@example.com

Пароль: Pass123!

Имя: Екатерина

Ожидаемый результат: Пользователь зарегистрирован, имя корректно отображается в профиле.

27. TT_FS_TL_TS001_CL027_PS. Создание задачи с кириллическим описанием в форме создания задачи

Вводимые данные:

Заголовок: Встреча

Описание: Обсудить проект на русском языке

Ожидаемый результат: Задача создана, описание полностью отображается в списке.

28. TT_FS_TL_TS001_CL028_PS. Редактирование задачи с удалением описания в форме редактирования

Вводимые данные:

Описание: пусто

Ожидаемый результат: Задача обновлена, описание удалено, остальные данные сохранены.

29. TT_FS_TL_TS001_CL029_PS. Создание задачи без даты выполнения в форме создания задачи

Вводимые данные:

Заголовок: Написать письмо

Ожидаемый результат: Задача создана, поле даты пустое, отображается корректно.

30. TT_FS_TL_TS001_CL030_PS. Создание задачи с низким приоритетом в форме создания задачи

Вводимые данные:

Заголовок: Проверить почту

Приоритет: Низкий

Ожидаемый результат: Задача создана, приоритет отображается как «Низкий».

31. TT_FS_TL_TS001_CL031_PS. Фильтрация задач при большом списке в списке задач

Вводимые данные:

Фильтр: «Активные» (100 задач)

Ожидаемый результат: Отображаются только активные задачи, список загружается без задержек.

32. TT_FS_TL_TS001_CL032_PS. Вход пользователя в форме входа на мобильном устройстве

Вводимые данные:

Email: test@example.com

Пароль: Pass123!

Ожидаемый результат: Пользователь авторизуется, интерфейс адаптивный, список задач отображается.

33. TT_FS_TL_TS001_CL033_PS. Регистрация пользователя с максимальной длиной имени в форме регистрации

Вводимые данные:

Имя: строка из 50 кириллических символов

Ожидаемый результат: Пользователь зарегистрирован, имя полностью сохраняется и отображается.

34. TT_FS_TL_TS001_CL034_PS. Пометка задачи как активной в списке задач при снятии чекбокса

Ожидаемый результат: Задача возвращается в статус «Активные», отображается в соответствующем фильтре.

35. TT_FS_TL_TS001_CL035_PS. Удаление задачи на мобильном устройстве в списке задач

Вводимые данные:

Кнопка «Удалить» → Подтверждение

Ожидаемый результат: Задача удалена, интерфейс адаптивный, уведомление отображается.

Негативные проверки

1. TT_FS_TL_TS001_CL001_NS. Регистрация пользователя в форме регистрации при существующем email

Вводимые данные:

Email: test@example.com

Пароль: Pass123!

Ожидаемый результат: Ошибка: «Email уже зарегистрирован», регистрация не выполняется.

2. TT_FS_TL_TS001_CL002_NS. Регистрация пользователя в форме регистрации при пустом email

Вводимые данные:

Email: *пусто*

Пароль: Pass123!

Ожидаемый результат: Ошибка: «Поле email не может быть пустым», регистрация не выполняется.

3. TT_FS_TL_TS001_CL003_NS. Регистрация пользователя в форме регистрации при некорректном email

Вводимые данные:

Email: test@

Пароль: Pass123!

Ожидаемый результат: Ошибка: «Некорректный формат email», регистрация не выполняется.

4. TT_FS_TL_TS001_CL004_NS. Регистрация пользователя в форме регистрации при слабом пароле

Вводимые данные:

Email: new@example.com

Пароль: pass

Ожидаемый результат: Ошибка: «Слабый пароль», регистрация не выполняется.

5. TT_FS_TL_TS001_CL005_NS. Регистрация пользователя в форме регистрации при пустом пароле

Вводимые данные:

Email: new@example.com
Пароль: *пусто*

Ожидаемый результат: Ошибка: «Поле пароль не может быть пустым», регистрация не выполняется.

6. TT_FS_TL_TS001_CL006_NS. Вход пользователя в форме входа при неверном пароле

Вводимые данные:

Email: test@example.com
Пароль: Wrong123!

Ожидаемый результат: Ошибка: «Неверный email или пароль», вход не выполняется.

7. TT_FS_TL_TS001_CL007_NS. Вход пользователя в форме входа при несуществующем email

Вводимые данные:

Email: unknown@example.com
Пароль: Pass123!

Ожидаемый результат: Ошибка: «Неверный email или пароль», вход не выполняется.

8. TT_FS_TL_TS001_CL008_NS. Вход пользователя в форме входа при пустом email

Вводимые данные:

Email: пусто
Пароль: Pass123!

Ожидаемый результат: Ошибка: «Поле email не может быть пустым», вход не выполняется.

9. TT_FS_TL_TS001_CL009_NS. Вход пользователя в форме входа после 5 неверных попыток

Вводимые данные:

Email: test@example.com
Пароль: Wrong123! (5 раз)

Ожидаемый результат: Ошибка: «Слишком много попыток, попробуйте через 15 минут».

10. TT_FS_TL_TS001_CL010_NS. Сброс пароля в форме «Забыли пароль» при несуществующем email

Вводимые данные:

Email: unknown@example.com

Ожидаемый результат: Ошибка: «Email не найден», письмо не отправляется.

11. TT_FS_TL_TS001_CL011_NS. Создание задачи в форме создания задачи при пустом заголовке

Вводимые данные:

Заголовок: пусто

Ожидаемый результат: Ошибка: «Заголовок не может быть пустым», задача не создается.

12. TT_FS_TL_TS001_CL012_NS. Создание задачи в форме создания задачи при дате в прошлом

Вводимые данные:

Дата: 13.05.2025

Ожидаемый результат: Ошибка: «Дата выполнения не может быть в прошлом», задача не создается.

13. TT_FS_TL_TS001_CL013_NS. Создание задачи в форме создания задачи при превышении лимита

Вводимые данные:

1001 задача

Ожидаемый результат: Ошибка: «Достигнут лимит задач», задача не создается.

14. TT_FS_TL_TS001_CL014_NS. Создание задачи в форме создания задачи с заголовком > 255 символов

Вводимые данные:

Заголовок: *строка 256 символов*

Ожидаемый результат: Ошибка: «Заголовок слишком длинный», задача не создается.

15. TT_FS_TL_TS001_CL015_NS. Создание задачи в форме создания задачи с описанием > 1000 символов

Вводимые данные:

Описание: *строка 1001 символ*

Ожидаемый результат: Ошибка: «Описание слишком длинное», задача не создается.

16. TT_FS_TL_TS001_CL016_NS. Редактирование задачи в форме редактирования при пустом заголовке

Вводимые данные:

Заголовок: *пусто*

Ожидаемый результат: Ошибка: «Заголовок не может быть пустым», задача не обновляется.

17. TT_FS_TL_TS001_CL017_NS. Удаление задачи в списке задач при отмене подтверждения

Вводимые данные:

Кнопка «Удалить» → Отмена

Ожидаемый результат: Задача не удаляется, список остается без изменений.

18. TT_FS_TL_TS001_CL018_NS. Ввод SQL-инъекции в поле email в форме регистрации

Вводимые данные:

Email: test' OR 1=1; --

Ожидаемый результат: Ошибка: «Некорректный формат email», регистрация не выполняется, БД не затронута.

19. TT_FS_TL_TS001_CL019_NS. Ввод XSS-скрипта в поле заголовка в форме создания задачи

Вводимые данные:

Заголовок: <script>alert('test')</script>

Ожидаемый результат: Ошибка или экранирование ввода, скрипт не выполняется, задача не создается или сохраняется безопасно.

20. TT_FS_TL_TS001_CL020_NS. Ввод некорректной даты в форме создания задачи

Вводимые данные:

Дата: 32.05.2025

Ожидаемый результат: Ошибка: «Некорректный формат даты», задача не создается.

21. TT_FS_TL_TS001_CL021_NS. Ввод некорректных символов в поле имени в форме регистрации

Вводимые данные:

Имя: #\$\$%^

Ожидаемый результат: Ошибка: «Имя содержит недопустимые символы», регистрация не выполняется.

22. TT_FS_TL_TS001_CL022_NS. Регистрация при очень длинном имени в форме регистрации

Вводимые данные:

Имя: *строка 51 символ*

Ожидаемый результат: Ошибка: «Имя слишком длинное», регистрация не выполняется.

23. TT_FS_TL_TS001_CL023_NS. Создание задачи при некорректном формате даты в форме создания задачи

Вводимые данные:

Дата: 2025-05-15

Ожидаемый результат: Ошибка: «Некорректный формат даты», задача не создается.

24. TT_FS_TL_TS001_CL024_NS. Фильтрация задач в списке задач при отсутствии задач

Вводимые данные:

Фильтр: «Активные»

Ожидаемый результат: Отображается: «Нет задач, соответствующих фильтру».

25. TT_FS_TL_TS001_CL025_NS. Вход в форме входа при отключенном JavaScript

Ожидаемый результат: Интерфейс отображается, но вход невозможен, отображается сообщение: «Включите JavaScript».

26. TT_FS_TL_TS001_CL026_NS. Ввод пробелов в поле email в форме регистрации

Вводимые данные:

Email: test@example.com

Ожидаемый результат: Ошибка: «Некорректный формат email» или пробелы обрезаются, регистрация выполняется.

27. TT_FS_TL_TS001_CL027_NS. Создание задачи с некорректным приоритетом в форме создания задачи

Вводимые данные:

Приоритет: Сверхвысокий

Ожидаемый результат: Ошибка: «Недопустимое значение приоритета», задача не создается.

28. TT_FS_TL_TS001_CL028_NS. Редактирование задачи с датой в прошлом в форме редактирования

Вводимые данные:

Дата: 13.05.2025

Ожидаемый результат: Ошибка: «Дата выполнения не может быть в прошлом», задача не обновляется.

29. TT_FS_TL_TS001_CL029_NS. Ввод HTML-тегов в поле описания в форме создания задачи

Вводимые данные:

Описание: Текст

Ожидаемый результат: Теги экранируются или отклоняются, задача создается безопасно или не создается.

30. TT_FS_TL_TS001_CL030_NS. Регистрация с email максимальной длины в форме регистрации

Вводимые данные:

Email: *строка 254 символов*

Ожидаемый результат: Ошибка: «Email слишком длинный» или регистрация выполняется, если длина допустима.

31. TT_FS_TL_TS001_CL031_NS. Вход с пустым паролем в форме входа

Вводимые данные:

Email: test@example.com

Пароль: *пусто*

Ожидаемый результат: Ошибка: «Поле пароль не может быть пустым», вход не выполняется.

32. TT_FS_TL_TS001_CL032_NS. Создание задачи с некорректными символами в заголовке в форме создания задачи

Вводимые данные:

Заголовок: Задача 😊

Ожидаемый результат: Ошибка: «Заголовок содержит недопустимые символы» или символы экранируются.

33. TT_FS_TL_TS001_CL033_NS. Сброс пароля с некорректным форматом пароля в форме сброса

Вводимые данные:

Новый пароль: pass

Ожидаемый результат: Ошибка: «Слабый пароль», пароль не обновляется.

34. TT_FS_TL_TS001_CL034_NS. Ввод длинного email в форме «Забыли пароль»

Вводимые данные:

Email: строка 255 символов

Ожидаемый результат: Ошибка: «Некорректный формат email» или «Email слишком длинный».

35. TT_FS_TL_TS001_CL035_NS. Фильтрация задач при некорректном значении фильтра в списке задач

Вводимые данные:

Фильтр: Неверный (если UI позволяет)

Ожидаемый результат: Ошибка или возврат к дефолтному фильтру («Все»), список отображается корректно.

Баг-репорты для веб-приложения «Список задач»

Баг-репорт 1: Ошибка 500 при создании задачи с пустым заголовком через API

- **ID:** TT_FS_TL_B_BR001
- **Заголовок:** Ошибка 500 при отправке POST-запроса на /tasks с пустым заголовком.
- **Описание:** При отправке POST-запроса на endpoint /tasks с пустым заголовком (title: "") сервер возвращает ошибку 500 вместо ожидаемой ошибки валидации (400). Это указывает на необработанное исключение на стороне сервера, что может привести к нестабильности API и блокировке создания задач через API.
- **Шаги воспроизведения:**
 1. Авторизуйтесь через POST /login с валидными данными (email: test@example.com, password: Pass123!), получите JWT-токен.
 2. Отправьте POST-запрос на /tasks с телом { "title": "", "description": "Тест", "dueDate": "15.05.2025", "priority": "medium" }, используя заголовок Authorization: Bearer <token>.
 3. Проверьте ответ сервера.
- **Фактический результат:** Сервер возвращает ошибку 500 Internal Server Error с сообщением: «Internal server error».
- **Ожидаемый результат:** Сервер возвращает ошибку 400 Bad Request с сообщением: «Заголовок не может быть пустым».
- **Критичность:** **Критический.** Баг блокирует корректное создание задач через API, что является ключевой функциональностью для интеграций и автоматизации. Затрагивает всех пользователей API.

Баг-репорт 2: Отображается неверный статус задачи на мобильном устройстве после редактирования

- **ID:** TT_FS_TL_F_BR002
- **Заголовок:** Задача отображается как «Активная» после пометки как «Выполненная» на мобильном устройстве.
- **Описание:** При пометке задачи как выполненной на мобильном устройстве (разрешение 320x568) через чекбокс статус задачи не обновляется в UI, хотя запрос к серверу успешен. Это сбивает пользователей, так как задача остается в фильтре «Активные», несмотря на завершение.
- **Шаги воспроизведения:**
 1. Откройте приложение в Chrome на мобильном устройстве (разрешение 320x568).
 2. Авторизуйтесь (email: test@example.com, password: Pass123!).
 3. Перейдите к списку задач, выберите фильтр «Активные».
 4. Создайте задачу (заголовок: «Тестовая задача»).
 5. Поставьте чекбокс «Выполнено» для задачи.
 6. Обновите страницу или переключите фильтр на «Выполненные».
- **Фактический результат:** Задача остается в фильтре «Активные», чекбокс не отмечен. При переключении на «Выполненные» задача отсутствует.
- **Ожидаемый результат:** Задача перемещается в фильтр «Выполненные», чекбокс отмечен, и задача отображается в правильном статусе.
- **Критичность:** **Высокий.** Баг нарушает ключевую функциональность (управление задачами) на мобильных устройствах, затрагивая значительную часть пользователей (целевая аудитория — офисные сотрудники и фрилансеры, использующие мобильные). Обходной путь — обновление через десктоп.

Баг-репорт 3: Некорректное сообщение об ошибке при регистрации с некорректным email

- **ID:** TT_FS_TL_FaB_BR003
- **Заголовок:** Непонятное сообщение об ошибке при регистрации с некорректным email.
- **Описание:** При попытке регистрации с некорректным email (например, test@) форма отображает сообщение: «Ошибка сервера», вместо ожидаемого «Некорректный формат email». Это сбивает пользователей и не указывает на конкретную проблему, снижая UX.
- **Шаги воспроизведения:**
 1. Откройте приложение в Chrome на десктопе (1920x1080).
 2. Перейдите к форме регистрации.
 3. Введите email: test@, пароль: Pass123!, имя: Иван.
 4. Нажмите кнопку «Зарегистрироваться».
 5. Проверьте отображаемое сообщение.
- **Фактический результат:** Отображается сообщение: «Ошибка сервера».
- **Ожидаемый результат:** Отображается сообщение: «Некорректный формат email».
- **Критичность: Средний.** Баг не блокирует регистрацию (валидный email работает), но ухудшает UX, сбивая пользователей. Затрагивает новых пользователей, но есть обходной путь (ввести корректный email).

Баг-репорт 4: Некорректное отображение длинного заголовка задачи на мобильном устройстве

- **ID:** TT_FS_TL_F_BR004
- **Заголовок:** Длинный заголовок задачи обрезается без прокрутки на мобильном устройстве.
- **Описание:** При создании задачи с заголовком максимальной длины (255 символов) на мобильном устройстве (320x568) заголовок обрезается в списке задач, и нет возможности прокрутки или просмотра полного текста. Это снижает удобство чтения задач с длинными заголовками.
- **Шаги воспроизведения:**
 1. Откройте приложение в Chrome на мобильном устройстве (320x568).
 2. Авторизуйтесь (email: test@example.com, password: Pass123!).
 3. Создайте задачу с заголовком из 255 кириллических символов.
 4. Перейдите к списку задач, выберите фильтр «Активные».
 5. Проверьте отображение заголовка задачи.
- **Фактический результат:** Заголовок обрезается, нет прокрутки или всплывающей подсказки.
- **Ожидаемый результат:** Заголовок либо полностью отображается (с переносом строк), либо доступна прокрутка/всплывающая подсказка для просмотра полного текста.
- **Критичность: Низкий.** Баг косметический, затрагивает только задачи с длинными заголовками на мобильных, не блокирует функциональность. Пользователи могут редактировать задачу, чтобы увидеть полный заголовок.

Тесты API для веб-приложения «Список задач»

Эндпоинт: POST /login

Цель: Проверка аутентификации пользователей.

Позитивные сценарии

1. TT_FS_TL_TS002_API_P/I_CL001_PS. Успешный вход с валидными данными

Запрос:

```
{ "email": "test@example.com", "password": "Pass123!" }
```

Ожидаемый результат: Код 200, ответ: { "token": "<JWT>" }, время ответа < 500 мс.

2. TT_FS_TL_TS002_API_P/I_CL002_PS. Вход с минимальным паролем (8 символов)

Запрос:

```
{ "email": "test@example.com", "password": "Pass123!" }
```

Ожидаемый результат: Код 200, ответ: { "token": "<JWT>" }.

3. TT_FS_TL_TS002_API_P/I_CL003_PS. Вход с длинным email (254 символа)

Запрос:

```
{ "email": "*<254 символа, валидный>*", "password": "Pass123!" }
```

Ожидаемый результат: Код 200, ответ: { "token": "<JWT>" }.

4. TT_FS_TL_TS002_API_P/I_CL004_PS. Вход с пробелами в email (игнорируются/обрезаются)

Запрос:

```
{ "email": " test@example.com ", "password": "Pass123!" }
```

Ожидаемый результат: Код 200, ответ: { "token": "<JWT>" }.

5. TT_FS_TL_TS002_API_P/I_CL005_PS. Проверка времени ответа при валидном входе

Запрос:

```
{ "email": "test@example.com", "password": "Pass123!" }
```

Ожидаемый результат: Код 200, время ответа < 500 мс.

6. TT_FS_TL_TS002_API_P/I_CL006_PS. Вход с кириллическим email (если поддерживается)

Запрос:

```
{ "email": "тест@пример.рф", "password": "Pass123!" }
```

Ожидаемый результат: Код 200, ответ: { "token": "<JWT>" }.

7. TT_FS_TL_TS002_API_P/I_CL007_PS. Вход с максимальным размером пароля

Запрос:

```
{ "email": "test@example.com", "password": "*<100 символов>*" }
```

Ожидаемый результат: Код 200, ответ: { "token": "<JWT>" }.

Негативные сценарии

1. TT_FS_TL_TS002_API_P/I_CL001_NS. Вход с некорректным паролем

Запрос:

```
{ "email": "test@example.com", "password": "Wrong123!" }
```

Ожидаемый результат: Код 401, сообщение: «Неверный email или пароль».

2. TT_FS_TL_TS002_API_P/I_CL002_NS. Вход с несуществующим email

Запрос:

```
{ "email": "unknown@example.com", "password": "Pass123!" }
```

Ожидаемый результат: Код 401, сообщение: «Неверный email или пароль».

3. TT_FS_TL_TS002_API_P/I_CL003_NS. Вход с пустым email

Запрос:

```
{ "email": "", "password": "Pass123!" }
```

Ожидаемый результат: Код 400, сообщение: «Поле email не может быть пустым».

4. TT_FS_TL_TS002_API_P/I_CL004_NS. Вход с пустым паролем

Запрос:

```
{ "email": "test@example.com", "password": "" }
```

Ожидаемый результат: Код 400, сообщение: «Поле пароль не может быть пустым».

5. TT_FS_TL_TS002_API_P/I_CL005_NS. Вход с некорректным форматом email

Запрос:

```
{ "email": "test@", "password": "Pass123!" }
```

Ожидаемый результат: Код 400, сообщение: «Некорректный формат email».

6. TT_FS_TL_TS002_API_P/I_CL006_NS. Вход после 5 неверных попыток

Запрос:

5 запросов: { "email": "test@example.com", "password": "Wrong123!" }

Ожидаемый результат: Код 429, сообщение: «Слишком много попыток, попробуйте через 15 минут».

7. TT_FS_TL_TS002_API_P/I_CL007_NS. Вход с SQL-инъекцией в email

Запрос:

```
{ "email": "test' OR 1=1; --", "password": "Pass123!" }
```

Ожидаемый результат: Код 400, сообщение: «Некорректный формат email», БД не затронута.

8. TT_FS_TL_TS002_API_P/I_CL008_NS. Вход с XSS в пароле

Запрос:

```
{ "email": "test@example.com", "password": "<script>alert('test')</script>" }
```

Ожидаемый результат: Код 401, сообщение: «Неверный email или пароль», скрипт не выполняется.

Эндпоинт: GET /tasks, POST /tasks

Цель: Проверка получения и создания задач.

Позитивные сценарии

1. TT_FS_TL_TS002_API_G/t_CL001_PS. Получение всех задач с валидным токеном

Запрос:

GET /tasks?status=all, Authorization: Bearer <token>

Ожидаемый результат: Код 200, ответ: массив задач [{ "id": 1, "title": "...", ... }], время < 500 мс.

2. TT_FS_TL_TS002_API_G/t_CL002_PS. Получение активных задач

Запрос:

GET /tasks?status=active, Authorization: Bearer <token>

Ожидаемый результат: Код 200, ответ: массив только активных задач.

3. TT_FS_TL_TS002_API_G/t_CL003_PS. Получение выполненных задач

Запрос:

GET /tasks?status=completed, Authorization: Bearer <token>

Ожидаемый результат: Код 200, ответ: массив только выполненных задач.

4. TT_FS_TL_TS002_API_G/t_CL004_PS. Получение задач при пустом списке

Запрос:

GET /tasks?status=all, Authorization: Bearer <token>

Ожидаемый результат: Код 200, ответ: [].

5. TT_FS_TL_TS002_API_P/t_CL005_PS. Создание задачи с валидными данными

Запрос:

POST /tasks, { "title": "Тест", "description": "Описание", "dueDate": "15.05.2025", "priority": "high" }, Authorization: Bearer <token>

Ожидаемый результат: Код 201, ответ: { "id": 1, "title": "Тест", ... }, время < 500 мс.

6. TT_FS_TL_TS002_API_P/t_CL006_PS. Создание задачи с минимальными данными

Запрос:

POST /tasks, { "title": "Тест" }, Authorization: Bearer <token>

Ожидаемый результат: Код 201, ответ: { "id": 1, "title": "Тест", "priority": "medium", ... }.

7. TT_FS_TL_TS002_API_P/t_CL007_PS. Проверка времени ответа при создании задачи

Запрос:

POST /tasks, { "title": "Тест" }, Authorization: Bearer <token>

Ожидаемый результат: Код 201, время ответа < 500 мс.

Негативные сценарии

1. TT_FS_TL_TS002_API_G/t_CL001_NS. Получение задач без токена

Запрос:

GET /tasks?status=all

Ожидаемый результат: Код 401, сообщение: «Требуется авторизация».

2. TT_FS_TL_TS002_API_G/t_CL002_NS. Получение задач с истекшим токеном

Запрос:

GET /tasks?status=all, Authorization: Bearer <expired_token>

Ожидаемый результат: Код 401, сообщение: «Недействительный токен».

3. TT_FS_TL_TS002_API_G/t_CL003_NS. Получение задач с некорректным статусом

Запрос:

GET /tasks?status=invalid, Authorization: Bearer <token>

Ожидаемый результат: Код 400, сообщение: «Недопустимое значение статуса».

4. TT_FS_TL_TS002_API_P/t_CL004_NS. Создание задачи с пустым заголовком

Запрос:

POST /tasks, { "title": "", "description": "Тест" }, Authorization: Bearer <token>

Ожидаемый результат: Код 400, сообщение: «Заголовок не может быть пустым».

5. TT_FS_TL_TS002_API_P/t_CL005_NS. Создание задачи с датой в прошлом

Запрос:

POST /tasks, { "title": "Тест", "dueDate": "13.05.2025" }, Authorization: Bearer <token>

Ожидаемый результат: Код 400, сообщение: «Дата выполнения не может быть в прошлом».

6. TT_FS_TL_TS002_API_P/t_CL006_NS. Создание задачи при превышении лимита (1001 задача)

Запрос:

POST /tasks, { "title": "Тест" } (после 1000 задач), Authorization: Bearer <token>

Ожидаемый результат: Код 400, сообщение: «Достигнут лимит задач».

7. TT_FS_TL_TS002_API_P/t_CL007_NS. Создание задачи с SQL-инъекцией в заголовке

Запрос:

POST /tasks, { "title": "Тест' OR 1=1; --" }, Authorization: Bearer <token>

Ожидаемый результат: Код 400, сообщение: «Недопустимые символы в заголовке», БД не затронута.

8. TT_FS_TL_TS002_API_P/t_CL008_NS. Создание задачи с длинным заголовком (256 символов)

Запрос:

POST /tasks, { "title": "<256 символов>" }, Authorization: Bearer <token>

Ожидаемый результат: Код 400, сообщение: «Заголовок слишком длинный».

Эндпоинт: GET /tasks/{id}, PUT /tasks/{id}, DELETE /tasks/{id}

Цель: Проверка операций с конкретной задачей.

Позитивные сценарии

1. TT_FS_TL_TS002_API_G/t/i_CL001_PS. Получение задачи по валидному ID

Запрос:

GET /tasks/1, Authorization: Bearer <token>

Ожидаемый результат: Код 200, ответ: { "id": 1, "title": "...", ... }, время < 500 мс.

2. TT_FS_TL_TS002_API_P/t/i_CL002_PS. Обновление задачи с валидными данными

Запрос:

PUT /tasks/1, { "title": "Новое название", "status": "completed" }, Authorization: Bearer <token>

Ожидаемый результат: Код 200, ответ: { "id": 1, "title": "Новое название", ... }.

3. TT_FS_TL_TS002_API_D/t/i_CL003_PS. Удаление задачи по валидному ID

Запрос:

DELETE /tasks/1, Authorization: Bearer <token>

Ожидаемый результат: Код 204, тело ответа пустое, время < 500 мс.

4. TT_FS_TL_TS002_API_G/t/i_CL004_PS. Проверка времени ответа при получении задачи

Запрос:

GET /tasks/1, Authorization: Bearer <token>

Ожидаемый результат: Код 200, время ответа < 500 мс.

5. TT_FS_TL_TS002_API_P/t/i_CL005_PS. Обновление задачи с минимальными данными

Запрос:

PUT /tasks/1, { "title": "Тест" }, Authorization: Bearer <token>

Ожидаемый результат: Код 200, ответ: { "id": 1, "title": "Тест", ... }.

Негативные сценарии

1. TT_FS_TL_TS002_API_G/t/i_CL001_NS. Получение несуществующей задачи

Запрос:

GET /tasks/999, Authorization: Bearer <token>

Ожидаемый результат: Код 404, сообщение: «Задача не найдена».

2. TT_FS_TL_TS002_API_G/t/i_CL002_NS. Получение чужой задачи

Запрос:

GET /tasks/2 (ID чужой задачи), Authorization: Bearer <token>

Ожидаемый результат: Код 404, сообщение: «Задача не найдена» (или 403: «Недостаточно прав»).

3. TT_FS_TL_TS002_API_P/t/i_CL003_NS. Обновление задачи с пустым заголовком

Запрос:

PUT /tasks/1, { "title": "" }, Authorization: Bearer <token>

Ожидаемый результат: Код 400, сообщение: «Заголовок не может быть пустым».

4. TT_FS_TL_TS002_API_P/t/i_CL004_NS. Обновление несуществующей задачи

Запрос:

PUT /tasks/999, { "title": "Тест" }, Authorization: Bearer <token>

Ожидаемый результат: Код 404, сообщение: «Задача не найдена».

5. TT_FS_TL_TS002_API_P/t/i_CL004_NS. Удаление без токена

Запрос:

DELETE /tasks/1

Ожидаемый результат: Код 401, сообщение: «Требуется авторизация».

Пример запросов/ответов в Postman

POST /login (Успешный вход)

```
{
  "method": "POST",
  "url": "https://api.todolist.example.com/login",
  "headers": {
    "Content-Type": "application/json"
  },
  "body": {
    "email": "test@example.com",
    "password": "Pass123!"
  }
}
```

Ожидаемый ответ:

```
{
  "status": 200,
  "body": {
    "token": "<JWT>"
  }
}
```

GET /tasks (Получение активных задач)

```
{
  "method": "GET",
  "url": "https://api.todolist.example.com/tasks?status=active",
  "headers": {
    "Authorization": "Bearer <JWT>"
  }
}
```

Ожидаемый ответ:

```
{
  "status": 200,
  "body": [
    {
      "id": 1,
```

```
    "title": "Тестовая задача",
    "description": "Описание задачи",
    "dueDate": "15.05.2025",
    "priority": "high",
    "status": "active"
  }
]
```

PUT /tasks/{id} (Обновление задачи)

```
{
  "method": "PUT",
  "url": "https://api.todolist.example.com/tasks/1",
  "headers": {
    "Content-Type": "application/json",
    "Authorization": "Bearer <JWT>"
  },
  "body": {
    "title": "Новое название",
    "status": "completed"
  }
}
```

Ожидаемый ответ:

```
{
  "status": 200,
  "body": {
    "id": 1,
    "title": "Новое название",
    "description": "Описание задачи",
    "dueDate": "15.05.2025",
    "priority": "high",
    "status": "completed"
  }
}
```

Python-скрипты для автоматизации API(Примеры)

Общие настройки

```
import requests
import time
import json

BASE_URL = "https://api.todolist.example.com"
TOKEN = None

def measure_time(func):
    def wrapper(*args, **kwargs):
        start = time.time()
        response = func(*args, **kwargs)
        elapsed = (time.time() - start) * 1000 # мс
        assert elapsed < 500, f"Время ответа {elapsed} мс превышает 500 мс"
        return response
    return wrapper
```

POST /login

```
@measure_time
def test_login_success():
    response = requests.post(f"{BASE_URL}/login", json={
        "email": "test@example.com",
        "password": "Pass123!"
    })
    assert response.status_code == 200, f"Ожидался код 200, получен {response.status_code}"
    assert "token" in response.json(), "Токен отсутствует в ответе"
    global TOKEN
    TOKEN = response.json()["token"]

@measure_time
def test_login_invalid_password():
    response = requests.post(f"{BASE_URL}/login", json={
        "email": "test@example.com",
        "password": "Wrong123!"
    })
    assert response.status_code == 401, f"Ожидался код 401, получен {response.status_code}"
```

```

        assert response.json()["message"] == "Неверный email или пароль"

@measure_time
def test_login_sql_injection():
    response = requests.post(f"{BASE_URL}/login", json={
        "email": "test' OR 1=1; --",
        "password": "Pass123!"
    })
    assert response.status_code == 400, f"Ожидался код 400, получен {response.status_code}"
    assert response.json()["message"] == "Некорректный формат email"

```

POST /tasks

```

@measure_time
def test_create_task_valid():
    response = requests.post(f"{BASE_URL}/tasks", json={
        "title": "Тестовая задача",
        "description": "Описание",
        "dueDate": "15.05.2025",
        "priority": "high"
    }, headers={"Authorization": f"Bearer {TOKEN}"})
    assert response.status_code == 201, f"Ожидался код 201, получен {response.status_code}"
    assert response.json()["title"] == "Тестовая задача"

@measure_time
def test_create_task_empty_title():
    response = requests.post(f"{BASE_URL}/tasks", json={
        "title": "",
        "description": "Тест"
    }, headers={"Authorization": f"Bearer {TOKEN}"})
    assert response.status_code == 400, f"Ожидался код 400, получен {response.status_code}"
    assert response.json()["message"] == "Заголовок не может быть пустым"

```

GET /tasks

```

@measure_time
def test_get_active_tasks():
    response = requests.get(f"{BASE_URL}/tasks?status=active",
        headers={"Authorization": f"Bearer {TOKEN}"})

```

```

    assert response.status_code == 200, f"Ожидался код 200, получен
{response.status_code}"
    assert isinstance(response.json(), list), "Ожидался массив задач"
    for task in response.json():
        assert task["status"] == "active", "Включена неактивная задача"

@measure_time
def test_get_tasks_no_token():
    response = requests.get(f"{BASE_URL}/tasks?status=all")
    assert response.status_code == 401, f"Ожидался код 401, получен
{response.status_code}"
    assert response.json()["message"] == "Требуется авторизация"

```

PUT /tasks/{id}, DELETE /tasks/{id}

```

@measure_time
def test_update_task_valid():
    response = requests.put(f"{BASE_URL}/tasks/1", json={
        "title": "Новое название",
        "status": "completed"
    }, headers={"Authorization": f"Bearer {TOKEN}"})
    assert response.status_code == 200, f"Ожидался код 200, получен
{response.status_code}"
    assert response.json()["title"] == "Новое название"

@measure_time
def test_delete_task_valid():
    response = requests.delete(f"{BASE_URL}/tasks/1",
headers={"Authorization": f"Bearer {TOKEN}"})
    assert response.status_code == 204, f"Ожидался код 204, получен
{response.status_code}"
    assert not response.text, "Ожидалось пустое тело ответа"

```

JavaScript скрипты для полуавтоматизации (для упрощенной работы с коллекциями в Postman, автоматический запрос и получение JWT токена)

```
// Postman Pre-request Script for automatic JWT token management
// Project: Todolist API
// Version: 1.0 (Beta)
// Requirements: Handles token fetching, expiration, multiple users, and
CI/CD compatibility

// Configuration
const API_BASE_URL = pm.environment.get("base_url") ||
"https://api.todolist.example.com";
const LOGIN_ENDPOINT = `${API_BASE_URL}/login`;
const TOKEN_VARIABLE = "token";
const EXPIRATION_VARIABLE = "token_expiration";
const TOKEN_TTL_MS = 24 * 60 * 60 * 1000; // 24 hours
const USERS = {
  default: {
    email: pm.environment.get("user_email") || "test@example.com",
    password: pm.environment.get("user_password") || "Pass123!"
  },
  admin: {
    email: pm.environment.get("admin_email") || "admin@example.com",
    password: pm.environment.get("admin_password") || "Admin123!"
  }
};

// Get current user role from environment (default: 'default')
const CURRENT_USER_ROLE = pm.environment.get("user_role") || "default";
const CURRENT_USER = USERS[CURRENT_USER_ROLE] || USERS.default;
// Utility functions
const log = (message, level = "INFO") => {
  console.log(`[${level}] ${new Date().toISOString()} - ${message}`);
};
const throwError = (message) => {
  log(message, "ERROR");
  throw new Error(message);
};
// Check if token is valid
const token = pm.environment.get(TOKEN_VARIABLE);
const tokenExpiration = parseInt(pm.environment.get(EXPIRATION_VARIABLE) ||
0);
```

```

const isValidToken = token && Date.now() < tokenExpiration;
if (isValidToken) {
  log(`Using existing token for user: ${CURRENT_USER.email}`);
} else {
  log(`Token missing or expired for user: ${CURRENT_USER.email}. Fetching
new token...`);

  // Login request configuration
  const loginRequest = {
    url: LOGIN_ENDPOINT,
    method: "POST",
    header: {
      "Content-Type": "application/json"
    },
    body: {
      mode: "raw",
      raw: JSON.stringify({
        email: CURRENT_USER.email,
        password: CURRENT_USER.password
      })
    }
  };

  // Send login request
  pm.sendRequest(loginRequest, (err, response) => {
    if (err) {
      throwError(`Failed to fetch token: ${err.message}`);
    }
    const status = response.code;
    const responseBody = response.json();
    switch (status) {
      case 200:
        const newToken = responseBody.token;
        if (!newToken) {
          throwError("Token not found in response");
        }
        pm.environment.set(TOKEN_VARIABLE, newToken);
        pm.environment.set(EXPIRATION_VARIABLE, Date.now() +
TOKEN_TTL_MS);
        log(`Token fetched successfully for user:
${CURRENT_USER.email}`);
        break;
      case 400:
        throwError(`Bad request: ${responseBody.message} || "Invalid

```

```

request format"}`);
    break;
    case 401:
        throwError(`Authentication failed: ${responseBody.message
|| "Invalid email or password"}`);
        break;
    case 429:
        throwError(`Rate limit exceeded: ${responseBody.message ||
"Too many login attempts"}`);
        break;
    default:
        throwError(`Unexpected response: ${status} -
${responseBody.message || "Unknown error"}`);
    }
});
}

```

Риски и приоритеты для веб-приложения “Список задач”

Таблица рисков и приоритетов для API

№	Функция	Риск	Последствия	Приоритет	План смягчения
1	Аутентификация (POST /login)	Сбой аутентификации из-за неверной обработки учетных данных	Пользователи не могут войти, блокировка всей функциональности	Критический	<ul style="list-style-type: none"> - Добавить тесты на граничные случаи (длинный email, пробелы). - Проверить серверную валидацию и обработку ошибок (400, 401). - Настроить мониторинг ошибок 401 в продакшене.

2	Аутентификация (POST /login)	Уязвимость к SQL-инъекциям или XSS в полях email/пароль	Компрометация данных пользователей, утечка сессий	Критический	<ul style="list-style-type: none"> - Расширить тесты на инъекции (тест №9, №10). - Внедрить серверную санитизацию входных данных. - Использовать параметризованные запросы в БД.
3	Аутентификация (POST /login)	Превышение лимита запросов (100/мин)	Ошибка 429, пользователи временно заблокированы	Высокий	<ul style="list-style-type: none"> - Добавить тесты на лимит запросов (тест №13). - Настроить информативное сообщение об ошибке 429. - Рассмотреть увеличение лимита для продакшена.
4	Управление задачами (POST /tasks)	Ошибка 500 при создании задачи с некорректными данными (например, пустой заголовок)	Блокировка создания задач, нестабильность API	Критический	<ul style="list-style-type: none"> - Исправить серверную валидацию (баг-репорт №1). - Добавить тесты на пустые/граничные поля (тест №25, №29). - Внедрить try-catch в маршруте /tasks.
5	Управление задачами (POST /tasks)	Превышение лимита задач (1000 на пользователя)	Ошибка 400, невозможность создать новую задачу	Высокий	<ul style="list-style-type: none"> - Добавить тест на лимит задач (тест №27). - Уточнить UX для сообщения об ошибке. - Рассмотреть механизм очистки старых задач.

6	Управление задачами (GET /tasks)	Некорректная фильтрация задач по статусу	Пользователи видят неверные списки задач, снижение UX	Средний	<ul style="list-style-type: none"> - Расширить тесты на фильтрацию (тест №17, №18). - Проверить серверную логику фильтров. - Добавить клиентскую проверку статуса.
7	Управление задачами (PUT /tasks/{id})	Неверное обновление статуса задачи в UI	Пользователи видят некорректный статус, путаница в управлении	Высокий	<ul style="list-style-type: none"> - Исправить синхронизацию UI (баг-репорт №2). - Добавить тесты на обновление статуса (тест №34). - Внедрить логирование клиентских ошибок.
8	Управление задачами (GET/PUT/DELETE /tasks/{id})	Доступ к чужим задачам	Утечка данных, нарушение конфиденциальности	Критический	<ul style="list-style-type: none"> - Добавить тесты на доступ к чужим задачам (тест №33). - Проверить серверную проверку прав доступа. - Внедрить аудит логов доступа.
9	Безопасность (все эндпоинты)	Истечение токена во время сессии	Ошибка 401, прерывание работы пользователей	Высокий	<ul style="list-style-type: none"> - Добавить тесты на истекший токен (тест №24). - Внедрить механизм автоматического обновления токена на клиенте. - Увеличить срок действия токена для продакшена (например, 7 дней).

10	Производительность (все эндпоинты)	Время ответа > 500 мс при высокой нагрузке	Замедление работы, снижение UX	Высокий	<ul style="list-style-type: none"> - Провести нагрузочное тестирование (расширить тесты №15, №22, №30, №34). - Оптимизировать запросы к БД (индексы, кэширование). - Настроить мониторинг производительности.
11	Локализация (все эндпоинты)	Непонятные или отсутствующие сообщения об ошибках	Снижение UX, путаница для пользователей	Средний	<ul style="list-style-type: none"> - Исправить сообщения об ошибках (баг-репорт №3). - Добавить тесты на локализацию ошибок (400, 401, 429). - Проверить переводы на русском.
12	Адаптивность (GET /tasks)	Некорректное отображение длинных заголовков на мобильных устройствах	Снижение UX на мобильных, ограниченный доступ к данным	Низкий	<ul style="list-style-type: none"> - Исправить CSS-стили (баг-репорт №4). - Добавить тесты UI на мобильных разрешениях (320px). - Рассмотреть ограничение длины заголовка в UI.

Таблица рисков и приоритетов для Frontend

№	Критерий	Риск	Проверка	Приоритет	Обоснование
1	Регистрация	Уязвимость к XSS	Проверить, что ввод <code><script>alert('test')</script></code> в поле заголовка задачи экранируется и не выполняется (TT_FS_TL_TS 001_CL019_NS).	Критический	Проверяет защиту от XSS, предотвращая выполнение вредоносных скриптов, что критично для безопасности пользователей
2	Регистрация	Уязвимость к SQL-инъекциям	Проверить, что ввод <code>test' OR 1=1; --</code> в поле email вызывает ошибку «Некорректный формат email» (TT_FS_TL_TS 001_CL018_NS).	Критический	Проверяет защиту от SQL-инъекций, предотвращая несанкционированный доступ к данным
3	Регистрация	Блокировка регистрации	Проверить, что при существующем email отображается ошибка «Email уже зарегистрирован» (TT_FS_TL_TS 001_CL001_NS).	Критический	Проверяет корректность обработки дублирующихся email, предотвращая конфликты пользователей
4	Регистрация	Валидация ввода	Проверить, что пустое поле email вызывает ошибку «Поле email не может быть пустым» (TT_FS_TL_TS	Критический	Проверяет валидацию обязательного поля email, предотвращая некорректную регистрацию

			001_CL002_NS).		
5	Регистрация	Валидация ввода	Проверить, что некорректный email (test@) вызывает ошибку «Некорректный формат email» (TT_FS_TL_TS 001_CL003_NS).	Критически	Проверяет валидацию формата email, обеспечивая корректность данных
6	Аутентификация	Блокировка входа	Проверить, что неверный пароль вызывает ошибку «Неверный email или пароль» (TT_FS_TL_TS 001_CL006_NS).	Критически	Проверяет обработку неверных учетных данных, предотвращая несанкционированный доступ
7	Аутентификация	Блокировка входа	Проверить, что несуществующий email вызывает ошибку «Неверный email или пароль» (TT_FS_TL_TS 001_CL007_NS).	Критически	Проверяет обработку несуществующих учетных данных, предотвращая доступ
8	Аутентификация	Валидация ввода	Проверить, что пустое поле email вызывает ошибку «Поле email не может быть пустым» (TT_FS_TL_TS 001_CL008_NS).	Критически	Проверяет валидацию обязательного поля email, предотвращая некорректный вход

9	Аутентификация	Безопасность	Проверить, что 5 неверных попыток входа вызывают ошибку «Слишком много попыток» (TT_FS_TL_TS 001_CL009_NS).	Критически	Проверяет защиту от brute-force атак, обеспечивая безопасность системы
10	Создание задачи	Валидация ввода	Проверить, что пустой заголовок вызывает ошибку «Заголовок не может быть пустым» (TT_FS_TL_TS 001_CL011_NS).	Критически	Проверяет валидацию обязательного поля заголовка, предотвращая создание некорректных задач
11	Создание задачи	Валидация ввода	Проверить, что дата в прошлом вызывает ошибку «Дата выполнения не может быть в прошлом» (TT_FS_TL_TS 001_CL012_NS).	Критически	Проверяет валидацию даты, обеспечивая корректность задач
12	Создание задачи	Лимит задач	Проверить, что 1001-я задача вызывает ошибку «Достигнут лимит задач» (TT_FS_TL_TS 001_CL013_NS).	Критически	Проверяет ограничение на количество задач, предотвращая перегрузку системы

13	Регистрация	Валидация ввода	Проверить, что пустое поле пароля вызывает ошибку «Поле пароль не может быть пустым» (TT_FS_TL_TS 001_CL005_NS).	Критически	Проверяет валидацию обязательного поля пароля, предотвращая некорректную регистрацию
14	Аутентификация	Валидация ввода	Проверить, что пустой пароль вызывает ошибку «Поле пароль не может быть пустым» (TT_FS_TL_TS 001_CL031_NS).	Критически	Проверяет валидацию обязательного поля пароля, предотвращая некорректный вход
15	Создание задачи	Безопасность	Проверить, что HTML-теги в описании экранируются (Текст → Текст) (TT_FS_TL_TS 001_CL029_NS).	Критически	Проверяет защиту от XSS в описании задач, предотвращая выполнение скриптов
16	Аутентификация	Функциональность	Проверить, что валидные данные (email, пароль) авторизуют и перенаправляют к списку задач (TT_FS_TL_TS 001_CL003_PS).	Высокий	Проверяет корректность входа, обеспечивая доступ к основным функциям приложения

17	Создание задачи	Функциональность	Проверить, что задача с полными данными создается и отображается в списке (TT_FS_TL_TS 001_CL007_PS)	Высокий	Проверяет создание задач, что является основной функцией приложения
18	Создание задачи	Функциональность	Проверить, что задача с минимальными данными (заголовок) создается с приоритетом «средний» (TT_FS_TL_TS 001_CL008_PS)	Высокий	Проверяет минимальное создание задач, обеспечивая гибкость ввода
19	Редактирование задачи	Функциональность	Проверить, что изменение всех полей задачи обновляет список (TT_FS_TL_TS 001_CL009_PS)	Высокий	Проверяет редактирование задач, что критично для управления задачами
20	Список задач	Статус задачи	Проверить, что пометка задачи как выполненной перемещает её в «Выполненные» (TT_FS_TL_TS 001_CL010_PS)	Высокий	Проверяет корректность отображения статуса задачи, влияющего на UX
21	Список задач	Статус задачи	Проверить, что снятие чекбокса возвращает	Высокий	Проверяет переключение статуса

			задачу в «Активные» (TT_FS_TL_TS 001_CL034_PS) .		задачи, критично для UX
22	Удаление задачи	Функциональность	Проверить, что удаление задачи с подтверждением убирает её из списка (TT_FS_TL_TS 001_CL011_PS) .	Высокий	Проверяет удаление задач, что является ключевой функцией
23	Список задач	Функциональность	Проверить, что фильтр «Все» отображает все задачи (TT_FS_TL_TS 001_CL012_PS) .	Высокий	Проверяет фильтрацию задач, обеспечивая удобство просмотра
24	Список задач	Функциональность	Проверить, что фильтр «Активные» отображает только невыполненные задачи (TT_FS_TL_TS 001_CL013_PS) .	Высокий	Проверяет фильтрацию активных задач, улучшая UX
25	Список задач	Функциональность	Проверить, что фильтр «Выполненные» отображает только выполненные задачи (TT_FS_TL_TS 001_CL014_PS) .	Высокий	Проверяет фильтрацию выполненных задач, улучшая UX

26	Регистрация	Функциональность	Проверить, что регистрация с валидными данными отображает сообщение «Email подтвержден» после клика по ссылке (TT_FS_TL_TS 001_CL002_PS)	Высокий	Проверяет процесс подтверждения email, обеспечивая завершение регистрации
27	Аутентификация	Функциональность	Проверить, что опция «Запомнить меня» сохраняет сессию 30 дней (TT_FS_TL_TS 001_CL004_PS)	Высокий	Проверяет удобство входа, улучшая UX за счет сохранения сессии
28	Сброс пароля	Функциональность	Проверить, что сброс пароля для существующего email отображает форму нового пароля (TT_FS_TL_TS 001_CL005_PS)	Высокий	Проверяет функциональность сброса пароля, важную для восстановления доступа
29	Сброс пароля	Функциональность	Проверить, что новый пароль обновляет доступ и перенаправляет к входу (TT_FS_TL_TS 001_CL006_PS)	Высокий	Проверяет завершение сброса пароля, обеспечивая доступ

30	Список задач	Производительность	Проверить, что фильтрация 100 задач выполняется < 200 мс (TT_FS_TL_TS 001_CL031_PS)	Высокий	Проверяет производительность фильтрации, обеспечивая быстрый отклик UI
31	Список задач	Производительность	Проверить, что список задач рендерится < 200 мс на десктопе 1920x1080 (TT_FS_TL_TS 001_CL015_PS)	Высокий	Проверяет производительность рендеринга, улучшая UX
32	Аутентификация	UX	Проверить, что вход на мобильном устройстве (320x568) отображает адаптивный список задач (TT_FS_TL_TS 001_CL032_PS)	Высокий	Проверяет адаптивность входа, обеспечивая доступность на мобильных устройствах
33	Удаление задачи	UX	Проверить, что отмена удаления сохраняет задачу в списке (TT_FS_TL_TS 001_CL017_NS)	Высокий	Проверяет защиту от случайного удаления, улучшая UX
34	Создание задачи	Валидация ввода	Проверить, что заголовок > 255 символов вызывает ошибку «Заголовок слишком длинный»	Высокий	Проверяет валидацию длины заголовка, предотвращая некорректные данные

			(TT_FS_TL_TS 001_CL014_NS).		
35	Создание задачи	Валидация ввода	Проверить, что описание > 1000 символов вызывает ошибку «Описание слишком длинное» (TT_FS_TL_TS 001_CL015_NS).	Высокий	Проверяет валидацию длины описания, предотвращая некорректные данные
36	Редактирование задачи	Валидация ввода	Проверить, что дата в прошлом вызывает ошибку «Дата выполнения не может быть в прошлом» (TT_FS_TL_TS 001_CL028_NS).	Высокий	Проверяет валидацию даты при редактировании, обеспечивая корректность
37	Создание задачи	Валидация ввода	Проверить, что некорректная дата (32.05.2025) вызывает ошибку «Некорректный формат даты» (TT_FS_TL_TS 001_CL020_NS).	Высокий	Проверяет валидацию формата даты, предотвращая ошибки ввода
38	Регистрация	Валидация ввода	Проверить, что слабый пароль вызывает ошибку «Слабый пароль» (TT_FS_TL_TS	Высокий	Проверяет валидацию пароля, обеспечивая безопасность учетной записи

			001_CL004_NS).		
39	Сброс пароля	Валидация ввода	Проверить, что слабый пароль вызывает ошибку «Слабый пароль» (TT_FS_TL_TS 001_CL033_NS).	Высокий	Проверяет валидацию нового пароля, обеспечивая безопасность
40	Сброс пароля	Валидация ввода	Проверить, что несуществующий email вызывает ошибку «Email не найден» (TT_FS_TL_TS 001_CL010_NS).	Высокий	Проверяет обработку несуществующих email, улучшая UX сброса пароля
41	Список задач	Локализация	Проверить, что сообщение «Нет задач, соответствующих фильтру» отображается при пустом списке (TT_FS_TL_TS 001_CL019_PS).	Средний	Проверяет локализацию сообщения, улучшая UX для русскоязычных пользователей
42	Список задач	Локализация	Проверить, что фильтр «Выполненные» при отсутствии задач показывает «Нет задач» (TT_FS_TL_TS 001_CL024_NS).	Средний	Проверяет локализацию сообщения для фильтра, улучшая UX

43	Список задач	Локализация	Проверить, что фильтр «Активные» при отсутствии задач показывает «Нет задач» (TT_FS_TL_TS 001_CL024_NS).	Средний	Проверяет локализацию сообщения для фильтра, улучшая UX
44	Регистрация	Локализация	Проверить, что кириллическое имя (Екатерина) отображается в профиле (TT_FS_TL_TS 001_CL026_PS).	Средний	Проверяет поддержку кириллицы, обеспечивая локализацию
45	Создание задачи	Локализация	Проверить, что кириллическое описание отображается в списке (TT_FS_TL_TS 001_CL027_PS).	Средний	Проверяет поддержку кириллицы в задачах, улучшая локализацию
46	Список задач	UX	Проверить, что обновление страницы (F5) сохраняет список и фильтры (TT_FS_TL_TS 001_CL020_PS).	Средний	Проверяет стабильность UI при обновлении, улучшая UX
47	UI/UX	Навигация	Проверить, что навигация по интерфейсу с Tab работает для всех элементов (TT_FS_TL_TS	Средний	Проверяет доступность интерфейса, улучшая UX для пользователей с клавиатурой

			001_CL018_PS) .		
48	Создание задачи	Функциональность	Проверить, что задача с максимальной длиной заголовка (255 символов) отображается полностью (TT_FS_TL_TS 001_CL021_PS) .	Средний	Проверяет обработку граничных значений заголовка, обеспечивая корректность отображения
49	Создание задача	Функциональность	Проверить, что задача с максимальной длиной описания (1000 символов) сохраняется (TT_FS_TL_TS 001_CL022_PS) .	Средний	Проверяет обработку граничных значений описания, обеспечивая корректность
50	Редактирование задачи	Функциональность	Проверить, что изменение только приоритета обновляет задачу (TT_FS_TL_TS 001_CL023_PS) .	Средний	Проверяет частичное редактирование, улучшая гибкость управления задачами
51	Создание задачи	Функциональность	Проверить, что задача с текущей датой отображается корректно (TT_FS_TL_TS 001_CL025_PS) .	Средний	Проверяет обработку текущей даты, обеспечивая корректность задач

52	Регистрация	Функциональность	Проверить, что имя из 50 символов сохраняется в профиле (TT_FS_TL_TS 001_CL033_PS) .	Средний	Проверяет обработку граничных значений имени, обеспечивая корректность профиля
53	Редактирование задачи	Функциональность	Проверить, что удаление описания сохраняет остальные данные (TT_FS_TL_TS 001_CL028_PS) .	Средний	Проверяет частичное редактирование, улучшая гибкость
54	Создание задача	Функциональность	Проверить, что задача без даты создается с пустым полем даты (TT_FS_TL_TS 001_CL029_PS) .	Средний	Проверяет опциональность даты, улучшая гибкость ввода
55	Создание задача	Функциональность	Проверить, что задача с низким приоритетом отображает «Низкий» (TT_FS_TL_TS 001_CL030_PS) .	Средний	Проверяет корректность установки приоритета, улучшая UX
56	Регистрация	Валидация ввода	Проверить, что некорректные символы в имени вызывают ошибку «Имя содержит недопустимые символы» (TT_FS_TL_TS	Средний	Проверяет валидацию имени, предотвращая некорректные данные

			001_CL021_NS).		
57	Регистрация	Валидация ввода	Проверить, что имя > 50 символов вызывает ошибку «Имя слишком длинное» (TT_FS_TL_TS 001_CL022_NS).	Средний	Проверяет валидацию длины имени, обеспечивая корректность профиля
58	Создание задача	Валидация ввода	Проверить, что некорректный формат даты (2025-05-15) вызывает ошибку «Некорректный формат даты» (TT_FS_TL_TS 001_CL023_NS).	Средний	Проверяет валидацию формата даты, предотвращая ошибки
59	Создание задача	Валидация ввода	Проверить, что некорректный приоритет вызывает ошибку «Недопустимое значение приоритета» (TT_FS_TL_TS 001_CL027_NS).	Средний	Проверяет валидацию приоритета, обеспечивая корректность задач
60	Создание задача	Валидация ввода	Проверить, что некорректные символы в заголовке вызывают ошибку или экранируются (TT_FS_TL_TS 001_CL032_NS).	Средний	Проверяет валидацию заголовка, предотвращая некорректные данные

61	Адаптивность	Мобильный UX	Проверить, что список задач отображается корректно на 320x568 (TT_FS_TL_TS 001_CL016_PS) .	Низкий	Проверяет адаптивность интерфейса, улучшая UX на мобильных устройствах
62	Адаптивность	Мобильный UX	Проверить, что создание задачи на планшете (768x1024) отображает адаптивную форму (TT_FS_TL_TS 001_CL017_PS) .	Низкий	Проверяет адаптивность формы создания, улучшая UX на планшетах
63	Адаптивность	Мобильный UX	Проверить, что удаление задачи на мобильном устройстве (320x568) отображает адаптивное уведомление (TT_FS_TL_TS 001_CL035_PS) .	Низкий	Проверяет адаптивность удаления, улучшая мобильный UX
64	UI/UX	JavaScript	Проверить, что при отключенном JavaScript отображается «Включите JavaScript» (TT_FS_TL_TS 001_CL025_NS).	Низкий	Проверяет обработку отсутствия JavaScript, обеспечивая информирование

65	Регистрация	Валидация ввода	Проверить, что email с пробелами вызывает ошибку «Некорректный формат email» или обрезается (TT_FS_TL_TS 001_CL026_NS).	Низкий	Проверяет валидацию email, предотвращая мелкие ошибки ввода
66	Регистрация	Валидация ввода	Проверить, что email > 254 символов вызывает ошибку «Email слишком длинный» (TT_FS_TL_TS 001_CL030_NS).	Низкий	Проверяет обработку граничных значений email, предотвращая некорректные данные
67	Сброс пароля	Валидация ввода	Проверить, что email > 254 символов вызывает ошибку «Email слишком длинный» (TT_FS_TL_TS 001_CL034_NS).	Низкий	Проверяет валидацию email при сбросе, предотвращая ошибки
68	Список задач	Фильтрация	Проверить, что некорректный фильтр возвращает дефолтный «Все» (TT_FS_TL_TS 001_CL035_NS).	Низкий	Проверяет устойчивость фильтрации, предотвращая сбои UI

69	Регистрация	Функциональность	Проверить, что регистрация отображает форму с полями email, пароль, имя (TT_FS_TL_TS 001_CL001_PS) .	Низкий	Проверяет базовое отображение формы регистрации, обеспечивая доступность UI
70	Список задач	UX	Проверить, что список задач отображается корректно на десктопе 1920x1080 (TT_FS_TL_TS 001_CL015_PS) .	Низкий	Проверяет корректность отображения списка задач, улучшая десктопный UX