## Документация по Task Manager

### 1. Введение

Task Manager — это система управления задачами, разработанная для оптимизации рабочих процессов внутри компании. Система позволяет создавать, назначать, отслеживать и анализировать задачи, а также взаимодействовать с коллегами в рамках проектов. Использовать будет сотрудники компании и руководители.

Зачем Task Manager? Оптимизация процессов, улучшение эффективности, упрощение коммуникации

### 2. Основные функции

#### 2.1. Создание задач

* **Название задачи**: Краткое описание задачи.
* **Описание**: Детальное описание задачи, включая цели и ожидаемые результаты.
* **Директор/Холдинг**: Назначение задачи конкретному отделу или группе.
* **Руководитель отдела**: создает задачу сотруднику или группе.
* **Сроки**: Установка дедлайна для выполнения задачи.
* **Приоритет**: Выбор приоритета задачи (низкий, средний, высокий).
* **Теги**: Добавление тегов для удобной категоризации задач.
* **Приоритет задач**: Можно добавить разъяснение, какие факторы влияют на выбор приоритета (например, важность задачи для бизнеса, сроки и т.д.).
* **Автоматизация**: Если планируется автоматизация каких-либо процессов (например, автоматическая смена статуса задачи по времени)
* **Дублирование задач**: Возможность создания шаблонов задач и их дублирования для повторяющихся задач.

#### 2.2. Управление задачами

* **Статусы задач**: Возможность изменять статус задачи (например, "Новая", "В работе", "На проверке", "Завершена").
* **Комментарии**: Обсуждение задачи в режиме реального времени.
* **Вложения**: Добавление файлов, документов или изображений к задаче.
* **Подзадачи**: Разделение задачи на более мелкие подзадачи.

#### 2.3. Проекты

* **Создание проектов**: Группировка задач по проектам.
* **Участники проекта**: Добавление сотрудников к проекту.
* **Доски задач**: Визуализация задач в виде канбан-досок.

#### 2.4. Коммуникация

* Чат компании – групповое обсуждение рабочих вопросов.
* Личные сообщения – переписка между сотрудниками.
* Упоминания – возможность пригласит коллег в обсуждениях.

#### 2.5. Календарь

* **Система автоматически уведомляет** о днях рождения сотрудников.
* **Планирование задач**: Назначение задач на конкретные даты.
* **Напоминания**: Уведомления о предстоящих дедлайнах и про мероприятиях.
* **Обычный календарь с наступающим праздниках**
* **Интеграция** с задачами и напоминаниями

#### 2.6. Отчеты и аналитика

* **Отчеты по задачам**: Анализ выполнения задач по сотрудникам, проектам или срокам.
* **Диаграммы Ганта**: Визуализация сроков выполнения задач.

### 3. Руководство пользователя

#### 3.1. Регистрация и вход

1. Перейдите на страницу входа.
2. Введите логин и пароль, предоставленные администратором.
3. Если вы новый пользователь, зарегистрируйтесь, используя корпоративную почту.

#### 3.2. Создание задачи

1. Нажмите кнопку "Создать задачу".
2. Заполните все необходимые поля (название, описание, исполнитель, сроки и т.д.).
3. Нажмите "Сохранить".

#### 3.3. Работа с проектами

1. Перейдите в раздел "Проекты".
2. Нажмите "Создать проект".
3. Добавьте задачи и участников проекта.

#### 3.4. Использование календаря

1. Перейдите в раздел "Календарь".
2. Перетащите задачу на нужную дату.
3. Установите напоминание, если необходимо.

##### *3.5. Лента Новостей*

1. Сотрудники может посмотреть событья дня/недели/месяца.
2. Коментарий: Сотрудики может коментировать/поздравить и тд.

### 4. Руководство администратора

#### 4.1. Настройка системы

* **Роли и права доступа**: Настройка прав доступа для сотрудников.
* **Шаблоны задач**: Создание шаблонов для часто повторяющихся задач.
* **Интеграции**: Подключение сторонних сервисов (например, почта, CRM).

#### 4.2. Управление пользователями

* **Добавление новых пользователей**.
* **Назначение ролей и прав доступа**.
* **Блокировка или удаление пользователей**.

### 5. Интеграции

* **Электронная почта**: Возможность создания задач из писем.
* **Мессенджеры**: Интеграция с **Telegram** и другими платформами.
* **CRM**: Синхронизация с CRM-системой для управления задачами, связанными с клиентами.

6. Отправка вложении

* **Пользователи может прикрепи вложении на коментарий задач**
* **Пользователи может отправиьт вложении через чат**

### 7. Часто задаваемые вопросы (FAQ)

#### 7.1. Как изменить статус задачи?

* Откройте задачу и выберите новый статус в выпадающем меню.

#### 7.2. Как добавить участника в проект?

* Перейдите в настройки проекта и нажмите "Добавить участника".

#### 7.3. Как настроить уведомления?

* Перейдите в "Настройки профиля" и выберите раздел "Уведомления".

### 8. Поддержка

Если у вас возникли вопросы или проблемы, обратитесь в службу поддержки:

* **Электронная почта**: [support@company.com](mailto:support@company.com)
* **Телефон**: +7 (XXX) XXX-XX-XX
* **Чат поддержки**: Доступен в правом нижнем углу экрана.

**Требование к Backend**

**1. База данных (PostgreSQL):**

Основная БД для хранения всех данных: пользователи, задачи, проекты, комментарии и т.д.

JSONB-поля для тегов и метаданных задач, чтобы избежать множества связей.

Индексы на ключевые поля (status, priority, deadline) для быстрого поиска.

Резервное копирование базы данных: Происходит резервное копирование данных, чтобы предотвратить потерю информации.

**2. API и Реалтайм (Django REST Framework + Django Channels):**

REST API (DRF) → CRUD-операции для задач, проектов, пользователей.

WebSockets (Django Channels + Redis) → Чат, уведомления в реальном времени.

**3. Фоновая обработка задач (Celery + Redis):**

Отправка уведомлений (почта, Telegram).

Генерация отчетов и диаграмм.

Очистка старых данных.

**4. Аутентификация (SimpleJWT):**

JWT-токены для API.

Refresh-токен для автоматического обновления.

**5. Кэширование (Redis):**

Кэширование популярных запросов (списки задач, проектов).

Rate-limiting для API.

**6. Роли и доступ (Django Permissions):**

Админы – управление пользователями и настройками.

Руководители – назначение задач, контроль.

Сотрудники – выполнение задач.

**7. Логирование и мониторинг**

Django Logging + Sentry для ошибок.

Prometheus + Grafana для метрик.

**8. Тестирование**

Django unittest тестирование

### ****8. Развертывание (Docker + Nginx)****

**Backend** → Gunicorn + Uvicorn (Django + ASGI)  
 **Frontend** → Nginx (статические файлы Vue)  
 **БД + Redis + Celery** → Оркестрация через Docker Compose

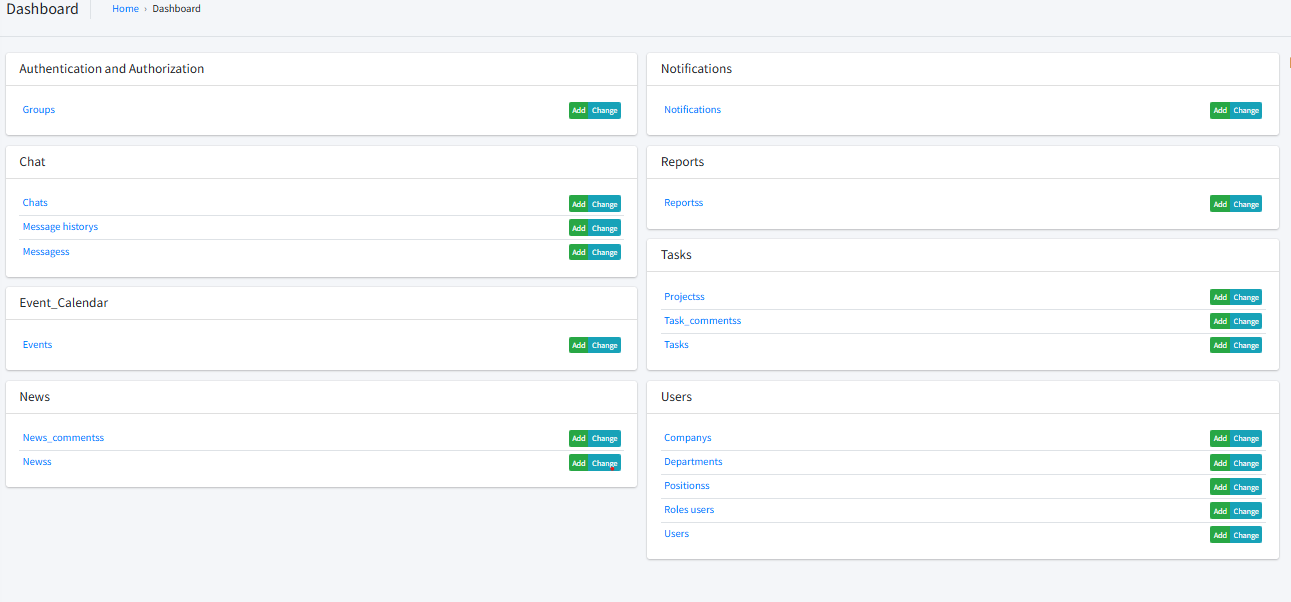
### ****Итог****

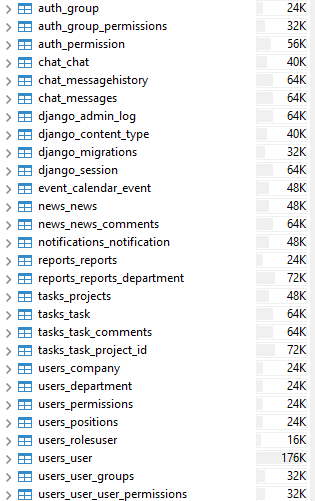
### ****Django API + WebSockets + Celery**** → Надежный и быстрый backend ****Vue 3 + Ant Design**** → Удобный и красивый frontend ****Redis + Celery**** → Реалтайм и фоновые задачи ****Docker + Nginx**** → Гибкое развертывание

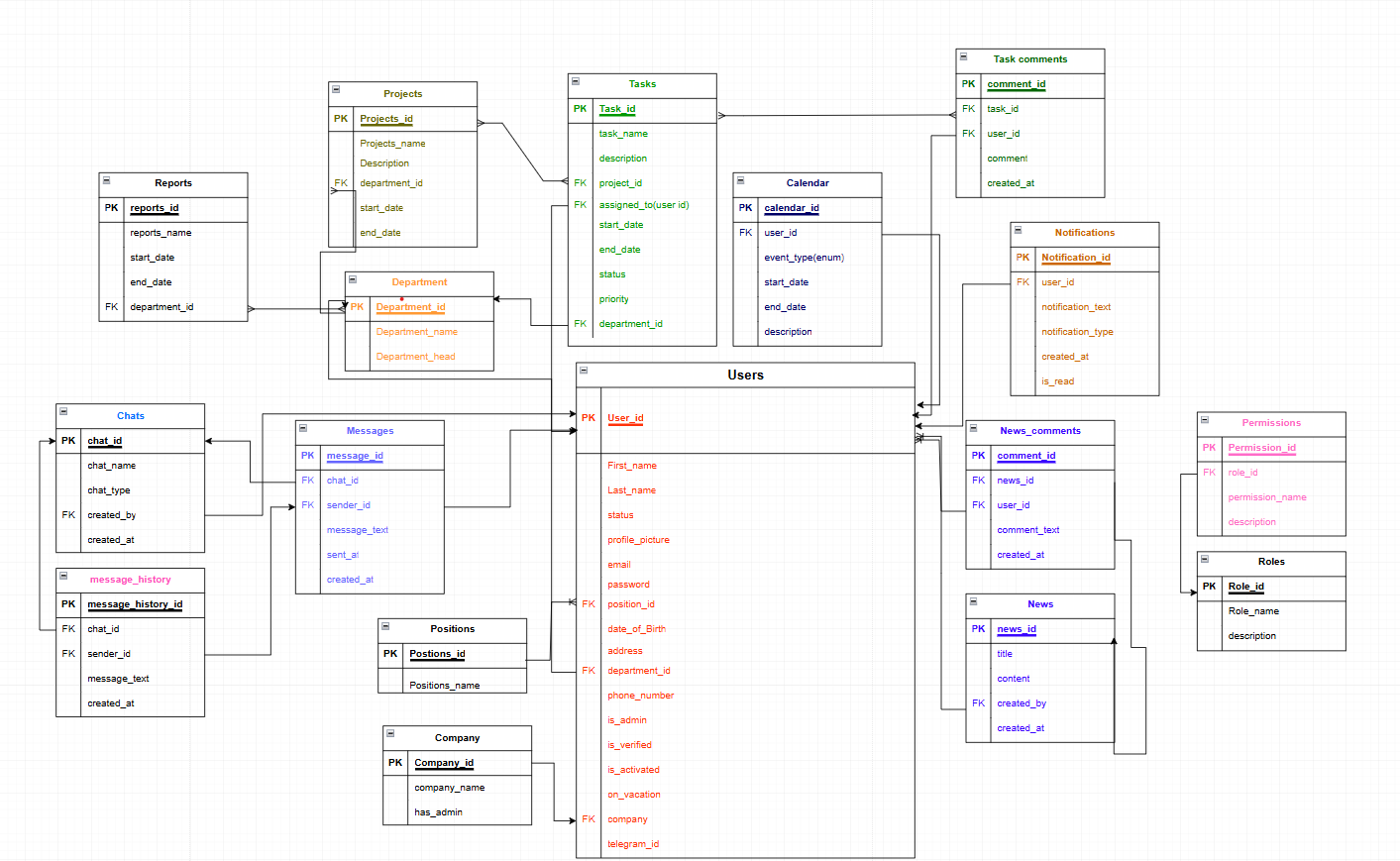
### Авторизация

### 

**Таблицы**

****

****



**Требование к Front-end**

**1. Аутентификация и авторизация**

* **Регистрация и вход**: Должен быть экран для логина/регистрации, с формами для ввода логина и пароля.
* **Роли и доступ**: На основе ролей пользователей (например, руководитель, сотрудник) фронтенд должен отображать разные разделы интерфейса. Например, доступ к созданию задач или проектов — только для определённых ролей.
* **Токены**: JWT для авторизации через API, чтобы сохранять сессию и проверять доступ.

**2. Главный интерфейс**

* **Dashboard**: После входа пользователя, нужно показывать главную страницу с обзором всех задач, проектов, уведомлений. Это должна быть основная точка входа.
* **Навигация**: Навигация по проектам, задачам, календарю и новостям. Это может быть боковое меню с иконками и названиями разделов.

**3. Работа с задачами**

* **Создание и редактирование задач**: Форма для создания задачи с полями: название, описание, приоритет, сроки, исполнитель, теги и т. п.
* **Управление статусами**: Возможность изменять статус задачи через UI (например, с помощью выпадающего меню или переключателей).
* **Комментарии**: Реализация раздела для комментирования задач с отображением времени, кто оставил комментарий.
* **Подзадачи**: Возможность добавлять подзадачи к основной задаче. Это может быть в виде вложенных списков или раскрывающихся блоков.
* **Файлы**: Возможность прикреплять файлы к задачам (документы, изображения и т. д.).

**4. Проекты**

* **Список проектов**: Страница, где пользователи могут видеть все свои проекты, с фильтрацией и сортировкой.
* **Создание проекта**: Форма для создания проектов, с полями для указания участников, описания и задач.
* **Участники проекта**: Возможность добавлять или удалять участников из проекта.

**5. Календарь**

* **Календарь с задачами**: где задачи будут отображаться по датам. У пользователя должна быть возможность перетащить задачу на новую дату (drag-and-drop).
* **Напоминания**: Уведомления о предстоящих дедлайнах. Могут быть в виде баннера, всплывающего окна или просто уведомления в UI.

**6. Новости и коммуникация**

* **Лента новостей**: Панель новостей.
* **Комментарии к новостям**: Возможность комментировать новости. Пользователи должны иметь возможность оставлять свои комментарии или поздравления.
* **Чат**: Встроенная система чатов для общения между сотрудниками, как минимум для общения внутри проектов и задач.

**7. Уведомления**

* **Уведомления**: Каждый раз, когда происходят важные изменения (новая задача, комментарий, изменение статуса), должно быть уведомление.
* **Интерфейс уведомлений**: Плоский список уведомлений, с кнопкой "Прочитано", чтобы отмечать прочитанное. Можно добавить фильтрацию по типам уведомлений.
* **Оповещения в реальном времени**: Через WebSocket или другие технологии для обновлений в реальном времени.

**8. Отчёты и аналитика**

* **Диаграммы Ганта**: Графическое представление выполнения задач в проекте с возможностью перетаскивать задачи, изменяя их сроки.
* **Отчёты**: Страница с фильтруемыми отчетами по задачам: кто сколько сделал, какие задачи просрочены, по проектам и срокам.

**9. UI/UX**

* **Адаптивность**: Интерфейс должен быть адаптивным и работать как на мобильных устройствах, так и на десктопе.
* **Удобство использования**: Обеспечение плавные анимации, удобные формы ввода, автозаполнение, подсказки.
* **Обратная связь**: Уведомления об ошибках или успехах. Например, когда задача создана или комментарий отправлен — показывай визуальный фидбек.
* **Темная тема**: Дай пользователям возможность переключаться между светлой и тёмной темой.

**10. Оптимизация и производительность**

* **Загрузка данных**: Ленивая загрузка — не грузить всё сразу, а подгружать данные по мере прокрутки (infinite scrolling).
* **Минимизация запросов к серверу**: Оптимизировать работу с API, чтобы не делать излишних запросов. Например, все изменения в задачах/проектах должны быть отправлены за один запрос, а не раздельно.

**11. Тестирование**

* **UI тесты**: Написание тесты для всех форм и элементов на фронтенде.
* **Интеграционные тесты**: Проверять работу всех функций вместе (например, создание задачи, добавление комментариев, уведомления).

**12. Безопасность**

* **Скрипты**: Очищать все данные, приходящие от пользователя (XSS защита).
* **CORS**: Правильная настройка CORS на сервере, чтобы не было дыр в безопасности при запросах с фронтенда.

**13. Интернационализация**

* Если в будущем планируется поддержка нескольких языков, стоит сразу предусмотреть возможность локализации интерфейса.