

Техническое задание

Проект: Система управления заявками (Service Desk / HelpDesk)

1. Цель работы

Разработать веб-приложение для регистрации, обработки и контроля заявок пользователей (аналог внутреннего HelpDesk): создание заявок, назначение ответственных, статусы, комментарии и история изменений.

2. Роли пользователей

- Заявитель: создаёт заявку, отслеживает статус, добавляет комментарии/файлы.
- Исполнитель (специалист поддержки): берёт заявку в работу, отвечает, меняет статусы.
- Администратор: управляет пользователями, категориями, правами, смотрит отчёты.

3. Основные сущности

- Заявка: номер, тема, описание, категория, приоритет, статус, заявитель, исполнитель, даты (создана/обновлена/закрыта).
- Комментарий: текст, автор, дата (возможность прикрепить файл — опционально).
- Категория: тип заявки (IT, бухгалтерия, пропуск, оборудование и т.п.).
- Приоритет: низкий/средний/высокий/критический.
- Журнал изменений (audit log): фиксация смены статусов/исполнителя/приоритета.

4. Функциональные требования (заявитель)

- Регистрация/вход.
- Создание заявки через форму (категория, тема, описание, приоритет).
- Просмотр списка своих заявок с фильтрами (статус, период).
- Просмотр карточки заявки (детали + переписка/комментарии).
- Добавление комментариев в заявку.
- Получение уведомлений о смене статуса (в интерфейсе; email — опционально).

5. Функциональные требования (исполнитель)

- Просмотр очереди заявок (новые/в работе/ожидают).
- Взять заявку в работу (назначить себя) или назначить другого исполнителя (если есть права).

- Изменение статуса заявки (Новая → В работе → Ожидает ответа → Решена/Закрыта).
- Добавление комментариев/ответов.
- Фильтры и поиск по заявкам (категория, приоритет, статус, заявитель).

6. Функциональные требования (администратор)

- CRUD пользователей и ролей.
- CRUD категорий заявок и приоритетов/статусов (можно фиксированный список статусов).
- Просмотр всех заявок и отчётов: количество заявок по статусам/категориям, среднее время закрытия.
- Просмотр журнала действий (audit log).

7. Статусы заявок (минимум)

- Новая
- В работе
- Ожидает ответа
- Решена
- Закрыта

8. Интерфейс (UI/UX)

- 3 основных раздела: «Мои заявки» (заявитель), «Очередь» (исполнитель), «Админка» (админ).
- Карточка заявки должна содержать: заголовок, статус, приоритет, категорию, исполнителя, таймлайн/историю, комментарии.
- Адаптивность: Desktop/Tablet/Mobile.

9. База данных (MySQL, минимум)

- users: id, name, email/login, password_hash, role, created_at.
- categories: id, name.
- tickets: id, ticket_no, title, description, category_id, priority, status, creator_id, assignee_id, created_at, updated_at, closed_at.
- comments: id, ticket_id, author_id, text, attachment_url (опц.), created_at.
- audit_log: id, ticket_id, action, before_json, after_json, author_id, created_at.

10. Технические требования

- Frontend: React (рекомендуется) или HTML/CSS/JS.
- Backend: Node.js или PHP.
- База данных: MySQL.

- REST API для работы интерфейсов.
- Валидация данных и обработка ошибок.
- README: запуск проекта, миграции/создание таблиц, тестовые пользователи.

11. Ограничения

- SLA/автоматическая эскалация — не требуется.
- Интеграции с почтой/Telegram — опционально.
- Достаточно одного проекта/организации (multi-tenant не нужен).

12. Результат работы

- Рабочее приложение: создание заявки → назначение исполнителя → переписка → смена статусов → закрытие.
- Демонстрация минимум 10 тестовых заявок в базе.
- Исходный код и база данных.

13. Критерии оценки

- Корректная логика статусов и прав доступа.
- Удобная карточка заявки и фильтры.
- Наличие истории изменений (audit log).
- Аккуратность UI и адаптивность.
- Качество структуры проекта и документации (README).