Носова Виктория ИСП-311

Тема: Система управления заявками службы технической поддержки

Цель разработки: Обеспечить автоматизацию процесса приема, распределения, выполнения и контроля заявок пользователей.

Функциональные требования

1. Регистрация и управление пользователями:

* Регистрация пользователей через веб-интерфейс
* Авторизация с использованием email и пароля
* Роли пользователей:
  + Пользователь: создает заявки
  + Оператор: обрабатывает заявки
  + Администратор: управляет пользователями и контролирует работу системы

1. Работа с заявками:

* Создание новой заявки:
  + Поля:
    - Тема заявки
    - Подробное описание
    - Приоритет (низкий, средний, высокий)
    - Категория (например, «техническая проблема», «сетевая ошибка»)
    - Вложения (файлы до 20 МБ)
* Изменение статуса заявки («Новая», «В работе», «На проверке», «Закрыта», «Отменена»)
* Возможность добавления комментариев оператором и пользователем
* Автоматическое уведомление о смене статуса заявки

1. Управление исполнителями:

* Назначение исполнителей на заявки:
  + Автоматически, на основе загрузки сотрудников
  + Вручную, оператором или администратором

1. Уведомления:

* Email-уведомления для пользователей и сотрудников:
  + Подтверждение регистрации
  + Изменение статуса заявки
* Push-уведомления в системе для внутренних пользователей

1. Отчеты и аналитика:

* Автоматическая генерация отчетов:
  + Количество выполненных заявок за период
  + Среднее время обработки заявок
  + Загруженность исполнителей
* Возможность экспорта отчетов в PDF или Excel

1. Технические требования:

* Поддержка всех популярных браузеров (Chrome, Firefox, Safari, Edge)
* Респонсивный дизайн для работы с мобильных устройств

Характеристики работ и технологии

1. Frontend (интерфейс):

* Разработка интерфейса на React.js или Vue.js
* Использование HTML, CSS (SCSS) и JavaScript для стилизации и логики

1. Backend (серверная часть):

* Реализация серверной логики на Node.js или Django (Python)
* REST API для взаимодействия с фронтендом

1. База данных:

* PostgreSQL для хранения данных о пользователях, заявках и отчетах
* Redis для кеширования уведомлений и повышения производительности

1. Инфраструктура:

* Использование Docker для контейнеризации
* Размещение на облачных платформах (AWS, Google Cloud или Azure)

1. Безопасность:

* SSL для шифрования передачи данных
* Хранение паролей с использованием bcrypt

1. Документация:

* Руководство пользователя
* API-документация

Сроки выполнения

|  |  |  |
| --- | --- | --- |
| Этап | Описание работ | Срок выполнения |
| 1. Анализ требований | Сбор и анализ требований заказчика | 2 недели |
| 2. Проектирование | Разработка архитектуры системы, ER-диаграмм | 2 недели |
| 3. Разработка прототипа | Создание прототипа интерфейса | 3 недели |
| 4. Backend-разработка | Реализация серверной части и API | 4 недели |
| 5. Интеграция | Связывание фронтенда и бэкенда | 3 недели |
| 6. Тестирование | Функциональное и нагрузочное тестирование | 2 недели |
| 7. Развертывание | Установка на сервер, подготовка инфраструктуры | 1 неделя |
| 8. Документация | Подготовка руководств и обучающих материалов | 1 неделя |

Общий срок выполнения работ: 18 недель (4,5 месяца).



