Техническое Задание

Разработка Веб-приложения "Система управления больницей"

1. Введение

1.1 Цель документа

Этот документ описывает требования к разработке веб-приложения "Система управления больницей", предназначенного для автоматизации процессов записи пациентов на прием, управления данными врачей и администраторов, а также обработки медицинских данных.

1.2 Актуальность и назначение проекта

Современные больницы сталкиваются с проблемами координации пациентов, записи на прием, управления расписаниями врачей и обработки медицинских данных. Целью системы является автоматизация этих процессов для повышения качества обслуживания пациентов и эффективности работы медицинских учреждений.

1.3 Основные пользователи системы

Пациенты: запись на прием, просмотр доступных врачей, управление персональными данными.

Врачи: доступ к карточкам пациентов, управление статусом записей.

Администраторы: управление пользователями, врачами, направлениями и расписанием.

2. Цели и задачи проекта

2.1 Основные цели

Автоматизация записи пациентов.

Централизованное хранение данных пациентов, врачей и записей.

Обеспечение безопасного и удобного доступа к данным.

2.2 Конкретные задачи

Разработка функциональности для записи пациентов на прием и управления записями.

Реализация модулей управления врачами и направлениями для администраторов.

Организация доступа врачей к карточкам пациентов.

Разработка функционала редактирования данных и создания отчетов.

3. Требования к системе

3.1 Функциональные требования

Для пациентов:

Регистрация и вход в систему.

Просмотр свободного времени врачей для записи.

Запись на прием и управление записями.

Редактирование личных данных.

Для врачей:

Просмотр и изменение статуса записей пациентов.

Доступ к карточкам пациентов.

Для администраторов:

Управление врачами, направлениями и специализациями.

Удаление и редактирование данных о пользователях.

Просмотр и управление записями пациентов.

3.2 Нефункциональные требования

Производительность: Отклик интерфейса не более 2 секунд.

Масштабируемость: Поддержка до 10 000 пользователей.

Безопасность: Шифрование данных через HTTPS, защита от SQL-инъекций.

4. Требования к пользовательскому интерфейсу

4.1 Основные экраны

Для пациентов:

Главная страница с функционалом записи и списком доступных врачей.

Личный кабинет с историей записей.

Для врачей:

Панель управления записями.

Просмотр и редактирование данных пациентов.

Для администраторов:

Управление пользователями, врачами и направлениями.

Панель для анализа данных.

4.2 Навигация

Простое и понятное меню с основными разделами: "Мои записи", "Врачи", "Администрирование", "Личный кабинет".

4.3 Юзабилити

Цветовое кодирование статусов записей: зеленый — подтверждено, желтый — ожидает подтверждения, красный — отменено.

5. Требования к технической реализации

5.1 Языки и технологии

Backend: Node JS.

Frontend: React.

База данных: phpmyadmin.

5.2 Архитектура системы

Микросервисная архитектура.

REST API для взаимодействия между модулями.

5.3 Интеграции

Подключение email-уведомлений для подтверждения записей.

Интеграция с системами авторизации OAuth.

6. Требования к безопасности

6.1 Аутентификация и авторизация

Разделение прав доступа: пациенты, врачи, администраторы.

Защита персональных данных через шифрование.

6.2 Политики доступа

Ограничение прав доступа к данным пациентов только для авторизованных врачей.

7. Ограничения и допущения

Система должна поддерживать мобильные устройства.

В рамках бюджета используются только open-source решения.

8. Требования к тестированию и приемке

8.1 Типы тестирования

Функциональное: проверка записи на прием, управления данными.

Нагрузочное: тестирование работы при 10 000 пользователях.

8.2 Критерии приемки

Полная реализация функциональных требований.

Успешное прохождение не менее 95% тестов.