1. Перечень технологий, используемых для реализации варианта задания

• Frontend:

• HTML/CSS

• JavaScript (React или Angular)

• Bootstrap (или другой CSS-фреймворк для стилизации)

• Backend:

• Node.js с Express (или Python с Flask/Django)

• RESTful API

• База данных:

• PostgreSQL или MySQL

• Инструменты для разработки:

• Git (для контроля версий)

• Docker (для контейнеризации и развертывания приложения)

• Тестирование:

• Jest (для тестирования JavaScript)

• Postman (для тестирования API)

2. Обоснование выбора технологий и удобство их использования

• Frontend:

• React/Angular: Оба фреймворка широко используются, имеют большое сообщество и множество готовых компонентов, что ускоряет разработку.

• HTML/CSS: Стандартные технологии для создания веб-интерфейсов.

• Bootstrap: Упрощает стилизацию и адаптивность интерфейса.

• Backend:

• Node.js с Express: Высокая производительность и возможность написания кода на JavaScript как на клиенте, так и на сервере.

• RESTful API: Позволяет организовать взаимодействие между клиентом и сервером через стандартные HTTP-запросы.

• База данных:

• PostgreSQL/MySQL: Надежные реляционные базы данных с поддержкой сложных запросов и транзакций.

• Инструменты для разработки:

• Git: Позволяет отслеживать изменения в коде, работать в команде и управлять версиями.

• Docker: Обеспечивает изоляцию окружения, что упрощает развертывание и масштабирование.

3. Декомпозиция разрабатываемой системы

1. Модуль регистрации и авторизации:

• Регистрация пользователей

• Авторизация

• Восстановление пароля

2. Модуль поиска врачей:

• Поиск по специализациям

• Фильтрация по местоположению

• Отображение профиля врача

3. Модуль записи на прием:

• Выбор времени

• Подтверждение записи

• Уведомления о записи

4. Модуль управления записями:

• Просмотр записей

• Отмена/перенос записей

• Напоминания о визитах

5. Модуль обратной связи:

• Оставление отзывов

• Оценка врачей

6. Модуль статистики и аналитики:

• Сбор данных о записях

• Генерация отчетов

▎4. Набор задач для реализации ПО

1. Анализ требований (приоритет: высокий)

2. Проектирование архитектуры приложения (приоритет: высокий)

3. Разработка модуля регистрации и авторизации (приоритет: высокий)

4. Создание базы данных и её структуры (приоритет: высокий)

5. Разработка модуля поиска врачей (приоритет: средний)

6. Разработка модуля записи на прием (приоритет: высокий)

7. Создание модуля управления записями (приоритет: средний)

8. Разработка модуля обратной связи (приоритет: низкий)

9. Создание модуля статистики и аналитики (приоритет: низкий)

10. Тестирование приложения (приоритет: высокий)

11. Подготовка к развертыванию и настройка сервера (приоритет: средний)