**Тема**

**Автосервис. Управление нарядами, ремонтом и обслуживанием автомобилей.**

**Планируемый стек разработки**

**Клиентская часть:**

* HTML/CSS — для создания разметки и стилей страницы.
* JavaScript — для взаимодействия с сервером через REST API, обработки событий, работы с UI.

**Серверная часть:**

* Java — основной язык для серверной логики.
* MySQL — для хранения данных.
* JDBC — для работы с базой данных.
* **Maven** — для сборки и управления зависимостями
* Spring – для разработки REST API

**Наполнение проекта**

Роли:

1. **клиент**
2. **механик**

Описание серверной части

**Для клиента:**

1. Регистрация и авторизация.
2. Добавление автомобиля.
3. Создание наряда (выбор услуг и привязка автомобиля).
4. Просмотр списка нарядов и статуса.
5. Просмотр детализации наряда.

#### **Для механика:**

1. Авторизация.
2. Просмотр списка нарядов.
3. Изменение статуса наряда (в процессе, завершен и т.д.).
4. Добавление услуг в наряд.

#### **Структура базы данных:**

* 1. **Пользователи (users):**
* id
* имя
* email
* роль (клиент/механик)
* пароль (захеширован)
  1. **Автомобили (cars):**
* id
* марка
* модель
* год выпуска
* номерной знак
* владелец (ссылка на users)
  1. **Наряды (orders):**
* id
* статус (в процессе, завершен, отменен)
* дата создания
* дата завершения
* клиент (ссылка на users)
* механик (ссылка на users)
* автомобиль (ссылка на cars)
  1. **Услуги (services):**
* id
* название услуги
* описание
* стоимость
  1. **Наряд-Услуга (order\_services):**
* id
* наряд (ссылка на orders)
* услуга (ссылка на services)
* количество

#### **Описание клиентской части**

#### **1. Авторизация и регистрация**

* Форма логина и регистрации.
* Валидация данных (на клиенте и сервере).

#### 2. **Кабинет клиента**

* **Список автомобилей:** С возможностью добавлять/удалять.
* **Создание наряда:**
  + Выбор автомобиля.
  + Список услуг с ценами.
  + Отправка на сервер.
* **История нарядов:**
  + Таблица с датами, статусом и ссылкой на детали.

#### 3. **Кабинет механика**

* **Список нарядов:** Фильтрация по статусу (новые, в процессе, завершенные).
* **Обновление статуса:** Возможность менять статус наряда.
* **Добавление услуг:** К существующему наряду.

**Этапы разработки**

#### 4.1. **Бэкенд**

1. **Создать REST API:**
   * Разработать контроллеры для обработки запросов.
   * Написать сервисы и DAO для работы с базой данных.
   * Подключить JWT для авторизации и аутентификации.
2. **Тестировать каждый API-эндпоинт:**
   * Использовать Postman/Swagger для проверки запросов.

#### 4.2. **Фронтэнд**

1. **Создать пользовательский интерфейс:**
   * Разметить основные страницы (авторизация, кабинет клиента/механика, создание наряда).
   * Стилизовать элементы интерфейса с использованием CSS/Bootstrap.
2. **Подключить к серверу:**
   * Настроить fetch-запросы для взаимодействия с REST API.
   * Реализовать обработку ответов от сервера (успех/ошибка).
3. **Добавить динамику:**
   * Сделать формы интерактивными (валидация, обработка ошибок).
   * Обновлять данные на странице без перезагрузки.

4.3. Написать Unit-тесты