**Twili: Z3RMULL41KX2KGCECW9P6YJ5**

**2. Функциональные требования**

**2.1 Пользователи и авторизация**

* **Роли:**
  + Клиент (запись на мойку, оплата, история посещений).
  + Администратор (управление заказами, сотрудниками, аналитика).
  + Сотрудник (выполнение заказов, отметка о завершении).
* **Функции:**
  + Регистрация (email/телефон, пароль).
  + Авторизация (JWT-токен).
  + Восстановление пароля.
  + Редактирование профиля.

**2.2 Управление записями на автомойку**

* Клиент может выбрать время, дату и тип мойки.
* Админ может просматривать, редактировать и отменять бронирования.
* Сотрудник может видеть список активных записей.
* Напоминания клиенту о записи (SMS/email).

**2.3 Онлайн-оплата**

* Подключение платежного шлюза (Stripe/LiqPay/PayPal/Kaspi).
* Оплата через сайт.
* Выставление электронных чеков.

**2.4 Программа лояльности**

* Кэшбэк или бонусные баллы за посещения.
* Реферальная система (клиент получает скидку за приглашенного друга).

**2.5 CRM и маркетинг**

* Хранение базы клиентов.
* История заказов и предпочтений клиента.
* Автоматическая рассылка акций и скидок.
* Интеграция с Meta Pixel / Google Analytics.

**2.6 Управление услугами и ценами**

* Админ может добавлять/редактировать услуги.
* Возможность устанавливать динамические скидки (например, на утреннее время).

**2.7 Аналитика и отчёты**

* Статистика по заказам, доходу, загруженности.
* Выгрузка данных в Excel/PDF.

**3. API (основные эндпоинты)**

| **Метод** | **URL** | **Описание** | **Доступ** |
| --- | --- | --- | --- |
| POST | /api/auth/register | Регистрация пользователя | Все |
| POST | /api/auth/login | Авторизация | Все |
| GET | /api/user/profile | Получение профиля | Авторизованные |
| POST | /api/bookings | Запись на мойку | Клиент |
| GET | /api/bookings | Список записей | Админ, Сотрудник |
| POST | /api/payments | Оплата заказа | Клиент |
| GET | /api/loyalty | Программа лояльности | Клиент |
| POST | /api/marketing/send | Запуск рекламной рассылки | Админ |
| GET | /api/analytics | Получение отчётов | Админ |

**4. База данных (основные таблицы)**

**users**

| **id** | **name** | **email** | **phone** | **role** | **password** |
| --- | --- | --- | --- | --- | --- |
| 1 | Иван | ivan@mail.com | +77011234567 | client | hash |

**bookings**

| **id** | **user\_id** | **date** | **time** | **service\_id** | **status** |
| --- | --- | --- | --- | --- | --- |
| 1 | 1 | 2025-03-09 | 14:00 | 2 | pending |

**services**

| **id** | **name** | **price** | **duration** |
| --- | --- | --- | --- |
| 1 | Комплекс | 5000 | 60 мин |

**payments**

| **id** | **user\_id** | **amount** | **status** | **payment\_method** |
| --- | --- | --- | --- | --- |
| 1 | 1 | 5000 | success | Kaspi |

**5. Не-функциональные требования**

* **Безопасность:** Защита JWT-токенов, шифрование паролей (bcrypt).
* **Производительность:** Кеширование популярных запросов через Redis.
* **Логирование:** Запись ошибок и действий пользователей в файлы логов.

Начнём с клиентской части. Я напишу описание API, базу данных и ключевые функциональные моменты для клиента, а потом перейдём к другим ролям.

**Клиентская часть (Client)**

**1. Функции пользователя (Клиент)**

* **Регистрация и авторизация**
  + Регистрация через email или телефон (с верификацией).
  + Авторизация через JWT.
  + Восстановление пароля через email/SMS.
  + Редактирование профиля (изменение email, телефона, пароля).
* **Запись на мойку**
  + Выбор даты и времени.
  + Выбор типа услуги.
  + Получение уведомлений (SMS/email) о подтверждении записи.
  + Просмотр истории записей.
* **Оплата**
  + Онлайн-оплата через Kaspi/Stripe/PayPal/LiqPay.
  + Выставление электронных чеков.
* **Программа лояльности**
  + Начисление бонусов за каждую мойку.
  + Реферальная система (пригласил друга → получил скидку).
* **История заказов**
  + Просмотр завершённых и предстоящих записей.
  + Отзывы о посещении.

**Техническое задание (ТЗ) на разработку веб-приложения для автомойки**

**1. Описание проекта**

Веб-приложение предназначено для записи клиентов на автомойку, автоматизации работы сотрудников и администраторов, а также управления платежами, лояльностью и маркетингом.

**2. Функциональные требования**

**2.1 Роли пользователей**

* **Клиент**
  + Запись на мойку (выбор времени, даты, услуги).
  + Оплата заказа.
  + Просмотр истории посещений.
  + Получение уведомлений о записи.
  + Участие в программе лояльности.
  + Оценка качества обслуживания.
* **Администратор**
  + Управление записями (просмотр, редактирование, отмена).
  + Управление сотрудниками (назначение смен, распределение заказов).
  + Настройка списка услуг и цен.
  + Аналитика и отчёты.
  + Управление маркетингом (SMS/email-рассылки, акции).
* **Сотрудник**
  + Просмотр списка активных записей.
  + Отметка о начале и завершении мойки.
  + Получение уведомлений о новых заказах.

**2.2 Запись на автомойку**

* Клиент выбирает **дату, время и тип мойки**.
* Клиенту приходит **подтверждение (SMS/email/уведомление в ЛК)**.
* В день записи клиент может **подтвердить прибытие** (QR-код/пин-код).
* Система отображает **очередь** сотрудников и их статус.
* Администратор может **просматривать, редактировать и отменять записи**.

**2.3 Оплата**

* Подключение платежного шлюза (Stripe, LiqPay, PayPal, Kaspi).
* Оплата онлайн через сайт.
* Выставление электронных чеков.

**2.4 Очередь клиентов**

* Клиенты с записью имеют **приоритет** перед незаписанными.
* Если клиент прибыл, но **опаздывает**, система может предложить слот другому.
* Живая очередь для незаписанных клиентов (если есть свободные окна).
* **Статусы заказов**: "Ожидание", "Прибыл", "В работе", "Завершено".

**2.5 CRM и маркетинг**

* Хранение базы клиентов.
* История заказов и предпочтений.
* Автоматическая рассылка акций и скидок.
* Интеграция с Meta Pixel / Google Analytics.

**2.6 Программа лояльности**

* Кэшбэк или бонусные баллы за посещения.
* Реферальная система (скидка за приглашенного друга).

**2.7 Управление услугами и ценами**

* Админ может добавлять/редактировать услуги.
* Динамические скидки (например, утренние скидки).

**2.8 Аналитика и отчёты**

* Статистика по заказам, доходу, загруженности.
* Выгрузка данных в Excel/PDF.

**3. API (основные эндпоинты)**

| **Метод** | **URL** | **Описание** | **Доступ** |
| --- | --- | --- | --- |
| **POST** | /api/auth/register | Регистрация пользователя | Все |
| **POST** | /api/auth/login | Авторизация (JWT) | Все |
| **GET** | /api/user/profile | Получение профиля | Авторизованные |
| **POST** | /api/bookings | Запись на мойку | Клиент |
| **GET** | /api/bookings | Список записей | Админ, Сотрудник |
| **POST** | /api/payments | Оплата заказа | Клиент |
| **GET** | /api/loyalty | Программа лояльности | Клиент |
| **POST** | /api/marketing/send | Запуск рекламной рассылки | Админ |
| **GET** | /api/analytics | Получение отчётов | Админ |

**4. База данных (основные таблицы)**

**Таблица users (пользователи)**

| **id** | **name** | **email** | **phone** | **role** | **password** |
| --- | --- | --- | --- | --- | --- |
| 1 | Иван | ivan@mail.com | +77011234567 | client | hash |

**Таблица bookings (записи на мойку)**

| **id** | **user\_id** | **date** | **time** | **service\_id** | **status** |
| --- | --- | --- | --- | --- | --- |
| 1 | 1 | 2025-03-09 | 14:00 | 2 | pending |

**Таблица services (услуги)**

| **id** | **name** | **price** | **duration** |
| --- | --- | --- | --- |
| 1 | Комплекс | 5000 | 60 мин |

**Таблица payments (платежи)**

| **id** | **user\_id** | **amount** | **status** | **payment\_method** |
| --- | --- | --- | --- | --- |
| 1 | 1 | 5000 | success | Kaspi |

В целом, твой routes/web.php уже выглядит хорошо:  
✔ Есть авторизация (JWT)  
✔ Есть работа с пользователями (регистрация, редактирование)  
✔ Есть бронирование  
✔ Есть история заказов  
✔ Есть админка и сотрудники

Но для дипломки, особенно если хочешь сделать её крутой, я бы посмотрел на:

**1️⃣ Роли и права доступа (RBAC)**

Сейчас ты проверяешь админов через check.admin, но можно добавить роли (Admin, User, Employee) и **гибкую систему прав**. Например:

* **Админы** могут управлять всеми пользователями, услугами, бронированиями
* **Сотрудники** могут только видеть и выполнять заказы
* **Пользователи** могут бронировать, но не могут редактировать чужие записи

Можно использовать Spatie Laravel Permissions — это гибкий пакет для ролей.

**2️⃣ CRUD для услуг и категорий**

Ты можешь добавлять категории и услуги **только вручную через БД**?  
Лучше добавить CRUD для Admin:  
✔ Создавать / редактировать / удалять услуги  
✔ Управлять категориями

**3️⃣ Фильтры и поиск**

Сейчас у тебя есть filter-with-name, но:

* Можно добавить **поиск по категориям, цене, названию**
* **Сортировку** (по рейтингу, цене, популярности)
* **Пагинацию**, если услуг много

Используй Eloquent Scopes или [Laravel Scout](https://laravel.com/docs/11.x/scout) для быстрого поиска.

**4️⃣ Уведомления (Notifications)**

📩 **E-mail / Telegram / SMS** пользователю о:  
✔ Успешном бронировании  
✔ Напоминании перед визитом  
✔ Акциях и промо

Можно использовать [Laravel Notifications](https://laravel.com/docs/11.x/notifications) или интегрировать Twilio/SMS-рассылки.

**5️⃣ Статусы заказов (Enum)**

Сейчас у тебя только confirm / cancel, но можно сделать:

* pending (ожидает)
* confirmed (подтвержден)
* in\_progress (выполняется)
* completed (завершен)
* canceled (отменен)

Так будет легче отслеживать процесс заказа.

**6️⃣ Отзывы и рейтинг**

После завершения мойки клиент может оставить **отзыв и оценку (1-5 звёзд)**.  
✔ Средний рейтинг будет показываться в списке услуг  
✔ Это добавит интерактивности и доверия к сервису

💡 **Вывод**  
Если просто сдать дипломку – то текущего кода хватит.  
Но если хочешь, чтобы она была **крутой**, то добавь **роли, фильтры, уведомления и отзывы**.

Что думаешь? 💡