ChatGPT

Структура проекта «Barber PRESTIGE»

Проект структурирован в виде отдельного frontend-приложения с учётом параллакса, 3D-элементов и фронтенд-бронирования. Общая структура каталогов:

```
BarberPRESTIGE/
├── index.html
├── css/
│   ├── style.css
│   ├── animations.css
│   └── booking.css
├── js/
│   ├── main.js
│   ├── scroll.js
│   └── gsap-plugins.js
├── images/
│   ├── hero-bg.jpg
│   ├── layers/
│   │   ├── layer1.png
│   │   └── layer2.png
│   ├── gallery/
│   │   ├── salon1.jpg
│   │   └── salon2.jpg
│   └── masters/
│       ├── master1.jpg
│       └── master2.jpg
├── fonts/
│   ├── YatraOne-Regular.woff2
│   ├── Cinzel-Regular.woff2
│   └── CormorantGaramond-Regular.woff2
├── booking/
│   ├── index.html
│   ├── booking.js
│   ├── calendar.js
│   └── queue.js
└── videos/
    ├── hero-bg.mp4
    └── gallery1.webm
```

HTML (index.html и booking/index.html)

- **index.html** — главная страница со всеми разделами: «hero», услуги, галерея, отзывы и т.д. Включает ссылки на CSS и JS.
- **booking/index.html** — отдельная страница (или модуль) с формой бронирования. Используется при клике «Записаться».

- Общий подход — SPA-подобная навигация. Главная страница содержит кнопку/ссылку, открывающую раздел бронирования (можно через модальное окно или отдельный html).

Пример (index.html):

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Barber PRESTIGE</title>
  <link rel="stylesheet" href="css/style.css">
  <link rel="stylesheet" href="css/animations.css">
  <link rel="stylesheet" href="css/booking.css">
</head>
<body>
  <div class="hero-section">
    <!-- Видео-фон -->
    <video class="hero-video" src="videos/hero-bg.mp4" autoplay loop muted></
video>
    <div class="hero-content">
      <h1>Barber <span>PRESTIGE</span></h1>
      <p>Высокая культура стрижки</p>
      <button onclick="openBooking()">Записаться</button>
    </div>
    <!-- 3D Canvas -->
    <canvas id="heroCanvas"></canvas>
  </div>
  <!-- Блок услуг с parallax-эффектами -->
  <section id="services">
    <!-- ...параллакс-слои, текст, иконки -->
  </section>
  <!-- Галерея -->
  <section id="gallery">
    <!-- Изображения высокого разрешения -->
  </section>
  <!-- … другие секции … -->
  <script src="js/gsap-plugins.js"></script>
  <script src="js/scroll.js"></script>
  <script src="js/main.js"></script>
</body>
</html>
```

CSS (/css)

- **style.css** — основные стили: шрифты, цветовая схема (черно-золотая), базовая сетка и расположение элементов. Применяются CSS-переменные для палитры (`--gold: #d4af37; --black: #000;` и т.д.).
- **animations.css** — стили для анимаций: keyframes для SVG-анимаций, 3D-трансформаций. Также стили параллакса: позиционирование слоёв с `position: absolute; z-index`.

- **booking.css** — стили для формы бронирования: календаря, полей ввода, списка очереди. Используются псевдоэлементы для эффектов иконок.
- Используем Google Fonts или локальные файлы из `/fonts/` с `@font-face` для премиальных шрифтов.

Пример (css/style.css):

```
:root {
  --gold: #d4af37;
  --black: #000;
  --glass: rgba(255, 255, 255, 0.1);
}
body {
  margin: 0;
  background: var(--black);
  color: var(--gold);
  font-family: 'Cinzel', serif;
}
h1 span {
  color: var(--gold);
  text-shadow: 0 0 10px rgba(212,175,55,0.8);
}
/* Параллакс-слои */
.parallax-layer {
  position: absolute;
  width: 100%;
  height: 100%;
}
```

JavaScript (/js)

- **gsap-plugins.js** — подключение библиотек: GSAP (ScrollTrigger, SplitText), Locomotive Scroll. Это отдельный бандл или CDN-ссылки.
- **scroll.js** — инициализация scroll-driven эффектов. Например:

```
gsap.registerPlugin(ScrollTrigger);
const locoScroll = new LocomotiveScroll({ el: document.body, smooth:
true });
locoScroll.on('scroll', ScrollTrigger.update);
ScrollTrigger.scrollerProxy(document.body, {
  scrollTop(value) {
    return arguments.length ? locoScroll.scrollTo(value, 0, 0) :
locoScroll.scroll.instance.scroll.y;
  },
  getBoundingClientRect() {
    return { top: 0, left: 0, width: window.innerWidth, height:
window.innerHeight };
  }
});
```

```
ScrollTrigger.addEventListener("refresh", () => locoScroll.update());
ScrollTrigger.refresh();
```

- **main.js** — основная логика сайта: запуск 3D-сцены (Three.js), обработка кликов, открытие формы бронирования (e.g. функция `openBooking()`), плавная навигация.
- В `main.js` добавлены эффекты GSAP для отдельных секций. Например:

```
gsap.to(".hero-section h1", {
  y: -50,
  scrollTrigger: {
    trigger: ".hero-section",
    start: "top top",
    scrub: true
  }
});
```

- Раздельный скрипт **calendar.js** (в папке /booking) реализует календарь (можно использовать HTML `<input type="date">` и `<input type="time">` или стороннюю библиотеку flatpickr).
- **queue.js** управляет локальным списком бронирования: при подтверждении записи данные (дата, время, имя) сохраняются в `localStorage`, отображается номер в очереди (количество записей + 1). При загрузке формы подсчитывается текущий размер очереди.

Пример функции записи (js/booking.js):

```
function bookAppointment(formData) {
  let queue = JSON.parse(localStorage.getItem('bookingQueue') || '[]');
  const record = { date: formData.date, time: formData.time, name:
formData.name };
  queue.push(record);
  localStorage.setItem('bookingQueue', JSON.stringify(queue));
  alert(`Ваша запись подтверждена. Номер в очереди: ${queue.length}`);
}
```

Папка /booking

- **booking/index.html** — отдельный HTML-файл (или модальный шаблон) с формой: поля «Имя», «Телефон», выбор даты/времени. Подключает `booking.js`, `calendar.js`, `booking.css`.
- Календарь: выбираем дату, появление доступных слотов (можно закладывать, что слоты генерируются скриптом). Формы с `required` и проверкой формата (pattern).
- При отправке формы вызывается JS-функция (из примера выше) для сохранения в `localStorage`.

Пример (booking/index.html):

```
<form id="bookingForm">
  <label>Имя: <input type="text" name="name" required></label>
```

```
  <label>Телефон: <input type="tel" name="phone" required></label>
  <label>Дата: <input type="date" name="date" id="datePicker" required></
label>
  <label>Время:
    <select name="time" id="timePicker" required>
      <option>10:00</option>
      <option>12:00</option>
      <!-- ... -->
    </select>
  </label>
  <button type="submit">Записаться</button>
</form>
<script src="booking/booking.js"></script>
```

Изображения (/images) и видео (/videos)

- Используются высококачественные изображения (разрешение 2–4K) в фонах и галерее. Например, 5–10 фонов по 10–20 МБ каждый (до ~200 МБ).
- Параллакс-слои (`/images/layers/`) — PNG с прозрачным фоном, детали (расстрижка, текстуры). Несколько слоёв для каждого сектора.
- Фото мастеров (`/images/masters/`) и салона (`/images/gallery/`) — по несколько изображений 4K (~15–30 МБ каждое, общее до ~200 МБ).
- Видео-фон в hero (`/videos/hero-bg.mp4` и/или webm) — 4K видеоролик о барбершопе, около 100–200 МБ (WebM и MP4-копии).
- Видеофайлы вебмастера могут занимать до ~300 МБ.
- Объём изображений + видео направлен на итоговую ~1 ГБ (например, 600–700 МБ медиа, остальное – шрифты, скрипты).

Шрифты (/fonts)

- Выбранные лицензированные шрифты: **Yatra One**, **Cinzel**, **Cormorant Garamond** (подчёркивают премиальность). Форматы WOFF2 (легковеснее) и WOFF.
- Каждого шрифта (несколько начертаний) по ~1–3 МБ. Итого шрифты около 10–15 МБ.
- Подключение через `@font-face` в CSS:

```
@font-face {
  font-family: 'YatraOne';
  src: url('../fonts/YatraOne-Regular.woff2') format('woff2');
}
h1 { font-family: 'YatraOne', serif; }
```

3D и SVG-анимации

- **Three.js** — подключается для 3D-элементов на холсте (`heroCanvas`). Например, модель бритвы или ножниц, которые медленно вращаются или реагируют на скролл. Код и шейдеры лежат в `js/threejs/` .
- Если не полноценный 3D, можно использовать CSS3-трансформации для эффекта глубины: например, текст с `transform: translateZ()` в параллаксе.

- **SVG-анимации** — декоративные элементы (узоры, иконки) анимируются через SMIL или CSS анимации (клевающая кисть, движущиеся узоры на фоне).
- Эти эффекты улучшают премиальный вид (стеклянные панели с полупрозрачностью, блики, тени).

Архитектурные особенности

- **SPA-подобная организация**: основная страница грузится однажды, подгружаются секции по скроллу или при кликах (для плавности UX).
- **Разделение ответственности**: отдельные модули для анимаций, логики скролла и бронирования. Это облегчает поддержку.
- **Без серверной части**: форма бронирования полностью на фронтенде, данные хранятся в `localStorage`. Можно добавить экспорт/импорт или уведомления по email при необходимости.
- **Параллакс и скролл**: обеспечивает эффект глубины. Используем `position: fixed/absolute`, gsap ScrollTrigger и Locomotive для синхронизации.
- **Детализация стиля**: много теней и блёсток (черное с золотом), стеклянные (glassmorphism) панели меню.

Примерные размеры ресурсов

- **Видео-фон (hero-bg.mp4/webm)**: ~150–300 МБ (4K, компрессия H.264/H.265 или WebM VP9).
- **Изображения фонов/галереи**: 5–10 изображений @4K, каждая 15–20 МБ → 300–600 МБ.
- **PNG-параллакс-слои**: по нескольку слоёв (общий вес ~50–100 МБ).
- **Шрифты**: ~10–15 МБ.
- **JS/CSS**: скрипты GSAP, Three.js, анимации – ~1–2 МБ (без учёта того, что GSAP обычно подключают с CDN).
- **Итого**: около **1 ГБ** (большая часть — графика и видео). Конечно, для реального сайта стоит оптимизировать, но по заданию акцент на «премиальный вес».

Все элементы структуры организованы по папкам для простого обслуживания. Главный файл `index.html` тянет на себя стили и скрипты, а раздел бронирования изолирован в `/booking/`. Это позволяет легко развивать сайт: например, подключать новые секции или обновлять визуал без глобального рефакторинга.
