

ТЗ Календарь

1. Технический стек (фикс)

- Frontend: **Vue 3, TypeScript, Composition API**
- Стили: **TailwindCSS**
- State: **Pinia**
- Формы/валидация: **VeeValidate** (использовать для всех форм)
- UI-ек, дизайн: **по мотивам Material UI** (не обязателен MUI, но поведение/spacing/компоненты как в MUI)
- Backend / Auth / DB: **Firebase** (Auth, Firestore, Functions (option), Storage, Hosting (опционально))
- i18n: **vue-i18n** (начально: en, es, ru)
- Разработка/хостинг: **Replit** (конфиг для dev и production)

2. Главная структура проекта (папки — предложенная)

```
/src
  /assets
  /components
    /base      # базовые переиспользуемые компоненты
      /forms
        BaseInput.vue
        BaseTextarea.vue
        BaseSelect.vue
        BaseCheckbox.vue
        BaseRadio.vue
        BaseSwitch.vue
        BaseDatepicker.vue
        BaseTimepicker.vue
        BaseFileUploader.vue
    /buttons
      PrimaryButton.vue
      IconButton.vue
      TextButton.vue
    /modals
      BaseModal.vue
      ConfirmModal.vue
    /layoutPieces
      Logo.vue
      Avatar.vue
      NotificationBell.vue
      ThemeToggle.vue
      LanguageSwitcher.vue
    /layouts
      DefaultLayout.vue
      AuthLayout.vue
      FullscreenLayout.vue
    /pages
```

```
CalendarPage.vue
MastersListPage.vue
MasterPage.vue
SalonsListPage.vue
SalonPage.vue
ClientsListPage.vue
ClientPage.vue
LoginPage.vue
JoinPage.vue
ForgotPasswordPage.vue
NotFound.vue
Forbidden.vue
ProfilePage.vue
SettingsPage.vue
AdminDashboard.vue
ReportsPage.vue
/router
  index.ts
/store
  index.ts
  auth.ts
  user.ts
  notifications.ts
  calendar.ts
  ui.ts
/i18n
  index.ts
  en.json
  es.json
  ru.json
/services
  firebase.ts
  api.ts (thin wrapper)
  calendarUtils.ts
  permission.ts
/utils
  date.ts
  validators.ts
/composables
  useAuth.ts
  useCalendar.ts
  useModal.ts
  useI18n.ts
/types
/styles
  tailwind.css
  themes.css (theme variables)
```

main.ts

3. Обязательные базовые компоненты (детализировано)

- **Buttons:** Primary, Secondary, Icon, Link style, Loading state, disabled state
- **Inputs:** text, email, password, number, search, phone; plus textarea, masked input (phone/time), file upload
- **Selects:** basic select, async select, creatable select (если нужно)
- **Switch/Toggle:** для темы и др.
- **Date & Time pickers:** поддержка выбора времени с шагом 30 минут, диапазонов (для рабочего времени)
- **Checkbox/Radio**
- **Form wrapper:** компонент-обёртка для интеграции с VeeValidate (авто-валидация, ошибки)
- **Modals:** Base modal, Confirm / Info modal, Form modal (используется в календаре для записи)
- **Notification / Toast:** глобальные уведомления (store -> отображение)
- **Avatar / UserCard:** кликабельные ссылки на профиль
- **Skeleton / Loading:** универсальные для загрузки страниц/списков
- **Table / List:** базовый список/таблица с пагинацией, фильтрами, сортировкой

Все формы должны использовать VeeValidate (на уровне Form компонента или через composable), правила вынести в **validators.ts**.

4. Лэйауты и навигация

- **layouts:** DefaultLayout (с левым сайдбаром и хедером), AuthLayout (чистая страница для логина/регистрации), FullscreenLayout (для модалей/спец-страниц)
- **Header:** логотип (ссылка на корень), блок нотификаций (иконка + dropdown), переключатель темы, переключатель языков, кнопка логина/логаута, аватар пользователя
- **Left Sidebar:** навигация по основным разделам. Сайдбар должен быть коллапсируемым, поддерживать активный маршрут, и быть адаптивным (в мобилке — drawer).
- **Route meta** использовать для защиты: meta: { requiresAuth: true, layout: 'default', roles: ['admin','manager'] }

5. Аутентификация и доступ

- Firebase Auth (email/password, возможно OAuth later)
- **Роли:** admin, manager, master, client, guest — хранить в users/{uid} в Firestore.
- **Guard:** глобальный route guard (navigation guard) — redirect неавторизованных на Login/Join. Для неавторизованных разрешены только /login, /join, /forgot.
- **View-level guard:** v-if / permission check (например, скрывать кнопки управления для неподходящей роли).

- **Server-side security:** Firebase Security Rules для Firestore (подробно ниже).

6. Store (Pinia) — структура и ключевые модули

- auth — токен/uid, roles, isAuthenticated, refresh, login/logout, register, resetPassword
- user — профиль пользователя, настройки (preferredLanguage, theme), методы обновления профиля
- notifications — список уведомлений (серверные + локальные), методы markRead, pushNotification
- calendar — выбранныйSalonId, selectedMasterId, visibleRange, appointments cache, blockedTimes, workingHours
- ui — sidebar state (collapsed), theme, locale, global modals state

7. Страницы — полный перечень

Актуальные (у вас)

- Calendar (с режимами month/week/day)
- Masters List
- Master Profile (карточка/редактирование, график)
- Salons List
- Salon Page
- Clients List
- Client Page
- Login
- Join (Register)
- Forgot Password
- Profile (user own profile) — редактирование инфо и смена пароля.
- Settings — аккаунтные/приложенные настройки (язык, уведомления, часовой пояс).
- Admin Dashboard — управление пользователями, салонами, мастерами, метрики.
- Reports — загрузка/экспорт расписаний и доходов (если применимо).
- Bookings List — список всех записей (фильтры по мастеру/салону/ клиенту/дате).
- Booking Details Page — отдельная страница (если нужно) помимо модального окна.
- Forbidden / 404 — для доступа и ненайденных роутов.
- Onboarding/Welcome — краткий турориал для новых юзеров (опционально).

9. i18n (локализация)

- Библиотека: **vue-i18n** (composition API).
- Файлы: en.json, es.json, ru.json (всё UI-тексты, валидационные сообщения, ошибки).
- Сохранять выбор языка в user.settings и localStorage.
- Переводы для динамических сообщений (date formats) —

использовать Intl.DateTimeFormat с локалью.

- VeeValidate — интеграция с i18n для локализованных ошибок.

10. Темы (светлая / тёмная)

- Implement theme tokens (CSS variables). Файлы themes.css с :root и .dark модификатором.
- ThemeToggle компонент — переключение сохраняется в store и localStorage + user.settings (если залогинен).
- Поддержка переключения авто (system preference).

11. Firebase — детали

- Использовать: **Auth**, **Firebase**, **Storage**, (опционально) **Functions**, **Analytics**, **Crashlytics** (если нужно).
- Collections (предложение):
 - users/{uid} — profile, role, settings, meta
 - salons/{salonId} — info, address, managers, services
 - masters/{masterId} — userRef, salonRef(s), workingHours, services
 - clients/{clientId} — profile, contacts, notes
 - appointments/{id} — masterRef, clientRef, salonRef, start, end, serviceId, status
 - notifications/{id} — userRef, payload, read, createdAt
- **Security Rules:** подробно:
 - Только авторизованные могут читать свои данные; роли — для доступа к управлению серверами; менеджеры админяют салон; мастера видят и редактируют свои записи; клиенты только свои записи.
 - Validate input structure and types (timestamps, uids).
- **Indexes:** добавить composite indexes для запросов (по master + date range и т.п.).
- **Offline / sync:** Firestore offline может пригодиться; важно обеспечить актуальность календаря при оффлайн-редактировании (опционально).
- **Backups / Export:** настроить регулярные экспорты или Function для бэкапа (production).

12. Безопасность и конфиденциальность

- Не хранить секреты в клиентском коде. Использовать Firebase config из env.
- Ограничить read/write в rules.
- Rate-limiting на Cloud Functions (если используется).
- XSS/Content security — sanitize inputs where needed.

13. Accessibility (A11y)

- Все компоненты — keyboard accessible, proper aria-attributes.
- Color-contrast для тем.
- Skip-links, focus management при открытии модалей.

Страницы

1. Страница календаря (CalendarPage.vue)

Цель

Управление расписанием салона: записи клиентов, рабочими часами мастеров, переносами записей и блокировками.

Основной функционал

- Просмотр расписания в режимах **Месяц / Неделя / День**.
- Фильтры:
 - по салону (select в верхней панели),
 - по мастерам (multi-select),
 - по услуге (опционально).
- Драг-н-дроп для переноса записей:
 - разрешён только в пустые интервалы;
 - проверка длительности записи и рабочий график мастера;
 - проверка пересечений с другими записями.
- Создание новой записи:
 - при клике на пустой слот,
 - модальное окно создания (клиент, мастер, время, услуга, комментарий).
- Клик по записи:
 - открывается модал детальной информации.
- Управление рабочими часами мастера (в отдельном режиме или боковой панели).
- Возможность поставить блок / недоступное время.
- Быстрый поиск клиента (по имени/телефону).
- Синхронизация в реальном времени (Firestore subscriptions).
- Поддержка часовых поясов.

Компоненты

- CalendarView (month/week/day)
- CalendarHeader
- AppointmentCard
- CreateAppointmentModal
- AppointmentDetailsModal
- BlockTimeModal
- WorkingHoursPanel
- MasterSelect, SalonSelect
- DateNavigator
- Toast notifications

Доступ

- admin — полный доступ
- manager — всё в рамках своего салона
- master — видит только своё расписание
- client — нет доступа

Данные

- appointments

- masters
- salons
- workingHours
- blockedSlots

2. Страница списка мастеров (MastersListPage.vue)

Цель

Отображение всех мастеров, фильтрация, переход в профиль мастера.

Функционал

- Список мастеров карточками или таблицей.
- Поиск по имени, телефону, email.
- Фильтр по салону, статусу, услуге.
- Пагинация или lazy-load.
- Кнопка «Добавить мастера» (роль admin/manager).
- У каждого мастера:
 - аватар
 - ФИО
 - салон(ы)
 - график на сегодня
 - количество записей на сегодня

Данные

- masters collection
- salons collection (для отображения названий)

Доступ

- admin, manager
- мастеру доступно только просмотр списка (без кнопок редактирования)

3. Страница мастера (MasterPage.vue)

Цель

Показ полной информации о мастере и управление его графиком.

Функционал

- Общая информация:
 - аватар
 - имя, контакты
 - салон(ы)
 - статус
 - услуги, длительность услуг
- Управление графиком мастера:
 - рабочие часы по дням
 - исключения (отпуска/выходные)
- Список всех записей мастера (таблица с датами).
- Быстрый переход в календарь, отфильтрованный по мастеру.
- Добавить/изменить услуги мастера.
- Кнопка «Отправить уведомление мастеру».

- Кнопка «Удалить мастера» (только admin).

Доступ

- admin
- manager
- master (только просмотр и редактирование своего профиля)

4. Страница списка салонов (SalonsListPage.vue)

Цель

Показ всех салонов, навигация между ними.

Функционал

- Список салонов в виде карточек/таблицы.
- Добавление салона (admin).
- Фильтры по городу/району.
- Поиск по названию.
- Кол-во мастеров, клиентов, записей.

5. Страница салона (SalonPage.vue)

Цель

Управление конкретным салоном.

Функционал

- Информация о салоне:
 - название
 - адрес
 - часы работы
 - контактные данные
- Список мастеров этого салона.
- Список клиентов салона.
- Услуги салона.
- Метрики (записи за сегодня/неделю/месяц).
- Настройки салона.
- Кнопка перехода в календарь этого салона.

Доступ

- admin — полный
- manager — только своего салона
- master/client — нет доступа

6. Страница списка клиентов (ClientsListPage.vue)

Цель

Показ всех клиентов, управление их картотекой.

Функционал

- Поиск по фамилии, телефону.
- Фильтр по мастеру, салону.
- Добавить клиента вручную.
- Показ:
 - имя
 - телефон

- последний визит
- количество записей
- предпочтительный мастер

7. Страница клиента (ClientPage.vue)

Цель

Полный профиль клиента.

Функционал

- Основная информация: ФИО, контакты, примечания.
- Список записей клиента (таблица с датами).
- История посещений.
- Баланс/депозиты (если нужны).
- Быстро создать новую запись для клиента.
- Метки: VIP, проблемный, новый, постоянный.

8. Страница Login (LoginPage.vue)

Функционал

- Авторизация через email+password Firebase.
- Валидация VeeValidate.
- Кнопка «Забыли пароль?»
- Кнопка «Создать аккаунт».
- Ошибки Firebase отображаются в читабельном виде.
- Автоматический redirect при успешном входе.

Доступ

- Только незалогиненным.

9. Страница Join Us / Register (JoinPage.vue)

Цель

Создание аккаунта.

Функционал

- Форма: email, пароль, повтор пароля, имя, язык, согласие с правилами.
- Роль по умолчанию: client / или по настройкам проекта.
- Автоматический вход после регистрации.

10. Страница восстановления пароля (ForgotPasswordPage.vue)

Функционал

- Ввод email → отправка reset email через Firebase.
- Показывается сообщение об успехе.

11. Страница профиля пользователя (ProfilePage.vue)

Цель

Управление собственным профилем пользователя.

Функционал

- Изменение личных данных.
- Изменение языка интерфейса.

- Изменение темы.
- Смена пароля.
- Уведомления (on/off).

12. Страница настроек (SettingsPage.vue)

Функционал

- Настройки приложения:
 - язык
 - тема
 - часовой пояс
 - часовой формат (12/24)
 - упрощённый режим календаря
- Настройки доступа (admin).

13. Админ-панель (AdminDashboard.vue)

Цель

Центральная панель управления системой.

Функционал

- Количество салонов, мастеров, клиентов.
- Активные записи сегодня.
- Быстрые действия: добавить салон, добавить мастера.
- Топ мастеров (по загрузке).
- Логи активности (если нужно).

14. Страница отчётов (ReportsPage.vue)

Цель

Аналитика для салона/администратора.

Функционал

- Отчёты по:
 - загрузке мастеров
 - доходу
 - видам услуг
 - количеству клиентов
 - retention rate
- Экспорт в csv/pdf.
- Фильтры по дате/мастеру/салону.

15. Список всех записей (BookingsListPage.vue)

Функционал

- Таблица всех записей.
- Фильтры:
 - мастер
 - салон
 - клиент
 - дата
 - статус

- Быстрые действия: изменить/удалить/перейти.

16. Страница конкретной записи (`BookingDetailsPage.vue`) (Опционально, но желательно)

Функционал

- Полная информация о записи:
 - мастер, клиент, салон
 - время
 - услуга
 - комментарии
- Возможность переноса (как в календаре).
- Удаление записи.

17. `NotFoundPage.vue (404)`

Стандартная страница "Страница не найдена".

18. `ForbiddenPage.vue (403)`

Показывается, если у пользователя нет прав на просмотр.