تراویکا ، همسفر هوشمند درمان

مستندات فنی و پیاده‌سازی سامانه توریسم سلامت تراویکا

توسعه توسط مریم صمیمی

دانشجوی کارشناسی نرم افزار مهندسی کامپیوتر

پاییز 1404

معرفی استارتاپ تراویکا

تراویکا یک پلتفرم جامع توریسم سلامت است که با هدف اتصال مستقیم بیماران بین‌المللی به کلینیک‌ها و مراکز درمانی معتبر ایران طراحی شده است. این پلتفرم فرآیند انتخاب درمان، دریافت مشاوره، برآورد هزینه، هماهنگی سفر، خدمات اقامتی و پیگیری‌های بعد از درمان را در یک مسیر شفاف و قابل اعتماد قرار می‌دهد.

تراویکا تلاش می‌کند تا تجربه پزشکی-گردشگری را برای بیماران خارجی ساده، قابل اعتماد و بدون سردرگمی کند. این پلتفرم در نسخه MVP بر پایه وردپرس اجرا شده و قابلیت توسعه به یک سیستم مستقل (Headless + App) را دارد.

توریسم سلامت

توریسم سلامت یا Health Tourism به سفر افراد به کشورهای دیگر برای دریافت خدمات پزشکی، زیبایی، درمانی یا توان‌بخشی گفته می‌شود.

سه زیرشاخه اصلی دارد:

۱. Medical Tourism: جراحی‌ها و خدمات درمانی تخصصی مثل ارتوپدی، قلب، دندان، IVF

۲. Wellness Tourism: خدمات سبک زندگی سالم (اسپا، ماساژ درمانی, رژیم درمانی)

۳. Cosmetic Tourism: خدمات زیبایی (لیفت، رینوپلاستی، کاشت مو و...)

ایران یکی از قطب‌های مهم توریسم سلامت در منطقه است به دلیل:

- کیفیت پزشکی بالا
- متخصصان باتجربه
- هزینه کمتر از ترکیه، امارات، اروپا
- خدمات زیبایی و درمانی سطح بالا

مشکلات موجود در بازار ایران و خارج

مشکلات بیماران:

- نبود اطلاعات شفاف درباره درمان و قیمت
- نداشتن اعتماد به کلینیک‌ها
- سختی در پیدا کردن پزشک معتبر
- نبود یک پلتفرم رسمی برای هماهنگی سفر، پرواز و هتل
- ناهماهنگی بین درمان و اقامت
- دشواری در ارسال مدارک پزشکی
- نبود همراه مجرب برای مدیریت پروسه درمانی

مشکلات کلینیک‌ها:

- نداشتن ابزار استاندارد برای جذب بیماران خارجی
- نبود سیستم مدیریت لیدهای خارجی
- ضعف در ارائه محتوا به زبان‌های جهانی
- مشکل در هماهنگی سفر و اقامت بیماران
- وابستگی به واسطه‌های غیرحرفه‌ای

`

مشکلات بازار جهانی:

- رقابت شدید کشورهایی مثل ترکیه
- تمرکز شدید بر تبلیغات گمراه‌کننده
- کیفیت خدمات برابر با هزینه‌ها نیست
- کمبود پلتفرم‌های شفاف و داده‌محور

چرا این پلتفرم لازم است؟

زیرا هیچ پلتفرم ایرانی تا امروز **یک مسیر ۳۶۰ درجه شامل درمان + سفر + اقامت + پشتیبانی کامل** ارائه نداده است.

Travicaبه شکاف‌های اصلی بازار پاسخ می‌دهد:

- شفافیت کامل قیمت‌ها
- ارائه مسیر مشخص درمان
- رزرو رسمی پرواز و هتل در یک جا
- همراهی بیمار توسطCare Assistant
- محتوای تخصصی و قابل اعتماد
- ارتباط مستقیم و بدون واسطه با کلینیک‌های معتبر

این پلتفرم هم برای بیمار ارزش دارد، هم برای کلینیک، هم برای اقتصاد سلامت.

هدف پروژه

اهداف کلان:

- ایجاد یک مسیر استاندارد برای ورود بیماران خارجی به سیستم درمانی ایران
- افزایش اعتماد و کاهش ریسک برای بیمار
- ترکیب خدمات درمانی + سفر + اقامت در یک تجربه یکپارچه
- بهینه‌سازی فرآیند لید و مدیریت بیمار برای کلینیک‌ها

اهداف عملیاتی:

- ایجاد یک وب‌سایت چندزبانه
- طراحی فرم‌های تخصصی دریافت پرونده
- معرفی کلینیک‌ها و خدمات درمانی
- امکان افزودن خدمات جانبی مانند: رزرو پرواز ، رزرو هتل و هماهنگی همراه بیمار
- صفحه بررسی هزینه اولیه (Cost Estimation Static)
- یک ساختار قابل توسعه برای نسخه‌های بعدی

مخاطبان هدف

Persona ۱-بیمار بین‌المللی

- از عراق، عمان، کویت، قطر، امارات یا اروپا
- جراحی زیبایی / درمان تخصصی
- دنبال قیمت مناسب و کیفیت بالا
- نیازمند اقامت، پرواز و یک همراه امین

Persona ۲-کلینیک یا پزشک

- خدمات زیبایی، پوست، IVF، دندان، جراحی‌های تخصصی
- نیازمند ابزار حرفه‌ای برای جذب بیماران خارجی
- مشکل در مدیریت لید و محتوای چندزبانه

Persona ۳-همراه بیمار

- مترجم
- مسئول هماهنگی‌های اقامتی و بیمارستانی
- نقش کلیدی در کاهش استرس بیمار

Persona ۴-تیم پشتیبانی تراویکا

- بررسی مدارک
- هماهنگی درمان و سفر
- ارتباط با بیمار و کلینیک

محدوده پروژه(Project Scope)

- وب‌سایت وردپرس
- سیستم فرم‌های مشاوره
- ماژول مدیریت درخواست‌ها
- صفحات معرفی خدمات درمانی
- صفحات معرفی کلینیک و پزشک
- محتوای آموزشی (بلاگ)
- وب‌سایت چندزبانه

قدم های بعدی

- اپلیکیشن موبایل
- سیستم پرداخت بین‌المللی
- پنل اختصاصی کلینیک‌ها
- AI Treatment Recommendation
- سیستم رزرو مبتنی برAPI

مزیت رقابتی(USP)

- چیزی که تراویکا را متفاوت می‌کند:
- شفافیت روند درمان
- تمرکز روی کیفیت و اعتبار پزشکی
- پشتیبانی واقعی و انسان‌محور
- اتصال مستقیم بیمار به کلینیک بدون واسطه
- محتوای غنی پزشکی با استاندارد بین‌المللی
- تجربه کاربری ساده برای بیماران غیرایرانی

تکنولوژی‌های مورد استفاده

- WordPress ۶+
- PHP ۸+
- Theme Custom (Elementor + ACF)
- Custom Plugin for Lead Management
- MySQL Database
- WPML
- Cloudflare CDN
- Google Tag Manager

در نسخه توسعه:

- React / Next.js
- Node.js
- Headless WordPress API
- MongoDB
- Microservices

معماری کلی

۱ .لایه نمایش (Frontend Layer)

- صفحات درمانی
- صفحات کلینیک
- صفحات خدمات جانبی (هتل، پرواز، همراه بیمار)
- فرم‌های چندزبانه

۲ .لایه برنامه(Application Layer)

- ماژول مدیریت درخواست‌ها
- پلاگین آپلود مدارک
- Routingچندزبانه
- سیستم اطلاع‌رسانی و پیام‌رسان

۳ .لایه داده (Data Layer)

- دیتابیسMySQL
- جداول اختصاصی برای:
  - درخواست‌ها
  - مدارک
  - اطلاعات سفر و اقامت

۴ .لایه ارتباطات (Integration Layer)

- ایمیل
- واتساپ
- در آینده:
  - APIپرواز
  - APIهتل

تحلیل نیازمندی‌ها

تحلیل نیازمندی‌ها بخش بنیادی در طراحی و توسعه سامانه‌های نرم‌افزاری است. هدف اصلی آن، تبدیل اهداف کسب‌وکار، مشکلات موجود، محدودیت‌ها و انتظارات کاربران به مجموعه‌ای دقیق از الزامات عملکردی و غیرعملکردی است که مبنای طراحی معماری، ساختار پایگاه داده، توسعه ماژول‌ها و تست نرم‌افزار قرار گیرد.

در این پروژه، هدف طراحی یک سیستم توریسم سلامت است که خدمات زیر را یکپارچه ارائه می‌دهد:

- درخواست و رزرو درمان
- رزرو پرواز
- رزرو هتل
- برنامه‌ریزی سفر پزشکی
- ارائه همراه مجرب و مترجم
- مدیریت پرونده پزشکی
- پنل بیمار، پزشک، کلینیک و مدیریت
- ماژول پرداخت، پیگیری وضعیت، گزارش‌گیری و**CRM**

تحلیل ذی‌نفعان (STAKEHOLDER ANALYSIS)

نقش	نیازها و انتظارات	ذی‌نفع
مصرف‌کننده خدمات	فرآیند ساده رزرو درمان و سفر، قیمت شفاف، پیگیری لحظه‌ای، ارتباط با پشتیبان	بیماران (کاربر اصلی)
ارائه‌دهنده خدمات	دریافت پرونده بیمار، زمان‌بندی، ویزیت آنلاین، مدیریت درخواست‌ها	کلینیک‌ها و پزشکان
عامل اجرایی	دریافت برنامه سفر و وظایف، هماهنگی روزانه	همراهان مجرب / مترجم
مدیریت جریان کار	مدیریت درخواست‌ها، چت، بارگذاری مدارک	اپراتور پشتیبانی
مالک پلتفرم	کنترل تمام ماژول‌ها، گزارش‌گیری، تعریف بسته‌ها	مدیر سیستم
ارائه‌دهنده خدمات جانبی	دریافت سفارش پرواز/هتل، تایید رزرو	آژانس مسافرتی همکار

تحلیل کاربران و نقش‌ها

1. بیمار
   - ثبت‌نام، احراز هویت، تکمیل فرم درمان
   - انتخاب پکیج درمان، مشاهده قیمت
   - رزرو پرواز، رزرو هتل
   - مشاهده همراه اختصاصی
   - بارگذاری مدارک پزشکی
   - پیگیری وضعیت درخواست
   - پرداخت آنلاین
2. کلینیک/دکتر
   - بررسی فرم درمان
   - اعلام امکان‌پذیری
   - پیشنهاد قیمت

- ثبت زمان ویزیت
- تایید نهایی

3.همراه مجرب

- دریافت برنامه روزانه
- تایید حضور
- گزارش‌دهی

4.ادمین پلتفرم

- مدیریت کاربران
- مدیریت سفارش‌ها
- مدیریت خدمات پرواز/هتل
- گزارش‌گیری از سیستم

نیازمندی‌های غیرعملکردی(Non-Functional Requirements)

قابلیت اطمینان(Reliability)

- Uptimeحداقل ۹۹٪
- پشتیبان‌گیری خودکار روزانه
- تحمل خطا در رزرو پرواز و هتل

امنیت(Security)

- رمزنگاری داده‌های حساس(SSL – AES256)
- ذخیره‌سازی امن فایل‌های پزشکی
- نقش‌بندی دقیق سطح دسترسی(RBAC)
- لاگ‌گیری کامل اقدامات کاربران

کارایی (Performance)

- زمان پاسخ API کمتر از ۲ ثانیه
- قابلیت سرویس‌دهی به ۱۰هزار کاربر همزمان
- بهینه‌سازی پایگاه داده برای جستجوی سریع

مقیاس‌پذیری (Scalability)

- پشتیبانی از چند کشور و چند زبان
- قابلیت اتصال به چندین آژانس پرواز و هتل
- افقی (Load Balancing) و عمودی (CPU/RAM Increase)

تجربه کاربری (Usability)

- فرآیند ثبت درخواست در کمتر از ۶ مرحله
- استفاده آسان برای افراد مسن

- طراحی Mobile First

سازگاری (Compatibility)

- سازگار با:
  - وب
  - iOS
  - Android
- پشتیبانی از مرورگرهای Chrome ، Safari، Edge

نگهداشت‌پذیری (Maintainability)

- استفاده از REST API
- ساختار ماژولار
- کد تمیز و قابل تست

مدل‌سازیUse Case های سیستم

در این سیستم ۵ نقش داریم:

- بیمار
- کلینیک
- همراه
- اپراتور
- مدیر

Use Caseهای اصلی:

1. ثبت درخواست درمان
2. بررسی درخواست توسط پزشک

3. رزرو هتل

4. رزرو پرواز

5. تخصیص همراه

6. پرداخت

7. پیگیری سفر

8. گفت‌وگو با پشتیبانی

تحلیل جریان داده– DFD(سطح ۰ و ۱)

سطح ۰:

- کاربر ← ارسال درخواست درمان ← پردازش سیستم ← کلینیک
- سیستم ← مدیریت سفر ← پرواز / هتل
- بیمار ← وضعیت و نتایج ← سیستم

سطح ۱:

- پردازش درخواست
- مدیریت رزرو
- مدیریت پرداخت
- مدیریت همراه
- مدیریت پرونده

تحلیل جریان کار(Workflow)

Workflowکلی:

1. ثبت‌نام
2. ارسال درخواست درمان
3. بررسی پزشک
4. تایید هزینه
5. رزرو سفر
6. پرداخت
7. تخصیص همراه
8. اجرای سفر
9. تکمیل درمان
10. بازگشت

نیازمندی‌های سیستمی(System Requirements)

Frontend

- **React / Next.js**
- **UI**ساده، سازگار با موبایل
- ارتباط با **API** از طریق**Axios**

Backend

- **Python (Django REST)**یا**PHP (Laravel)**
- مدیریت کاربران
- سیستم پرداخت
- ماژول رزرو
- ماژول پیام‌رسانی
- ماژول مدیریت پرونده

Database

- **PostgreSQL**یا**MySQL**
- ذخیره‌سازی امن فایل

سناریوهای کاربران (User Scenarios)

ساختار سناریوها

هر سناریو شامل بخش‌های زیر است:

۱. **عنوان سناریو(Scenario Title)**

۲. **کاربر(Actor)**

۳. **هدف(Goal)**

۴. **شرط شروع(Trigger / Precondition)**

۵. **مراحل اجرایی(Flow / Steps)**

۶. **استثناها(Exceptions)**

۷. **نتیجه نهایی(Outcome)**

۸. **توضیحات اضافی(Notes)**

سناریو ۱ — درخواست درمان و مشاوره اولیه

- **Actor:** بیمار بین‌المللی
- **Goal:** ارسال درخواست درمان و دریافت مشاوره
- **Trigger:** کاربر وارد صفحه "درخواست مشاوره" می‌شود

Flow / Steps

۱. انتخاب نوع درمان (جراحی، زیبایی، توان‌بخشی)
۲. پر کردن فرم اطلاعات شخصی
۳. آپلود مدارک پزشکی موجود
۴. انتخاب زبان (انگلیسی/عربی)
۵. ارسال درخواست
۶. دریافت کد پیگیری و تایید ایمیل

Exceptions

- مدارک ناقص ⟶ درخواست رد می‌شود
- فرم پر نشده ⟶ کاربر هشدار می‌گیرد

Outcome

- درخواست در سیستم ذخیره می‌شود
- ایمیل و نوتیفیکیشن برای تیم Travica ارسال می‌شود

Notes

- این مرحله شروع مسیر بیمار است و اطلاعات برای کل فرآیند بعدی استفاده می‌شود

سناریو ۲ — بررسی درخواست توسط تیم تراویکا

- **Actor**: تیم پشتیبانیTravica
- **Goal**: تایید و بررسی مدارک بیمار
- **Trigger**: درخواست جدید توسط بیمار ارسال شده است

Flow / Steps

۱. ورود به داشبورد مدیریت درخواست‌ها
۲. بررسی مدارک و فرم‌ها
۳. تایید یا رد درخواست
۴. اضافه کردن یادداشت برای تیم کلینیک
۵. تخصیص همراه بیمار(Care Assistant)

Exceptions

- مدارک ناقص ← پیام اصلاح برای بیمار ارسال می‌شود

Outcome

- درخواست آماده مرحله بعدی (رزرو خدمات و هماهنگی سفر) می‌شود

سناریو ۳ — رزرو هتل و پرواز

- **Actor**: تیم پشتیبانی / همراه بیمار
- **Goal**: رزرو اقامت و پرواز برای بیمار
- **Trigger**: درخواست تایید شده است

Flow / Steps

۱. بررسی مقصد و زمان درمان
۲. انتخاب هتل از لیست همکاران
۳. رزرو پرواز با هماهنگی بیمار
۴. اطلاع‌رسانی به بیمار و ثبت اطلاعات در سیستم

Exceptions

- ظرفیت هتل کافی نیست → جایگزین پیشنهاد می‌شود
- پرواز مناسب یافت نشد → زمان دیگری پیشنهاد می‌شود

Outcome

- رزرو انجام شده و اطلاعات برای بیمار و کلینیک ارسال می‌شود

سناریو ۴ — همراه بیمار(Care Assistant)

- **Actor**: همراه بیمار
- **Goal**: مدیریت حضور بیمار در طول درمان
- **Trigger**: بیمار وارد ایران می‌شود

Flow / Steps

۱. خوش‌آمدگویی و هماهنگی ترانسفر فرودگاهی
۲. همراهی بیمار تا هتل
۳. هماهنگی رفتن به کلینیک و پزشک
۴. کمک به تکمیل مدارک و فرم‌ها
۵. ارائه گزارش وضعیت روزانه به تیم پشتیبانی

Exceptions

- بیمار تاخیر دارد → همراه زمان‌بندی را به روز می‌کند
- نیاز پزشکی اضطراری → تیم پشتیبانی اطلاع داده می‌شود

Outcome

- بیمار به راحتی و بدون مشکل به درمان دسترسی پیدا می‌کند

سناریو ۵ — پرداخت و تسویه هزینه‌ها

- **Actor**: بیمار و تیم پشتیبانی
- **Goal**: پرداخت هزینه درمان، هتل و خدمات همراه
- **Trigger**: مرحله نهایی رزرو خدمات انجام شده است

Flow / Steps

۱. نمایش فاکتور کامل برای بیمار
۲. ارسال لینک پرداخت امن
۳. تایید پرداخت توسط تیمTravica
۴. بروزرسانی وضعیت درخواست در سیستم

Exceptions

- پرداخت ناموفق → بیمار دوباره هدایت می‌شود
- اختلاف مبلغ → تیم پشتیبانی بررسی می‌کند

Outcome

- پرداخت کامل و اطلاعات در سیستم ثبت می‌شود

سناریو ۶ — پیگیری درمان و گزارش نهایی

- **Actor**: بیمار، همراه و کلینیک
- **Goal**: ثبت و پیگیری نتایج درمان
- **Trigger**: درمان شروع شده است

Flow / Steps

۱. همراه و بیمار گزارش وضعیت روزانه را ثبت می‌کنند
۲. پزشک گزارش درمان را وارد سیستم می‌کند
۳. تیم پشتیبانی بررسی می‌کند و یادداشت‌های تکمیلی اضافه می‌شود
۴. گزارش نهایی به بیمار و کلینیک ارسال می‌شود

Exceptions

- خطای ثبت گزارش → تیم پشتیبانی مداخله می‌کند

Outcome

- پرونده بیمار تکمیل شده و اطلاعات برای تحلیل‌های بعدی ذخیره می‌شود

سناریو ۷ — پشتیبانی و خدمات پس از درمان

- **Actor**: تیم پشتیبانیTravica
- **Goal**: ارائه خدمات پشتیبانی پس از درمان
- **Trigger**: بیمار به کشور خود بازگشته است

Flow / Steps

۱. ارسال پیام follow-up به بیمار
۲. دریافت بازخورد و امتیازدهی
۳. پاسخ به مشکلات و سوالات بیمار
۴. ثبت داده‌ها برای بهبود خدمات

Exceptions

- بازخورد منفی → تیم حل مشکل را آغاز می‌کند

Outcome

- تجربه بیمار کامل و اطلاعات برای تحلیل تجربه جمع‌آوری می‌شود

لیست ماژول‌ها(Modules)

هر ماژول شامل بخش‌های زیر است:

۱. **عنوان ماژول(Module Title)**
۲. **شرح ماژول(Description)**
۳. **ورودی‌ها(Inputs)**
۴. **خروجی‌ها(Outputs)**
۵. **وابستگی‌ها(Dependencies)**
۶. **نکات مهم و محدودیت‌ها(Notes / Constraints)**

ماژول ۱ — ماژول درخواست درمان

- **Description:**

این ماژول امکان ثبت درخواست درمان توسط بیمار، آپلود مدارک پزشکی و انتخاب زبان را فراهم می‌کند. تمامی اطلاعات به داشبورد مدیریت درخواست‌ها ارسال می‌شود.

- **Inputs:**
  - اطلاعات شخصی بیمار
  - نوع درمان انتخابی
  - مدارک پزشکی آپلود شده
  - زبان انتخابی
- **Outputs:**
  - ثبت درخواست در سیستم
  - کد پیگیری برای بیمار
  - اطلاع‌رسانی به تیم پشتیبانی
- **Dependencies:**
  - ماژول مدیریت کاربران
  - سیستم نوتیفیکیشن(Email/SMS)
  - ماژول آپلود مدارک
- **Notes:**
  - فرم باید چندزبانه باشد
  - آپلود مدارک محدود به فرمت‌های PDF و تصاویر پزشکی

ماژول ۲ — ماژول مدیریت درخواست‌ها

- **Description:**

این ماژول برای تیم Travica طراحی شده و امکان بررسی، تایید، رد یا تخصیص همراه بیمار به هر درخواست را فراهم می‌کند.

- **Inputs:**
  - درخواست‌های ارسال شده توسط بیماران
  - مدارک آپلود شده
  - اطلاعات اختصاص همراه بیمار
- **Outputs:**
  - وضعیت درخواست (تایید / رد / نیاز به اصلاح)
  - اطلاع‌رسانی برای بیمار
  - آماده‌سازی اطلاعات برای رزرو خدمات جانبی
- **Dependencies:**
  - ماژول درخواست درمان
  - ماژول همراه بیمار
  - سیستم نوتیفیکیشن
- **Notes:**
  - باید داشبورد به‌صورت real-time بروزرسانی شود
  - ثبت تاریخ و زمان بررسی درخواست‌ها الزامی است

ماژول ۳ — ماژول رزرو هتل و پرواز

- **Description:**

این ماژول هماهنگی اقامت و سفر بیمار را انجام می‌دهد و اطلاعات رزرو را در سیستم ثبت می‌کند.

- **Inputs:**
  - اطلاعات بیمار
  - تاریخ درمان
  - مقصد و محل اقامت پیشنهادی
  - ترجیح بیمار در انتخاب هتل و پرواز
- **Outputs:**
  - رزرو هتل
  - رزرو پرواز
  - تاییدیه به همراه بیمار و تیم پشتیبانی
- **Dependencies:**
  - ماژول مدیریت درخواست‌ها
  - ماژول همراه بیمار
  - ماژول نوتیفیکیشن
- **Notes:**
  - در MVP رزرو به صورت دستی توسط تیم پشتیبانی انجام می‌شود
  - نسخه آینده می‌تواند API بوکینگ هتل و پرواز را داشته باشد

ماژول ۴ — ماژول همراه بیمار

- **Description:**

این ماژول وظیفه مدیریت همراه بیمار، هماهنگی اقامت، رفت‌وآمد و خدمات کلینیک را دارد.

- **Inputs:**
  - اطلاعات بیمار
  - جزئیات رزرو هتل و پرواز
  - درخواست‌های درمان
- **Outputs:**
  - گزارش روزانه وضعیت بیمار
  - اطلاع‌رسانی تیم پشتیبانی
  - ثبت وظایف همراه در سیستم
- **Dependencies:**
  - ماژول رزرو سفر
  - ماژول مدیریت درخواست‌ها
- **Notes:**
  - همراه باید توانایی زبان انگلیسی و عربی داشته باشد
  - نقش کلیدی در تجربه کاربری بیمار دارد

ماژول ۵ — ماژول پرداخت

- **Description:**

مدیریت پرداخت هزینه درمان، رزرو هتل و پرواز، و خدمات همراه بیمار

- **Inputs:**
  - فاکتور خدمات بیمار
  - اطلاعات پرداخت
- **Outputs:**
  - تایید پرداخت
  - ثبت وضعیت تراکنش در سیستم
- **Dependencies:**
  - ماژول مدیریت درخواست‌ها
  - سیستم نوتیفیکیشن
- **Notes:**
  - در نسخه MVP پرداخت به صورت دستی و با هماهنگی تیم پشتیبانی انجام می‌شود
  - در نسخه بعدی API درگاه پرداخت آنلاین اضافه خواهد شد

ماژول ۶ — ماژول داشبورد بیمار

- **Description:**
  نمایش وضعیت درخواست‌ها، رزروها، پرونده پزشکی و اطلاعات همراه بیمار برای بیمار
- **Inputs:**
  - وضعیت درخواست
  - اطلاعات رزرو
  - مدارک پزشکی
- **Outputs:**
  - نمایش اطلاعات به بیمار به صورت امن
  - دسترسی به کد پیگیری و اطلاعات تماس
- **Dependencies:**
  - ماژول مدیریت درخواست‌ها
  - ماژول رزرو سفر
- **Notes:**
  - باید تجربه کاربری ساده و قابل فهم داشته باشد
  - دسترسی به اطلاعات حساس نیاز به احراز هویت دارد

ماژول ۷— ماژول نوتیفیکیشن

- **Description:**
  ارسال ایمیل، SMSو واتساپ برای اطلاع‌رسانی به بیماران، همراهان و تیم پشتیبانی
- **Inputs:**
  - رویدادهای سیستم (ثبت درخواست، تایید رزرو، پرداخت، گزارش روزانه)
- **Outputs:**
  - پیام به کاربران
  - ثبت لاگ پیام‌ها در سیستم
- **Dependencies:**
  - تمام ماژول‌های دیگر که نیاز به اطلاع‌رسانی دارند
- **Notes:**
  - ارسال به موقع پیام‌ها برای تجربه کاربری حیاتی است

لیست فیچرها(Features)

هر فیچر شامل بخش‌های زیر است:

1. **عنوان فیچر(Feature Title)**
2. **شرح فیچر(Description)**
3. **وابستگی‌ها(Dependencies / Related Modules)**
4. **نکات مهم و محدودیت‌ها(Notes / Constraints)**

فیچر ۱ — ثبت درخواست درمان

- **Description:**

امکان ثبت درخواست درمان توسط بیمار شامل انتخاب نوع درمان، آپلود مدارک پزشکی، تعیین زبان و ارسال درخواست به تیم پشتیبانی.

- **Dependencies:**
  - ماژول درخواست درمان
  - ماژول نوتیفیکیشن
- **Notes:**
  - فرم باید کاربرپسند و چندزبانه باشد
  - آپلود مدارک محدود به PDF و تصاویر پزشکی
  -

فیچر ۲ — مدیریت درخواست‌ها

- **Description:**

تیم پشتیبانی می‌تواند درخواست‌ها را بررسی، تایید یا رد کند و همراه بیمار را تخصیص دهد.

- **Dependencies:**
  - ماژول مدیریت درخواست‌ها
  - ماژول همراه بیمار
  - ماژول نوتیفیکیشن
- **Notes:**
  - ثبت تاریخ و زمان بررسی درخواست‌ها الزامی است
  - امکان اضافه کردن یادداشت داخلی برای تیم وجود دارد

فیچر ۳ — رزرو پرواز و هتل

- **Description:**

رزرو هتل و پرواز بیمار به صورت هماهنگی با تیم پشتیبانی و همراه بیمار.

- **Dependencies:**
  - ماژول رزرو هتل و پرواز
  - ماژول همراه بیمار
- **Notes:**
  - در MVP رزرو به صورت دستی انجام می‌شود
  - نسخه بعدی می‌تواند API بوکینگ هتل و پرواز داشته باشد

فیچر ۴ — همراه بیمار

- **Description:**

همراه بیمار در طول سفر، اقامت و درمان او را پشتیبانی می‌کند و گزارش روزانه ارائه می‌دهد.

- **Dependencies:**
  - ماژول همراه بیمار
  - ماژول رزرو سفر
- **Notes:**
  - همراه باید زبان انگلیسی/عربی بلد باشد
  - نقش حیاتی در تجربه کاربری بیمار دارد

فیچر ۵ — پرداخت و تسویه هزینه‌ها

- **Description:**

مدیریت پرداخت هزینه درمان، رزرو هتل و پرواز و خدمات همراه بیمار.

- **Dependencies:**
  - ماژول پرداخت
  - ماژول مدیریت درخواست‌ها
- **Notes:**
  - در MVP پرداخت به صورت دستی انجام می‌شود
  - در نسخه بعدی درگاه پرداخت آنلاین اضافه خواهد شد

فیچر ۶ — داشبورد بیمار

- **Description:**

نمایش وضعیت درخواست‌ها، رزروها، پرونده پزشکی و اطلاعات همراه بیمار به بیمار.

- **Dependencies:**
  - ماژول داشبورد بیمار
  - ماژول مدیریت درخواست‌ها
- **Notes:**
  - اطلاعات حساس نیاز به احراز هویت دارد
  - باید کاربرپسند و قابل فهم باشد

فیچر۷— نوتیفیکیشن(Notification System)

- **Description:**

ارسال ایمیل، SMSو واتساپ برای اطلاع‌رسانی بیماران، همراهان و تیم پشتیبانی در رویدادهای مهم سیستم.

- **Dependencies:**
  - ماژول نوتیفیکیشن
  - تمام ماژول‌های دارای رویداد
- **Notes:**
  - ارسال به موقع پیام‌ها برای تجربه کاربری حیاتی است

فیچر ۸ — گزارش‌دهی و تحلیل(Reporting & Analytics)

- **Description:**

تیم Travica می‌تواند گزارش‌های عملکرد، وضعیت درخواست‌ها و بازخورد بیماران را مشاهده و تحلیل کند.

- **Dependencies:**
  - ماژول مدیریت درخواست‌ها
  - ماژول داشبورد کلینیک
  - ماژول همراه بیمار
- **Notes:**
  - قابلیت توسعه برای گزارش‌های پیشرفته با نمودارها
  - اطلاعات باید امن ذخیره شوند

مستندات کد

Form-user

```
function __construct() {

        // enqueue
        add_action( 'admin_enqueue_scripts', array( $this, 'admin_enqueue_scripts' ) );
        add_action( 'login_form_register', array( $this, 'login_form_register' ) );

        // render
        add_action( 'show_user_profile', array( $this, 'render_edit' ) );
        add_action( 'edit_user_profile', array( $this, 'render_edit' ) );
        add_action( 'user_new_form', array( $this, 'render_new' ) );
        add_action( 'register_form', array( $this, 'render_register' ) );

        // save
        add_action( 'user_register', array( $this, 'save_user' ) );
        add_action( 'profile_update', array( $this, 'save_user' ) );

        // Perform validation before new user is registered.
        add_filter( 'registration_errors', array( $this, 'filter_registration_errors' ), 10, 3 );
    }
```

```
function admin_enqueue_scripts() {

    // bail early if not valid screen
    if ( ! acf_is_screen( array( 'profile', 'user', 'user-edit', 'profile-network', 'user-network', 'user-edit-network' ) ) ) {
        return;
    }

    // enqueue
    acf_enqueue_scripts();
}



function login_form_register() {

    // customize action prefix so that "admin_head" = "login_head"
    acf_enqueue_scripts(
        array(
            'context' => 'login',
        )
    );
```

```
function render_register() {

	// render
	$this->render(
	   array(
	      'user_id' => 0,
	      'view'    => 'register',
	      'el'      => 'div',
	   )
	);

function render_edit( $user ) {

	// add compatibility with front-end user profile edit forms such as bbPress
	if ( ! is_admin() ) {
	   acf_enqueue_scripts();
	}

	// render
	$this->render(
	   array(
```

```
                'user_id' => $user->ID,
                'view'    => 'edit',
                'el'      => 'tr',
            )
        );
    }
function render_new() {

        // Multisite uses a different 'user-new.php' form. Don't render fields here
        if ( is_multisite() ) {
            return;
        }

        // render
        $this->render(
            array(
                'user_id' => 0,
                'view'    => 'add',
                'el'      => 'tr',
            )
        );
```

```
function render( $args = array() ) {

	// Allow $_POST data to persist across form submission attempts.
	if ( isset( $_POST['acf'] ) ) { // phpcs:ignore WordPress.Security.NonceVerification.Missing
		add_filter( 'acf/pre_load_value', array( $this, 'filter_pre_load_value' ), ١٠, ٣ );
	}

	// defaults
	$args = wp_parse_args(
		$args,
		array(
			'user_id' => ٠,
			'view'    => 'edit',
			'el'      => 'tr',
		)
	);
```

Wp-login

```
require __DIR__ . '/wp-load.php';

// Redirect to HTTPS login if forced to use SSL.
if ( force_ssl_admin() && ! is_ssl() ) {
    if ( str_starts_with( $_SERVER['REQUEST_URI'], 'http' ) ) {
        wp_safe_redirect( set_url_scheme( $_SERVER['REQUEST_URI'], 'https' ) );
        exit;
    } else {
        wp_safe_redirect( 'https://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'] );
        exit;
    }
```

```
function login_header( $title = null, $message = '', $wp_error = null ) {
    global $error, $interim_login, $action;

    if ( null === $title ) {
        $title = __( 'Log In' );
    }

    // Don't index any of these forms.
    add_filter( 'wp_robots', 'wp_robots_sensitive_page' );
    add_action( 'login_head', 'wp_strict_cross_origin_referrer' );

    add_action( 'login_head', 'wp_login_viewport_meta' );

    if ( ! is_wp_error( $wp_error ) ) {
        $wp_error = new WP_Error();
    }
```

```
$shake_error_codes = apply_filters( 'shake_error_codes', $shake_error_codes );

    if ( $shake_error_codes && $wp_error->has_errors() && in_array( $wp_error->get_error_code(), $shake_error_codes, true ) ) {
        add_action( 'login_footer', 'wp_shake_js', ۱۲ );
    }

    $login_title = get_bloginfo( 'name', 'display' );

    /* translators: Login screen title. ۱: Login screen name, ۲: Network or site name. */
    $login_title = sprintf( __( '%۱$s &lsaquo; %۲$s &#۸۲۱۲; WordPress' ), $title, $login_title );

    if ( wp_is_recovery_mode() ) {
        /* translators: %s: Login screen title. */
        $login_title = sprintf( __( 'Recovery Mode &#۸۲۱۲; %s' ), $login_title );
    }
```

```
$login_title = apply_filters( 'login_title', $login_title, $title );

?><!DOCTYPE html>
<html <?php language_attributes(); ?>>
<head>
<meta http-equiv="Content-Type" content="<?php bloginfo( 'html_type' ); ?>; charset=<?php bloginfo( 'charset' ); ?>" />
<title><?php echo $login_title; ?></title>
<?php

wp_enqueue_style( 'login' );

if ( 'loggedout' === $wp_error->get_error_code() ) {
ob_start();
?>
<script>if("sessionStorage" in window){try{for(var key in sessionStorage){if(key.indexOf("wp-autosave-")!=-
1){sessionStorage.removeItem(key)}}}catch(e){}};</script>
<?php
wp_print_inline_script_tag( wp_remove_surrounding_empty_script_tags( ob_get_clean() ) );
}
```

```
$login_header_title = apply_filters_deprecated(
    'login_headertitle',
    array( $login_header_title ),
    '٥.٢.٠',
    'login_headertext',
    __( 'Usage of the title attribute on the login logo is not recommended for accessibility reasons. Use the link text instead.' )
);

$login_header_text = empty( $login_header_title ) ? __( 'Powered by WordPress' ) : $login_header_title;

$login_header_text = apply_filters( 'login_headertext', $login_header_text );

$classes = array( 'login-action-' . $action, 'wp-core-ui' );

if ( is_rtl() ) {
    $classes[] = 'rtl';
}

if ( $interim_login ) {
    $classes[] = 'interim-login';

    ?>
```

```
        <style type="text/css">html {background-color: transparent;}</style>
        <?php

        if ( 'success' === $interim_login ) {
            $classes[] = 'interim-login-success';
        }
    }

    $classes[] = ' locale-' . sanitize_html_class( strtolower( str_replace( '_', '-', get_locale() ) ) );

$classes = apply_filters( 'login_body_class', $classes, $action );

    ?>
    </head>
    <body class="login no-js <?php echo esc_attr( implode( ' ', $classes ) ); ?>">
    <?php
    wp_print_inline_script_tag( "document.body.className = document.body.className.replace('no-js','js');" );
    ?>
```

```
     * @param string $errors Login error messages.
     */
    $errors = apply_filters( 'login_errors', $errors );

    wp_admin_notice(
        $errors,
        array(
            'type'           => 'error',
            'id'             => 'login_error',
            'paragraph_wrap' => false,
        )
    );

if ( ! empty( $messages ) ) {
    /**
     * Filters instructional messages displayed above the login form.
     *
     * @since ٢.٥.٠
     *
     * @param string $messages Login messages.
     */
```

```
        $messages = apply_filters( 'login_messages', $messages );

        wp_admin_notice(
            $messages,
            array(
                'type'               => 'info',
                'id'                 => 'login-message',
                'additional_classes' => array( 'message' ),
                'paragraph_wrap'     => false,
            )
        );
    function login_footer( $input_id = '' ) {
global $interim_login;

// Don't allow interim logins to navigate away from the page.
if ( ! $interim_login ) {
    ?>
    <p id="backtoblog">
        <?php
        $html_link = sprintf(
            '<a href="%s">%s</a>',
            esc_url( home_url( '/' ) ),
```

```
            sprintf(
                /* translators: %s: Site title. */
                _x( '&larr; Go to %s', 'site' ),
                get_bloginfo( 'title', 'display' )
            )
        );
```

Classes/pages

```
<?php

if ( ! defined( 'ABSPATH' ) ) {
    exit; // disable direct access.
}

if ( ! class_exists( 'Mega_Menu_General' ) ) :

    /**
     * Handles the Mega Menu > Menu Settings page
     */
    class Mega_Menu_General {
```

```
/**
 * Constructor
 *
 * @since ١.٠
 */
public function __construct() {
    add_action( 'admin_post_megamenu_save_settings', array( $this, 'save_settings' ) );

    add_filter( 'megamenu_menu_tabs', array( $this, 'add_general_tab' ), ۴ );
    add_action( 'megamenu_page_general_settings', array( $this, 'general_settings_page' ) );
}

/**
 * Add the Menu Locations tab to our available tabs
 *
 * @param array $tabs
 * @since ٢.٨
 */
public function add_general_tab( $tabs ) {
    $tabs['general_settings'] = __( 'General Settings', 'megamenu' );
    return $tabs;
}
```

```
/**
 * Sanitize multidimensional array
 *
 * @since ٢.٧.٥
 */
public function sanitize_array( &$array ) {
   foreach ( $array as &$value ) {
      if ( ! is_array( $value ) ) {
         $value = sanitize_textarea_field( $value );
      } else {
         $this->sanitize_array( $value );
      }
   }
   return $array;
}

/**
 * Save menu general settings.
 *
 * @since ١.٠
 */
```

```
public function save_settings() {
    check_admin_referer( 'megamenu_save_settings' );

    if ( isset( $_POST['settings'] ) && is_array( $_POST['settings'] ) ) {
        $settings          = $this->sanitize_array( $_POST['settings'] );
        $submitted_settings = apply_filters( 'megamenu_submitted_settings', $settings );
        $existing_settings  = get_option( 'megamenu_settings' );
        $new_settings       = array_merge( (array) $existing_settings, $submitted_settings );

        update_option( 'megamenu_settings', $new_settings );
    }

    delete_option( 'megamenu_failed_to_write_css_to_filesystem' );

    do_action( 'megamenu_after_save_general_settings' );
    do_action( 'megamenu_delete_cache' );

    $url = isset( $_POST['_wp_http_referer'] ) ? $_POST['_wp_http_referer'] : admin_url( 'admin.php?page=maxmegamenu&saved=true' );

    $this->redirect( $url );
}
```

```
/**
 * Redirect and exit
 *
 * @since ١.٨
 */
public function redirect( $url ) {
    wp_redirect( $url );
    exit;
}

/**
 * Content for 'Settings' tab
 *
 * @since ١.٢
 */
public function general_settings_page( $saved_settings ) {

    $css = isset( $saved_settings['css'] ) ? $saved_settings['css'] : 'fs';

    $locations = get_registered_nav_menus();
```

```
    ?>

<div class='menu_settings menu_settings_general_settings'>

    <form action="<?php echo esc_attr( admin_url( 'admin-post.php' ) ); ?>" method="post">
        <input type="hidden" name="action" value="megamenu_save_settings" />
        <?php wp_nonce_field( 'megamenu_save_settings' ); ?>

        <h3 class='first'><?php esc_html_e( 'General Settings', 'megamenu' ); ?></h3>

        <table>
            <tr>
                <td class='mega-name'>
                    <?php esc_html_e( 'CSS Output', 'megamenu' ); ?>
                    <div class='mega-description'>
                    </div>
                </td>
                <td class='mega-value'>
                    <select name='settings[css]' id='mega_css'>
                        <option value='fs' <?php echo selected( 'fs' === $css ); ?>><?php esc_html_e( 'Save to filesystem', 'megamenu' ); ?>
                            <?php
                            if ( get_option( 'megamenu_failed_to_write_css_to_filesystem' ) ) {
```

```
                        echo ' ' . esc_html( '(Action required: Check upload folder permissions)', 'megamenu' );
                    }
                    ?>
                </option>
                <option value='head' <?php echo selected( 'head' === $css ); ?>><?php esc_html_e( 'Output in &lt;head&gt;', 'megamenu' ); ?></option>
                <option value='disabled' <?php echo selected( 'disabled' === $css ); ?>><?php esc_html_e( "Don't output CSS", 'megamenu' ); ?></option>
            <select>
            <div class='mega-description'>
                <div class='fs' style='display: <?php echo 'fs' === $css ? 'block' : 'none'; ?>'><?php esc_html_e( 'CSS will be saved to wp-content/uploads/maxmegamenu/style.css and enqueued from there.', 'megamenu' ); ?></div>
                <div class='head' style='display: <?php echo 'head' === $css ? 'block' : 'none'; ?>'><?php esc_html_e( 'CSS will be loaded from the cache in a &lt;style&gt; tag in the &lt;head&gt; of the page.', 'megamenu' ); ?></div>
                <div class='disabled' style='display: <?php echo 'disabled' === $css ? 'block' : 'none'; ?>'>
                    <?php esc_html_e( 'CSS will not be output, you must enqueue the CSS for the menu manually.', 'megamenu' ); ?>
                    <div class='fail'><?php esc_html_e( 'Selecting this option will effectively disable the theme editor and many of the features available in Max Mega Menu and Max Mega Menu Pro. Only enable this option if you fully understand the consequences.', 'megamenu' ); ?>
                </div>
            </div>
```

تست‌ها (Unit Tests + Functional Tests)

ابزارهای استفاده‌شده برای تست

PHPUnit

برای نوشتن تست‌های PHP در وردپرس:

- تست توابع Validation
- تست توابع Logic رزرو
- تست ذخیره‌سازی در دیتابیس
- تست رفتار توابع Hooks

WP-CLI Test Suite

WordPress Test Suiteامکان تست توابع مرتبط با WordPress Core و دیتابیس را فراهم می‌کند.

Postman / Thunder Client

برای تستEndpoint های:REST API

- book-treatment
- get-hotels
- book-flight
- verify-user

Manual Functional Testing

برای فرم‌های رزرو، پنل بیمار و کلینیک

تست عملکردی: رزرو درمان

سناریو:

1. کاربر وارد صفحه درمان می‌شود
2. فرم رزرو را پر می‌کند
3. API /book-treatment فراخوانی می‌شود
4. رزرو در دیتابیس ذخیره می‌شود
5. پیام موفقیت نمایش داده می‌شود

تست:Postman

POST → /wp-json/travica/v1/book-treatment

Body:

{

"user_id": 2,

"treatment_id": 12,

"country": "Turkey"

}

Expected Response:

{

"booking_id": 125

}

اگر booking_idبرنمی‌گردد → خطا.

تست عملکردی: لاگین بیمار

سناریو:

1. کاربر وارد فیلدهای لاگین می‌شود
2. هوک authenticateداده را بررسی می‌کند
3. کاربر به داشبورد بیمار منتقل می‌شود

تست:

- ورود کاربر بلاک شده
- ورود کاربر فعال
- ورود با رمز اشتباه

ثبت همراه مجرب

ورودی‌ها:

- نام همراه
- مدت زمان حضور
- مهارت‌ها
- زبان‌ها

API لازم:

/wp-json/travica/v1/add-assistant

تست‌های امنیتی(Security Tests)

تست‌ها شامل:

- جلوگیری از ارسال داده بدونToken
- بررسی نقش کاربر هنگام رزرو
- بررسی سطح دسترسی کلینیک در مشاهده درخواست‌ها
- SQL Injection Test

)با استفاده از ()$wpdb->prepareجلوگیری شده(

تست تزریق:SQL

ورودی:

{

"user_id": "1 OR 1=1"

}

خروجی: باید خطای invalid data بازگردد.

اهداف امنیتی(Security Objectives)

اهداف کلی:

- جلوگیری از دسترسی غیرمجاز به اطلاعات بیماران و سوابق درمانی.
- جلوگیری از جعل رزروها، درخواست خدمات یا پرداخت‌ها.
- جلوگیری از دست‌کاری مقاصد، قیمت‌ها، پزشکان و پکیج‌های درمانی.
- حفاظت از اطلاعات حساس مانند پاسپورت، ویزا، بلیط پرواز.
- حفظ سلامت فرایند رزرو و جلوگیری از رزروهای جعلی.
- محافظت ازAPI ها و جلوگیری از حملات.Brute Force
- ایجاد استانداردهای حداقلی امنیتی برای MVP وردپرسی.

دستاورد:

هدف ایجاد امنیت در حد قابل قبول برای نسخه اولیه است، اما ساختار قابلیت ارتقا به امنیت سازمانی را نیز دارد.

تهدیدهای اصلی سیستم(Threat Model)

تهدیدهای مربوط به کاربران

- سرقت اطلاعات سفر و هویت
- دسترسی غیرمجاز به اطلاعات پزشکی
- ثبت رزروهای جعلی با استفاده از فرم‌ها

تهدیدهای مربوط به اکانت‌ها

- Brute-forceروی صفحه لاگین
- ربودن نشست(Session Hijacking)
- دسترسی اشتباه به نقش‌های بالاتر

تهدیدهای مربوط به سیستم

- SQL Injectionدر بخش رزروها
- Cross-site scripting (XSS)داخل فرم‌های تماس
- API Abuseروی endpoint های رزرو
- CSRFروی فرم رزرو و فرم درخواست پزشک همراه

3.لایه‌های امنیتی

احراز هویت و نقش‌ها

- **فعال‌کردن نقش‌های سفارشی:**
  - patient
  - travel-agent
  - medical-companion
  - doctor-partner
  - admin

هر نقش سطح دسترسی متفاوتی دارد و نمی‌تواند به داده‌های حساس نقش دیگر دسترسی داشته باشد.

جلوگیری از حملات معمول

- CSRF Tokenبرای فرم رزرو
- محدودکردن تعداد درخواست‌های لاگین
- فعال‌سازی Google reCAPTCHA برای فرم‌های:
  - رزرو
  - درخواست پکیج
  - ثبت‌نام بیمار

محافظت از فایل‌ها

- جلوگیری از دسترسی مستقیم به فایل‌هایPHP
- قرار دادن فایل‌های محرمانه داخل پوشه uploads در حالتprivate
- ایجاد فایل htaccess.برای جلوگیری از لیست شدن دایرکتوری‌ها

امنیت سشن‌ها و کوکی‌ها

- فعال‌سازی HttpOnly و Secure cookies
- غیرفعال کردن REST API برای کاربران مهمان
- جلوگیری از دسترسی به پنل مدیریت برای همه نقش‌ها جز admin و agent

امنیت فایل‌ها و آپلودها

- محدودیت نوع فایل قابل آپلود

  (فقط = JPG/PNG/PDF پاسپورت)
- بررسی حجم فایل
- بررسی malware در فایل‌های آپلودی
- ذخیره‌سازی فایل‌ها در مسیر غیر عمومی

افزونه‌ها و ابزارهای امنیتی مورد استفاده

- Wordfence Security
- Limit Login Attempts Reloaded
- WP Activity Log
- Akismetبرای جلوگیری از فرم‌های اسپمی

پشتیبان‌گیری و بازیابی(Backup & Recovery)

- بکاپ‌گیری روزانه دیتابیس
- بکاپ‌گیری هفتگی فایل‌ها
- امکان بازیابی کامل در صورت حمله

استراتژی امنیت نسخه نهایی(After MVP)

برای نسخه بزرگ‌تر:

- احراز هویت دو مرحله‌ای برای مدیران
- رمزگذاری end-to-end اطلاعات پزشکی
- ایجاد فایروال اختصاصی
- مهاجرت به ساختار Microservices امن‌تر
- ایجاد Log Management حرفه‌ای(ELK)

اهداف بهینه‌سازی درTravica

هدف‌های اصلی بهینه‌سازی پرفورمنس:

- کاهش زمان لود صفحه به زیر ۲.۵ ثانیه
- افزایش سرعت جست‌وجو و رزرو
- جلوگیری از فشار روی سرور هنگام دریافت فرم‌های متعدد
- بهینه‌سازیAPI ها جهت پاسخ‌گویی سریع
- کاهش حجم عکس‌ها و فایل‌های رسانه‌ای برای جلوگیری از کندی
- افزایش مقیاس‌پذیری سایت برای تعداد بالای کاربران

معماری پرفورمنس

ساختار طوری طراحی شده که بتواند افزایش کاربران را مدیریت کند:

- اجرای Cache در چند لایه
- مدیریتQuery های دیتابیس
- استفاده ازLazy Loading
- بهینه‌سازی قالب اختصاصیTravica
- استفاده از CDN برای تصاویر

Lazy Loading

عکس‌های زیر Fold با تأخیر لود می‌شوند تا سرعت اولیه صفحه بالا برود.

بهینه‌سازی فونت‌ها

- بارگذاری فونت‌ها به صورتasync
- استفاده محدود از وزن‌های مختلف
- تبدیل فونت‌ها به فرمتWOFF2

استفاده ازCDN

برای عکس‌ها، فایل‌های تصویری بیمار، اسکن‌ها و فایل‌های رزرو، از CDN استفاده می‌شود.

مدیریت فایل‌ها و تصاویر(Media Optimization)

فشرده‌سازی تصاویر

- استفاده از فرمتWebP
- فشرده‌سازیLossless
- تنظیم حداکثر حجم تصویر 300 KB >

محدودیت آپلود

بیماران و همراهان فقط فایل‌های زیر را آپلود می‌کنند:

- JPG
- PNG
- PDF
- حداکثر ۵ مگابایت

Image CDN

استفاده از CDN برای:

- تصاویر پزشکان
- تصاویر مقصدهای درمانی
- عکس‌های پروفایل
- تصاویر پکیج‌ها

بهینه‌سازی سرعت رزرو و جست‌وجو

ویژگی‌های مهم:

- لیست پزشکانCached
- قیمت‌ها از قبل محاسبه و ذخیره می‌شوند
- فرم رزرو با AJAX ارسال می‌شود

نتیجه: سرعت بالا + جلوگیری ازRefresh های غیرضروری

مقیاس‌پذیری(Scalability)

روش‌های مقیاس‌پذیری MVP

- امکان انتقال از Shared Hosting → VPS
- امکان استفاده از Cloudflare برای کنترل ترافیک
- استفاده از Queue برای رزروهای سنگین

مقیاس‌پذیری نسخه نهایی

بعداً می‌توان سیستم را به ساختار Microservices با این اجزا ارتقا داد:

- Serviceپزشک
- Serviceرزرو
- Serviceهمراه مجرب
- Serviceپرواز
- Serviceپرداخت

Cache Strategy

Full Page Cache

صفحه‌های ثابت مثل:

- درباره Travica
- مقاصد درمانی
- قوانین و شرایط

کاملاً Cache می‌شوند.

Fragment Cache

بخش‌هایی مثل:

- پکیج‌ها
- لیست پزشکان

با TTL ۵ دقیقه کش می‌شوند.

UI/UX Design Documentation

اهداف طراحی تجربه کاربری(UX Goals)

هدف اصلی طراحی UI/UX در پلتفرم Travica ایجاد تجربه‌ای ساده، قابل‌اعتماد و امن برای کاربران در حوزه توریسم سلامت است. تجربه کاربری باید:

- فرآیند رزرو درمان + سفر را در کم‌ترین کلیک ممکن ارائه دهد
- کاربر را از سردرگمی بین پزشک، کلینیک، خدمات جانبی و پرواز نجات دهد
- اطلاعات پزشکی حساس را با نمایش واضح ولی محافظت‌شده ارائه کند
- زبان ساده، مسیرهای کوتاه و طراحی شفاف داشته باشد
- برای کاربران غیرمتخصص (بیماران و افراد مسن) قابل استفاده باشد
- قابلیت استفاده در موبایل را ٪۱۰۰ تضمین کند

پرسوناهای اصلی طراحی(Design Personas)

علی، بیمار نیازمند سفر درمانی

- ۴۲ ساله
- آشنایی متوسط با تکنولوژی
- هدف: درخواست مشاوره، انتخاب پزشک، رزرو درمان
- نیاز: مسیر ساده، توضیحات شفاف، اطمینان از امنیت اطلاعات

نرگس، همراه مجرب(Medical Companion)

- ۳۰ ساله
- باید درخواست‌های فوری را روی موبایل دریافت کند
- نیاز: پنل ساده، تایید سریع مأموریت، دیدن اطلاعات بیمار

کارگزار سفر(Travel Agent)

- رزرو پرواز + هتل + هماهنگی با بیمار
- نیاز: مشاهده وضعیت رزروها، مدیریت اسناد سفر، تایید پرداخت‌ها

پزشک / کلینیک

- **بررسی پرونده بیمار**
- **تایید اینکه آیا این بیمار کاندید درمان است یا نه**
- **نیاز: فایل‌ها، اسکن‌ها، سوابق و توضیحات واضح**

اصول طراحی(Design Principles)

Simplicityحداقل انتخاب‌ها در هر صفحه

Clarityنمایش اطلاعات مهم در بالا

Consistencyفونت، رنگ، سایز ثابت

Accessibilityرنگ‌های مناسب افراد کم‌بینا

Trustاستفاده از رنگ‌های آرام و طراحی رسمی برای بخش پزشکی

Speed-firstسبک‌سازی صفحه‌ها، استفاده از Skeleton Loader

سیستم طراحی(Design System / Style Guide)

رنگ‌ها(Color Palette)

پالت ترکیبی درمان + سفر:

- آبی تیره → (#003D5C)
- فیروزه‌ای → (#00B9C7)
- طلایی ملایم → (#F0C75E)
- سفید و خاکستری روشن

تایپوگرافی

- عنوان‌هاIRANSans Bold :
- پاراگراف‌هاIRANSans Regular :
- دکمه‌ها: سایز بزرگ برای کاربران مسن

فاصله‌گذاری‌ها

- Marginو Padding های زیاد جهت حس حرفه‌ای
- ارتفاع خطوط 1.7 برای خوانایی بهتر متن‌های پزشکی

Wireframeهای سطح بالا(High-Level Wireframes)

۵.۱ .صفحه اصلی

شامل:

- جست‌وجوی درمان یا پزشک
- لیست پرطرفدارترین پکیج‌های درمانی
- دکمه» → CTA رزرو مشاوره رایگان«
- بخش معرفی همراهان مجرب

صفحه پزشک(Doctor Profile)

بخش‌ها:

- عکس، نام، تخصص
- تجربه و توضیحات کوتاه
- اسلایدر نتایج درمان
- دکمه “درخواست بررسی پرونده”
- اطلاعات قیمت و زمان‌های قابل رزرو

صفحه پکیج درمانی

- شامل پرواز + هتل + درمان + همراه مجرب
- قیمت نهایی
- مدت‌زمان اقامت
- CTA → «ثبت درخواست»
- لیست کارهایی که در این پکیج انجام می‌شود

صفحه رزرو(Booking Flow)

Flowشامل ۳ مرحله:

۱ (اطلاعات بیمار

- نام
- پاسپورت
- فایل‌های پزشکی

۲ (انتخاب پزشک / پکیج / همراه مجرب

- نمایش کارت‌گونه
- انتخاب سریع(One tap select)

۳ (تایید نهایی و پرداخت

داشبورد بیمار(Patient Dashboard)

نمایش:

- رزروهای فعال
- وضعیت بررسی پزشک
- فایل‌ها
- چت با همراه مجرب
- یادآوری زمان پرواز و پذیرش در کلینیک

پنل همراه مجرب(Companion Panel)

نمایش:

- مأموریت‌های فعال
- اطلاعات بیمار
- مسیرهای سفر یا کلینیک
- فرم گزارش روزانه(Daily Report)

پنلTravel Agent

- رزروهای پرواز
- رزرو هتل
- پاسپورت و ویزا
- صدور بلیتPDF

الگوهای UI (Reusable UI Patterns)

- کارت پزشک
- کارت پکیج
- کارت همراه مجرب
- Steps Navigation
- فرم چندمرحله‌ای
- Modalبرای آپلود فایل پزشکی
- بخش نظرات پزشک

طراحی نسخه موبایل(Mobile-first)

- Navigationپایین(Bottom Navigation)
- CTAثابت پایین صفحه برای رزرو
- فرم‌ها به صورت تک‌ستونی
- دکمه‌ها بزرگ
- فواصل زیاد برای لمس راحت
- محافظت از حریم خصوصی با نمایش کمتر اطلاعات حساس

Accessibility

- کنتراست رنگ‌ها ۷:۱
- فونت درشت برای تیترها
- جلوگیری ازFlash های تند
- قابل استفاده با صفحه‌خوان(Screen Reader)
- توضیح alt برای تصاویر پزشکی و پزشکان

Design KPIs

- نرخ تکمیل رزرو
- زمان لازم برای تکمیل فرم درمان
- نرخ خروج(Bounce)
- رضایت بیمار از فرآیند رزرو
- سرعت پاسخ API در جست‌وجو