دانشگاه سمنان

دانشکدهٔ برق و کامپیوتر

پایان‌نامهٔ کارشناسی مهندسی کامپیوتر - گرایش نرم‌افزار

طراحی و پیاده‌سازی سامانهٔ یکپارچهٔ مدیریت باشگاه، مربی و ورزشکار (کوجیما)

نگارش:

.....

استاد راهنما:

.....

.....

بسم الله الرحمن الرحيم

دانشگاه سمنان

دانشکدهٔ برق و کامپیوتر

پایان‌نامهٔ کارشناسی مهندسی کامپیوتر - گرایش نرم‌افزار

طراحی و پیاده‌سازی سامانهٔ یکپارچهٔ مدیریت باشگاه، مربی و ورزشکار (کوجیما)

نگارش:

.....

استاد راهنما:

.....

.....

اینجانب(ان) ..... بدین‌وسیله اظهار می‌دارم که محتوای علمی این نوشتار با عنوان

«طراحی و پیاده‌سازی سامانهٔ یکپارچهٔ مدیریت باشگاه، مربی و ورزشکار (کوجیما)»

که به‌عنوان پایان‌نامهٔ کارشناسی مهندسی کامپیوتر - گرایش نرم‌افزار به دانشکدهٔ برق و کامپیوتر دانشگاه سمنان ارائه شده، دارای اصالت پژوهشی بوده و حاصل فعالیت علمی اینجانب(ان) است.

اینجانب(ان) می‌دانم که اگر خلاف ادعای بالا محرز شود، کلیهٔ حقوق مترتب بر این نوشتار از اینجانب(ان) سلب شده و مراتب قانونی مرتبط با آن نیز از طرف ذی‌ربط قابل پیگیری است.

نام و نام خانوادگی - شمارهٔ دانشجویی

تاریخ و امضا

نام و نام خانوادگی - شمارهٔ دانشجویی

تاریخ و امضا

این فرم به‌صورت دست‌نویس تکمیل می‌شود.

به نام خدا

دانشگاه سمنان

دانشکدهٔ برق و کامپیوتر

تأیید دفاع از پایان‌نامهٔ کارشناسی

پایان‌نامهٔ .....

برای اخذ درجهٔ **کارشناسی مهندسی کامپیوتر - گرایش نرم‌افزار**

با عنوان:

طراحی و پیاده‌سازی سامانهٔ یکپارچهٔ مدیریت باشگاه، مربی و ورزشکار (کوجیما)

در تاریخ ..... دفاع شد و مورد تأیید قرار گرفت.

تأییدکنندگان:

۱) استاد محترم داور ..... امضا

۲) استاد محترم راهنما ..... امضا

۳) مدیر محترم گروه مهندسی کامپیوتر ..... امضا

تقدیم به

همهٔ مربیانی که با حوصله راه را نشان می‌دهند

و همهٔ ورزشکارانی که هر روز از دیروزِ خود بهتر می‌شوند.

سپاس‌گزاری

از استاد راهنمای گرامی که در تمام مراحل تعریف مسئله، طراحی و پیاده‌سازی این سامانه با راهنمایی‌های دقیق و صبورانه همراه بودند، صمیمانه سپاس‌گزارم. همچنین از اساتید محترم داور که با نقد سازنده به بهبود این نوشتار کمک کردند، و از مربیان و مدیران باشگاه‌هایی که با صرف وقت، نیازها و مشکلات روزمرهٔ کار خود را برای ما توضیح دادند و در ارزیابی نسخه‌های آزمایشی سامانه مشارکت کردند، قدردانی می‌کنم.

در پایان از خانواده‌ام که در طول این مسیر پشتیبان بی‌دریغ من بودند، تشکر می‌کنم.

چکیده

گسترش فرهنگ تناسب‌اندام در سال‌های اخیر با رشد سریع تعداد باشگاه‌ها و مربیان خصوصی همراه بوده است؛ اما ابزارهای مدیریتی این حوزه همچنان پراکنده و جزیره‌ای‌اند: ورزشکار برنامهٔ تمرینی خود را روی کاغذ یا در پیام‌رسان دریافت می‌کند، مربی پیشرفت شاگردان را در دفترچه یا صفحه‌گسترده نگه می‌دارد و مدیر باشگاه برای اشتراک‌ها و پرداخت‌ها نرم‌افزار جداگانه‌ای دارد. افزون بر این، انتخاب مربی و باشگاه عملاً به محدودهٔ جغرافیایی محل سکونت فرد محدود می‌شود و سابقهٔ تمرینی ورزشکار با تغییر باشگاه از بین می‌رود.

در این پروژه سامانهٔ تحت وب «کوجیما» (KooGYMaa) طراحی و پیاده‌سازی شد که سه نقش مدیر باشگاه، مربی و عضو را در یک فضای کاری یکپارچه گرد هم می‌آورد. مدیر باشگاه اعضا و مربیان را می‌پذیرد، طرح‌های اشتراک تعریف می‌کند و پرداخت‌ها را با گزارش مالی به تفکیک ارز و خروجی CSV پایش می‌کند. مربی برنامه‌های تمرینی و غذایی ساخت‌یافته می‌سازد، آن‌ها را به شاگردان تخصیص می‌دهد، جلسات حضوری را با بررسی تداخل زمان‌بندی می‌کند و بازخورد می‌دهد. عضو برنامه‌ها را اجرا و ثبت می‌کند، پیشرفت بدنی و استریک تمرینی خود را می‌بیند، اشتراک می‌خرد و از هر باشگاه یا مربی در پلتفرم بهره می‌برد. سامانه با TypeScript، React 19، Next.js 16 و Prisma 7 روی پایگاه دادهٔ SQLite/libSQL پیاده‌سازی شده، کاملاً دوزبانه (فارسی راست‌به‌چپ با تقویم جلالی و انگلیسی) و واکنش‌گرا است، حالت روز/شب دارد و از احراز هویت مبتنی بر JWT با نشست‌های قابل ابطال، محدودیت نرخ درخواست، ثبت رخداد و اتوماسیون چرخهٔ عمر اشتراک‌ها پشتیبانی می‌کند. صحت منطق دامنه با ۷۶ آزمون واحد و واکنش‌گرایی رابط کاربری با ۱۹۸ ترکیب صفحه/اندازهٔ نمایش به‌صورت خودکار تأیید شده است.

نتیجهٔ کار، محصولی قابل استقرار است که با حذف پراکندگی ابزارها هزینهٔ عملیاتی باشگاه را کاهش می‌دهد، ارتباط مربی و ورزشکار را از محدودهٔ یک باشگاه فراتر می‌برد و سابقهٔ تمرینی فرد را مستقل از باشگاه حفظ می‌کند.

کلیدواژگان: مدیریت باشگاه ورزشی؛ سامانهٔ یکپارچهٔ تحت وب؛ برنامهٔ تمرینی و غذایی؛ مدیریت اشتراک و پرداخت؛ Next.js؛ Prisma؛ بومی‌سازی فارسی.

فهرست مطالب

عنوان	صفحه
چکیده	**ز**
مقدمه	**۱**
۱-۱- جایگاه موضوع	۱
۱-۲- بیان مسئله	۲
۱-۲-۱- نبود سیستم یکپارچه	۲
۱-۲-۲- محدود بودن ارتباط به باشگاه‌های یک منطقه	۳
۱-۲-۳- از دست رفتن سابقه و داده‌های پیشرفت	۳
۱-۲-۴- مشکلات عملیاتی و مالی باشگاه	۴
۱-۳- هدف پروژه	۴
۱-۴- سناریوهای کاربردی	۵
۱-۴-۱- سناریوی مدیر باشگاه	۵
۱-۴-۲- سناریوی مربی	۵
۱-۴-۳- سناریوی عضو	۶
۱-۵- مشخصات کلی سامانهٔ ساخته‌شده	۶
۱-۶- مروری بر کارهای مشابه	۸
۱-۷- روش انجام کار	۹
۱-۸- ساختار پایان‌نامه	۹
مبانی نظری و فناوری‌های به‌کاررفته	**۱۱**
۲-۱- سامانه‌های اطلاعاتی چندنقشی و مدل مجوزدهی	۱۱
۲-۲- معماری برنامه‌های وب مدرن	۱۲
۲-۲-۱- رندر سمت سرور و مؤلفه‌های سرور	۱۲
۲-۲-۲- مسیرهای API و معماری REST	۱۲
۲-۳- پایگاه داده و نگاشت شیء-رابطه	۱۳
۲-۳-۱- مدل رابطه‌ای و نرمال‌سازی	۱۳
۲-۳-۲- SQLite و libSQL	۱۳
۲-۳-۳- Prisma به‌عنوان ORM	۱۴
۲-۴- احراز هویت و مدیریت نشست	۱۵

۱-۴-۲- ذخیرهٔ امن گذرواژه ۱۵
۲-۴-۲- توکن JWT و کوکی HttpOnly ۱۶
۳-۴-۲- محدودیت نرخ درخواست ۱۶
۵-۲- پرداخت الکترونیک و خودتوانی عملیات ۱۶
۶-۲- بومی‌سازی، راست‌به‌چپ و تقویم جلالی ۱۷
۷-۲- طراحی واکنش‌گرا و دسترس‌پذیری ۱۷
۸-۲- آزمون نرم‌افزار ۱۸
۹-۲- جمع‌بندی فناوری‌های انتخاب‌شده ۱۸
طراحی، پیاده‌سازی و راهنمای استفادهٔ سامانهٔ کوجیما ۲۱
۱-۳- معماری کلی سامانه ۲۱
۱-۱-۳- ساختار پروژه ۲۳
۲-۳- مدل داده ۲۳
۱-۲-۳- هویت، عضویت و اشتراک ۲۴
۲-۲-۳- برنامه‌های تمرینی و غذایی ۲۵
۳-۲-۳- اجرا، پیشرفت و ارتباطات ۲۶
۴-۲-۳- انواع شمارشی و وضعیت‌ها ۲۷
۳-۳- احراز هویت و مدیریت حساب ۲۹
۱-۳-۳- ثبت‌نام ۲۹
۲-۳-۳- ورود و نشست ۳۰
۳-۳-۳- بازیابی حساب ۳۲
۴-۳- جریان‌های اصلی کسب‌وکار ۳۴
۱-۴-۳- درخواست عضویت و همکاری ۳۴
۲-۴-۳- خرید اشتراک و پرداخت ۳۵
۳-۴-۳- چرخهٔ برنامهٔ تمرینی: از طراحی تا بازخورد ۳۸
۴-۴-۳- زمان‌بندی جلسات حضوری ۳۹
۵-۳- امنیت و کنترل دسترسی ۳۹
۱-۵-۳- لایه‌های دفاعی ۳۹
۲-۵-۳- ابطال نشست ۴۰
۶-۳- بومی‌سازی: فارسی، راست‌به‌چپ و تقویم جلالی ۴۱
۱-۶-۳- چیدمان راست‌به‌چپ ۴۱
۲-۶-۳- تاریخ، اعداد و پول ۴۱

۳-۶-۳- فونت ۴۲
۳-۷- طراحی واکنش‌گرا و حالت روز /شب ۴۲
۳-۸- اتوماسیون، پایش و پشتیبان‌گیری ۴۴
۳-۹- آزمون و تضمین کیفیت ۴۷
۳-۱۰- راهنمای استفاده: ورود به سامانه ۴۸
۳-۱۰-۱- صفحهٔ اصلی ۴۸
۳-۱۰-۲- ثبت‌نام ۴۹
۳-۱۰-۳- ورود و بازیابی گذرواژه ۵۰
۳-۱۱- راهنمای مدیر باشگاه ۵۱
۳-۱۱-۱- فضاهای کاری و ساخت باشگاه ۵۱
۳-۱۱-۲- داشبورد ۵۱
۳-۱۱-۳- مدیریت اعضا ۵۲
۳-۱۱-۴- مدیریت مربیان ۵۳
۳-۱۱-۵- طرح‌های اشتراک ۵۴
۳-۱۱-۶- اشتراک‌ها ۵۵
۳-۱۱-۷- پرداخت‌ها و گزارش مالی ۵۶
۳-۱۱-۸- تنظیمات باشگاه ۵۷
۳-۱۲- راهنمای مربی ۵۸
۳-۱۲-۱- داشبورد مربی ۵۸
۳-۱۲-۲- تکمیل پروفایل و پیوستن به باشگاه ۵۹
۳-۱۲-۳- مدیریت شاگردان ۶۱
۳-۱۲-۴- ساخت برنامهٔ تمرینی ۶۲
۳-۱۲-۵- ساخت برنامهٔ غذایی ۶۳
۳-۱۲-۶- برنامهٔ زمانی و جلسات ۶۴
۳-۱۲-۷- پیگیری پیشرفت و بازخورد ۶۵
۳-۱۳- راهنمای عضو (ورزشکار) ۶۶
۳-۱۳-۱- داشبورد عضو ۶۶
۳-۱۳-۲- پیوستن به باشگاه و خرید اشتراک ۶۷
۳-۱۳-۳- انتخاب مربی ۶۹
۳-۱۳-۴- اجرای تمرین ۷۰
۳-۱۳-۵- تغذیه ۷۱

۶-۱۳-۳- برنامهٔ زمانی ۷۲
۷-۱۳-۳- پیگیری پیشرفت ۷۳
۸-۱۳-۳- اعلان‌ها و پروفایل ۷۴
۹-۱۳-۳- استفاده در گوشی همراه ۷۶
جمع‌بندی، نتیجه‌گیری و پیشنهادها ۷۷
۱-۴- جمع‌بندی نتایج ۷۷
۲-۴- محدودیت‌های سامانهٔ ساخته‌شده ۷۸
۳-۴- مزایا و معایب فناوری‌های به‌کاررفته ۷۹
۴-۴- تجربیات پروژه ۸۰
۱-۴-۴- تجربیات فنی ۸۰
۲-۴-۴- تجربیات غیرفنی ۸۱
۵-۴- پیشنهادهایی برای ادامهٔ کار ۸۱
۶-۴- سخن پایانی ۸۲
پیوست الف: فهرست مسیرهای API ۸۳
پیوست ب: جدول‌های پایگاه داده ۸۹
پیوست ج: راهنمای نصب و استقرار ۹۳
ج-۱- اجرای محلی ۹۳
ج-۲- متغیرهای محیطی ۹۴
ج-۳- استقرار عملیاتی ۹۴
ج-۴- حساب‌های نمونه ۹۵
منابع و مراجع ۹۷
چکیدهٔ انگلیسی ۱۰۰

فهرست شکل‌ها

عنوان **صفحه**

شکل ۲-۱: لایه‌های دسترسی به داده: از مؤلفهٔ سرور تا فایل پایگاه داده ۱۵
شکل ۳-۱: بلوک‌دیاگرام کلی سامانهٔ کوجیما در سطح لایه‌ها ۲۲
شکل ۳-۲: نقش‌های سامانه و تعامل آن‌ها حول موجودیت باشگاه ۲۲
شکل ۳-۳: نمای کلی روابط میان مدل‌های پایگاه داده (نمودار موجودیت-رابطه) ۲۴
شکل ۳-۴: خوشهٔ هویت، عضویت، اشتراک و پرداخت با فیلدهای اصلی ۲۶
شکل ۳-۵: خوشهٔ برنامه‌های تمرینی و غذایی و ثبت اجرای آن‌ها ۲۸
شکل ۳-۶: فلوچارت فرایند ثبت‌نام ۳۱
شکل ۳-۷: فلوچارت فرایند ورود ۳۳
شکل ۳-۸: نمودار توالی ورود و دسترسی به صفحهٔ محافظت‌شده ۳۴
شکل ۳-۹: الگوی درخواست-تأیید برای عضویت، همکاری مربی و رابطهٔ مربی-شاگرد ۳۵
شکل ۳-۱۰: فلوچارت خرید اشتراک و تأیید پرداخت ۳۷
شکل ۳-۱۱: نمودار حالت اشتراک (راست) و پرداخت (چپ) ۳۸
شکل ۳-۱۲: چرخهٔ برنامهٔ تمرینی از طراحی توسط مربی تا بازخورد ۳۸
شکل ۳-۱۳: نمودار توالی اعلام بازهٔ در دسترس و رزرو جلسه با تشخیص تداخل ۳۹
شکل ۳-۱۴: جریان بومی‌سازی از کوکی تا رندر نهایی ۴۱
شکل ۳-۱۵: نمای موبایل پنل عضو (راست) و منوی همبرگری باز (چپ) ۴۳
شکل ۳-۱۶: داشبورد مدیر باشگاه در حالت شب ۴۴
شکل ۳-۱۷: فلوچارت زمان‌بند روزانهٔ اشتراک‌ها ۴۶
شکل ۳-۱۸: چرخهٔ یکپارچه‌سازی و استقرار پیوسته ۴۷
شکل ۳-۱۹: صفحهٔ اصلی سامانه ۴۹
شکل ۳-۲۰: فرم ثبت‌نام با انتخاب نقش ۵۰
شکل ۳-۲۱: صفحهٔ ورود (راست) و فراموشی گذرواژه (چپ) ۵۰
شکل ۳-۲۲: فهرست فضاهای کاری مدیر و فرم ساخت باشگاه ۵۱
شکل ۳-۲۳: داشبورد مدیر باشگاه ۵۲
شکل ۳-۲۴: مدیریت اعضا و درخواست‌های عضویت ۵۳
شکل ۳-۲۵: مدیریت مربیان باشگاه ۵۴
شکل ۳-۲۶: تعریف و مدیریت طرح‌های اشتراک ۵۵
شکل ۳-۲۷: فهرست و مدیریت اشتراک‌ها ۵۶

شکل ۳-۲۸: پرداخت‌ها، گزارش مالی و خط زمانی حسابرسی ۵۷
شکل ۳-۲۹: تنظیمات باشگاه ۵۸
شکل ۳-۳۰: داشبورد مربی ۵۹
شکل ۳-۳۱: پروفایل حرفه‌ای مربی ۶۰
شکل ۳-۳۲: باشگاه‌های مربی و درخواست همکاری ۶۱
شکل ۳-۳۳: مدیریت شاگردان ۶۲
شکل ۳-۳۴: فهرست برنامه‌های تمرینی مربی ۶۳
شکل ۳-۳۵: برنامه‌های غذایی مربی ۶۴
شکل ۳-۳۶: برنامهٔ زمانی مربی: بازه‌های در دسترس و تقویم جلسات ۶۵
شکل ۳-۳۷: پیشرفت مراجعان ۶۶
شکل ۳-۳۸: داشبورد عضو ۶۷
شکل ۳-۳۹: کشف باشگاه‌ها ۶۸
شکل ۳-۴۰: اشتراک‌های عضو با امکان توقف، ازسرگیری، لغو و دریافت رسید ۶۹
شکل ۳-۴۱: یافتن مربی در سراسر پلتفرم ۷۰
شکل ۳-۴۲: برنامهٔ تمرینی عضو و سوابق اجرا ۷۱
شکل ۳-۴۳: برنامهٔ غذایی و ثبت وعده‌ها ۷۲
شکل ۳-۴۴: برنامهٔ زمانی و رزرو جلسه ۷۳
شکل ۳-۴۵: پیگیری پیشرفت: اندازه‌ها، نمودار، عکس‌ها و اهداف ۷۴
شکل ۳-۴۶: اعلان‌ها و تنظیمات یادآوری ۷۵
شکل ۳-۴۷: پروفایل عضو، تغییر گذرواژه و نشست‌های فعال ۷۶

فهرست جدول‌ها

عنوان **صفحه**

جدول ۱-۱: پراکندگی ابزارهای رایج در مدیریت باشگاه و مربی‌گری ۳
جدول ۱-۲: مشخصات کلی و قابلیت‌های سامانهٔ کوجیما ۷
جدول ۱-۳: مقایسهٔ دسته‌های نرم‌افزاری موجود با کوجیما ۸
جدول ۲-۱: قرارداد کدهای وضعیت HTTP در API کوجیما ۱۳
جدول ۲-۲: فناوری‌های به‌کاررفته و دلیل انتخاب ۱۸
جدول ۳-۱: ساختار پوشه‌های اصلی پروژه ۲۳
جدول ۳-۲: انواع شمارشی اصلی طرحواره ۲۹
جدول ۳-۳: لایه‌های امنیتی سامانه ۴۰
جدول ۳-۴: پوشش آزمون‌های واحد به تفکیک حوزه ۴۷
جدول ۴-۱: ارزیابی دستاوردها در برابر اهداف ۷۷
جدول ۴-۲: مزایا و معایب مشاهده‌شدهٔ فناوری‌های اصلی ۸۰
جدول الف-۱: مسیرهای API - احراز هویت ۸۴
جدول الف-۲: مسیرهای API - مدیر باشگاه ۸۴
جدول الف-۳: مسیرهای API - مربی ۸۵
جدول الف-۴: مسیرهای API - عضو ۸۶
جدول الف-۵: مسیرهای API - سیستمی ۸۷
جدول ب-۱: مدل‌های پایگاه داده ۸۹
جدول ج-۱: متغیرهای محیطی ۹۴
جدول ج-۲: حساب‌های دادهٔ نمونه ۹۵

فصل اول
مقدمه

در این فصل ابتدا جایگاه موضوع در بستر تغییرات صنعت تناسب‌اندام تشریح می‌شود؛ سپس مسئلهٔ اصلی پژوهش، یعنی پراکندگی ابزارها و محدودیت جغرافیایی ارتباط ورزشکار و مربی، به‌طور دقیق بیان می‌شود. در ادامه هدف پروژه، سناریوهای کاربردی، مشخصات کلی سامانهٔ ساخته‌شده و مروری بر کارهای مشابه آورده می‌شود و فصل با معرفی ساختار فصل‌های بعدی به پایان می‌رسد.

۱-۱- جایگاه موضوع

ورزش و تناسب‌اندام در دو دههٔ اخیر از یک فعالیت حاشیه‌ای به بخشی از سبک زندگی شهری تبدیل شده است. بر اساس گزارش انجمن جهانی سلامت و تناسب‌اندام، تعداد اعضای باشگاه‌های ورزشی در جهان پیش از همه‌گیری کووید-۱۹ به بیش از ۱۸۰ میلیون نفر رسیده بود و بازار این صنعت سالانه نزدیک به ۱۰۰ میلیارد دلار درآمد داشته است [IHRSA-R20]. در ایران نیز شمار باشگاه‌های دارای مجوز از سازمان‌های ورزشی طی سال‌های اخیر رو به افزایش بوده و مربی‌گری خصوصی، به‌ویژه در رشته‌های بدن‌سازی و آمادگی جسمانی، به یک حرفهٔ مستقل بدل شده است.

این رشد کمّی اما با رشد کیفی ابزارهای مدیریتی همراه نبوده است. مشاهدات میدانی ما در گفت‌وگو با مدیران و مربیان چند باشگاه نشان می‌دهد که بخش بزرگی از فرایندهای روزمره - از ثبت‌نام عضو و پیگیری تمدید اشتراک گرفته تا نوشتن برنامهٔ تمرینی و پایش پیشرفت - هنوز به روش‌های دستی، کاغذی یا با ابزارهای عمومی مانند پیام‌رسان‌ها و صفحه‌گسترده‌ها انجام می‌شود. در چنین وضعیتی، داده‌های ارزشمندِ سابقهٔ تمرینی افراد یا اصلاً ثبت نمی‌شود یا در جزیره‌های اطلاعاتی جداگانه پراکنده می‌ماند.

از سوی دیگر، نسل جدید ورزشکاران با اپلیکیشن‌های سلامت و ساعت‌های هوشمند خو گرفته‌اند و انتظار دارند تجربهٔ باشگاه نیز به همان اندازه دیجیتال، شفاف و شخصی‌سازی‌شده باشد. پژوهش‌ها نشان می‌دهد که خودپایشی [1] و بازخورد منظم مربی، دو عامل مهم در پایبندی بلندمدت به برنامهٔ تمرینی هستند [Burke-P11]؛ عواملی که تحقق آن‌ها بدون یک بستر نرم‌افزاری مشترک میان مربی و ورزشکار دشوار است.

۲-۱- بیان مسئله

مسئله‌ای که این پروژه به آن می‌پردازد، از چند کمبود به‌هم‌پیوسته ناشی می‌شود که در ادامه هر یک را جداگانه بررسی می‌کنیم.

۱-۲-۱- نبود سیستم یکپارچه

سه بازیگر اصلی اکوسیستم باشگاه - مدیر، مربی و ورزشکار - هر یک با ابزار متفاوتی کار می‌کنند. مدیر باشگاه معمولاً یک نرم‌افزار حضور و غیاب یا صندوق دارد که فقط ورود و خروج و پرداخت شهریه را ثبت می‌کند. مربی برنامهٔ تمرینی را در دفترچه، فایل ورد یا پیام‌رسان می‌نویسد و ورزشکار عکس آن را در گوشی خود نگه می‌دارد. هیچ‌یک از این ابزارها با دیگری ارتباط ندارد؛ در نتیجه مدیر نمی‌داند کدام عضو واقعاً تمرین می‌کند، مربی نمی‌داند کدام شاگردش اشتراک فعال دارد و ورزشکار نمی‌تواند پیشرفت خود را در طول زمان ببیند. جدول ۱-۱ این پراکندگی را خلاصه کرده است.

۱ Self-monitoring

جدول ۱-۱ پراکندگی ابزارهای رایج در مدیریت باشگاه و مربی‌گری

فرایند	ابزار رایج امروز	مشکل اصلی
ثبت‌نام و تمدید اشتراک	دفتر، نرم‌افزار صندوق، اکسل	بی‌خبری از انقضا، یادآوری دستی، خطای انسانی
پذیرش مربی در باشگاه	توافق شفاهی	نبود سابقهٔ قابل استناد، سردرگمی در تسویه
برنامهٔ تمرینی	کاغذ، عکس، پیام‌رسان	گم‌شدن، نبود نسخه‌بندی، عدم امکان ثبت اجرا
برنامهٔ غذایی	فایل PDF یا متن آزاد	عدم امکان پیگیری روزانه و اصلاح تدریجی
پایش پیشرفت	حافظهٔ مربی، عکس‌های پراکنده	مقایسهٔ زمانی دشوار، وابستگی به یک مربی
زمان‌بندی جلسات	تماس تلفنی، پیام	تداخل وقت، فراموشی، عدم ثبت
گزارش مالی	جمع دستی رسیدها	خطا، عدم تفکیک، نبود حسابرسی

۲-۲-۱- محدود بودن ارتباط به باشگاه‌های یک منطقه

انتخاب مربی امروز تابع جغرافیاست: ورزشکار به نزدیک‌ترین باشگاه می‌رود و از میان مربیانی که همان ساعت در همان سالن حضور دارند، یکی را انتخاب می‌کند. مربی نیز شاگردانش را فقط از میان اعضای همان باشگاه می‌یابد. این وابستگی به مکان چند پیامد دارد: نخست، مربی متخصص در یک رشتهٔ خاص (مثلاً توان‌بخشی یا آمادگی جسمانی ویژهٔ سالمندان) تنها به بازار کوچک محلهٔ خود دسترسی دارد. دوم، ورزشکاری که به شهر دیگری نقل مکان می‌کند یا باشگاهش را عوض می‌کند، ارتباط با مربی و کل سابقهٔ تمرینی خود را از دست می‌دهد. سوم، امکان ارائهٔ برنامه‌های تمرینی و غذایی از راه دور - که پس از همه‌گیری کووید-۱۹ به یک نیاز واقعی تبدیل شد - در چنین مدلی وجود ندارد.

۳-۲-۱- از دست رفتن سابقه و داده‌های پیشرفت

سابقهٔ تمرینی، اندازه‌های بدنی، وزنه‌های ثبت‌شده و بازخوردهای مربی مجموعاً دارایی ارزشمندی برای هر ورزشکارند؛ اما چون این داده‌ها در اختیار باشگاه یا مربی - و نه خودِ فرد -

نگهداری می‌شود، با هر تغییر از بین می‌رود. نبود این سابقه، طراحی برنامهٔ جدید را دشوار می‌کند و مربی جدید عملاً از صفر شروع می‌کند.

۴-۲-۱- مشکلات عملیاتی و مالی باشگاه

از دید مدیر باشگاه، نبود سامانهٔ یکپارچه به معنای بی‌اطلاعی از وضعیت واقعی اشتراک‌ها، تمدیدهای فراموش‌شده، جمع دستی درآمد و ناتوانی در پاسخ‌گویی به پرسش‌های ساده‌ای مانند «درآمد خالص شش ماه گذشته چقدر بوده است؟» یا «کدام مربی بیشترین شاگرد فعال را دارد؟» است. همچنین بدون ثبت رخدادها [۱]، رسیدگی به اختلاف‌های مالی یا اداری امکان‌پذیر نیست.

۳-۱- هدف پروژه

هدف اصلی این پروژه، طراحی و پیاده‌سازی یک سامانهٔ تحت وب است که سه نقش مدیر باشگاه، مربی و عضو را در یک فضای کاری مشترک قرار دهد و جریان اطلاعات میان آن‌ها را از ابتدا (ثبت‌نام و اشتراک) تا انتها (اجرای تمرین، پایش پیشرفت و گزارش مالی) به‌صورت یکپارچه پوشش دهد. اهداف جزئی‌تر عبارت‌اند از:

- **یکپارچگی داده:** هر رخداد فقط یک بار ثبت شود و همهٔ نقش‌های ذی‌نفع، همان داده را از زاویهٔ خود ببینند.
- **استقلال از مکان:** عضو بتواند در هر باشگاه عضو شود و هر مربیِ پلتفرم را - فارغ از محل - انتخاب کند؛ مربی بتواند با چند باشگاه همکاری کند.
- **مالکیت سابقه توسط فرد:** برنامه‌ها، لاگ‌های تمرین، اندازه‌های بدنی و عکس‌های پیشرفت به حساب کاربری عضو وابسته باشند، نه به باشگاه.
- **اتوماسیون عملیات تکراری:** انقضا و تمدید اشتراک، یادآوری جلسات و هشدار کم‌فعالیتی به‌صورت خودکار انجام شود.
- **شفافیت مالی:** درآمد، بازپرداخت و تمدیدها به تفکیک ارز محاسبه و با خروجی قابل حسابرسی ارائه شود.

۱ Audit log

- **بومی‌سازی کامل:** رابط کاربری فارسیِ راست‌به‌چپ با تقویم جلالی و ارقام فارسی، در کنار نسخهٔ انگلیسی.
- **دسترس‌پذیری روی موبایل:** بیشتر تعامل ورزشکار و مربی روی گوشی رخ می‌دهد؛ بنابراین طراحی واکنش‌گرا الزامی است.
- **امنیت و قابلیت اتکا:** احراز هویت امن، نشست‌های قابل ابطال، محدودیت نرخ درخواست و ثبت رخدادها.

در مقابل، برخی موضوعات آگاهانه خارج از دامنهٔ این پروژه تعریف شدند: اتصال به درگاه پرداخت واقعی (به‌جای آن یک درگاه نمایشی با همان قرارداد پیاده شد)، اپلیکیشن بومی موبایل، و اتصال به ساعت‌های هوشمند. این موارد در فصل چهارم به‌عنوان کارهای آینده بحث شده‌اند.

۴-۱- سناریوهای کاربردی

برای روشن شدن دامنهٔ کار، سه سناریوی نمونه از منظر سه نقش سامانه توصیف می‌شود.

۱-۴-۱- سناریوی مدیر باشگاه

خانم «الف» مدیر یک باشگاه بدن‌سازی است. او در سامانه ثبت‌نام می‌کند، باشگاه خود را می‌سازد و سه طرح اشتراک یک‌ماهه، سه‌ماهه و شش‌ماهه تعریف می‌کند. دو مربی درخواست همکاری می‌دهند که او پس از بررسی پروفایل، هر دو را می‌پذیرد. اعضا از طریق صفحهٔ عمومی باشگاه اشتراک می‌خرند؛ پرداخت‌ها به‌صورت خودکار ثبت می‌شود و خانم «الف» در داشبورد خود درآمد ماهانه، اشتراک‌های در آستانهٔ انقضا و درخواست‌های در انتظار را می‌بیند. پایان هر ماه، گزارش مالی را به‌صورت CSV برای حسابدار خروجی می‌گیرد.

۲-۴-۱- سناریوی مربی

آقای «ب» مربی آمادگی جسمانی است و با دو باشگاه در دو منطقهٔ شهر همکاری می‌کند. او یک برنامهٔ تمرینی چهارروزه به‌عنوان الگو می‌سازد، برای هر شاگرد نسخه‌ای از آن را شخصی‌سازی و تخصیص می‌دهد و بازه‌های در دسترس خود را برای جلسات حضوری اعلام می‌کند. وقتی

شاگردی تمرین را ثبت می‌کند، او در صفحهٔ پیشرفت مراجعان، وزنه‌ها و شدت درک‌شده [1] را می‌بیند و بازخورد می‌نویسد. سامانه شاگردانی را که بیش از یک هفته فعالیتی نداشته‌اند به او یادآوری می‌کند تا پیامی برایشان بفرستد.

۳-۴-۱- سناریوی عضو

«ج» دانشجویی است که تازه به شهر جدیدی آمده است. او در صفحهٔ کشف باشگاه‌ها، باشگاهی نزدیک دانشگاه می‌یابد، درخواست عضویت می‌دهد و پس از تأیید، اشتراک سه‌ماهه می‌خرد. سپس از میان مربیان پلتفرم - نه فقط مربیان همان باشگاه - مربی «ب» را که در رشتهٔ مورد علاقه‌اش تخصص دارد انتخاب می‌کند. برنامهٔ تمرینی روی گوشی او نمایش داده می‌شود؛ هنگام تمرین، ست‌ها و وزنه‌ها را با تایمر استراحت ثبت می‌کند و پس از چند هفته، نمودار پیشرفت و استریک تمرینی خود را می‌بیند. اگر سال بعد باشگاهش را عوض کند، همهٔ این سابقه در حساب او باقی می‌ماند.

۵-۱- مشخصات کلی سامانهٔ ساخته‌شده

سامانهٔ «کوجیما» [2] یک برنامهٔ وب چندنقشی است که قابلیت‌های آن به‌صورت فهرست‌وار در جدول ۱-۲ آمده است. جزئیات فنی هر قابلیت در فصل سوم تشریح می‌شود.

۱ Rating of Perceived Exertion (RPE)

۲ KooGYMaa

جدول ۱-۲ مشخصات کلی و قابلیت‌های سامانهٔ کوجیما

حوزه	قابلیت‌ها
حساب کاربری و امنیت	ثبت‌نام با سه نقش، ورود با «مرا به خاطر بسپار»، فراموشی و بازنشانی گذرواژه، تأیید ایمیل، مدیریت نشست‌های فعال، محدودیت نرخ درخواست، هش bcrypt
مدیر باشگاه	ساخت و تنظیم باشگاه، پذیرش/رد عضو و مربی، عملیات گروهی، طرح‌های اشتراک، مدیریت اشتراک‌ها، بازپرداخت، گزارش مالی به تفکیک ارز، خروجی CSV، ثبت رخداد
مربی	پروفایل حرفه‌ای، همکاری با چند باشگاه، ویرایشگر برنامهٔ تمرینی (روز/حرکت/ست/تکرار)، ویرایشگر برنامهٔ غذایی (روز/وعده/ماده)، الگو و نسخه‌بندی، تخصیص به شاگرد، بازه‌های در دسترس، زمان‌بندی جلسه با تشخیص تداخل، بازخورد، یادآور کم‌فعالیتی
عضو	کشف باشگاه و مربی، درخواست عضویت/مربی، خرید و توقف/ازسرگیری/لغو اشتراک، رسید، اجرای تمرین با تایمر و ثبت ست، ثبت تغذیه، اندازه‌های بدنی، عکس پیشرفت، اهداف، استریک، نمودار پیشرفت، اعلان‌ها و تنظیم یادآوری، امتیازدهی به باشگاه و مربی
پلتفرم	دو زبان (fa/en) با تقویم جلالی و ارقام فارسی، RTL/LTR، حالت روز/شب، واکنش‌گرا (۳۶۰ تا ۱۲۸۰ پیکسل) با منوی همبرگری، صفحات عمومی، سلامت‌سنجی، اتوماسیون کرون، پشتیبان‌گیری
کیفیت	TypeScript سخت‌گیرانه، ۷۶ آزمون واحد، ESLint، آزمون خودکار واکنش‌گرایی روی ۱۹۸ ترکیب صفحه/عرض

از نظر فنی، سامانه با چارچوب Next.js نسخهٔ ۱۶ و React نسخهٔ ۱۹ به زبان TypeScript نوشته شده، از Prisma نسخهٔ ۷ برای دسترسی به پایگاه دادهٔ SQLite/libSQL استفاده می‌کند و بدون وابستگی به سرویس‌های خارجی (حتی فونت وب) قابل استقرار است. کد پایه حدود ۱۶ هزار خط شامل ۴۷ صفحه، ۷۶ مسیر API و ۳۳ جدول پایگاه داده است.

۶-۱- مروری بر کارهای مشابه

نرم‌افزارهای مدیریت باشگاه در بازار جهانی سابقه‌ای طولانی دارند. محصولاتی مانند Mindbody، Glofox و Wodify بر عملیات باشگاه (رزرو کلاس، عضویت، صندوق) تمرکز دارند [Mindbody-W25]؛ در مقابل، ابزارهایی مانند Trainerize و TrueCoach مخصوص مربیان‌اند و برنامهٔ تمرینی و ارتباط مربی-شاگرد را پوشش می‌دهند [Trainerize-W25]. اپلیکیشن‌های سمت کاربر مانند Strong و Hevy نیز صرفاً ثبت تمرین شخصی را انجام می‌دهند. جدول ۱-۳ این دسته‌ها را با پروژهٔ حاضر مقایسه می‌کند.

جدول ۱-۳ مقایسهٔ دسته‌های نرم‌افزاری موجود با کوجیما

ویژگی	نرم‌افزار مدیریت باشگاه	ابزار مربی	اپ ثبت تمرین	کوجیما
مدیریت عضویت و اشتراک باشگاه	✓	-	-	✓
برنامهٔ تمرینی ساخت‌یافته با تخصیص	محدود	✓	خودساخته	✓
برنامهٔ غذایی	-	بعضاً	-	✓
ثبت اجرا توسط ورزشکار	-	✓	✓	✓
مربی مستقل از باشگاه / چندباشگاهی	-	✓	-	✓
سابقهٔ متعلق به فرد	-	وابسته به مربی	✓	✓
گزارش مالی و حسابرسی	✓	-	-	✓
فارسی، تقویم جلالی، ریال	-	-	-	✓
متن‌باز / قابل میزبانی داخلی	-	-	-	✓

چنان‌که دیده می‌شود، هیچ‌یک از دسته‌های موجود هر سه نقش را هم‌زمان پوشش نمی‌دهد؛ افزون بر این، محصولات خارجی به دلیل محدودیت‌های پرداخت بین‌المللی، نبود زبان فارسی و تقویم جلالی، و وابستگی به سرویس‌های ابری خارج از کشور عملاً برای باشگاه‌های ایرانی قابل

استفاده نیستند. نمونه‌های داخلی نیز عمدتاً نرم‌افزارهای صندوق و حضور و غیاب هستند و به سمت مربی و ورزشکار وارد نشده‌اند. این خلأ، انگیزهٔ اصلی تعریف پروژهٔ حاضر بود.

۱-۷- روش انجام کار

پروژه با رویکرد تکرارشونده [۱] انجام شد. ابتدا با مصاحبهٔ غیررسمی با چند مدیر باشگاه و مربی، فهرست نیازها استخراج و اولویت‌بندی شد. سپس مدل دادهٔ دامنه طراحی و در قالب مهاجرت‌های پایگاه داده پیاده شد. قابلیت‌ها در چند تکرار - احراز هویت و نقش‌ها، مدیریت باشگاه، برنامه‌های تمرینی و غذایی، اجرا و پیشرفت، پرداخت و اشتراک، بومی‌سازی و واکنش‌گرایی - افزوده شدند و در پایان هر تکرار، آزمون‌های واحد و بررسی بصری انجام گرفت. کنترل نسخه با Git و بررسی خودکار (بررسی نوع، تحلیل ایستا و آزمون) در هر ارسال کد انجام می‌شد.

۱-۸- ساختار پایان‌نامه

ادامهٔ این نوشتار در سه فصل تنظیم شده است. **فصل دوم** مبانی نظری و فناوری‌های به‌کاررفته را به اندازهٔ لازم برای درک فصل سوم معرفی می‌کند: معماری برنامه‌های وب مدرن، رندر سمت سرور، نگاشت شیء-رابطه، احراز هویت مبتنی بر توکن، اصول طراحی راست‌به‌چپ و واکنش‌گرا. **فصل سوم** بدنهٔ اصلی کار است و در دو بخش ارائه می‌شود: بخش فنی شامل معماری سامانه، مدل داده، جریان‌های اصلی با فلوچارت و نمودار توالی، امنیت، بومی‌سازی و آزمون؛ و بخش راهنمای استفاده که قدم‌به‌قدم و با تصاویر واقعی، کار با سامانه را برای مدیر، مربی و عضو شرح می‌دهد. **فصل چهارم** به جمع‌بندی، محدودیت‌ها، مزایا و معایب فناوری‌های انتخاب‌شده و پیشنهادهایی برای ادامهٔ کار می‌پردازد. در پیوست‌ها فهرست کامل مسیرهای API، جدول‌های پایگاه داده و راهنمای استقرار آمده است.

۱ Iterative

فصل دوم
مبانی نظری و فناوری‌های به‌کاررفته

این فصل مفاهیم و فناوری‌هایی را معرفی می‌کند که درک فصل سوم بر آن‌ها استوار است. هدف، آموزش این فناوری‌ها نیست؛ بلکه بیان مختصر مفاهیم، دلیل انتخاب هر فناوری و ارجاع به مراجع اصلی برای مطالعهٔ بیشتر است. خواننده‌ای که با توسعهٔ وب مدرن آشناست می‌تواند از این فصل عبور کند.

۲-۱- سامانه‌های اطلاعاتی چندنقشی و مدل مجوزدهی

در سامانه‌ای که چند دستهٔ کاربر با اختیارات متفاوت دارد، کنترل دسترسی معمولاً با مدل «کنترل دسترسی مبتنی بر نقش» [۱] پیاده می‌شود [Sandhu-P96]. در این مدل، مجوزها به نقش‌ها و نقش‌ها به کاربران منتسب می‌شوند. کوجیما سه نقش سراسری (مدیر، مربی، عضو) دارد؛ اما بسیاری از مجوزها به «زمینه» [۲] نیز وابسته‌اند: مدیر فقط باشگاه‌های خودش را می‌بیند، مربی فقط شاگردان خودش را و عضو فقط داده‌های خودش را. بنابراین در عمل ترکیبی از RBAC و بررسی مالکیت رکورد به کار رفته است که در بخش ۳-۵ شرح داده می‌شود.

۱ Role-Based Access Control (RBAC)

۲ Context / tenant

۲-۲- معماری برنامه‌های وب مدرن

۲-۲-۱- رندر سمت سرور و مؤلفه‌های سرور

برنامه‌های تک‌صفحه‌ای [۱] که در دهۀ ۲۰۱۰ رایج شدند، کل رابط کاربری را در مرورگر می‌سازند و داده را از API می‌گیرند. این رویکرد بار اولیۀ سنگین و زمان انتظار بیشتری برای کاربر دارد. چارچوب‌های نسل جدید مانند Next.js با «رندر سمت سرور» [۲] صفحۀ HTML آماده را از سرور می‌فرستند و سپس در مرورگر تعاملی می‌کنند [Next-W25]. React از نسخۀ ۱۹ مفهوم «مؤلفۀ سرور» [۳] را معرفی کرد: مؤلفه‌هایی که فقط روی سرور اجرا می‌شوند، مستقیماً به پایگاه داده دسترسی دارند و هیچ کد جاوااسکریپتی به مرورگر نمی‌فرستند [React-W25]. در کوجیما همۀ صفحات پنل‌ها مؤلفۀ سرورند و تنها بخش‌های تعاملی (فرم‌ها، ویرایشگر برنامه، تایمر، منوی موبایل) به‌صورت «جزیره‌های کلاینت» پیاده شده‌اند. این تصمیم حجم جاوااسکریپت ارسالی را کم و امنیت را بیشتر می‌کند، چون منطق دسترسی به داده هرگز مرورگر را ترک نمی‌کند.

۲-۲-۲- مسیرهای API و معماری REST

برای عملیات نوشتنی (ثبت‌نام، ساخت برنامه، پرداخت و...) از مسیرهای API با سبک REST استفاده شده است [Fielding-T00]: هر منبع یک URL دارد، افعال HTTP معنای استاندارد دارند (GET خواندن، POST ایجاد، PATCH ویرایش، DELETE حذف) و پاسخ‌ها با کد وضعیت مناسب (۲۰۰، ۲۰۱، ۴۰۰، ۴۰۱، ۴۰۳، ۴۰۴، ۴۰۹، ۴۲۹) همراه‌اند. جدول ۲-۱ قرارداد کدهای وضعیت در کوجیما را نشان می‌دهد.

۱ Single-Page Application (SPA)
۲ Server-Side Rendering (SSR)
۳ React Server Components (RSC)

جدول ۲-۱ قرارداد کدهای وضعیت HTTP در API کوجیما

کد	معنا	نمونهٔ کاربرد
۲۰۰ / ۲۰۱	موفق / ایجاد شد	دریافت فهرست شاگردان / ساخت برنامهٔ تمرینی
۴۰۰	دادهٔ نامعتبر	ایمیل بدون @، ست منفی، تاریخ نامعتبر
۴۰۱	احراز هویت نشده	کوکی نشست موجود نیست یا باطل شده
۴۰۳	مجاز نیست	مربی می‌خواهد برنامهٔ مربی دیگر را ویرایش کند
۴۰۴	یافت نشد	شناسهٔ باشگاه اشتباه
۴۰۹	تعارض	ایمیل تکراری، تداخل جلسه، پرداخت تکراری
۴۲۹	درخواست بیش از حد	بیش از ۲۰ تلاش ورود در دقیقه

۲-۳- پایگاه داده و نگاشت شیء-رابطه

۲-۳-۱- مدل رابطه‌ای و نرمال‌سازی

داده‌های کوجیما ذاتاً رابطه‌ای‌اند: کاربر با باشگاه، باشگاه با طرح، طرح با اشتراک و اشتراک با پرداخت مرتبط است. مدل رابطه‌ای [Codd-P70] با کلیدهای خارجی و قیود یکتایی، یکپارچگی این روابط را در سطح پایگاه داده تضمین می‌کند. طرحوارهٔ سامانه تا سطح سوم نرمال [۱] طراحی شده است؛ به‌جز چند مورد عمدی مانند ذخیرهٔ «قیمت پرداخت‌شده» در رکورد اشتراک، چون قیمت طرح ممکن است بعداً تغییر کند و سابقهٔ مالی باید ثابت بماند.

۲-۳-۲- SQLite و libSQL

SQLite یک موتور پایگاه دادهٔ رابطه‌ای درون‌فرایندی است که کل پایگاه را در یک فایل نگه می‌دارد، نیازی به سرور جداگانه ندارد و برای بارهای خواندنی متوسط بسیار سریع است [SQLite-W25]. libSQL شاخه‌ای متن‌باز از SQLite است که دسترسی شبکه‌ای و تکثیر [۲] را اضافه می‌کند و سرویس Turso آن را به‌صورت ابری ارائه می‌دهد. انتخاب این ترکیب به کوجیما اجازه می‌دهد در

۱ Third Normal Form (3NF)

۲ Replication

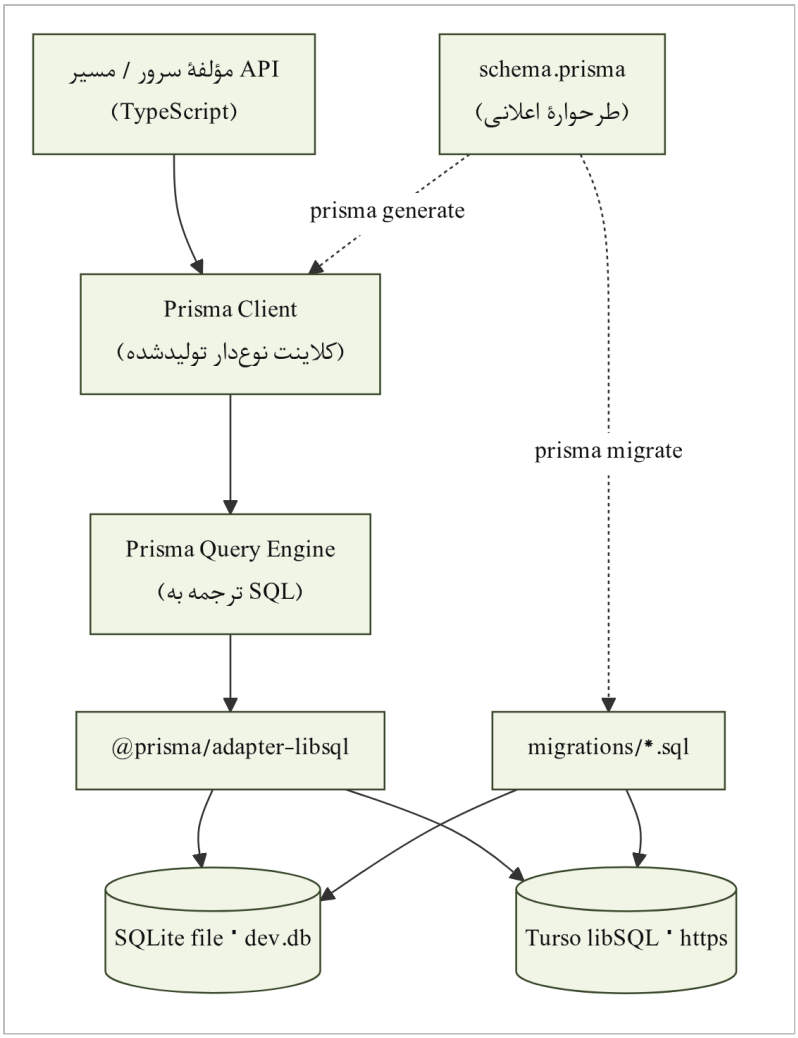
محیط توسعه با یک فایل ساده کار کند و در محیط عملیاتی بدون تغییر کد به پایگاه دادهٔ میزبانی‌شده متصل شود.

۳-۳-۲- Prisma به‌عنوان ORM

نگاشت شیء-رابطه [۱] لایه‌ای است که جدول‌ها را به اشیای زبان برنامه‌نویسی نگاشت می‌کند. Prisma طرحواره را در یک فایل اعلانی (schema.prisma) تعریف می‌کند، از روی آن مهاجرت‌های SQL و یک کلاینت کاملاً نوع‌دار [۲] تولید می‌کند [Prisma-W25]. مزیت اصلی برای این پروژه، کشف خطاهای دسترسی به داده در زمان کامپایل به‌جای زمان اجرا بود؛ برای مثال اگر فیلدی از مدل حذف شود، همهٔ کدهایی که از آن استفاده می‌کنند بلافاصله خطای نوع می‌دهند. شکل ۲-۱ جایگاه این لایه‌ها را نشان می‌دهد.

۱ Object-Relational Mapping (ORM)

۲ Type-safe



شکل ۲-۱ لایه‌های دسترسی به داده: از مؤلفهٔ سرور تا فایل پایگاه داده

۲-۴- احراز هویت و مدیریت نشست

۲-۴-۱- ذخیرهٔ امن گذرواژه

گذرواژه هرگز نباید به‌صورت متن ساده یا حتی با هش سریع (مانند MD5 یا SHA-256) ذخیره شود، زیرا با سخت‌افزار امروزی میلیاردها حدس در ثانیه ممکن است. توابع «مشتق‌سازی کلید» مانند bcrypt عمداً کند و دارای نمک [۱] تصادفی‌اند [Provos-P99]. کوجیما از bcrypt با ضریب هزینهٔ ۱۲ استفاده می‌کند که هر بررسی را حدود ۲۵۰ میلی‌ثانیه طول می‌دهد - برای کاربر نامحسوس، اما برای حمله‌کننده بازدارنده.

۱ Salt

۲-۴-۲- توکن JWT و کوکی HttpOnly

«توکن وب JSON» [۱] رشته‌ای امضاشده است که ادعاهایی [۲] مانند شناسهٔ کاربر و نقش او را حمل می‌کند و سرور می‌تواند بدون مراجعه به پایگاه داده اصالت آن را بررسی کند [Jones-R15]. اگر توکن در کوکی با پرچم HttpOnly ذخیره شود، جاوااسکریپت مرورگر به آن دسترسی ندارد و حملهٔ سرقت توکن از طریق XSS [۳] خنثی می‌شود؛ پرچم SameSite=Lax نیز جلوی بیشتر حملات CSRF [۴] را می‌گیرد [OWASP-W25]. ضعف شناخته‌شدهٔ JWT ناتوانی در ابطال پیش از انقضاست؛ کوجیما این ضعف را با ثبت شناسهٔ نشست در توکن و نگهداری جدول نشست‌های قابل ابطال جبران کرده است (بخش ۳-۵-۲).

۲-۴-۳- محدودیت نرخ درخواست

برای مقابله با حملات حدس گذرواژه و سوءاستفاده از API، تعداد درخواست هر کلاینت در بازهٔ زمانی مشخص محدود می‌شود. الگوریتم «پنجرهٔ ثابت» [۵] ساده‌ترین روش است: برای هر کلید (مثلاً آدرس IP) شمارنده‌ای نگه داشته می‌شود که در پایان هر پنجره صفر می‌شود. کوجیما از این الگوریتم با دو سطل جداگانه برای مسیرهای احراز هویت (۲۰ درخواست در دقیقه) و سایر مسیرهای نوشتنی (۱۲۰ درخواست در دقیقه) استفاده می‌کند.

۲-۵- پرداخت الکترونیک و خودتوانی عملیات

در یکپارچه‌سازی با درگاه پرداخت، دو اصل اهمیت ویژه دارند. نخست، «خودتوانی» [۶]: اگر درخواست پرداخت به هر دلیل تکرار شود (مثلاً کاربر دوبار کلیک کند یا شبکه قطع و وصل شود)، نباید دو پرداخت ثبت شود. راه‌حل استاندارد، همراه کردن هر درخواست با یک «کلید خودتوانی» یکتا و رد کردن کلید تکراری در سطح پایگاه داده است [Stripe-W25]. دوم، «تأیید سمت سرور»:

۱ JSON Web Token (JWT), RFC 7519

۲ Claims

۳ Cross-Site Scripting

۴ Cross-Site Request Forgery

۵ Fixed window

۶ Idempotency

نتیجهٔ پرداخت هرگز نباید بر اساس پارامترهای بازگشتی مرورگر پذیرفته شود؛ بلکه باید از طریق فراخوان برگشتی [1] امضاشدهٔ درگاه یا استعلام مستقیم تأیید شود. کوجیما هر دو اصل را با کلید خودتوانی یکتا در جدول پرداخت و امضای HMAC-SHA256 روی فراخوان برگشتی رعایت می‌کند.

۲-۶- بومی‌سازی، راست‌به‌چپ و تقویم جلالی

بومی‌سازی [2] فراتر از ترجمهٔ رشته‌هاست و جهت نوشتار، قالب اعداد و تاریخ، واحد پول و حتی چیدمان عناصر را در بر می‌گیرد. برای زبان‌های راست‌به‌چپ، CSS «ویژگی‌های منطقی» [3] مانند margin-inline-start را معرفی کرده که به‌جای چپ/راست فیزیکی، بر اساس جهت نوشتار عمل می‌کنند [CSSWG-W25]؛ با این کار یک برگهٔ سبک برای هر دو جهت کافی است. تقویم جلالی (هجری شمسی) تقویم رسمی ایران است و تبدیل دقیق آن به میلادی به الگوریتم‌های شناخته‌شده‌ای نیاز دارد که مبنای ۳۳ ساله و سال‌های کبیسهٔ نامنظم را در نظر بگیرند [Borkowski-P96]. کوجیما هم از API استاندارد Intl با تقویم persian و هم از پیاده‌سازی مستقل برای محاسبات روزشمار استفاده می‌کند تا نتایج در همهٔ محیط‌های اجرا یکسان باشد.

۲-۷- طراحی واکنش‌گرا و دسترس‌پذیری

طراحی واکنش‌گرا [4] یعنی یک صفحه با استفاده از شبکهٔ سیال، تصاویر انعطاف‌پذیر و پرس‌وجوهای رسانه‌ای [5] خود را با عرض‌های مختلف نمایشگر تطبیق دهد [Marcotte-W10]. در کوجیما سه نقطهٔ شکست اصلی تعریف شده است: تا ۶۴۰ پیکسل (گوشی؛ سرصفحهٔ چسبان، منوی همبرگری و نوار زبانه‌ها)، ۶۴۱ تا ۸۲۰ پیکسل (تبلت؛ ریل آیکنی) و بالاتر (رایانه؛ نوار کناری کامل). از نظر دسترس‌پذیری، رهنمودهای WCAG اندازهٔ حداقل ۴۴×۴۴ پیکسل برای اهداف لمسی، نسبت

۱ Webhook
۲ Localization (l10n)
۳ Logical properties
۴ Responsive Web Design
۵ Media queries

کنتراست کافی، برچسب‌گذاری ARIA برای عناصر تعاملی و پشتیبانی از ترجیح «کاهش حرکت» را توصیه می‌کنند [WCAG-W23] که همگی در طراحی رعایت شده‌اند.

۲-۸- آزمون نرم‌افزار

هرم آزمون [Cohn-B09] توصیه می‌کند بیشترین تعداد آزمون در سطح واحد (توابع خالص و سریع)، تعداد کمتری در سطح یکپارچگی و کمترین تعداد در سطح رابط کاربری نوشته شود. کوجیما از این الگو پیروی می‌کند: منطق دامنه (اعتبارسنجی ورودی‌ها، محاسبات مالی، تشخیص تداخل زمانی، استریک، توکن‌های حساب) در توابع خالص جدا شده و با Vitest آزمون می‌شود؛ بررسی نوع TypeScript نقش آزمون یکپارچگی ایستا را ایفا می‌کند؛ و رفتار واکنش‌گرایی رابط کاربری با اسکریپت مرورگر بدون سر [۱] روی همهٔ صفحات بررسی می‌شود.

۲-۹- جمع‌بندی فناوری‌های انتخاب‌شده

جدول ۲-۲ فناوری‌های اصلی پروژه و دلیل انتخاب هر یک را خلاصه می‌کند. ملاک‌های انتخاب عبارت بودند از: بلوغ و پشتیبانی جامعه، نوع‌دار بودن، قابلیت اجرا بدون وابستگی به سرویس خارجی، و هزینهٔ صفر برای میزبانی در مقیاس کوچک.

جدول ۲-۲ فناوری‌های به‌کاررفته و دلیل انتخاب

لایه	فناوری	نسخه	دلیل انتخاب
زبان	TypeScript	۵	کشف خطا در زمان کامپایل، مستندسازی ضمنی قراردادها
چارچوب وب	Next.js (App Router)	۱۶	SSR و مؤلفه‌های سرور، مسیرهای API در همان پروژه، استقرار ساده
رابط کاربری	React	۱۹	مؤلفه‌های سرور، اکوسیستم بزرگ
سبک‌دهی	CSS سفارشی + Tailwind	۴	کنترل کامل RTL و تم، بدون وابستگی به کتابخانهٔ UI
ORM	Prisma	۷	طرحوارهٔ اعلانی، مهاجرت، کلاینت نوع‌دار

۱ Headless browser

پایگاه داده	SQLite / libSQL (Turso)	-	بدون سرور در توسعه، میزبانی ابری بدون تغییر کد
احراز هویت	bcryptjs + jsonwebtoken	۳ / ۹	استاندارد صنعتی، بدون سرویس خارجی
آزمون	Vitest	۴	سریع، سازگار با TypeScript و ESM
فونت	Estedad (رابط) ˙ IRNazanin (سند)	-	آزاد (OFL)، خودمیزبان، بدون نیاز به Google Fonts
ابزار کیفیت	ESLint ˙ tsc ˙ Playwright	-	تحلیل ایستا، بررسی نوع، آزمون مرورگر

فصل سوم
طراحی، پیاده‌سازی و راهنمای استفادهٔ سامانهٔ کوجیما

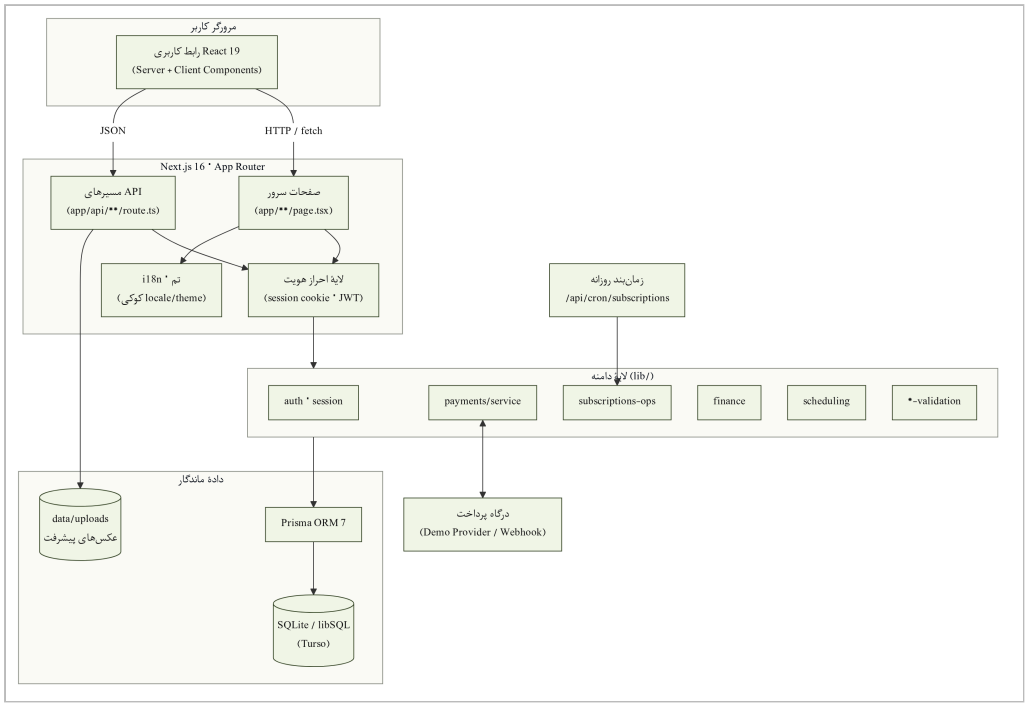
این فصل بدنهٔ اصلی پایان‌نامه است و در دو بخش ارائه می‌شود. بخش نخست (۳-۱ تا ۳-۹) جنبه‌های فنی را پوشش می‌دهد: معماری کلی، مدل داده، جریان‌های اصلی سامانه با فلوچارت و نمودار توالی، امنیت، بومی‌سازی، طراحی واکنش‌گرا، اتوماسیون و آزمون. بخش دوم (۳-۱۰ تا ۳-۱۳) راهنمای استفادهٔ قدم‌به‌قدم برای سه نقش مدیر باشگاه، مربی و عضو است که با تصاویر واقعی از سامانه همراه شده است. برای درک بخش نخست، آشنایی با مفاهیم فصل دوم کافی است.

۳-۱- معماری کلی سامانه

کوجیما یک برنامهٔ وب یکپارچه [۱] است که رابط کاربری، منطق دامنه و API را در یک پروژهٔ Next.js نگه می‌دارد. شکل ۳-۱ بلوک‌دیاگرام کلی سامانه را در چهار لایه نشان می‌دهد: مرورگر، چارچوب Next.js (صفحات سرور و مسیرهای API)، لایهٔ دامنه و دادهٔ ماندگار. جریان یک درخواست معمولی چنین است: مرورگر صفحه‌ای را درخواست می‌کند؛ مؤلفهٔ سرور مربوط پس از خواندن نشست کاربر، مستقیماً از طریق Prisma داده را می‌خواند و HTML آماده را بازمی‌گرداند. عملیات

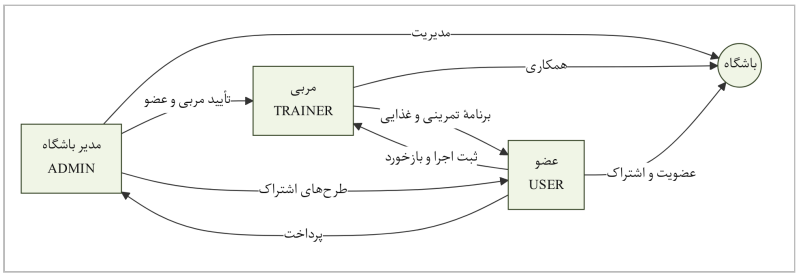
۱ Monolith

نوشتنی از طریق جزیره‌های کلاینت به مسیرهای API ارسال می‌شود که پس از احراز هویت، محدودیت نرخ و اعتبارسنجی، توابع لایهٔ دامنه را فرامی‌خوانند.



شکل ۳-۱ بلوک‌دیاگرام کلی سامانهٔ کوجیما در سطح لایه‌ها

این معماری در برابر معماری ریزسرویس [۱] انتخاب شد، زیرا برای تیمی کوچک و مقیاس هدف (ده‌ها باشگاه و هزاران کاربر) هزینهٔ عملیاتی بسیار کمتری دارد و همچنان با جداسازی منطق دامنه در پوشهٔ /lib، در صورت نیاز آینده قابل تفکیک است. شکل ۳-۲ ارتباط سه نقش سامانه با یکدیگر و با موجودیت باشگاه را نشان می‌دهد.



شکل ۳-۲ نقش‌های سامانه و تعامل آن‌ها حول موجودیت باشگاه

۱ Microservices

۳-۱-۱- ساختار پروژه

ساختار پوشه‌های پروژه از قراردادهای App Router پیروی می‌کند و در جدول ۳-۱ خلاصه شده است. نکتهٔ کلیدی این ساختار، تفکیک صریح سه لایه است: /app فقط مسیریابی و ترکیب صفحه را بر عهده دارد، /components واحدهای رابط کاربری قابل استفادهٔ مجدد را نگه می‌دارد و /lib منطق دامنه را بدون هیچ وابستگی به React پیاده می‌کند تا مستقلاً قابل آزمون باشد.

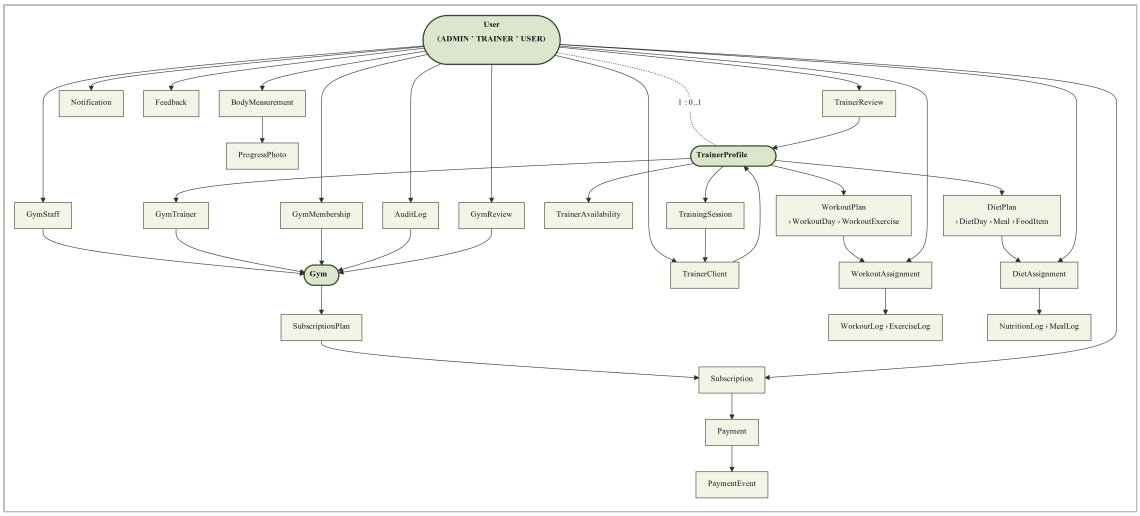
جدول ۳-۱ ساختار پوشه‌های اصلی پروژه

مسیر	محتوا	تعداد فایل
app/	صفحات (page.tsx)، چیدمان‌ها (layout.tsx)، مسیرهای API (route.ts) به تفکیک نقش: admin، trainer، user، auth، cron، webhooks	۴۷ صفحه، ۷۶ مسیر API
components/	مؤلفه‌های رابط کاربری: ناوبری هر نقش، فرم‌ها، ویرایشگرهای برنامه، تقویم جلسات، نمودار پیشرفت، منوی موبایل، سوییچ زبان و تم	۵۰+
lib/	منطق دامنه: احراز هویت و نشست، اعتبارسنجی‌ها، مالی، زمان‌بندی، اشتراک، استریک، تقویم جلالی، i18n، پرداخت، محدودیت نرخ، پایش	۴۰+
prisma/	طرحواره (۳۳ مدل، ۱۶ شمارش)، ۶ مهاجرت SQL، اسکریپت دادهٔ اولیه	-
tests/	آزمون‌های واحد Vitest	۱۸ فایل، ۷۶ آزمون
docs/	مستندات پایگاه داده، پرداخت و عملیات	۳
public/fonts/	فونت خودمیزبان Estedad (OFL)	-

۳-۲- مدل داده

طرحوارهٔ پایگاه داده ۳۳ مدل و ۱۶ نوع شمارشی دارد و در پنج خوشهٔ مفهومی سازمان یافته است: هویت و عضویت، اشتراک و پرداخت، برنامه‌های تمرینی و غذایی، اجرا و پیشرفت، و ارتباطات و

حسابرسی. شکل ۳-۳ نمای کلی روابط میان همهٔ مدل‌ها را نشان می‌دهد و شکل‌های ۳-۴ و ۳-۵ دو خوشهٔ اصلی را با جزئیات فیلدها ترسیم می‌کنند.



شکل ۳-۳ نمای کلی روابط میان مدل‌های پایگاه داده (نمودار موجودیت-رابطه)

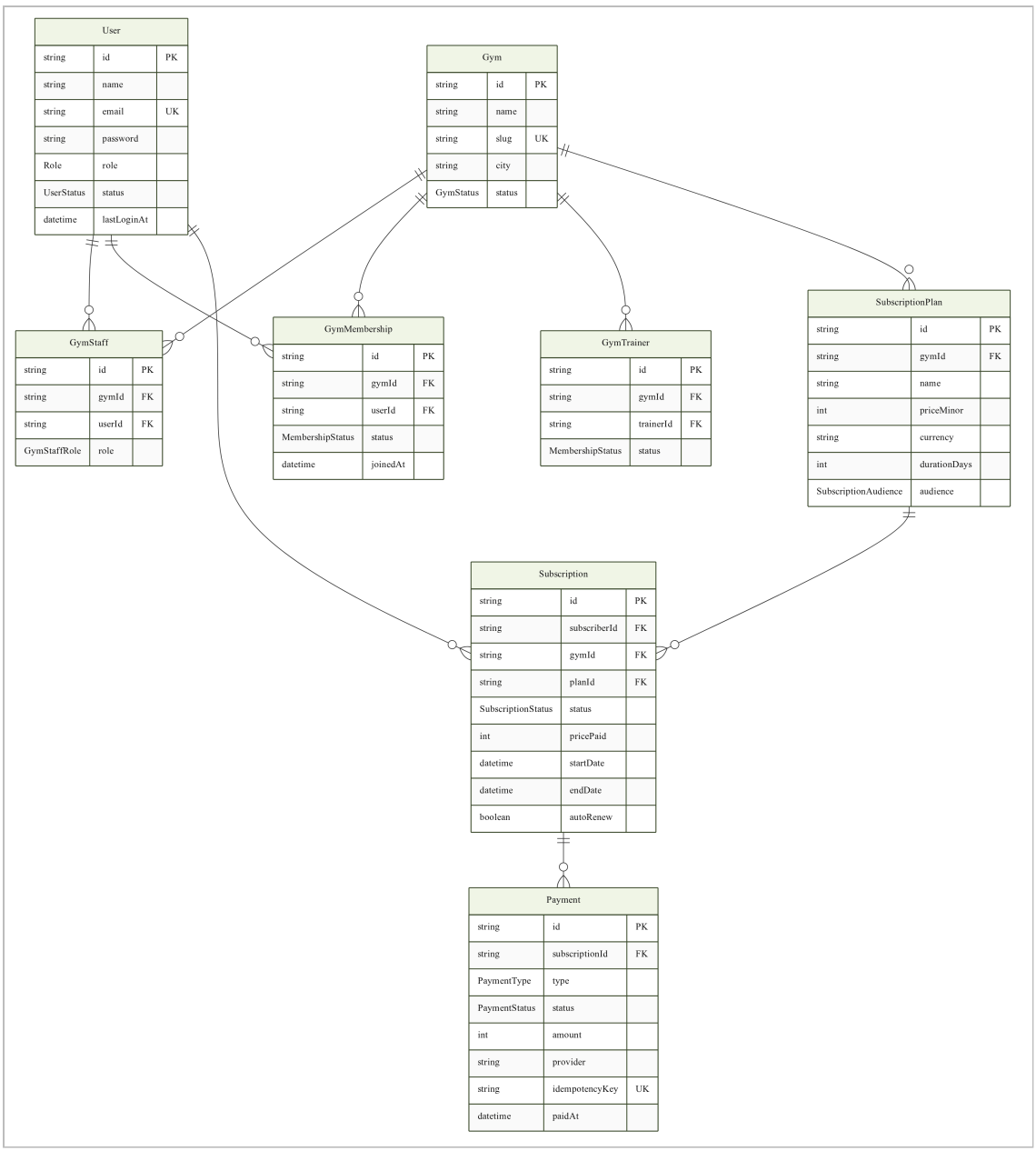
۳-۲-۱- هویت، عضویت و اشتراک

مدل User هستهٔ سامانه است و با فیلد role یکی از سه نقش را می‌گیرد. رابطهٔ کاربر با باشگاه بسته به نقش از سه جدول متفاوت می‌گذرد: GymStaff برای مدیران، GymTrainer برای مربیان و GymMembership برای اعضا. این تفکیک به مربی اجازه می‌دهد هم‌زمان با چند باشگاه همکاری کند و به عضو اجازه می‌دهد در چند باشگاه عضو باشد. رابطهٔ مستقیم مربی و شاگرد - مستقل از باشگاه - در TrainerClient نگهداری می‌شود که یکی از تصمیم‌های کلیدی طراحی برای رفع «محدودیت جغرافیایی» بیان‌شده در فصل اول است.

هر باشگاه چند SubscriptionPlan با قیمت (به کوچک‌ترین واحد پول، مثلاً ریال)، ارز، مدت و مخاطب (عضو یا مربی) تعریف می‌کند. خرید یک طرح، رکورد Subscription با وضعیت PENDING و یک Payment می‌سازد. قیمت پرداخت‌شده در خود اشتراک کپی می‌شود تا تغییر بعدی قیمت طرح، سابقهٔ مالی را خراب نکند. رخدادهای درگاه در PaymentEvent ثبت می‌شوند و کلید خودتوانی یکتا از پرداخت تکراری جلوگیری می‌کند.

۲-۲-۳- برنامه‌های تمرینی و غذایی

برنامهٔ تمرینی ساختاری سه‌سطحی دارد: WorkoutPlan ← WorkoutDay ← WorkoutExercise. هر حرکت نام، تعداد ست، الگوی تکرار (مثلاً «۸-۱۲») و زمان استراحت دارد. برنامه می‌تواند «الگو» باشد و با عمل کپی برای شاگرد خاص شخصی‌سازی شود؛ فیلد version امکان نگهداری نسخه‌های متوالی را می‌دهد. تخصیص برنامه به شاگرد از طریق WorkoutAssignment با تاریخ شروع و وضعیت انجام می‌شود. برنامهٔ غذایی ساختار مشابهی دارد: DietPlan ← DietDay ← Meal ← FoodItem با هدف کالری. شکل ۳-۵ این خوشه را همراه با جدول‌های ثبت اجرا نشان می‌دهد.



شکل ۳-۴ خوشهٔ هویت، عضویت، اشتراک و پرداخت با فیلدهای اصلی

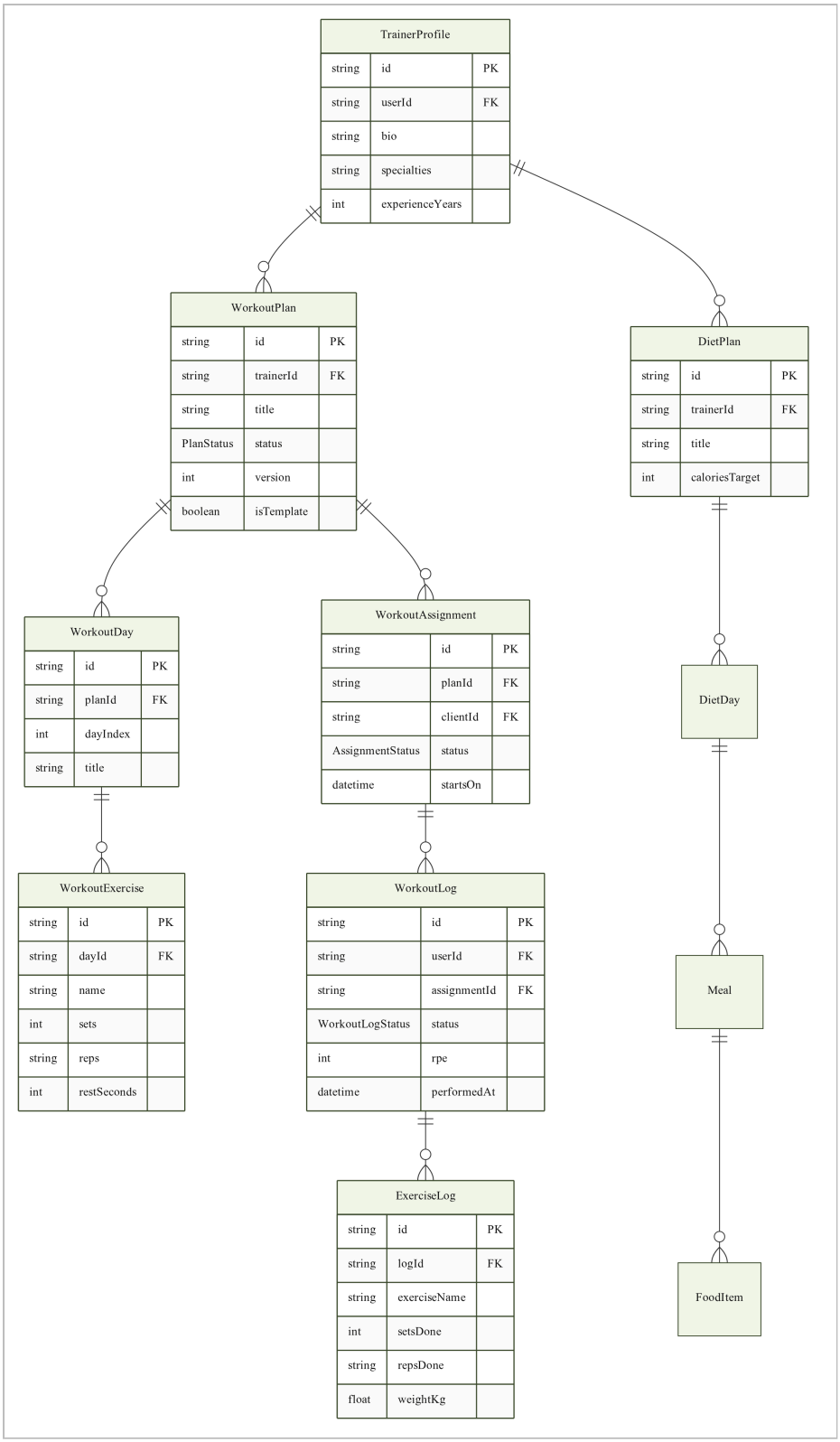
۳-۲-۳- اجرا، پیشرفت و ارتباطات

وقتی عضو تمرینی را انجام می‌دهد، یک WorkoutLog با وضعیت، شدت درک‌شده و زمان اجرا و به‌ازای هر حرکت یک ExerciseLog (ست، تکرار و وزنهٔ واقعی) ثبت می‌شود. ثبت تغذیه با MealLog/NutritionLog، اندازه‌های بدنی با BodyMeasurement و عکس‌ها با ProgressPhoto نگهداری می‌شوند. همهٔ این جدول‌ها کلید خارجی به User دارند نه به باشگاه - تجلی اصل

«مالکیت سابقه توسط فرد». ارتباطات شامل Notification (پنج نوع)، Feedback (گفت‌وگوی مربی و شاگرد روی یک لاگ)، GymReview و TrainerReview است و AuditLog هر عمل اداری را با کنشگر، موجودیت و جزئیات ثبت می‌کند.

۴-۲-۳- انواع شمارشی و وضعیت‌ها

وضعیت‌های چرخهٔ عمر موجودیت‌ها به‌صورت نوع شمارشی در طرحواره تعریف شده‌اند تا مقادیر نامعتبر در سطح پایگاه داده رد شوند. جدول ۳-۲ مهم‌ترین آن‌ها را فهرست می‌کند.



شکل ۳-۵ خوشهٔ برنامه‌های تمرینی و غذایی و ثبت اجرای آن‌ها

جدول ۳-۲ انواع شمارشی اصلی طرحواره

شمارش	مقادیر	کاربرد
Role	ADMIN, TRAINER, USER	نقش سراسری کاربر
MembershipStatus	PENDING, APPROVED, REJECTED, SUSPENDED	درخواست عضویت عضو / همکاری مربی در باشگاه
TrainerClientStatus	PENDING, ACTIVE, PAUSED, ENDED	رابطهٔ مربی-شاگرد
SubscriptionStatus	PENDING, ACTIVE, PAUSED, EXPIRED, CANCELLED	چرخهٔ عمر اشتراک
PaymentStatus	PENDING, PROCESSING, SUCCEEDED, FAILED, REFUNDED, CANCELLED	چرخهٔ عمر پرداخت
PaymentType	SUBSCRIPTION, RENEWAL	تفکیک خرید اولیه از تمدید
PlanStatus	DRAFT, PUBLISHED, ARCHIVED	برنامهٔ تمرینی/غذایی
SessionStatus	SCHEDULED, COMPLETED, CANCELLED, NO_SHOW	جلسهٔ حضوری
NotificationType	PLAN_ASSIGNED, FEEDBACK, SESSION_REMINDER, SUBSCRIPTION_EXPIRING, GENERAL	دسته‌بندی اعلان‌ها

۳-۳- احراز هویت و مدیریت حساب

۳-۳-۱- ثبت‌نام

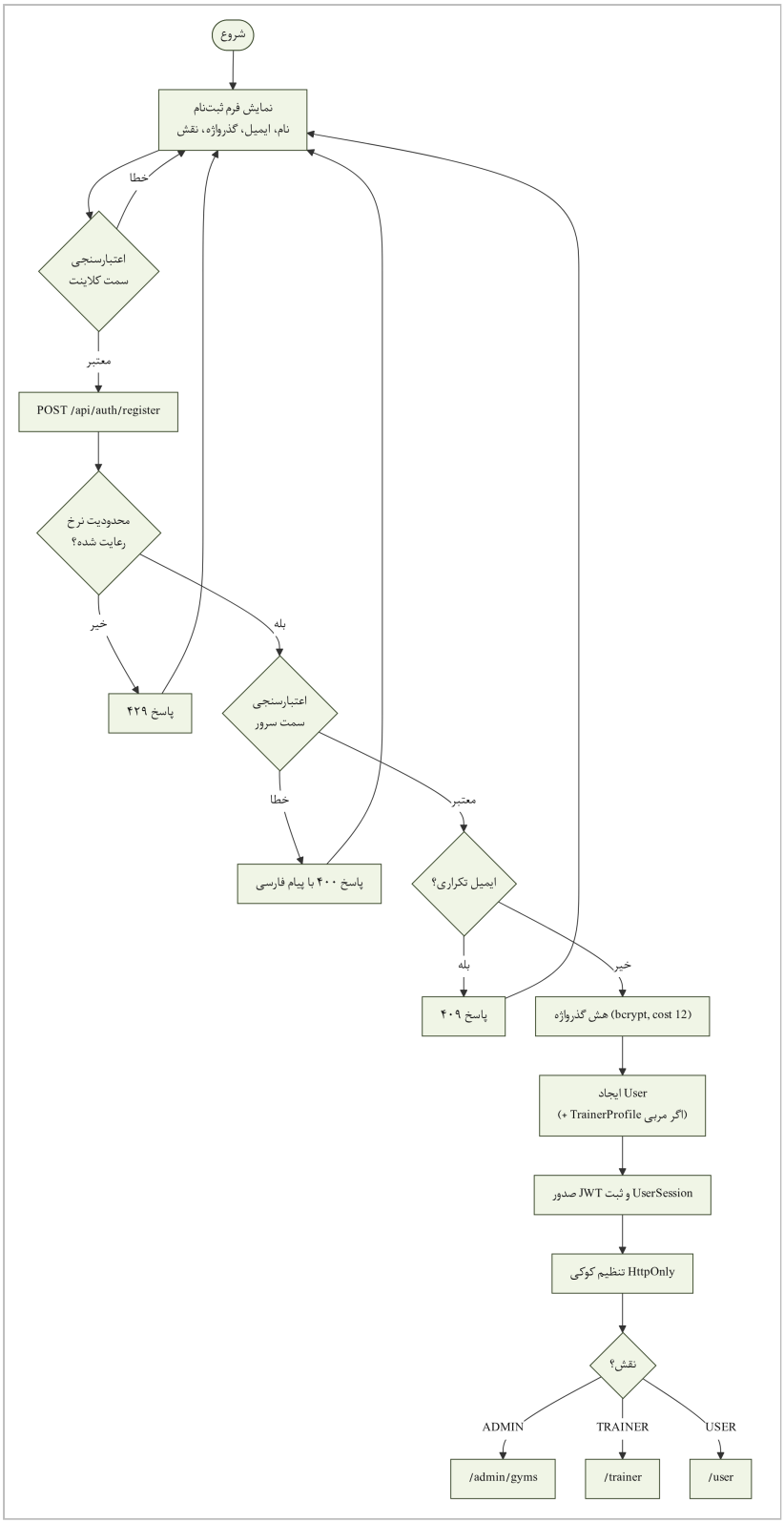
فلوچارت شکل ۳-۶ فرایند ثبت‌نام را نشان می‌دهد. اعتبارسنجی دو بار انجام می‌شود: در مرورگر برای بازخورد فوری و در سرور به‌عنوان مرجع نهایی (چون ورودی مرورگر هرگز قابل اعتماد نیست). قواعد اعتبارسنجی در lib/auth-validation.ts به‌صورت توابع خالص نوشته شده و مستقلاً آزمون

می‌شوند. اگر نقش انتخابی «مربی» باشد، هم‌زمان با کاربر یک TrainerProfile خالی نیز ساخته می‌شود. پس از ثبت موفق، کاربر بلافاصله وارد سامانه شده و به پنل نقش خود هدایت می‌شود.

۲-۳-۳- ورود و نشست

شکل ۳-۷ فلوچارت ورود و شکل ۳-۸ نمودار توالی آن را در تعامل با لایه‌های داخلی نشان می‌دهد. سه نکتهٔ امنیتی در این جریان قابل توجه است. نخست، اگر ایمیل واردشده وجود نداشته باشد، سامانه باز هم یک مقایسهٔ bcrypt با هش ساختگی انجام می‌دهد تا زمان پاسخ برای ایمیل موجود و ناموجود یکسان باشد و حمله‌کننده نتواند از اختلاف زمان، وجود حساب را تشخیص دهد [۱]. دوم، برای هر ورود موفق یک رکورد UserSession با آدرس IP و عامل کاربر ساخته و شناسهٔ آن در ادعاهای JWT قرار می‌گیرد؛ در هر درخواست بعدی، ابطال‌نشده بودن این نشست بررسی می‌شود. سوم، مدت اعتبار کوکی بسته به انتخاب «مرا به خاطر بسپار» ۲۴ ساعت یا ۷ روز است.

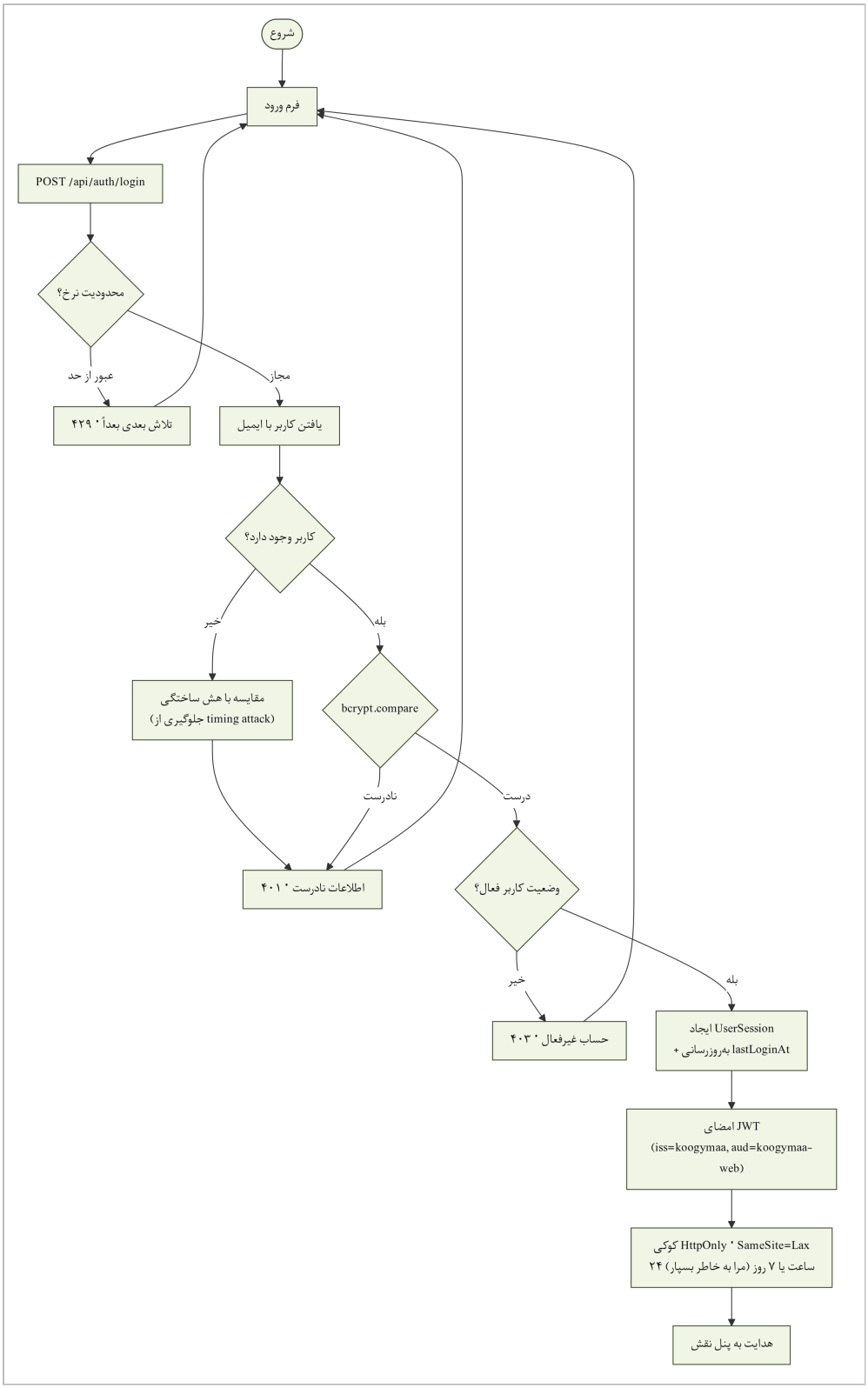
۱ Timing attack mitigation



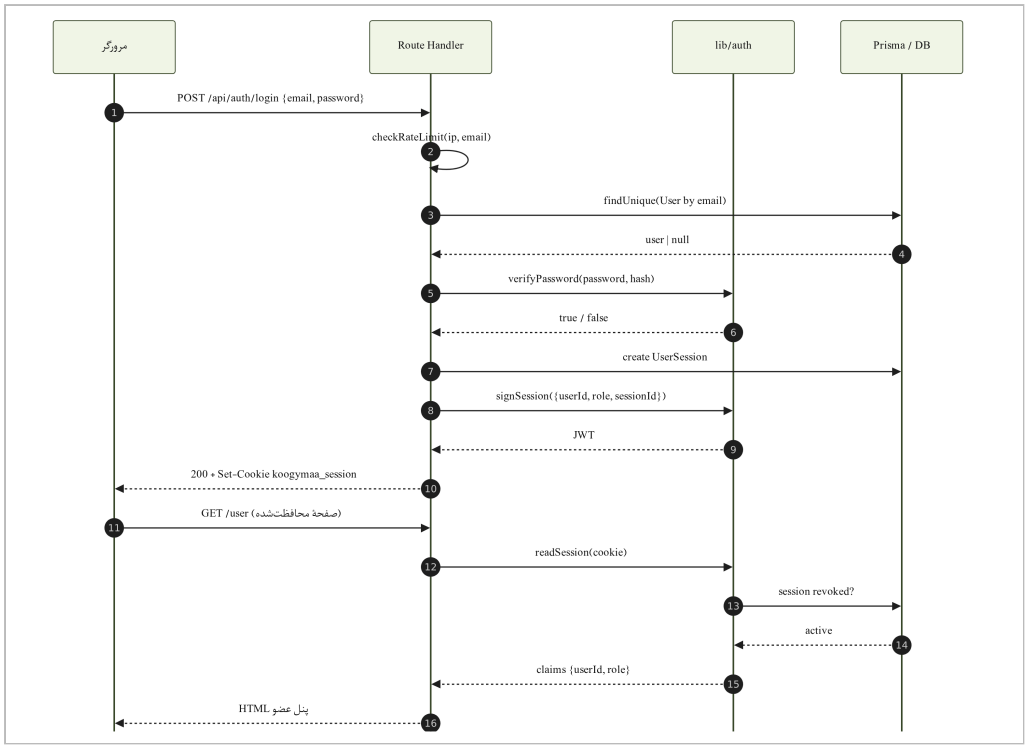
شکل ۳-۶ فلوچارت فرایند ثبت‌نام

۳-۳-۳- بازیابی حساب

فراموشی گذرواژه با توکن یک‌بارمصرف پیاده شده است: سرور یک توکن تصادفی می‌سازد، هش آن را همراه با زمان انقضا (۳۰ دقیقه) ذخیره می‌کند و توکن خام را برای کاربر می‌فرستد. هنگام بازنشانی، هش توکن دریافتی با مقدار ذخیره‌شده مقایسه می‌شود؛ بنابراین حتی با دسترسی به پایگاه داده نمی‌توان توکن معتبر ساخت. همین سازوکار برای تأیید ایمیل به کار رفته است. کاربر همچنین می‌تواند از داخل پنل، گذرواژه را تغییر دهد و فهرست نشست‌های فعال خود را ببیند و هر یک را (یا همه به‌جز نشست فعلی را) باطل کند.



شکل ۳-۷ فلوچارت فرایند ورود

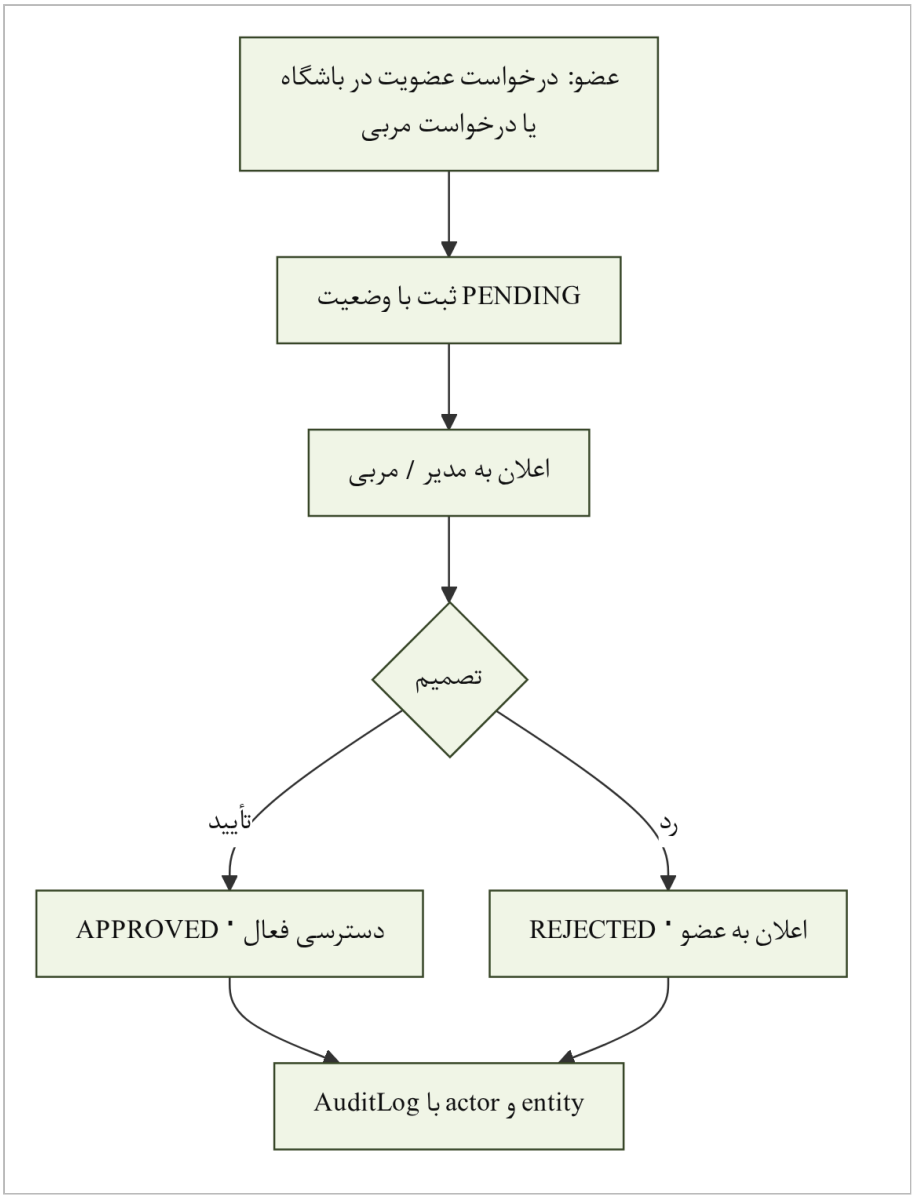


شکل ۳-۸ نمودار توالی ورود و دسترسی به صفحهٔ محافظت‌شده

۳-۴- جریان‌های اصلی کسب‌وکار

۳-۴-۱- درخواست عضویت و همکاری

ورود عضو به باشگاه و پیوستن مربی به باشگاه هر دو با الگوی «درخواست-تأیید» انجام می‌شود (شکل ۳-۹). درخواست با وضعیت PENDING ثبت و اعلانی برای مدیر ساخته می‌شود. مدیر می‌تواند تک‌تک یا گروهی تصمیم بگیرد؛ هر تصمیم در AuditLog ثبت می‌شود و به درخواست‌دهنده اعلان می‌رود. همین الگو برای درخواست مربی‌گری از سوی عضو نیز به کار می‌رود، با این تفاوت که تصمیم‌گیرنده خودِ مربی است.



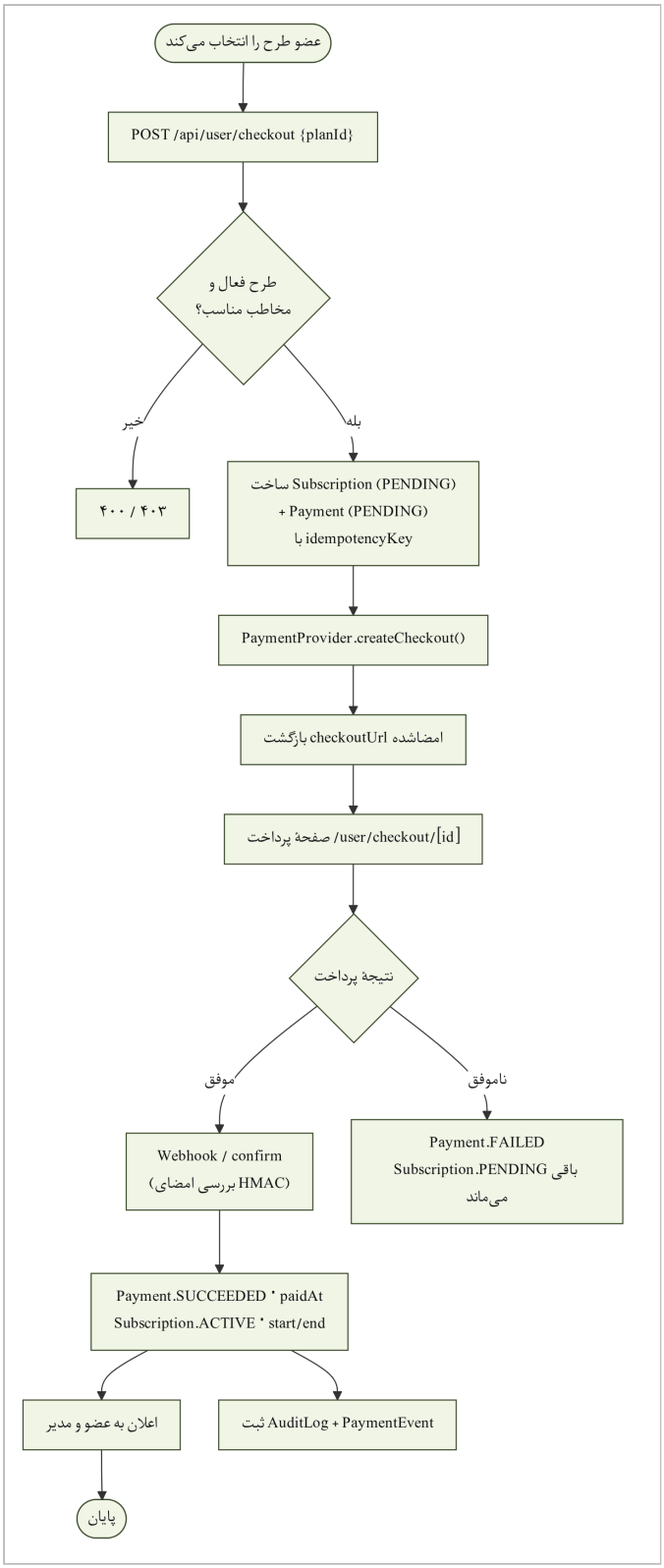
شکل ۳-۹ الگوی درخواست-تأیید برای عضویت، همکاری مربی و رابطهٔ مربی-شاگرد

۲-۴-۳- خرید اشتراک و پرداخت

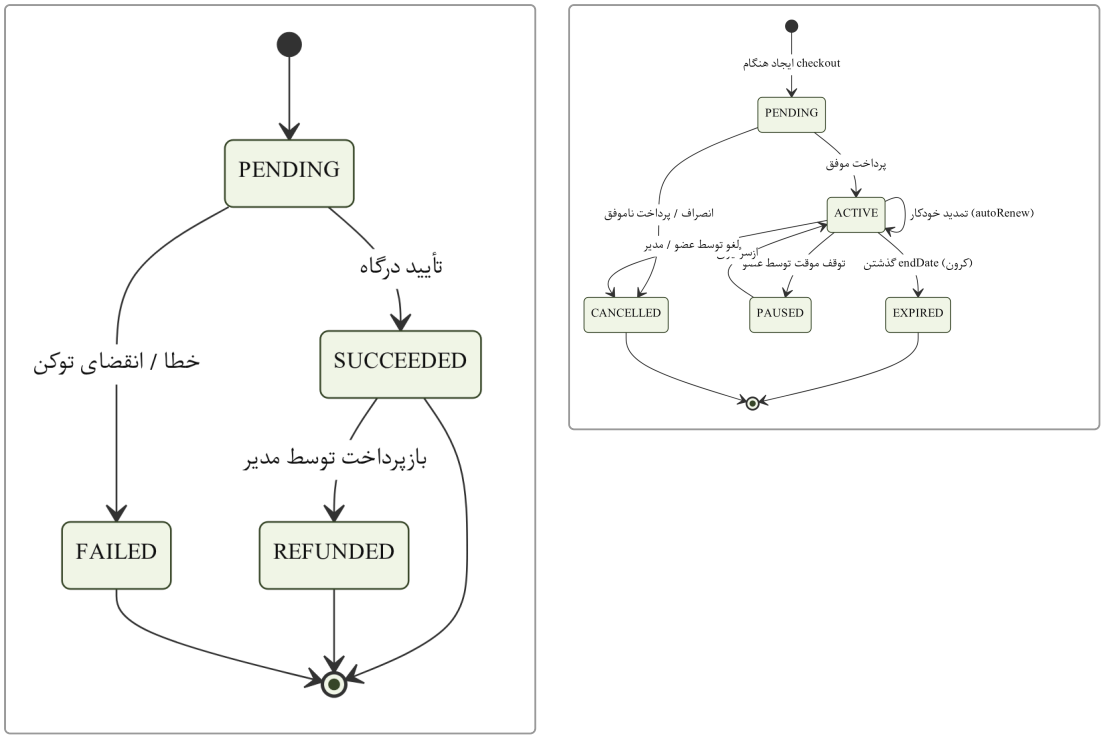
شکل ۳-۱۰ فلوچارت کامل خرید اشتراک را نشان می‌دهد. سرویس پرداخت (lib/payments/service.ts) پشت یک واسط PaymentProvider قرار دارد که فقط یک متد createCheckout دارد؛ در نسخهٔ فعلی یک «درگاه نمایشی» این واسط را پیاده می‌کند که توکن امضاشده با HMAC می‌سازد و صفحهٔ پرداخت داخلی را نمایش می‌دهد. اتصال به درگاه واقعی (مثلاً زرین‌پال یا درگاه‌های بانکی) تنها به پیاده‌سازی همین واسط نیاز دارد. نتیجهٔ پرداخت از

طریق مسیر /api/webhooks/payments/[provider] دریافت و امضای آن با timingSafeEqual بررسی می‌شود؛ سپس در یک تراکنش پایگاه داده، پرداخت SUCCEEDED و اشتراک ACTIVE می‌شود و تاریخ پایان بر اساس مدت طرح محاسبه می‌گردد.

نمودارهای حالت شکل ۳-۱۱ چرخهٔ عمر اشتراک و پرداخت را نشان می‌دهند. گذار «توقف موقت» به عضو اجازه می‌دهد در سفر یا بیماری، روزهای باقی‌ماندهٔ اشتراک را حفظ کند؛ هنگام ازسرگیری، تاریخ پایان به اندازهٔ مدت توقف جلو می‌رود. بازپرداخت فقط توسط مدیر و فقط برای پرداخت موفق ممکن است و در گزارش مالی از درآمد خالص کسر می‌شود.



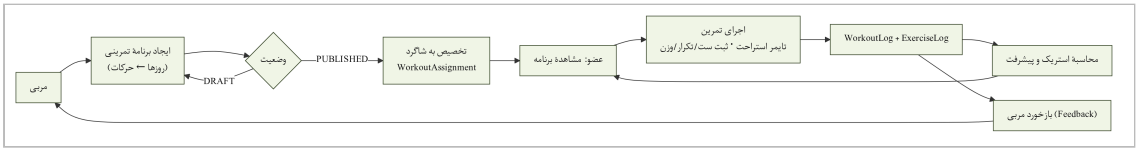
شکل ۳-۱۰ فلوچارت خرید اشتراک و تأیید پرداخت



شکل ۳-۱۱ نمودار حالت اشتراک (راست) و پرداخت (چپ)

۳-۴-۳- چرخهٔ برنامهٔ تمرینی: از طراحی تا بازخورد

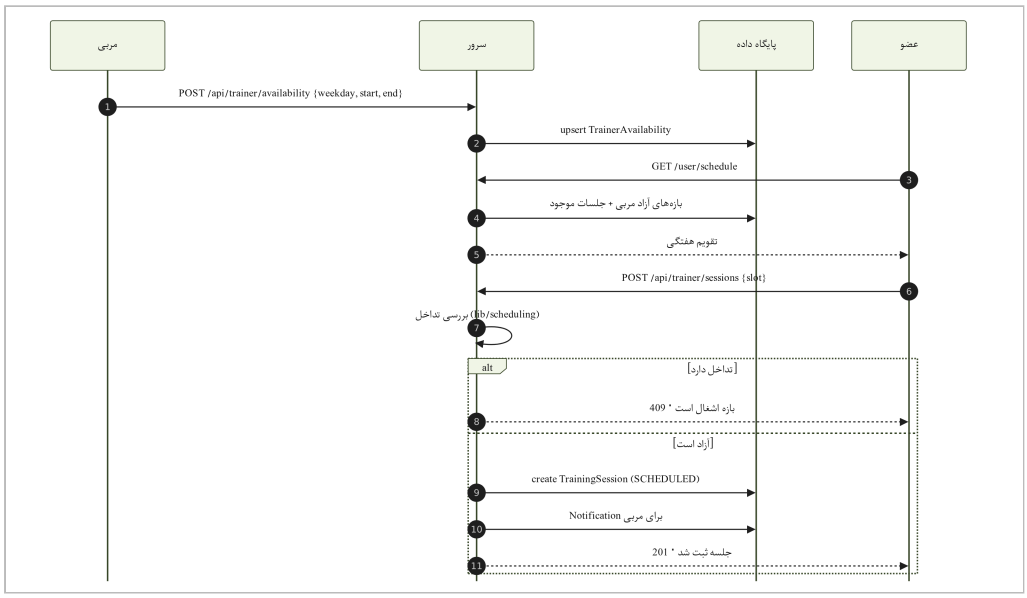
شکل ۳-۱۲ چرخهٔ کامل یک برنامهٔ تمرینی را نشان می‌دهد. مربی برنامه را در حالت پیش‌نویس می‌سازد، پس از انتشار آن را به یک یا چند شاگرد تخصیص می‌دهد، شاگرد اجرا و ثبت می‌کند و لاگ‌ها هم برای محاسبهٔ استریک و نمودار پیشرفت عضو و هم برای بازخورد مربی استفاده می‌شوند. ویرایشگر برنامه (workout-plan-editor.tsx) یک مؤلفهٔ کلاینت است که کل ساختار روز/حرکت را در حافظه نگه می‌دارد و با یک درخواست ذخیره می‌کند؛ کتابخانهٔ حرکات (lib/exercises.ts) با جست‌وجوی فارسی/انگلیسی به انتخاب سریع کمک می‌کند.



شکل ۳-۱۲ چرخهٔ برنامهٔ تمرینی از طراحی توسط مربی تا بازخورد

۳-۴-۴- زمان‌بندی جلسات حضوری

مربی بازه‌های در دسترس هفتگی خود را اعلام می‌کند و عضو از میان آن‌ها جلسه رزرو می‌کند (شکل ۳-۱۳). پیش از ثبت، تابع findConflicts در lib/scheduling.ts تداخل بازهٔ درخواستی را با جلسات موجود مربی بررسی و تابع isWithinAvailability قرار گرفتن آن در بازهٔ اعلام‌شده را تأیید می‌کند. هر دو تابع خالص‌اند و با آزمون واحد پوشش داده شده‌اند. یادآوری جلسه بر اساس تنظیم شخصی هر کاربر (پیش‌فرض ۲۴ ساعت قبل) به‌صورت اعلان ارسال می‌شود.



شکل ۳-۱۳ نمودار توالی اعلام بازهٔ در دسترس و رزرو جلسه با تشخیص تداخل

۳-۵- امنیت و کنترل دسترسی

۳-۵-۱- لایه‌های دفاعی

امنیت سامانه به‌صورت چندلایه [۱] طراحی شده و جدول ۳-۳ این لایه‌ها را خلاصه می‌کند. هر مسیر API ابتدا از requireSession می‌گذرد که کوکی را می‌خواند، امضای JWT را با صادرکننده و مخاطب مشخص بررسی می‌کند و ابطال‌نشده بودن نشست را می‌سنجد. سپس توابع دسترسی

۱ Defense in depth

مخصوص نقش (admin-access.ts، trainer-access.ts) مالکیت رکورد را بررسی می‌کنند؛ مثلاً requireGymAdmin(gymId) تأیید می‌کند کاربر فعلی در GymStaff آن باشگاه باشد.

جدول ۳-۳ لایه‌های امنیتی سامانه

لایه	سازوکار	محل پیاده‌سازی
ذخیرهٔ گذرواژه	bcrypt با ضریب هزینهٔ ۱۲، هش ساختگی برای یکسان‌سازی زمان پاسخ	lib/auth.ts
نشست	JWT امضاشده (iss/aud)، کوکی HttpOnly + SameSite=Lax، جدول نشست قابل ابطال	lib/session.ts, lib/user-sessions.ts
محدودیت نرخ	پنجرهٔ ثابت؛ ۲۰/دقیقه برای احراز هویت، ۱۲۰/دقیقه برای API عمومی	lib/rate-limit.ts
اعتبارسنجی ورودی	توابع خالص برای هر دامنه؛ پیام خطای فارسی	lib/*-validation.ts
کنترل دسترسی	نقش سراسری + بررسی مالکیت رکورد (باشگاه، شاگرد، لاگ)	lib/admin-access.ts, trainer-access.ts
پرداخت	کلید خودتوانی یکتا، امضای HMAC-SHA256 با مقایسهٔ زمان‌ثابت	lib/payments/
حسابرسی	ثبت کنشگر، موجودیت، عمل و جزئیات برای هر عمل اداری	lib/audit.ts
آپلود فایل	بررسی نوع MIME و اندازه، نام‌گذاری تصادفی، خارج از ریشهٔ وب	progress-photos/upload
پایش	بافر حلقوی خطاهای سرور، شاخص‌های سلامت	lib/monitoring.ts, /api/health

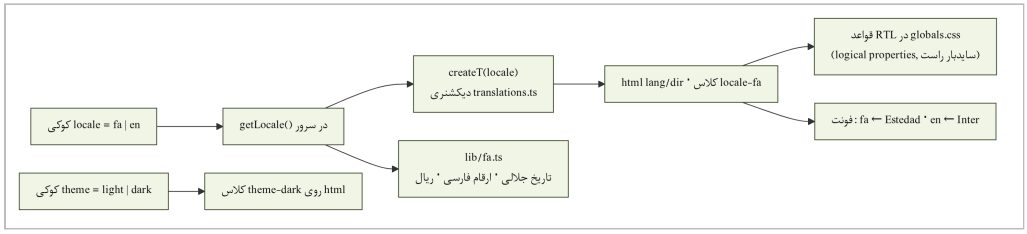
۲-۵-۳- ابطال نشست

چنان‌که در فصل دوم گفته شد، JWT به‌خودی‌خود قابل ابطال نیست. راه‌حل کوجیما، قرار دادن شناسهٔ نشست در توکن و نگهداری جدول UserSession با فیلد revokedAt است. هزینهٔ این کار یک

پرس‌وجوی سبک به‌ازای هر درخواست است که با شاخص روی کلید اصلی قابل چشم‌پوشی است. در مقابل، کاربر می‌تواند از صفحهٔ پروفایل، دستگاه‌های واردشده را ببیند و «خروج از سایر دستگاه‌ها» را بزند - قابلیتی که برای حساب‌های مدیر باشگاه اهمیت ویژه دارد.

۳-۶- بومی‌سازی: فارسی، راست‌به‌چپ و تقویم جلالی

سامانه به‌طور کامل دوزبانه است و زبان فعال در کوکی locale نگهداری می‌شود (شکل ۳-۱۴). در سرور، تابع getLocale() این کوکی را می‌خواند و createT(locale) تابع ترجمه را از دیکشنری translations.ts (بیش از ۵۰۰ کلید برای هر زبان) می‌سازد؛ در کلاینت، همین دیکشنری از طریق Context در دسترس است. عنصر html ویژگی‌های lang و dir و کلاس locale-fa را می‌گیرد و بقیهٔ کار را CSS انجام می‌دهد.



شکل ۳-۱۴ جریان بومی‌سازی از کوکی تا رندر نهایی

۳-۶-۱- چیدمان راست‌به‌چپ

به‌جای نگهداری دو برگهٔ سبک، از ویژگی‌های منطقی CSS استفاده شده است؛ در نتیجه نوار کناری پنل‌ها در فارسی به‌طور خودکار در سمت راست و در انگلیسی در سمت چپ قرار می‌گیرد. تنها استثناها عناصر تزئینی هستند (مانند نشان کارت صفحهٔ اصلی) که با موقعیت فیزیکی ثابت طراحی شده‌اند. آیکن‌های جهت‌دار (فلش‌ها) در حالت فارسی با scaleX(-1) آینه می‌شوند.

۳-۶-۲- تاریخ، اعداد و پول

ماژول lib/fa.ts سه دسته تابع دارد: تبدیل میلادی به جلالی (toJalali) با الگوریتم مستقل و قالب‌بندی با Intl.DateTimeFormat('fa-IR-u-ca-persian') در منطقهٔ زمانی تهران؛ تبدیل ارقام

لاتین به فارسی (toFaDigits) برای متن‌های ترکیبی؛ و قالب‌بندی پول (formatMoney) که مبلغ را از کوچک‌ترین واحد به ریال/تومان با جداکنندهٔ هزارگان فارسی تبدیل می‌کند. برچسب‌های وضعیت که از انواع شمارشی پایگاه داده می‌آیند، از طریق نگاشت‌های faStatus و مانند آن ترجمه می‌شوند تا هیچ مقدار انگلیسی خامی در رابط فارسی دیده نشود.

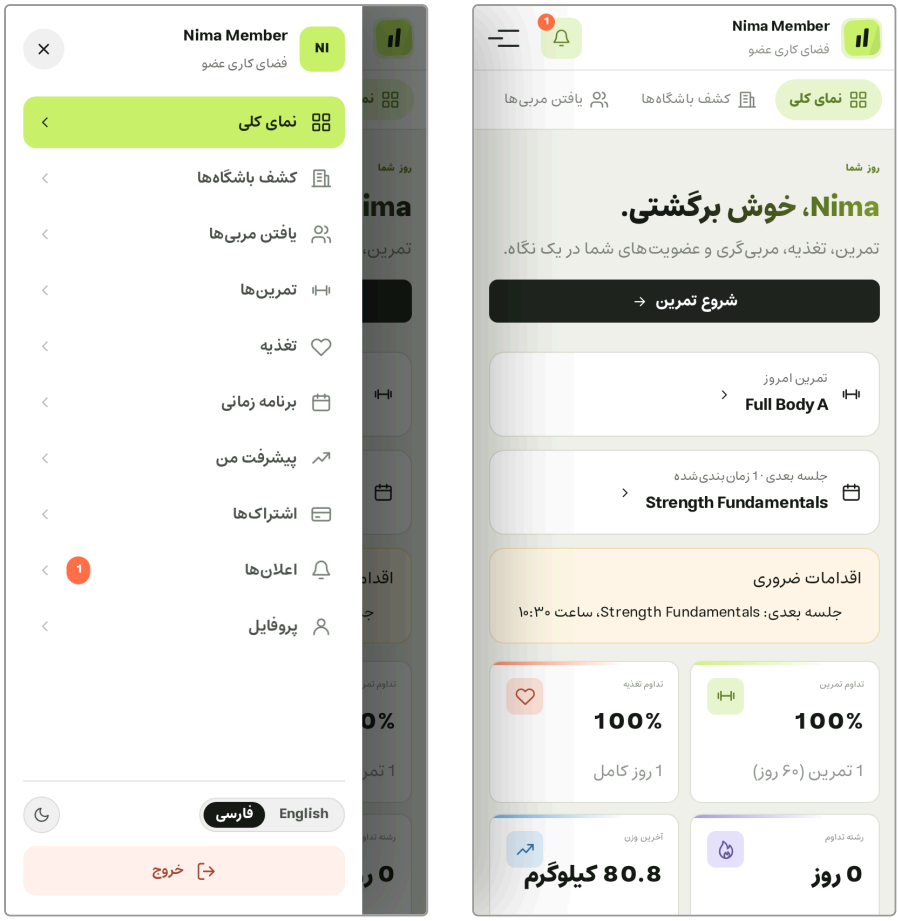
۳-۶-۳- فونت

فونت رابط کاربری فارسی، «استعداد» [۱] با مجوز آزاد OFL است که به‌صورت متغیر (وزن ۱۰۰ تا ۹۰۰) در /public/fonts میزبانی و با next/font/local بارگذاری می‌شود. خودمیزبانی به این دلیل انتخاب شد که سرویس Google Fonts در بسیاری از شبکه‌های داخل کشور در دسترس نیست و وابستگی به آن باعث بازگشت بی‌صدا به فونت پیش‌فرض سیستم می‌شد. متن انگلیسی با فونت Inter نمایش داده می‌شود.

۳-۷- طراحی واکنش‌گرا و حالت روز/شب

رابط کاربری برای عرض‌های ۳۶۰ تا ۱۲۸۰ پیکسل طراحی شده است. در عرض‌های تا ۶۴۰ پیکسل، نوار کناری پنل‌ها با یک سرصفحهٔ چسبان ۶۰ پیکسلی جایگزین می‌شود که شامل نشان برند، نام کاربر، دسترسی سریع (اعلان‌ها یا شاگردان) و دکمهٔ منوی همبرگری است؛ زیر آن یک نوار زبانه‌های افقی قابل پیمایش برای جابه‌جایی سریع قرار دارد. منوی همبرگری (components/mobile-menu.tsx) یک کشوی کناری است که از سمت شروع نوشتار باز می‌شود، با کلید Escape، لمس پس‌زمینه یا تغییر مسیر بسته می‌شود، پیمایش صفحه را قفل می‌کند و ویژگی‌های aria-expanded و role=dialog را برای فناوری‌های کمکی دارد. شکل ۳-۱۵ نمای موبایل پنل عضو را با منوی بسته و باز نشان می‌دهد.

۱ Estedad, by Amin Abedi (github.com/aminabedi68/Estedad)



شکل ۳-۱۵ نمای موبایل پنل عضو (راست) و منوی همبرگری باز (چپ)

حالت شب با کلاس theme-dark روی عنصر ریشه و متغیرهای CSS برای رنگ‌های سطح، متن و خطوط پیاده شده و انتخاب کاربر در کوکی theme ذخیره می‌شود تا در رندر سرور نیز اعمال شود و «پرش» رنگ هنگام بارگذاری رخ ندهد. شکل ۳-۱۶ داشبورد مدیر را در حالت شب نشان می‌دهد.

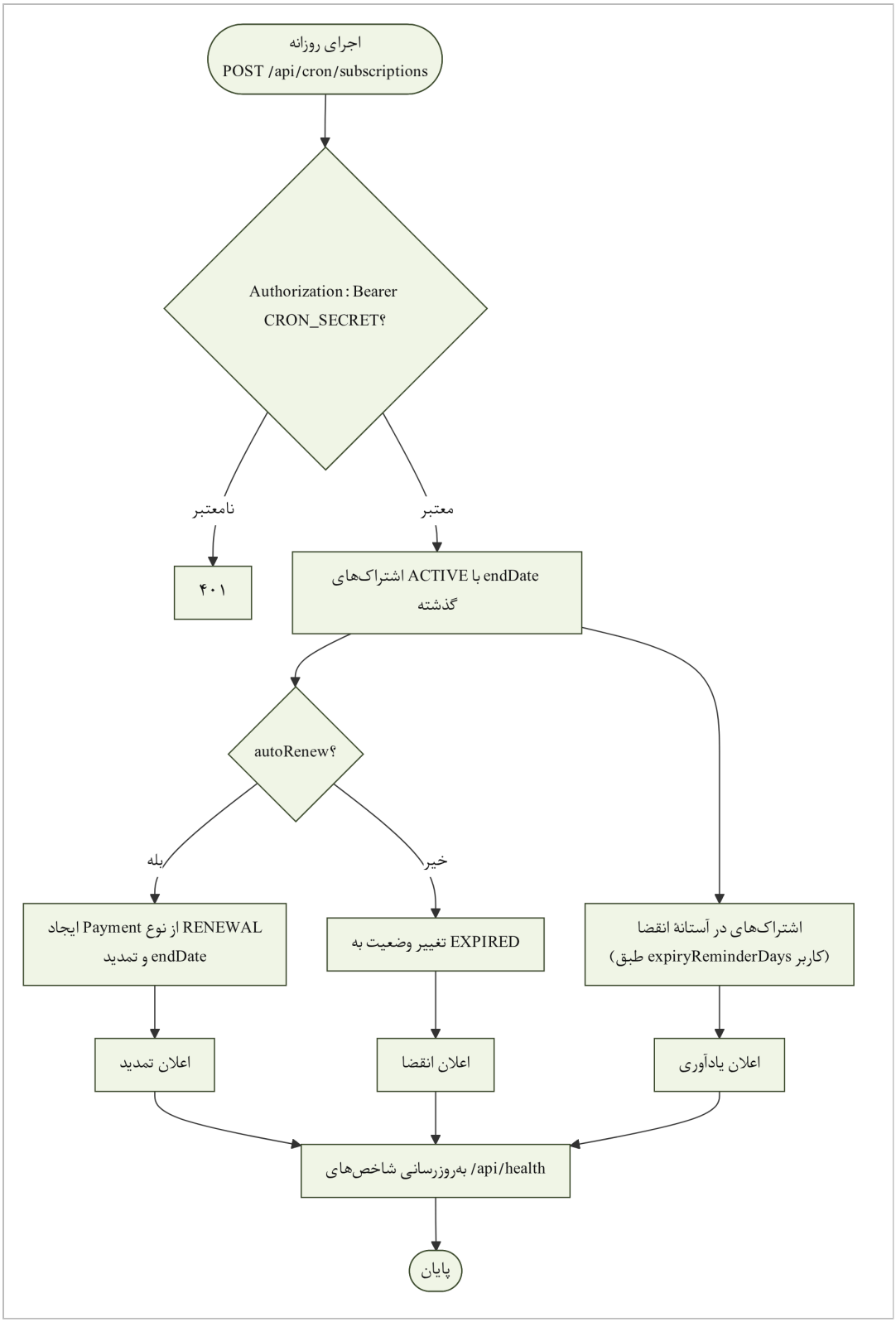
شکل ۳-۱۶ داشبورد مدیر باشگاه در حالت شب

صحت واکنش‌گرایی به‌صورت خودکار بررسی شده است: یک اسکریپت Playwright همهٔ ۳۳ صفحهٔ سامانه را در سه عرض (۳۶۰، ۷۶۸، ۱۲۸۰) و دو زبان بارگذاری و وجود سرریز افقی (scrollWidth > innerWidth) را می‌سنجد. در نسخهٔ نهایی، هیچ‌یک از ۱۹۸ ترکیب سرریز ندارد.

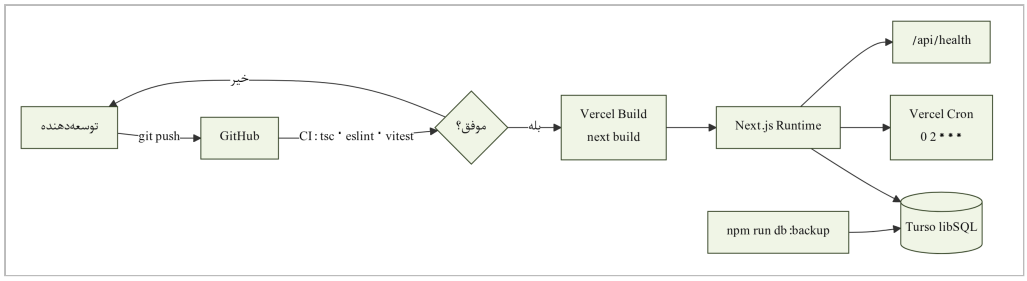
۳-۸- اتوماسیون، پایش و پشتیبان‌گیری

عملیات تکراری با یک مسیر کرون روزانه انجام می‌شود (شکل ۳-۱۷). این مسیر با سرآیند Authorization: Bearer CRON_SECRET محافظت شده و سه کار انجام می‌دهد: اشتراک‌های فعالِ منقضی را - بسته به پرچم تمدید خودکار - یا تمدید می‌کند (با ثبت پرداخت از نوع RENEWAL) یا EXPIRED می‌کند؛ و برای اشتراک‌های در آستانهٔ انقضا طبق تنظیم هر کاربر یادآوری می‌فرستد. مسیر /api/health تعداد اشتراک‌های عقب‌افتاده (که پس از هر اجرا باید صفر شود) و پنج خطای اخیر سرور را گزارش می‌کند تا با ابزارهای پایش بیرونی قابل هشداردهی باشد.

اسکریپت npm run db:backup از فایل پایگاه داده (یا در حالت میزبانی‌شده، از محتوای جدول‌ها به‌صورت JSON) و پوشهٔ آپلودها نسخهٔ پشتیبان می‌گیرد، یک مانیفست با اندازه و چکیدهٔ هر فایل می‌نویسد و در پایان یک «آزمون بازیابی خشک» انجام می‌دهد تا از خوانا بودن نسخه اطمینان حاصل شود. شکل ۳-۱۸ چرخهٔ استقرار را نشان می‌دهد.



شکل ۳-۱۷ فلوچارت زمان‌بند روزانهٔ اشتراک‌ها



شکل ۳-۱۸ چرخهٔ یکپارچه‌سازی و استقرار پیوسته

۳-۹- آزمون و تضمین کیفیت

سه سطح بررسی خودکار در پروژه وجود دارد که پیش از هر ادغام کد اجرا می‌شوند: بررسی نوع با tsc --noEmit در حالت سخت‌گیرانه، تحلیل ایستا با ESLint (شامل قواعد React Compiler برای جلوگیری از رندرهای آبشاری) و ۷۶ آزمون واحد Vitest در ۱۸ فایل. جدول ۳-۴ پوشش آزمون‌ها را به تفکیک حوزه نشان می‌دهد. زمان اجرای کل مجموعه کمتر از ۶ ثانیه است که اجرای مکرر آن را در حین توسعه عملی می‌کند.

جدول ۳-۴ پوشش آزمون‌های واحد به تفکیک حوزه

فایل آزمون	حوزه	نمونهٔ موارد آزمون
auth.test.ts, auth-validation	احراز هویت	هش و بررسی گذرواژه، امضا و اعتبار JWT، قواعد ایمیل/گذرواژه
account-tokens.test.ts	بازیابی حساب	تولید، هش و انقضای توکن یک‌بارمصرف
admin/trainer/member-validation	اعتبارسنجی ورودی	طرح اشتراک، پروفایل مربی، اندازه‌های بدنی، مقادیر مرزی
plan-validation.test.ts	برنامه‌ها	ساختار روز/حرکت، ست و تکرار، کالری
scheduling.test.ts	زمان‌بندی	تداخل بازه‌ها، قرار گرفتن در بازهٔ در دسترس
finance-report.test.ts	مالی	جمع به تفکیک ارز، فقط پرداخت موفق، کسر بازپرداخت، درآمد ماهانه
streak.test.ts, engagement.test.ts	درگیری کاربر	استریک با منطقهٔ زمانی تهران، شاگردان کم‌فعال، گفت‌وگوهای بی‌پاسخ

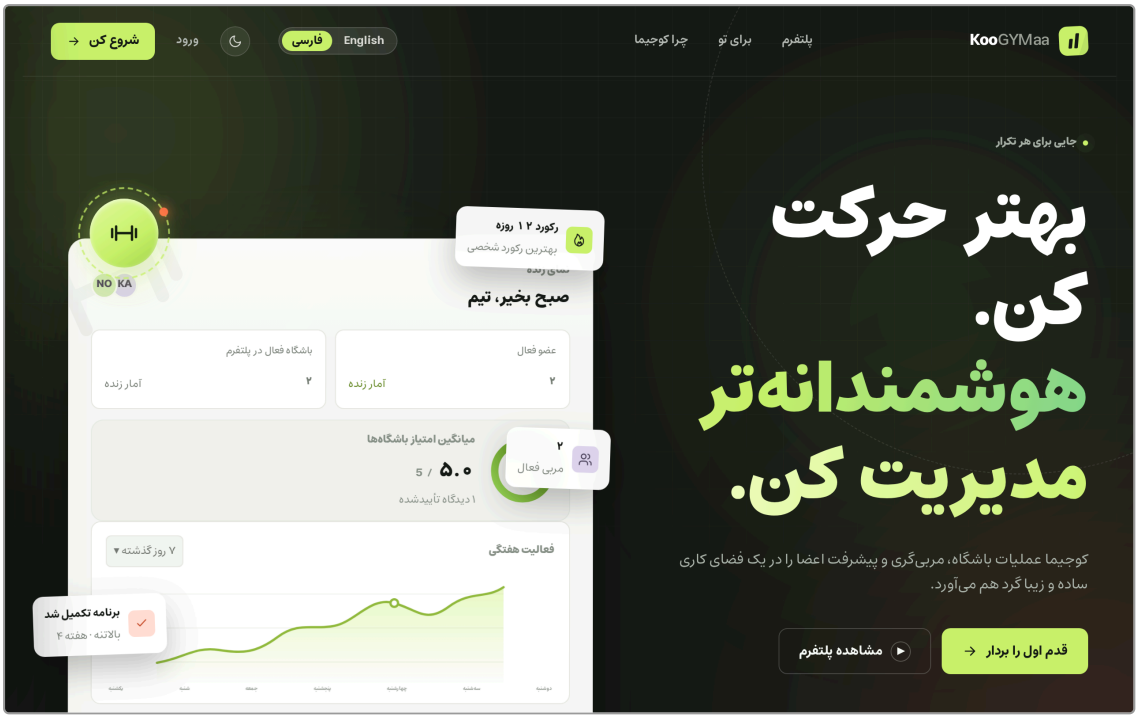
fa-utils.test.ts	بومی‌سازی	تبدیل جلالی، ارقام فارسی، قالب پول
payment-provider.test.ts	پرداخت	توکن نمایشی، امضای HMAC، رد امضای نامعتبر
csv-export.test.ts, schema.test.ts, hosted-sql	زیرساخت	گریز کاراکترهای CSV، سازگاری طرحواره، اتصال libSQL

۱۰-۳- راهنمای استفاده: ورود به سامانه

بخش دوم این فصل، راهنمای گام‌به‌گام کار با سامانه است. تصاویر از نسخهٔ فارسی سامانه با داده‌های نمونه گرفته شده‌اند. ابتدا مراحل مشترک (ثبت‌نام و ورود) و سپس راهنمای اختصاصی هر نقش ارائه می‌شود.

۱-۱۰-۳- صفحهٔ اصلی

با ورود به نشانی سامانه، صفحهٔ اصلی (شکل ۳-۱۹) نمایش داده می‌شود. نوار بالای صفحه شامل پیوندهای بخش‌ها، سوییچ زبان (فارسی/English)، دکمهٔ روز/شب، «ورود» و «شروع کن» است. در ادامهٔ صفحه، معرفی قابلیت‌ها برای هر سه نقش، شمارنده‌های زنده و بخش پرسش‌های متداول قرار دارد.



شکل ۳-۱۹ صفحهٔ اصلی سامانه

۲-۱۰-۳- ثبت‌نام

برای ساخت حساب، روی «شروع کن» یا «حساب کاربری بساز» کلیک کنید (شکل ۳-۲۰) و مراحل زیر را دنبال کنید:

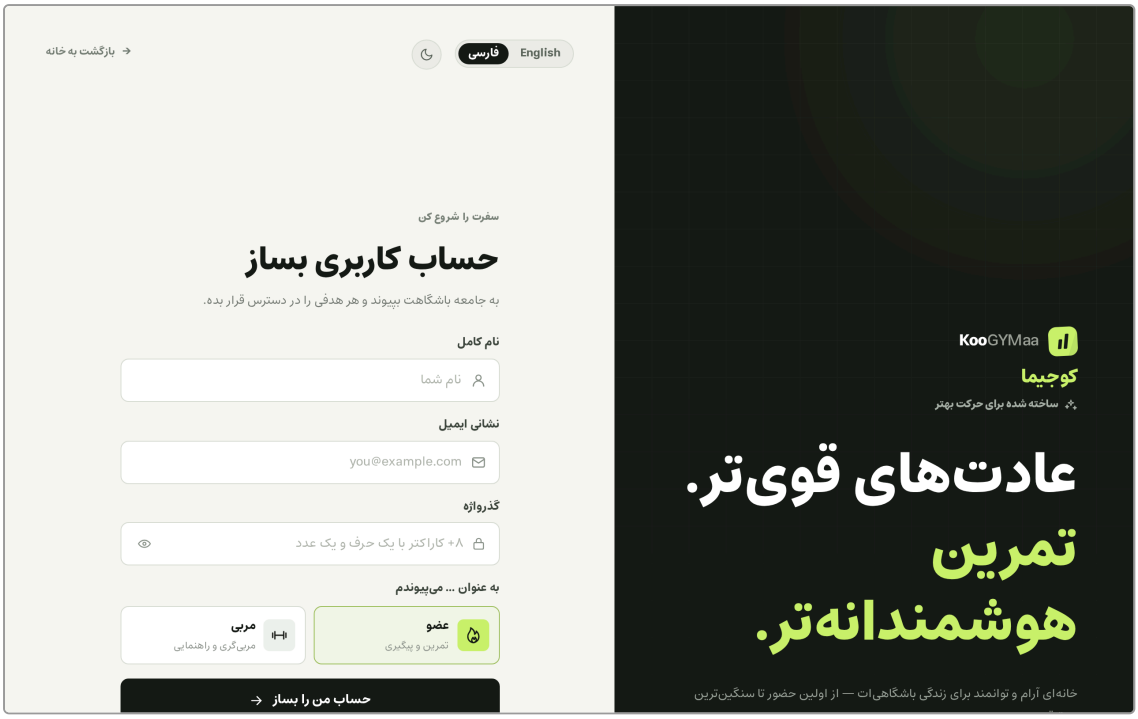
۱. نام کامل خود را وارد کنید.

۲. نشانی ایمیل معتبر وارد کنید؛ این نشانی برای ورود و بازیابی گذرواژه استفاده می‌شود.

۳. گذرواژه‌ای با حداقل ۸ کاراکتر شامل حرف و عدد انتخاب کنید.

۴. نقش خود را انتخاب کنید: «عضو» برای ورزشکاران و «مربی» برای مربیان. حساب مدیر باشگاه با انتخاب «مدیر» ساخته می‌شود و پس از ورود می‌توانید باشگاه خود را بسازید.

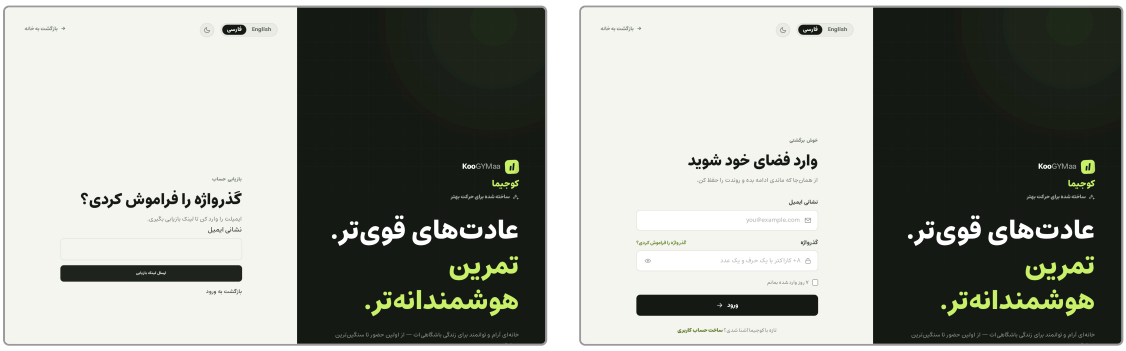
۵. روی «حساب من را بساز» کلیک کنید. پس از ثبت موفق، به‌طور خودکار وارد پنل نقش خود می‌شوید.



شکل ۳-۲۰ فرم ثبت‌نام با انتخاب نقش

۳-۱۰-۳- ورود و بازیابی گذرواژه

در صفحهٔ ورود (شکل ۳-۲۱) ایمیل و گذرواژه را وارد کنید. با فعال کردن «مرا به خاطر بسپار» نشست شما ۷ روز (به‌جای ۲۴ ساعت) معتبر می‌ماند. اگر گذرواژه را فراموش کرده‌اید، پیوند «فراموشی گذرواژه» شما را به فرمی می‌برد که با وارد کردن ایمیل، پیوند بازنشانی ۳۰ دقیقه‌ای دریافت می‌کنید. دکمه‌های زبان و روز/شب در گوشهٔ بالای این صفحات نیز در دسترس‌اند.

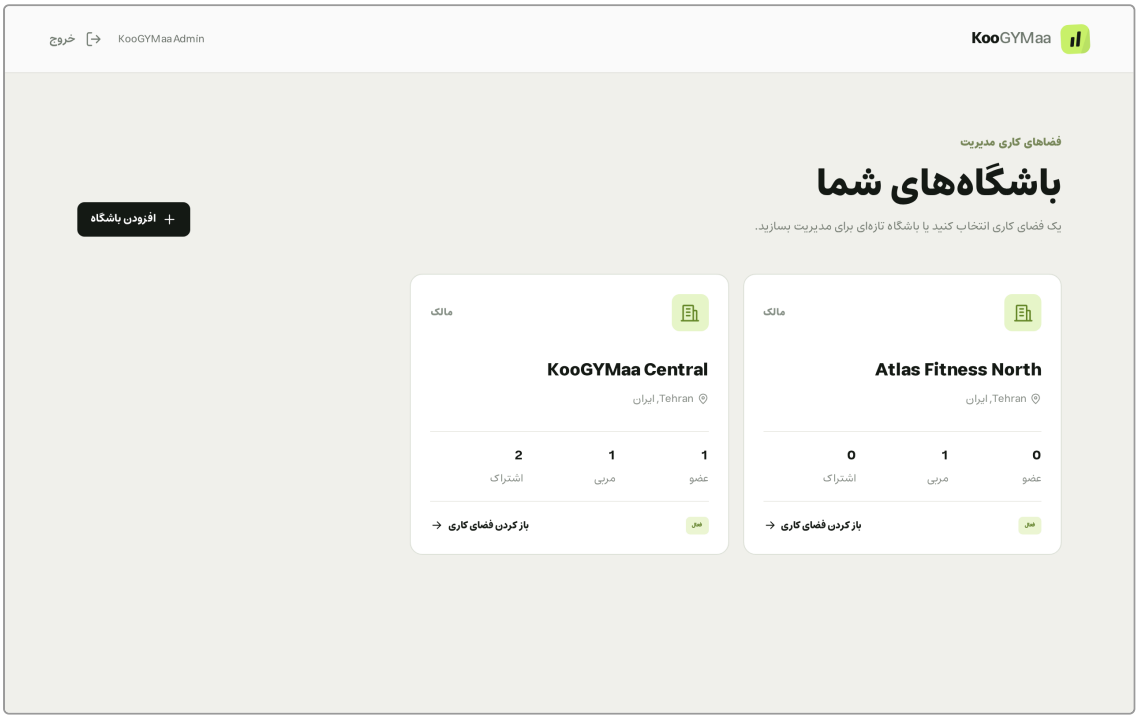


شکل ۳-۲۱ صفحهٔ ورود (راست) و فراموشی گذرواژه (چپ)

۳-۱۱- راهنمای مدیر باشگاه

۳-۱۱-۱- فضاهای کاری و ساخت باشگاه

پس از ورود، مدیر فهرست باشگاه‌های خود را می‌بیند (شکل ۳-۲۲). برای ساخت باشگاه جدید، در فرم سمت راست نام، شهر و توضیح کوتاه را وارد و «ساخت باشگاه» را بزنید. هر باشگاه یک «فضای کاری» مستقل با داشبورد، اعضا، مربیان و مالی جداگانه است و مدیر می‌تواند چند باشگاه را اداره کند. با کلیک روی کارت هر باشگاه وارد فضای کاری آن می‌شوید.

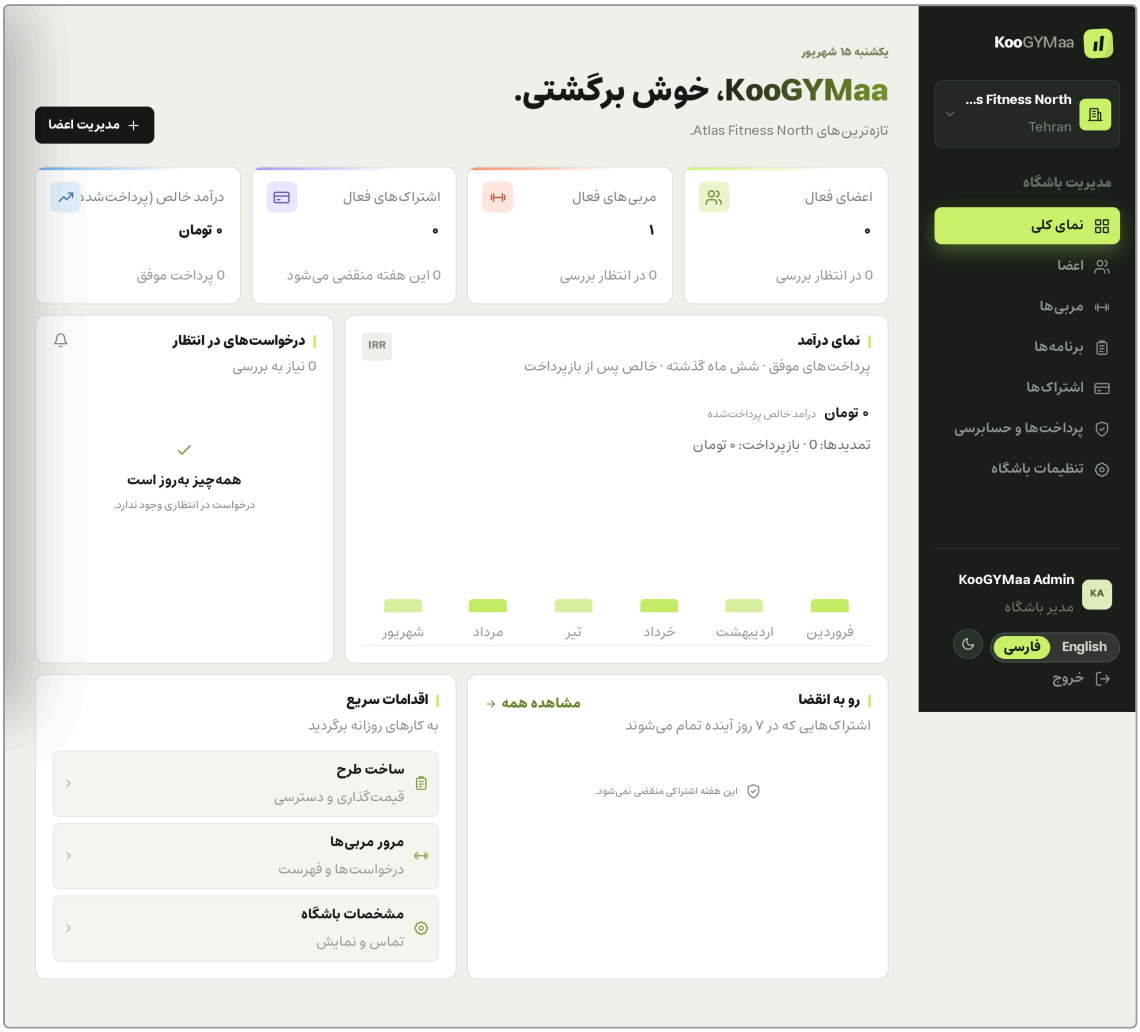


شکل ۳-۲۲ فهرست فضاهای کاری مدیر و فرم ساخت باشگاه

۳-۱۱-۲- داشبورد

داشبورد (شکل ۳-۲۳) در یک نگاه وضعیت باشگاه را نشان می‌دهد: تعداد اعضا و مربیان فعال، اشتراک‌های فعال و در آستانهٔ انقضای این هفته، درآمد خالص، نمودار درآمد شش ماه گذشته، درخواست‌های در انتظار بررسی و «اقدامات سریع». نوار کناری سمت راست دسترسی به

بخش‌های اعضا، مربی‌ها، برنامه‌ها، اشتراک‌ها، پرداخت‌ها و تنظیمات را فراهم می‌کند و در پایین آن سوییچ زبان، دکمهٔ روز/شب و خروج قرار دارد.

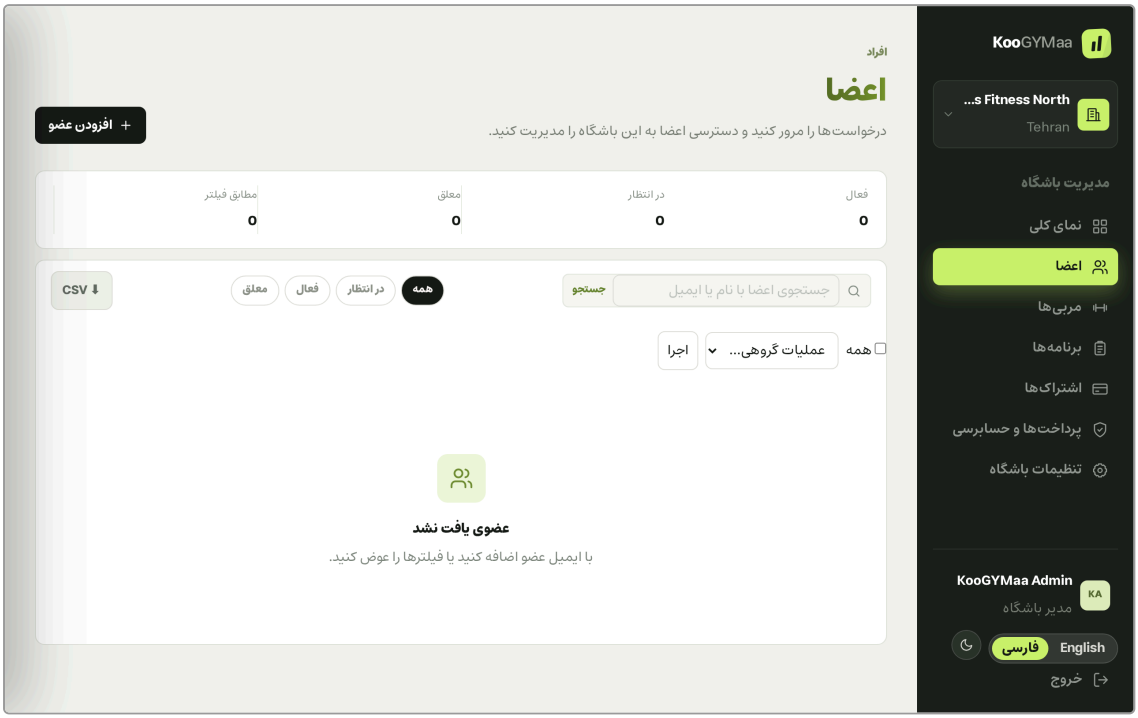


شکل ۳-۲۳ داشبورد مدیر باشگاه

۳-۱۱-۳- مدیریت اعضا

در بخش «اعضا» (شکل ۳-۲۴) فهرست همهٔ درخواست‌ها و عضویت‌ها با زبانه‌های وضعیت (همه، در انتظار، تأییدشده، تعلیق) و جست‌وجو نمایش داده می‌شود. برای هر درخواست می‌توانید «تأیید» یا «رد» را بزنید، یا با انتخاب چند ردیف، عملیات گروهی انجام دهید. دکمهٔ «افزودن عضو» برای ثبت‌نام حضوری است: ایمیل عضو را وارد کنید تا اگر حساب داشته باشد مستقیماً عضو شود و

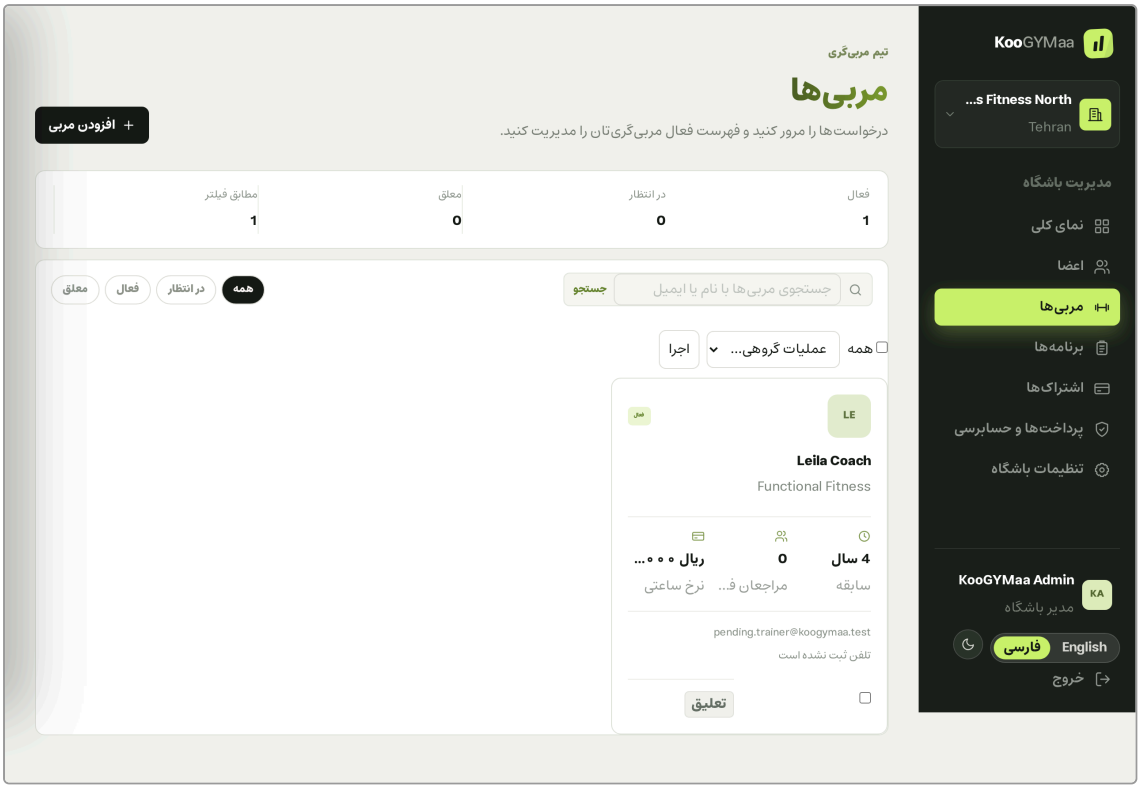
در غیر این صورت دعوت‌نامه دریافت کند. ستون «اشتراک» وضعیت فعلی اشتراک هر عضو را نشان می‌دهد.



شکل ۳-۲۴ مدیریت اعضا و درخواست‌های عضویت

۴-۱۱-۳- مدیریت مربیان

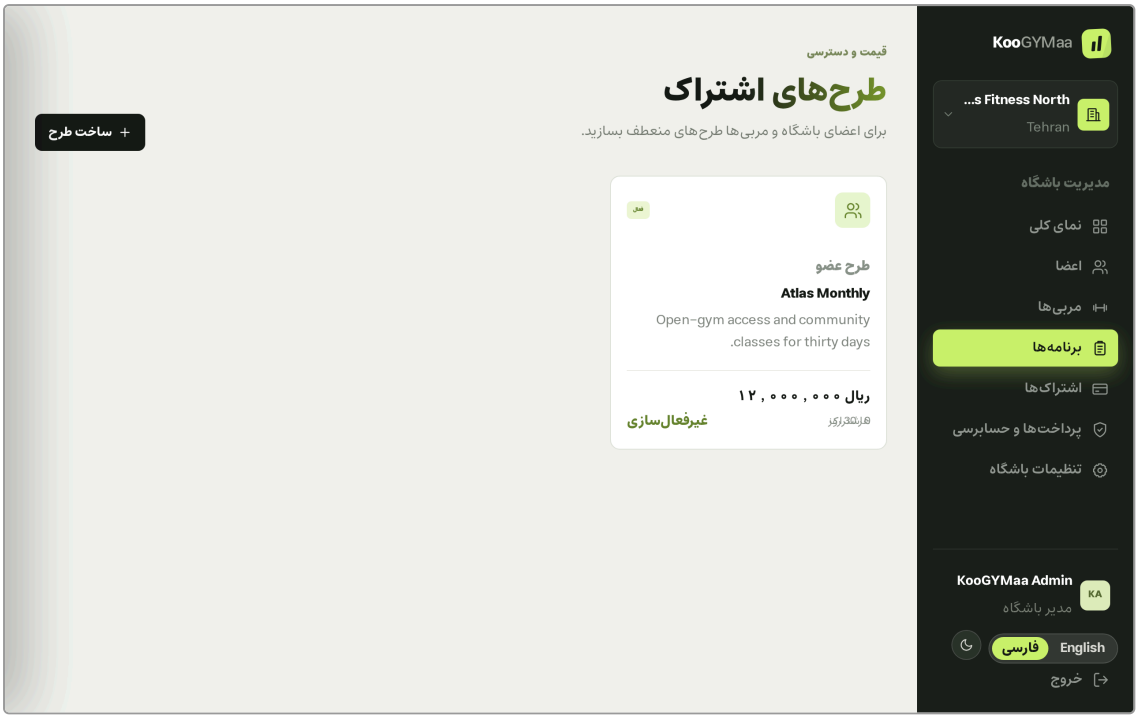
بخش «مربی‌ها» (شکل ۳-۲۵) مشابه اعضا عمل می‌کند: درخواست‌های همکاری مربیان را ببینید، پروفایل و تخصص‌هایشان را بررسی و تأیید یا رد کنید. برای هر مربی تعداد شاگردان فعال در این باشگاه نمایش داده می‌شود. مربی تأییدشده در صفحهٔ عمومی باشگاه به اعضا معرفی می‌شود.



شکل ۳-۲۵ مدیریت مربیان باشگاه

۵-۱۱-۳- طرح‌های اشتراک

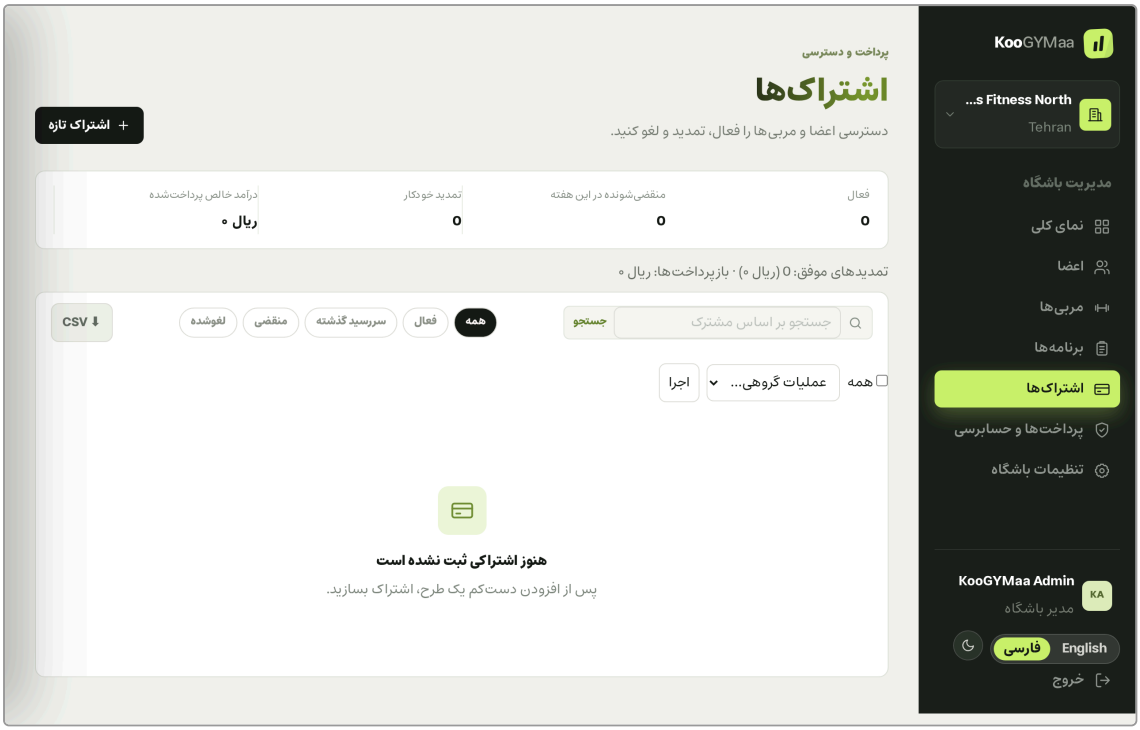
در بخش «برنامه‌ها» (شکل ۳-۲۶) طرح‌های اشتراک باشگاه را تعریف کنید. برای هر طرح نام، مدت (روز)، قیمت، ارز و مخاطب (عضو یا مربی) را مشخص کنید. طرح‌ها را می‌توان غیرفعال کرد بدون آنکه اشتراک‌های خریداری‌شدهٔ قبلی آسیب ببینند، زیرا قیمت پرداخت‌شده در خود اشتراک ذخیره می‌شود.



شکل ۳-۲۶ تعریف و مدیریت طرح‌های اشتراک

۶-۱۱-۳- اشتراک‌ها

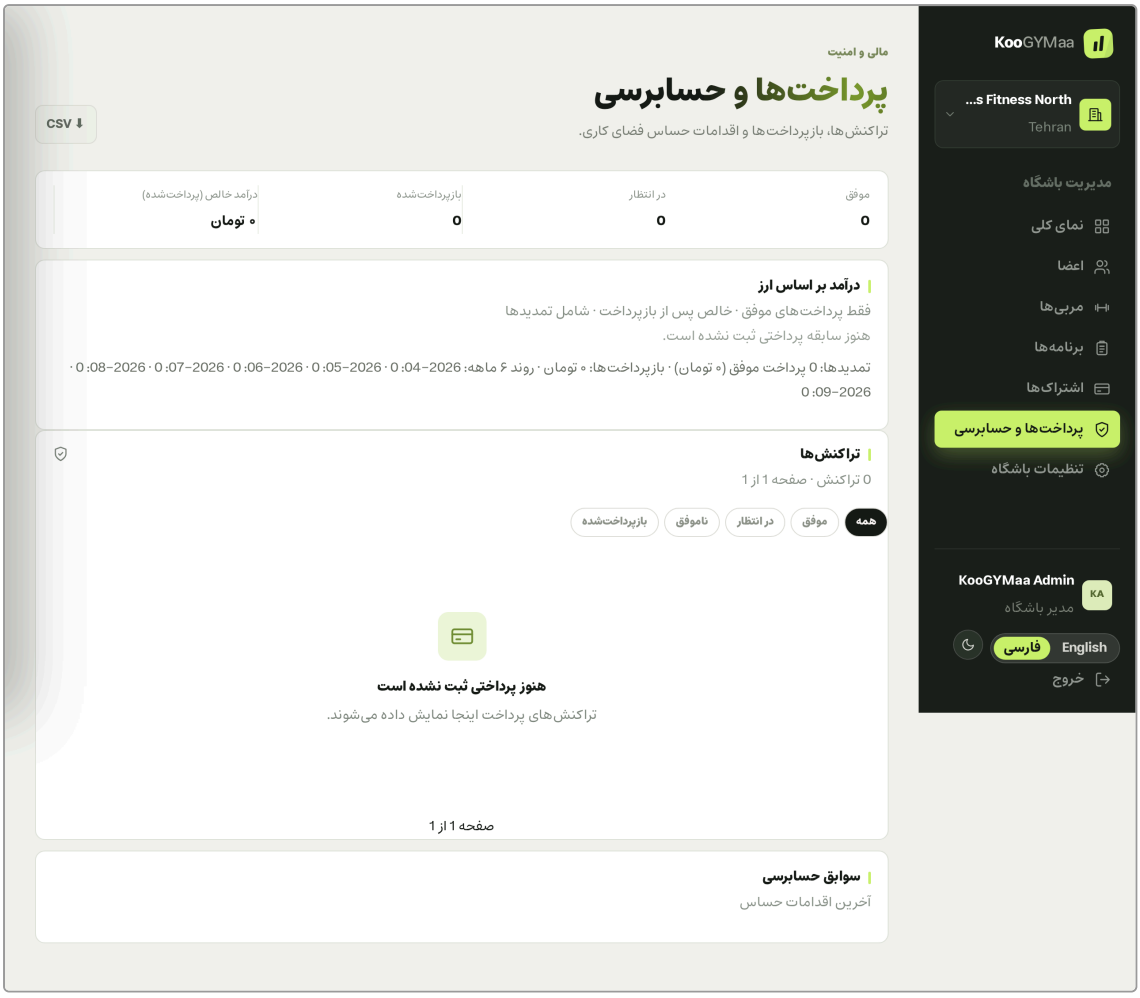
بخش «اشتراک‌ها» (شکل ۳-۲۷) همهٔ اشتراک‌های اعضا را با وضعیت، تاریخ شروع و پایان و روزهای باقی‌مانده فهرست می‌کند. می‌توانید اشتراکی را دستی فعال، تمدید یا لغو کنید (مثلاً برای پرداخت نقدی) و با عملیات گروهی چند اشتراک را هم‌زمان تغییر دهید. اشتراک‌های در آستانهٔ انقضا با رنگ هشدار مشخص شده‌اند.



شکل ۳-۲۷ فهرست و مدیریت اشتراک‌ها

۷-۱۱-۳- پرداخت‌ها و گزارش مالی

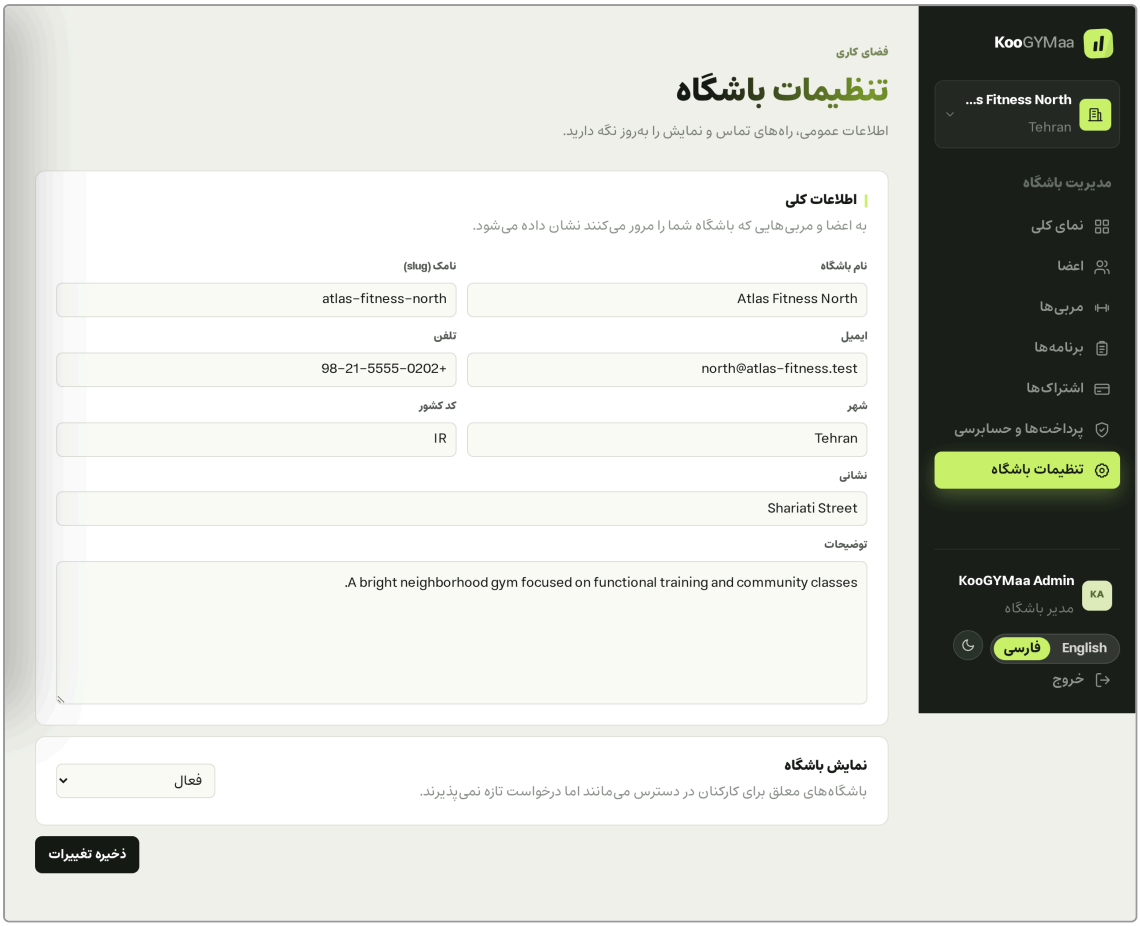
صفحهٔ «پرداخت‌ها و حسابرسی» (شکل ۳-۲۸) سه بخش دارد: خلاصهٔ مالی به تفکیک ارز (درآمد ناخالص، بازپرداخت، خالص، تعداد تمدیدها)، فهرست پرداخت‌ها با امکان بازپرداخت پرداخت‌های موفق، و خط زمانی رخدادهای اداری (چه کسی، چه زمانی، چه کاری کرد). دکمهٔ «خروجی CSV» فهرست اعضا، مربیان، اشتراک‌ها یا پرداخت‌ها را برای استفاده در اکسل یا نرم‌افزار حسابداری دانلود می‌کند. توجه کنید که فقط پرداخت‌های موفق در درآمد شمرده می‌شوند و پرداخت‌های ناموفق یا در انتظار هرگز جمع را افزایش نمی‌دهند.



شکل ۳-۲۸ پرداخت‌ها، گزارش مالی و خط زمانی حسابرسی

۸-۱۱-۳- تنظیمات باشگاه

در «تنظیمات باشگاه» (شکل ۳-۲۹) می‌توانید نام، شهر، نشانی، توضیحات، ساعت کاری و وضعیت باشگاه (فعال/غیرفعال) را ویرایش کنید. غیرفعال کردن باشگاه، آن را از صفحهٔ کشف باشگاه‌ها پنهان می‌کند اما داده‌ها حفظ می‌شود.

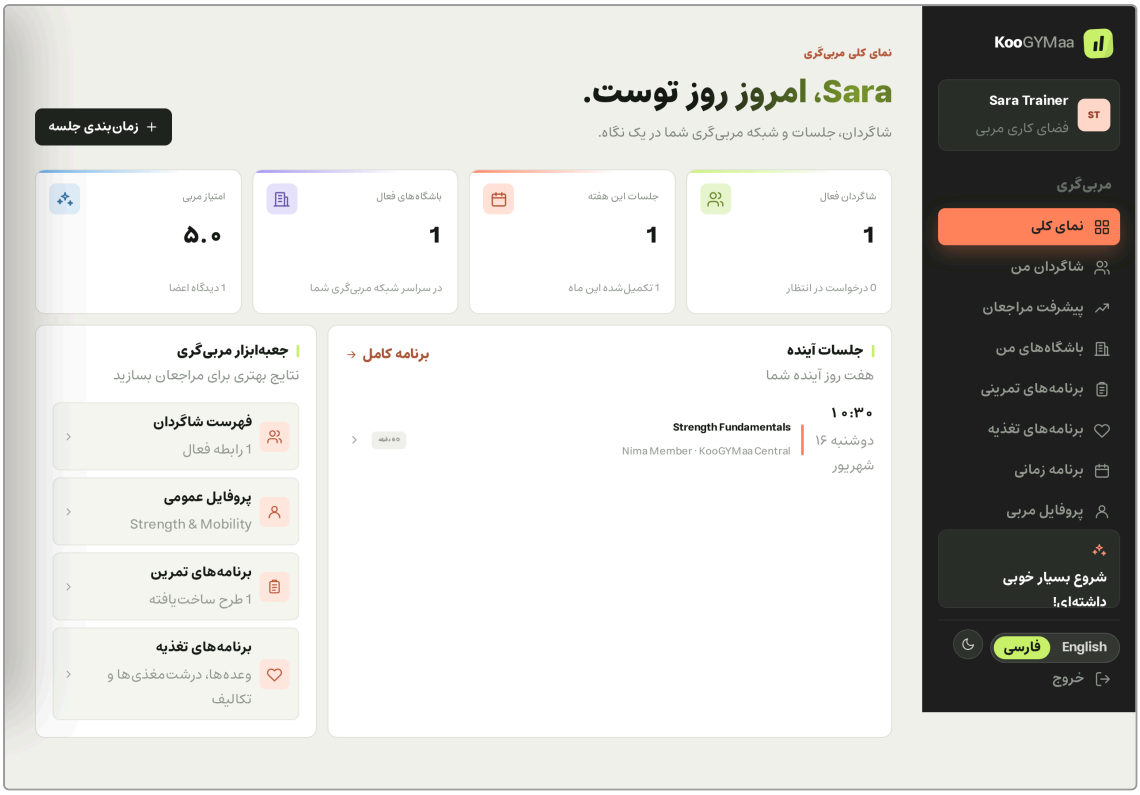


شکل ۳-۲۹ تنظیمات باشگاه

۳-۱۲- راهنمای مربی

۳-۱۲-۱- داشبورد مربی

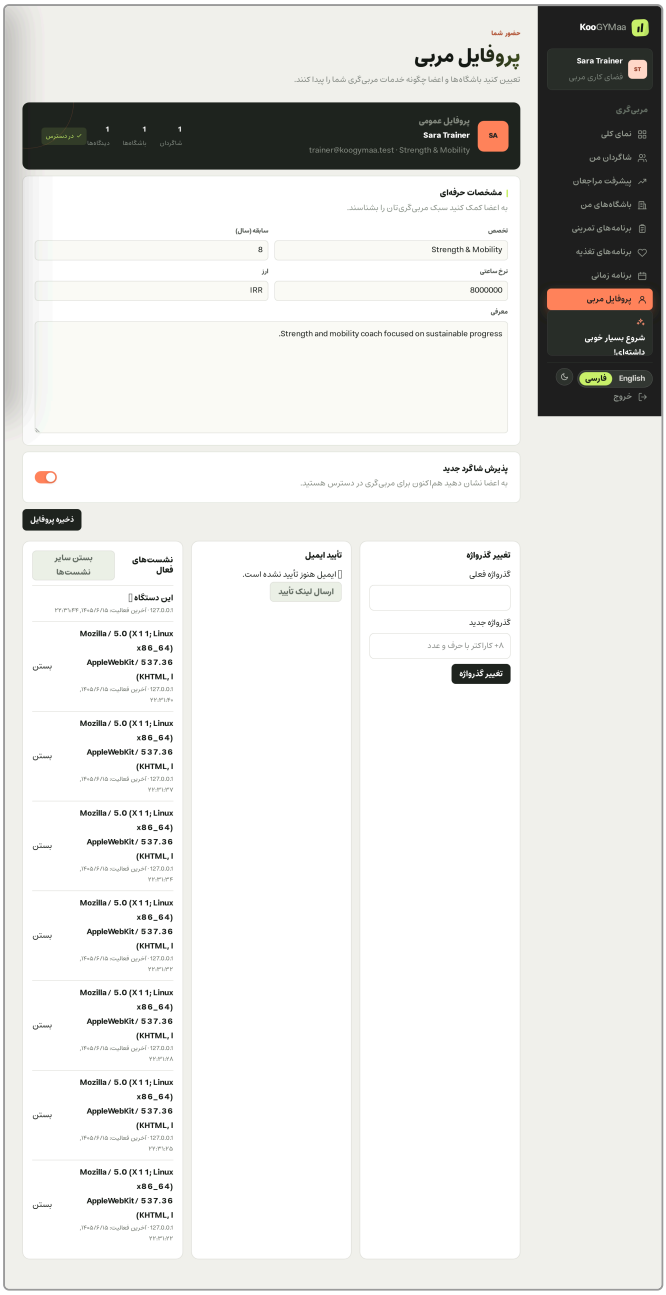
داشبورد مربی (شکل ۳-۳۰) تعداد شاگردان فعال، درخواست‌های در انتظار، جلسات امروز و برنامه‌های منتشرشده را نشان می‌دهد و شاگردانی را که بیش از یک هفته فعالیت نداشته‌اند برجسته می‌کند تا با یک کلیک برایشان یادآور بفرستید. نوار کناری (نارنجی، برای تمایز از پنل مدیر) شامل شاگردان، پیشرفت مراجعان، باشگاه‌ها، برنامه‌های تمرینی و غذایی، برنامهٔ زمانی و پروفایل است.



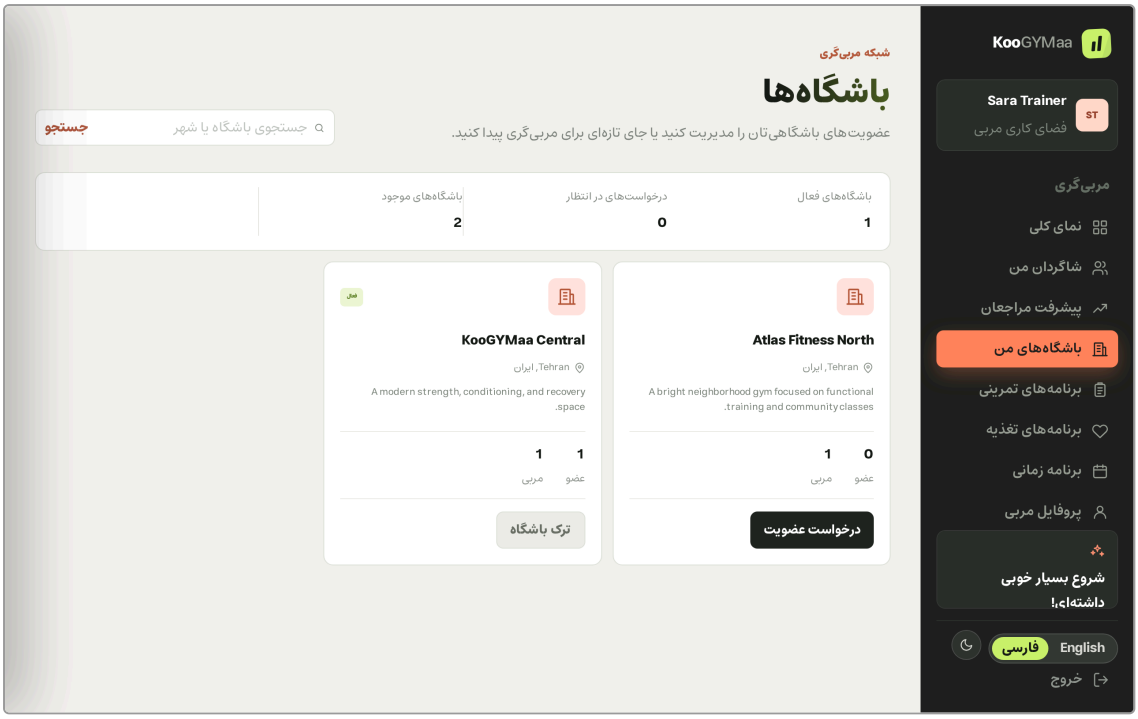
شکل ۳-۳۰ داشبورد مربی

۲-۱۲-۳- تکمیل پروفایل و پیوستن به باشگاه

پیش از هر چیز، پروفایل حرفه‌ای خود را در «پروفایل مربی» (شکل ۳-۳۱) تکمیل کنید: بیوگرافی، تخصص‌ها، سال‌های تجربه و مدارک. این اطلاعات در صفحهٔ عمومی مربیان به اعضا نمایش داده می‌شود. سپس در «باشگاه‌های من» (شکل ۳-۳۲) باشگاه‌های فعال را ببینید و برای همکاری «درخواست» بدهید؛ پس از تأیید مدیر، شاگردان آن باشگاه می‌توانند شما را انتخاب کنند. می‌توانید هم‌زمان با چند باشگاه همکاری کنید.



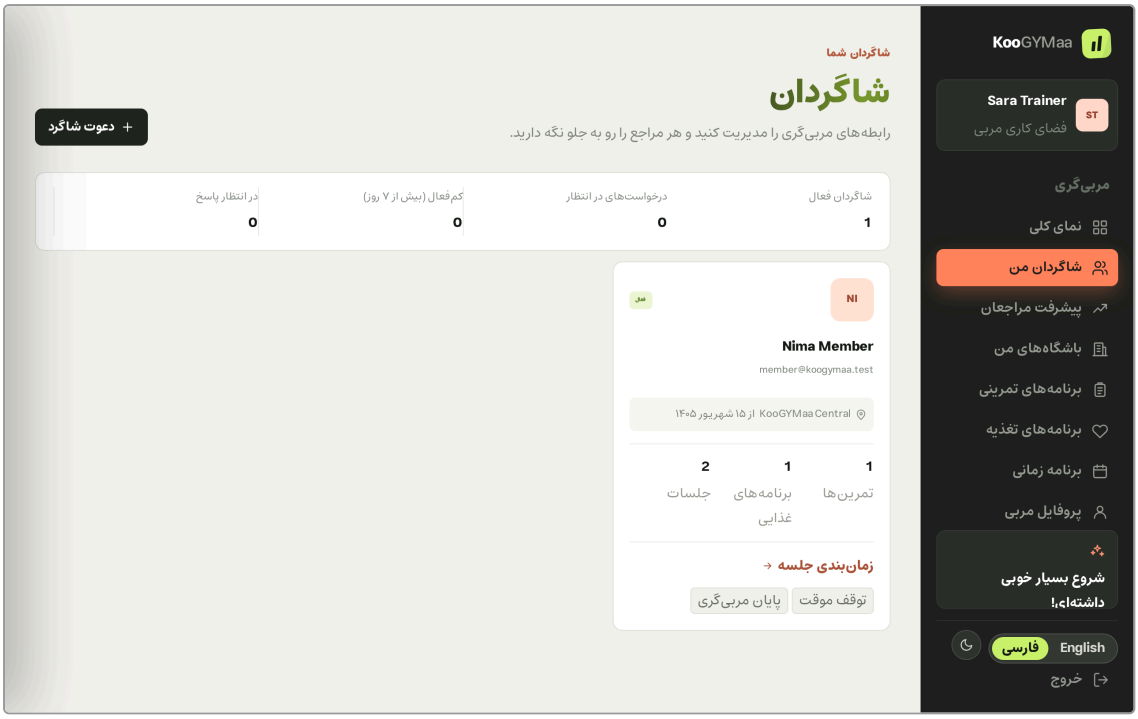
شکل ۳-۳۱ پروفایل حرفه‌ای مربی



شکل ۳-۳۲ باشگاه‌های مربی و درخواست همکاری

۳-۱۲-۳- مدیریت شاگردان

در «شاگردان» (شکل ۳-۳۳) درخواست‌های مربی‌گری اعضا را بپذیرید یا رد کنید و فهرست شاگردان فعال را ببینید. برای هر شاگرد، تعداد تمرین‌ها، برنامه‌های غذایی و جلسات و آخرین فعالیت نمایش داده می‌شود. از همین‌جا می‌توانید جلسه زمان‌بندی کنید، رابطه را موقتاً متوقف یا پایان دهید. دکمهٔ «دعوت شاگرد» به شما امکان می‌دهد با ایمیل، عضوی را مستقیماً دعوت کنید.



شکل ۳-۳۳ مدیریت شاگردان

۴-۱۲-۳- ساخت برنامهٔ تمرینی

در «برنامه‌های تمرینی» (شکل ۳-۳۴) روی «برنامهٔ جدید» کلیک کنید و مراحل زیر را دنبال کنید:

۱. عنوان و توضیح برنامه و هدف آن (قدرت، کاهش وزن، آمادگی عمومی و...) را وارد کنید.

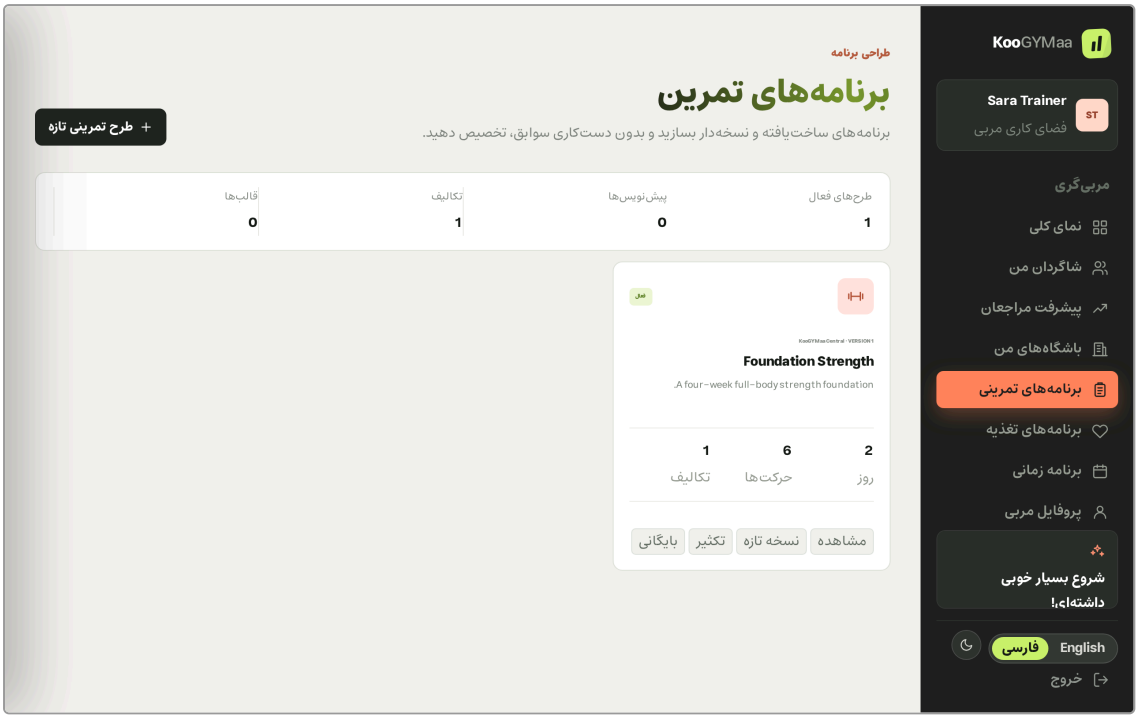
۲. روزهای تمرین را اضافه کنید (مثلاً «روز اول - بالاتنه»).

۳. برای هر روز، حرکات را از کتابخانه انتخاب یا دستی وارد کنید و ست، تکرار، وزنهٔ پیشنهادی و زمان استراحت را مشخص کنید.

۴. برنامه را ذخیره کنید؛ در حالت «پیش‌نویس» فقط شما آن را می‌بینید.

۵. با «انتشار»، برنامه آمادهٔ تخصیص می‌شود. برای شخصی‌سازی برای هر شاگرد، از «کپی» استفاده کنید تا الگو دست‌نخورده بماند.

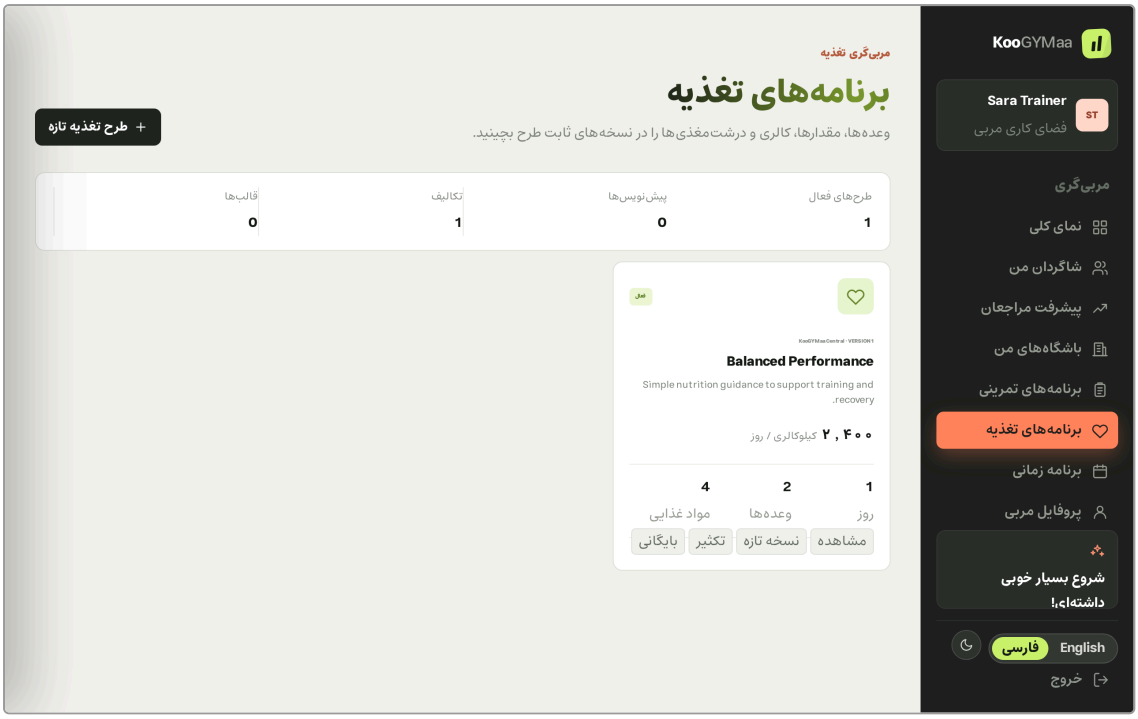
۶. در «تخصیص»، شاگرد و تاریخ شروع را انتخاب کنید. شاگرد اعلان دریافت می‌کند و برنامه در پنل او ظاهر می‌شود.



شکل ۳-۳۴ فهرست برنامه‌های تمرینی مربی

۵-۱۲-۳- ساخت برنامهٔ غذایی

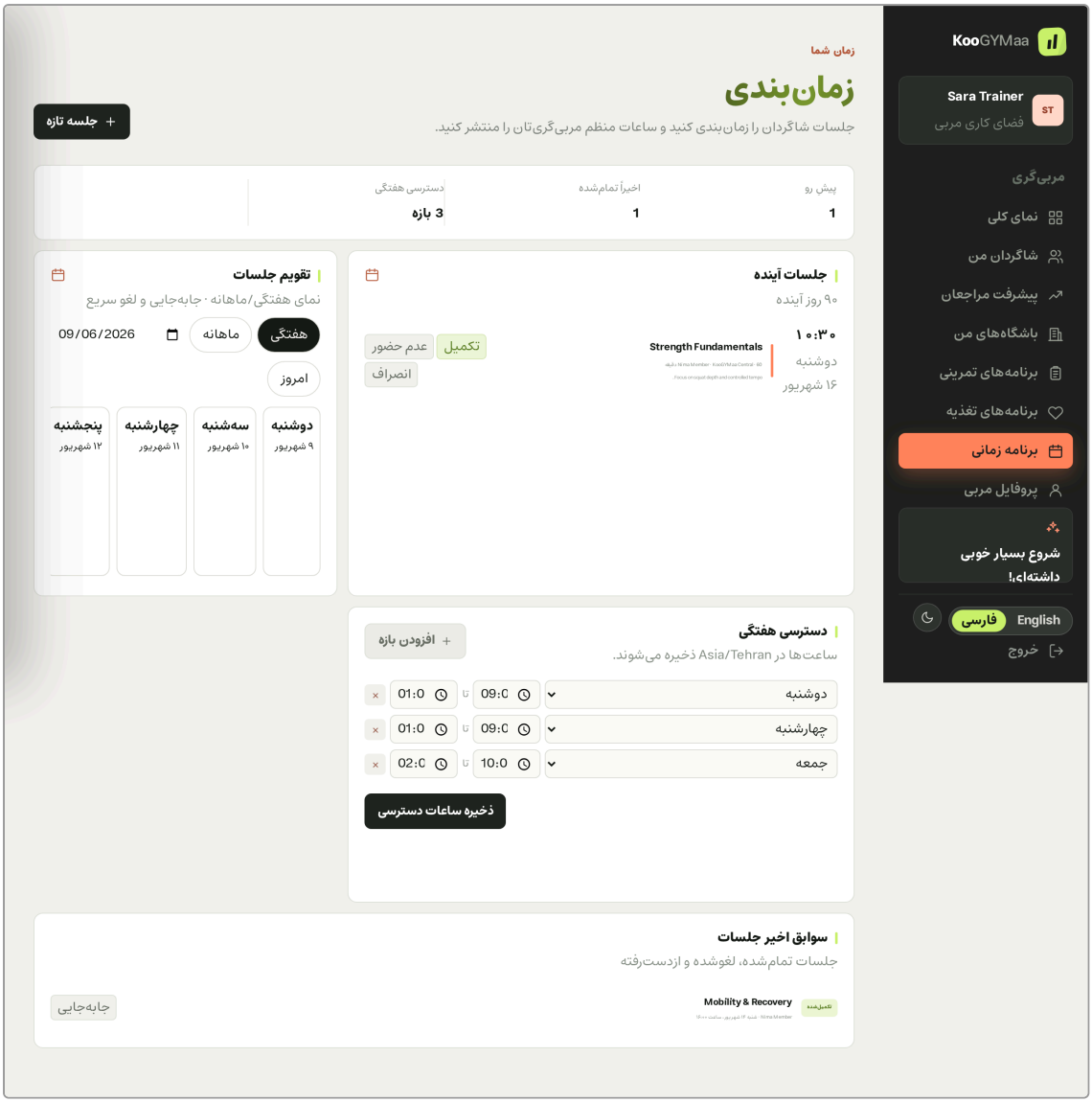
«برنامه‌های غذایی» (شکل ۳-۳۵) ساختار مشابهی دارد: روزها، وعده‌ها (صبحانه، ناهار، میان‌وعده و...) و مواد غذایی هر وعده با مقدار و کالری. هدف کالری روزانه در بالای برنامه تنظیم می‌شود و جمع کالری هر روز به‌طور خودکار محاسبه می‌گردد. تخصیص و کپی مانند برنامهٔ تمرینی است.



شکل ۳-۳۵ برنامه‌های غذایی مربی

۶-۱۲-۳- برنامهٔ زمانی و جلسات

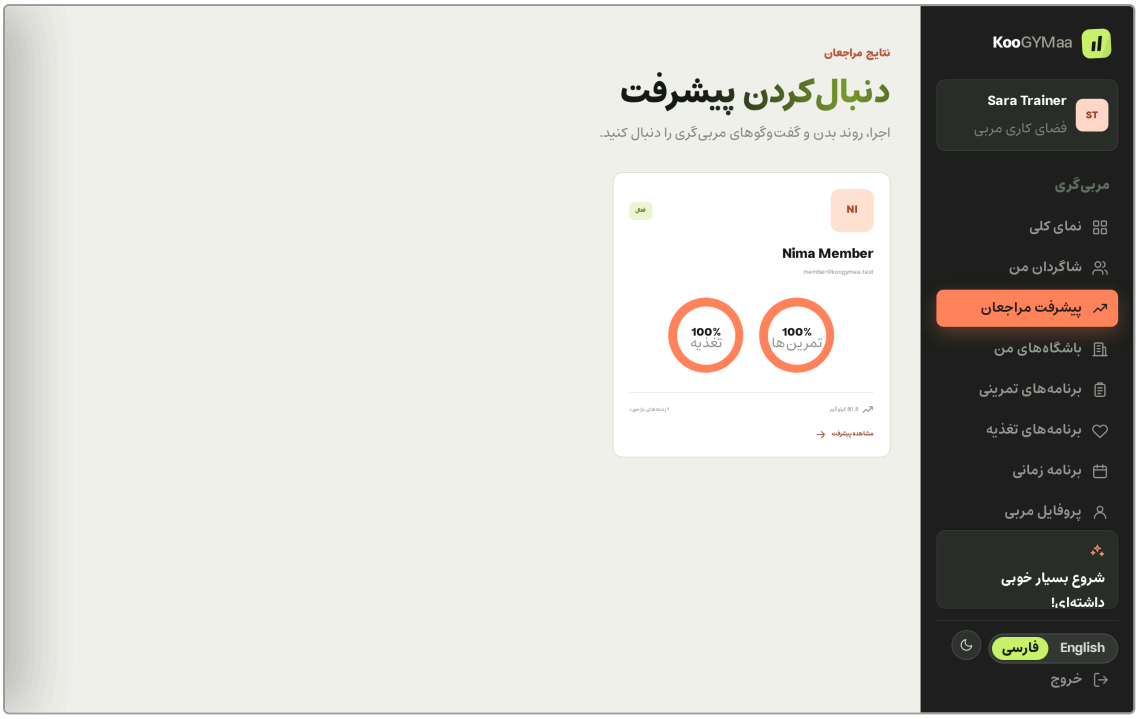
در «برنامهٔ زمانی» (شکل ۳-۳۶) ابتدا بازه‌های در دسترس هفتگی خود را تعریف کنید (روز هفته، ساعت شروع و پایان). سپس جلسات رزروشده را در تقویم ببینید، جلسهٔ جدید برای شاگرد ثبت کنید (سامانه تداخل را بررسی می‌کند) و وضعیت جلسات گذشته را به «انجام‌شده» یا «غیبت» تغییر دهید. یادآوری جلسه به‌طور خودکار برای شاگرد ارسال می‌شود.



شکل ۳-۳۶ برنامهٔ زمانی مربی: بازه‌های در دسترس و تقویم جلسات

۷-۱۲-۳- پیگیری پیشرفت و بازخورد

«پیشرفت مراجعان» (شکل ۳-۳۷) برای هر شاگرد خلاصهٔ تمرین‌های ثبت‌شده، شدت درک‌شده، اندازه‌های بدنی و عکس‌های پیشرفت را نشان می‌دهد. با کلیک روی هر شاگرد، جزئیات هر جلسهٔ تمرین (ست‌ها، تکرارها و وزنه‌های واقعی) را می‌بینید و می‌توانید زیر آن بازخورد بنویسید؛ شاگرد اعلان می‌گیرد و می‌تواند پاسخ دهد. گفت‌وگوهایی که بیش از ۲۴ ساعت بی‌پاسخ مانده‌اند در داشبورد یادآوری می‌شوند.

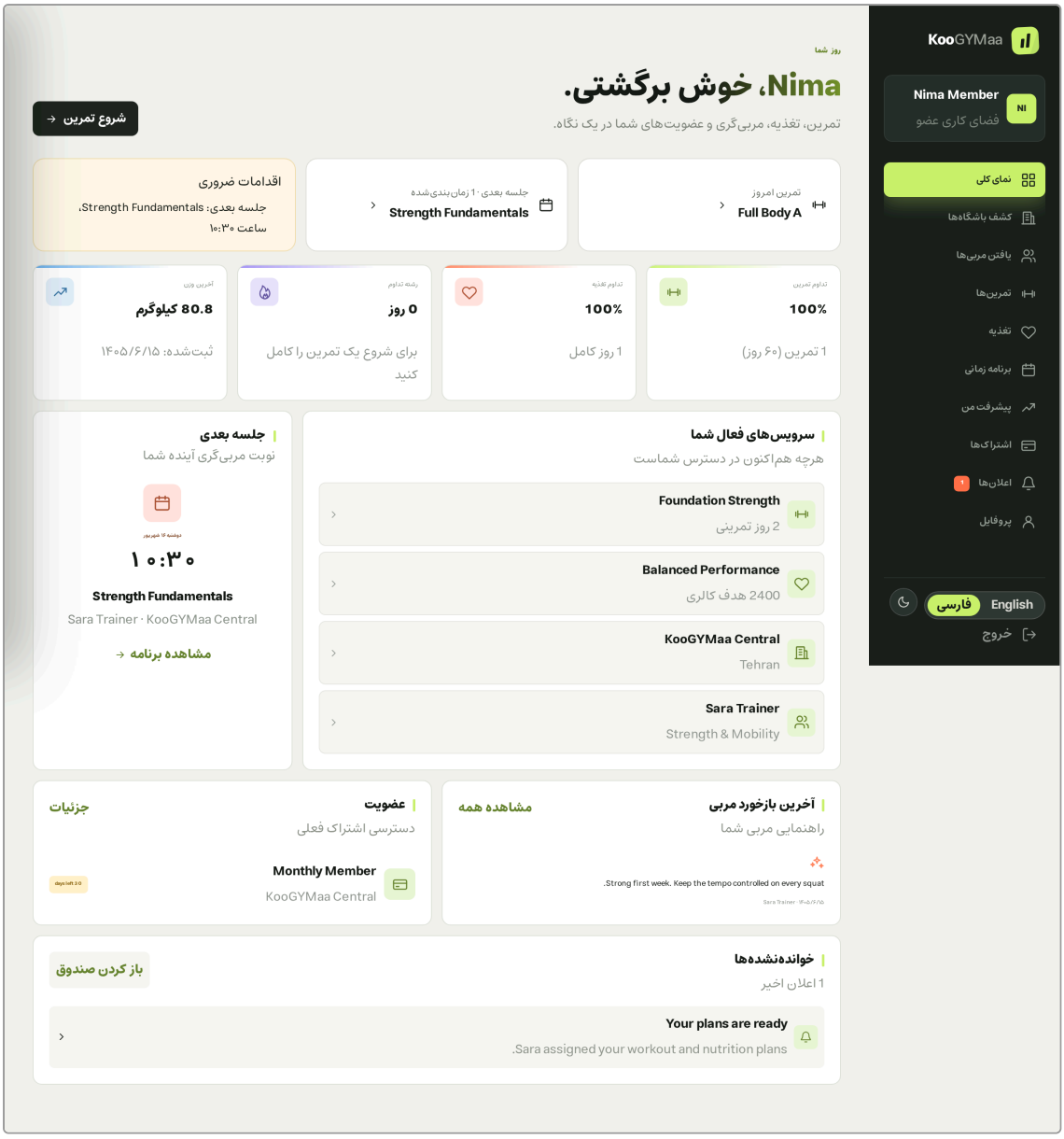


شکل ۳-۳۷ پیشرفت مراجعان

۳-۱۳- راهنمای عضو (ورزشکار)

۳-۱۳-۱- داشبورد عضو

داشبورد عضو (شکل ۳-۳۸) با سلام و تاریخ شمسی امروز آغاز می‌شود و تمرین امروز، استریک (روزهای پیاپی تمرین)، اشتراک فعال و روزهای باقی‌ماندهٔ آن، اهداف و آخرین اعلان‌ها را نشان می‌دهد. نوار کناری (سبز لیمویی) شامل کشف باشگاه‌ها، یافتن مربی‌ها، تمرین‌ها، تغذیه، برنامهٔ زمانی، پیشرفت من، اشتراک‌ها، اعلان‌ها و پروفایل است.



شکل ۳-۳۸ داشبورد عضو

۲-۱۳-۳- پیوستن به باشگاه و خرید اشتراک

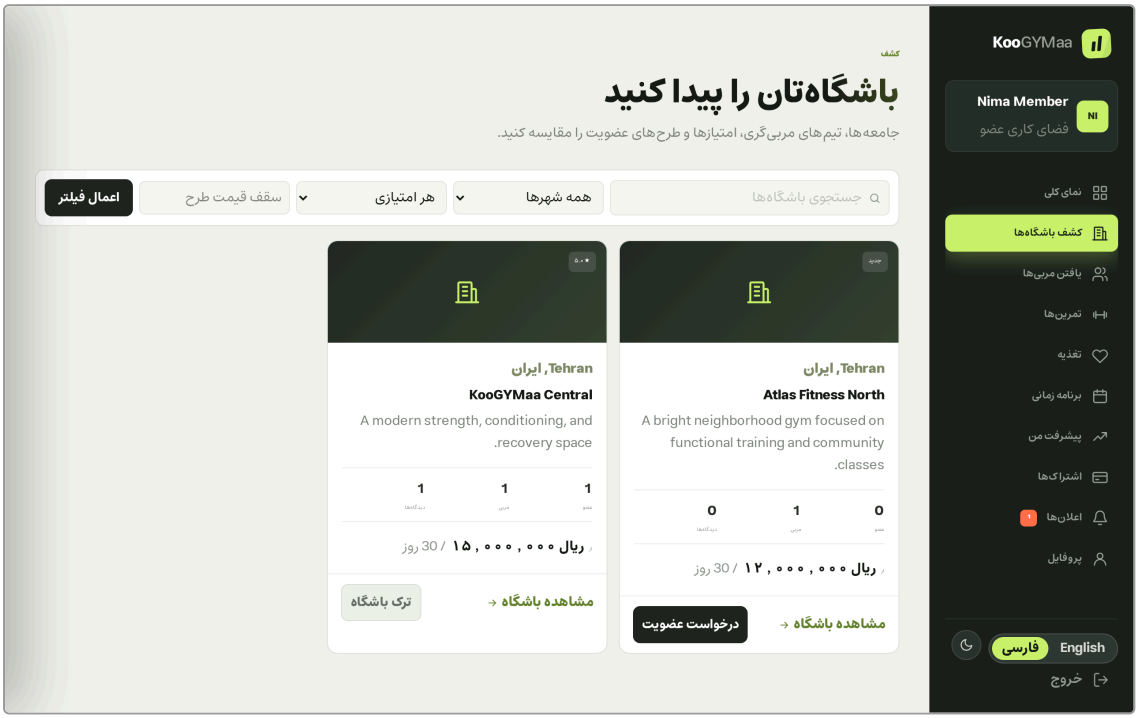
در «کشف باشگاه‌ها» (شکل ۳-۳۹) باشگاه‌های فعال را با شهر، امتیاز، تعداد مربیان و طرح‌ها ببینید. با ورود به صفحهٔ باشگاه:

۱. «درخواست عضویت» را بزنید و منتظر تأیید مدیر بمانید (اعلان دریافت می‌کنید).

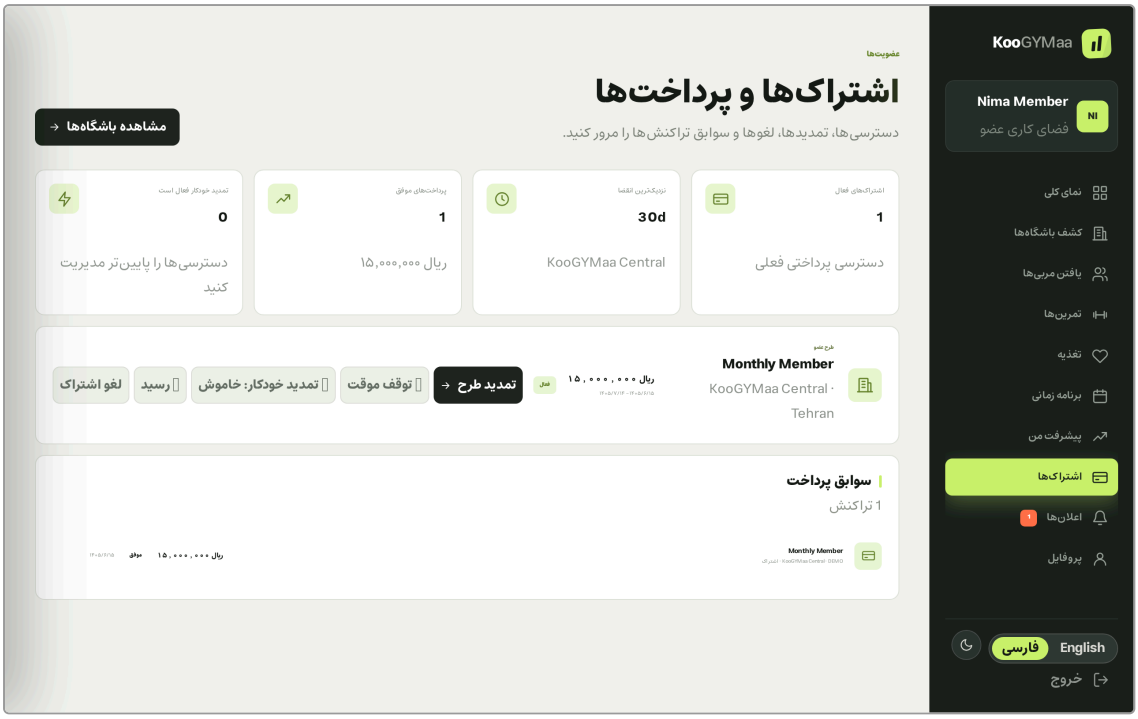
۲. پس از تأیید، یکی از طرح‌های اشتراک را انتخاب و «خرید» را بزنید.

۳. به صفحهٔ پرداخت هدایت می‌شوید؛ پس از پرداخت موفق، اشتراک بلافاصله فعال می‌شود.

۴. در «اشتراک‌ها» (شکل ۳-۴۰) می‌توانید رسید را دانلود کنید، تمدید خودکار را روشن/خاموش کنید، اشتراک را موقتاً متوقف (مثلاً در سفر) و بعداً ازسرگیری کنید، یا آن را لغو کنید.



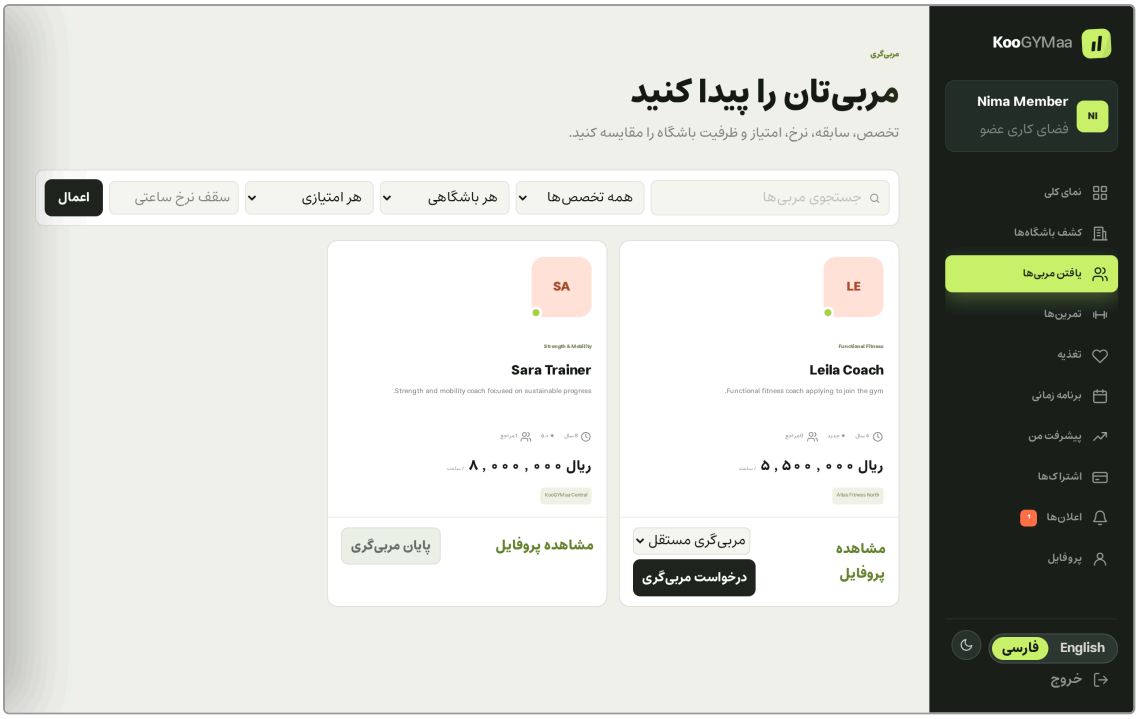
شکل ۳-۳۹ کشف باشگاه‌ها



شکل ۳-۴۰ اشتراک‌های عضو با امکان توقف، ازسرگیری، لغو و دریافت رسید

۳-۱۳-۳- انتخاب مربی

در «یافتن مربی‌ها» (شکل ۳-۴۱) همهٔ مربیان پلتفرم - نه فقط مربیان باشگاه شما - با تخصص، تجربه و امتیاز فهرست شده‌اند. پروفایل مربی را ببینید و «درخواست مربی‌گری» بدهید. پس از پذیرش، مربی می‌تواند برایتان برنامه بنویسد و جلسه بگذارد. می‌توانید هم‌زمان با بیش از یک مربی (مثلاً یکی برای تمرین و یکی برای تغذیه) کار کنید.



شکل ۳-۴۱ یافتن مربی در سراسر پلتفرم

۴-۱۳-۳- اجرای تمرین

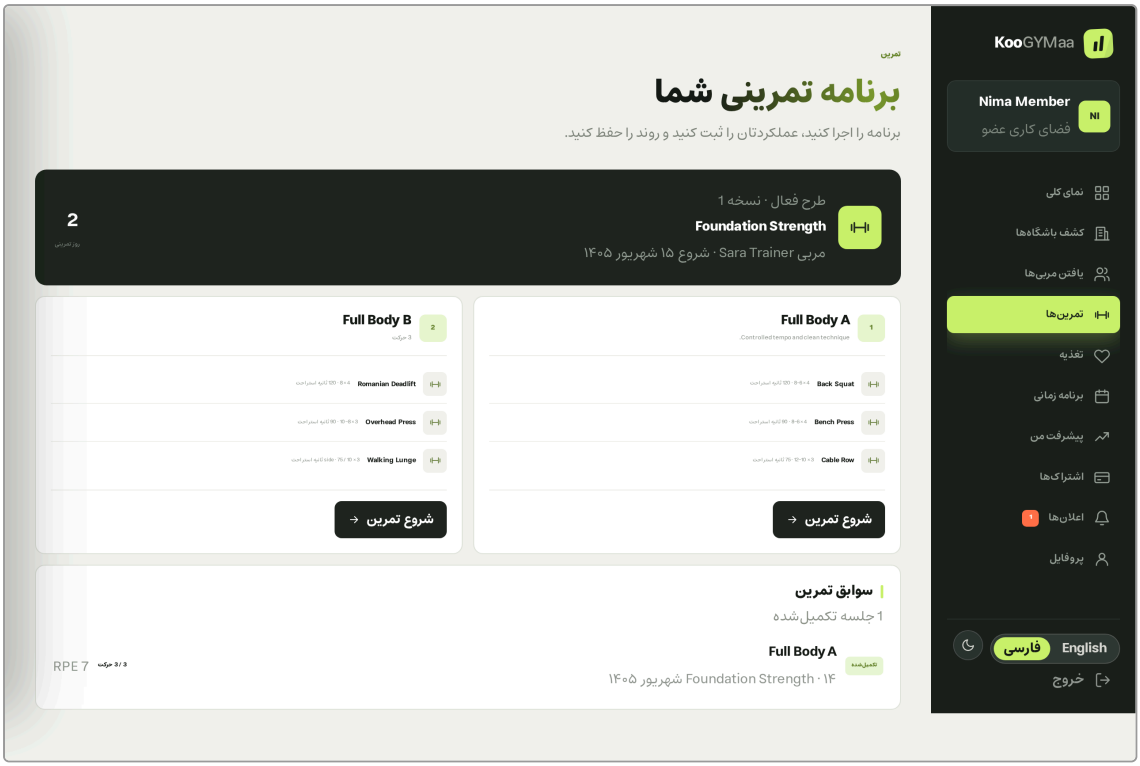
«تمرین‌ها» (شکل ۳-۴۲) برنامهٔ فعال شما را با روزهای آن نشان می‌دهد. برای اجرای یک روز تمرینی:

۱. روی «شروع تمرین» روز مورد نظر کلیک کنید.

۲. برای هر حرکت، پس از انجام هر ست، تعداد تکرار و وزنهٔ واقعی را وارد کنید. تایمر استراحت به‌طور خودکار با زمان تعیین‌شده توسط مربی شروع می‌شود.

۳. در پایان، شدت درک‌شدهٔ کل جلسه (۱ تا ۱۰) و یادداشت اختیاری را ثبت کنید.

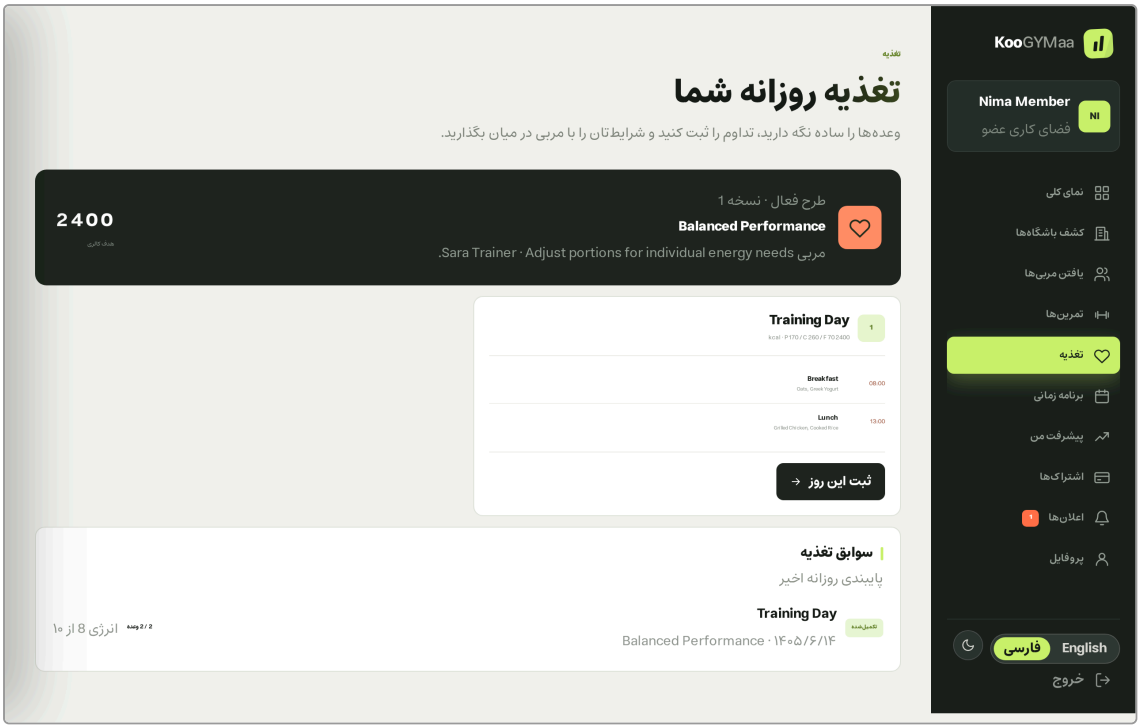
۴. «پایان تمرین» را بزنید. جلسه در «سوابق تمرین» ثبت، استریک شما به‌روز و مربی مطلع می‌شود.



شکل ۳-۴۲ برنامهٔ تمرینی عضو و سوابق اجرا

۵-۱۳-۳- تغذیه

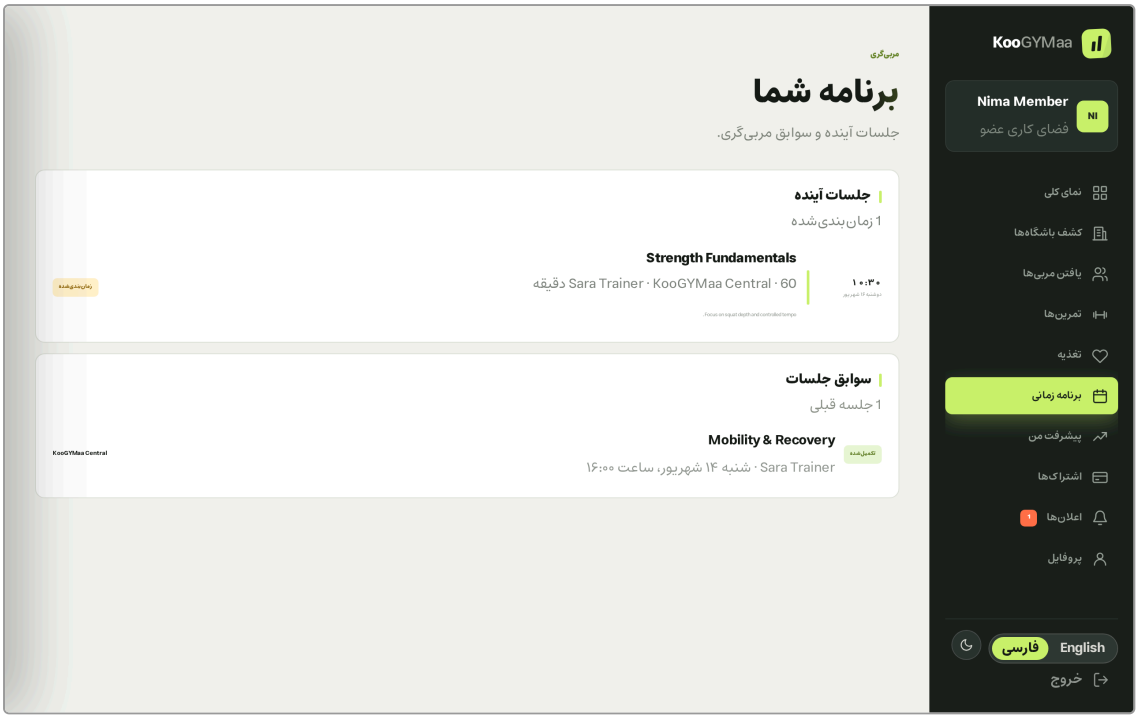
در «تغذیه» (شکل ۳-۴۳) برنامهٔ غذایی تخصیص‌یافته را ببینید و هر وعده را پس از مصرف «تیک» بزنید یا مقدار واقعی را ثبت کنید. جمع کالری روز در مقایسه با هدف نمایش داده می‌شود و سابقهٔ روزهای گذشته برای مربی قابل مشاهده است.



شکل ۳-۴۳ برنامهٔ غذایی و ثبت وعده‌ها

۶-۱۳-۳- برنامهٔ زمانی

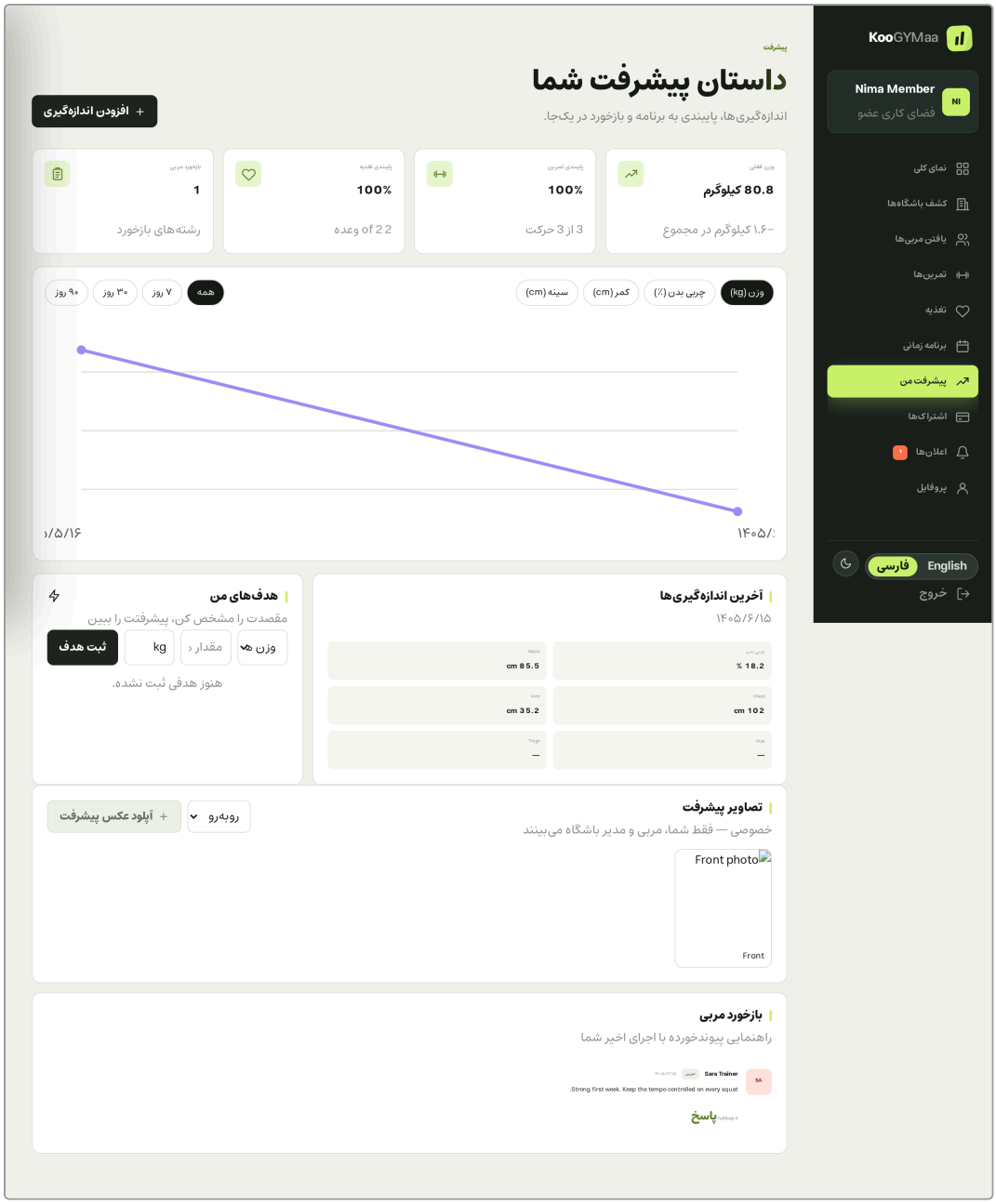
«برنامهٔ زمانی» (شکل ۳-۴۴) جلسات حضوری شما با مربی را در تقویم هفتگی نشان می‌دهد. برای رزرو جلسهٔ جدید، مربی و یکی از بازه‌های آزاد او را انتخاب کنید؛ اگر بازه اشغال شده باشد سامانه اطلاع می‌دهد. یادآوری جلسه طبق تنظیم شما (پیش‌فرض ۲۴ ساعت قبل) ارسال می‌شود.



شکل ۳-۴۴ برنامهٔ زمانی و رزرو جلسه

۷-۱۳-۳- پیگیری پیشرفت

«پیشرفت من» (شکل ۳-۴۵) سه ابزار دارد: ثبت اندازه‌های بدنی (وزن، دور کمر، درصد چربی و...) با نمودار روند؛ آپلود عکس پیشرفت با تاریخ برای مقایسهٔ بصری؛ و تعریف اهداف (مثلاً «رسیدن به وزن ۷۵ کیلوگرم تا پایان تابستان») با نمایش درصد پیشرفت. همهٔ این داده‌ها متعلق به حساب شماست و با تغییر باشگاه یا مربی حفظ می‌شود.



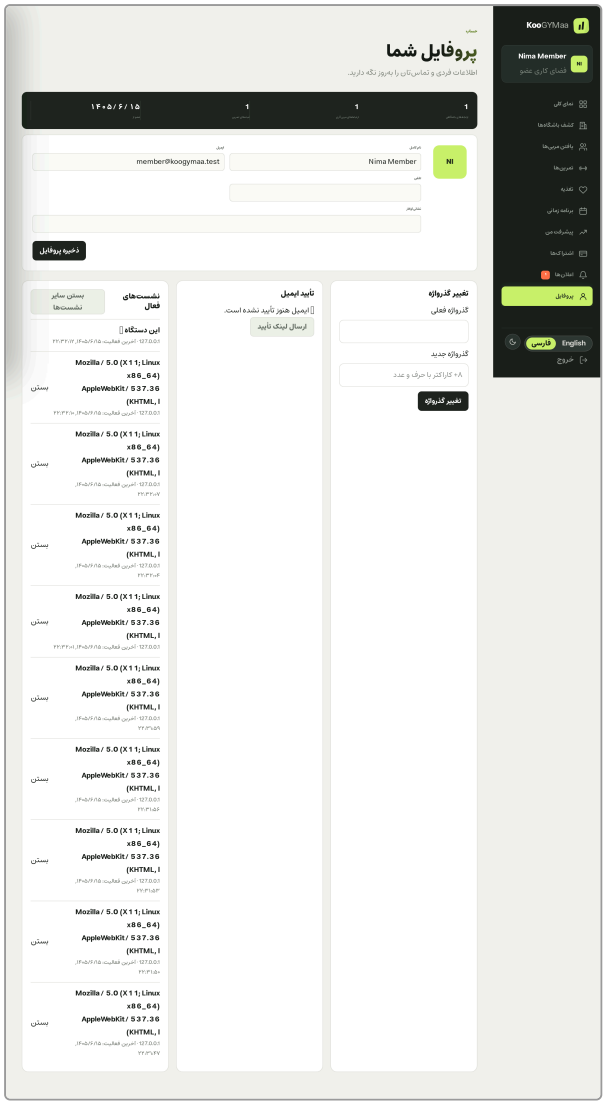
شکل ۳-۴۵ پیگیری پیشرفت: اندازه‌ها، نمودار، عکس‌ها و اهداف

۸-۱۳-۳- اعلان‌ها و پروفایل

«اعلان‌ها» (شکل ۳-۴۶) همهٔ رویدادها - تخصیص برنامه، بازخورد مربی، یادآوری جلسه، هشدار انقضای اشتراک - را فهرست می‌کند و تعداد خوانده‌نشده‌ها روی آیکن زنگ نمایش داده می‌شود. در بخش تنظیمات همین صفحه می‌توانید زمان یادآوری جلسه (ساعت)، هشدار انقضا (روز) و

یادآور بی‌فعالیتی را تنظیم یا خاموش کنید. در «پروفایل» (شکل ۳-۴۷) اطلاعات شخصی، تغییر گذرواژه و فهرست دستگاه‌های واردشده با امکان «خروج از سایر دستگاه‌ها» قرار دارد.

شکل ۳-۴۶ اعلان‌ها و تنظیمات یادآوری



شکل ۳-۴۷ پروفایل عضو، تغییر گذرواژه و نشست‌های فعال

۹-۱۳-۳- استفاده در گوشی همراه

همهٔ صفحات بالا در گوشی همراه نیز قابل استفاده‌اند. در عرض‌های کوچک، منو به دکمهٔ همبرگری در سرصفحه منتقل می‌شود و با لمس آن، کشوی ناوبری با همهٔ بخش‌ها، سوییچ زبان، دکمهٔ روز/شب و خروج باز می‌شود (شکل ۳-۱۵). نوار زبانه‌های زیر سرصفحه نیز دسترسی سریع به بخش‌های پرکاربرد را فراهم می‌کند. فرم‌ها با اندازهٔ قلم مناسب طراحی شده‌اند تا مرورگر گوشی هنگام تمرکز روی فیلد، صفحه را بزرگ‌نمایی نکند.

فصل چهارم
جمع‌بندی، نتیجه‌گیری و پیشنهادها

در این فصل ابتدا دستاوردهای پروژه در برابر اهداف فصل اول ارزیابی می‌شود؛ سپس محدودیت‌های سامانهٔ ساخته‌شده، مزایا و معایب فناوری‌های انتخاب‌شده و تجربیات فنی و غیرفنی پروژه بیان می‌گردد و در پایان، پیشنهادهایی برای ادامهٔ کار ارائه می‌شود.

۴-۱- جمع‌بندی نتایج

جدول ۴-۱ اهداف تعیین‌شده در بخش ۱-۳ را با آنچه محقق شد مقایسه می‌کند. همهٔ اهداف اصلی محقق شده‌اند و دو مورد (درگاه واقعی و اپ بومی) که از ابتدا خارج از دامنه تعریف شده بودند، به کارهای آینده موکول شده‌اند.

جدول ۴-۱ ارزیابی دستاوردها در برابر اهداف

هدف	وضعیت	شاهد
یکپارچگی داده میان سه نقش	محقق شد	۳۳ مدل مرتبط در یک پایگاه داده؛ هر رخداد یک بار ثبت و از سه پنل دیده می‌شود
استقلال از مکان	محقق شد	رابطهٔ مربی-شاگرد مستقل از باشگاه (TrainerClient)؛ مربی چندباشگاهی؛ کشف سراسری مربیان

مالکیت سابقه توسط فرد	محقق شد	همهٔ لاگ‌ها، اندازه‌ها و عکس‌ها کلید خارجی به User دارند نه Gym
اتوماسیون عملیات تکراری	محقق شد	کرون روزانهٔ انقضا/تمدید/یادآوری؛ یادآور کم‌فعالیتی؛ تنظیم شخصی یادآوری‌ها
شفافیت مالی	محقق شد	گزارش به تفکیک ارز، فقط پرداخت موفق، بازپرداخت، CSV، خط زمانی حسابرسی
بومی‌سازی کامل	محقق شد	۵۰۰+ کلید ترجمه، RTL با ویژگی‌های منطقی، تقویم جلالی، ارقام فارسی، فونت خودمیزبان
دسترس‌پذیری روی موبایل	محقق شد	۰ سرریز در ۱۹۸ ترکیب صفحه/عرض/زبان؛ منوی همبرگری؛ اهداف لمسی ≤ ۴۰ پیکسل
امنیت و قابلیت اتکا	محقق شد	JWT، bcrypt + نشست قابل ابطال، محدودیت نرخ، حسابرسی، ۷۶ آزمون سبز
درگاه پرداخت واقعی	خارج از دامنه	واسط PaymentProvider آماده؛ پیاده‌سازی نمایشی با همان قرارداد
اپلیکیشن بومی موبایل	خارج از دامنه	وب واکنش‌گرا پوشش می‌دهد؛ PWA در کارهای آینده

از نظر کمّی، محصول نهایی شامل ۴۷ صفحه، ۷۶ مسیر ۳۳ ،API جدول پایگاه داده، حدود ۱۶ هزار خط کد TypeScript/CSS و ۷۶ آزمون واحد است و در ۱۱ مرحلهٔ اصلی (کامیت) توسعه یافته است. مهم‌تر از این اعداد، آن است که سامانه به‌صورت یک محصول کامل - نه یک نمونهٔ اولیه - قابل استقرار است: مستندات عملیاتی، اسکریپت پشتیبان‌گیری، سلامت‌سنجی و بررسی خودکار کیفیت در هر تغییر، همگی بخشی از تحویل‌اند.

۲-۴- محدودیت‌های سامانهٔ ساخته‌شده

صادقانه باید به محدودیت‌های نسخهٔ فعلی اشاره کرد:

- **درگاه پرداخت نمایشی:** پرداخت واقعی انجام نمی‌شود. اگرچه قرارداد و جریان تأیید کاملاً پیاده شده، اتصال به درگاه بانکی نیازمند قرارداد تجاری و پیاده‌سازی واسط است.

- **ارسال ایمیل و پیامک:** توکن‌های بازیابی و اعلان‌ها در سامانه ثبت می‌شوند اما سرویس ارسال ایمیل/پیامک متصل نیست؛ در محیط عملیاتی باید یک ارائه‌دهنده پیکربندی شود.
- **محدودیت نرخ درون‌حافظه‌ای:** شمارنده‌ها در حافظهٔ فرایند نگهداری می‌شوند و در استقرار چندنمونه‌ای [۱] میان نمونه‌ها مشترک نیستند؛ برای مقیاس بزرگ باید به Redis یا مشابه منتقل شوند.
- **مقیاس پایگاه داده:** SQLite/libSQL برای ده‌ها باشگاه و هزاران کاربر کافی است، اما برای بار نوشتنی بسیار سنگین، مهاجرت به PostgreSQL (که Prisma آن را با تغییر یک خط پشتیبانی می‌کند) توصیه می‌شود.
- **ارزیابی با کاربران واقعی:** سامانه با داده‌های نمونه و بازخورد غیررسمی چند مربی و مدیر ارزیابی شده است؛ مطالعهٔ کاربردپذیری [۲] رسمی با نمونهٔ آماری انجام نشده است.
- **آزمون یکپارچگی سرتاسری:** منطق دامنه با آزمون واحد و رابط کاربری با بررسی خودکار سرریز پوشش داده شده، اما سناریوهای سرتاسری (مثلاً «ثبت‌نام تا خرید اشتراک») هنوز به‌صورت خودکار آزمون نمی‌شوند.

۳-۴- مزایا و معایب فناوری‌های به‌کاررفته

انتخاب‌های فنی پروژه در عمل نقاط قوت و ضعفی نشان دادند که در جدول ۴-۲ خلاصه شده است. مهم‌ترین درس این بود که «نوع‌دار بودن سرتاسری» - از طرحوارهٔ پایگاه داده تا مؤلفهٔ رابط کاربری - بیشترین بازده را در جلوگیری از خطا داشت؛ تقریباً همهٔ بازسازی‌های [۳] بزرگ (مانند تغییر مدل پرداخت) بدون خطای زمان اجرا انجام شدند.

۱ Multi-instance / horizontally scaled

۲ Usability study

۳ Refactoring

جدول ۴-۲ مزایا و معایب مشاهده‌شدهٔ فناوری‌های اصلی

فناوری	مزایا در این پروژه	معایب / چالش‌ها
Next.js + مؤلفه‌های سرور	دسترسی مستقیم به داده در صفحه، حجم کم جاوااسکریپت، یک پروژه برای همه چیز	منحنی یادگیری مرز سرور/کلاینت؛ برخی خطاها فقط در زمان اجرا آشکار می‌شوند
TypeScript سخت‌گیرانه	کشف زودهنگام خطا، بازسازی ایمن، مستندسازی ضمنی	زمان بیشتر برای تعریف نوع‌ها در ابتدا
Prisma	طرحوارهٔ خوانا، مهاجرت خودکار، کلاینت نوع‌دار	برخی پرس‌وجوهای پیچیده به SQL خام نیاز داشتند؛ اندازهٔ بستهٔ تولیدشده
SQLite / libSQL	بدون سرور، پشتیبان‌گیری با کپی فایل، سرعت بالا	قفل نوشتن سراسری؛ نامناسب برای هم‌زمانی نوشتنی بسیار بالا
CSS سفارشی با ویژگی‌های منطقی	کنترل کامل RTL و تم، بدون وابستگی	حجم زیاد برگهٔ سبک (بیش از ۳۵۰۰ خط)؛ نگهداری دشوارتر از سیستم‌های طراحی آماده
JWT + جدول نشست	بدون حالت در سرور برای اعتبارسنجی، ولی قابل ابطال	یک پرس‌وجوی اضافی به‌ازای هر درخواست
Vitest + Playwright	آزمون سریع، بررسی بصری خودکار	نیاز به مرورگر بدون سر در محیط CI

۴-۴- تجربیات پروژه

۴-۴-۱- تجربیات فنی

- **منطق دامنه را از چارچوب جدا کنید.** قرار دادن محاسبات مالی، زمان‌بندی و اعتبارسنجی در توابع خالص پوشهٔ /lib باعث شد بدون راه‌اندازی پایگاه داده یا مرورگر قابل آزمون باشند و همین توابع در سرور و کلاینت مشترک استفاده شوند.
- **راست‌به‌چپ را از روز اول در نظر بگیرید.** افزودن RTL به رابطی که با موقعیت‌های فیزیکی چپ/راست نوشته شده، بسیار پرهزینه‌تر از استفاده از ویژگی‌های منطقی از ابتداست.

- **به سرویس‌های خارجی اعتماد نکنید.** وابستگی به Google Fonts در شبکه‌های داخلی بی‌صدا شکست می‌خورد؛ خودمیزبانی دارایی‌ها هم قابلیت اطمینان و هم حریم خصوصی را بهبود داد.
- **وضعیت‌ها را در پایگاه داده قید کنید.** تعریف انواع شمارشی به‌جای رشتهٔ آزاد، دستهٔ کاملی از خطاها را حذف کرد.
- **بررسی خودکار بصری ارزشمند است.** اسکریپت سادهٔ سنجش سرریز افقی روی همهٔ صفحات، ده‌ها مشکل چیدمان را پیش از دیده شدن توسط کاربر آشکار کرد.

۲-۴-۴- تجربیات غیرفنی

- گفت‌وگو با کاربران واقعی (مدیران و مربیان) پیش از طراحی، اولویت‌ها را به‌کلی تغییر داد؛ برای مثال «یادآوری انقضای اشتراک» که در ابتدا کم‌اهمیت به نظر می‌رسید، پرتکرارترین درخواست مدیران بود.
- توسعهٔ تکرارشونده با تحویل‌های کوچک و قابل نمایش، بازخورد سریع‌تر و انگیزهٔ بیشتری ایجاد کرد تا تلاش برای تکمیل همهٔ قابلیت‌ها پیش از نخستین نمایش.
- مستندسازی هم‌زمان با کد (README، پوشهٔ docs) هزینهٔ بازگشت به بخش‌های قدیمی را به‌شدت کاهش داد.

۴-۵- پیشنهادهایی برای ادامهٔ کار

پیشنهادهای زیر مستقیماً از محدودیت‌های بخش ۴-۲ و بازخورد کاربران برخاسته‌اند و به ترتیب اولویت پیشنهادی مرتب شده‌اند:

۱. **اتصال به درگاه پرداخت واقعی:** پیاده‌سازی واسط PaymentProvider برای یکی از درگاه‌های داخلی (مانند زرین‌پال یا درگاه مستقیم بانکی) و پشتیبانی از کیف پول داخلی برای پرداخت‌های خرد.

۲. **ارسال ایمیل و پیامک:** اتصال به سرویس پیامک داخلی برای یادآوری‌ها و کدهای تأیید، که برای کاربران ایرانی کاربردی‌تر از ایمیل است.

۳. **برنامهٔ وب پیش‌رونده**[1]**:** افزودن مانیفست، سرویس‌ورکر و حالت آفلاین برای ثبت تمرین در سالن‌هایی که اینترنت ضعیف دارند، همراه با اعلان فشاری[2].

۴. **اتصال به دستگاه‌های پوشیدنی:** دریافت ضربان قلب و کالری از ساعت‌های هوشمند از طریق Google Fit / Apple HealthKit برای غنی‌سازی لاگ تمرین.

۵. **پیشنهاددهی هوشمند:** استفاده از سابقهٔ لاگ‌ها برای پیشنهاد افزایش وزنه (اضافه‌بار تدریجی)، تشخیص افت عملکرد و پیشنهاد روز استراحت؛ در گام بعد، تولید پیش‌نویس برنامه با مدل‌های زبانی برای تأیید مربی.

۶. **مطالعهٔ کاربردپذیری رسمی:** اجرای آزمون با نمونه‌ای از هر سه نقش، اندازه‌گیری زمان انجام وظایف کلیدی و نرخ خطا، و بازطراحی بر اساس نتایج.

۷. **چندشعبه‌ای و مجوزهای ریزدانه:** پشتیبانی از زنجیرهٔ باشگاه با گزارش تجمیعی و نقش‌های میانی (پذیرش، حسابدار) با مجوزهای محدودتر از مدیر.

۸. **مقیاس‌پذیری:** مهاجرت اختیاری به PostgreSQL، انتقال محدودیت نرخ و صف اعلان‌ها به Redis، و افزودن آزمون‌های سرتاسری خودکار برای سناریوهای اصلی.

۹. **بازار برنامه‌ها:** امکان انتشار برنامه‌های تمرینی/غذایی مربیان به‌صورت عمومی یا پولی، که هم درآمد مربی و هم دسترسی اعضا به تخصص‌های نادر را افزایش می‌دهد.

۶-۴- سخن پایانی

این پروژه نشان داد که با فناوری‌های متن‌باز امروزی و بدون وابستگی به سرویس‌های خارجی، می‌توان سامانه‌ای ساخت که سه بازیگر اصلی اکوسیستم باشگاه را در یک فضای مشترک قرار دهد، سابقهٔ تمرینی را به خودِ ورزشکار برگرداند و ارتباط مربی و شاگرد را از دیوارهای یک سالن فراتر ببرد. کوجیما نقطهٔ پایان نیست؛ بستری است که با پیشنهادهای بالا می‌تواند به محصولی تجاری برای باشگاه‌های ایرانی تبدیل شود.

۱ Progressive Web App (PWA)

۲ Push notification

پیوست الف
فهرست مسیرهای API

جدول‌های الف-۱ تا الف-۵ همهٔ ۷۶ مسیر API سامانه را به تفکیک نقش فهرست می‌کنند. همهٔ مسیرها به‌جز ثبت‌نام، ورود، فراموشی/بازنشانی گذرواژه، فراخوان برگشتی و سلامت‌سنجی به نشست معتبر نیاز دارند و پاسخ‌ها با قالب JSON و کدهای وضعیت جدول ۲-۱ برگردانده می‌شوند. علامت «…» به معنی پیشوند /api/admin/gyms/[gymId] یا مسیر والد بلافاصله قبل است.

جدول الف-۱ مسیرهای API - احراز هویت

فعل	مسیر	شرح
POST	/api/auth/register	ثبت‌نام با نقش
POST	/api/auth/login	ورود، صدور کوکی نشست
POST	/api/auth/logout	خروج و ابطال نشست
GET	/api/auth/me	اطلاعات کاربر جاری
POST	/api/auth/forgot-password	درخواست توکن بازنشانی
POST	/api/auth/reset-password	بازنشانی با توکن
POST	/api/auth/change-password	تغییر گذرواژه
POST	/api/auth/verify-email	تأیید ایمیل
GET/DELETE	/api/auth/sessions[/id]	فهرست و ابطال نشست‌ها

جدول الف-۲ مسیرهای API - مدیر باشگاه

فعل	مسیر	شرح
GET/POST	/api/admin/gyms	فهرست و ساخت باشگاه
GET/PATCH	/api/admin/gyms/[gymId]	جزئیات و تنظیمات باشگاه
GET/POST	…/members	اعضا و افزودن عضو
PATCH	…/members/[membershipId]	تأیید/رد/تعلیق عضویت
POST	…/members/bulk	عملیات گروهی اعضا
GET/POST	…/trainers	مربیان و دعوت
PATCH	…/trainers/[gymTrainerId]	تأیید/رد همکاری
POST	…/trainers/bulk	عملیات گروهی مربیان
GET/POST	…/plans	طرح‌های اشتراک
PATCH/DELETE	…/plans/[planId]	ویرایش/حذف طرح

GET/POST	…/subscriptions	اشتراک‌ها و ثبت دستی
PATCH	…/subscriptions/[subscriptionId]	فعال/تمدید/لغو
POST	…/subscriptions/bulk	عملیات گروهی
POST	…/payments/[paymentId]/refund	بازپرداخت
GET	…/reports	گزارش مالی به تفکیک ارز
GET	…/export?type=	خروجی CSV

جدول الف-۳ مسیرهای API - مربی

فعل	مسیر	شرح
GET/PATCH	/api/trainer/profile	پروفایل حرفه‌ای
GET/POST	/api/trainer/gyms	باشگاه‌ها و درخواست همکاری
DELETE	/api/trainer/gyms/[gymTrainerId]	پایان همکاری
GET/POST	/api/trainer/clients	شاگردان و دعوت
PATCH	/api/trainer/clients/[clientId]	پذیرش/توقف/پایان
POST	/api/trainer/clients/[clientId]/nudge	ارسال یادآور
GET/POST	/api/trainer/workouts	برنامه‌های تمرینی
GET/PATCH/DELETE	/api/trainer/workouts/[planId]	جزئیات/ویرایش/حذف
POST	…/workouts/[planId]/clone	کپی برنامه
POST	…/workouts/[planId]/assignments	تخصیص به شاگرد
PATCH	/api/trainer/workout-assignments/[id]	تغییر وضعیت تخصیص
GET/POST	/api/trainer/nutrition	برنامه‌های غذایی
GET/PATCH/DELETE	/api/trainer/nutrition/[planId]	جزئیات/ویرایش/حذف
POST	…/nutrition/[planId]/clone | assignments	کپی / تخصیص
GET/PUT	/api/trainer/availability	بازه‌های در دسترس

GET/POST	/api/trainer/sessions	جلسات و رزرو با بررسی تداخل
PATCH/DELETE	/api/trainer/sessions/[sessionId]	تغییر وضعیت/لغو
POST	/api/trainer/feedback	بازخورد روی لاگ شاگرد

جدول الف-۴ مسیرهای API - عضو

فعل	مسیر	شرح
GET/PATCH	/api/user/profile	پروفایل
POST	/api/user/gyms/[gymId]/membership	درخواست عضویت
POST	/api/user/trainers/[trainerId]/request	درخواست مربی‌گری
POST	/api/user/checkout	شروع خرید اشتراک
POST	/api/user/payments/[paymentId]/confirm	تأیید پرداخت نمایشی
POST	…/subscriptions/[id]/pause | resume | cancel	توقف/ازسرگیری/لغو
GET	…/subscriptions/[id]/receipt	رسید چاپی
GET/POST	/api/user/workout-logs	لاگ تمرین
PATCH/DELETE	/api/user/workout-logs/[logId]	ویرایش/حذف
GET/POST	/api/user/nutrition-logs	لاگ تغذیه
GET/POST	/api/user/measurements	اندازه‌های بدنی
POST	/api/user/progress-photos/upload	آپلود عکس
GET/DELETE	/api/user/progress-photos/[photoId]	دریافت/حذف عکس
GET/POST	/api/user/goals	اهداف
PATCH/DELETE	/api/user/goals/[goalId]	به‌روزرسانی/حذف هدف
GET/PATCH	/api/user/notifications	اعلان‌ها و خوانده‌شدن
GET/PUT	/api/user/notifications/settings	تنظیم یادآوری‌ها
POST	/api/user/feedback | …/[id]/reply	بازخورد و پاسخ

POST	/api/user/reviews/gyms/[gymId] | trainers/[id]	امتیازدهی

جدول الف-۵ مسیرهای API - سیستمی

فعل	مسیر	شرح
POST/GET	/api/cron/subscriptions	زمان‌بند روزانه (CRON_SECRET)
POST	/api/webhooks/payments/[provider]	فراخوان برگشتی درگاه (امضای HMAC)
GET	/api/health	سلامت‌سنجی و خطاهای اخیر

پیوست ب
جدول‌های پایگاه داده

طرحوارهٔ کامل در فایل prisma/schema.prisma (۸۷۳ خط) تعریف شده و از طریق شش مهاجرت SQL اعمال می‌شود. جدول ب-۱ مدل‌ها را با نقش و کلیدهای مهم خلاصه می‌کند. چهار جدول کمکی (UserSession, EmailVerification, FitnessGoal, NotificationPreference) در نخستین استفاده با CREATE TABLE IF NOT EXISTS ساخته می‌شوند تا پایگاه‌های موجود بدون مهاجرت جدید کار کنند.

جدول ب-۱ مدل‌های پایگاه داده

مدل	خوشه	کلیدها و شاخص‌های مهم	شرح
User	هویت	email UK; (role,status)	کاربر با نقش سراسری
TrainerProfile	هویت	userId UK	پروفایل حرفه‌ای مربی
Gym	باشگاه	slug UK; status	باشگاه
GymStaff	باشگاه	(gymId,userId) UK	مدیران باشگاه
GymTrainer	باشگاه	(gymId,trainerId) UK; status	همکاری مربی با باشگاه

GymMembership	باشگاه	(gymId,userId) UK; status	عضویت عضو در باشگاه
TrainerClient	باشگاه	(trainerId,clientId) UK	رابطهٔ مربی-شاگرد (مستقل از باشگاه)
SubscriptionPlan	اشتراک	gymId; audience	طرح اشتراک
Subscription	اشتراک	(subscriberId,status); (endDate,status)	اشتراک خریداری‌شده
Payment	اشتراک	idempotencyKey UK; providerReference UK	پرداخت
PaymentEvent	اشتراک	paymentId	رخدادهای درگاه
WorkoutPlan / WorkoutDay / WorkoutExercise	برنامه	trainerId; (planId,dayIndex)	برنامهٔ تمرینی سه‌سطحی
WorkoutAssignment	برنامه	(planId,clientId); status	تخصیص برنامهٔ تمرینی
DietPlan / DietDay / Meal / FoodItem	برنامه	trainerId	برنامهٔ غذایی چهارسطحی
DietAssignment	برنامه	(planId,clientId)	تخصیص برنامهٔ غذایی
TrainerAvailability	زمان‌بندی	(trainerId,weekday)	بازه‌های در دسترس
TrainingSession	زمان‌بندی	(trainerId,startsAt); status	جلسهٔ حضوری
WorkoutLog / ExerciseLog	اجرا	(userId,performedAt)	ثبت اجرای تمرین و ست‌ها
NutritionLog / MealLog	اجرا	(userId,date)	ثبت تغذیه
BodyMeasurement	پیشرفت	(userId,measuredAt)	اندازه‌های بدنی
ProgressPhoto	پیشرفت	userId	عکس پیشرفت (فایل خارج از وب‌روت)

اعلان‌ها	(userId,readAt)	ارتباط	Notification
بازخورد مربی/ شاگرد	(recipientId,createdAt)	ارتباط	Feedback
امتیاز و نظر	(gymId|trainerId,userId) UK	ارتباط	GymReview / TrainerReview
ثبت اعمال اداری	(gymId,createdAt); actorId	حسابرسی	AuditLog
نشست‌های قابل ابطال	userId; revokedAt	امنیت	UserSession
تأیید ایمیل، اهداف، تنظیم یادآوری	userId	کمکی	EmailVerification, FitnessGoal, NotificationPreference

پیوست ج
راهنمای نصب و استقرار

ج-۱- اجرای محلی

پیش‌نیازها: Node.js نسخهٔ ۲۰ یا بالاتر و npm. مراحل زیر سامانه را با پایگاه دادهٔ SQLite محلی و داده‌های نمونه (جدول ج-۲) راه‌اندازی می‌کند؛ متغیرهای محیطی لازم در جدول ج-۱ آمده‌اند:

```
git clone https://github.com/aghrabooti/KooGYMaa.git && cd KooGYMaa
npm install                 # نصب وابستگی‌ها و تولید کلاینت Prisma
cp .env.example .env        # سپس JWT_SECRET را مقداردهی کنید
npx prisma migrate dev      # ساخت پایگاه داده از مهاجرت‌ها
npm run db:seed             # داده نمونه: admin@ / trainer@ / member@koogymaa.test
npm run dev                 # http://localhost:3000
```

ج-۲- متغیرهای محیطی

جدول ج-۱ متغیرهای محیطی

متغیر	الزامی	شرح
DATABASE_URL	بله (توسعه)	مسیر فایل SQLite، مانند file:./dev.db
LIBSQL_DATABASE_URL	بله (عملیاتی)	نشانی پایگاه دادهٔ میزبانی‌شدهٔ Turso؛ بر DATABASE_URL اولویت دارد
LIBSQL_DATABASE_AUTH_TOKEN	بله (عملیاتی)	توکن دسترسی Turso
JWT_SECRET	بله	رشتهٔ تصادفی حداقل ۳۲ کاراکتر برای امضای نشست
PAYMENT_PROVIDER	خیر	شناسهٔ درگاه؛ پیش‌فرض demo
PAYMENT_WEBHOOK_SECRET	بله	کلید HMAC برای امضای فراخوان برگشتی
CRON_SECRET	بله (عملیاتی)	توکن محافظ مسیر کرون
APP_URL	بله	نشانی عمومی برنامه برای ساخت پیوندها
SEED_PASSWORD	خیر	گذرواژهٔ حساب‌های نمونه

ج-۳- استقرار عملیاتی

استقرار توصیه‌شده روی Vercel با پایگاه دادهٔ Turso است: مخزن را به Vercel متصل کنید، متغیرهای جدول بالا را تعریف کنید، با npm run db:turso:setup مهاجرت‌ها را روی پایگاه میزبانی‌شده اجرا کنید و یک Vercel Cron با زمان‌بندی * * * 2 0 به مسیر /api/cron/subscriptions اضافه کنید. سلامت سامانه از /api/health قابل پایش است و

npm run db:backup نسخهٔ پشتیبان می‌گیرد. دستور npm run check پیش از هر استقرار، تحلیل ایستا، آزمون‌ها، بررسی نوع و ساخت را یک‌جا اجرا می‌کند.

ج-۴- حساب‌های نمونه

جدول ج-۲ حساب‌های دادهٔ نمونه

نقش	ایمیل	توضیح
مدیر باشگاه	admin@koogymaa.test	مدیر دو باشگاه نمونه (Atlas Fitness North، KooGYMaa Central)
مربی	trainer@koogymaa.test	مربی با یک شاگرد فعال و برنامه‌های نمونه
عضو	member@koogymaa.test	عضو با اشتراک فعال و برنامهٔ تمرینی تخصیص‌یافته

منابع و مراجع

منابع به ترتیب حروف الفبای نام نویسندهٔ اول، جداگانه برای منابع فارسی و غیرفارسی، فهرست شده‌اند. حرف پس از خط تیره نوع منبع را نشان می‌دهد: م = مقاله، ک = کتاب، و = صفحهٔ وب، گ = گزارش، پ = پایگاه اطلاعاتی (به ترتیب D، R، W، B، P و T = رساله برای منابع انگلیسی).

[سمیعی-ک۸۰] سمیعی (گیلانی)، احمد؛ نگارش و ویرایش؛ چاپ سوم، انتشارات سمت، تهران؛ ۱۳۸۰.

[دانشگاه سمنان-گ۸۷] دانشکدهٔ برق و کامپیوتر دانشگاه سمنان؛ شیوه‌نامهٔ تدوین و ارائهٔ پایان‌نامه؛ ویرایش دوم؛ شهریور ۱۳۸۷.

[روحانی-ک۹۴] روحانی رانکوهی، سید محمدتقی؛ مفاهیم بنیادی پایگاه داده‌ها؛ ویرایش سوم، انتشارات جلوه، تهران؛ ۱۳۹۴.

[Borkowski-P96] Borkowski, K. M.; “The Persian Calendar for 3000 Years;” *Earth, Moon, and Planets*; vol. 74, 1996; pp. 223–230.

[Burke-P11] Burke, L. E.; Wang, J.; Sevick, M. A.; “Self-Monitoring in Weight Loss: A Systematic Review of the Literature;” *Journal of the American Dietetic Association*; vol. 111, no. 1, 2011; pp. 92–102.

[Codd-P70] Codd, E. F.; “A Relational Model of Data for Large Shared Data Banks;” *Communications of the ACM*; vol. 13, no. 6, 1970; pp. 377–387.

[Cohn-B09] Cohn, M.; *Succeeding with Agile: Software Development Using Scrum*; Addison-Wesley; 2009.

[CSSWG-W25] W3C CSS Working Group; *CSS Logical Properties and Values Level 1*; Available at https://www.w3.org/TR/css-logical-1/; Accessed Aug. 2026.

[Fielding-T00] Fielding, R. T.; *Architectural Styles and the Design of Network-based Software Architectures*; Ph.D. Dissertation, University of California, Irvine; 2000.

[IHRSA-R20] International Health, Racquet & Sportsclub Association; *The 2020 IHRSA Global Report*; IHRSA, Boston; 2020.

[Jones-R15] Jones, M.; Bradley, J.; Sakimura, N.; *JSON Web Token (JWT)*; RFC 7519, IETF; May 2015.

[Marcotte-W10] Marcotte, E.; "Responsive Web Design;" *A List Apart*; no. 306, 2010; Available at https://alistapart.com/article/responsive-web-design/; Accessed Aug. 2026.

[Mindbody-W25] Mindbody Inc.; *Mindbody Business Software*; Available at https://www.mindbodyonline.com/; Accessed Aug. 2026.

[Next-W25] Vercel Inc.; *Next.js Documentation: App Router*; Available at https://nextjs.org/docs; Accessed Aug. 2026.

[OWASP-W25] OWASP Foundation; *Session Management Cheat Sheet*; Available at https://cheatsheetseries.owasp.org/; Accessed Aug. 2026.

[Prisma-W25] Prisma Data Inc.; *Prisma ORM Documentation*; Available at https://www.prisma.io/docs; Accessed Aug. 2026.

[Provos-P99] Provos, N.; Mazières, D.; "A Future-Adaptable Password Scheme;" *Proceedings of the USENIX Annual Technical Conference*; 1999; pp. 81–92.

[React-W25] Meta Open Source; *React Documentation: Server Components*; Available at https://react.dev/reference/rsc/server-components; Accessed Aug. 2026.

[Sandhu-P96] Sandhu, R. S.; Coyne, E. J.; Feinstein, H. L.; Youman, C. E.; "Role-Based Access Control Models;" *IEEE Computer*; vol. 29, no. 2, 1996; pp. 38–47.

[SQLite-W25] Hipp, D. R.; *SQLite Documentation*; Available at https://www.sqlite.org/docs.html; Accessed Aug. 2026.

[Stripe-W25] Stripe Inc.; *Idempotent Requests – Stripe API Reference*; Available at https://stripe.com/docs/api/idempotent_requests; Accessed Aug. 2026.

[Trainerize-W25] ABC Fitness Solutions; *ABC Trainerize*; Available at https://www.trainerize.com/; Accessed Aug. 2026.

[WCAG-W23] W3C Web Accessibility Initiative; *Web Content Accessibility Guidelines (WCAG) 2.2*; W3C Recommendation; Oct. 2023.

Abstract

The rapid growth of the fitness industry has not been matched by its management tooling. Gym owners, trainers and athletes typically rely on disconnected instruments—cash-register software, spreadsheets, paper programs and messaging apps—so training history is fragmented, subscription renewals are tracked by hand, and the choice of a coach is effectively limited to whoever happens to work at the nearest gym. When an athlete changes gyms, the relationship with the coach and the entire training record are usually lost.

This thesis presents the design and implementation of **KooGYMaa**, an integrated web platform that brings the three roles—gym administrator, trainer and member—into a single shared workspace. Administrators approve members and trainers, define subscription plans, monitor payments with per-currency financial reports and export CSV data. Trainers author structured workout and nutrition programs, assign them to clients, schedule in-person sessions with conflict detection and give feedback. Members execute and log their programs, track body measurements, streaks and goals, purchase subscriptions and may work with any gym or trainer on the platform, while their history remains attached to their own account. The system is built with Next.js 16, React 19, TypeScript and Prisma 7 on SQLite/libSQL; it is fully bilingual (right-to-left Persian with the Jalali calendar, and English), responsive with a dark mode, and includes JWT authentication with revocable sessions, rate limiting, audit logging and automated subscription life-cycle management. Domain logic is verified by 76 unit tests and the responsive layout by 198 automated page/viewport checks.

The result is a deployable product that removes tool fragmentation, lowers gym operating overhead, extends the coach–athlete relationship beyond the walls of a single facility and preserves the athlete's training record independently of any gym.

Keywords: Gym management system; Integrated web platform; Workout and nutrition programs; Subscription and payment management; Next.js; Prisma; Persian localization.

Semnan University
Faculty of Electrical & Computer Engineering

B.Sc. Thesis in Computer Engineering – Software

Design & Implementation of an Integrated Gym, Trainer and Athlete Management Platform (KooGYMaa)

By:

.....

Supervisor:

.....

.....

Semnan University
Faculty of Electrical & Computer Engineering

B.Sc. Thesis in Computer Engineering – Software

Design & Implementation of an Integrated Gym, Trainer and Athlete Management Platform (KooGYMaa)

By:

.....

Supervisor:

.....

.....