

# مستند فنی تدریسینو

معماری، ساختار کد، مدل داده، API ها، دستیار هوش مصنوعی و داده‌ی Qdrant، گیمیفیکیشن، امنیت، استقرار و مسیر مقیاس‌پذیری – برای خواننده‌ای که با پروژه آشنایی قبلی ندارد و می‌خواهد آن را توسعه دهد یا نگهداری کند.

|          |                                                                                                                                                                |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| تاریخ    | ۲۲ مهر ۱۴۰۵ (۱۴ اکتبر ۲۰۲۶)                                                                                                                                    |
| نویسنده  | علیرضا زارعیان – بک‌اند و فرانت‌اند                                                                                                                            |
| مخاطب    | توسعه‌دهنده‌ی تازه‌وارد، مدیر فنی، و هر کسی که باید درباره‌ی ادامه‌ی پروژه تصمیم بگیرد                                                                         |
| کد مبنا  | شاخه‌ی <code>main</code> هر دو مخزن در همین تاریخ: <code>Tadrisino/MathApp</code> (بک‌اند) و <code>Tadrisino/MathApp_frontend</code> (فرانت و پوسته‌ی اندروید) |
| نشانی‌ها | اپ: <code>math.tadrisino2.ir</code> · API: <code>mathapi.tadrisino2.ir</code> · مستندات زنده‌ی API: <code>/api/docs/</code>                                    |

این فایل نسخه‌ی Markdown «مستند-فنی-تدریسینو» است. نسخه‌ی PDF با همان محتوا و صفحه‌آرایی کامل کنار همین فایل است: `technical-handbook.pdf`.

## ۱. معرفی پروژه در یک نگاه

تدریسینو یک اپ آموزش ریاضی به سبک دولینگو است: دانش آموز روی نقشه‌ای از «جزیره»ها جلو می‌رود، هر جزیره یک آزمون کوتاه از مسئله‌های خود کتاب درسی است، و هر جا گیر کرد از دستیار هوش مصنوعی («پی») می‌پرسد که جوابش را از روی همان کتاب می‌سازد.

| شاخص                                              | مقدار            |
|---------------------------------------------------|------------------|
| پایه‌های سوم تا دوازدهم                           | ۱۰ پایه          |
| کتاب (ریاضی، حسابان، هندسه، آمار و احتمال، گسسته) | ۱۵ کتاب در ۵ درس |
| سؤال چهارگزینه‌ای                                 | ۱۸,۵۲۲           |
| عملیات API، مستند در Swagger                      | ۶۱               |
| تست خودکار، همه سبز                               | ۹۷۳              |
| اپ جنگو، هرکدام یک مسئولیت                        | ۱۴               |
| تکه‌ی کتاب در (Qdrant ۱۸ collection)              | ۵,۸۱۸            |
| همراه گیمیفیکیشن، هرکدام ۶ حالت چهره              | ۴                |
| بک‌اند و فرانت؛ یک قرارداد API                    | ۲ مخزن           |

## کاربرها و کاری که می‌کنند

| کاربر               | چه می‌کند                                                                                                                                                                                                                                                                           | کجای کد                                                                   |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| دانش آموز           | از هر سنی (بزرگسالی که فقط یک مبحث را می‌خواهد هم پایه‌ی همان مبحث را انتخاب می‌کند). با شماره‌ی موبایل و کد پیامکی وارد می‌شود؛ جزیره‌ها را بازی می‌کند، از دستیار می‌پرسد، امتیاز و سکه و زنجیره‌ی روزانه جمع می‌کند، همراه انتخاب می‌کند، اشتراک می‌خرد، تیکت پشتیبانی می‌فرستد. | <code>progress/</code><br><code>content/</code><br><code>ai_tutor/</code> |
| والد                | حساب جدا ندارد: «گزارش والدین» را در حساب خود دانش آموز می‌بیند — زمان تمرین، سوال‌ها و مقایسه با هفته‌ی پیش، با دو نمودار.                                                                                                                                                         | <code>progress/report.py</code>                                           |
| ادمین (کارمند شرکت) | از پنل <code>/admin/</code> سؤال، درس، پلن، همراه و تیکت را بدون برنامه‌نویس ویرایش می‌کند؛ سؤال‌ها را فله‌ای وارد و خارج می‌کند.                                                                                                                                                   | <code>*/admin.py</code>                                                   |

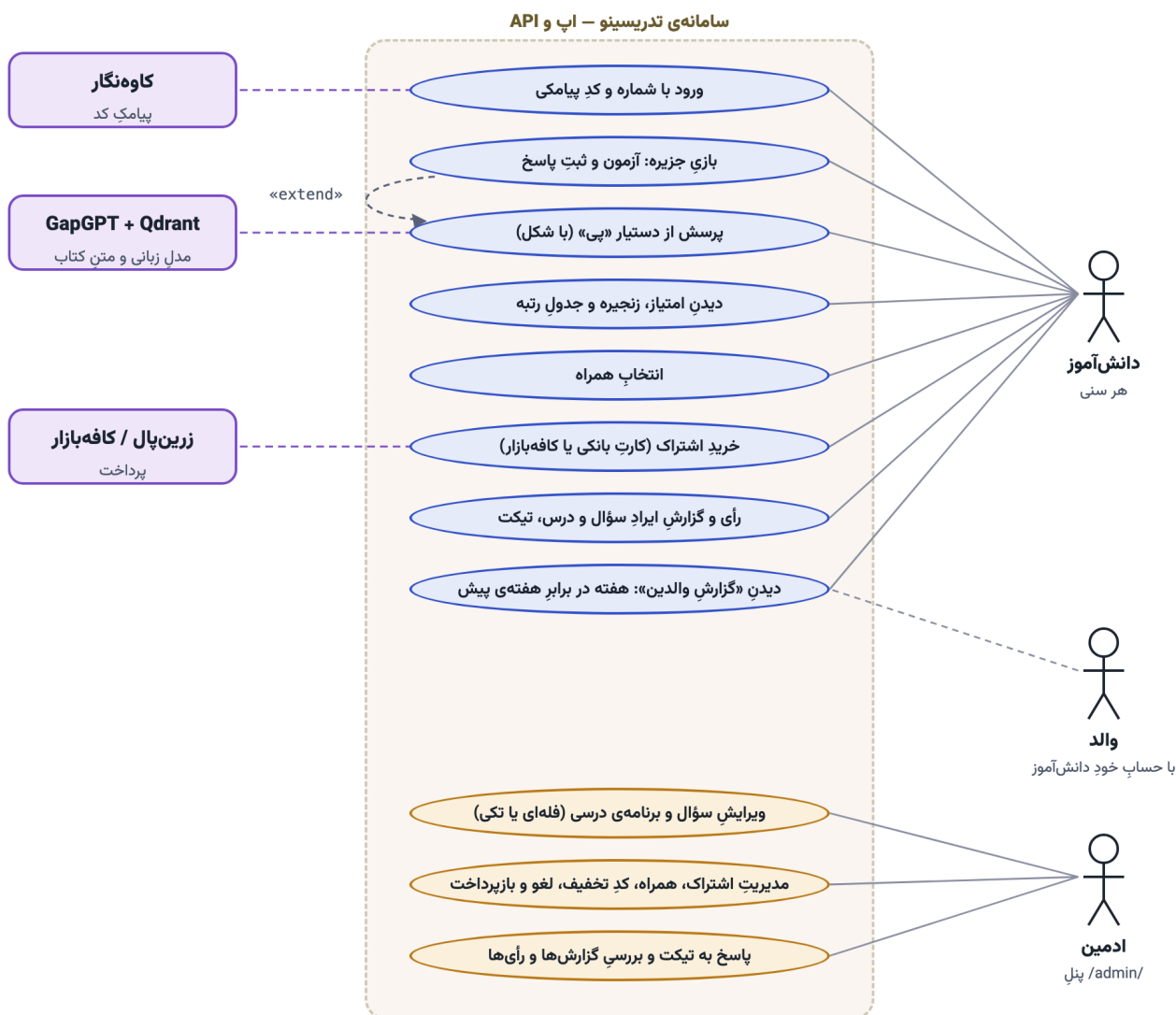
## تاریخچه‌ی کوتاه

| زمان       | اتفاق                                                                                                               |
|------------|---------------------------------------------------------------------------------------------------------------------|
| مرداد ۱۴۰۵ | طراحی معماری و مدل‌های داده؛ هویت کاربر؛ جداسازی اپ‌ها. بازبینی امنیتی اول و بستن حفره‌ها، محدودیت نرخ در کل سیستم. |

| زمان        | اتفاق                                                                                                                                                                                                                                                                                            |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| شهریور      | دستیار از پاسخ ساختگی به RAG واقعی روی کتاب رسید؛ اتصال زرین پال؛ گیمیفیکیشن؛ خواندن پنج کتاب ریاضی و ساخت هزاران سؤال تأییدشده؛ حسابان ۱۱ و ۱۲.                                                                                                                                                 |
| ۲ تا ۷ مهر  | فرانت به ما سپرده شد و از نو ساخته شد؛ هر دو روی دامنه‌ی تازه زنده شدند؛ پیامک واقعی؛ شکل در جواب دستیار؛ همراه چهارم؛ بازبینی نهایی روی سایت زنده.                                                                                                                                              |
| ۸ تا ۲۲ مهر | ورود مهمان حذف شد؛ کلید امضای تازه؛ PostgreSQL و Redis اختصاصی و کپی زنده‌ی دیتابیس روی سرور دوم؛ دیوار پرداخت و اشتراک یک/سه/شش ماهه از زرین پال و کافه بازار؛ پایه‌های چهارم و پنجم و پنج کتاب تازه (هندسه، آمار، گسسته)؛ جزیره‌های تمرین و ۱۸ هزار سؤال؛ سخت کردن سرور و آزمون بار هزار نفره. |

## مدل سناریو (Use Case)

چه کسانی با سامانه کار می‌کنند و هرکدام چه کاری انجام می‌دهند. خط چین بنفش یعنی آن کار به یک سرویس بیرونی وابسته است. «extend» یعنی دانش آموز از دل آزمون، پس از پاسخ نادرست، مستقیم از دستیار می‌پرسد «چرا غلط شد؟». جزئیات گام به گام مهم ترین سناریوها در بخش ۶ آمده است.



شکل ۱-۱ – مدل سناریو: نقش‌های کاربری (والد با حساب خود دانش آموز)، کارهای اصلی، و سرویس‌های بیرونی.

## واحدهای محتوا (این واژه‌ها در کل سند تکرار می‌شوند)

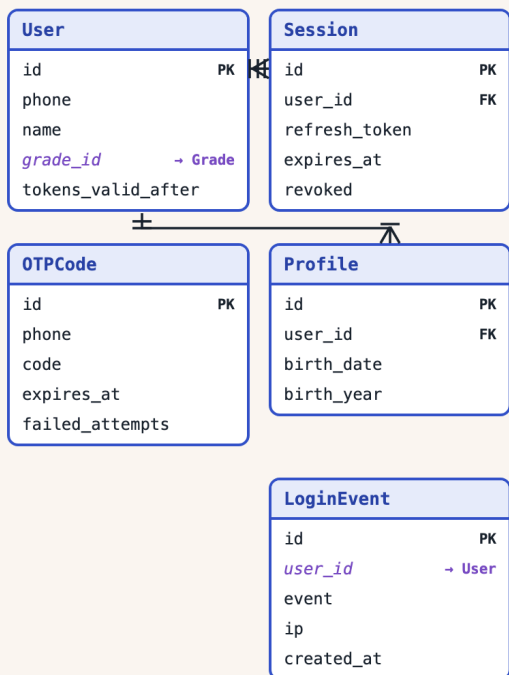
پایه ۱۰ ← کتاب ۵ Subject ← فصل ۸۹ Unit ← جزیره ۱,۱۴۵ Node ← درس ۵,۴۰۶ Lesson ← سؤال Question  
۱۸,۵۲۲ ← گزینه Choice

«درس» در این پروژه یعنی یک «تیپ مسئله»ی کتاب، نه یک جلسه‌ی کلاس. برای همین درس خیلی ریز است و اپ در سطح جزیره (Node) بازی می‌شود: آزمون هر جزیره از درس‌هایش سؤال برمی‌دارد.

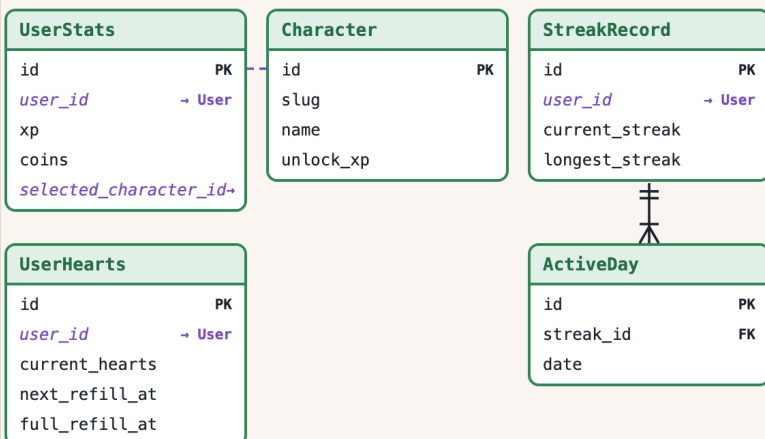
## مدل رابطه‌ای (ER)

جدول‌های اصلی دیتابیس و ارتباط میان آن‌ها، دسته‌بندی شده بر اساس اپ. خط ممتد یک کلید خارجی واقعی (FK) در داخل یک اپ است: دو خط موازی سمت «دقیقاً یک» و پای کلاغی سمت «صفر یا چند» (مثلاً هر پایه چند فصل دارد). فیلد بنفش با → ارجاع سبک به جدول اپ دیگر است: فقط شناسه ذخیره می‌شود و قید دیتابیزی ندارد تا هر اپ مستقل بماند (بخش ۳). خط چین بنفش دو نمونه از همین ارجاع‌های سبک را نشان می‌دهد. فیلدهای کم‌اهمیت حذف شده‌اند؛ فهرست کامل در بخش ۷ و در `<app>/models.py`.

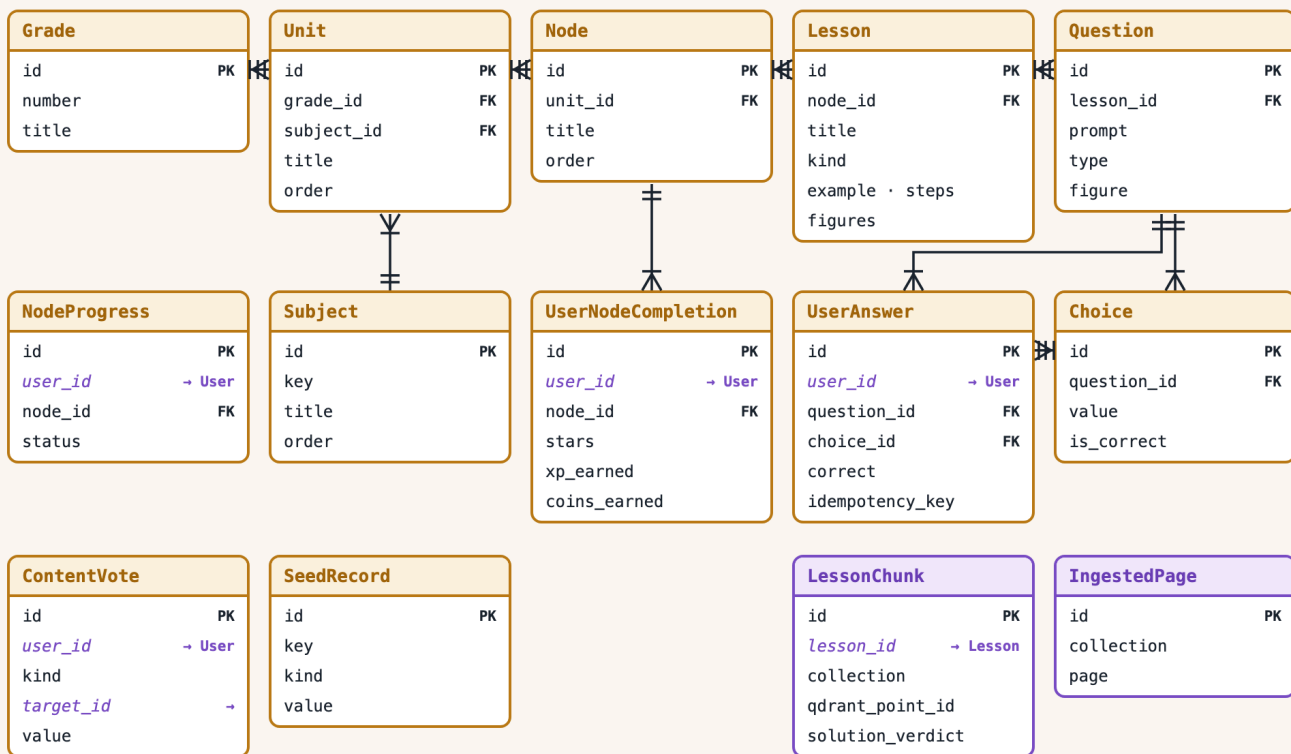
## accounts · هویت



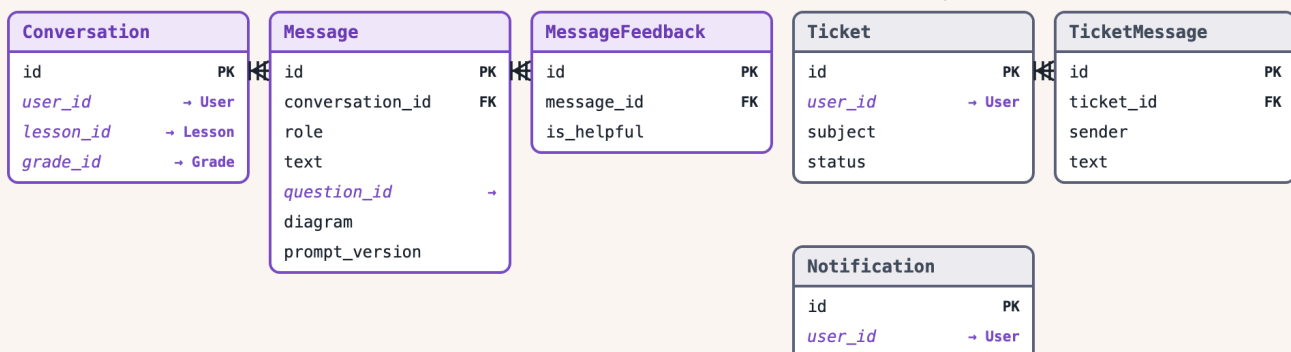
## progress · بازی و پیشرفت

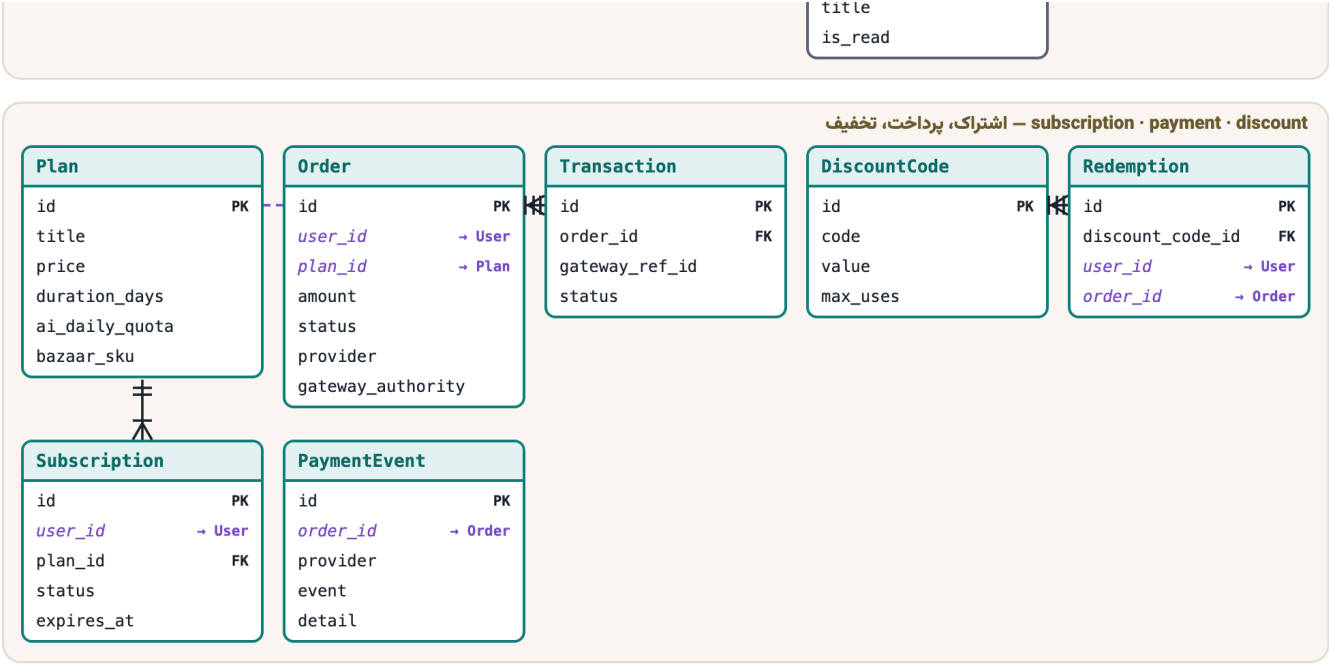


## content · rag – محتوا، فعالیت کاربر و کتاب



## ai\_tutor · support · notifications – دستیار، پشتیبانی، اعلان





## ۲. انتخاب‌های فناوری و چرایی‌شان

هر انتخاب یک دلیل و یک هزینه دارد. ستون آخر هزینه را صادقانه می‌گوید، چون همان چیزی است که نفر بعدی باید بداند.

| لایه         | انتخاب                                            | چرا                                                                                                                                                                                                                                                                                                                                | چرا نه جایگزین                                                                                                                                                                                            | هزینه / ریسک                                                                                                                               |
|--------------|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| بک‌اند       | Django 5.2 + Django REST Framework                | ORM و migration و پنل ادمین آماده در یک بسته؛ پنل ادمین یعنی محتوا بدون برنامه‌نویس ویرایش می‌شود. DRF احراز هویت، مجوز، محدودیت نرخ و سریال‌سازی را استاندارد می‌دهد. پایتون همان زبان کتابخانه‌های هوش مصنوعی است ( <code>pymupdf</code> ، <code>qdrant-client</code> ، <code>openai</code> ).                                   | FastAPI سریع‌تر و async است ولی ادمین، ORM و migration را باید جدا سرهم کرد. Node.js زبان بک‌اند را از زبان لایه‌ی AI جدا می‌کرد.                                                                         | جنگو پیش‌فرض همگام (sync) است؛ جواب طولانی دستیار یک نخ را نکه می‌دارد. با <code>worker × ۳۲ ۴</code> نخ و سهم جدای دستیار حل شد (بخش ۱۴). |
| قرارداد API  | OpenAPI با drf-spectacular                        | اسکیما از خود کد ساخته می‌شود، پس هیچ‌وقت از کد عقب نمی‌ماند. Swagger <code>/api/docs/</code> قرارداد مشترک دو مخزن است.                                                                                                                                                                                                           | سند دستی یا «در ذهن کسی» – که دقیقاً باعث ناهمخوانی اولیه‌ی فرانت و بک‌اند شد.                                                                                                                            | –                                                                                                                                          |
| فرانت        | React 18 + TypeScript (strict) + Vite + Tailwind  | فرانت اولیه را توسعه‌دهنده‌ی قبلی با React ساخته بود؛ در بازسازی نکه داشته شد چون بزرگ‌ترین اکوسیستم را دارد و نیروی جایگزینش راحت پیدا می‌شود. TypeScript ناهمخوانی شکل داده با API را در زمان build می‌گیرد. Vite build سریع و Tailwind رنگ و فاصله‌ی یکدست. وب موبایل محور یعنی بدون نصب و بدون فروشگاه روی هر گوشی باز می‌شود. | React Native / Flutter نیاز انتشار در فروشگاه و build جدا داشت. نسخه‌ی اندروید یک پوسته‌ی روی WebView همین سایت است که فقط پرداخت کافه‌بازار را بومی انجام می‌دهد ( <code>android/</code> در مخزن فرانت). | بدون کتابخانه‌ی router و state؛ کمتر وابستگی، ولی الگوی داخلی را باید خواند (بخش ۴).                                                       |
| دیتابیس و کش | PostgreSQL 14 (cluster اختصاصی) + Redis (اختصاصی) | قفل ردیفی واقعی برای پرداخت، قلب و سهمیه؛ چند نویسنده‌ی هم‌زمان؛ کپی زنده‌ی لحظه‌ای به سرور دوم با ابزار خود PostgreSQL. Redis سقف‌های نرخ و کش را بین workerها مشترک می‌کند. توسعه‌ی محلی و تست‌ها روی SQLite می‌مانند.                                                                                                           | Redis و PostgreSQL موجود ماشین مال تیم‌های دیگرند؛ برای این‌که به داده‌ی دیگران دست نزنیم، نمونه‌ی جدای خودمان را داریم.                                                                                  | دو سرویس بیشتر برای نگهداری؛ اتصال‌ها از pool هر worker می‌آیند چون <code>max_connections</code> سرور ۶۰ است.                              |

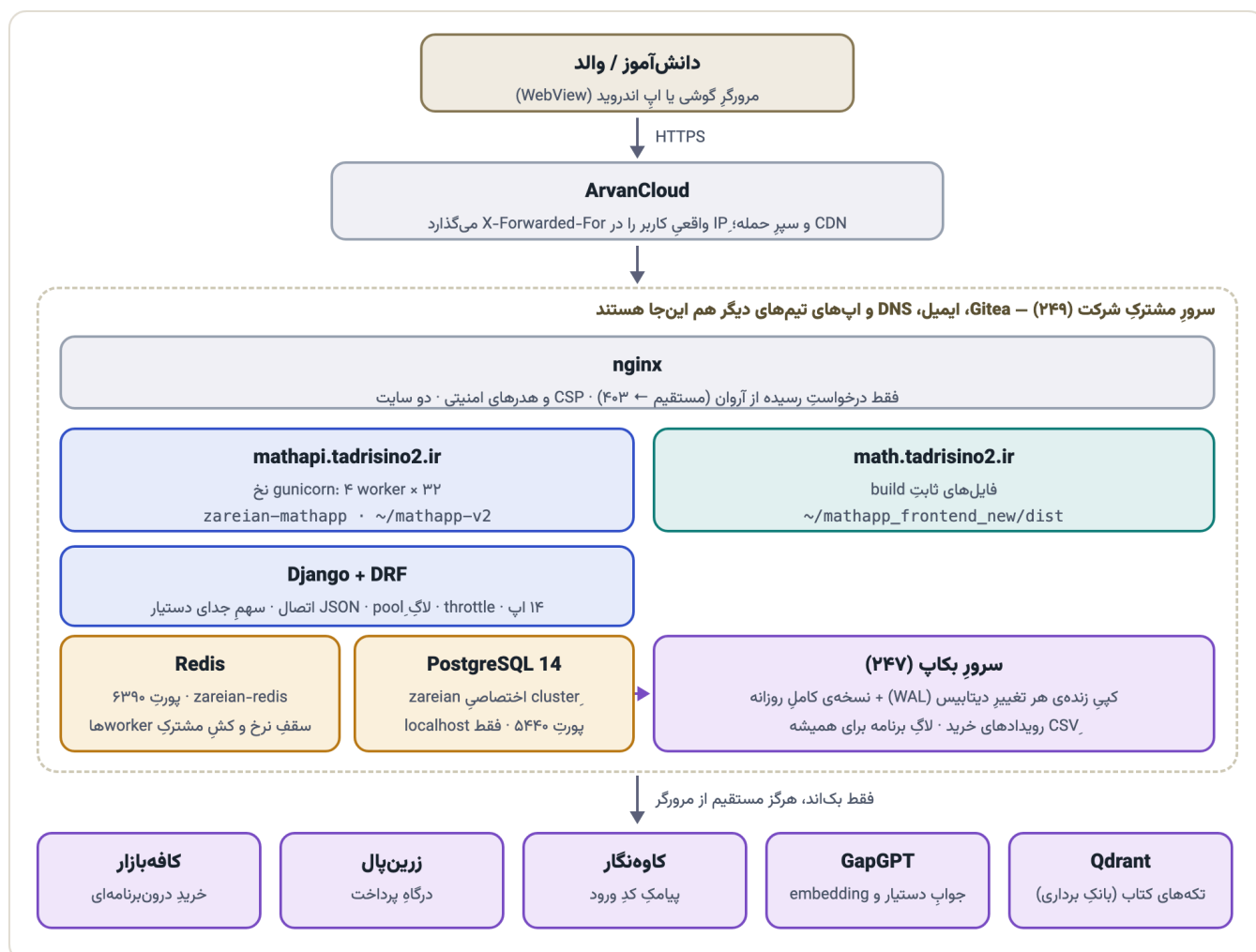
| لایه           | انتخاب                                      | چرا                                                                                                                                                                                                                                                                                                                                                              | چرا نه جایگزین                                                                                    | هزینه / ریسک                                                                            |
|----------------|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| جست‌وجوی کتاب  | Qdrant (بانک برداری)                        | تیم دیگر شرکت کتاب‌های سوم/ششم/هفتم را از قبل در یک Qdrant مشترک گذاشته بود؛ همان را ادامه دادیم تا داده دوباره ساخته نشود. فیلتر روی فیلد «فصل» دارد.                                                                                                                                                                                                           | pgvector به Postgres نیاز داشت؛ FAISS سرور ندارد و بین دو تیم اشتراک‌پذیر نیست.                   | سرویس مشترک با تیم دیگر؛ پاک‌کردن باید با احتیاط باشد (بخش ۹).                          |
| مدل زبانی      | GapGPT ← gpt-4.1-mini                       | GapGPT یک واسط سازگار با OpenAI است که از داخل ایران جواب می‌دهد. مدل با سنجش انتخاب شد، نه با جدول قیمت: روی ۱۰۲ مسئله‌ی تصحیح‌شده ۹۵-۹۶ درست در برابر ۸۵-۹۰ برای gpt-4o-mini؛ هر جواب با پرامپت امروز حدود ۲/۰ سنت (≈ ۶۰۰ تومان؛ docs/ai-cost-and-guardrails.md). دستیار فقط درباره‌ی ریاضی حرف می‌زند و سقف روزانه‌اش برای کاربر رایگان ۲ و برای مشترک ۴ است. | OpenAI مستقیم از ایران در دسترس نیست. مدل‌های گران‌تر در دور اول سنجش (۱۴ مدل) مزیتی نشان ندادند. | گنبد سرویس (حتی «سلام» ۷ ثانیه) دست ما نیست؛ اعتبار دو بار تمام شده.                    |
| ورود           | توکن امضا شده‌ی خودمان + کد پیامکی          | کاربرها با شماره وارد می‌شوند، نه نام کاربری/رمز؛ مدل کاربر جنگو به این نمی‌خورد. توکن با <code>django.core.signing</code> امضا می‌شود؛ یک وابستگی کمتر از کتابخانه‌ی JWT.                                                                                                                                                                                       | JWT همان کار را می‌کند با یک کتابخانه‌ی اضافه.                                                    | امنیتش به کلید امضا بسته است؛ کلید روی خود سرور ساخته شده و در هیچ مخزنی نیست (بخش ۱۲). |
| پیامک / پرداخت | کاوه‌نگار · زرین‌پال · کافه‌بازار           | سرویس‌های ایرانی رایج؛ کاوه‌نگار با «ارسال با الگو» که روی خط شرکت فعال است؛ زرین‌پال برای وب، بازار برای خرید درون‌برنامه‌ای اندروید.                                                                                                                                                                                                                           | —                                                                                                 | پرداخت واقعی تا انتها هنوز آزموده نشده.                                                 |
| اجرا           | gunicorn (gthread) پشت nginx پشت ArvanCloud | الگوی استاندارد اجرای جنگو. آروان CDN و سپر حمله‌ی ایرانی است و nginx فقط درخواست رسیده از آروان را می‌پذیرد. WhiteNoise فایل‌های استاتیک ادمین را از خود جنگو سرو می‌کند.                                                                                                                                                                                       | —                                                                                                 | سرور مشترک است؛ مال ما فقط پوشه‌ها و سرویس‌های خودمان (بخش ۱۳).                         |

### اصل کلی انتخاب‌ها

هر جا ممکن بود، ابزار آماده و رایج انتخاب شده است؛ هر جا انتخاب بر کیفیت اثر مستقیم داشت (مانند مدل هوش مصنوعی)، گزینه‌ها پیش از انتخاب با داده‌ی واقعی سنجیده شده‌اند.

### ۳. معماری کلی

دو برنامه‌ی جدا که فقط از راه HTTP با هم حرف می‌زنند: فرانت (فایل‌های ثابت در مرورگر) و بک‌اند (API). هر دو روی یک سرور مشترک شرکت‌اند.



شکل ۱-۳ - نقشه‌ی استقرار. همه‌ی کلیدهای سرویس‌های بیرونی فقط در `.env` سرورند و هیچ‌وقت به مرورگر نمی‌رسند.

#### درون بک‌اند: «یک برنامه، چند اپ مستقل» (Modular Monolith)

بک‌اند یک برنامه‌ی جنگوست که به ۱۴ «اپ» تقسیم شده. هر اپ یک پوشه است با مدل‌ها، API‌ها و تست‌های خودش. این همان الگویی است که اجازه می‌دهد چند نفر هم‌زمان روی بک‌اند کار کنند (بخش ۱۴).

| اپ       | مسئولیت                                                                             | مدل‌ها (جدول‌ها)                                                                                                                                                        |
|----------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| accounts | هویت، ورود با کد پیامکی، نشست ورود، پروفایل                                         | User Profile Session OTPCode LoginEvent                                                                                                                                 |
| content  | کتاب‌ها، برنامه‌ی درسی، سؤال، آزمون جزیره، ثبت جواب و پیشرفت، رأی و گزارش دانش‌آموز | Grade Subject Unit Node Lesson Question Choice UserAnswer<br>UserLessonCompletion UserNodeCompletion NodeProgress<br>ContentVote QuestionReport LessonReport SeedRecord |

| اپ            | مسئولیت                                                                    | مدل‌ها (جدول‌ها)                                                          |
|---------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------|
| progress      | امتیاز، سکه، قلب، زنجیره، جدول رتبه، همراه‌ها                              | UserStats UserHearts StreakRecord ActiveDay<br>LeaderboardEntry Character |
| ai_tutor      | گفتگو با دستیار، سهمیه، بازخورد 👍/👍                                        | Conversation Message MessageFeedback AskIdempotency                       |
| rag           | خواندن کتاب، Qdrant، ساخت پرامپت، شکل‌ها                                   | LessonChunk LessonChunkImage IngestedPage                                 |
| subscription  | پلن‌های اشتراک و اشتراک هر کاربر                                           | Plan Subscription                                                         |
| payment       | سفارش، تراکنش، زرین‌پال، بازار، لغو و بازپرداخت از ادمین، دفتر رویداد خرید | Order Transaction PaymentEvent BazaarCredential                           |
| discount      | کد تخفیف و پیشنهادهای                                                      | DiscountCode Redemption Offer UserOffer                                   |
| affiliate     | معرفی به دوستان و پورسانت                                                  | AffiliateAccount Conversion                                               |
| notifications | اعلان‌های داخل اپ                                                          | Notification                                                              |
| support       | تیکت و گزارش باگ                                                           | Ticket TicketMessage BugReport                                            |
| campaigns     | پیام گروهی (ذخیره و هدف‌گیری؛ زمان‌بندی خودکار ندارد)                      | MessageCampaign CampaignDelivery                                          |
| statuslog     | ثبت عمومی «وضعیت فلان چیز از X به Y رفت»                                   | StatusChangeLog                                                           |
| config        | تنظیمات، مسیرهای اصلی، middleware، سلامت، مدیریت خطا                       | —                                                                         |

## دو قاعده‌ی معماری که از کد پیدا نیست

### ۱. بین اپ‌ها کلید خارجی (FK) نیست

اپ progress کاربر را با یک عدد ساده نگه می‌دارد، نه با اتصال دیتابیزی به accounts :

```
# progress/models.py
class UserStats(models.Model):
    user_id = models.IntegerField(unique=True) # ارجاع سبک
    xp = models.PositiveIntegerField(default=0)
    coins = models.PositiveIntegerField(default=0)
```

چرا: اگر روزی «پرداخت» یا «دستیار» سرویس جدا شد، زنجیره‌ای از وابستگی پاره نمی‌شود. داخل یک اپ FK آزاد است (مثلاً Question → Lesson).

## ۲. اپ‌ها با «رویداد» خبر می‌دهند

وقتی پرداخت تأیید شد، `payment` خودش اشتراک و اعلان را نمی‌سازد؛ رویداد می‌فرستد یا تابع سرویسِ اپِ دیگر را صدا می‌زند، و هر اپ کار خودش را می‌کند. فایل‌های مربوط:

`*/signals.py` رویدادهایی که یک اپ می‌فرستد · `*/receivers.py` شنونده‌ها · `*/services.py` توابعی که اپ دیگر می‌تواند صدا بزند (مثلاً `notifications.services.notify`، `affiliate.services.record_paid_order`).

## ۴. ساختار فایل‌ها و نقش هر فایل

این بخش نقشه‌ی راه است: هر فایل چه چیزی دارد و به کدام فایل وصل است.

### ۴-۱ مخزن بک‌اند ( Tadrino/MathApp )

```
MathApp/
├── config/
│   ├── settings.py
│   ├── urls.py
│   ├── middleware.py
│   ├── exceptions.py
│   ├── health.py
│   ├── db.py
│   └── cache.py
├── renderers.py
├── log_formatters.py
├── accounts/
├── content/
├── progress/
├── ai_tutor/
├── rag/
├── subscription/
├── payment/
├── discount/
├── affiliate/
├── notifications/
├── support/
├── campaigns/
├── statuslog/
├── content/seed/
├── data/textbooks/
├── templates/tadrino/
├── tools/
├── ops/
├── docs/
├── authoring. ...
├── deploy.sh
├── gunicorn.conf.py
├── requirements.txt
├── ...
├── docker-compose.yml
└── manage.py
```

تنظیمات و سیم‌کشی کلی پروژه

AI ها؛ کلید امضا؛ مدل‌های throttle: نرخ env. همه‌ی تنظیمات؛ خواندن هر اپ include و /api/health، /api/docs، /admin: ریشه‌ی مسیرها سقف ورود ادمین، no-store، هر درخواست JSON واقعی فقط از آروان، لای IP یکدست، بی‌جزئیات داخلی JSON هر خطا → پاسخ

GET /api/health: بکاپ، هشدارها، pool، کش، migration، دیتابیس

pool اتصال Postgres (به‌جای ۵۰۰ ۵۰۳)، Redis

بیا اعراب و بی «ی» اضافه JSON خروجی

لای JSON قالب

git کل محتوا به‌صورت فایل در - curriculum.json و questions.json

README نیست؛ فقط git در) کتاب‌ها PDF

Swagger و ReDoc پوسته‌ی راست‌چپ

fake\_openai.py (آزمون بار) · loadtest/ (جایگزین GapGPT)

و آرشیو لاگ روی سرورها Postgres بکاپ، nginx نسخه‌ی اصلی

operations، architecture، security-before-production، content-

کل انتشار در یک اسکریپت (بخش ۱۳)

۱۲۰ timeout، ۳۲ worker × ۴

Django 5.2، DRF، drf-spectacular، openai، qdrant-client، pymupdf.

برای دیدن لای Loki + Alloy + Grafana: فقط محلی

### ۴-۲ آناتومی یک اپ (همه‌ی اپ‌ها همین شکل‌اند)

مثال: اپ content. ترتیب جدول همان ترتیبی است که یک درخواست از آن رد می‌شود.

| فایل           | چه چیزی در آن است                                                                            | نمونه‌ی واقعی                                              |
|----------------|----------------------------------------------------------------------------------------------|------------------------------------------------------------|
| urls.py        | نگاشت آدرس به کلاس ویو                                                                       | → questions/<int:question_id>/answer<br>QuestionAnswerView |
| views.py       | منطق هر اندپوینت: چه کسی مجاز است (permission_classes)، چند بار (throttle_scope)، و کار اصلی | ، NodeQuizView، QuestionAnswerView<br>NodeQuizCompleteView |
| serializers.py | شکل ورودی و خروجی؛ اعتبارسنجی؛ همین فایل Swagger را می‌سازد                                  | AnswerRequestSerializer (فقط)<br>(selectedOptionId)        |
| services.py    | منطق مشترکی که چند ویو یا اپ دیگر لازم دارند                                                 | درصد پیشرفت، جزیره‌ی بعدی                                  |
| models.py      | جدول‌های دیتابیس و قیدهای یکتایی                                                             | UserAnswer با قید یکتا روی کلید idempotency                |

| فایل                 | چه چیزی در آن است                                          | نمونه‌ی واقعی                                       |
|----------------------|------------------------------------------------------------|-----------------------------------------------------|
| migrations/          | تاریخچه‌ی تغییر جدول‌ها؛ migrate روی سرور اعمال‌شان می‌کند | id: 0008 هر پایه = شماره‌ی پایه                     |
| admin.py             | صفحه‌های پنل ادمین، اکشن‌ها، واردکردن فله‌ای               | «وارد کردن سؤال‌ها از فایل»                         |
| receivers.py         | واکنش به رویداد اپ‌های دیگر                                | باطل کردن کش نقشه وقتی سؤال عوض شد                  |
| management/commands/ | دستورهای خط فرمان (python manage.py ...)                   | seed_content، generate_questions_from_chunks        |
| tests*.py            | تست‌ها؛ هر باگ پیدا شده یک تست دارد                        | tests_frontend_contract.py – نهبان قرارداد با فرانت |

### ۳-۴ مخزن فرانت ( Tadrino/MathApp\_frontend )

|                |                                                                  |
|----------------|------------------------------------------------------------------|
| src/           |                                                                  |
| └─ main.tsx    | را سوار می‌کند App نقطه‌ی شروع؛                                  |
| └─ config.ts   | نشانی API: dist/config.js ← VITE_API_BASE_URL ← /api             |
| └─ api/        | نمی‌زند fetch تنها در ارتباط با سرور – هیچ جای دیگر              |
| │ └─ http.ts   | SSE، Idempotency-Key، تکنس‌های refresh، GET تکرار، timeout، توکن |
| │ └─ index.ts  | api.nodeQuiz، api.lesson.answer، ... یک تابع برای هر اندپوینت    |
| │ └─ types.ts  | شکل پاسخ‌ها، از روی پاسخ واقعی سرور                              |
| └─ app/        | App.tsx: تب‌ها و صفحه‌ها، MainApp.tsx: جریان ورود                |
| └─ nav/        | گوشی back ناوبری پشته‌ای خودمان، هماهنگ با دکمه‌ی                |
| └─ state/      | داده‌ی مشترک (پروفایل، نقشه) و گفتگوی دستیار                     |
| └─ screens/    | HomeScreen، QuizScreen، AskScreen، ReportScreen، ...             |
| └─ components/ | اجزای مشترک: مسیر جزیره‌ها، برگه‌ی پایین، نوار پایین             |
| └─ mascots/    | از کد، ۶ حالت چهره SVG – سیگما، دلتا، پی، امگا                   |
| └─ diagram/    | مختصات، ریشه، زاویه اینجا حساب می‌شود SVG → JSON شکل دستیار      |
| └─ lib/        | پیام خطای دوستانه، صدا، MathText، رقم فارسی، تاریخ شمسی          |

قاعده‌ی فرانت: صفحه‌ها هرگز مستقیم سرور را صدا نمی‌زنند؛ همیشه `screens/* → api/index.ts → api/http.ts`

## ۵. سفر یک درخواست، فایل به فایل

نمونه: ثبت پاسخ یک دانش آموز به یک سؤال در آزمون جزیره. در ادامه مسیر کامل این درخواست از رابط کاربری تا دیتابیس و بازگشت پاسخ آمده است. سایر درخواست‌ها نیز همین مسیر را طی می‌کنند و تنها ویو پاسخ‌دهنده تفاوت دارد.

1. فرانت · صفحه — `src/screens/QuizScreen.tsx` — کاربر گزینه‌ای را انتخاب می‌کند و صفحه تابع API مربوط را فراخوانی می‌کند. صفحه نمی‌داند نشانی سرور چیست.

```
const r = await api.lesson.answer(question.id, optionId, answerKey);
```

2. فرانت · api — `src/api/index.ts` — نام تابع را به نشانی API ترجمه می‌کند. برای هر جواب یک کلید یکتا (Idempotency-Key) ساخته می‌شود تا اگر اینترنت قطع شد و دوباره فرستاده شد، دوبار حساب نشود.

```
answer: (questionId, selectedOptionId, idempotencyKey = newIdempotencyKey()) =>
  http.create(`/questions/${questionId}/answer`, { selectedOptionId }, { idempotencyKey })
```

3. فرانت · http — `src/api/http.ts` — توکن را از `localStorage` می‌خواند، هدرها را تنظیم می‌کند و درخواست را با محدودیت زمانی (timeout) می‌فرستد. اگر ۴۰۱ برگشت، یک بار توکن را تازه می‌کند و دوباره می‌فرستد.

```
POST https://mathapi.tadrisino2.ir/api/questions/1834/answer
Authorization: Bearer 10:1tQx9k:Xy3... ← user_id: امضا
Idempotency-Key: 7f3c... {"selectedOptionId": 7290}
```

4. شبکه — اینترنت ← `nginx` ← `ArvanCloud` روی سرور (فقط درخواست رسیده از آروان) ← سوکت یونیکس ← یک نخ آزاد از `gunicorn` (`gunicorn.conf.py`).

5. بک‌اند · `middleware` — `config/settings.py` → `MIDDLEWARE` — لایه‌های عمومی به ترتیب: امنیت، `CORS` (آیا درخواست از `math.tadrisino2.ir` آمده؟)، فایل‌های استاتیک، سقف ورود ادمین، ...، `APINoStoreMiddleware` (پاسخ در کش آروان نماند)، و در آخر `APIRequestLoggingMiddleware` که در پایان یک خط JSON لاگ می‌نویسد (`config/middleware.py`).

6. بک‌اند · مسیر — `config/urls.py` همه‌ی `/api/` را به اپ‌ها می‌سپارد؛ `content/urls.py` این آدرس را پیدا می‌کند:

```
path('api/', include('content.urls')), # config/urls.py
path('questions/<int:question_id>/answer', QuestionAnswerView.as_view()), # content/urls.py
```

7. بک‌اند · هویت — `accounts/authentication.py` — امضا و انقضای توکن را بی‌رجوع به دیتابیس می‌سنجد و `request.user` را می‌سازد. بی‌هدر یعنی «درخواست ناشناس»، نه خطا؛ ویو تصمیم می‌گیرد.

```
value = TimestampSigner(salt='tadrisino-access-token').unsign(token, max_age=3600)
if user.tokens_valid_after and issued_at(token) < user.tokens_valid_after: raise
AuthenticationFailed
```

8. بک‌اند · مجوز و سقف — `content/views.py` → `QuestionAnswerView` — فقط کاربر وارد شده، حداکثر ۶۰ جواب در دقیقه (نرخ در `settings.DEFAULT_THROTTLE_RATES`).

```
permission_classes = [IsAuthenticated]
throttle_scope = 'question_answer' # 'question_answer': '60/min'
```

9. بک‌اند · اعتبارسنجی – `content/serializers.py` – ورودی باید دقیقاً یک عدد باشد؛ هر چیز دیگر ۴۰۰ می‌گیرد.  
`userId` هرگز از کلاینت خوانده نمی‌شود، فقط از توکن.

```
class AnswerRequestSerializer(serializers.Serializer):
    selectedOptionId = serializers.IntegerField()
```

10. بک‌اند · دیتابیس – ثبت جواب و کم کردن قلب «همه یا هیچ» در یک تراکنش ( `content/models.py` )  
 : ( `progress/models.py` )

```
with transaction.atomic():
    UserAnswer.objects.create(user_id=user_id, question=question, choice=choice,
                              correct=choice.is_correct, idempotency_key=idempotency_key)
    if not choice.is_correct:
        hearts = UserHearts.objects.select_for_update().get(user_id=user_id)
        hearts.apply_regen(); hearts.lose_heart(); hearts.save()
```

قید یکتای ( `user_id, question, idempotency_key` ) در دیتابیس آخرین خط دفاع است: درخواست تکراری نمی‌تواند ردیف دوم بسازد. بعد از روز فعال زنجیره ثبت می‌شود ( `progress/services.py` ).

11. بک‌اند · پاسخ

```
{"correct": false, "correctOptionId": "7288", "xpEarned": 0,
 "heartsRemaining": 4, "mascotMood": "encourage"}
```

اگر هر جا خطا شد، `config/exceptions.py` آن را به JSON یکدست تبدیل می‌کند و traceback فقط در لاگ می‌ماند.

12. فرانت · نمایش – `QuizScreen.tsx` گزینه‌ی نادرست را قرمز و گزینه‌ی درست را سبز نمایش می‌دهد، قلب را کم نشان می‌دهد، چهره‌ی همراه را عوض می‌کند و دکمه‌ی «از پی بپرس چرا غلط شد» را نشان می‌دهد. پیام خطا همیشه از `lib/errors.ts` می‌آید، نه متن خام سرور.

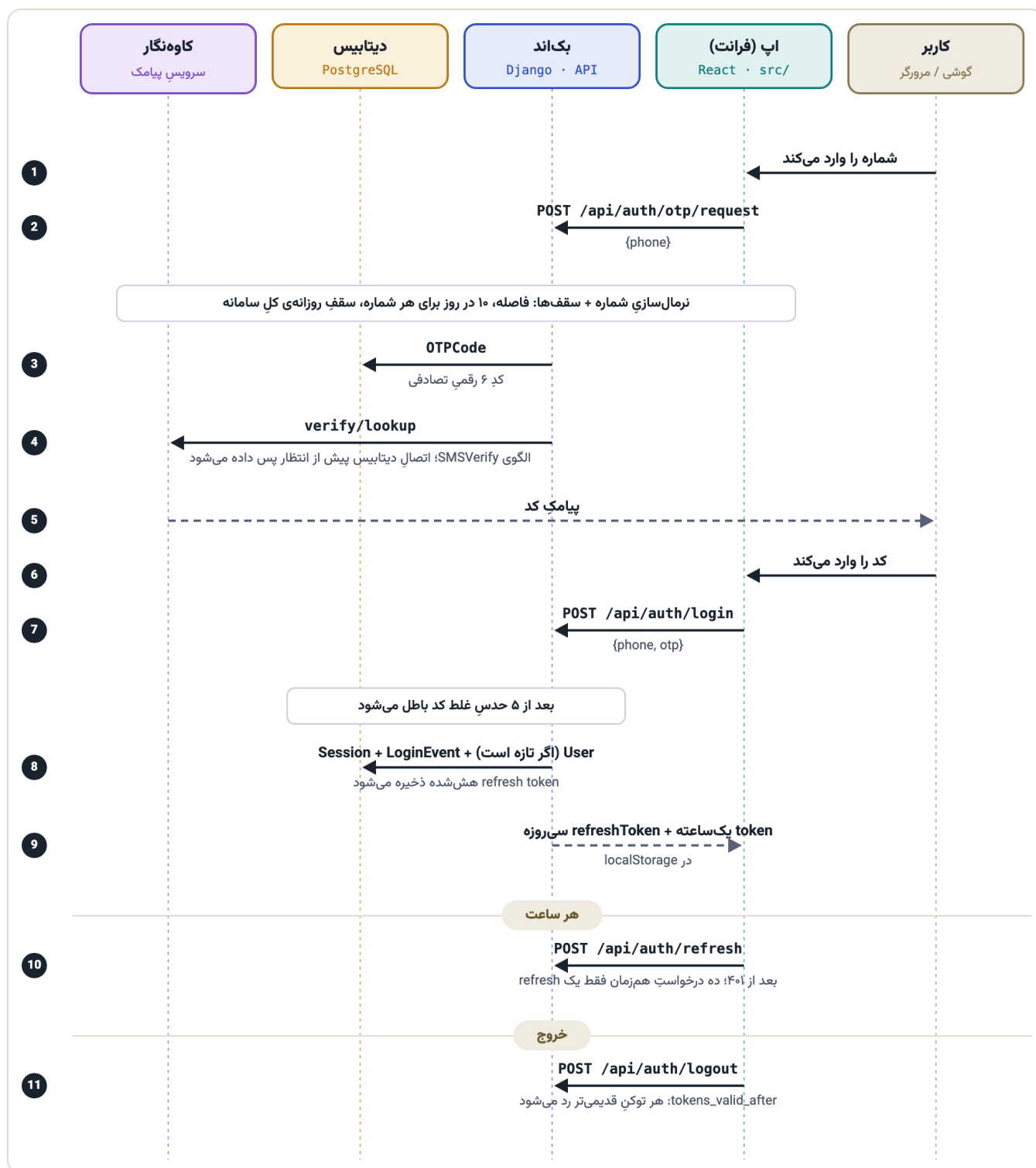
### خلاصه‌ی مسیر درخواست

صفحه تابع API را صدا می‌زند؛ `http.ts` توکن را می‌گذارد و می‌فرستد. از آروان و nginx به gunicorn می‌رسد. جنگو از `urls.py` ویو را پیدا می‌کند، توکن در `authentication.py` سنجیده می‌شود، ویو مجوز و سقف را چک می‌کند، serializer ورودی را، و منطق در یک تراکنش اتمیک در دیتابیس می‌نویسد. پاسخ JSON است و برای هر درخواست یک خط لاگ JSON ثبت می‌شود.

## ۶. سناریوهای اصلی

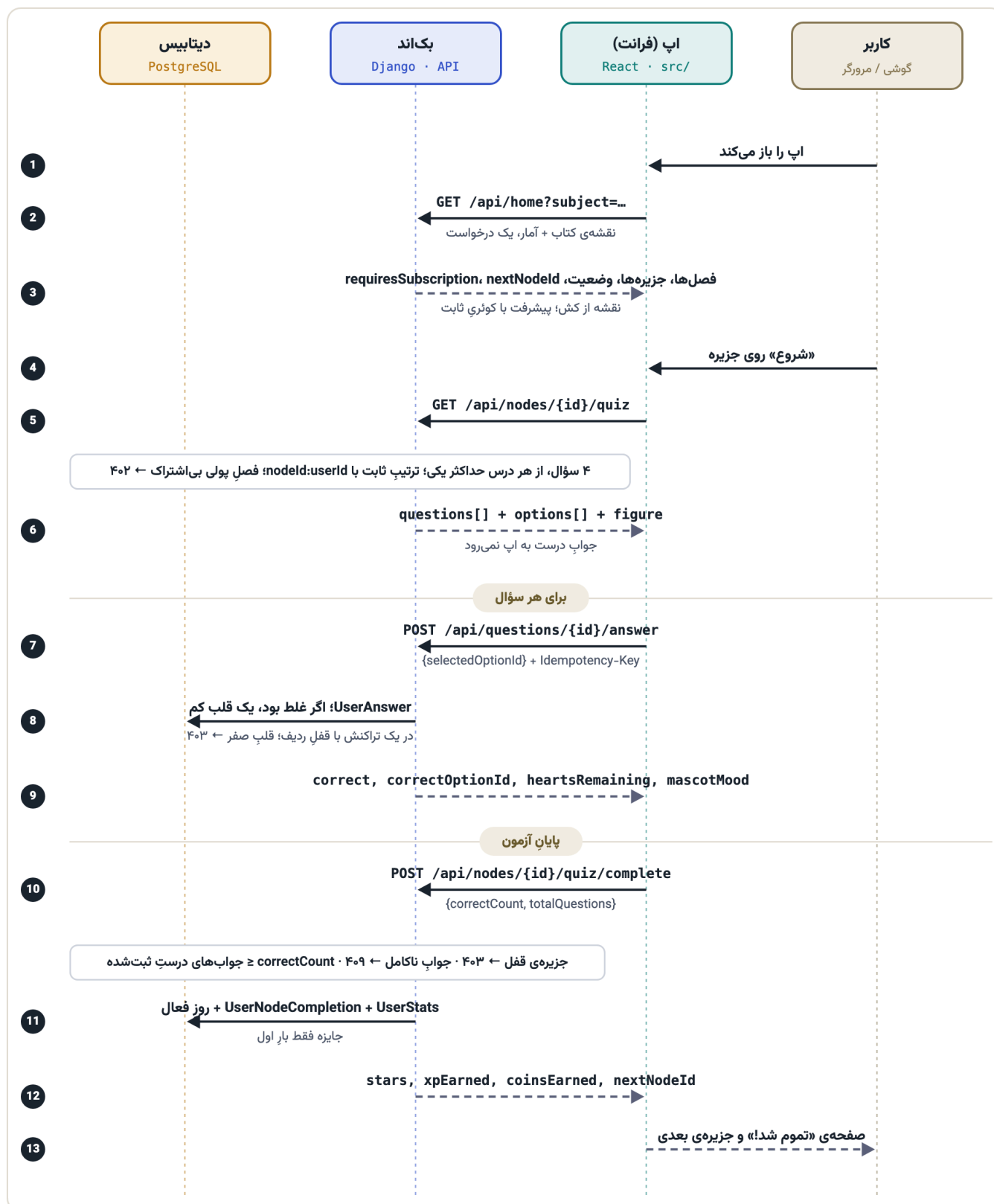
هر نمودار از راست به چپ خوانده می‌شود: ستون‌ها بازیگرها هستند، پیکان‌ها درخواست‌ها. پیکان خط‌چین یعنی پاسخ. شماره‌ها ترتیب زمانی‌اند.

### ۶-۱ ورود با شماره و کد پیامکی



کد: accounts/views.py · accounts/phone.py (نرمال‌سازی شماره) · accounts/services.py (ارسال پیامک) · accounts/authentication.py (توکن) · تست‌ها: accounts/tests\_otp\_limits.py

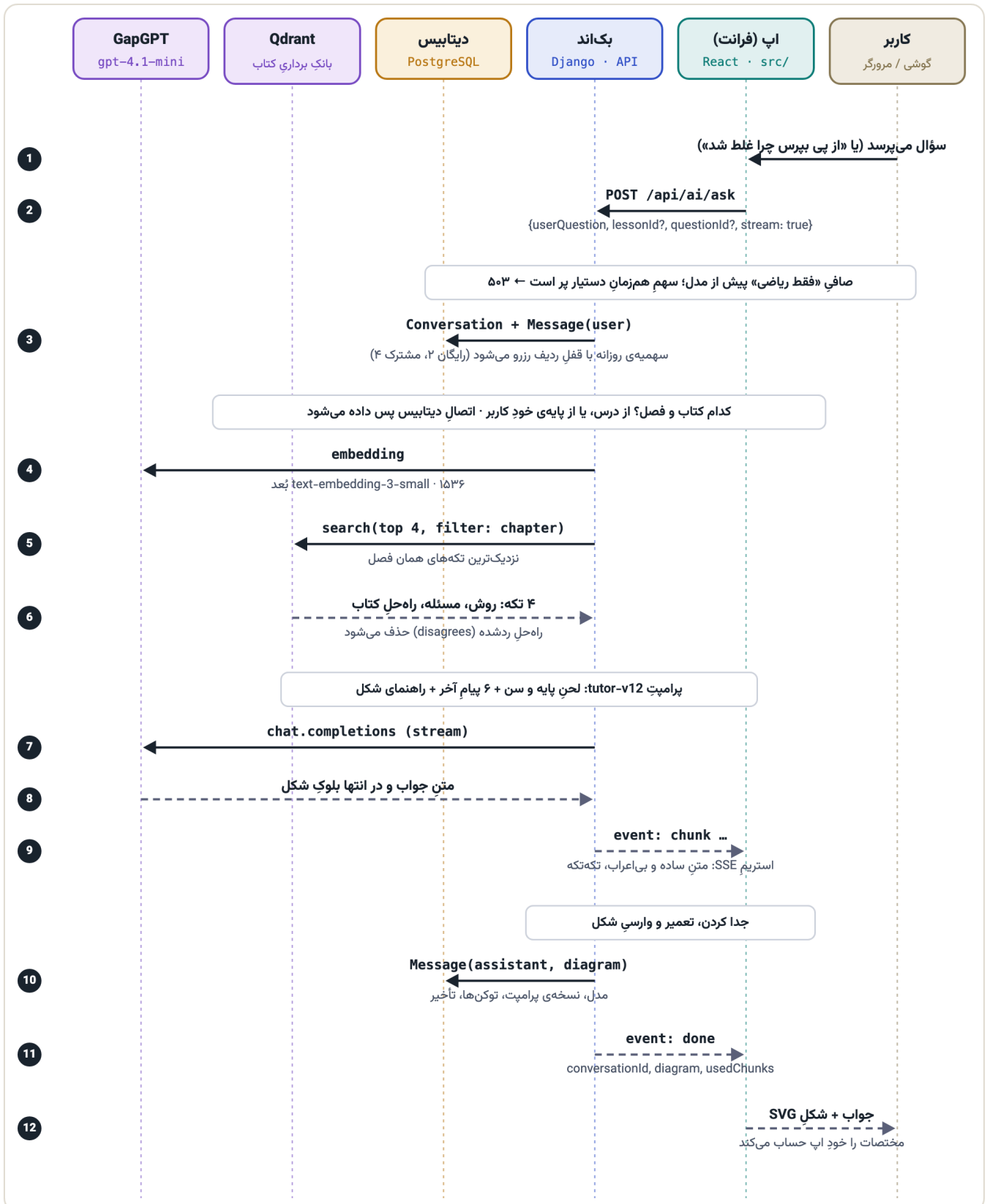
## ۶-۲ بازی یک جزیره



کد: content/views.py ( NodeQuizCompleteView , QuestionAnswerView , NodeQuizView , HomeView ) · فرانت:

screens/QuizScreen.tsx , screens/HomeScreen.tsx

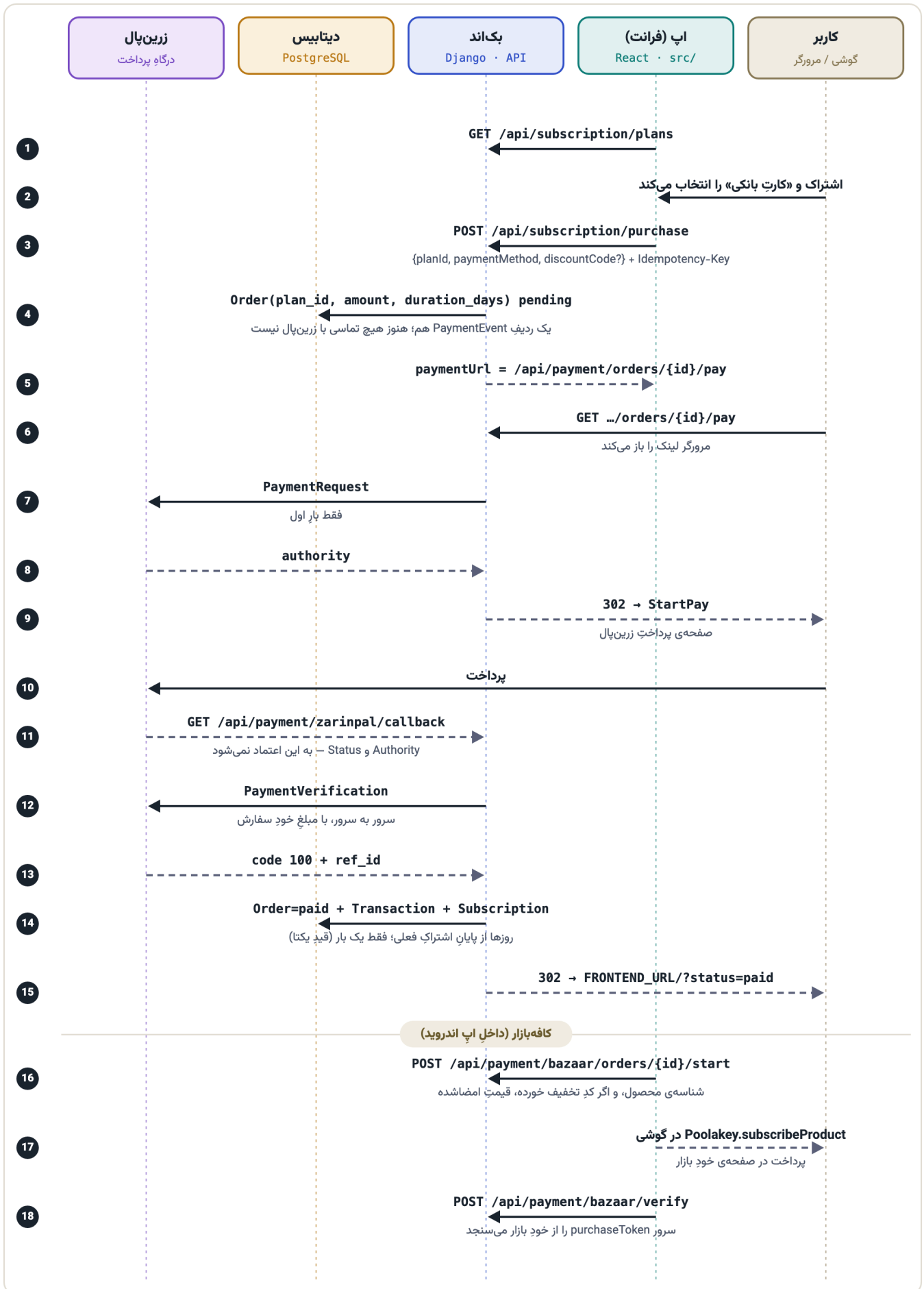
### ۳-۶ «از پی پیرس» – دستیار با کتاب



کد: `ai_tutor/views.py` → `AskAIView` · `rag/services/lesson_context.py` · `rag/services/retrieval.py` · فرانت: `rag/services/qdrant_store.py` · `rag/services/generation.py` · `rag/services/diagram.py` · `diagram/` · `screens/AskScreen.tsx`

سه نکته‌ی طراحی: (۱) تماس با GapGPT بیرون از تراکنش دیتابیس است، وگرنه ۳۰ ثانیه انتظار دیتابیس را قفل می‌کرد؛ (۲) اگر GapGPT خطا داد، سهمیه‌ی رزرو شده پس داده می‌شود و کاربر ۵۰۳ می‌گیرد، نه ۵۰۰؛ (۳) اولین پیام هر گفتگو کش می‌شود – سؤال تکراری از کش ۰/۰۱ ثانیه‌ای برمی‌گردد در برابر ۱۱~ ثانیه.

## ۴-۶ خرید اشتراک (زرین پال یا کافه بازار)



کد: `subscription/services.py` → `payment/services.py` · `payment/providers.py` · `subscription/views.py`  
`activate_subscription` ( `payment/views.py` ، `PaymentRedirectView` ، `ZarinpalCallbackView` )  
 تست‌ها: `payment/bazaar.py` · `payment/zarinpal.py` · ( `BazaarVerifyPurchaseView` ، `BazaarStartPurchaseView` )  
`payment/tests_refund.py` · `payment/tests_bazaar_subscription.py` · `payment/tests_audit_fixes.py`

فقط اشتراک فروخته می‌شود: یک / سه / شش ماهه به ۵۹۸ / ۸۹۸ / ۱,۳۹۸ هزار تومان، هر سه با تمامی فصل‌ها، قلب نامحدود و ۴ سوال در روز از دستیار. کاربر در برگه‌ی پرداخت بین «کارت بانکی» (زرین‌پال، پیش‌فرض) و «کافه‌بازار» (فقط داخل اپ اندروید) انتخاب می‌کند. پرداخت تأییدشده روزهای همان پلن را از پایان اشتراک فعلی اضافه می‌کند. تفاوت دو راه پرداخت در `payment/providers.py` است (هر راه یک کلاس).

### چرا این قدر پیچیده؟

- ساخت سفارش به زرین‌پال وابسته نیست: اگر زرین‌پال قطع باشد، سفارش ساخته می‌شود و فقط رفتن به درگاه عقب می‌افتد.
  - برگشت کاربر هیچ چیزی را ثابت نمی‌کند: هر کس می‌تواند آدرس برگشت را با `Status=OK` بسازد. فقط تأیید سرور - به - سرور سفارش را «پرداخت‌شده» می‌کند.
  - پرداخت جاافتاده بازسنجی می‌شود: اگر زرین‌پال موقع برگشت جواب نداد، هر بار که اپ وضعیت خرید را می‌خواند سفارش دوباره سنجیده می‌شود (و دستور `payment_reconcile`).
  - پلن و مدت روی خود سفارش است؛ کد تخفیف ۱۰۰٪ بی‌درگاه تمام می‌کند؛ تأیید تکراری روز دوباره نمی‌دهد (قید یکتا روی تراکنش).
  - کافه‌بازار: اپ با Poolakey اشتراک می‌خرد و سرور توکن را از خود بازار می‌سند؛ بازار تمدید خودکار را خبر نمی‌دهد، پس سرور نزدیک پایان دوره خودش می‌پرسد (`sync_bazaar_renewals`). کد تخفیف با «تخفیف پویا»ی بازار به قیمت می‌رسد.
  - لغو و بازپرداخت از پنل ادمین: دو action روی سفارش‌ها؛ بازپرداخت پس از صفحه‌ی تأیید روزهای اشتراک همان سفارش را پس می‌گیرد. خود پول از پنل درگاه برگردانده می‌شود (`docs/operations.md`).
- آزموده‌نشده: زرین‌پال درخواست پرداخت را با کد ۱۰۰ می‌پذیرد، ولی یک پرداخت واقعی تا انتها هنوز انجام نشده؛ خرید بازار منتظر نسخه‌ی تازه‌ی اپ اندروید است.

## ۵-۶ گزارش والدین

والد حساب جدا ندارد؛ صفحه‌ی «گزارش والدین» در پروفایل خود دانش‌آموز است.

| # | درخواست                                  | چه می‌شود                                                                                                                                               |
|---|------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| ۱ | <code>GET /api/user/weekly-report</code> | برای هر یک از ۷ روز اخیر و ۷ روز قبلش: زمان تمرین، سوال حل‌شده و درست، امتیاز، درس و جزیره‌ی تمام‌شده، سوال از دستیار. فقط دیتای خود صاحب توکن.         |
| ۲ | —                                        | فرانت جمله‌ی مقایسه («آفرین! این هفته ۲۰ دقیقه بیشتر...») و دو نمودار می‌سازد: زمان هر روز در برابر همان روز هفته‌ی پیش، و سوال‌های درست/نادرست هر روز. |
| ۳ | <code>GET /api/progress/nodes</code>     | هر کاربر فقط پیشرفت خودش را می‌بیند؛ شناسه‌ی کاربر دیگر پذیرفته نمی‌شود.                                                                                |

**تخمینی:** «زمانِ تمرین» اندازه‌گیری نیست: از فاصله‌ی جواب‌ها و پرسش‌ها حساب می‌شود (فاصله‌ی بیش از ۱۰ دقیقه وقفه است). خواندنِ درس بدونِ جواب دادن در آن نیست، پس عدد معمولاً از زمانِ واقعی کمتر است.

## ۷. مدل داده

جدول‌های اصلی، دسته‌بندی‌شده بر اساس اپ. → یعنی کلید خارجی واقعی (داخل همان اپ)، و `user_id` یعنی ارجاع سبک به اپ دیگر.

### accounts هویت

- `User phone, name, grade_id, tokens_valid_after`
- `Profile → User · birth_date` (شمسی در API)، `birth_year`
- `Session → User · refresh_token` (هش‌شده)، `expires_at, revoked`
- `OTPCode phone, code, expires_at, failed_attempts`
- `LoginEvent` ثبت هر ورود

### content محتوا

- `Grade id = (۸ = هشتم)` شماره‌ی پایه
- `Subject key, title` (ریاضی، حسابان، هندسه، آمار و احتمال، گسسته)
- `Unit → Grade, → Subject · title, blurb, order`
- `Node → Unit · order` (جزیره)
- `Lesson → Node · kind, example, steps, figures`
- `Question → Lesson · prompt, type, figure`
- `Choice → Question · value, is_correct`
- `SeedRecord` تحویل داده (تا کار ادمین برگردانده نشود) `seed` دفتر آنچه

### content فعالیت کاربر

- `UserAnswer user_id, → Question, → Choice`
- `UserLessonCompletion user_id, → Lesson`
- `UserNodeCompletion user_id, → Node, stars, xp_earned, coins_earned`
- `NodeProgress user_id, → Node, status`
- `ContentVote user_id, ۱ - / ۱` سؤال یا درس،
- `QuestionReport · LessonReport` «ایراد دارد»

### progress بازی

- `UserStats user_id, xp, coins, selected_character_id`
- `UserHearts current_hearts, max_hearts=5, next_refill_at, full_refill_at`
- `StreakRecord current_streak, longest_streak`
- `ActiveDay → StreakRecord, date`
- `LeaderboardEntry period, xp_in_period, rank`
- `Character slug, name, unlock_xp`

### ai\_tutor · rag دستیار

- Conversation user\_id, lesson\_id, grade\_id
- Message → Conversation · role, text, diagram, model\_name, prompt\_version, latency\_ms
- MessageFeedback → Message · is\_helpful
- LessonChunk problem, solution, solution\_verdict, qdrant\_point\_id
- IngestedPage collection, page

### subscription · payment پول

- Plan price, duration\_days, on\_sale, ai\_daily\_quota, bazaar\_sku
- Subscription user\_id, → Plan, status, expires\_at
- Order user\_id, plan\_id, subscription\_id, amount, list\_price, plan\_title, duration\_days, status (... canceled, refunded), provider, gateway\_authority, bazaar\_purchase\_token, bazaar\_valid\_until
- Transaction → Order · gateway\_ref\_id, status, paid\_at
- PaymentEvent order\_id, event, detail – هر قدم مسیر خرید
- BazaarCredential توکن اتصال به بازار

### discount · affiliate بازاریابی

- DiscountCode code, type, value, max\_uses
- Redemption یکتا: هر کاربر × هر کد
- Offer / UserOffer پیشنهاد خودکار
- AffiliateAccount user\_id, referral\_code, commission\_rate
- Conversion → AffiliateAccount · order\_id, commission\_amount

### support · notifications ارتباط

- Ticket user\_id, subject, status
- TicketMessage → Ticket · sender, text
- BugReport
- Notification user\_id, title, body, is\_read

### campaigns · statuslog

- MessageCampaign, CampaignDelivery یکتا: کمپین × کاربر × کانال
- StatusChangeLog label, object\_id, old\_status → new\_status

فهرست فیلدها خلاصه است؛ منبع کامل <app>/models.py است. نامها و ارتباطها از کد خوانده شده‌اند.

## قیدهایی که در خود دیتابیس گذاشته شده‌اند (نه فقط در کد)

| قید                                                  | از چه جلوگیری می‌کند                  |
|------------------------------------------------------|---------------------------------------|
| یکتا: UserAnswer(user_id, question, idempotency_key) | ثبت دوباره‌ی یک جواب با تکرار درخواست |

| از چه جلوگیری می‌کند                                                    | قید                                                                                |
|-------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| استفاده‌ی دوباره از یک کد تخفیف                                         | یکتا: (کاربر، کد) Redemption                                                       |
| روز دوباره وقتی تأیید پرداخت دو بار برسد، یا یک توکن خرید برای دو سفارش | یکتا: Transaction(gateway, gateway_ref_id)                                         |
| پیام تکراری وقتی کمپین دو بار همزمان اجرا شود                           | یکتا: (کمپین، کاربر، کانال) CampaignDelivery                                       |
| تکه‌ی تکراری در Qdrant وقتی خواندن کتاب از سر گرفته می‌شود              | یکتا: LessonChunk(collection, qdrant_point_id) و<br>IngestedPage(collection, page) |

**کد و محتوا دو مسیر جدا دارند.** دیتابیس در git نیست. محتوا (برنامه‌ی درسی و سؤال‌ها) به‌صورت JSON در `content/seed/` است و `deploy.sh` در هر انتشار آن را وارد می‌کند. داده‌ی کاربرها فقط در Postgres سرور (و کپی زنده‌اش روی سرور بکاپ) است.

## ۸. API: قواعد و فهرست کامل

مستند زنده و قابل امتحان: <https://mathapi.tadrisino2.ir/api/docs/> (Swagger). جدول‌های زیر از همان اسکیمای استخراج شده‌اند.

### قواعد مشترکِ همه‌ی اندپوینت‌ها

| موضوع      | قاعده                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| نشانی پایه | <code>https://mathapi.tadrisino2.ir/api/...</code> — دامنه در کد نیامده و از تنظیمات ( <code>PUBLIC_BASE_URL</code> ) خوانده می‌شود.                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| ورود       | هدر <code>Authorization: Bearer &lt;token&gt;</code> . توکن یک ساعت اعتبار دارد و با <code>POST /api/auth/refresh</code> تازه می‌شود.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| قالب داده  | JSON؛ نام فیلدها camelCase ( <code>selectedOptionId</code> )، شناسه‌ها در پاسخ رشته‌اند.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| خطا        | همیشه JSON با <code>detail</code> (و گاهی <code>code</code> مثل <code>no_hearts</code> ، <code>node_locked</code> ، <code>answers_missing</code> ، <code>subscription_required</code> ): <code>400</code> ورودی بد، <code>401</code> توکن نامعتبر، <code>402</code> اشتراک لازم است، <code>403</code> اجازه نداری، <code>404</code> ، <code>409</code> پیش‌شرط برقرار نیست، <code>429</code> زیاد درخواست دادی، <code>503</code> سرویس بیرونی یا ظرفیت در دسترس نیست (با <code>Retry-After</code> ).<br>متن فارسی پاسخ‌ها بی‌اعراب و بی «ی» اضافه است، و هیچ LaTeX یا markdown نمی‌دارد. |
| تکرار امن  | هدر اختیاری <code>Idempotency-Key</code> روی خرید، دستیار، جواب، پایان درس و جزیره: درخواست تکراری با همان کلید، همان نتیجه‌ی قبلی را برمی‌گرداند.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| سقف نرخ    | هر اندپوینت حساس یک <code>throttle_scope</code> دارد؛ نرخ‌ها همه یک‌جا در <code>config/settings.py</code> → <code>DEFAULT_THROTTLE_RATES</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| استریم     | <code>POST /api/ai/ask</code> با <code>"stream": true</code> پاسخ را SSE می‌فرستد: رویدادهای <code>chunk</code> ، <code>done</code> ، <code>error</code> .                                                                                                                                                                                                                                                                                                                                                                                                                               |

### فهرست اندپوینت‌ها (۶۱ عملیات)

عمومی بی‌ورود · ورود توکن لازم · داخلی با هدر کلید داخلی (سرور-به-سرور)

| روش                                    | مسیر                               | دسترسی | کار                                                       |
|----------------------------------------|------------------------------------|--------|-----------------------------------------------------------|
| هویت و پروفایل — <code>accounts</code> |                                    |        |                                                           |
| POST                                   | <code>/api/auth/otp/request</code> | عمومی  | فرستادن کد ورود                                           |
| POST                                   | <code>/api/auth/login</code>       | عمومی  | ورود با شماره و کد                                        |
| POST                                   | <code>/api/auth/refresh</code>     | عمومی  | توکن تازه با <code>refresh token</code>                   |
| POST                                   | <code>/api/auth/logout</code>      | ورود   | خروج؛ توکن‌های قبلی باطل                                  |
| GET · PATCH                            | <code>/api/user/profile</code>     | ورود   | خواندن و ویرایش پروفایل (تاریخ تولد شمسی، شناسه‌ی نمایشی) |

| روش                      | مسیر                                                   | دسترسی | کار                                                                    |
|--------------------------|--------------------------------------------------------|--------|------------------------------------------------------------------------|
| POST                     | /api/user/delete                                       | ورود   | حذف حساب با کد پیامکی تازه                                             |
| POST                     | /api/user/grade                                        | ورود   | ثبت پایه                                                               |
| محتوا و آزمون – content  |                                                        |        |                                                                        |
| GET                      | /api/grades                                            | عمومی  | فهرست پایه‌ها                                                          |
| GET                      | /api/home                                              | ورود   | نقشه‌ی جزیره‌ها + آمار، در یک درخواست؛ ?subject= کتاب را انتخاب می‌کند |
| GET                      | /api/nodes/{id}/lessons                                | عمومی  | درس‌های یک جزیره                                                       |
| GET                      | /api/nodes/{id}/quiz                                   | عمومی  | آزمون ۴ سؤالی جزیره                                                    |
| POST                     | /api/nodes/{id}/quiz/complete                          | ورود   | پایان جزیره؛ ستاره، امتیاز، سکه، جزیره‌ی بعد                           |
| GET                      | /api/lessons/{id}/intro                                | عمومی  | مثال و قدم‌های درس                                                     |
| GET                      | /api/lessons/{id}/questions                            | عمومی  | سؤال‌های یک درس                                                        |
| POST                     | /api/lessons/{id}/complete                             | ورود   | پایان درس                                                              |
| POST                     | /api/questions/{id}/answer                             | ورود   | ثبت جواب                                                               |
| POST                     | /api/questions/{id}/report<br>/api/lessons/{id}/report | ورود   | گزارش ایراد سؤال یا درس                                                |
| POST                     | /api/questions/{id}/vote<br>/api/lessons/{id}/vote     | ورود   | 👍 / 👎 روی سؤال یا درس (value ۱، -۱ یا ۰)                               |
| GET                      | /api/progress/nodes                                    | ورود   | وضعیت خود کاربر روی جزیره‌ها                                           |
| پیشرفت و بازی – progress |                                                        |        |                                                                        |
| GET                      | /api/user/summary                                      | ورود   | پروفایل + آمار + زنجیره در یک درخواست                                  |
| GET                      | /api/user/hearts                                       | ورود   | قلب‌ها و زمان پر شدن                                                   |
| GET                      | /api/user/streak/calendar                              | ورود   | تقویم روزهای فعال                                                      |
| GET                      | /api/leaderboard                                       | عمومی  | جدول رتبه: هفتگی یا همه‌ی زمان‌ها                                      |
| GET                      | /api/characters                                        | ورود   | همراه‌ها و باز/قفل بودنشان                                             |
| POST                     | /api/characters/select                                 | ورود   | انتخاب همراه (فقط بازنده)                                              |

| روش                        | مسیر                                  | دسترسی | کار                                                            |
|----------------------------|---------------------------------------|--------|----------------------------------------------------------------|
| دستیار –<br>ai_tutor و rag |                                       |        |                                                                |
| POST                       | /api/ai/ask                           | ورود   | پرسش؛ JSON یا استریم                                           |
| GET                        | /api/ai/quota                         | ورود   | سوال‌های باقی امروز                                            |
| GET                        | /api/ai/conversations/{id}/messages   | ورود   | تاریخچه‌ی گفتگو                                                |
| POST                       | /api/ai/messages/{id}/feedback        | ورود   | 👍 / 👎 روی یک جواب                                              |
| POST                       | /pdf · /api/rag/ingest/text           | داخلی  | افزودن متن یا PDF به Qdrant                                    |
| GET                        | /api/rag/chunks                       | داخلی  | مرور تکه‌های ذخیره‌شده                                         |
| اشتراک، پرداخت،<br>تخفیف   |                                       |        |                                                                |
| GET                        | /api/subscription/plans               | عمومی  | اشتراک‌های در حال فروش                                         |
| POST                       | /api/subscription/purchase            | ورود   | شروع خرید؛ سفارش و لینک پرداخت                                 |
| GET                        | /api/subscription/status              | ورود   | اشتراک فعال و پایانش (و بازسنجی پرداخت جافتاده و تمدید بازار)  |
| POST                       | /api/payment/bazaar/orders/{id}/start | ورود   | شروع خرید بازار (SKU و قیمت امضاشده)                           |
| GET                        | /api/payment/orders/{id}              | ورود   | وضعیت سفارش                                                    |
| GET                        | /api/payment/invoices                 | ورود   | پرداخت‌ها و فاکتورهای کاربر (شماره‌ی فاکتور، تخفیف، کد پیگیری) |
| GET                        | /api/payment/orders/{id}/pay          | عمومی  | رفتن به درگاه زرین‌پال                                         |
| GET                        | /api/payment/zarinpal/callback        | عمومی  | برگشت از درگاه + تأیید سرور-به-سرور                            |
| POST                       | /api/payment/bazaar/verify            | ورود   | تأیید خرید کافه‌بازار                                          |
| GET                        | /api/discount/validate                | ورود   | سنجش کد تخفیف                                                  |
| POST                       | /api/discount/redeem                  | ورود   | ثبت مصرف کد                                                    |
| GET · POST                 | /redeem · /api/discount/offers/mine   | ورود   | پیشنهادهای کاربر                                               |

| روش                | مسیر                                      | دسترسی                       | کار                                           |
|--------------------|-------------------------------------------|------------------------------|-----------------------------------------------|
| اجتماعی و پشتیبانی |                                           |                              |                                               |
| GET · POST         | /api/affiliate/me · /track · /conversions | ورود                         | کد معرف، ثبت معرف، پورسانت‌ها                 |
| GET                | /api/notifications                        | ورود                         | اعلان‌ها                                      |
| PATCH              | /api/notifications/{id}/read · /read-all  | ورود                         | خوانده‌شده                                    |
| GET · POST         | /api/support/tickets                      | ورود                         | تیکت‌های من / تیکت تازه                       |
| GET · POST         | /api/support/tickets/{id} · /messages     | ورود                         | جزئیات و پیام تازه                            |
| POST               | /api/support/bug-reports                  | ورود                         | گزارش باگ                                     |
| GET                | /api/user/weekly-report                   | ورود                         | «گزارش والدین»: ۷ روز اخیر در کنار ۷ روز قبلی |
| سیستم              |                                           |                              |                                               |
| GET                | /api/health                               | عمومی (جزئیات با کلید داخلی) | سلامت سرویس (بخش ۱۳)                          |

## ۹. دستیار هوش مصنوعی و داده‌ی Qdrant

دستیار «از حافظه‌ی مدل» جواب نمی‌دهد؛ اول تکه‌های مرتبطِ همان کتاب و همان فصل را پیدا می‌کند و مدل جواب را بر پایه‌ی آن‌ها می‌نویسد. به این روش (Retrieval-Augmented Generation) RAG می‌گویند.

### ۹-۱ Qdrant چیست و چه چیزی در آن است

Qdrant یک «بانک برداری» است: هر تکه‌ی کتاب به یک بردار (۱۵۳۶ عدد) تبدیل می‌شود که «معنا»ی آن را نشان می‌دهد؛ سؤال دانش‌آموز هم به بردار تبدیل می‌شود و نزدیک‌ترین تکه‌ها پیدا می‌شوند. هر **collection** یک کتاب است و هر **نقطه** (point) یک تکه – معمولاً یک تیپ‌مسئله از یک صفحه، با روش حل، صورت مسئله و راه حل کتاب.

| collection (کتاب)                            | تکه                | صفحه‌ی خوانده‌شده | چه کسی ساخت                                           |
|----------------------------------------------|--------------------|-------------------|-------------------------------------------------------|
| ریاضی چهارم · پنجم                           | ۵۹۴ · ۵۹۰          | ۱۹۶ · ۱۵۴         | ما – خواندن PDF با مدلِ بینایی                        |
| ریاضی هشتم · نهم · دهم                       | ۴۵۰ · ۵۱۸<br>۴۵۸   | ۱۷۰ · ۱۴۳ · ۱۵۰   | ما                                                    |
| حسابان یازدهم · دوازدهم                      | ۳۲۶ · ۳۷۶          | ۱۴۶ · ۱۵۱         | ما                                                    |
| هندسه دهم · یازدهم · دوازدهم                 | ۰ ۱۷۱ · ۲۲۹<br>۱۹۲ | ۷۸ · ۷۲ · ۸۸      | ما                                                    |
| آمار و احتمال یازدهم · گسسته دوازدهم         | ۲۵۴ · ۲۷۱          | ۸۴ · ۱۲۱          | ما                                                    |
| ریاضی سوم · ششم · هفتم                       | ۰ ۴۳۷ · ۴۳۶<br>۳۷۹ | –                 | تیم دیگر شرکت؛ با rag_import_collection خوانده شد     |
| علوم تجربی سوم · مطالعات اجتماعی ششم و چهارم | ۲۷ · ۴۰ · ۷۰       | –                 | تیم دیگر (در اپ ریاضی استفاده نمی‌شود)                |
| جمع                                          | ۵,۸۱۸              | ۱,۷۵۳             | نشانی: qdrant.tadrisino2.ir – سرویس مشترک با تیم دیگر |

### مشخصات فنی

|                         |                                     |
|-------------------------|-------------------------------------|
| مدل بردار (embedding)   | text-embedding-3-small              |
| ابعاد و فاصله           | ۱۵۳۶ · کسینوسی                      |
| تعداد تکه در هر جست‌وجو | ۴ ( RAG_RETRIEVAL_TOP_K )           |
| فیلتر ریاضی             | فیلتر chapter = «فصل ۲: عددهای اول» |
| نام collection          | "{پایه} {درس}" مثل «ریاضی نهم»      |

```
{ "chapter": "فصل ۸: حجم و مساحت",
  "grade": "نهم", "type": "تعریف کره",
  "keywords": ["کره", "شعاع", "مرکز"],
  "generalSolution": "تعریف دایره را ۱۰.",
  "problem": "...", "solution": "...",
  "solution_verdict": "agrees",
  "source_page": 139, "chapter_number": 8 }
```

ساخته در `rag/services/ingest.py` → `_payload`

## ۹-۲ داده چطور وارد شد: از PDF تا Qdrant

1. کتاب PDF `/data/textbooks` + نقشه‌ی فصل‌ها در `textbooks.py`

2. تصویر صفحه ۲۰۰ DPI `pdf2image`

3. مدل بینایی `gpt-4.1-mini` ~ ۰/۲۲ سنت برای هر صفحه

4. JSON ساختاریافته نوع، کلیدواژه، روش، مسئله، راه‌حل

5. LessonChunk ردیف در دیتابیس (منبع اصلی، با بردار)

6. Qdrant upsert نقطه + `payload` IngestedPage برای ازسرگیری

قاعده‌ی مهم پرامپت ( `rag/services/pdf.py` ) : اگر مسئله به «شکل زیر» ارجاع می‌دهد، مدل باید اندازه‌ها و برجسب‌های شکل را داخل متن بنویسد. این قاعده سهم مسئله‌های بی‌استفاده‌ی وابسته به شکل را از ~۴۲٪ به ~۱۴٪ رساند. هزینه‌ی خواندن پنج کتاب اول در یک روز حدود ۳/۴۴ دلار بود.

## ۹-۳ کنترل کیفیت: راستی‌آزمایی راه‌حل‌های کتاب

مدل بینایی گاهی عدد را بد می‌خواند یا راه‌حل را به مسئله‌ی کناری می‌چسباند. دستور `rag_verify_solutions` هر مسئله را مستقل دوباره حل می‌کند و رأی سه‌حالتی می‌دهد:

| رأی                                | تعداد | اثر                                                  |
|------------------------------------|-------|------------------------------------------------------|
| agrees راه‌حل کتاب درست است        | ۴,۱۹۱ | فقط از این‌ها سؤال ساخته می‌شود                      |
| disagrees راه‌حل کتاب غلط است      | ۶۴۴   | راه‌حلش به دستیار داده نمی‌شود؛ فقط روش و صورت مسئله |
| unjudgeable بی‌شکل قابل قضاوت نیست | ۷۴۲   | کنار می‌ماند                                         |
| بررسی نشده                         | ۲۴۱   | بیشترش علوم و مطالعات تیم دیگر                       |

متن اصلی کتاب هیچ‌وقت بازنویسی نمی‌شود؛ جواب مستقل جدا در `checked_answer` ذخیره می‌شود تا آدم بتواند داوری کند.

## ۹-۴ افزودن و پاک کردن داده

| کار                                    | روش                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | هزینه                 |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| افزودن کتاب تازه                       | <p>(۱) PDF را در <code>data/textbooks/</code> بگذارید و فصل‌ها و <code>page_offset</code> را در <code>rag/services/textbooks.py</code> ثبت کنید.</p> <p>(۲) <code>python manage.py rag_ingest_book --grade نهم --plan</code> ← فقط نشان می‌دهد چه چیزی و با چه هزینه‌ای خوانده می‌شود.</p> <p>(۳) همان دستور بدون <code>--plan</code> (قابل قطع و ازسرگیری؛ صفحه‌ی خوانده‌شده دوباره خرید نمی‌شود).</p> <p>(۴) <code>rag_verify_solutions</code> ← (۵) <code>seed_curriculum_from_textbook</code> ← (۶) ساخت سؤال (بخش ۱۰).</p> | <p>~۲۲۰ه/سنت/صفحه</p> |
| افزودن یک متن یا PDF                   | <code>POST /api/rag/ingest/text</code> یا <code>/pdf</code> با هدر <code>X-Internal-Key</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                | کم                    |
| آوردن <code>collection</code> تیم دیگر | <code>rag_import_collection --collection "ریاضی ششم"</code> – فقط می‌خواند، چیزی در Qdrant نمی‌نویسد.                                                                                                                                                                                                                                                                                                                                                                                                                           | رایگان                |
| بازخوانی صفحه‌های معیوب                | <code>rag_ingest_book --grade نهم --refresh figures</code> یا <code>--refresh-pages 12,15,20-24</code> . هر صفحه اول پاک می‌شود (اول Qdrant، بعد دیتابیس – تابع <code>clear_page</code> ) و دوباره خوانده می‌شود.                                                                                                                                                                                                                                                                                                               | مثلی خواندن           |
| پاک کردن                               | دستور جداگانه‌ی «فقط پاک کن» نداریم. برای یک صفحه تابع <code>rag.services.ingest.clear_page(collection, page)</code> از <code>manage.py shell</code> برای کل <code>collection</code> ، <code>delete_collection</code> کلاینت Qdrant – فقط با هماهنگی تیم دیگر، چون Qdrant مشترک است.                                                                                                                                                                                                                                            | –                     |

### دو احتیاط (از خواندن کد؛ آزموده نشده)

- پاک کردن یک `LessonChunk` از پنل ادمین نقطه‌ی Qdrant آن را پاک نمی‌کند (هیچ شنونده‌ای برایش نیست)؛ نقطه بی‌صاحب می‌ماند و در جست‌وجو پیدا می‌شود. از `clear_page` استفاده کنید.
- ردیف‌های `LessonChunk` (نسخه‌ی ساختاریافته با بردار) در دیتابیس محلی توسعه‌اند، نه روی سرور؛ سرور فقط از Qdrant می‌خواند. از آن فایل بکاپ جدا نگه دارید – با آن، Qdrant بی‌هزینه‌ی embedding دوباره ساخته می‌شود.

## ۹-۵ از سؤال تا جواب (خلاصه‌ی سناریوی ۳-۶)

| مرحله      | فایل                                                                                 | نکته                                                                            |
|------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| کدام کتاب؟ | <code>rag/services/lesson_context.py</code>                                          | از <code>lessonId</code> زنجیره‌ی درس → جزیره → فصل → پایه؛ یا پایه‌ی خود کاربر |
| جست‌وجو    | <code>retrieval.py</code> · <code>qdrant_store.py</code> · <code>embedding.py</code> | متن جست‌وجو = سؤال + سؤال قبلی کاربر (برای «نفهمیدم»)                           |

| مرحله                    | فایل                            | نکته                                                                                                                                                                               |
|--------------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| پرامپت                   | generation.py                   | نسخه‌ی tutor-v12-age-tone؛ فقط ریاضی؛ لحن سه‌سطحی (دبستان/متوسطه‌ی اول/دبیرستان) و لحن بزرگسال برای ۱۸+؛ متن ساده بی‌LaTeX؛ حل مرحله‌به‌مرحله با واری و «✓ جواب»؛ ۶ پیام آخر گفتگو |
| جواب درست/ غلط دانش‌آموز | generation.py → question_focus  | با questionId دستیار صورت سؤال و آخرین جواب دانش‌آموز را می‌بیند و می‌گوید اشتباه کجا بود                                                                                          |
| تمیزکاری                 | mathtext.py                     | هرچه LaTeX از دست مدل در برود به یونیکد برمی‌گردد ( $\sqrt{3}$ ، $x^2$ ) و اعراب حذف می‌شود                                                                                        |
| شکل                      | diagram.py ← فرانت src/diagram/ | مدل فقط توصیف JSON می‌نویسد؛ سرور جدا، تعمیر و با فهرست سفید واری می‌کند؛ اپ مختصات را حساب و SVG رسم می‌کند. ۱۵ نوع شکل – شرح کامل در کادر بنفش پایین.                            |
| ثبت                      | ai_tutor/models.py → Message    | مدل، نسخه‌ی پرامپت، تعداد تکه، تأخیر و شکل برای هر جواب ذخیره می‌شود ← کیفیت قابل اندازه‌گیری است                                                                                  |

## مهم‌ترین ایده‌ی فنی دستیار

### شکل را هوش مصنوعی نمی‌کشد: مدل می‌گوید «چه شکلی»، اپ آن را با فرمول می‌کشد

هوش مصنوعی ریاضی را می‌فهمد ولی در حساب دقیق مختصات اشتباه می‌کند؛ کامپیوتر برعکس، هیچ‌وقت در حساب اشتباه نمی‌کند ولی مسئله را نمی‌فهمد. پس کار بین این دو تقسیم شد: **مدل فقط توصیف شکل را به صورت داده (JSON) می‌نویسد، سرور آن را واری می‌کند، و کد اپ رأس‌ها، ریشه‌ها و زاویه‌ها را خودش حساب و رسم می‌کند.** نام این الگو در مهندسی نرم‌افزار: «توصیف اعلانی + رسم قطعی» (Declarative Spec + Deterministic Rendering) – همان ایده‌ی پشت Vega-Lite و Mermaid، این‌جا مخصوص کتاب‌های ریاضی ایران، با ۱۵ نوع شکل.

### گام ۱: سرور → مدل

سرور به مدل می‌گوید اگر شکل لازم است، فقط JSON بنویسد – و فقط بخش مربوط از راهنمای شکل‌ها را کنار سؤال می‌گذارد (برای «۳۴۵ × ۱۲» هیچ). rag/services/diagram.py

متن واقعی قاعده (DIAGRAM\_RULE): «در پایان پاسخ دقیقاً یک بلوک `<diagram> ... </diagram>` بگذار که داخلش فقط یک JSON است، نه SVG و نه HTML. برای حساب عددی یا ساده‌کردن عبارت جبری که شکلی ندارد، شکل نگذار.»

### گام ۲: مدل

مدل حل قدم‌به‌قدم را می‌نویسد و در آخر فقط توصیف شکل (~۱۵۰ توکن، نه صدها خط کد تصویر):

```
... ✓ جواب: BC = 10
<diagram>{"type":"triangle", "names":["A","B","C"],
"sides":{"AB":6,"AC":8}, "angles":{"A":90}, "labels":{"BC":"x"}}</diagram>
```

### گام ۳: سرور

سرور بلوک را از متن جدا، اشتباه‌های قالبی مدل را تعمیر، و با فهرست سفید واری می‌کند (رقم فارسی در عدد،  $2\pi$ ، کامنت، گیومه‌ی جافتاده؛ نوع و رنگ و کلید ناشناخته رد؛ عددها محدود). شکل خراب فقط حذف می‌شود و جواب متنی سر جایش می‌ماند. نتیجه در `Message.diagram` ذخیره و در رویداد `done` فرستاده می‌شود. کار سرور ~۴/۰ میلی‌ثانیه.

```
# rag/services/diagram.py
def extract(text) -> tuple[str, dict | None]:
    """(text without the block, spec or None)"""
    match = _BLOCK.search(text) # <diagram>(.*?)</diagram>
    spec = parse(match.group(1)) if match else None # _repair -> json -> _normalize -> validate
    return _BLOCK.sub('', text).rstrip(), spec
```

#### گام ۴۰ اپ (گوشی)

اپ هر عددی را که قابل حساب است خودش حساب می‌کند و SVG می‌کشد: رأس‌های مثلث با قانون کسینوس‌ها، ریشه‌ی نمودار با نصف‌کردن بازه تا دقت  $10^{-11}$ . رسم روی گوشی خود دانش‌آموز انجام می‌شود، پس هزار کاربر یعنی بارِ صفر روی سرور.

`src/diagram/expr.ts` `src/diagram/shapes.ts`

```
// src/diagram/shapes.ts - law of cosines: third side from two sides + angle
s[z] = Math.sqrt(s[x]**2 + s[y]**2 - 2*s[x]*s[y]*Math.cos(g[opp[z]]*RAD));
// src/diagram/expr.ts - roots: scan 800 points, bisect where the sign flips
if (Math.sign(y) !== Math.sign(py)) { const r = bisect(f, px, x); if (Math.abs(f(r)) < 1e-6)
out.push(r); }
```

مثال ۱۰ مثلث قائم‌الزاویه

مدل می‌نویسد

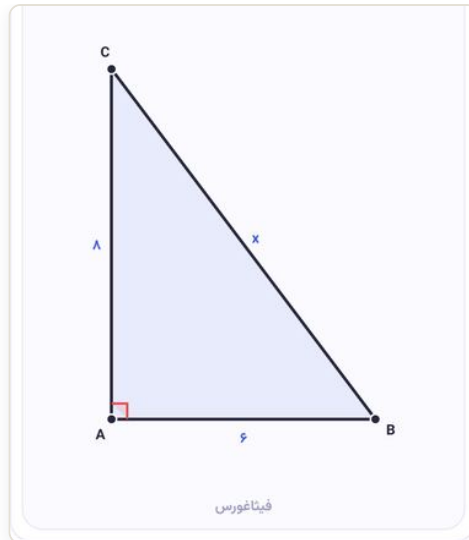
```
{"type":"triangle","names":["A","B","C"],
"sides":{"AB":6,"AC":8},"angles":{"A":90},
"labels":{"BC":"x"}}
```

اپ حساب می‌کند

قانون کسینوس‌ها:  $BC^2 = 6^2 + 8^2 - 2 \cdot 6 \cdot 8 \cdot \cos 90^\circ = 100$

پس  $BC = 10$  و رأس‌ها  $A(0,0) \cdot B(6,0) \cdot C(0,8)$

دانش‌آموز می‌بیند



مثال ۲. ریشه و رأس سهمی

مدل می‌نویسد

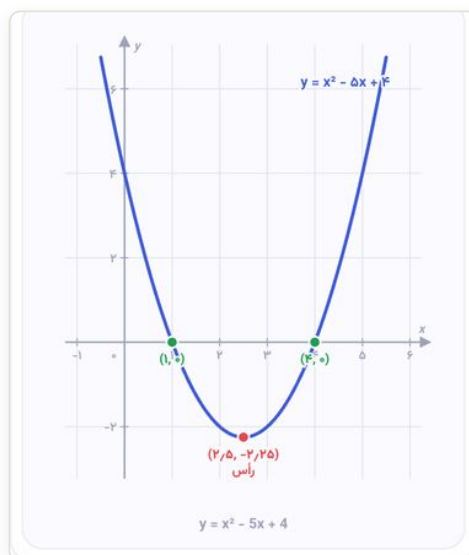
```
{"type": "function", "expr": "x^2-5*x+4",  
"marks": ["roots", "vertex"]}
```

اپ حساب می‌کند

۸۰۰ نقطه بررسی می‌شود؛ هر جا علامت عوض شد، با نصف کردن بازه ریشه پیدا می‌شود:  $x = 1$  و  $x = 4$

جایی که شیب صفر است، رأس:  $(2.5, -2.25)$

دانش‌آموز می‌بیند



| روشی رایج: مدل خودش SVG/HTML بنویسد | روشی ما: مدل توصیف، اپ رسم                                         |                                                  |
|-------------------------------------|--------------------------------------------------------------------|--------------------------------------------------|
| امنیت                               | کد مدل در صفحه اجرا می‌شود؛ متن مخرب در سؤال می‌تواند کد تزریق کند | هیچ کدی از مدل اجرا نمی‌شود؛ فقط JSON واریسی شده |
| دقت                                 | مختصات را حدس می‌زند؛ پنج ضلعی کج، ریشه جابه‌جا                    | با فرمول حساب می‌شود                             |

| هزینه                      | روشی رایج: مدل خودش SVG/HTML بنویسد                                         | روشی ما: مدل توصیف، اپ رسم               |
|----------------------------|-----------------------------------------------------------------------------|------------------------------------------|
| چند برابر توکن برای هر شکل | ~۱۵۰ توکن؛ برای سؤال بی‌شکل صفر (۰/۰۰۰۳ تا ۰/۰۰۰۸ دلار در هر جوابِ شکل‌دار) |                                          |
| اگر خراب شد                | صفحه یا چت به هم می‌ریزد                                                    | فقط شکل نمی‌آید؛ جوابِ متنی سالم می‌ماند |



## از پی بپرس

دستیار ریاضی، از روی کتاب خودت

گفتگوی تازه

۴) پس ۲۰ درصد از یک ساعت برابر است با ۱۲ دقیقه.

✓ جواب: ۱۲ دقیقه

در شکی پایین، ۲۰ خانه از ۱۰۰ خانه رنگ شده که نشان‌دهنده ۲۰ درصد است.

حالا خودت امتحان کن: ۱۵ درصد از ۸۰ دقیقه چند دقیقه می‌شود؟ 😊



۲۰ خانه از ۱۰۰ خانه، یعنی ۲۰٪  
20 درصد



سوالت رو بنویس...

امروز ۱۲ سوال دیگه می‌تونی بپرسی



من



اشتراک



دستیار



نقشه



خانه



## از پی بپرس

دستیار ریاضی، از روی کتاب خودت

گفتگوی تازه

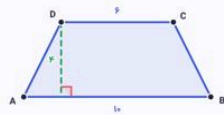
۶) حالا تقسیم بر ۲:

$$۶۴ \div ۲ = ۳۲$$

✓ جواب: مساحت دوزنقه برابر ۳۲ واحد مربع است.

در شکی پایین، دوزنقه با قاعده‌های ۱۰ و ۶ و ارتفاع ۴ نشان داده شده است که به فهمیدن بهتر کمک می‌کند.

حالا خودت امتحان کن: مساحت دوزنقه‌ای با قاعده‌های ۸ و ۵ و ارتفاع ۳ چقدر است؟



دوزنقه با قاعده‌های 10 و 6 و ارتفاع 4



سوالت رو بنویس...

امروز ۱۴ سوال دیگه می‌تونی بپرسی



من



اشتراک



دستیار



نقشه



خانه



فایل‌ها – بک‌اند: `rag/services/diagram.py` (قاعده، تشخیص، تعمیر، واریسی) · `rag/services/generation.py` (پرامپت، جدا کردن در استریم) · `ai_tutor/models.py` → `Message.diagram` · تست‌ها `rag/tests_diagram.py` – فرانت: `src/diagram/index.tsx` (انتخاب نوع، محافظ خطا) · `plane.tsx` · `shapes.ts` · `expr.ts`

## ۹-۶ کیفیت با عدد

| سنجش                                              | نتیجه                                                                                                                                                             |
|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| انتخاب مدل جواب: ۱۰۲ مسئله‌ی تصحیح‌شده‌ی دستی     | <code>gpt-4.1-mini</code> : ۹۵-۹۶ درست · <code>gpt-4o-mini</code> : ۸۵-۹۰                                                                                         |
| شکل‌ها: ۶۴ مسئله‌ی تصادفی کتاب، واریسی چشمی تک‌تک | ۴۰ درست، ۱۲ قابل قبول، ۷ غلط، ۵ بی‌شکل – حدود یکی از پنج هنوز ایراد دارد؛ برای همین شکل همیشه کنار حل متنی می‌آید                                                 |
| سرعت                                              | جواب کامل با شکل ~۳۰ ثانیه؛ اندازه‌گیری شد که در خود GapGPT است («فقط بگو سلام» = ۷ ثانیه)                                                                        |
| سقف‌ها                                            | ۲ سؤال رایگان در روز (مشترک ۴)؛ ۲۰ در ساعت؛ ۳,۰۰۰ در روز برای کل سامانه؛ حداکثر ۱۲ جواب هم‌زمان در هر worker (بیشتر از آن ۵۰۳ «دستیار شلوغه» پیش از کم شدن سهمیه) |

## ۹-۷ فقط ریاضی، و چه چیزی هزینه دارد

دستیار فقط درباره‌ی ریاضی حرف می‌زند ( `rag/services/guardrails.py` ). سه لایه دارد: (۱) صافی سرور پیش از مدل و بی‌هزینه، فقط برای ترفندهای آشکار («دستورهای قبلی را نادیده بگیر»، «پرامپت را بگو»); (۲) قاعده‌ی سخت در پرامپت که هر

چیزِ غیرِ ریاضی را با علامتی ثابتِ جواب می‌دهد؛ (۳) سرور آن علامت را با متنِ ثابتِ رد عوض می‌کند. جمله‌ای درباره‌ی آسیب به خود رد نمی‌شود و جوابی مهربان با شماره‌ی اورژانسیِ اجتماعی (۱۲۳) و صدای مشاور (۱۴۸۰) می‌گیرد؛ از ۱۰ مهر هر شکلِ نوشتنش (با فاصله، بی‌نیم‌فاصله) گرفته می‌شود. آزمونِ زنده: ۷ از ۷ تلاشِ خروج از ریاضی رد شد.

| قاعده‌ی<br>خرج |                                                                                                                               |
|----------------|-------------------------------------------------------------------------------------------------------------------------------|
| ترتیب          | فقط سقفِ روزانه (۲ رایگان، مشترک ۴؛ Message.charge : none / free )                                                            |
| خرج<br>می‌شود  | هر سوالی که به مدل برسد، از جمله رد «غیر ریاضی» توسطِ مدل                                                                     |
| خرج<br>نمی‌شود | رد صافیِ سرور، جوابِ کش‌شده، جوابی که ساخته نشد، و تکرارِ درخواست با همان Idempotency-Key                                     |
| هم‌زمانی       | دو سوالِ هم‌زمان با یک اعتبار فقط یک جواب می‌گیرند؛ با دو thread واقعی آزموده می‌شود ( ai_tutor/tests_credit_concurrency.py ) |
| هزینه          | هر سوال حدود ۰/۲ سنت؛ قیمتِ اندازه‌گیری‌شده‌ی هر مدل و تاریخچه‌اش در docs/model-prices.md                                     |

## ۱۰. محتوای آموزشی و ساختِ سؤال

هدف: محتوا بدون برنامه‌نویس قابل ویرایش باشد، و یک سرور خالی با یک دستور پر شود.

### محتوا کجا زندگی می‌کند

| فایل                         | محتوا                                                                      |
|------------------------------|----------------------------------------------------------------------------|
| content/seed/curriculum.json | ۱۰ پایه، ۵ کتاب، ۸۹ فصل، ۱,۱۴۵ جزیره، ۵,۴۰۶ درس، با مثال، قدم‌ها و شکل درس |
| content/seed/questions.json  | ۱۸,۵۲۲ سؤال چهارگزینه‌ای، بعضی با شکل                                      |

- `seed_content` در هر انتشار اجرا می‌شود؛ هیچ‌وقت چیزی پاک نمی‌کند و اجرای دوباره‌اش بی‌خطر است.
- سؤال با مسیر **عنوان** (پایه > فصل > جزیره > درس) به درس وصل می‌شود، نه با شناسه — چون شناسه‌ها بین دو دیتابیس فرق دارند.
- کار ادمین با انتشار بعدی برنمی‌گردد: دفتر `SeedRecord` به یاد دارد `seed` چه چیزی تحویل داده، پس سؤال پاک‌شده یا فصلی تغییرنام‌داده دوباره ساخته نمی‌شود؛ مثال، قدم‌ها و شکل فقط خانه‌ی خالی را پر می‌کنند.

### کارهای روزمره از پنل ادمین

| کار                           | کجا                                                                         |
|-------------------------------|-----------------------------------------------------------------------------|
| افزودن یا ویرایش یک سؤال      | <code>/admin/content/question/</code> ← «افزودن»                            |
| وارد کردن فله‌ای (همه یا هیچ) | همان صفحه ← «وارد کردن سؤال‌ها از فایل»                                     |
| خروجی سؤال یا برنامه‌ی درسی   | انتخاب ← اکشن «دریافت فایل» (همان قالبی که ورودی می‌پذیرد)                  |
| پایه‌ی تازه                   | <code>/admin/content/grade/add/</code> — عنوان «پایه چهارم»، شناسه خودکار ۴ |
| پلن، همراه، تیکت، کد تخفیف    | صفحه‌ی اپ مربوط در پنل                                                      |
| بازخورد دانش‌آموز             | ستون 👍 / 🙌 در فهرست سؤال‌ها و درس‌ها؛ صف «گزارش‌های سوال» و «گزارش‌های درس» |

### سؤال‌ها چگونه ساخته شدند

۱. تکه‌های `agrees` فقط راه‌حل‌های تأییدشده
  ۲. `export generate_questions_from_chunks` رایگان
  ۳. نویسنده‌ی سؤال `subagent` های Claude (Sonnet) با همان قواعد پرامپت
  ۴. `import` با اعتبارسنجی `content/question_io.py`
  ۵. `commit` و `questions.json seed_content --export`
- بازده حدود ۸۷–۹۲٪ بود؛ بقیه عمداً رد شدند (بیشتر مسئله‌هایی که بی‌شکلی کتاب خودکفا نمی‌شوند). همین مسیر با API پولی هم ممکن است ولی اندازه‌گیری‌شده ~۷۷٪ سنت برای هر سؤال گران‌تر بود بی‌کیفیت بهتر. نویسنده‌های سؤال چند بار خطای حساب خود کتاب را هم گرفتند.

**جزیره‌های تمرین:** آخر هر فصل از پایه‌ی سوم تا دوازدهم یک تا چند جزیره‌ی «تمرین» است (۱۶۰ جزیره، ~۱۳/۷ هزار سؤال). جواب این سؤال‌ها با برنامه **محاسبه** شده، نه حفظ، و گزینه‌های غلط از جنس خطای رایج دانش‌آموزند. قالب سؤال‌های هر خانواده یکی است و فقط عددها عوض می‌شوند؛ بازبینی معلم هنوز انجام نشده است.

**شکل درس:** درس‌های هندسه، آمار و گسسته شکل ثابت دارند ( Lesson. figures )، دستی نوشته و با چشم بازبینی شده؛ هنگام نمایش هیچ مدلی صدا زده نمی‌شود.

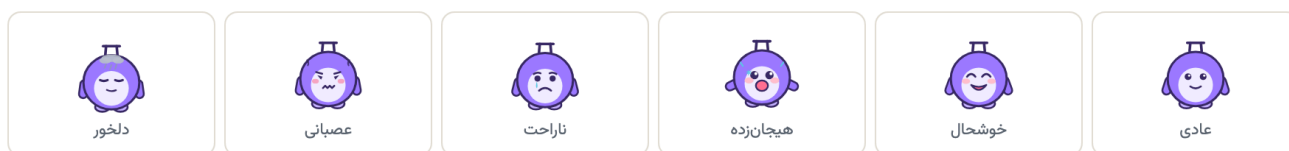
## ۱۱. گیمیفیکیشن و همراه‌ها

سازوکارهای شناخته‌شده‌ی اپ‌های آموزشی موفق (مانندِ دولینگو): مسیرِ مرحله‌به‌مرحله، امتیاز، قلب، زنجیره‌ی روزانه، رقابت و شخصیت‌هایی که با دانش‌آموز واکنش نشان می‌دهند. همه در سرور حساب می‌شوند، نه در گوشی، تا قابلِ تقلب نباشند.

### همراه‌ها



شش حالتِ چهره‌ی هر همراه؛ اپ بر اساسِ نتیجه‌ی یکی را نشان می‌دهد (سرور در پاسخِ هر جواب `mascotMood` می‌فرستد):



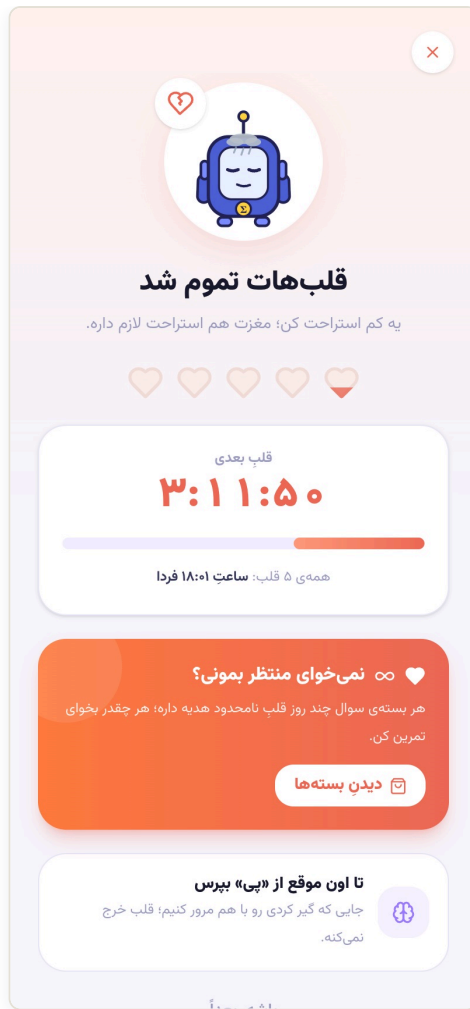
| بخش     | چطور کار می‌کند                                                                                                                                             |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| کاتالوگ | جدول <code>Character</code> در <code>progress/models.py</code> ؛ ادمین می‌سازد و آستانه‌ی <code>unlock_xp</code> را تعیین می‌کند (مثل پلن‌ها).              |
| باز شدن | <code>UserStats.xp ≥ unlock_xp</code> – جدول مالکیت جدا ندارد؛ ساده و بی‌خطا.                                                                               |
| انتخاب  | <code>POST /api/characters/select</code> فقط همراهِ باز شده را می‌پذیرد و در <code>UserStats.selected_character_id</code> می‌نشانند.                        |
| تصویر   | هیچ فایل تصویری نیست: <code>src/mascots/seyar.ts</code> هر همراه و هر حالت را از کد به SVG می‌سازد (تصاویر بالا همان خروجی‌اند) ← حجم کم، واضح روی هر صفحه. |

### قاعده‌های بازی

| سازوکار     | قاعده                                                                                                                                                                   | کجای کد                                                           |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| امتیاز (XP) | پایانِ درس یا جزیره: ۱۰ × جوابِ درست، فقط بارِ اول. عددِ جوابِ درستی که اپ می‌فرستد به جواب‌های درستِ ثبت‌شده محدود می‌شود؛ صدا زدنِ مستقیمِ API امتیازِ مجانی نمی‌دهد. | <code>content/views.py</code> → <code>NodeQuizCompleteView</code> |
| ستاره       | ۹۰٪ به بالا ۳۰٪ به بالا ۲۰٪ بیشتر از صفر ۱                                                                                                                              | همان                                                              |
| سکه         | ۵ × ستاره، فقط بارِ اول                                                                                                                                                 | همان                                                              |

| سازوکار         | قاعده                                                                                                                                                                                                                                                                                                  | کجای کد                                               |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| قلب             | ۵ قلب؛ هر جواب غلط یکی کم؛ هر ۵ ساعت یکی برمی‌گردد و ۲۴ ساعت بعد از اولین قلب از دست‌رفته همه پر می‌شوند. با قلب صفر نه جوابی پذیرفته می‌شود نه پایان درس یا جزیره‌ی بی‌جواب ( 403 no_hearts ). اشتراکِ فعال = قلب نامحدود.                                                                            | progress/models.py →<br>content/views.py · UserHearts |
| زنجیره (Streak) | جواب، پایان درس یا جزیره امروز را (به وقتِ تهران) روز فعال می‌کند. اگر آخرین روز فعال نه امروز باشد نه دیروز، صفر نشان داده می‌شود.                                                                                                                                                                    | progress/services.py                                  |
| جدول رتبه       | هفتگی (امتیاز ۷ روز اخیر) و همه‌ی زمان‌ها؛ زنده حساب و ۶۰ ثانیه کش می‌شود.                                                                                                                                                                                                                             | progress/leaderboard.py                               |
| پیشرفت روی نقشه | پایانِ آزمون، جزیره را با هر نمره‌ای تمام می‌کند و جزیره‌ی بعدی ( nextNodeId ) باز می‌شود؛ نمره فقط ستاره و امتیاز را تعیین می‌کند (تصمیم محصولی: ساده بماند). جزیره‌ی قفل ( 403 node_locked ) و آزمون جواب‌نداده ( 409 answers_missing ) پذیرفته نمی‌شوند. آزمون هر جزیره برای هر دانش‌آموز ثابت است. | content/views.py                                      |
| دیوار پرداخت    | فصل اول هر کتاب رایگان؛ بقیه با اشتراکِ فعال ( 402 subscription_required ).                                                                                                                                                                                                                            | content/views.py →<br>unit_is_free                    |
| اعلان           | «جزیره را با ۳ ستاره تمام کردی و ۴۰ امتیاز گرفتی» – با رقمِ فارسی.                                                                                                                                                                                                                                     | notifications/services.py                             |

**قلب واقعی است ( HEARTS\_ENFORCED )** پیش‌فرض روشن): قلبِ صفر بازی را متوقف می‌کند تا قلبِ بعدی برگردد. چکِ قلب داخلِ همان تراکنشی است که قلب را کم می‌کند، پس دو جوابِ هم‌زمان با یک قلبِ باقی‌مانده هر دو پذیرفته نمی‌شوند؛ و پایانِ آزمون بی‌جواب دادن هم بسته است. کسی که آخرین قلبش را روی آخرین سوال از دست داده، جزیره را تمام می‌کند. تست‌ها: content/tests\_hearts\_gate.py .



شکل ۱۱-۳ – صفحه‌ی «قلبهات تموم شد»: شمارش معکوس زنده تا قلب بعدی، قلبی که کم‌کم پر می‌شود، ساعت پر شدن همه، پیشنهاد اشتراک و «تا اون موقع از پی بی‌رس» (پرسیدن قلب خرج نمی‌کند). وقتی وقتش رسید، صفحه خودش از سرور می‌پرسد و «ادامه بده» را نشان می‌دهد. فرانت: `src/components/NoHeartsScreen.tsx`.

## ۱۲. امنیت و احراز هویت

چند دور بازیبنی بدبینانه روی کلِ بک‌اند انجام شد (مرداد، ۳، ۱۰ و ۲۱ مهر)، با واریسی روی سایت و سرورِ زنده، با این سؤال: «یک دانش‌آموزِ متقلب، یک مهاجم یا یک والدِ سردرگم چطور این را می‌شکند؟» هر یافته با یک تست بسته شد. این بخش صادقانه می‌گوید چه بسته است و چه هنوز باز.

### ۱۲-۱ هویت و توکن

۱. کد پیامکی ۶ رقم، تصادفی رمزنگارانه؛ ۵ حدسِ غلط = باطل
  ۲. access token user\_id: زمان: امضا ۱ ساعت؛ در دیتابیس نیست
  ۳. refresh token در جدولِ Session ؛ ۳۰ روز؛ قابلِ باطل کردن
  ۴. خروج tokens\_valid\_after : هر توکنِ قدیمی‌تر رد می‌شود
- کاربرهای اپ ( accounts.User ) عمداً از کاربرهای پنلِ ادمین (کاربرِ استاندارد جنگو، با رمز) جدا هستند؛ دانش‌آموز هیچ راهی به پنل ندارد.

### ۱۲-۲ آنچه رعایت شده

| حوزه                       | چه کرده‌ایم                                                                                                                                                                                                                                                          | کجا                                                         |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| دسترسی به داده‌ی دیگران    | شناسه‌ی کاربر همیشه از توکن خوانده می‌شود، هرگز از ورودی؛ هر کاربر فقط داده‌ی خودش را می‌بیند (گزارش والدین هم در حسابِ خود دانش‌آموز است).                                                                                                                          | همه‌ی views.py ها                                           |
| سقفِ نرخ                   | روی همه‌ی نقاطِ حساس (ورود، دستیار، خرید، کدِ تخفیف، جواب...); درخواستِ بی‌ورود بر اساسِ IP واقعی، که فقط از پروکسی‌های آروان پذیرفته می‌شود (جعلِ X-Forwarded-For اثری ندارد). nginx هر درخواستِ نرسیده از آروان را رد می‌کند.                                      | config/middleware.py → TrustedProxyMiddleware<br>ops/nginx/ |
| پیامک و هزینه‌اش           | شماره به 09XXXXXXXXX نرمال می‌شود؛ فاصله بین دو درخواست، ۱۰ در روز برای هر شماره، سقفِ روزانه برای کلِ سامانه (روی سرور ۲,۰۰۰، ضدِ SMS pumping) — شمارش در دیتابیس، نه حافظه‌ی موقت.                                                                                 | accounts/phone.py<br>tests_otp_limits.py                    |
| پرداخت                     | فقط تأییدِ سرور-به-سرور؛ برگشتِ جعلی اثری ندارد؛ webhook عمومی ناامن حذف شد؛ هر درگاهی بی‌کلید «بسته» می‌ماند (fail-closed)، نه موفقیتِ ساختگی. هر سفارش فقط یک بار روز می‌دهد (قیدِ یکتا روی تراکنش)؛ وضعیتِ سفارش در ادمین فقط با action لغو/بازپرداخت عوض می‌شود. | subscription/ · payment/                                    |
| تقلبِ امتیاز               | امتیاز فقط برای جوابِ ثبت‌شده؛ جایزه فقط بار اول.                                                                                                                                                                                                                    | content/views.py                                            |
| درخواستِ تکراری و هم‌زمانی | Idempotency-Key در شش نقطه؛ transaction.atomic ؛ قیدِ یکتا در دیتابیس؛ قفلِ ردیفی Postgres.                                                                                                                                                                          | config/settings.py                                          |

| حوزه       | چه کرده‌ایم                                                                                                                                                                                                                                       | کجا                                                                  |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| خطا و لاگ  | هر خطای JSON یکدست بی‌جزئیات داخلی؛ لاگ بدون بدنه، هدر، توکن یا رمز.                                                                                                                                                                              | · <code>config/exceptions.py</code><br><code>middleware.py</code>    |
| محتوای مدل | HTML یا SVG تولید مدل هرگز وارد صفحه نمی‌شود؛ فقط JSON واریسی‌شده با فهرست سفید. دستیار فقط ریاضی (بخش ۹-۷).                                                                                                                                      | · <code>rag/services/diagram.py</code><br><code>guardrails.py</code> |
| حذف حساب   | با کد پیامکی تازه (همان سقف حدس ورود)؛ سفارش‌ها برای حسابداری می‌مانند؛ توکن قبلی بعد از حذف کار نمی‌کند.                                                                                                                                         | <code>accounts/views.py</code> →<br><code>DeleteAccountView</code>   |
| مرورگر     | CORS فقط برای دامنه‌ی فرانت؛ CSP و هدرهای امنیتی روی سایت فرانت؛ پاسخ API با <code>no-store</code> در کش آروان نمی‌ماند؛ نام نمایشی از شماره و لینک پاک می‌شود؛ سقف حدس رمز ادمین برای هر IP.                                                     | · <code>config/middleware.py</code><br><code>ops/nginx/</code>       |
| کلیدها     | همه در <code>.env</code> ، سرور؛ کلید امضای توکن روی خودِ سرور ساخته شده و در هیچ مخزنی نیست؛ <code>/api/health</code> بی‌کلید داخلی فقط «ok/error» می‌گوید و Swagger فقط برای ادمین وارد شده باز است. هیچ ورود آزمایشی با کد ثابت روی سرور نیست. | <code>.env.example</code>                                            |

### ۳-۱۲ آنچه هنوز باز است (پیش از کاربر عمومی باید بسته شود)

فهرست کامل با خطر و راه بستن هرکدام در `docs/security-before-production.md` است؛ مهم‌ترین‌ها:

| مورد                                                                                                | خطر   | راه بستن                                                            |
|-----------------------------------------------------------------------------------------------------|-------|---------------------------------------------------------------------|
| سقف روزانه‌ی پیامک کل سامانه را یک ناشناس با شماره‌های تصادفی پر می‌کند و ورود برای همه بسته می‌شود | بالا  | CAPTCHA یا مانع مشابه پس از یک حد؛ شمارش بر اساس شبکه – تصمیم طراحی |
| دسترسی سرور: حساب مشترک با رمز و <code>sudo</code> کامل                                             | بالا  | کلید SSH و حساب محدود – کار ادمین سرور                              |
| پنل <code>/admin/</code> روی دامنه‌ی عمومی (با سقف تلاش، بی ۲FA)                                    | متوسط | محدود به IP یا ۲FA، VPN                                             |
| با دانستن شماره‌ی کسی، ورود او تا ۲۴ ساعت بسته می‌شود                                               | متوسط | گره زدن کد به درخواست‌کننده – تصمیم طراحی                           |
| کد پیامکی بی‌هش در دیتابیس؛ توکن در <code>localStorage</code> مرورگر                                | کم    | هش کردن؛ کوکی <code>httpOnly</code>                                 |
| کلید جواب هر سوال با حساب دوم درمی‌آید                                                              | کم    | گزینه‌ی درست فقط پس از پایان آزمون – تصمیم محصولی                   |

#### جمع‌بندی وضعیت امنیت

لایه‌ی برنامه بازبینی و تست شده است: دسترسی، سقف نرخ، جعل IP، پرداخت، تکرار درخواست و تقلب امتیاز بسته‌اند؛ کلید امضا عوض شده، حساب‌ها و ورودهای نمایشی پاک شده‌اند و `nginx` فقط از آروان می‌پذیرد. آنچه مانده بیشتر تصمیم طراحی (پیامک و ورود) یا کار ادمین سرور است.

## ۱۳. استقرار (Deploy) و نگهداری

هدف: هیچ مرحله‌ی دستی یادآوردنی در انتشار نباشد. یک دستور، و اسکریپت خودش ثابت می‌کند که کار شد.

### ۱۳-۱ انتشار بک‌اند

```
ssh -t administrator@<server> 'cd ~/mathapp-v2 && ./deploy.sh; sudo systemctl restart zareian-mathapp'
```

۱.۱ fetch از Gitea

۲.۲ reset تغییر دستی ← patch در /deploy-backups.

۳.۳ pip install

۴.۴ بک‌آپ دیتابیس (pg\_dump) و migrate

۵.۵ seed\_content محتوا

۶.۶ collectstatic

۷.۷ restart + اثبات ری‌استارت

۸.۸ health poll تا سالم

- **سرور را تعمیر می‌کند، جلوش نمی‌ایستد:** فایلی که کسی دستی روی سرور عوض کرده ذخیره و بعد دور ریخته می‌شود؛ سرور دقیقاً روی چیزی می‌رود که در Gitea است.
- **ری‌استارت را ثابت می‌کند:** زمان شروع سرویس را قبل و بعد مقایسه می‌کند. یک بار سرور دو روز کد قدیمی سرو کرد درحالی‌که همه‌ی فایل‌ها تازه بودند – از آن به بعد «فایل روی دیسک» را «منتشرشده» حساب نمی‌کنیم.
- ری‌استارت جداست چون `sudo` داخل اسکریپت نمی‌تواند رمز بپرسد.

### ۱۳-۲ انتشار فرانت

```
ssh -t administrator@<server> 'cd ~/mathapp_frontend_new && ./deploy.sh'
```

fetch ← `npm ci` ← بررسی نوع‌ها ← build در `dist.next` ← جابه‌جایی بی‌وقفه؛ build قبلی در `dist.prev` می‌ماند و برگشت یک دستور است. نشانی API در `dist/config.js` است و بی‌build عوض می‌شود.

### ۱۳-۳ پیکربندی ( .env سرور )

| متغیر                                                     | اگر نباشد                                                                                                                            |
|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <code>PUBLIC_BASE_URL</code> · <code>FRONTEND_URL</code>  | دامنه‌های فعلی. همه‌ی تنظیمات دامنه (CORS، callback، زرین‌پال، ...) از همین دو ساخته می‌شوند؛ جابه‌جایی دامنه یعنی عوض کردن همین دو. |
| <code>DJANGO_SECRET_KEY</code>                            | یک کلید تصادفی ساخته و در <code>django_secret_key</code> نگه داشته می‌شود (حالت مطلوب).                                              |
| <code>QDRANT_API_KEY</code> · <code>OPENAI_API_KEY</code> | دستیار ۵۰۳ صریح می‌دهد؛ جواب نمایشی فقط در حالت توسعه.                                                                               |

| متغیر                                                                  | اگر نباشد                                                                                                |
|------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| KAVENEGAR_OTP_TEMPLATE · KAVENEGAR_API_KEY                             | درخواست کد ۵۰۳ صریح می‌گیرد. الگو باید SMSVerify بماند.                                                  |
| BAZAAR_* · ZARINPAL_MERCHANT_ID                                        | پرداخت بسته می‌ماند (۵۰۳).                                                                               |
| CACHE_URL                                                              | روی سرور به Redis اختصاصی خودمان اشاره می‌کند. بی آن کش برای هر worker جداست و /api/health هشدار می‌دهد. |
| DB_HOST · DB_PASSWORD · DB_USER · DB_NAME · DB_POOL_MAX_SIZE · DB_PORT | Postgres و اندازه‌ی pool هر worker. بی DB_NAME برنامه SQLite را برمی‌دارد (فقط برای محلی).               |
| TEST_LOGIN_PHONES · HEARTS_ENFORCED                                    | روشن (قلب واقعی؛ 0 خاموشش می‌کند) · خالی؛ ورود آزمایشی فقط با کد یک‌بارمصرفی که ابرکاربر در پنل می‌سازد. |
| AI_DAILY_QUOTA_FREE                                                    | ۲ سوال رایگان در روز (پیش‌فرض).                                                                          |
| DJANGO_DEBUG                                                           | خاموش – و هرگز روی سرور روشن نشود (کد ورود همگانی 0000 را فعال می‌کند).                                  |

مقدارهای `.env` با پیش‌فرض‌های کد ادغام می‌شوند؛ یک `.env` ناقص سرویس را نمی‌خواهاند. شرح کامل در `docs/operations.md` و `.env.example`.

## ۱۳-۴ پایش، بکاپ و قواعد سرور مشترک

### GET /api/health

بی‌ورود فقط «ok/error» می‌گوید. با هدر `X-Internal-Key`: دیتابیس سالم است؟ migration جامانده هست؟ کش و pool اتصال؟ کپی زنده‌ی بکاپ عقب نیست؟ کدام سرویس بیرونی تنظیم است و اعتبار دستیار تمام شده (aiProvider)؟ هشدارهای امنیتی و ترافیک اخیر. ۲۰۰ = سالم، ۵۰۳ = خرابی واقعی.

### بکاپ

هر تغییر دیتابیس لحظه‌ای به سرور دوم (۲۴۷) می‌رود ( `pg_receivewal` از راه تونل محدود)، با یک نسخه‌ی کامل روزانه؛ بازیابی تا هر لحظه از ۷ روز اخیر ممکن است و آزموده شده. `deploy.sh` هم پیش از هر `migrate` یک `pg_dump` می‌گیرد. دفتر رویدادهای خرید هر ۶ ساعت به CSV و لاگ برنامه هر شب فشرده و برای همیشه روی ۲۴۷ نگه داشته می‌شود. رویه: `docs/operations.md`.

## سرور مالِ ما نیست؛ ما یکی از مستأجرهاییم

- مالِ ما فقط پوشه‌های `~/mathapp-v2` و `~/mathapp_frontend_new`، سرویس‌های `zareian-*` و `cluster`، `Postgres` `zareian` است. `gunicorn` خالی سرویس تیم دیگری است.
- هرگز `git clean` یا «پاک کن و از نو clone کن»: `.env`، کلید امضا، لاگ‌ها و بکاپ‌ها در همان پوشه‌اند و در `git` نیستند.
- `nginx` مشترک است: `nginx -t` و بعد `reload`، هرگز `restart`.
- پیش از هر کار پرحجم `df -h /`: پر شدن دیسک `Gitea` و ایمیل شرکت را هم می‌خواهاند.
- `.env` فقط از راه جنگو خوانده شود؛ مقدارش هرگز در چت یا گزارش نیاید.
- قواعد کامل برای هر کسی (یا agentی) که روی سرور کار می‌کند: `CLAUDE.md`، بخش «Working on the servers».

## ۱۴. مقیاس‌پذیری و پیوستن توسعه‌دهنده‌ی تازه

دو معنا دارد: کد چقدر برای تیم بزرگ‌تر آماده است، و سیستم چقدر برای کاربر بیشتر.

### ۱۴-۱ آیا دو سه نفر دیگر می‌توانند به بک‌اند اضافه شوند؟ بله – به این دلایل

| ویژگی             | چرا کار هم‌زمان را ممکن می‌کند                                                                                                     |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------|
| ۱۴ اپ با مرز روشن | هر نفر صاحب چند اپ می‌شود (مثلاً یکی پرداخت و اشتراک، یکی محتوا و گیمیفیکیشن، یکی دستیار)؛ تغییر هر کس کمتر به فایل دیگری می‌خورد. |
| بی‌FK بین اپ‌ها   | تغییر جدول‌های یک اپ migration اپ دیگر را نمی‌شکند؛ هر اپ بعداً قابل جدا شدن به سرویس مستقل است.                                   |
| قرارداد OpenAPI   | فرانت و بک‌اند بی‌جلسه روی یک شکل داده توافق دارند؛ <code>tests_frontend_contract.py</code> تغییر ناخواسته را می‌گیرد.             |
| ۹۷۳ تست خودکار    | تازه‌وارد بی‌ترس تغییر می‌دهد: اگر چیزی را شکست، تست قبل از سرور خبر می‌دهد.                                                       |
| قواعد یک‌جا       | نرخ‌ها، دامنه‌ها و تنظیمات فقط در <code>config/settings.py</code> ؛ پیام خطا، لاگ و هویت یک مسیر مشترک دارند.                      |
| مستندات در مخزن   | <code>docs/</code> + همین سند + Swagger + توضیح «چرا» کنار کد حساس.                                                                |

### ۱۴-۲ دستورالعمل: افزودن یک اندپوینت تازه

۱. اپ درست را انتخاب کنید (یا اپ تازه با `python manage.py startapp` و افزودن به `INSTALLED_APPS` و `config/urls.py`).
۲. مدل در `models.py` – ارجاع به اپ دیگر فقط با `user_id` یا شناسه‌ی عددی – و `python manage.py makemigrations`.
۳. ورودی/خروجی در `serializers.py`، با نام `camelCase`.
۴. ویو در `views.py` با `permission_classes`؛ اگر حساس است `throttle_scope` و ردیف متناظر در `DEFAULT_THROTTLE_RATES`؛ اگر اثر ماندگار دارد، `Idempotency-Key` و `transaction.atomic`؛ تماس شبکه‌ای بیرون از تراکنش.
۵. مسیر در `urls.py` و `@extend_schema` تا در Swagger بیاید.
۶. تست در `tests.py` – از جمله «کاربر دیگری نمی‌تواند».
۷. `python manage.py test` ← `push` ← `deploy.sh` ← `/api/health`.
۸. فرانت: نوع در `src/api/types.ts` از روی پاسخ واقعی، تابع در `src/api/index.ts`.

### ۱۴-۳ مقیاس اجرا: کجا گلوگاه است و بعد چه

وضع فعلی: ۴ worker × ۳۲؛ هر worker حداکثر ۸ اتصال Postgres از pool قرض می‌گیرد و حداکثر ۱۲ نخ به دستیار می‌دهد، تا رگبار سوال نقشه و آزمون را نخواست. توکن stateless است، پس هر تعداد نسخه‌ی دیگر از بک‌اند پشت `nginx` بی‌تغییر کد کار می‌کند.

آزمون بار روی خودِ سرور (بازیکنان شبیه‌سازی‌شده که هر ۱ تا ۵ ثانیه یک درخواست می‌زنند): ۳۰۰ نفر ۸۸ درخواست در ثانیه با میانه‌ی ۰/۱ ثانیه؛ ۶۰۰ نفر ۱۳۷ درخواست با میانه‌ی ۱/۴ ثانیه و بی‌خطا؛ ۱,۰۰۰ نفر به سقفِ ~۱۳۵ درخواست در ثانیه رسید (صدکِ ۹۵ ۱۵ ثانیه). گلوگاهِ CPU ماشینِ مشترک است، نه دیتابیس: دو سرویسِ خرابِ تیم‌های دیگر مدام ری‌استارت می‌شوند و حدودِ ۳ از ۸ هسته را می‌گیرند.

| # | گلوگاه                         | راه‌حلِ ساده‌ی بعدی                                                                 | نیاز       |
|---|--------------------------------|-------------------------------------------------------------------------------------|------------|
| ۱ | CPU ماشینِ مشترک               | رفعِ دو سرویسِ خرابِ دیگران، هسته‌ی بیشتر و ۸ worker (برآورد ~۲۷۰ درخواست در ثانیه) | ادمین سرور |
| ۲ | جوابِ دستیار نخ را نگه می‌دارد | صف کار یا ویوی async برای <code>/api/ai/ask</code>                                  | کد         |
| ۳ | یک سرور برای همه               | چند نسخه‌ی gunicorn پشتِ nginx یا سرورِ دوم؛ فرانت از CDN آروان                     | زیرساخت    |
| ۴ | هزینه‌ی دستیار با کاربرِ زیاد  | کش، سقفِ روزانه (هست)، و سنجشِ مدلِ ارزان‌تر با <code>rag_model_benchmark</code>    | سنجش       |

بهینه‌سازی‌های انجام‌شده: رفعِ N+1 در صفحه‌ی اصلی، کشِ نقشه و جدولِ رتبه، کشِ جوابِ دستیار، pool اتصال (پیش از آن ۱۰۰ کاربرِ هم‌زمان ۲۱٪ خطای ۵۰۰ می‌گرفتند؛ بعد صفر)، ادغامِ چند درخواست در یکی ( `/api/home` ، `/api/user/summary` ، `/api/nodes/{id}/quiz` )، و حجمِ بارِ اولِ فرانت از ۵۰۵ به ۸۵ کیلوبایت.

## ۱۵. تست و کیفیت

| چه                | چطور                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| تست خودکار بک‌اند | ۹۷۳ تست، همه سبز ( <code>python manage.py test --noinput</code> ). روی فایل دیتابیس ( <code>.test-db.sqlite3</code> ) اجرا می‌شوند و روی Postgres هم سبزند. هر باگ پیداشده در بازبینی‌ها یک تست اختصاصی دارد ( <code>tests_audit_fixes.py</code> ، <code>tests_otp_limits.py</code> ، <code>tests_hearts_gate.py</code> ، ... ). هم‌زمانی با دو thread واقعی آزموده می‌شود (اعتبارِ دستیار، آخرین قلب). |
| قرارداد با فرانت  | <code>content/tests_frontend_contract.py</code> و build با TypeScript_strict در فرانت.                                                                                                                                                                                                                                                                                                                  |
| تست روی سایت زنده | با حساب‌های آزمونگر روی سرور واقعی؛ مرورگرِ خودکار (puppeteer) برای واریسِ صفحه‌ها؛ آزمونِ بار با <code>tools/loadtest/</code> .                                                                                                                                                                                                                                                                        |
| کیفیت دستیار      | <code>rag_model_benchmark</code> (مسئله‌های واقعی کتاب از مسیر واقعی، با هزینه‌ی واقعی)، <code>ai_feedback_report</code> (👍) / 👉 به تفکیک نسخه‌ی پرامپت)، واریسِ چشمی نمونه پیش از اعتماد به هر اجرای انبوه.                                                                                                                                                                                            |
| بی‌هزینه آزمون    | <code>tools/fake_openai.py</code> جای GapGPT می‌نشیند.                                                                                                                                                                                                                                                                                                                                                  |
| پایش              | لاگِ JSON هر درخواست + <code>/api/health</code> ؛ محلی با Grafana/Loki ( <code>docs/docker.md</code> ).                                                                                                                                                                                                                                                                                                 |

**فرهنگِ کد:** توضیح‌ها کنارِ کد «چرا» را می‌گویند، نه «درست است». قاعده‌ی مخزن: کامنت ادعاست، نه مدرک – هر ادعا با تست یا اجرا سنجیده می‌شود. چند باگِ واقعی دقیقاً زیرِ کامنتی پیدا شد که کد را «امن» می‌خواند.

## ۱۶. وضعیت فعلی، محدودیت‌ها و تصمیم‌های باز

وضعیت در تاریخ نگارش سند (۲۲ مهر ۱۴۰۵، روی سرور زنده)

| بخش          | وضعیت                    | توضیح                                                                             |
|--------------|--------------------------|-----------------------------------------------------------------------------------|
| اپ و API     | زنده                     | با حساب‌های آزمونگر واری می‌شود؛ هنوز به روی کاربر عمومی باز نشده                 |
| محتوا        | کامل برای سوم تا دوازدهم | ۱۵ کتاب؛ همه‌ی جزیره‌ها سؤال دارند؛ جزیره‌های تمرین آخر هر فصل                    |
| پیامک ورود   | کار می‌کند               | الگوی SMSVerify؛ هر پیامک هزینه دارد؛ سقف روزانه‌ی کل سامانه ۲,۰۰۰                |
| دستیار + شکل | کار می‌کند               | فقط ریاضی؛ ~یک شکل از پنج ایراد دارد                                              |
| زرین پال     | نیمه                     | درخواست پرداخت پذیرفته می‌شود؛ پرداخت واقعی تا انتها آزموده نشده                  |
| کافه‌بازار   | منتظر APK تازه           | سه اشتراک (۱۰۳۴۲-۱۰۳۴۴) روی پلن‌ها؛ کد اپ آماده ولی build و خرید واقعی انجام نشده |
| اشتراک و قلب | زنده                     | فصل اول هر کتاب رایگان؛ قلب صفر بازی را متوقف می‌کند                              |
| بکاپ         | خودکار                   | کپی زنده روی سرور دوم، بازیابی آزموده شده                                         |
| ظرفیت        | سنجیده                   | ۶۰۰ بازیکن هم‌زمان بی‌خطا؛ سقف CPU ماشین مشترک                                    |
| پیام گروهی   | پایه                     | ذخیره و هدف‌گیری هست؛ ارسال خودکار نه                                             |

### تصمیم‌هایی که با مدیریت است، نه با کد

| موضوع                           | چرا تصمیم لازم است                                                                        |
|---------------------------------|-------------------------------------------------------------------------------------------|
| مانع در برابر پر کردن سقف پیامک | CAPTCHA تجربه‌ی ورود را سخت‌تر می‌کند؛ بی‌آن یک مهاجم ورود را برای همه می‌بندد.           |
| سقف دستیار و سهمیه‌ی رایگان     | بعد از دو هفته مصرف واقعی بازبینی شود؛ سود در docs/ai-cost-and-guardrails.md حساب شده.    |
| یک فصل رایگان برای هر کتاب      | تعبیر ما از «یک فصل رایگان» برای پایه‌های چندکتابی؛ تأیید نهایی با مدیریت.                |
| سیاست بازپرداخت                 | الان روزهای اشتراک همان سفارش پس گرفته می‌شود؛ پول از پنل درگاه.                          |
| پرداخت درگاه داخل اپ بازار      | بازار پرداخت محتوای دیجیتال بیرون از پرداخت خودش را محدود می‌کند؛ پیش از انتشار روشن شود. |
| CPU و دسترسی SSH                | کار ادمین سرور.                                                                           |
| پیشرفت با هر نمره               | در حال حاضر جزیره با هر نمره‌ای تمام می‌شود؛ اگر حد قبولی لازم است، یک تصمیم آموزشی است.  |

## کارِ بعدی، به ترتیبِ ارزش

1. یک پرداختِ واقعیِ کامل با زرین‌پال، و ساختِ APK تازه و یک خریدِ واقعی از کافه‌بازار.
2. بستنِ مواردِ باز در `docs/security-before-production.md` (پیامک، پنلِ ادمین، دسترسیِ سرور).
3. بازبینیِ سؤال‌های جزیره‌های تمرین با چشمِ معلم، به کمکِ رأی و گزارشِ دانش‌آموزان.
4. رفعِ گلوگاهِ CPU با ادمینِ سرور پیش از هر لانچِ بزرگ.
5. صفِ کار برای دستیار؛ سنجشِ شکل روی نمونه‌ی بزرگ‌تر.

| واژه               | معنی در این پروژه                                                                      |
|--------------------|----------------------------------------------------------------------------------------|
| API / اندپوینت     | نشانی‌ای که اپ برای گرفتن یا فرستادن داده صدا می‌زند، مثل <code>GET /api/home</code> . |
| اپ جنگو            | یک پوشه‌ی مستقل با مدل، ویو و تست خودش؛ یک مسئولیت.                                    |
| مدل / migration    | تعریف یک جدول در پایتون / اسکرپتی که تغییر جدول را روی دیتابیس اعمال می‌کند.           |
| ویو / serializer   | کدی که درخواست را جواب می‌دهد / کدی که شکل ورودی و خروجی را تعریف و واری می‌کند.       |
| middleware         | لایه‌ای که همه‌ی درخواست‌ها پیش و پس از ویو از آن رد می‌شوند.                          |
| توکن               | «کارت شناسایی» امضاشده که اپ با هر درخواست می‌فرستد.                                   |
| throttle           | سقف تعداد درخواست در زمان.                                                             |
| Idempotency-Key    | کلیدی که باعث می‌شود تکرار یک درخواست دو بار اثر نکند.                                 |
| تراکنش اتمیک       | چند نوشتن که یا همه انجام می‌شوند یا هیچ‌کدام.                                         |
| RAG                | اول پیدا کردن متن مرتبط از کتاب، بعد نوشتن جواب بر پایه‌ی آن.                          |
| embedding / بردار  | تبدیل متن به عددی که معنا را نشان می‌دهند تا متن‌های هم‌معنا کنار هم بیفتند.           |
| chunk / collection | یک تکه‌ی کتاب (یک تیپ‌مسئله) / همه‌ی تکه‌های یک کتاب در Qdrant.                        |
| SSE (استریم)       | فرستادن جواب تکه‌تکه، تا کاربر پیش از تمام شدن چیزی ببیند.                             |
| seed               | پر کردن دیتابیس از فایل‌های محتوای داخل مخزن.                                          |
| fail-closed        | اگر تنظیمات نبود، بسته بماند (خطای صریح)، نه اینکه وانمود کند کار کرد.                 |

## ب . دستوره‌ای پرکاربرد

```
# توسعه‌ی محلی
python3 -m venv venv && source venv/bin/activate && pip install -r requirements.txt
cp .env.example .env # برای محلی DJANGO_DEBUG=1
python manage.py migrate && python manage.py seed_content && python manage.py runserver
python manage.py test --noinput # کل تست‌ها
python manage.py spectacular --file schema.yaml # قرارداد API

# محتوا و دستیار
python manage.py seed_content --export --only questions
python manage.py rag_catalog # collection های Qdrant (فقط خواندن)
python manage.py rag_ingest_book --grade نهم --plan
python manage.py rag_verify_solutions
python manage.py generate_questions_from_chunks --subject ریاضی --grade نهم --export /tmp/b
python manage.py ai_feedback_report
python manage.py rebuild_leaderboard
python manage.py payment_reconcile --days 3

# فرانت
npm install && npm run dev # http://localhost:5173
npm run build # type-check + build در dist/
```

## ج . نقشه‌ی مستندات موجود

| سند                                | برای                                                                         |
|------------------------------------|------------------------------------------------------------------------------|
| README.md (هر دو مخزن)             | در ورودی و راه‌اندازی                                                        |
| docs/architecture.md               | تصمیم‌های معماری و دلیلشان، با جزئیات                                        |
| docs/operations.md                 | انتشار، متغیرها، بکاپ، جابه‌جایی دامنه، ورود تیم                             |
| docs/security-before-production.md | موارد امنیتی باز با خطر و راه بستن، و حفاظ‌های موجود                         |
| docs/content-authoring.md          | ویرایش محتوا و ساخت انبوه سؤال                                               |
| docs/extraction-playbook.md        | خواندن کتاب با مدل بینایی: هزینه و درس‌ها                                    |
| docs/ai-cost-and-guardrails.md     | هزینه‌ی هر سوال دستیار، سود هر اشتراک و قاعده‌ی خرج، محافظ «فقط ریاضی»       |
| docs/model-prices.md               | دفتر قیمت اندازه‌گیری‌شده‌ی مدل‌ها، با تاریخ؛ اگر قیمت عوض شد ردیف اضافه شود |
| data/textbooks/README.md           | افزودن کتاب تازه                                                             |
| tools/loadtest/README.md           | آزمون بار، روی لپ‌تاپ و روی سرور مشترک                                       |
| CLAUDE.md                          | راهنمای agent‌های کدنویس و قواعد کار روی سرور مشترک                          |
| /api/docs/                         | قرارداد زنده و قابل امتحان API                                               |

تدریسینو . مستند فنی . علیرضا زارعیان . مهر ۱۴۰۵  
عددها از کد، اسکیمای API، دیتابیس و سنجش‌های ثبت‌شده در مخزن استخراج شده‌اند.