

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

گروه مهندسی نرم افزار

## پایان نامه کارشناسی رشته مهندسی کامپیوتر

عنوان پایان نامه

طراحی و پیاده سازی سامانه آنلاین آموزش تایپ

استاد راهنما:

دکتر مجتبی مهدوی

دکتر احسان مهدوی

پژوهشگر:

سپهر شیرازی

محمد حسین ملکی

شهریور ۱۴۰۴



دانشگاه اصفهان  
دانشکده مهندسی کامپیوتر  
گروه مهندسی نرم افزار

## پروژه کارشناسی رشته‌ی مهندسی کامپیوتر گرایش نرم افزار / فناوری اطلاعات

آقای سپهر شیرازی / محمد حسین ملکی

تحت عنوان

طراحی و پیاده سازی سامانه آنلاین آموزش تایپ

در تاریخ / / ۱۴۰۴ توسط هیأت داوران زیر بررسی و با نمره به تصویب نهایی رسید.

۱- استاد راهنمای پروژه:

دکتر

امضا

۲- استاد داور :

دکتر

امضا

امضای مدیر گروه

## تشکر و قدردانی

در ابتدا از پدر و مادر عزیزم که در این مسیر همواره حامی و پشتیبان من بودند و سپس از استادان راهنمای عزیز دکترا احسان و مجتبی مهدوی که در هر شرایط از کمک به من فروگذار نکرده اند کمال تشکر و قدردانی را دارم.

سپهر شیرازی

محمد حسین ملکی

تقدیم به

همه ی عزیزانی که در سخت ترین لحظات زندگی،

همیشه یاور من بوده اند.

## چکیده:

در دنیای امروز، تسلط بر مهارت تایپ ده انگشتی برای افزایش بهره‌وری در فعالیت‌های گوناگون ضروری است. با وجود پلتفرم‌های متعدد آنلاین برای آموزش و تمرین تایپ، بسیاری از آن‌ها فاقد بخش آموزش ساختاریافته برای یادگیری حروف به صورت گام به گام هستند یا در زمینه گزارش‌گیری جامع و امکانات شخصی‌سازی محدودیت دارند.

هدف این پروژه، طراحی و پیاده‌سازی یک سامانه جامع و کاربرپسند برای آموزش و تمرین آنلاین تایپ ده انگشتی است که این نیازها را برطرف کند. این سامانه به صورت یک وب‌سایت پویا با استفاده از فناوری‌های نوین توسعه یافته است؛ Vue.js برای ساخت رابط کاربری منعطف و واکنش‌گرا، Fastify بر بستر Node.js برای پیاده‌سازی API‌های پرسرعت و مدیریت سمت سرور، و Prisma برای تسهیل ارتباط با پایگاه داده PostgreSQL و مدیریت داده‌ها.

قابلیت‌های اصلی سامانه شامل ارائه آموزش درس به درس تایپ حروف همراه با راهنمای تصویری انگشتان، امکان انجام تست سرعت تایپ با انتخاب متن‌های متنوع و سطوح دشواری مختلف (مبتدی، متوسط، پیشرفته)، گزارش‌گیری دقیق و کامل پس از هر تست (شامل سرعت، دقت و تعداد اشتباهات)، نمایش نمودار روند پیشرفت کاربر در طول زمان، و امکانات شخصی‌سازی گسترده رابط کاربری مانند انتخاب تم‌های رنگی و تنظیم پارامترهای نمایش متن تست (اندازه فونت و طول پاراگراف) می‌باشد. همچنین، سیستم مدیریت کاربران با امکان ثبت‌نام، ورود امن (با احراز هویت) و بازیابی رمز عبور برای ذخیره و پیگیری پیشرفت کاربران در نظر گرفته شده است.

تفاوت‌های کلیدی این پروژه با نمونه‌های مشابه، به خصوص پلتفرم‌های ایرانی، در ارائه ماژول آموزش گام‌به‌گام جامع با راهنمای بصری، قابلیت‌های گزارش‌گیری پیشرفته و نمایش نمودار پیشرفت، امکانات شخصی‌سازی وسیع‌تر، و بهره‌گیری از معماری و تکنولوژی‌های مدرن برای ارائه تجربه‌ای کارآمدتر و با رابط کاربری جذاب‌تر است. نتیجه نهایی این پروژه، ابزاری مؤثر برای کمک به کاربران در تمام سطوح جهت یادگیری و بهبود مهارت تایپ به شیوه‌ای ساختاریافته، تعاملی و قابل پیگیری است.

**واژگان کلیدی:** پروژه کارشناسی، آموزش تایپ آنلاین، تایپ ده انگشتی، سامانه وب آموزشی

## فهرست مطالب

عنوان	صفحه
فصل اول پیشگفتار.....	۸
۱-۱- مقدمه .....	۸
۱-۲- بیان مسئله: شکاف موجود در سامانه‌های آموزش تایپ .....	۸
۱-۳- هدف پروژه .....	۹
۱-۴- کاربردهای پروژه .....	۱۰
۱-۵- دامنه پروژه .....	۱۲
۱-۶- ساختار پایان‌نامه .....	۱۲
فصل دوم: مفاهیم .....	۱۴
۱-۲- مقدمه .....	۱۴
۲-۲- مبانی نظری تایپ ده‌انگشتی .....	۱۴
۱-۲-۲- اصول و اهمیت تایپ لمسی .....	۱۴
۲-۲-۲- استانداردهای ارزیابی (WPM و دقت) .....	۱۶
۳-۲- پیشینه تحقیق: بررسی سامانه‌های موجود .....	۱۷
۱-۳-۲- بررسی نمونه‌های خارجی .....	۱۹
۲-۳-۲- تحلیل نمونه‌های ایرانی .....	۲۰
۳-۳-۲- جمع‌بندی و تحلیل شکاف‌های موجود .....	۲۰
۴-۲- فناوری‌های مورد استفاده در پروژه .....	۲۰
۱-۴-۲- فناوری‌های سمت کاربر (Frontend) .....	۲۱
۲-۴-۲- فناوری‌های سمت سرور (Backend) .....	۲۲
۵-۲- جمع‌بندی .....	۲۵
فصل سوم: شرح پروژه .....	۲۶
۱-۳- مقدمه .....	۲۶
۲-۳- شرح نیازمندی‌ها .....	۲۶
۱-۲-۳- نیازمندی‌های عملکردی .....	۲۷
۲-۲-۳- نیازمندی‌های غیرعملکردی .....	۳۱
۳-۳- معماری سامانه .....	۳۲

## فهرست مطالب

عنوان	صفحه
۳-۳-۱- نمای کلی معماری (معماری کلاینت-سرور) .....	۳۲
۳-۳-۲- معماری سمت کاربر (Frontend) .....	۳۴
۳-۳-۳- معماری سمت سرور (Backend) .....	۳۴
۳-۴- طراحی پایگاه داده .....	۳۶
۳-۴-۱- مدل موجودیت-رابطه (ER) .....	۳۷
۳-۴-۲- طراحی شمای جداول (Table Schema) .....	۳۸
۳-۵- طراحی رابط کاربری (UI/UX Design) .....	۳۹
۳-۵-۱- فلسفه و رویکرد طراحی .....	۴۱
۳-۵-۲- عناصر طراحی بصری .....	۴۲
۳-۶- طراحی API .....	۴۳
۳-۶-۱- اصول طراحی (استفاده از اصول RESTful) .....	۴۴
۳-۶-۲- تشریح مسیرهای اصلی (Endpoints) .....	۴۴
۳-۷- جمع‌بندی .....	۴۷
<b>فصل چهارم: پیاده‌سازی سامانه .....</b>	<b>۴۸</b>
۴-۱- مقدمه .....	۴۸
۴-۲- پیاده‌سازی بخش فرانت‌اند (Vue.js) .....	۴۹
۴-۲-۱- سازوکار واکنشی و نقش DOM مجازی در Vue.js .....	۴۹
۴-۲-۲- ساختار پروژه فرانت‌اند و معماری مبتنی بر کامپوننت .....	۵۰
۴-۲-۳- ساختار پروژه بک‌اند و ماژول‌های اصلی .....	۵۴
۴-۳- پیاده‌سازی بخش بک‌اند (Fastify) .....	۵۴
۴-۳-۱- ساختار پروژه بک‌اند و ماژول‌های اصلی .....	۵۴
۴-۳-۲- پیاده‌سازی API‌ها و منطق سمت سرور .....	۵۵
۴-۳-۳- پیاده‌سازی احراز هویت و امنیت .....	۵۵
۴-۴- تست و اعتبارسنجی سامانه .....	۵۶
۴-۴-۱- تست واحد در فرانت‌اند با Vitest .....	۵۶
۴-۴-۲- تست واحد در بک‌اند با Jest .....	۵۷



## فهرست مطالب

عنوان	صفحه
۵-۴- چالش‌های پیاده‌سازی و راهکارها .....	۵۷
۶-۴- جمع‌بندی .....	۵۸
<b>فصل پنجم: ارزیابی و نتایج .....</b>	<b>۵۹</b>
۱-۵- مقدمه .....	۵۹
۲-۵- روش ارزیابی سامانه .....	۵۹
۱-۲-۵- ارزیابی فنی و عملکردی .....	۶۰
۲-۲-۵- ارزیابی کاربردپذیری .....	۶۲
۳-۵- نتایج ارزیابی .....	۶۳
۱-۳-۵- نتایج عملکرد فنی .....	۶۴
۲-۳-۵- تحلیل نتایج کاربردپذیری و بازخوردهای کاربران .....	۶۵
۳-۳-۵- نمایش محصول نهایی .....	۶۷
۴-۵- طرح تجاری و امکان‌سنجی .....	۷۴
۱-۴-۵- مدل کسب‌وکار .....	۷۵
۲-۴-۵- تحلیل بازار هدف .....	۷۵
۳-۴-۵- امکان‌سنجی اولیه .....	۷۶
۵-۵- جمع‌بندی .....	۷۶
<b>فصل ششم: نتیجه‌گیری و پیشنهادات .....</b>	<b>۷۷</b>
۱-۶- نتیجه‌گیری کلی .....	۷۷
۲-۶- محدودیت‌های پروژه .....	۷۸
۳-۶- کارهای آتی و پیشنهادات برای توسعه آینده .....	۸۰
<b>پیوست .....</b>	<b>۸۲</b>
منابع: .....	۸۳

## فهرست شکل‌ها

### عنوان

### صفحه

شکل ۱-۲: نمای شماتیک از قرارگیری صحیح انگشتان روی کیبورد در حالت "ردیف پایه" .....	۱۶
شکل ۲-۲: نمای کلی رابط کاربری وبسایت 10fastfingers.com .....	۱۸
شکل ۲-۳: نمای کلی رابط کاربری وبسایت typing.com .....	۱۹
شکل ۱-۳: نمودار موارد کاربرد (Use Case) سامانه .....	۲۹
شکل ۲-۳: نمودار کلاس‌های اصلی سیستم بر اساس مدل‌های داده .....	۳۰
شکل ۳-۳: نمودار توالی (Sequence) برای فرآیند انجام تست تایپ .....	۳۱
شکل ۴-۳: نمودار معماری کلی سامانه (کلاینت-سرور) .....	۳۳
شکل ۵-۳: نمودار فعالیت برای جریان کاری انجام تست تایپ .....	۳۶
شکل ۶-۳: نمودار موجودیت-رابطه (ERD) پایگاه داده سامانه .....	۳۸
شکل ۷-۳: نمودار اجزای (Component) نرم‌افزار در معماری سامانه .....	۴۰
شکل ۸-۳: نمونه‌ای از وزن‌های مختلف فونت Vazirmatn .....	۴۲
شکل ۹-۳: پالت رنگی اصلی استفاده شده در سامانه .....	۴۳
شکل ۱۰-۳: نمودار استقرار (Deployment) سامانه .....	۴۶
شکل ۱۱-۳: نمودار ماشین حالات (State Machine) برای فرآیند تست تایپ .....	۴۷
شکل ۱-۴: معماری واکنشی Vue.js و نقش DOM مجازی .....	۴۹
شکل ۲-۴: نمودار مفهومی معماری کامپوننت‌ها بر اساس متدولوژی Atomic Design .....	۵۰
شکل ۳-۴: ساختار پوشه‌ها و دایرکتوری‌های پروژه فرانت‌اند .....	۵۱
شکل ۱-۵: نمودار میانگین امتیاز سادگی وظایف از دید کاربران .....	۶۶
شکل ۲-۵: صفحه ثبت‌نام کاربر در سامانه .....	۶۷
شکل ۳-۵: صفحه اصلی بخش آموزش با نمایش درس‌ها .....	۶۸
شکل ۴-۵: محیط تعاملی یک درس آموزشی با کیبورد راهنما .....	۶۸
شکل ۵-۵: صفحه انجام تست سرعت با قابلیت‌های شخصی‌سازی .....	۶۹
شکل ۶-۵: جدول رتبه‌بندی کاربران بر اساس بیشترین سرعت ثبت‌شده .....	۶۹
شکل ۷-۵: صفحه دستاوردهای سامانه برای ایجاد انگیزه در کاربر .....	۷۰
شکل ۸-۵: صفحه نمایش نتیجه نهایی یک آزمون .....	۷۱

## فهرست شکل‌ها

### عنوان

### صفحه

شکل ۵-۹: صفحه گزارش پیشرفت کاربر با نمایش نمودار و تاریخچه نتایج.....	۷۱
شکل ۵-۱۰: صفحه ویرایش اطلاعات پروفایل کاربری .....	۷۲
شکل ۵-۱۱: لوگوی رسمی سامانه آموزش تایپ .....	۷۲
شکل ۵-۱۲: داشبورد اصلی پنل مدیریت سامانه .....	۷۳
شکل ۵-۱۳: صفحه مدیریت محتوای آموزشی (درس‌ها) در پنل ادمین .....	۷۳
شکل ۵-۱۴: مودال افزودن یک پاراگراف جدید در پنل ادمین .....	۷۳

## فهرست جدول‌ها

### عنوان

### صفحه

جدول ۳-۱: لیست مسیرهای (Endpoints) اصلی API سامانه .....	۴۴
جدول ۵-۱: نتایج ارزیابی عملکرد فرانت‌اند با ابزار Lighthouse .....	۶۵

API	Application Programming Interface
CMS	Content Management System
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DOM	Document Object Model
ERD	Entity-Relationship Diagram
HTTP	Hypertext Transfer Protocol
JWT	JSON Web Token
LCP	Largest Contentful Paint
ORM	Object-Relational Mapping
OTP	One-Time Password
RDBMS	Relational Database Management System
REST	Representational State Transfer
SEO	Search Engine Optimization
SPA	Single-Page Application
SQL	Structured Query Language
SSR	Server-Side Rendering
UI	User Interface
UX	User Experience
WPM	Words Per Minute
XP	Experience Points

## فصل اول

### پیشگفتار

#### ۱-۱- مقدمه

در جهان معاصر، با دیجیتالی شدن فزاینده‌ی فرآیندها، مهارت کار با کامپیوتر از یک مزیت رقابتی به یک ضرورت انکارناپذیر تبدیل شده است. در این میان، تایپ ده‌انگشتی<sup>۱</sup> یا تایپ صحیح و سریع، به عنوان یکی از بنیادی‌ترین و کاربردی‌ترین مهارت‌های دیجیتال<sup>۲</sup>، نقشی حیاتی در افزایش بهره‌وری<sup>۳</sup> و کارایی افراد در حوزه‌های مختلف ایفا می‌کند. این مهارت دیگر محدود به مشاغل منشی‌گری یا تایپیست‌ها نیست؛ بلکه دانشجویانی که در حال نگارش مقالات و پایان‌نامه‌های خود هستند، برنامه‌نویسانی که روزانه هزاران خط کد می‌نویسند، محققانی که با حجم عظیمی از داده‌ها سروکار دارند، و کارمندانی که وظایفشان با مکاتبات اداری و گزارش‌نویسی گره خورده است، همگی می‌توانند از مزایای آن بهره‌مند شوند. تسلط بر تایپ ده‌انگشتی نه تنها منجر به صرفه‌جویی چشمگیر در زمان می‌شود، بلکه با کاهش خطاها و افزایش تمرکز بر محتوا به جای فرآیند نوشتن، کیفیت کار را نیز به شکل قابل توجهی ارتقا می‌دهد. از این رو، وجود ابزارهای کارآمد، در دسترس و جذاب برای یادگیری این مهارت از اهمیت بالایی برخوردار است.

#### ۱-۲- بیان مسئله<sup>۴</sup>: شکاف موجود در سامانه‌های آموزش تایپ

با وجود اهمیت ذکر شده، و با وجود اینکه ابزارها و وبسایت‌های متعددی برای آموزش و تمرین تایپ در دسترس هستند، بررسی دقیق آن‌ها نشان‌دهنده‌ی یک شکاف عملکردی و تجربه‌ی کاربری<sup>۵</sup> است. بسیاری از

---

<sup>۱</sup> Ten-finger Typing

<sup>۲</sup> Digital Skills

<sup>۳</sup> Productivity

<sup>۴</sup> Problem Statement

<sup>۵</sup> User Experience (UX)

این پلتفرم‌ها در ارائه یک راهکار جامع و یکپارچه که تمام نیازهای کاربر را از سطح مبتدی تا پیشرفته پوشش دهد، دارای کاستی‌هایی هستند. این مسئله به ویژه برای کاربران فارسی‌زبان که به دنبال محتوای آموزشی استاندارد و رابط کاربری بهینه شده هستند، بیشتر احساس می‌شود.

تحلیل سامانه‌های رقیب، این چالش‌ها را به وضوح نشان می‌دهد:

- **نمونه‌های خارجی:** وبسایت‌های محبوبی مانند **10fastfingers**، تمرکز اصلی خود را بر روی بخش تست سرعت<sup>۱</sup> قرار داده‌اند و ابزاری قدرتمند برای سنجش مهارت فعلی کاربران محسوب می‌شوند؛ اما این پلتفرم فاقد یک بخش آموزشی مدون و جداگانه برای یادگیری گام‌به‌گام حروف و اصول اولیه تایپ است. این نقص باعث می‌شود کاربری که هیچ پیش‌زمینه‌ای ندارد، نتواند مسیر یادگیری خود را از صفر در این سایت آغاز کند. از سوی دیگر، وبسایت **typing.com** که بخش آموزشی را ارائه می‌دهد، در بخش گزارش‌گیری<sup>۲</sup> و ارائه آمار تحلیلی از روند پیشرفت کاربر، عملکرد مطلوبی ندارد. این ضعف باعث می‌شود کاربر نتواند نقاط ضعف و قوت خود را به درستی شناسایی کرده و برای بهبود آن‌ها برنامه‌ریزی کند.

### ۱-۳- هدف پروژه

در راستای حل مسئله بیان شده و برای تحقق چشم‌انداز نهایی پروژه، اهداف اصلی و کاربردی زیر به صورت دقیق تعریف شده‌اند. این اهداف نقشه راه توسعه سامانه را تشکیل داده و هر یک به دنبال افزودن ارزشی مشخص به تجربه کاربر هستند:

- **ایجاد یک بستر آموزشی ساختاریافته و تعاملی:**
  - طراحی بخش‌های جداگانه برای آموزش گام‌به‌گام و درس‌به‌درس تمامی حروف.
  - ارائه تمرین‌های عملی و راهنمای آموزشی تصویری برای نمایش حالت صحیح قرارگیری انگشت‌ها جهت تسهیل یادگیری.
- **توسعه یک مازول پیشرفته و قابل تنظیم برای تست سرعت:**
  - طراحی تست‌های سرعت تایپ با استفاده از پاراگراف‌های متنوع و جذاب.
  - فراهم کردن امکان انتخاب متن‌ها با سطوح دشواری متفاوت (مبتدی، متوسط، پیشرفته).

<sup>۱</sup> Speed Test

<sup>۲</sup> Reporting

- ارائه گزارش‌های کامل و فوری پس از هر تست، شامل معیارهای دقیق مانند تعداد کلمه در دقیقه (WPM)<sup>۱</sup>، درصد دقت<sup>۲</sup>، تعداد اشتباهات و زمان صرف شده.
- **پیاده‌سازی یک سیستم گزارش‌گیری جامع و بصری:**
  - ایجاد قابلیت برای کاربران تا بتوانند از روند پیشرفت خود در تایپ گزارش‌های تحلیلی دریافت کنند.
  - نمایش پیشرفت سرعت و دقت تایپ در طول زمان با استفاده از نمودارهای بصری و قابل فهم.
- **فراهم آوردن سطح بالایی از شخصی‌سازی برای بهبود تجربه کاربری:**
  - امکان انتخاب تم سایت از میان ۱۵ رنگ دلخواه جهت ایجاد محیطی جذاب برای کاربر.
  - قابلیت تغییر اندازه فونت پاراگراف‌های تست برای راحتی بیشتر، به ویژه برای افرادی که مشکلات بینایی دارند.
  - امکان انتخاب طول پاراگراف تست در سه گزینه کوتاه، متوسط و بلند.
- **ایجاد یک سیستم کاربری امن و پایدار:**
  - پیاده‌سازی صفحه ثبت‌نام و ورود کاربر جهت ذخیره و نگهداری سوابق، رکوردها و درس‌های گذرانده شده.
  - تضمین امنیت در تبادل اطلاعات با سرور از طریق احراز هویت درخواست‌ها با استفاده از توکن<sup>۳</sup> (JWT).
  - افزایش امنیت حساب کاربری با پیاده‌سازی قابلیت بازیابی رمز عبور از طریق رمز یکبار مصرف<sup>۴</sup> (OTP).
- **تضمین دسترسی گسترده و پشتیبانی دوزبانه:**
  - پشتیبانی کامل از دو زبان فارسی و انگلیسی در تمام بخش‌های سامانه تا کاربران مختلف بتوانند از آن استفاده کنند.

#### ۴-۱- کاربردهای پروژه

سامانه آموزش تایپ ده انگشتی به عنوان یک ابزار آموزشی و تمرینی، دارای کاربردهای متنوعی برای طیف گسترده‌ای از مخاطبان، از افراد مبتدی تا کاربران حرفه‌ای است. این پلتفرم به گونه‌ای طراحی شده است که نه تنها یک ابزار سنجش، بلکه یک مسیر یادگیری کامل باشد. در ادامه، کاربردهای اصلی این پروژه تشریح می‌شود:

<sup>1</sup> Words Per Minute

<sup>2</sup> Accuracy

<sup>3</sup> JSON Web Token

<sup>4</sup> One-Time Password



- ابزاری بنیادین برای یادگیری از صفر:

کاربرانی که هیچ آشنایی قبلی با اصول تایپ صحیح ندارند، می‌توانند از این سامانه به عنوان نقطه شروع استفاده کنند. بخش آموزش درس به درس و تعاملی حروف، به این دسته از کاربران کمک می‌کند تا به صورت ساختاریافته و گام به گام، موقعیت صحیح هر کلید و انگشت متناظر با آن را فرا بگیرند<sup>۱</sup>. این ویژگی، سامانه را به یک کلاس درس مجازی برای مبتدیان تبدیل می‌کند.

- پلتفرمی برای تمرین و افزایش مهارت:

کاربرانی که با اصول اولیه آشنا هستند اما قصد دارند سرعت و دقت خود را افزایش دهند، می‌توانند از ماژول<sup>۱</sup> تست سرعت استفاده کنند. امکان انتخاب متن‌های متنوع با سطوح دشواری مختلف (مبتدی، متوسط و پیشرفته) و قابلیت تنظیم طول پاراگراف، به کاربران اجازه می‌دهد تا تمرینات خود را متناسب با سطح و نیازشان شخصی‌سازی کرده و خود را به چالش بکشند.

- ابزاری برای ارزیابی و رصد پیشرفت:

یکی از کاربردهای کلیدی این پروژه، ارائه بازخوردهای دقیق و قابل تحلیل به کاربر است. پس از هر تست، گزارش کاملی از عملکرد شامل سرعت (WPM)، دقت، تعداد اشتباهات و زمان نمایش داده می‌شود. علاوه بر این، کاربران با مراجعه به پروفایل شخصی خود می‌توانند روند پیشرفت خود را در طول زمان از طریق نمودارها مشاهده کنند و نقاط ضعف خود را برای تمرین بیشتر شناسایی نمایند.

- محیطی شخصی‌سازی شده و انگیزشی:

برای جلوگیری از خستگی و افزایش تعامل کاربر، امکانات شخصی‌سازی متعددی در نظر گرفته شده است. کاربران می‌توانند تم سایت را از میان ۱۵ رنگ مختلف انتخاب کرده و اندازه فونت را تغییر دهند. همچنین، جدول رتبه‌بندی<sup>۲</sup> با نمایش برترین رکوردها، یک فضای رقابتی سالم ایجاد کرده و به کاربران انگیزه بیشتری برای بهبود مهارت‌هایشان می‌دهد.

- پلتفرمی<sup>۳</sup> دوزبانه برای مخاطبان گسترده:

این سامانه با پشتیبانی کامل از دو زبان فارسی و انگلیسی، نه تنها برای فارسی‌زبانان، بلکه برای کاربرانی که قصد تمرین و تقویت مهارت تایپ انگلیسی خود را دارند نیز قابل استفاده است و دایره مخاطبان گسترده‌تری را پوشش می‌دهد.

- ایجاد یک پنل مدیریت<sup>۴</sup> جامع:

طراحی و پیاده‌سازی یک بخش مدیریتی امن برای کنترل کامل بر محتوای سامانه (درس‌ها و پاراگراف‌ها)

<sup>1</sup> Module

<sup>2</sup> Ranking Table

<sup>3</sup> Platform

<sup>4</sup> Admin Panel

و مدیریت کاربران، جهت تضمین پایداری و سهولت در نگهداری بلندمدت پروژه.

#### ۱-۵- دامنه پروژه<sup>۱</sup> (محدوده و قابلیت‌ها)

به منظور مدیریت مؤثر منابع<sup>۲</sup> و تمرکز بر ارائه یک محصول هسته‌ای کارآمد، تعیین دقیق دامنه پروژه امری ضروری است. این بخش مرزهای پروژه را با تشریح قابلیت‌های اصلی که پیاده‌سازی می‌شوند (درون محدوده) و همچنین مواردی که به صورت عامدانه خارج از محدوده فعلی قرار گرفته‌اند، مشخص می‌کند.

#### قابلیت‌های اصلی سامانه (درون محدوده)

دامنه این پروژه بر طراحی و پیاده‌سازی یک وب اپلیکیشن<sup>۳</sup> جامع متمرکز است که شامل سه ماژول کلیدی و یکپارچه است:

۱. **ماژول مدیریت کامل حساب کاربری:** این بخش، زیربنای یک تجربه شخصی‌سازی شده را فراهم می‌کند و شامل فرآیندهای کامل ثبت‌نام، ورود امن با استفاده از توکن JWT، قابلیت ویرایش پروفایل و بازیابی رمز عبور از طریق OTP می‌باشد.
۲. **ماژول آموزش و تست سرعت:** این ماژول، هسته اصلی سامانه است و یک مسیر یادگیری کامل را پوشش می‌دهد؛ از درس‌های تعاملی و گام‌به‌گام برای مبتدیان گرفته تا آزمون‌های پیشرفته و کاملاً قابل تنظیم (از نظر سطح دشواری، طول متن و اندازه فونت) برای کاربران حرفه‌ای. این بخش همچنین شامل ارائه گزارش‌های تحلیلی فوری و نمودارهای پیشرفت بصری است.
۳. **پنل مدیریت محتوا و کاربران:** برای تضمین پایداری و سهولت در نگهداری بلندمدت، یک پنل مدیریتی امن طراحی شده است که به مدیران اجازه می‌دهد محتوای آموزشی (درس‌ها و پاراگراف‌ها) را مدیریت کرده و بر فعالیت کاربران نظارت داشته باشند.

#### ۱-۶- ساختار پایانه

ساختار این پایان نامه به این صورت است:

- **فصل دوم،** به بررسی مبانی نظری، پیشینه تحقیق<sup>۴</sup> و فناوری‌های به کار رفته در پروژه اختصاص دارد. در این فصل، ضمن مرور مفاهیم کلیدی مرتبط با تایپ ده‌انگشتی، به تحلیل و مقایسه

<sup>1</sup> Project Scope

<sup>2</sup> Resources

<sup>3</sup> Web Application

<sup>4</sup> Literature Review

سامانه‌های مشابه داخلی و خارجی پرداخته شده و در نهایت، فناوری‌های مورد استفاده در دو بخش فرانت‌اند<sup>۱</sup> و بک‌اند<sup>۲</sup> معرفی و بررسی می‌شوند.

- **فصل سوم، فرآیند تحلیل و طراحی سامانه** را به تفصیل تشریح می‌کند. این فصل با تعریف دقیق نیازمندی‌های عملکردی<sup>۳</sup> و نیازمندی‌های غیرعملکردی<sup>۴</sup> آغاز شده و سپس به تشریح معماری کلی سامانه، طراحی ساختار پایگاه داده، طراحی رابط کاربری (UI)<sup>۵</sup> و همچنین طراحی واسطه‌های برنامه‌نویسی کاربردی (API)<sup>۶</sup> می‌پردازد.
- **فصل چهارم، جزئیات کامل مراحل پیاده‌سازی سامانه** را پوشش می‌دهد. در این فصل، ساختار پروژه‌های فرانت‌اند و بک‌اند، نحوه پیاده‌سازی ماژول‌های اصلی مانند بخش آموزش، تست سرعت و گزارش‌گیری، و همچنین چالش‌های فنی مواجه شده در طول فرآیند توسعه ارائه می‌گردد.
- **فصل پنجم، بر ارزیابی و تحلیل نتایج حاصل از سامانه پیاده‌سازی شده تمرکز دارد.** در این فصل، روش‌های به کار رفته برای تست و ارزیابی سامانه بیان شده و نتایج عملکردی آن، همراه با طرح تجاری پیش‌بینی شده برای پروژه، مورد بحث قرار می‌گیرد.
- **در نهایت، فصل ششم، به نتیجه‌گیری کلی از دستاوردهای پروژه، بیان محدودیت‌های موجود و ارائه پیشنهاداتی برای توسعه و بهبود سامانه در آینده اختصاص یافته است.**

---

<sup>1</sup> Frontend

<sup>2</sup> Backend

<sup>3</sup> Functional Requirements

<sup>4</sup> Non- Functional Requirements

<sup>5</sup> User Interface

<sup>6</sup> Application Programming Interface

## فصل دوم

### مفاهیم

#### ۲-۱- مقدمه

پیش از ورود به مراحل طراحی و پیاده‌سازی سامانه، درک عمیق مبانی نظری، بررسی کارهای پیشین و آشنایی با ابزارهای فنی مورد استفاده، امری ضروری است. این فصل به منظور ایجاد یک شالوده علمی و فنی مستحکم برای پروژه تدوین شده است. در این راستا، ابتدا به تشریح مفاهیم کلیدی و اصول مرتبط با مهارت تایپ ده‌انگشتی و اهمیت آن پرداخته خواهد شد. سپس، در بخش پیشینه تحقیق، به بررسی و تحلیل دقیق سامانه‌های آموزش تایپ موجود، اعم از نمونه‌های مطرح خارجی و داخلی، می‌پردازیم تا ضمن شناسایی نقاط قوت و ضعف آن‌ها، نوآوری‌ها و وجه تمایز پروژه حاضر به روشنی مشخص گردد. در ادامه، فناوری‌ها و ابزارهای به کار رفته در توسعه سامانه به تفکیک دو بخش فرانت‌اند و بک‌اند معرفی می‌شوند؛ در بخش فرانت‌اند، فریم‌ورک Vue.js و در بخش بک‌اند، فناوری‌هایی نظیر Fastify (Node.js)، Prisma (ORM)، پایگاه داده PostgreSQL و مفاهیم امنیتی مرتبط با احراز هویت (JWT) و (OTP) مورد بررسی قرار خواهند گرفت. در نهایت، این فصل با یک جمع‌بندی، زمینه لازم برای درک تصمیمات اتخاذ شده در فصول طراحی و پیاده‌سازی را فراهم می‌آورد.

#### ۲-۲- مبانی نظری تایپ ده‌انگشتی

##### ۲-۲-۱- اصول و اهمیت تایپ لمسی<sup>۲</sup>

تایپ لمسی به روشی از تایپ اطلاق می‌شود که در آن، کاربر بدون نگاه کردن به کلیدهای کیبورد و صرفاً با اتکا به حافظه عضلانی<sup>۳</sup> انگشتان، عمل تایپ را انجام می‌دهد. این مهارت بر پایه یک سری اصول استاندارد و

---

<sup>۱</sup> Object-Relational Mapper

<sup>۲</sup> Touch Typing

<sup>۳</sup> Muscle Memory

مشخص بنا شده است که هدف آن بهینه‌سازی حرکت انگشتان و به حداقل رساندن جابجایی دست‌ها بر روی کیبورد است.

اساس و شالوده این روش، "ردیف پایه"<sup>۱</sup> است. در کیبوردهای استاندارد<sup>۲</sup> QWERTY، این ردیف شامل کلیدهای (A, S, D, F) برای دست چپ و کلیدهای (J, K, L, ;) برای دست راست می‌باشد. در حالت استراحت، انگشتان اشاره هر دست بر روی کلیدهای برجسته F و J قرار می‌گیرند و سایر انگشتان به ترتیب در کنار آنها مستقر می‌شوند. از این موقعیت مرکزی، هر انگشت مسئولیت فشردن مجموعه‌ای از کلیدهای مشخص در بالا، پایین و اطراف خود را بر عهده می‌گیرد. این تقسیم وظایف سیستماتیک، پایه و اساس شکل‌گیری حافظه عضلانی است؛ به این معنا که پس از تمرین کافی، مغز به صورت ناخودآگاه دستور حرکت صحیح را به انگشت مربوطه ارسال می‌کند، بدون آنکه نیازی به جستجوی چشمی کلیدها باشد.

اهمیت کسب این مهارت در دنیای دیجیتال امروز از چند جنبه قابل بررسی است:

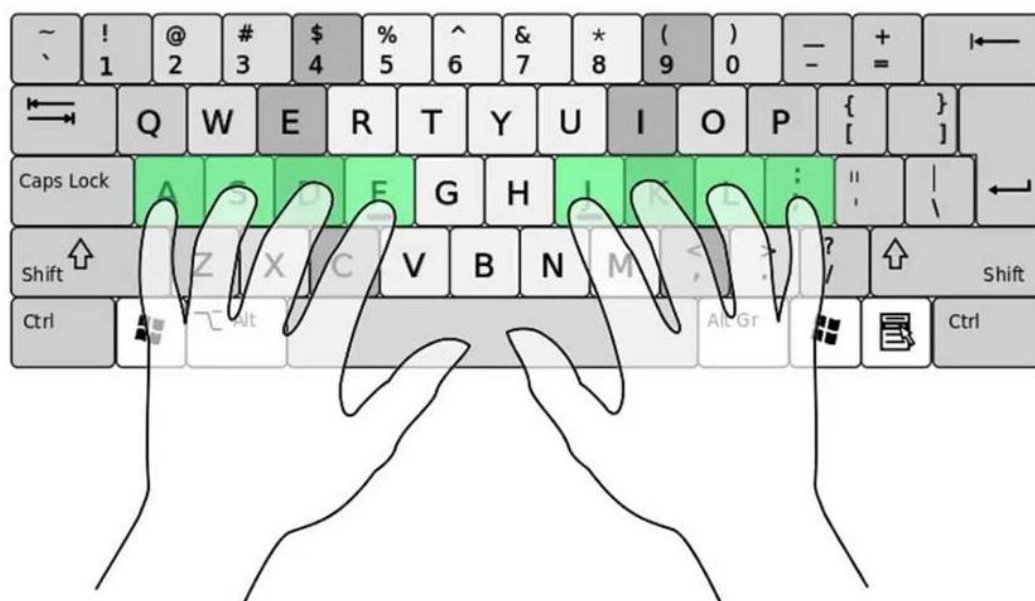
- **افزایش چشمگیر سرعت و بهره‌وری:** مهم‌ترین و ملموس‌ترین مزیت تایپ لمسی، افزایش قابل توجه سرعت ورود اطلاعات است. این امر مستقیماً به صرفه‌جویی در زمان و افزایش بهره‌وری در انجام اموری مانند نامه‌نگاری، برنامه‌نویسی، و تولید محتوا منجر می‌شود.
- **افزایش دقت و کاهش خطا:** از آنجایی که در این روش، نگاه کاربر به جای کیبورد بر روی صفحه نمایش متمرکز است، خطاها به محض وقوع، شناسایی و تصحیح می‌شوند. این تمرکز مداوم بر خروجی، به طور طبیعی منجر به کاهش تعداد خطاهای تایپی می‌گردد.
- **کاهش بار شناختی<sup>۳</sup> و افزایش تمرکز:** با خودکار شدن فرآیند تایپ، مغز از وظیفه طاق‌فرسای "پیدا کردن کلیدها" آزاد می‌شود. این منابع شناختی آزاد شده، به طور کامل به فرآیند تفکر، خلاقیت و تمرکز بر روی محتوایی که در حال تولید است، اختصاص می‌یابد.
- **بهبود جنبه‌های ارگونومیک<sup>۴</sup> و سلامت:** حرکت مداوم سر و گردن برای نگاه کردن بین صفحه نمایش و کیبورد، در بلندمدت می‌تواند باعث خستگی و دردهای عضلانی شود. تایپ لمسی با حذف این حرکات اضافی و ترویج یک وضعیت نشستن استاندارد، به حفظ سلامت فیزیکی کاربر کمک شایانی می‌کند. در ادامه و شکل ۲-۱، نحوه صحیح قرارگیری دست‌ها و انگشتان بر روی ردیف پایه را به صورت بصری نمایش می‌دهد.

<sup>۱</sup> Home Row

<sup>۲</sup> Standard QWERTY Keyboards

<sup>۳</sup> Cognitive Load

<sup>۴</sup> Ergonomics



شکل ۱-۲: نمای از قرار گرفتن انگشتان روی کیبورد

## ۲-۲-۲- استانداردهای ارزیابی (WPM و دقت)

برای اینکه بتوان مهارت تایپ یک کاربر را به صورت عینی، قابل اندازه گیری و قابل مقایسه ارزیابی کرد، از معیارهای استاندارد استفاده می شود که در سطح جهانی پذیرفته شده اند. این معیارها به کاربر اجازه می دهند تا عملکرد خود را در دو بعد اصلی سرعت و دقت بسنجند. در این پروژه نیز از همین استانداردها برای ارزیابی کاربران استفاده می شود.

### سرعت: کلمه در دقیقه

مشهورترین و اصلی ترین معیار برای سنجش سرعت تایپ، WPM است. این شاخص نشان می دهد که یک کاربر به طور متوسط در یک دقیقه چه تعداد کلمه را می تواند تایپ کند. نکته حائز اهمیت در محاسبه این معیار، تعریف استاندارد "کلمه" است. در این استاندارد، یک "کلمه" لزوماً معادل یک واژه در لغت نامه نیست؛ بلکه به صورت قراردادی، هر پنج کاراکتر تایپ شده (شامل حروف، اعداد، علائم نگارشی و فاصله ها) به عنوان یک کلمه در نظر گرفته می شود. این تعریف استاندارد، امکان مقایسه عادلانه سرعت تایپ را حتی در زبان ها و متن های مختلف فراهم می کند.

فرمول محاسبه WPM به صورت زیر است:

$$WPM = \frac{\left( \frac{\text{تعداد کل کاراکترهای تایپ شده}}{5} \right)}{\text{زمان کل به دقیقه}}$$

در ارزیابی سرعت، دو نوع WPM قابل تعریف است:

- **سرعت خام<sup>۱</sup>:** این معیار صرفاً سرعت تایپ کاربر را بدون در نظر گرفتن اشتباهات محاسبه می‌کند.
- **سرعت خالص<sup>۲</sup>:** این معیار که تصویر دقیق‌تری از مهارت کاربر ارائه می‌دهد، تعداد خطاهای تایپی را از سرعت خام کسر می‌کند. فرمول آن معمولاً به صورت (سرعت خام - (تعداد خطاها / زمان به دقیقه)) محاسبه می‌شود و نشان‌دهنده سرعت تایپ مفید و مؤثر کاربر است.

## دقت<sup>۳</sup>

سرعت بالا به تنهایی نشان‌دهنده مهارت تایپ نیست. یک تایپیست ماهر باید بتواند با حداقل خطا تایپ کند. معیار **دقت**، درصدی از کاراکترها را که به درستی تایپ شده‌اند، نسبت به کل کاراکترهای تایپ شده نشان می‌دهد. این شاخص به صورت مستقیم کیفیت کار تایپیست را منعکس می‌کند.

فرمول محاسبه دقت به شرح زیر است:

$$\text{Accuracy}(\%) = \frac{\text{تعداد کاراکترهای صحیح}}{\text{تعداد کل کاراکترهای تایپ شده}} \times 100$$

در نهایت، یک ارزیابی کامل از مهارت تایپ، نیازمند در نظر گرفتن هر دو معیار سرعت و دقت به صورت همزمان است. سرعت بالا تنها زمانی ارزشمند است که با دقت بالایی همراه باشد، زیرا تصحیح اشتباهات خود فرآیندی زمان‌بر است که می‌تواند مزیت ناشی از سرعت بالا را از بین ببرد. سامانه طراحی شده در این پروژه نیز با ارائه همزمان این دو معیار، بازخوردی جامع از عملکرد کاربر ارائه می‌دهد.

## ۳-۲- پیشینه تحقیق: بررسی سامانه‌های موجود

هیچ پروژه‌ای در خلا شکل نمی‌گیرد. برای درک بهتر جایگاه، اهمیت و نوآوری یک پروژه جدید، ضروری است که ابتدا وضعیت فعلی حوزه مورد نظر و راهکارهای موجود در آن را به دقت بررسی کنیم. این بخش به عنوان پیشینه تحقیق، به تحلیل و مقایسه سامانه‌های مطرح آموزش تایپ در سطح جهانی و داخلی می‌پردازد. هدف از این بررسی، شناسایی نقاط قوت و ضعف رقبا و در نهایت، روشن ساختن شکافی است که پروژه حاضر قصد دارد آن را پر کند.

۳-۲-۱- بررسی وبسایت 10fastfingers.com

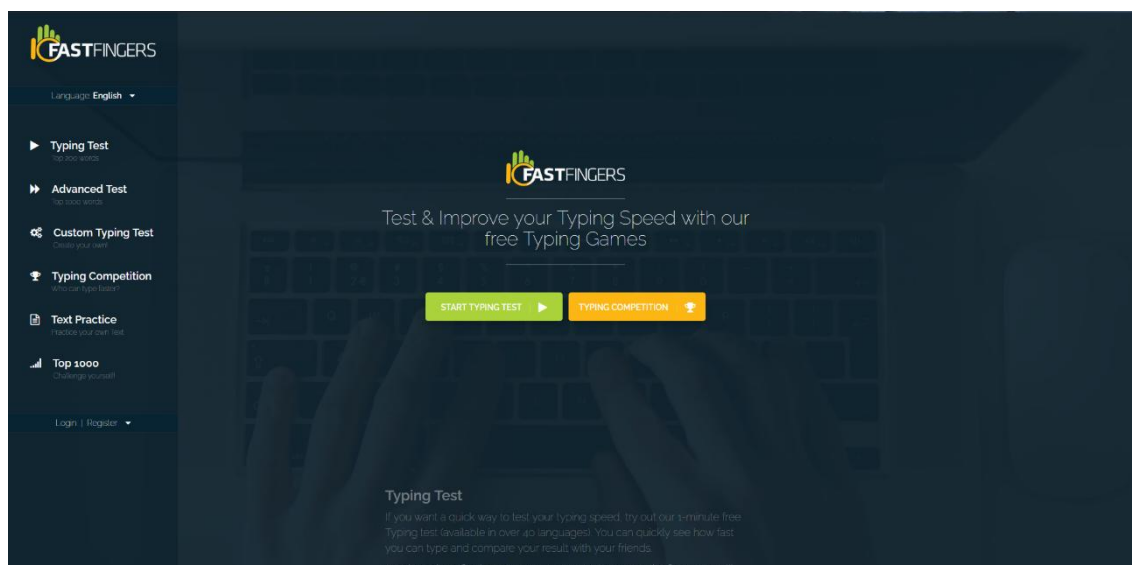
<sup>1</sup> Gross WPM

<sup>2</sup> Net WPM

<sup>3</sup> Accuracy

وبسایت 10fastfingers.com یکی از محبوب‌ترین پلتفرم‌ها در جهان برای ارزیابی سرعت تایپ است. شهرت اصلی این وبسایت به دلیل تمرکز ویژه بر روی تست‌های سریع و ایجاد یک محیط رقابتی است. کاربران می‌توانند در آزمون‌های یک دقیقه‌ای شرکت کرده، سرعت خود را بسنجند و نتیجه را با میلیون‌ها کاربر دیگر در سراسر جهان مقایسه کنند. وجود جدول‌های رتبه‌بندی و امکان شرکت در مسابقات، انگیزه بالایی برای تمرین مستمر و بهبود رکورد در کاربران ایجاد می‌کند.

با وجود این نقطه قوت، این سامانه دارای ضعف‌های قابل توجهی نیز می‌باشد. مهم‌ترین کاستی آن، این است که این وبسایت «فقط مخصوص تست سرعت است و بخشی برای یادگیری جداگانه حروف ندارد». این موضوع باعث می‌شود که 10fastfingers.com برای کاربران مبتدی که به دنبال یک مسیر آموزشی ساختاریافته برای یادگیری اصول از پایه هستند، ابزار مناسبی نباشد. علاوه بر این، «رابط کاربری قدیمی‌ای دارد» که در مقایسه با استانداردهای طراحی مدرن، ممکن است برای کاربران امروزی به اندازه کافی جذاب و کاربرپسند نباشد. نمای کلی این وبسایت در شکل ۲-۲ نشان داده شده است.



شکل ۲-۲: نمای کلی سایت 10fastfingers

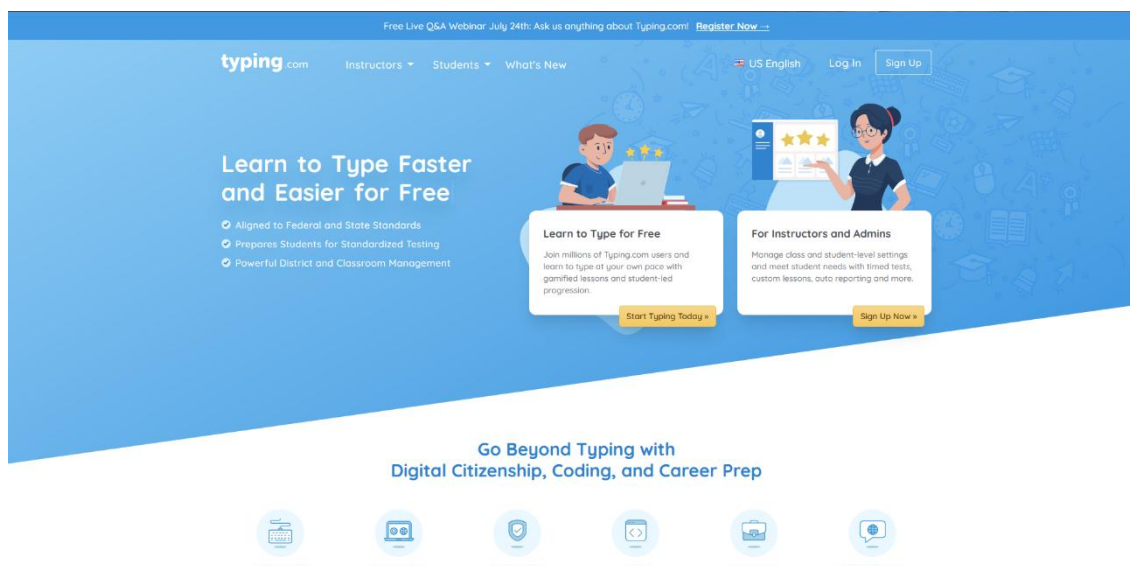
## ۲-۳-۱- بررسی وبسایت typing.com

وبسایت typing.com رویکردی متفاوت از 10fastfingers.com اتخاذ کرده و تمرکز اصلی خود را بر جنبه آموزشی قرار داده است. این پلتفرم یک مسیر یادگیری کامل و طبقه‌بندی شده از سطح مبتدی تا پیشرفته



ارائه می‌دهد که شامل درس‌های ویدیویی، تمرینات تعاملی و بازی‌های آموزشی است. این ساختار منسجم، typing.com را به گزینه‌ای عالی برای مدارس و کاربرانی تبدیل کرده است که به دنبال یادگیری اصولی و گام‌به‌گام هستند.

با این حال، این سامانه نیز در بخش کلیدی دیگری دچار ضعف است. بر اساس بررسی‌های انجام شده، این وبسایت «فاقد بخش گزارش‌گیری مناسبی است». اگرچه کاربر پس از هر تمرین بازخورد دریافت می‌کند، اما سیستم جامعی برای رصد و تحلیل پیشرفت بلندمدت وجود ندارد. نبود نمودارهای پیشرفت و آمارهای تحلیلی دقیق باعث می‌شود کاربر نتواند به سادگی عملکرد خود را در طول زمان ارزیابی کرده و نقاط ضعف خود را برای بهبود هدفمند شناسایی کند. در ادامه نمای کلی این وبسایت در شکل ۲-۳ قابل مشاهده است.



شکل ۲-۳: نمای کلی سایت typing.com

## ۲-۳-۲- تحلیل نمونه‌های ایرانی

علاوه بر نمونه‌های بین‌المللی، بررسی سامانه‌های توسعه‌یافته در داخل کشور نیز درک بهتر نیازهای کاربر ایرانی و فضای رقابتی موجود حائز اهمیت است. این سامانه‌ها اغلب با در نظر گرفتن ویژگی‌های زبان فارسی طراحی شده و می‌توانند دیدگاه‌های ارزشمندی را ارائه دهند.

## ۲-۳-۱- بررسی وبسایت Typeo.ir

در میان پلتفرم‌های ایرانی، وبسایت Typeo.ir یکی از شناخته‌شده‌ترین ابزارها برای تمرین تایپ فارسی است. نقطه قوت اصلی این سامانه، فراهم آوردن «تمام امکانات پایه» مورد نیاز برای یک کاربر است.<sup>۱</sup> این وبسایت

به خوبی توانسته است نیازهای اساسی مانند ارائه متون فارسی برای تمرین و محاسبه سرعت را پوشش دهد و به عنوان یک ابزار کار راه انداز در دسترس کاربران فارسی زبان قرار گیرد.

با این حال، این سامانه نیز فضایی برای بهبود و ارائه خدمات پیشرفته تر باقی گذاشته است. ضعف اصلی آن در مقایسه با اهداف پروژه حاضر، کمبود در دو حوزه کلیدی است. اول، این وبسایت فاقد «امکانات شخصی سازی بیشتری» است.<sup>۱</sup> قابلیت هایی مانند تغییر تم، تنظیم فونت یا انتخاب طول متن که می توانند تجربه کاربری را به شکل قابل توجهی بهبود بخشیده و انگیزه کاربر را برای تمرین های طولانی مدت حفظ کنند، در این پلتفرم به صورت محدود وجود دارد یا اصلاً وجود ندارد. دوم، این پروژه در نظر دارد تا «رابط کاربری چشمگیری» را طراحی کند<sup>۲</sup>، که این موضوع نشان می دهد رابط کاربری فعلی Typeo با وجود کاربردی بودن، می تواند از نظر زیبایی شناسی و استانداردهای طراحی مدرن، جذاب تر و پویاتر باشد.

## ۲-۳-۳- جمع بندی و تحلیل شکاف های موجود

تحلیل و مقایسه سامانه های پیشین، چه در سطح بین المللی و چه داخلی، نشان می دهد که با وجود ارائه خدمات ارزشمند توسط هر یک، هیچ کدام راهکاری کامل و همه جانبه ارائه نمی دهند. هر پلتفرم بر بخشی از نیازهای کاربر تمرکز کرده و در بخش های دیگر دچار کاستی است. این بررسی، یک شکاف مشخص در بازار ابزارهای آموزش تایپ را آشکار می سازد که می توان آن را در سه حوزه اصلی خلاصه کرد:

۱. شکاف یکپارچگی<sup>۱</sup>: پلتفرم ها عمدتاً یا بر آموزش (مانند typing.com) یا بر سنجش (مانند 10fastfingers.com) متمرکز هستند. جای خالی یک سامانه یکپارچه که مسیر یادگیری را از آموزش اصولی و گام به گام آغاز کرده و به تمرین های پیشرفته و تست های رقابتی ختم کند، به وضوح احساس می شود.

۲. شکاف تحلیل و بازخورد<sup>۲</sup>: بسیاری از سامانه ها، از جمله typing.com، فاقد بخش گزارش گیری مناسب برای تحلیل بلندمدت عملکرد کاربر هستند. کاربران ابزار قدرتمندی برای مشاهده بصری روند پیشرفت خود، تحلیل خطاها و دریافت بازخورد هوشمند برای بهبود نقاط ضعف خود در اختیار ندارند.

۳. شکاف تجربه کاربری شخصی سازی<sup>۳</sup>: همانطور که در تحلیل Typeo.ir نیز مشهود بود، بسیاری از پلتفرم ها «امکانات شخصی سازی» محدودی ارائه می دهند و از نظر طراحی رابط کاربری جای پیشرفت دارند.<sup>۴</sup> ارائه یک محیط جذاب، مدرن و کاملاً قابل تنظیم می تواند به طور چشمگیری انگیزه و رضایت کاربر را افزایش دهد.

<sup>1</sup> Integration Gap

<sup>2</sup> Analytics Gap

<sup>3</sup> Customization Gap

## ۴-۲- فناوری‌های مورد استفاده در پروژه

انتخاب پشته فناوری<sup>۱</sup> مناسب، یکی از مهم‌ترین تصمیمات در مهندسی هر پروژه نرم‌افزاری است که تأثیر مستقیمی بر کیفیت، عملکرد و قابلیت توسعه‌پذیری محصول نهایی دارد. این پروژه با بهره‌گیری از یک معماری مدرن **fullstack**<sup>۲</sup> توسعه یافته است. در این بخش، به تشریح دقیق ابزارها، کتابخانه‌ها و فریم‌ورک‌هایی که در دو بخش اصلی سامانه، یعنی سمت کاربر (Frontend) و سمت سرور (Backend)، به کار گرفته شده‌اند، می‌پردازیم.

### ۴-۲-۱- فناوری‌های سمت کاربر (Frontend)

بخش فرانت‌اند، به عنوان رابط کاربری سامانه، مسئولیت اصلی تعامل مستقیم با کاربر و ارائه یک تجربه روان و جذاب را بر عهده دارد. برای دستیابی به این هدف، از مجموعه‌ای از فناوری‌های مدرن و هماهنگ بر پایه اکوسیستم Vue.js استفاده شده است.

### ۴-۲-۱-۱- معرفی فریم‌ورک Vue.js و Nuxt 3

هسته اصلی رابط کاربری این پروژه بر پایه **Vue 3**، یکی از محبوب‌ترین کتابخانه‌های جاوا اسکریپت برای ساخت واسط‌های کاربری، بنا شده است. معماری کامپوننت-محور<sup>۳</sup> و سیستم واکنش‌گرایی<sup>۴</sup> در Vue، توسعه UI های پیچیده را ساده و مدیریت وضعیت آن‌ها را بهینه می‌سازد.

برای بهره‌گیری حداکثری از قابلیت‌های Vue 3 و پوشش نیازهای یک اپلیکیشن مدرن، از فریم‌ورک **Nuxt 3** استفاده شده است. Nuxt یک فریم‌ورک سطح بالاتر بر پایه Vue است که امکانات قدرتمندی نظیر **رندر سمت سرور**<sup>۵</sup> (SSR) و ساخت اپلیکیشن‌های **تک‌صفحه‌ای**<sup>۶</sup> (SPA) را به صورت پیش‌فرض فراهم می‌کند. استفاده از SSR می‌تواند به بهبود سئو<sup>۷</sup> (SEO) و کاهش زمان بارگذاری اولیه صفحه کمک شایانی کند که برای یک پلتفرم آموزشی آنلاین مزیت مهمی محسوب می‌شود.

<sup>۱</sup> Technology Stack

<sup>۲</sup> Full-stack Architecture

<sup>۳</sup> Component-Based

<sup>۴</sup> Reactivity

<sup>۵</sup> Server-Side Rendering

<sup>۶</sup> Single-Page Application

<sup>۷</sup> Search Engine Optimization

## ۲-۴-۱-۲- معرفی کتابخانه‌های UI

به منظور تسریع فرآیند توسعه، ایجاد یک طراحی یکپارچه و افزودن قابلیت‌های بصری، در کنار فریم‌ورک اصلی از مجموعه‌ای از کتابخانه‌ها و ابزارهای UI استفاده شده است:

- **TailwindCSS**: این پروژه برای استایل‌دهی از فریم‌ورک **TailwindCSS** بهره می‌برد که یک فریم‌ورک **utility-first** است. این رویکرد به توسعه‌دهنده اجازه می‌دهد تا بدون خروج از فایل HTML و نوشتن CSS سفارشی، با استفاده از کلاس‌های کاربردی از پیش تعریف‌شده، رابط‌های کاربری پیچیده و واکنش‌گرا را به سرعت طراحی کند.
- **Nuxt UI و Headless UI**: برای کامپوننت‌های آماده، از کتابخانه **nuxt/ui@** استفاده شده که مجموعه‌ای هماهنگ با **Nuxt** و **TailwindCSS** را ارائه می‌دهد. علاوه بر این، در موارد نیاز به کامپوننت‌های کاملاً سفارشی از نظر ظاهری، از **Headless UI** استفاده شده است. این کتابخانه، منطق و دسترسی‌پذیری<sup>۱</sup> کامپوننت‌های تعاملی مانند **Switch** یا **Modal** را فراهم می‌کند، در حالی که استایل‌دهی کامل آن به توسعه‌دهنده واگذار می‌شود.
- **nuxt-icon**: برای مدیریت و استفاده بهینه از آیکون‌ها در سراسر پروژه، از کتابخانه **nuxt-icon** استفاده شده است که امکان دسترسی به مجموعه‌های متنوعی از آیکون‌ها را به صورت یکپارچه فراهم می‌آورد.

## ۲-۴-۲- فناوری‌های سمت سرور (Backend)

بخش بک‌اند به عنوان هسته مرکزی سامانه، وظیفه مدیریت منطق کسب‌وکار، پردازش داده‌ها، ارتباط با پایگاه داده و تأمین امنیت را بر عهده دارد. برای این بخش، یک پشته فناوری مبتنی بر **Node.js** انتخاب شده است که به دلیل عملکرد بالا و اکوسیستم غنی، برای ساخت API های سریع و مقیاس‌پذیر گزینه‌ای ایده‌آل محسوب می‌شود.

## ۲-۴-۳- معرفی فریم‌ورک Fastify

برای ساخت API های این پروژه، از فریم‌ورک **Fastify** استفاده شده است. **Fastify** یک فریم‌ورک وب مدرن و بسیار سریع برای **Node.js** است که با تمرکز بر سربار<sup>۲</sup> پایین و عملکرد بالا طراحی شده است. معماری مبتنی بر پلاگین<sup>۳</sup> آن، توسعه‌پذیری بالایی را فراهم می‌آورد. در این پروژه از افزونه‌های کلیدی مانند **fastify/cors@**

<sup>۱</sup> Accessibility

<sup>۲</sup> overhead

<sup>۳</sup> Plugin

برای مدیریت دسترسی‌های بین مبدأ<sup>۱</sup> (CORS)، `fastify/autoload@` برای بارگذاری خودکار مسیرها و `fastify/sensible@` برای مدیریت خطاها و پاسخ‌های HTTP<sup>۲</sup> استاندارد استفاده شده است.

## ۲-۲-۴-۲ معرفی Prisma ORM

ارتباط میان منطق برنامه و پایگاه داده از طریق یک ORM نسل جدید به نام **Prisma** مدیریت می‌شود. پرزما با فراهم کردن یک لایه انتزاعی امن و شهودی، کار با پایگاه داده را ساده‌تر می‌کند. تعریف اسکیمای به صورت اعلانی<sup>۳</sup>، تولید خودکار کلاینت کاملاً تایپ-سیف<sup>۴</sup> و ابزارهای قدرتمند برای مایگریشن<sup>۵</sup>، از ویژگی‌های برجسته پرزما هستند که فرآیند توسعه و نگهداری پایگاه داده را به شکل چشمگیری بهبود می‌بخشند.

## ۳-۲-۴-۲ معرفی پایگاه داده PostgreSQL

پایگاه داده اصلی این پروژه **PostgreSQL** است. این سیستم مدیریت پایگاه داده رابطه‌ای<sup>۶</sup> (RDBMS) به دلیل قابلیت اطمینان بالا، استحکام، پشتیبانی از انواع داده‌های پیچیده و قابلیت‌های پیشرفته، به عنوان یکی از قدرتمندترین پایگاه‌های داده متن‌باز در جهان شناخته می‌شود و انتخابی مناسب برای پروژه‌هایی است که نیازمند پایداری و مقیاس‌پذیری<sup>۸</sup> هستند.

## ۴-۲-۴-۲ تشریح سازوکار احراز هویت و امنیت

امنیت کاربران و داده‌ها یکی از اولویت‌های اصلی و ستون‌های بنیادین در معماری این پروژه است. برای تضمین یک محیط امن و قابل اعتماد، از مجموعه‌ای از ابزارها و روش‌های استاندارد صنعتی استفاده شده است که هر یک وظیفه مشخصی را در حفاظت از اطلاعات و مدیریت دسترسی کاربران بر عهده دارند.

## احراز هویت بدون حالت<sup>۹</sup> با JWT

فرآیند احراز هویت در این سامانه به صورت **بدون حالت** و با استفاده از استاندارد توکن‌های وب جیسون (JWT) پیاده‌سازی شده است. JWT یک استاندارد باز (RFC 7519) و یک روش فشرده و مستقل برای انتقال امن اطلاعات بین دو طرف به صورت یک شیء JSON است. این رویکرد به سرور اجازه می‌دهد تا بدون نیاز به

<sup>1</sup> Cross-Origin Resource Sharing (CORS)

<sup>2</sup> routes

<sup>3</sup> <sup>1</sup> Hypertext Transfer Protocol

<sup>4</sup> declarative

<sup>5</sup> Type-Safe

<sup>6</sup> migration

<sup>7</sup> Relational Database Management System

<sup>8</sup> Scalability

<sup>9</sup> Stateless

ذخیره اطلاعات نشست<sup>۱</sup> در حافظه یا پایگاه داده، هویت کاربر را در هر درخواست اعتبارسنجی کند، که این امر مقیاس‌پذیری سامانه را به شدت افزایش می‌دهد.

یک توکن JWT از سه بخش اصلی تشکیل شده است که با نقطه (.) از هم جدا می‌شوند:

۱. سرآیند<sup>۲</sup>: این بخش معمولاً شامل دو قسمت است: نوع توکن (که JWT است) و الگوریتم امضای استفاده شده برای رمزنگاری، مانند HS256.

۲. محموله<sup>۳</sup>: این بخش حاوی "ادعاها"<sup>۴</sup> است. ادعاها اطلاعاتی درباره یک موجودیت (معمولاً کاربر) و متادیتای توکن هستند. این اطلاعات می‌توانند شامل شناسه‌ی کاربر، نقش‌ها و مجوزها، و همچنین اطلاعات استاندارد مانند زمان انقضای توکن (exp) باشند.

۳. امضا<sup>۵</sup>: برای تأیید صحت و عدم دستکاری توکن، یک امضا با استفاده از سرآیند رمزگذاری شده، محموله رمزگذاری شده و یک کلید مخفی<sup>۶</sup> که تنها در سمت سرور نگهداری می‌شود، ایجاد می‌گردد. هرگونه تغییر در سرآیند یا محموله، امضا را نامعتبر می‌سازد.

در این پروژه، کتابخانه **jsonwebtoken** برای پیاده‌سازی این فرآیند به کار رفته است.<sup>۲</sup> جریان کاری به این صورت است که پس از ورود موفق کاربر، سرور یک JWT حاوی شناسه‌ی کاربر تولید و امضا کرده و به کلاینت (Nuxt 3) ارسال می‌کند. کلاینت این توکن را ذخیره کرده و در هدر Authorization تمام درخواست‌های بعدی خود به سرور ارسال می‌نماید. سرور با اعتبارسنجی امضای توکن، هویت کاربر را تأیید کرده و به درخواست او پاسخ می‌دهد.

## ذخیره‌سازی امن رمز عبور با Bcrypt

یکی از مهم‌ترین اصول امنیتی، عدم ذخیره‌سازی رمزهای عبور به صورت متن ساده<sup>۷</sup> است. برای این منظور، از یک الگوریتم هشینگ<sup>۸</sup> قوی و استاندارد استفاده می‌شود. در این پروژه، کتابخانه **bcryptjs** برای این کار انتخاب شده است.<sup>۳</sup> Bcrypt به طور خاص برای هش کردن رمز عبور طراحی شده و دو ویژگی کلیدی دارد:

---

<sup>1</sup> Session

<sup>2</sup> Header

<sup>3</sup> Payload

<sup>4</sup> Claims

<sup>5</sup> Signature

<sup>6</sup> Secret Key

<sup>7</sup> Plain Text

<sup>8</sup> Hashing

- **کند بودن عمدی:** این الگوریتم به گونه‌ای طراحی شده که از نظر محاسباتی کند باشد. این ویژگی، حملات جستجوی فراگیر<sup>۱</sup> را بسیار زمان‌بر و پرهزینه می‌کند.
- **استفاده از Salt:** Bcrypt به صورت خودکار یک رشته تصادفی منحصر به فرد به نام "نمک"<sup>۲</sup> به هر رمز عبور قبل از هش کردن اضافه می‌کند. این کار تضمین می‌کند که حتی اگر دو کاربر رمز عبور یکسانی داشته باشند، هش نهایی آن‌ها متفاوت خواهد بود و از حملات مبتنی بر جداول رنگین‌کمانی<sup>۳</sup> جلوگیری می‌کند.

### ارسال ایمیل برای فرآیندهای حساس

برای فرآیندهای حساسی مانند بازیابی رمز عبور که نیازمند تأیید هویت کاربر خارج از بستر اصلی سایت است، از سازوکار ارسال ایمیل استفاده می‌شود. کتابخانه **nodemailer** به عنوان یک ماژول قدرتمند Node.js، مسئولیت ارسال ایمیل از سمت سرور به کاربران را بر عهده دارد.<sup>۴</sup> هنگامی که کاربر فرآیند بازیابی رمز عبور را آغاز می‌کند، سرور یک توکن یکبار مصرف و زمان‌دار تولید کرده و آن را از طریق ایمیل برای کاربر ارسال می‌کند تا کاربر بتواند هویت خود را تأیید و رمز عبور جدیدی تنظیم نماید.

### ۲-۵- جمع‌بندی

در این فصل، شالوده علمی و فنی پروژه از طریق بررسی سه حوزه کلیدی بنا نهاده شد. ابتدا، مبانی نظری مهارت تایپ ده‌انگشتی، شامل اصول تایپ لمسی و استانداردهای جهانی ارزیابی مانند WPM و دقت، مورد بحث قرار گرفت تا اهمیت و معیارهای سنجش این مهارت به وضوح مشخص شود. در ادامه، در بخش پیشینه تحقیق، با تحلیل و مقایسه سامانه‌های مطرح داخلی و خارجی، نقاط قوت و ضعف آن‌ها شناسایی و شکاف‌های موجود در زمینه‌های یکپارچگی، تحلیل بازخورد و تجربه کاربری آشکار گردید. این تحلیل، ضرورت و نوآوری پروژه حاضر را توجیه کرد. در نهایت، پشته فناوری منتخب برای توسعه سامانه به تفکیک دو بخش فرانت‌اند و بک‌اند معرفی شد؛ انتخاب ابزارهای مدرنی همچون Nuxt 3، Fastify و Prisma با هدف دستیابی به عملکرد بالا، امنیت و قابلیت توسعه‌پذیری صورت گرفت. مجموع مباحث مطرح شده در این فصل، زمینه لازم برای درک عمیق تصمیمات اتخاذ شده در فصول طراحی و پیاده‌سازی را فراهم می‌آورد.

<sup>1</sup> Brute-force

<sup>2</sup> Salt

<sup>3</sup> Rainbow Table Attacks

## فصل سوم

### شرح پروژه

#### ۳-۱- مقدمه

پس از بررسی مبانی نظری و تحلیل پیشینه تحقیق در فصل گذشته، اینک در فصل سوم، وارد مرحله حیاتی طراحی سامانه می‌شویم. این فصل به مثابه نقشه فنی و بلوپرینت<sup>۱</sup> پروژه عمل می‌کند و تمامی تصمیمات کلان معماری و جزئیات طراحی را پیش از شروع فرآیند پیاده‌سازی مشخص می‌سازد. هدف از این فصل، تبدیل نیازمندی‌های استخراج‌شده به یک طرح مهندسی دقیق و قابل اجرا است. در این راستا، ابتدا به تشریح کامل نیازمندی‌های عملکردی و غیرعملکردی می‌پردازیم. سپس، معماری کلی سیستم در دو لایه کلاینت و سرور تشریح شده و به دنبال آن، جزئیات طراحی پایگاه داده، طراحی رابط کاربری و در نهایت، طراحی رابط برنامه‌نویسی کاربردی مورد بحث و بررسی قرار خواهد گرفت.

#### ۳-۲- شرح نیازمندی‌ها

این بخش، سنگ بنای فرآیند طراحی سامانه را تشکیل می‌دهد. در اینجا، نیازمندی‌ها یا الزاماتی که سیستم باید برآورده سازد، به صورت دقیق و شفاف تعریف می‌شوند. این نیازمندی‌ها به دو دسته اصلی **عملکردی** و **غیرعملکردی** تقسیم می‌گردند تا تمامی جنبه‌های فنی و کیفی سیستم پوشش داده شود.

---

<sup>۱</sup> Blueprint



### ۳-۲-۱ نیازمندی‌های عملکردی

نیازمندی‌های عملکردی، رفتارها و قابلیت‌های مشخصی را که سیستم باید به کاربران خود ارائه دهد، توصیف می‌کنند. این نیازمندی‌ها مشخص می‌کنند که سامانه «چه کارهایی» باید انجام دهد. نیازمندی‌های عملکردی اصلی این پروژه به شرح زیر است:

#### • مدیریت کاربران و احراز هویت:

- کاربر باید بتواند با استفاده از اطلاعات شخصی در سامانه ثبت‌نام کرده و برای خود یک حساب کاربری ایجاد کند.
- کاربر باید بتواند با استفاده از نام کاربری و رمز عبور خود وارد حساب کاربری شود تا سوابق و رکوردهایش ذخیره گردد.
- کاربر باید بتواند اطلاعات حساب کاربری و تصویر پروفایل خود را ویرایش نماید.
- سامانه باید قابلیت بازیابی رمز عبور از طریق ارسال رمز یکبار مصرف به ایمیل یا شماره موبایل کاربر را فراهم کند.

#### • ماژول آموزش تعاملی:

- سامانه باید یک بخش مجزا برای یادگیری گام‌به‌گام و درس‌به‌درس تایپ حروف ارائه دهد.
- در بخش آموزش، باید راهنمای تصویری برای نمایش حالت صحیح قرارگیری انگشتان روی کیبورد وجود داشته باشد.

#### • ماژول تست سرعت و ارزیابی:

- کاربر باید بتواند در آزمون‌های تست سرعت تایپ شرکت کند.
- سامانه باید به کاربر اجازه دهد تا متن تست خود را از میان سطوح دشواری مختلف (مبتدی، متوسط، پیشرفته) انتخاب کند.
- کاربر باید بتواند اندازه پاراگراف متن تست را در سه حالت (کوتاه، متوسط، بلند) شخصی‌سازی نماید.
- کاربر باید قادر به تنظیم اندازه فونت متنی باشد که در حال تایپ آن است.

#### • گزارش‌گیری و بازخورد:

- پس از هر تست، سامانه باید گزارش کاملی از عملکرد کاربر شامل تعداد کلمه در دقیقه، دقت، تعداد اشتباهات و زمان کل را نمایش دهد.
- کاربر باید بتواند به تاریخچه‌ای از نتایج تست‌های خود دسترسی داشته باشد.

- سامانه باید روند پیشرفت کاربر در دو معیار سرعت و دقت را با استفاده از نمودار بصری نمایش دهد.
- یک جدول رتبه‌بندی عمومی باید برای نمایش برترین رکوردهای ثبت‌شده توسط کاربران وجود داشته باشد.

• **شخصی‌سازی پیشرفته رابط کاربری:**

- کاربر باید بتواند ظاهر بصری سامانه را با انتخاب از میان تم‌های رنگی از پیش تعریف‌شده، شخصی‌سازی کند.
- کاربر باید قادر به تنظیم اندازه فونت و طول پاراگراف (کوتاه، متوسط، بلند) متنی باشد که در حال تایپ آن است.

• **مدیریت سامانه (پنل ادمین):**

- مدیر باید بتواند لیست تمام کاربران را مشاهده کند.
- مدیر باید بتواند یک درس یا پاراگراف جدید به سامانه اضافه کند.
- مدیر باید بتواند محتوای درس‌ها و پاراگراف‌های موجود را ویرایش یا آن‌ها را حذف کند.
- مدیر اصلی سایت (Super Admin) باید بتواند نقش یک کاربر عادی را به ADMIN ارتقا دهد یا نقش یک مدیر را به USER تنزل دهد.

• **پشتیبانی دوزبانه:**

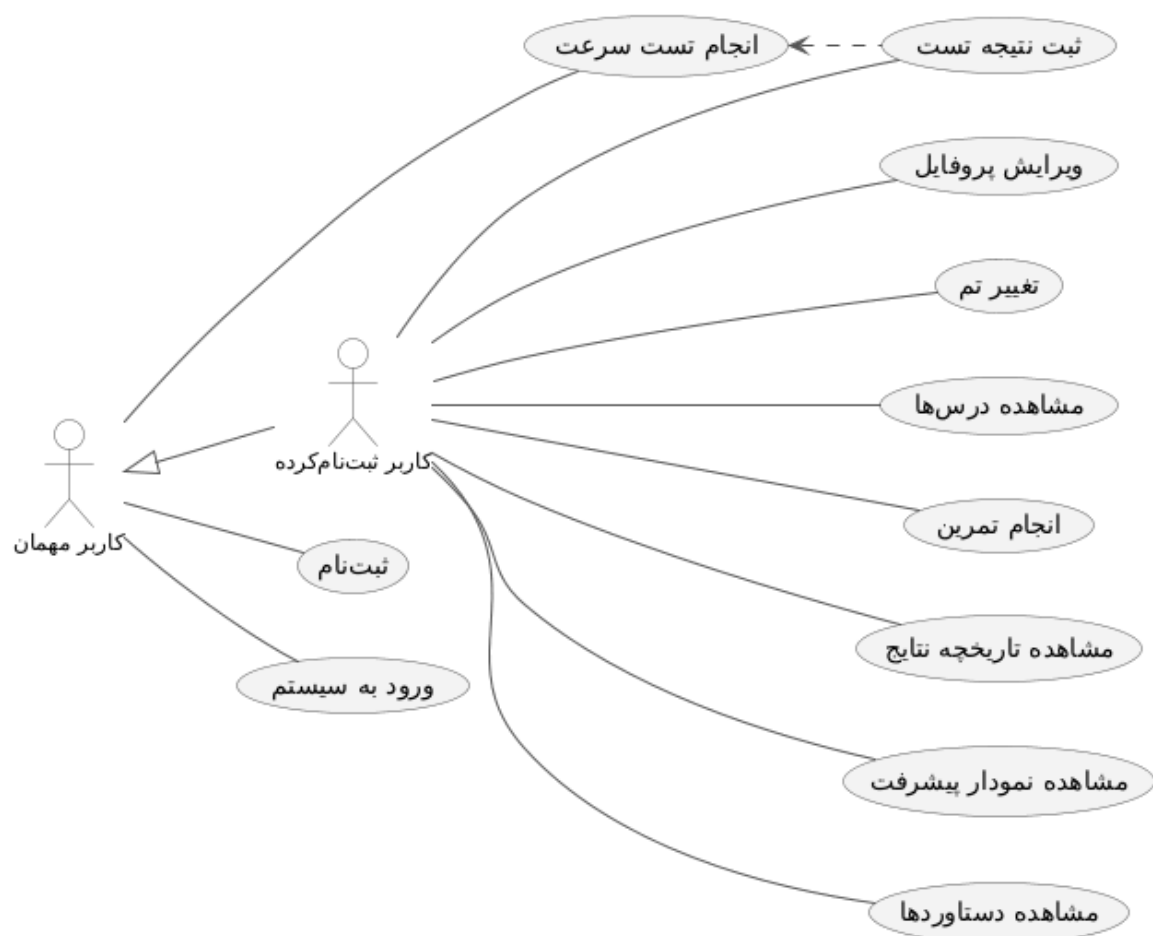
- تمام بخش‌های رابط کاربری سامانه باید به دو زبان فارسی و انگلیسی در دسترس باشند و کاربر بتواند به راحتی بین این دو زبان جابجا شود.

## نمودار موارد کاربرد<sup>۱</sup>

نمودار موارد کاربرد نشان می‌دهد که کاربران مختلف سیستم چگونه با قابلیت‌های مختلف آن تعامل دارند. همانطور که در شکل ۳-۱ مشاهده می‌شود، دو بازیگر اصلی شامل "کاربر مهمان" و "کاربر ثبت‌نام‌کرده" نمایش داده شده‌اند که به ترتیب به قابلیت‌هایی همچون ثبت‌نام، ورود، مشاهده درس‌ها، انجام تمرین، شرکت در تست تایپ و مشاهده گزارش‌ها دسترسی دارند. برای درک بهتر تعاملات میان کاربران و قابلیت‌های سامانه، از نمودار موارد کاربرد استفاده شده است.

---

<sup>۱</sup> Use Case

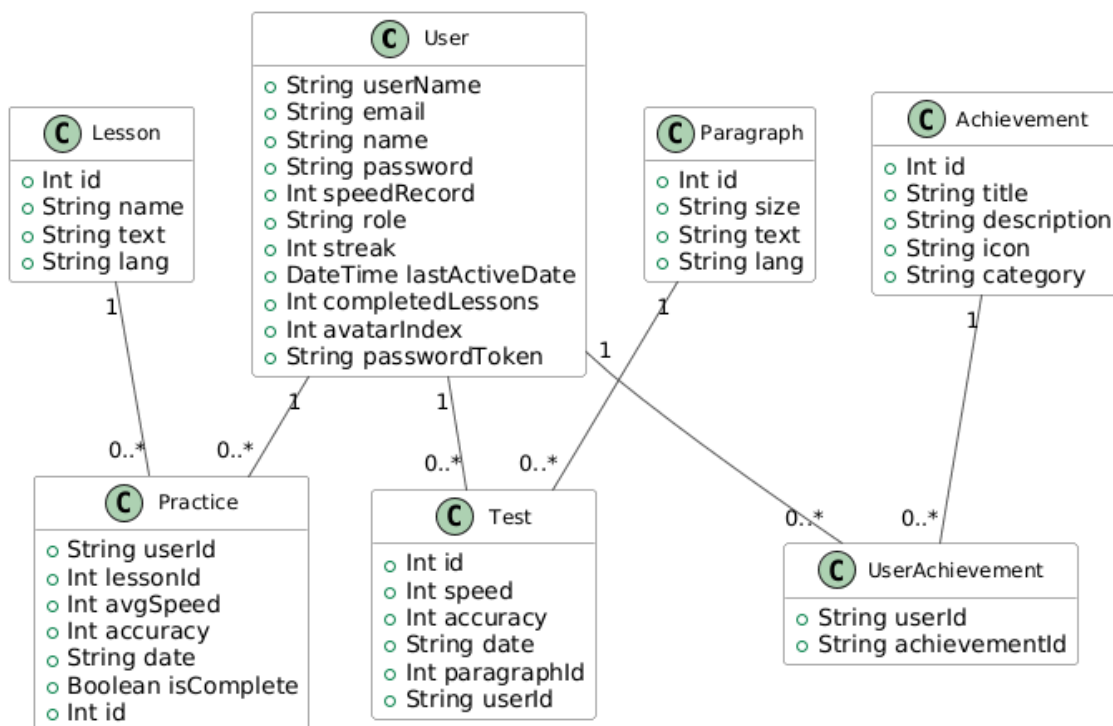


شکل ۳-۱: نمودار موارد کاربرد سامانه

### نمودار کلاس‌ها<sup>۱</sup>

نمودار کلاس سیستم ساختار داده‌ها و روابط بین موجودیت‌های اصلی پروژه را نمایش می‌دهد. شکل ۳-۲، این نمودار مبتنی بر مدل‌های تعریف‌شده در پایگاه داده (Prisma) و ساختار منطقی نرم‌افزار تهیه شده و شامل کلاس‌هایی مانند User، Test، Lesson، Practice و Achievement است.

<sup>۱</sup> Class Diagram



شکل ۳-۲: نمودار کلاس‌های اصلی سیستم بر اساس مدل‌های داده

### نمودار توالی<sup>۱</sup>

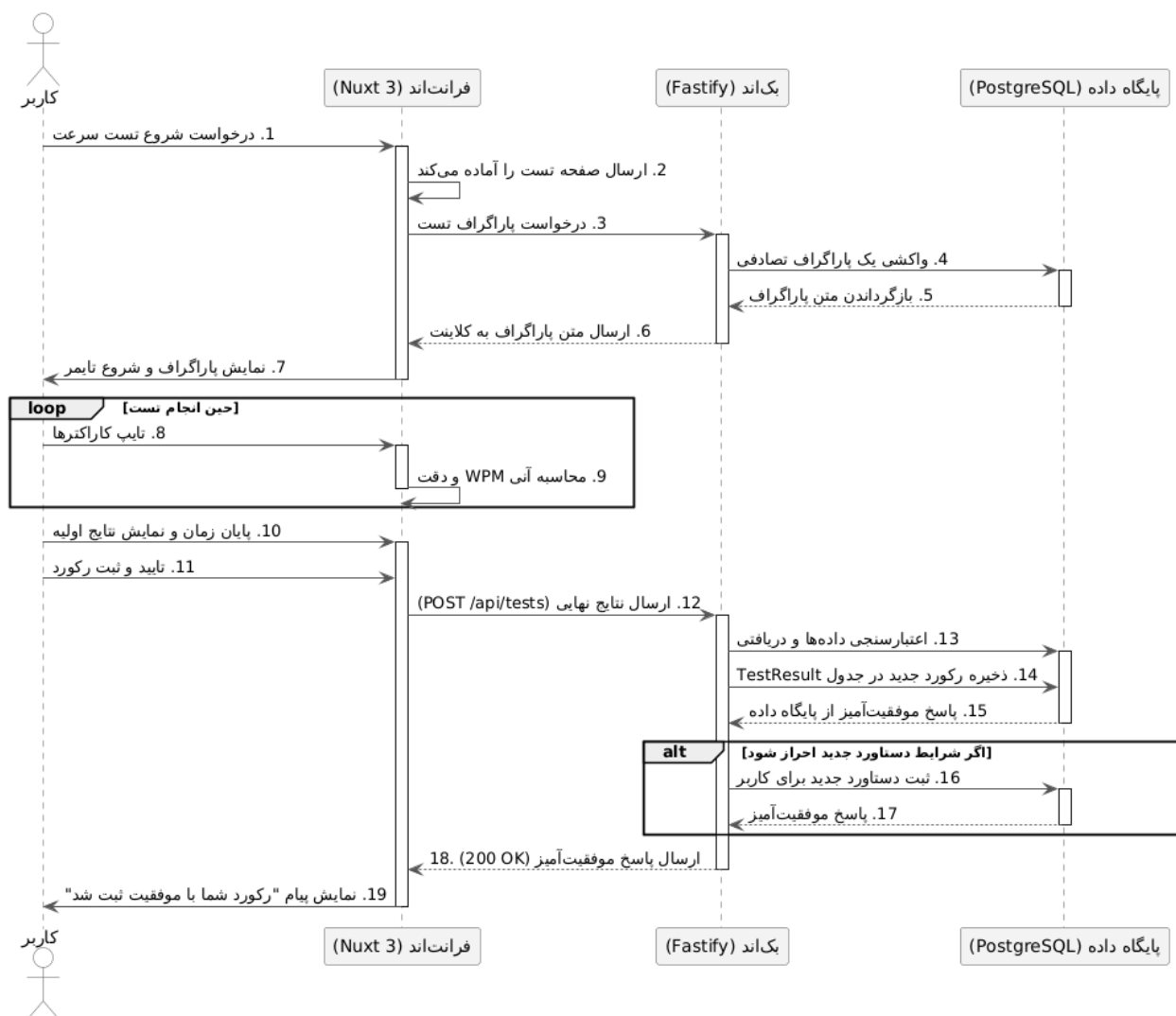
برای نمایش رفتار دینامیک و تعاملات زمانی بین اجزای سامانه، از نمودار توالی استفاده شده است. همانطور که در شکل ۳-۳ مشاهده می‌شود، این نمودار سناریوی کلیدی "انجام تست سرعت تایپ" را مدل‌سازی می‌کند که شامل چهار شرکت‌کننده اصلی است: کاربر، فرانت‌اند، بک‌اند و پایگاه داده. این فرآیند در سه فاز اصلی قابل تشریح است:

**فاز اول:** آماده‌سازی تست فرآیند با درخواست کاربر برای شروع تست آغاز می‌شود. فرانت‌اند پس از آماده‌سازی صفحه، یک درخواست برای دریافت متن آزمون به بک‌اند ارسال می‌کند. بک‌اند پس از واکنشی یک پاراگراف تصادفی از پایگاه داده، آن را به فرانت‌اند بازگردانده و تست برای کاربر نمایش داده می‌شود.

**فاز دوم:** تعامل آنی در کلاینت در حین انجام تست، تمام محاسبات آنی و بازخوردهای لحظه‌ای (مانند محاسبه و دقت) به صورت بهینه در سمت فرانت‌اند انجام می‌گردد. این تصمیم‌معماری، از ایجاد بار اضافی بر روی سرور جلوگیری کرده و یک تجربه کاربری کاملاً روان و با بازخورد فوری را تضمین می‌کند.

<sup>۱</sup> Sequence Diagram

**فاز سوم:** ثبت نهایی در سرور پس از اتمام آزمون، نتایج نهایی در قالب یک درخواست POST به بک‌اند ارسال می‌شود. در این مرحله، سرور مسئولیت اعتبارسنجی داده‌ها، ذخیره‌سازی رکورد در پایگاه داده، و اجرای منطق‌های کسب‌وکار (مانند بررسی برای کسب دستاوردهای جدید) را بر عهده دارد و در نهایت، پاسخ موفقیت‌آمیز را به کلاینت بازمی‌گرداند.



شکل ۳-۳: نمایی از نمودار توالی

### ۳-۲-۲- نیازمندی‌های غیرعملکردی

نیازمندی‌های غیرعملکردی، برخلاف نیازمندی‌های عملکردی، به تشریح «چگونگی» عملکرد سیستم می‌پردازند و بر ویژگی‌های کیفی و معیارهای کارایی آن تمرکز دارند. این نیازمندی‌ها برای اطمینان از ارائه یک تجربه کاربری باکیفیت، قابل اعتماد و امن، حیاتی هستند.

- **کارایی<sup>۱</sup>:** سامانه باید عملکردی سریع و روان داشته باشد. زمان بارگذاری اولیه صفحات، به ویژه صفحه شروع تست، باید بهینه و کوتاه باشد. همچنین، پاسخ‌دهی رابط کاربری به تعاملات کاربر (مانند فشردن کلیدها و نمایش بازخورد فوری) باید بدون تأخیر محسوس صورت گیرد و API های سمت سرور باید بتوانند در زمان کوتاهی به درخواست‌ها پاسخ دهند.
- **امنیت<sup>۲</sup>:** حفاظت از اطلاعات کاربران از اولویت‌های اصلی است. سیستم باید رمزهای عبور کاربران را با استفاده از الگوریتم‌های هشینگ امن (مانند **Bcrypt**) ذخیره کند<sup>۱</sup>. همچنین، دسترسی به مسیرهای محافظت‌شده (مانند پروفایل کاربری) باید از طریق سازوکار احراز هویت مبتنی بر توکن (مانند **JWT**) مدیریت شود تا از دسترسی‌های غیرمجاز جلوگیری گردد.
- **قابلیت استفاده<sup>۳</sup> یا کاربردپذیری:** رابط کاربری سامانه باید ساده، شهودی و کاربرپسند باشد. یک کاربر جدید باید بتواند بدون نیاز به آموزش پیچیده، به راحتی مسیر خود را در سامانه پیدا کرده، در درس‌ها شرکت کند و آزمون دهد. جریان‌های کاری باید منطقی و روان طراحی شوند تا کاربر در استفاده از سامانه دچار سردرگمی نشود.
- **مقیاس‌پذیری<sup>۴</sup>:** معماری سامانه باید به گونه‌ای طراحی شود که قابلیت مدیریت افزایش تعداد کاربران و حجم درخواست‌ها در آینده را داشته باشد. سیستم باید بتواند با افزایش بار، عملکرد پایدار خود را حفظ کرده و امکان توسعه و افزودن ویژگی‌های جدید در آینده را به راحتی فراهم سازد.

### ۳-۳- معماری سامانه

معماری سامانه، ساختار کلان و بلوک‌های اصلی تشکیل‌دهنده نرم‌افزار و نحوه تعامل آن‌ها با یکدیگر را تعریف می‌کند. انتخاب یک معماری مناسب، اساس یک پروژه قابل نگهداری، توسعه‌پذیر و کارآمد است.

#### ۳-۳-۱- نمای کلی معماری (معماری کلاینت-سرور)

این پروژه بر اساس یک معماری مدرن **کلاینت-سرور<sup>۵</sup>** طراحی شده است. در این مدل، مسئولیت‌ها به دو بخش اصلی و مستقل تفکیک شده‌اند:

<sup>1</sup> Performance

<sup>2</sup> Security

<sup>3</sup> Usability

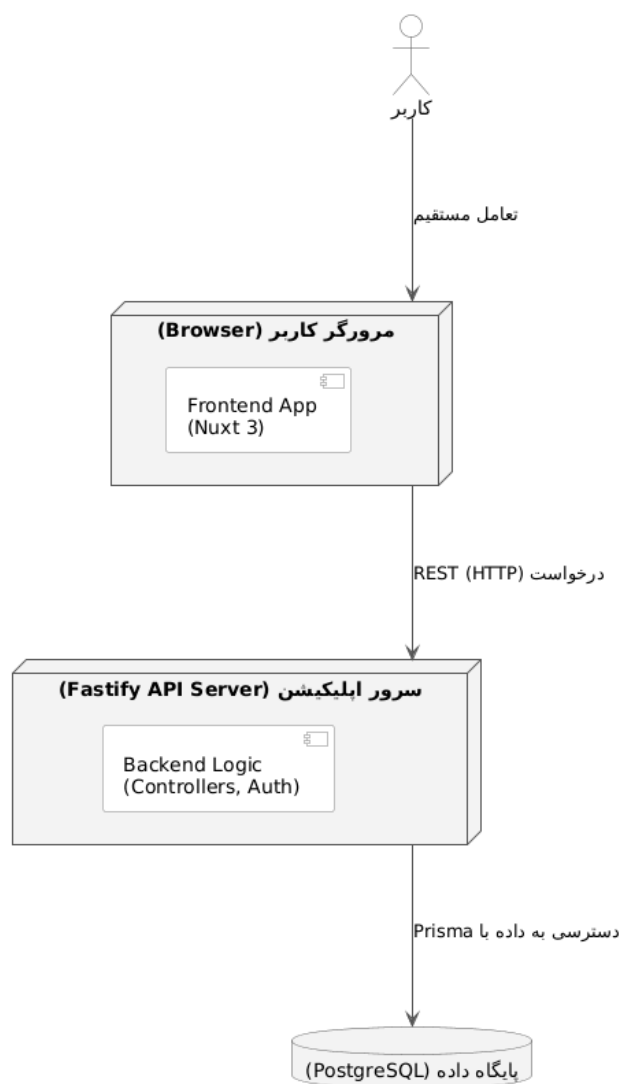
<sup>4</sup> Scalability

<sup>5</sup> Client-Server

۱. **بخش کلاینت یا فرانت اند:** این بخش، همان رابط کاربری است که در مرورگر کاربر اجرا می‌شود و مسئولیت تمام جنبه‌های بصری و تعاملی را بر عهده دارد. این اپلیکیشن با فریم‌ورک Nuxt 3 توسعه یافته است.

۲. **بخش سرور یا بک اند:** این بخش بر روی یک سرور اجرا شده و به عنوان مغز متفکر سیستم عمل می‌کند. وظایف آن شامل مدیریت منطق برنامه، پردازش داده‌ها، احراز هویت کاربران و ارتباط با پایگاه داده است. این بخش با فریم‌ورک Fastify پیاده‌سازی شده است.

برای درک بهتر ساختار معماری سامانه، در ادامه یک نمای کلی از نحوه تعامل اجزای اصلی سیستم شامل کلاینت، سرور و پایگاه داده ارائه شده است. این نمودار نحوه جریان داده بین بخش‌های مختلف را به صورت ساده و قابل فهم نمایش می‌دهد. جزئیات این معماری در شکل ۳-۴ ترسیم شده است.



شکل ۳-۴: معماری کلی سامانه آموزش تایپ

ارتباط میان این دو بخش از طریق یک **API** مبتنی بر اصول **RESTful**<sup>۱</sup> و با استفاده از پروتکل **HTTP** صورت می‌گیرد. کلاینت درخواست‌های خود را به سرور ارسال کرده و سرور پس از پردازش، پاسخ را در قالب **JSON** باز می‌گرداند.

### ۳-۳-۲- معماری سمت کاربر (Frontend)

معماری بخش فرانت‌اند با هدف ایجاد یک رابط کاربری ماژولار، قابل نگهداری و با عملکرد بالا طراحی شده است.

- **معماری مبتنی بر کامپوننت با Nuxt 3:** رابط کاربری با پیروی از الگوی معماری مبتنی بر کامپوننت ساخته شده است. تمام واسط کاربری به قطعات کوچک، مستقل و قابل استفاده مجدد به نام کامپوننت (مانند کامپوننت تایمر، کامپوننت متن آزمون، کامپوننت کارت درس و غیره) تقسیم می‌شود. این رویکرد که هسته اصلی **Vue 3** و فریم‌ورک **Nuxt 3** را تشکیل می‌دهد<sup>۵</sup>، مدیریت کدهای پیچیده را آسان‌تر کرده و توسعه و تست پروژه را بهینه می‌سازد.

- **الگوی مدیریت وضعیت با Pinia:** در اپلیکیشن‌های تک‌صفحه‌ای، داده‌ها و وضعیت‌های مشترکی بین کامپوننت‌های مختلف وجود دارد (مانند اطلاعات کاربر لاگین شده یا وضعیت آزمون فعلی). برای مدیریت متمرکز و قابل پیش‌بینی این داده‌ها، از کتابخانه **Pinia** به عنوان ابزار رسمی مدیریت وضعیت برای **Vue 3** استفاده شده است. **Pinia** یک "انبار"<sup>۲</sup> مرکزی فراهم می‌کند که کامپوننت‌ها می‌توانند داده‌های مورد نیاز خود را از آن خوانده و یا در آن بنویسند، بدون آنکه نیاز به پاس دادن داده‌ها از طریق زنجیره‌های طولانی کامپوننت‌ها باشد.

### ۳-۳-۳- معماری سمت سرور (Backend)

معماری بخش بک‌اند با هدف ارائه یک سرویس سریع، امن و قابل نگهداری طراحی شده است. این بخش به عنوان یک **API** یکپارچه عمل می‌کند که تمام منطق تجاری پروژه را در خود جای داده است.

- معماری **API** مبتنی بر **Fastify**:

<sup>۱</sup> RESTful Principles

<sup>۲</sup> Store



منطق اصلی سرور در قالب یک API مبتنی بر فریم‌ورک **Fastify** پیاده‌سازی شده است. معماری این API به صورت **مسیر-محور**<sup>۱</sup> است. بدین معنا که عملکردها بر اساس منابع مختلف (مانند کاربران، تست‌ها، دستاوردها)<sup>۲</sup> در فایل‌های مسیر جداگانه (test.routes.ts, achievement.routes.ts) سازماندهی شده‌اند. هر مسیر به یک کنترلر یا سرویس متصل است که وظیفه اجرای منطق خاص آن درخواست، مانند اعتبارسنجی ورودی‌ها و فراخوانی لایه‌های پایین‌تر را بر عهده دارد. این تفکیک مسئولیت، خوانایی و نگهداری کد را بهبود می‌بخشد.

● لایه دسترسی به داده<sup>۳</sup> با Prisma ORM :

برای جداسازی کامل منطق برنامه از نحوه تعامل با پایگاه داده، یک لایه دسترسی به داده با استفاده از **Prisma ORM** ایجاد شده است. Prisma به عنوان یک واسطه میان برنامه و پایگاه داده **PostgreSQL** عمل می‌کند. به جای نوشتن کوئری‌های خام SQL، منطق برنامه از کلاینت تایپ-سیف پریزما برای انجام عملیات **CRUD**<sup>۴</sup> استفاده می‌کند. این رویکرد نه تنها امنیت را در برابر حملات SQL Injection افزایش می‌دهد، بلکه تضمین می‌کند که تمام تعاملات با پایگاه داده با اسکیمای تعریف‌شده در فایل schema.prisma سازگار است.

### ۳-۲-۲- نمودار فعالیت‌ها<sup>۵</sup>

نمودار فعالیت نشان‌دهنده‌ی جریان کار از زمان ورود کاربر تا پایان یک فرآیند مشخص است. در این پروژه، فعالیت کاربر برای انجام تست سرعت تایپ به صورت گام به گام مدل‌سازی شده و مراحل مختلف مانند دریافت پاراگراف، شروع تایمر، ورود داده‌ها، و ثبت نتیجه در این نمودار گنجانده شده‌اند. این جریان کاری در شکل ۳-۵ قابل مشاهده است.

<sup>۱</sup> Route-based

<sup>۲</sup> achievement

<sup>۳</sup> Data Access Layer

<sup>۴</sup> Create, Read, Update, Delete

<sup>۵</sup> Activity Diagram



### ۳-۴-۱- مدل موجودیت-رابطه<sup>۱</sup> (ER Diagram)

مدل مفهومی داده‌ها با استفاده از نمودار موجودیت-رابطه (ER) طراحی شده است تا دیدی کلان از داده‌ها و ارتباطات میان آن‌ها ارائه دهد.

#### • تشریح موجودیت‌های اصلی:

- **User**: نماینده کاربر ثبت‌نام کرده در سیستم است و اطلاعات هویتی و پروفایل او را نگهداری می‌کند. این موجودیت علاوه بر اطلاعات هویتی، دارای یک فیلد نقش<sup>۲</sup> برای تعیین سطح دسترسی (کاربر عادی یا مدیر) است.
- **Lesson**: نمایانگر یک درس آموزشی مشخص است و شامل محتوای متنی و ترتیب آن درس می‌باشد.
- **TestResult**: یک نتیجه ثبت‌شده از آزمون تایپ کاربر را نشان می‌دهد و شامل معیارهای عملکردی مانند WPM و دقت است.
- **Achievement**: یک دستاورد یا نشان افتخار قابل کسب در سامانه را تعریف می‌کند (مانند رسیدن به سرعت مشخص).

#### • تشریح روابط بین موجودیت‌ها:

- رابطه یک-به-چند (**One-to-Many**) بین User و TestResult: هر کاربر می‌تواند نتایج تست‌های متعددی داشته باشد، اما هر نتیجه تست تنها متعلق به یک کاربر است.
- رابطه چند-به-چند<sup>۳</sup> بین کاربر و دستاوردها: هر کاربر می‌تواند دستاوردهای متعددی کسب کند و هر دستاورد نیز می‌تواند توسط کاربران متعددی به دست آید. این رابطه از طریق یک جدول واسطه<sup>۴</sup> پیاده‌سازی می‌شود.

### ۳-۲-۲- نمودار ERD (نمودار رابطه‌ای موجودیت‌ها)

نمودار ERD ساختار پایگاه داده پروژه را نمایش می‌دهد. این نمودار ارتباط بین موجودیت‌هایی نظیر

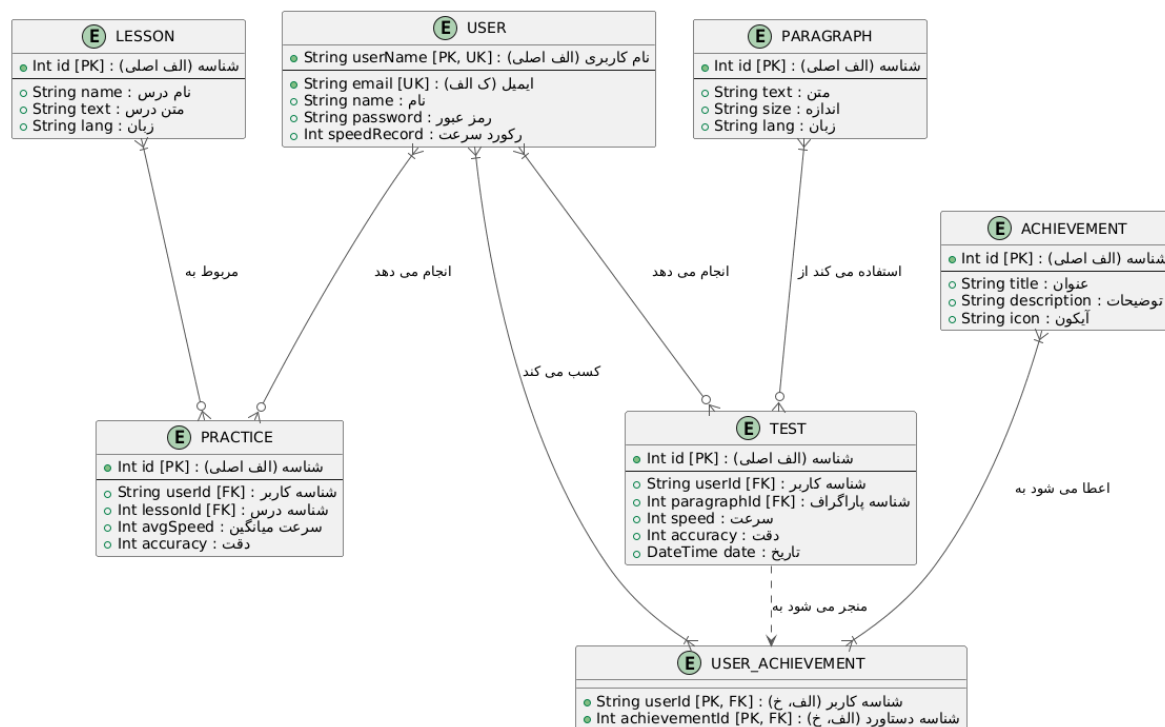
<sup>۱</sup> Entity-Relationship Model

<sup>۲</sup> role

<sup>۳</sup> Many-to-Many

<sup>۴</sup> Join Table

کاربران، تمرین‌ها، تست‌ها، درس‌ها و دستاوردها را به تفصیل مشخص کرده و کلیدهای اصلی و خارجی در آن لحاظ شده‌اند. شکل ۳-۶، این نمودار را که شامل تمام موجودیت‌های اصلی و روابط بین آن‌هاست، به تفصیل نشان می‌دهد.



شکل ۳-۶: نمودار موجودیت-رابطه (ERD) پایگاه داده سامانه

### ۳-۴-۲- طراحی شمای جداول (Table Schema)

بر اساس مدل ER، شمای منطقی جداول اصلی پایگاه داده به شرح زیر طراحی شده است:

#### • جدول users:

شامل فیلدهای username، password\_hash، email، id و همچنین فیلد role (از نوع شمارشی با مقادیر USER و ADMIN)

#### • جدول test\_results:

شامل فیلدهای id (کلید اصلی)، wpm (عدد صحیح)، accuracy (عدد اعشاری)، timestamp (زمان ثبت تست) و user\_id (کلید خارجی با ارجاع به جدول users).

### ۳-۵- طراحی رابط کاربری (UI/UX Design)

طراحی رابط و تجربه کاربری (UI/UX) با هدف ایجاد یک محیط ساده، جذاب و بدون حواس‌پرتی برای کاربر انجام شده است تا تمام تمرکز او بر روی فرآیند یادگیری و تمرین معطوف باشد. فلسفه اصلی طراحی، ارائه بازخورد فوری و تشویق کاربر از طریق عناصر بصری است. این فرآیند شامل طراحی وایرفریم‌ها برای صفحات کلیدی و ترسیم فلوچارت‌های گردش کار برای فرآیندهای اصلی سامانه بوده است.

#### ۳-۵-۱- وایرفریم‌ها<sup>۱</sup> و پروتوتایپ صفحات کلیدی

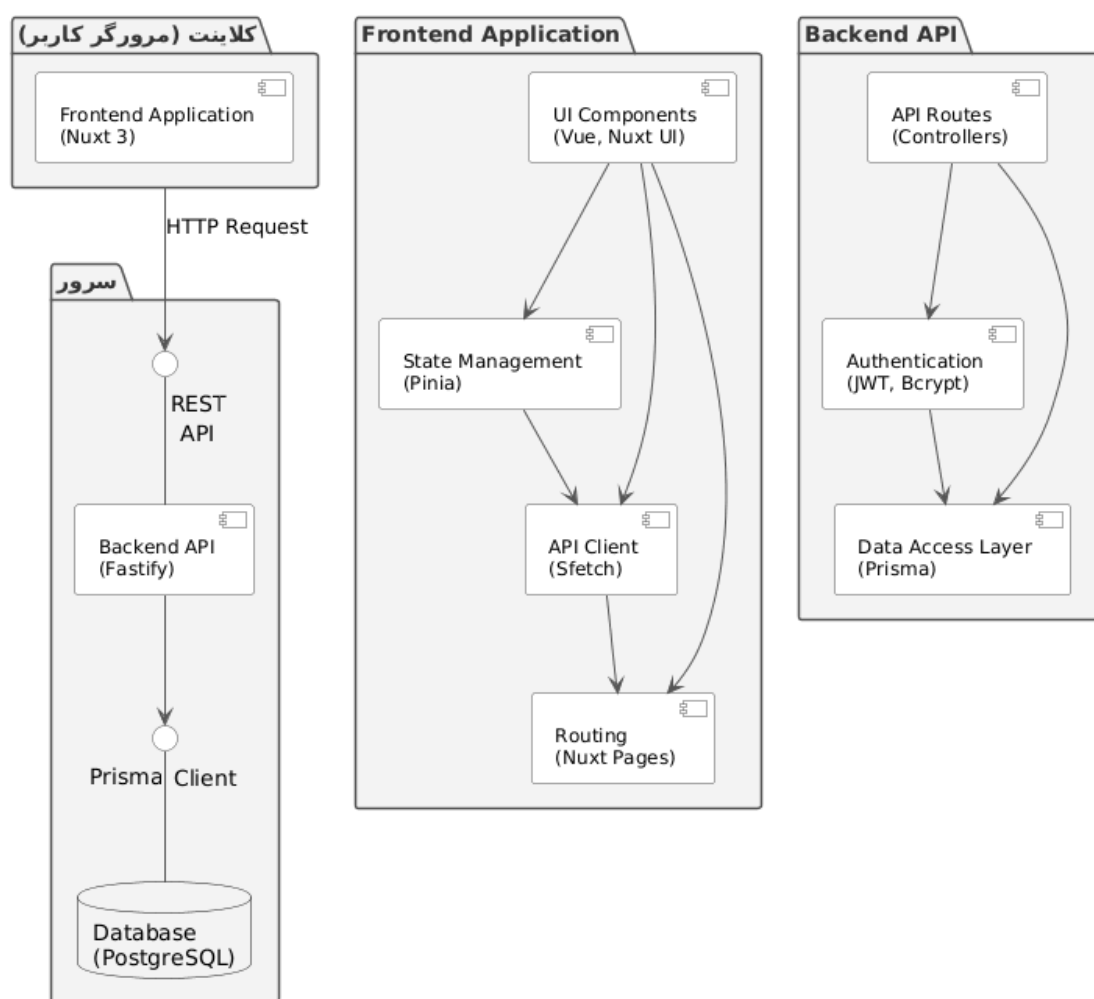
وایرفریم‌ها به عنوان طرح‌های اولیه و اسکلت‌بندی صفحات، ساختار کلی، چیدمان عناصر و سلسله‌مراتب اطلاعات را بدون درگیر شدن در جزئیات بصری مانند رنگ و فونت مشخص می‌کنند. در این پروژه، برای صفحات اصلی و کلیدی سامانه، وایرفریم‌هایی طراحی شده است تا جریان کاربری و محل قرارگیری هر عنصر مشخص گردد.

- **صفحه اصلی (داشبورد):** این صفحه به عنوان نقطه ورود کاربر پس از لاگین طراحی شده است. وایرفریم آن شامل بخش‌هایی برای نمایش آمار کلی کاربر (مانند میانگین سرعت و دقت)، یک کارت برای دسترسی سریع به "درس بعدی" و بخشی برای نمایش آخرین "دستاوردهای" کسب‌شده است.
- **صفحه درس‌ها و صفحه انجام تست:** طراحی این دو صفحه به دلیل ماهیت مشابه، بسیار نزدیک به هم است. تمرکز اصلی در طراحی آن‌ها، بر روی حداقل کردن حواس‌پرتی‌ها و معطوف کردن تمام توجه کاربر به متن و کیبورد مجازی است. طرح اولیه شامل یک بخش بزرگ برای نمایش متن، یک کیبورد راهنمای تصویری برای نمایش انگشتان، و نمایشگرهای زنده برای معیارهایی مانند زمان، سرعت و خطاها می‌باشد.
- **صفحه پروفایل کاربر و گزارش پیشرفت:** این صفحه به منظور ارائه یک نمای کامل از عملکرد و فعالیت‌های کاربر طراحی شده است. وایرفریم آن شامل سه بخش اصلی است: بخشی برای ویرایش اطلاعات شخصی، جدولی برای نمایش تاریخچه کامل نتایج تست‌ها، و یک بخش بصری شامل نمودار خطی برای نمایش روند پیشرفت سرعت و دقت در طول زمان.
- **صفحات پنل مدیریت:** شامل صفحاتی برای نمایش لیستی از کاربران، درس‌ها و پاراگراف‌ها به صورت جدول، همراه با فرم‌هایی برای ایجاد یا ویرایش هر یک از این موارد.

<sup>۱</sup> Wireframes

### ۳-۵-۱- وایرفریم‌ها و پروتوتایپ صفحات کلیدی

نمودار کامپوننت نمای کلی از اجزای نرم‌افزار را در دو بخش فرانت‌اند و بک‌اند نشان می‌دهد. این نمودار شامل بخش‌هایی مانند کامپوننت‌های UI، مدیریت وضعیت، روتینگ و کلاینت API در سمت کلاینت و لایه‌های کنترلر، احراز هویت و پایگاه داده در سمت سرور است. نمای کلی از اجزای نرم‌افزار و ارتباطات داخلی آن‌ها در نمودار اجزا (شکل ۳-۷) نمایش داده شده است.



شکل ۳-۷: استقرار اجزای سامانه روی زیرساخت

### ۳-۵-۲- فلوچارت گردش کار کاربر<sup>۱</sup>

برای درک بهتر و طراحی دقیق‌تر تعاملات کاربر با سیستم، فلوچارت‌هایی برای فرآیندهای اصلی ترسیم شده است. این فلوچارت‌ها مسیر گام‌به‌گام کاربر برای رسیدن به یک هدف مشخص را نمایش می‌دهند.

<sup>۱</sup> User Flow

- **فلوچارت فرآیند ثبت نام تا ورود:** این فلوچارت، مسیر کاربر جدید را از لحظه ورود به سایت، کلیک بر روی دکمه ثبت نام، تکمیل فرم، ارسال اطلاعات، اعتبارسنجی سمت سرور و در نهایت ورود موفق به داشبورد شخصی‌اش ترسیم می‌کند. همچنین مسیر ورود کاربر قبلی و فرآیند بازیابی رمز عبور نیز در این گردش کار در نظر گرفته شده است.
- **فلوچارت فرآیند شروع یک تست تا مشاهده نتیجه:** این گردش کار، مسیر کاربر را از انتخاب یک آزمون، بارگذاری صفحه تست، فرآیند تایپ کردن متن، ارسال خودکار نتایج پس از اتمام زمان، پردازش داده‌ها در بک‌اند و در نهایت، نمایش صفحه نتایج شامل آمار کامل عملکرد به کاربر را به تصویر می‌کشد.

### ۳-۵-۲- فلسفه و رویکرد طراحی

طراحی رابط کاربری این سامانه صرفاً یک تلاش برای زیباسازی بصری نیست؛ بلکه یک رویکرد مهندسی‌شده برای بهینه‌سازی فرآیند یادگیری و تمرین است. از آنجایی که هدف اصلی سامانه، آموزش یک مهارت شناختی-حرکتی (تایپ لمسی) است، فلسفه طراحی بر پایه دو اصل کلیدی بنا شده است: تمرکز حداکثری و بازخورد آنی.

- رویکرد کاربر-محور<sup>۱</sup>:

در مرکز تمام تصمیمات طراحی، کاربر و نیازهای او قرار دارد. هدف اصلی، ایجاد یک تجربه کاربری روان و بدون مانع است تا کاربر بتواند تمام انرژی ذهنی خود را بر روی یادگیری و تمرین متمرکز کند. این به معنای حذف تمام عناصر بصری زائد، ساده‌سازی ناوبری و اطمینان از این است که قابلیت‌های اصلی سامانه به راحتی در دسترس باشند.

- اهمیت یک نمای ظاهری خوب طراحی شده:

جذب و نگهداری کاربر: یک رابط کاربری تمیز و مدرن، کاربر را به کاوش و استفاده مستمر از سامانه ترغیب می‌کند.

- تعامل بهینه:

طراحی شهودی باعث می‌شود کاربر بدون نیاز به فکر کردن، بداند چگونه از سیستم استفاده کند. این امر به ویژه در یک ابزار آموزشی که نیازمند تمرین‌های مکرر است، اهمیت حیاتی دارد.

---

<sup>۱</sup> User-Centric Approach

• دیدگاه مثبت:

یک رابط کاربری مرتب و زیبا، حس اعتماد و حرفه‌ای بودن را به کاربر منتقل می‌کند و نشان‌دهنده توجه به جزئیات در توسعه آن است.

### ۳-۵-۲- عناصر طراحی بصری

فونت:

زبانی که بیشتر در وب‌سایت استفاده می‌شود، زبان فارسی است. برای متون فارسی، فونت Vazirmatn انتخاب شده است. این فونت به دلیل خوانایی بسیار بالا در اندازه‌های مختلف، ساختار منظم و طراحی مدرن، گزینه‌ای ایده‌آل برای محتوای متنی طولانی (مانند پاراگراف‌های تمرین) است. خوانایی بالا به کاربر کمک می‌کند تا بدون خستگی چشم، برای مدت طولانی‌تری به تمرین بپردازد. وزن‌های مختلف این فونت در شکل ۳-۸ نشان داده شده است.

Thin	سکولاریسم ملیگرایانه
ExtraLight	سکولاریسم ملیگرایانه
Light	سکولاریسم ملیگرایانه
Regular	سکولاریسم ملیگرایانه
Medium	سکولاریسم ملیگرایانه
SemiBold	سکولاریسم ملیگرایانه
Bold	سکولاریسم ملیگرایانه
ExtraBold	سکولاریسم ملیگرایانه
Black	سکولاریسم ملیگرایانه

شکل ۳-۸: وزن‌های مختلف فونت کلمه

پالت رنگی:

پالت رنگی با توجه به ماهیت آموزشی و تمرکز محور سامانه انتخاب شده است. رنگ‌های اصلی به



گونه‌ای هستند که ضمن ایجاد یک محیط آرام و جذاب، از ایجاد حواس‌پرتی برای کاربر جلوگیری کنند.

رنگ پس‌زمینه اصلی: یک رنگ خنثی و روشن (مانند #F7F9FC) برای جلوگیری از خستگی چشم.

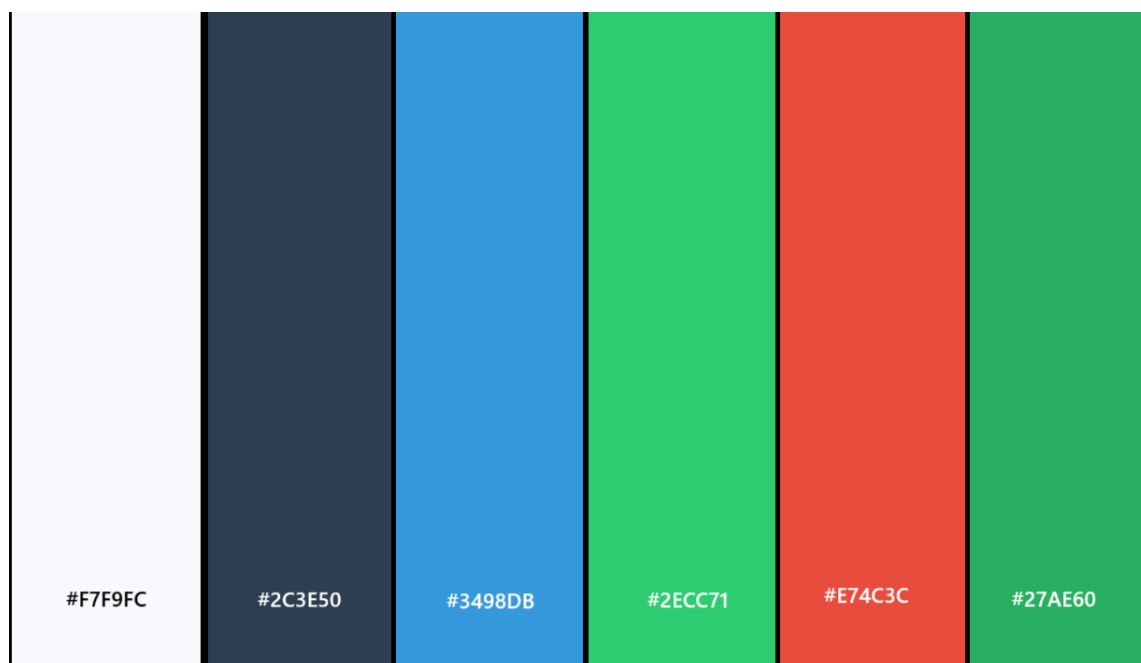
رنگ متن اصلی: خاکستری تیره (مانند #C3E50۲) برای حداکثر خوانایی.

رنگ تأکید<sup>۱</sup>: یک رنگ پرانرژی اما ملایم (مانند آبی #DB۳۴۹۸ یا سبز #ECC71۲) برای دکمه‌ها،

لینک‌ها و عناصر تعاملی.

رنگ‌های بازخورد: سبز (#AE60۲۷) برای نمایش کاراکترهای صحیح و قرمز (#E74C3C) برای

نمایش خطاهای تایپی. این پالت رنگی در شکل ۳-۹ به صورت بصری نمایش داده شده است.



شکل ۳-۹: پالت استفاده شده در سامانه

### ۳-۶- طراحی API

رابط برنامه‌نویسی کاربردی (API)، به عنوان پل ارتباطی میان کلاینت (فرانت‌اند) و سرور (بک‌اند)، نقش حیاتی در عملکرد صحیح سامانه دارد. طراحی یک API خوش‌ساختار، خوانا و کارآمد، برای توسعه و نگهداری آسان پروژه ضروری است.

<sup>۱</sup> Accent Color

### ۳-۶-۱- اصول طراحی (استفاده از اصول RESTful)

طراحی API این پروژه بر پایه اصول و الگوهای RESTful انجام شده است. این رویکرد به معنای استفاده از استانداردهای وب برای ساخت یک API قابل پیش‌بینی و مقیاس‌پذیر است. مهم‌ترین اصول رعایت‌شده عبارتند از:

- استفاده از متدهای استاندارد HTTP (مانند GET برای خواندن، POST برای ایجاد، PATCH/PUT برای به‌روزرسانی و DELETE برای حذف).
- آدرس‌دهی مبتنی بر منابع (Resource-Based URLs) که در آن هر URL به یک منبع مشخص اشاره دارد (مانند `api/users/` یا `api/tests/`).
- استفاده از فرمت JSON برای تبادل داده‌ها بین کلاینت و سرور.

### ۳-۶-۲- تشریح مسیرهای Endpoints اصلی

در ادامه، مسیرهای<sup>۱</sup> اصلی واسط برنامه‌نویسی کاربردی که برای پیاده‌سازی قابلیت‌های کلیدی سامانه طراحی شده‌اند، در جدول ۳-۱ تشریح می‌شوند. این جدول، متد HTTP، آدرس مسیر، توضیحات عملکرد و سطح دسترسی مورد نیاز برای هر Endpoint را به تفکیک مشخص می‌کند.

نیازمند احراز هویت	توضیحات عملکرد	(Endpoint) مسیر	متد HTTP
عمومی	ثبت‌نام یک کاربر جدید در سامانه.	<code>/api/auth/register</code>	POST
عمومی	احراز هویت کاربر و صدور توکن.	<code>/api/auth/login</code>	POST
کاربر (JWT)	ثبت نتیجه یک آزمون جدید توسط کاربر لاگین کرده.	<code>/api/tests</code>	POST
کاربر (JWT)	دریافت تاریخچه نتایج تمام تست‌های کاربر لاگین کرده.	<code>/api/tests/history</code>	GET
کاربر (JWT)	دریافت لیست دستاوردهای کسب‌شده توسط کاربر لاگین کرده.	<code>/api/achievements</code>	GET
(ADMIN JWT)	دریافت لیست تمام کاربران سامانه.	<code>/api/admin/users</code>	GET
مدیر اصلی	دریافت لیست تمام کاربران سامانه.	<code>/api/admin/users/:id/role</code>	PUT
(ADMIN JWT)	ویرایش یک درس آموزشی موجود.	<code>/api/admin/lessons</code>	POST
(ADMIN JWT)	ویرایش یک درس آموزشی موجود.	<code>/api/admin/lessons/:id</code>	PUT
(ADMIN JWT)	حذف یک درس آموزشی.	<code>/api/admin/lessons/:id</code>	DELETE

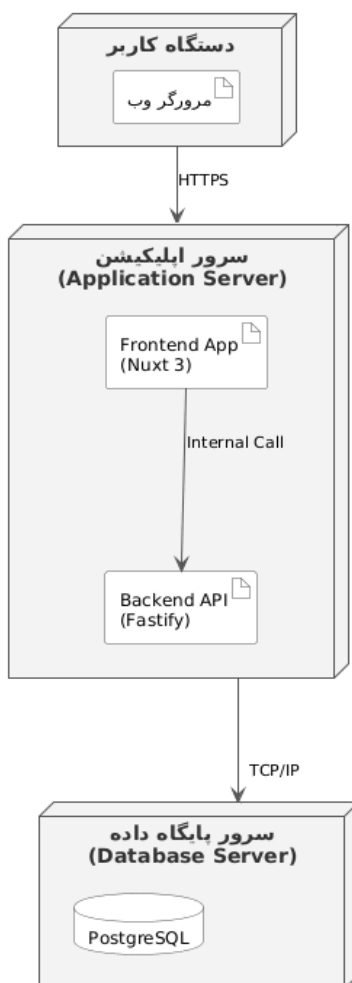
<sup>۱</sup> Endpoints

- **مسیرهای مربوط به کاربران (Authentication):**
  - `post /api/auth/register`: این Endpoint برای ثبت نام کاربر جدید استفاده می شود. بدنه درخواست شامل اطلاعات کاربر (نام، ایمیل، رمز عبور) بوده و در صورت موفقیت، پیام ثبت نام موفق را باز می گرداند.
  - `post /api/auth/login`: برای احراز هویت کاربر و ورود به سیستم طراحی شده است. بدنه درخواست شامل ایمیل و رمز عبور بوده و در صورت صحت اطلاعات، یک توکن JWT به کلاینت بازگردانده می شود.
- **مسیرهای مربوط به تست ها:**
  - `post /api/tests`: پس از اتمام یک آزمون توسط کاربر، نتایج (شامل wpm، دقت و...) از طریق این Endpoint به سرور ارسال و در پایگاه داده ذخیره می شود.<sup>۷</sup>
  - `get /api/tests/history`: برای دریافت تاریخچه نتایج تمام تست های یک کاربر استفاده می شود و لیستی از رکوردهای ثبت شده را باز می گرداند.
- **مسیرهای مربوط به دستاوردها:**
  - `get /api/achievements`: این Endpoint برای دریافت لیست دستاوردهای کسب شده توسط کاربر لاگین کرده استفاده می شود.
- **مسیرهای مربوط به مدیریت (Admin API) - نیازمند احراز هویت با نقش مدیر:**
  - `get /api/admin/users`: دریافت لیست تمام کاربران.
  - `put /api/admin/users/:id/role`: تغییر نقش یک کاربر (تنها توسط مدیر اصلی قابل دسترسی است).
  - `post /api/admin/lessons`: ایجاد یک درس جدید.
  - `put /api/admin/lessons/:id`: ویرایش یک درس موجود.
  - `delete /api/admin/lessons/:id`: حذف یک درس.

### ۳-۶-۳- نمودار استقرار<sup>۱</sup>

نمودار استقرار نمایش دهنده استقرار فیزیکی اجزای سیستم است. این نمودار نشان می دهد که نرم افزار چگونه روی زیرساخت ها مانند مرورگر کاربر، سرور اپلیکیشن، و سرور دیتابیس مستقر شده و از طریق پروتکل های ارتباطی با یکدیگر تعامل دارند. همانطور که در شکل ۳-۱۰ مشاهده می شود، این استقرار شامل سه لایه اصلی است.

<sup>۱</sup> Deployment Diagram

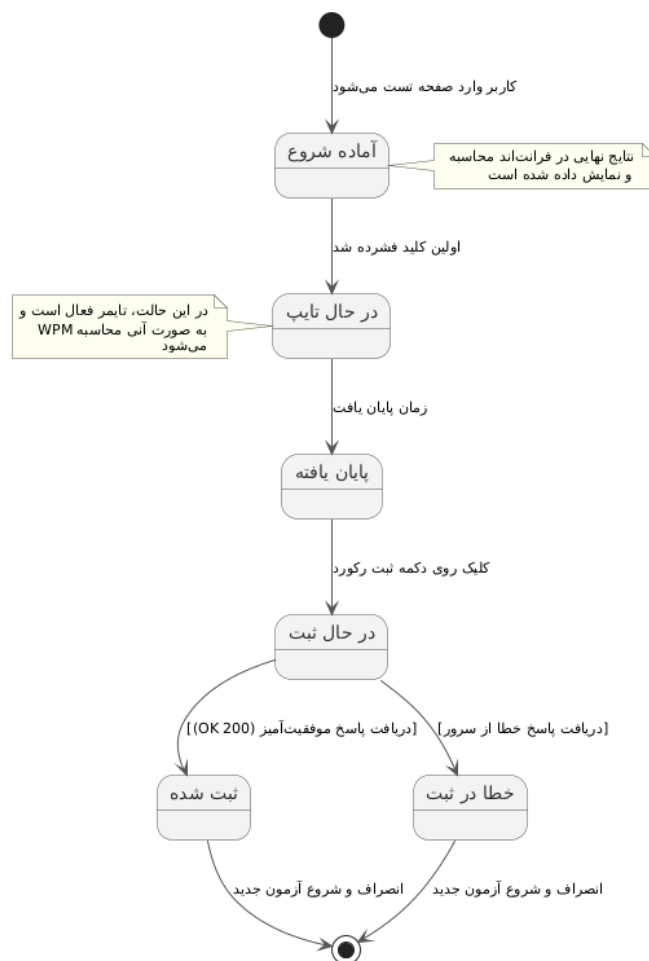


شکل ۳-۱۰: نمای کلی معماری کلاینت-سرور سامانه

### ۳-۶-۳- نمودار ماشین حالات<sup>۱</sup>

نمودار ماشین حالات وضعیت‌های مختلف یک موجودیت (مثلاً تست تایپ یا حساب کاربری) را در طی عمر خود نشان می‌دهد. این نمودار به درک بهتر رفتار سیستم در مواجهه با تغییرات وضعیت کمک می‌کند. این فرآیند برای سناریوی تست تایپ در شکل ۳-۱۱ مدل‌سازی شده است.

<sup>۱</sup> State Machine



شکل ۳-۱۱: ماشین حالات برای فرایند تست تایپ

### ۷-۳- جمع‌بندی

در این فصل، طرح جامع و چارچوب فنی سامانه آموزش تایپ ده‌انگشتی به تفصیل ارائه گردید. فرآیند طراحی با تعریف دقیق نیازمندی‌های عملکردی و غیرعملکردی آغاز شد تا اهداف و محدودیت‌های کیفی پروژه به وضوح مشخص شوند. بر این اساس، یک معماری مدرن کلاينت-سرور به عنوان ساختار اصلی انتخاب شد که در آن، مسئولیت‌های رابط کاربری و منطق سرور به طور کامل از یکدیگر تفکیک شده‌اند. در ادامه، ساختار پایگاه داده با ارائه مدل موجودیت-رابطه و شمای جداول اصلی طراحی شد. همچنین، با بررسی وایرفریم‌ها و فلوچارت‌های گردش کار، جنبه‌های مختلف طراحی تجربه کاربری مشخص گردید و در نهایت، طراحی API به عنوان پل ارتباطی میان دو بخش سیستم، با تشریح اصول RESTful و Endpoints اصلی آن نهایی شد. این طرح جامع، زمینه را برای ورود به فاز پیاده‌سازی که در فصل بعدی به آن پرداخته خواهد شد، به طور کامل فراهم می‌آورد.

## فصل چهارم

### پیاده‌سازی سامانه

#### ۴-۱- مقدمه

این فصل به قلب فنی پروژه، یعنی فرآیند پیاده‌سازی، اختصاص دارد. پس از تعریف نیازمندی‌ها و طراحی معماری در فصول گذشته، در این بخش، طرح‌ها و نقشه‌های فنی به کد واقعی و یک سامانه کاربردی تبدیل می‌شوند. این فصل به صورت دقیق، جزئیات و چگونگی ساخت دو بخش اصلی سامانه، یعنی فرانت‌اند و بک‌اند را تشریح می‌کند. هدف، ارائه یک تصویر شفاف از نحوه استفاده از تکنولوژی‌های منتخب برای تحقق اهداف پروژه و تبدیل طرح‌های مفهومی به قابلیت‌های عملی است. در ادامه، ابتدا به شرح پیاده‌سازی بخش فرانت‌اند و سپس بخش بک‌اند خواهیم پرداخت.

#### ۴-۲- پیاده‌سازی بخش فرانت‌اند (Vue.js)

بخش فرانت‌اند سامانه با استفاده از فریم‌ورک Nuxt 3 که بر پایه Vue 3 است، توسعه یافته است. این انتخاب به منظور بهره‌گیری از یک محیط توسعه مدرن، سریع و با قابلیت رندر سمت سرور (SSR) صورت گرفته است. در ادامه، جزئیات پیاده‌سازی ماژول‌های مختلف این بخش تشریح می‌گردد.

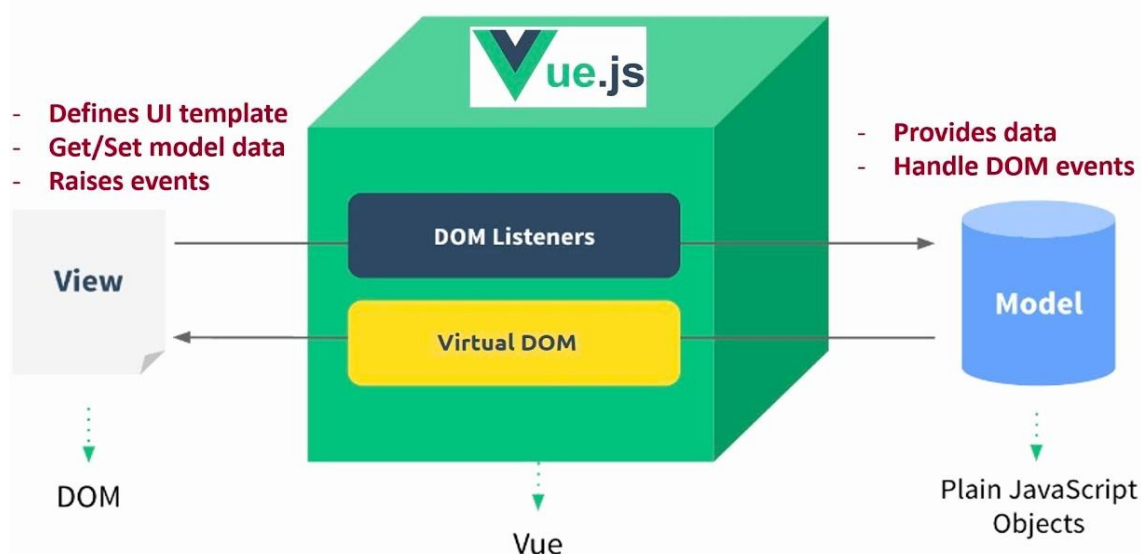
#### • سازوکار واکنشی و نقش DOM مجازی<sup>۱</sup> در Vue.js

یکی از دلایل اصلی عملکرد بالا و تجربه کاربری روان در اپلیکیشن‌های مدرن مبتنی بر Vue، سازوکار هوشمندانه آن در مدیریت تغییرات رابط کاربری است. همانطور که در شکل ۴-۱ نمایش داده شده است.

---

<sup>۱</sup> Virtual DOM

## Architecture of Vue.js app



شکل ۴-۱: معماری واکنشی Vue.js و نقش DOM مجازی

در این معماری:

مدل<sup>۱</sup>: شامل داده‌های خام برنامه است که به صورت اشیاء ساده جاوا اسکریپت<sup>۲</sup> نگهداری می‌شوند. این بخش مسئولیت تأمین داده‌ها و منطق برنامه را بر عهده دارد.

• نما<sup>۳</sup>: همان رابط کاربری نهایی است که کاربر در مرورگر خود مشاهده و با آن تعامل می‌کند. وظیفه آن نمایش داده‌ها و دریافت ورودی‌های کاربر (مانند کلیک یا تایپ) است.

هسته Vue.js به عنوان یک پل هوشمند بین این دو عمل می‌کند. هنگامی که کاربر با View تعامل می‌کند، DOM Listeners این رویدادها را دریافت کرده و Model را به‌روزرسانی می‌کنند. نکته کلیدی و وجه تمایز Vue در مرحله بعدی رخ می‌دهد: به جای دستکاری مستقیم و پرهزینه DOM واقعی، Vue ابتدا تغییرات را بر روی یک کپی سبک و مجازی از DOM که در حافظه نگهداری می‌شود و به آن DOM مجازی (Virtual DOM) می‌گویند، اعمال می‌کند.

سپس، Vue با استفاده از یک الگوریتم بهینه‌سازی شده، تفاوت بین نسخه جدید و قدیم DOM مجازی را محاسبه کرده (فرآیندی که به آن "Diffing" می‌گویند) و تنها و تنها همان تغییرات حداقلی و ضروری را به

<sup>۱</sup> Model

<sup>۲</sup> Plain JavaScript Objects

<sup>۳</sup> View

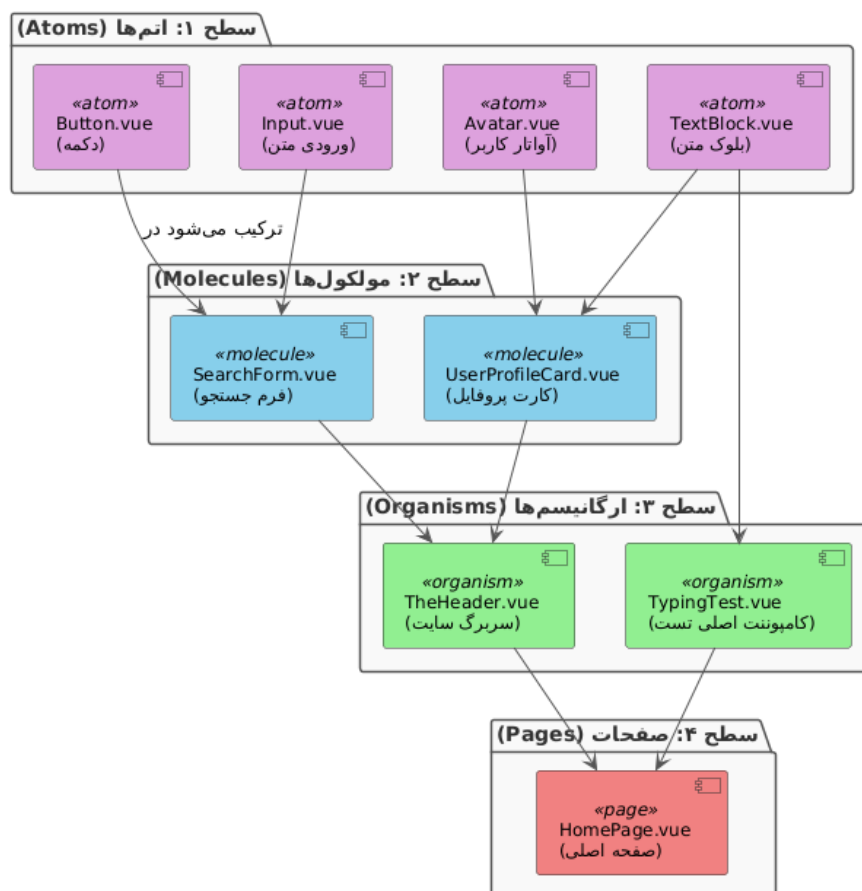
صورت یکجا بر روی DOM واقعی اعمال می‌کند. این سازوکار، تعداد عملیات باز (Re-rendering) و دستکاری‌های سنگین در مرورگر را به شدت کاهش داده و منجر به عملکردی بسیار سریع و روان می‌شود، به ویژه در اپلیکیشن‌های پیچیده‌ای مانند سامانه‌ها که نیازمند بازخوردهای آنی و مکرر به کاربر هستند.

#### ۴-۲-۱- ساختار پروژه فرانت‌اند و کامپوننت‌های اصلی

##### • معماری مبتنی بر کامپوننت در Nuxt 3

هسته اصلی رابط کاربری این پروژه بر پایه معماری کامپوننت-محور (Component-Based) بنا شده است. این رویکرد که توسط Vue 3 و فریم‌ورک Nuxt 3 به بهترین شکل پیاده‌سازی می‌شود، به ما اجازه می‌دهد تا رابط کاربری را به قطعات کوچک، مستقل و قابل استفاده مجدد به نام کامپوننت تقسیم کنیم. هر کامپوننت منطق و ظاهر مربوط به خود را کپسوله می‌کند.

برای نمایش بهتر این ساختار، از متدولوژی طراحی Atomic Design الهام گرفته شده است. همانطور که در شکل ۴-۲ مشاهده می‌شود، رابط کاربری از ترکیب لایه‌لایه کامپوننت‌ها در سطوح مختلف شکل می‌گیرد:



شکل ۴-۲: نمودار مفهومی معماری کامپوننت‌ها بر اساس متدولوژی Atomic Design



- **سطح ۱: اتم‌ها<sup>۱</sup>:** این‌ها کوچکترین و پایه‌ای‌ترین اجزای رابط کاربری هستند که به تنهایی قابل تفکیک نیستند. در پروژه ما، کامپوننت‌هایی مانند Button.vue (دکمه)، Input.vue (ورودی متن) و Avatar.vue (آواتار کاربر) در این سطح قرار می‌گیرند.
- **سطح ۲: مولکول‌ها<sup>۲</sup>:** از ترکیب چند اتم، یک کامپوننت کاربردی‌تر و پیچیده‌تر به نام مولکول ساخته می‌شود. به عنوان مثال، کامپوننت SearchForm.vue (فرم جستجو) از ترکیب یک اتم Input.vue و یک اتم Button.vue تشکیل شده است.
- **سطح ۳: ارگانیسم‌ها<sup>۳</sup>:** بخش‌های بزرگتر و مستقل رابط کاربری که از ترکیب مولکول‌ها و/یا اتم‌ها ساخته می‌شوند، در این سطح قرار دارند. کامپوننت TheHeader.vue (سربرگ سایت) که شامل لوگو، منوی ناوبری و کارت پروفایل کاربر است، یک نمونه از ارگانیسم می‌باشد.
- **سطح ۴: صفحات<sup>۴</sup>:** در نهایت، با کنار هم قرار دادن ارگانیسم‌ها و کامپوننت‌های دیگر، یک صفحه کامل مانند HomePage.vue (صفحه اصلی) شکل می‌گیرد.

این معماری ماژولار نه تنها مدیریت کدهای پیچیده و توسعه پروژه را آسان‌تر می‌کند، بلکه به دلیل قابلیت استفاده مجدد بالای کامپوننت‌ها، از تکرار کد جلوگیری کرده و پایداری و یکپارچگی کل سامانه را تضمین می‌نماید.

سازماندهی صحیح فایل‌ها و پوشه‌ها، یکی از اصول کلیدی برای توسعه یک پروژه قابل نگهداری است. ساختار پروژه فرانت‌اند بر اساس قراردادهای استاندارد Nuxt 3 به شرح زیر است. نمایی از این ساختار در شکل ۳-۴ قابل مشاهده است.

پوشه pages/: این پوشه مسئولیت مسیریابی (routing) مبتنی بر فایل را بر عهده دارد. صفحات اصلی مانند صفحه اصلی (index.vue)، صفحه درس‌ها (lessons/[id].vue) و پروفایل کاربر در این بخش قرار دارند.

پوشه components/: کامپوننت‌های قابل استفاده مجدد که در صفحات مختلف به کار می‌روند، در این پوشه نگهداری می‌شوند. کامپوننت‌های کلیدی شامل TypingTest.vue برای منطق اصلی آزمون، TheHeader.vue برای سربرگ سایت و AchievementCard.vue برای نمایش دستاوردها می‌باشند.

---

<sup>1</sup> Atoms

<sup>2</sup> Molecules

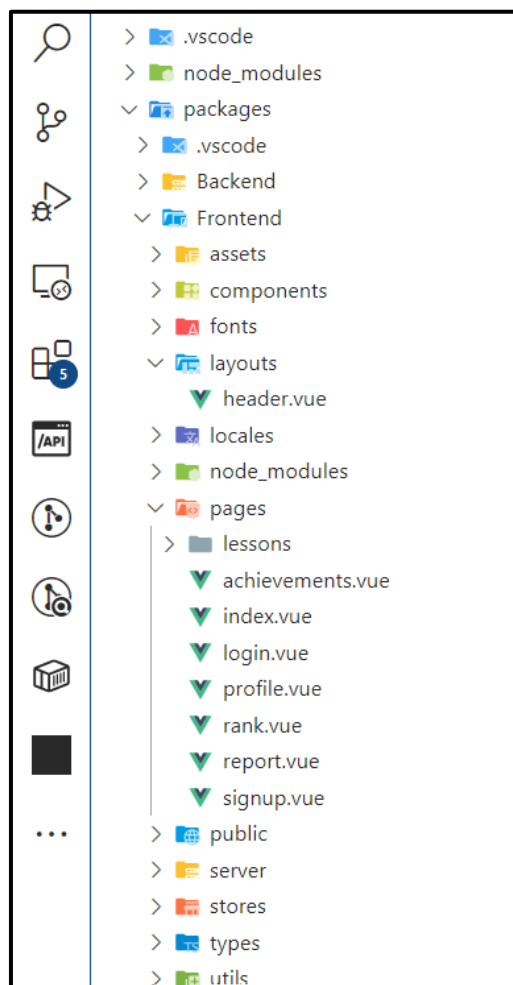
<sup>3</sup> Organisms

<sup>4</sup> Pages

پوشه `stores`: برای مدیریت متمرکز وضعیت برنامه از `Pinia` استفاده شده است. فایل‌های مربوط به هر "انبار" (`Store`) در این پوشه قرار دارند؛ به عنوان مثال، `user.ts` برای نگهداری وضعیت احراز هویت و اطلاعات کاربر و `settings.ts` برای ذخیره تنظیمات شخصی‌سازی کاربر (مانند تم انتخابی).

پوشه `/utils`: این پوشه شامل توابع کمکی و عمومی است که در بخش‌های مختلف پروژه استفاده می‌شوند، مانند تابعی برای محاسبه WPM یا تابعی برای بررسی کسب دستاوردهای جدید (`checkAchievements.ts`).

پوشه `/assets`: داده‌های ایستا و محتوای ثابت پروژه مانند محتوای متنی درس‌ها (`lessons.ts`) و تعریف دستاوردها (`achievements.ts`) در این بخش قرار دارند.



شکل ۳-۴: پوشه بندی و دایرکتوری پروژه

#### ۴-۲-۲- پیاده‌سازی ماژول آموزش تایپ

این ماژول، هسته آموزشی سامانه را تشکیل می‌دهد. در صفحه مربوط به هر درس، محتوای متنی از فایل مربوطه در پوشه `/assets` خوانده شده و به کاربر نمایش داده می‌شود. منطق اصلی این بخش، دریافت ورودی کاربر به ازای هر کلید فشرده شده، مقایسه آن با کاراکتر مورد انتظار و ارائه بازخورد بصری فوری است. کاراکترهای صحیح با رنگ سبز و کاراکترهای اشتباه با رنگ قرمز مشخص می‌شوند تا کاربر به سرعت متوجه خطای خود شود. همچنین، یک کامپوننت کیبورد مجازی، کلید بعدی که باید فشرده شود را برای راهنمایی بهتر کاربر هایلایت می‌کند.

#### ۴-۲-۳- پیاده‌سازی ماژول تست سرعت تایپ

این ماژول با استفاده از کامپوننت `TypingTest.vue` پیاده‌سازی شده است. با شروع آزمون، یک تایمر معکوس فعال شده و منطق محاسبه آنی سرعت (WPM) و دقت بر اساس ورودی‌های کاربر اجرا می‌گردد. این مقادیر به صورت زنده به کاربر نمایش داده می‌شوند تا از عملکرد لحظه‌ای خود آگاه باشد. پس از پایان زمان آزمون، نتایج نهایی (شامل WPM، دقت، تعداد خطاها و...) جمع‌بندی شده و با استفاده از ابزار `fetch$` به Endpoint مربوطه در API بک‌اند (`post/api/tests`) ارسال می‌گردد تا در پروفایل کاربر ثبت شود.

#### ۴-۲-۴- پیاده‌سازی بخش گزارش‌گیری و نمودارها

این بخش در صفحه پروفایل کاربر پیاده‌سازی شده است. با بارگذاری این صفحه، درخواستی از طریق `fetch$` به API بک‌اند (`get/api/tests/history`) ارسال می‌شود تا تاریخچه تمام نتایج تست‌های کاربر دریافت گردد. داده‌های دریافت شده در یک جدول نمایش داده می‌شوند. علاوه بر این، برای ارائه یک دید بصری از پیشرفت، از یک کتابخانه نمودارسازی (مانند `Chart.js`) استفاده شده است تا روند تغییرات WPM و دقت کاربر در طول زمان به صورت یک نمودار خطی ترسیم شود.

#### ۴-۲-۵- پیاده‌سازی مدیریت کاربر و شخصی‌سازی UI

این بخش شامل دو قسمت اصلی است. اول، کامپوننت‌های مربوط به فرم‌های ثبت‌نام و ورود که داده‌های کاربر را دریافت کرده و به API های احراز هویت (`api/auth/register/` و `api/auth/login/`) ارسال می‌کنند. پس

از ورود موفق، توکن JWT دریافت شده و در store مربوط به کاربر در Pinia ذخیره می‌شود تا در تمام درخواست‌های بعدی مورد استفاده قرار گیرد.

دوم، بخش شخصی‌سازی UI است که با استفاده از کتابخانه TailwindCSS و کامپوننت‌های @nuxt/ui پیاده‌سازی شده است. انتخاب‌های کاربر مانند تم رنگی در store تنظیمات ذخیره شده و به صورت پویا به کلاس‌های CSS سراسری اعمال می‌شود تا ظاهر کل سایت مطابق با سلیقه کاربر تغییر کند.

#### ۴-۲-۶- پیاده‌سازی پنل مدیریت

برای بخش مدیریت، یک مجموعه صفحات جدید تحت مسیر admin/ ایجاد شد. دسترسی به این مسیر و تمام زیرمجموعه‌های آن با استفاده از نگهبان مسیر<sup>۱</sup> در Nuxt محافظت می‌شود. این نگهبان، قبل از نمایش صفحه، نقش کاربر را از store پینیا خوانده و تنها در صورتی که کاربر دارای نقش ADMIN باشد، اجازه ورود به این بخش را می‌دهد. رابط کاربری پنل شامل جداولی برای نمایش داده‌ها و فرم‌های مودال برای عملیات ایجاد و ویرایش محتوا می‌باشد.

#### ۴-۳- پیاده‌سازی بخش بک‌اند (Fastify)

بخش بک‌اند (Backend) به عنوان مغز متفکر و هسته پردازشی سامانه، با استفاده از پلتفرم Node.js و فریم‌ورک Fastify پیاده‌سازی شده است. این بخش مسئولیت مدیریت تمام منطق‌های تجاری، ارتباط با پایگاه داده، پردازش درخواست‌ها و تأمین امنیت را بر عهده دارد. انتخاب این پشته فناوری با هدف دستیابی به حداکثر کارایی، سرعت و توسعه‌پذیری صورت گرفته است.

#### ۴-۳-۱- ساختار پروژه بک‌اند و ماژول‌های اصلی

معماری کد در بخش بک‌اند به صورت ماژولار و با تفکیک مسئولیت‌ها<sup>۲</sup> طراحی شده است تا خوانایی و نگهداری آن آسان باشد:

- پوشه **src/routes**: این پوشه قلب مسیریابی API را تشکیل می‌دهد. هر منبع اصلی (resource) در سامانه، مانند تست‌ها یا دستاوردها، فایل مسیریابی مخصوص به خود را دارد (مثلاً `test.routes.ts` یا `achievement.routes.ts`). این فایل‌ها وظیفه تعریف Endpoints، متدهای HTTP مجاز و اتصال آن‌ها به کنترلرهای مربوطه را بر عهده دارند.

<sup>۱</sup> Route Guard

<sup>۲</sup> Separation of Concerns

- **لایه کنترلر و سرویس (Controllers/Services):** برای هر مسیر، یک تابع کنترلر وجود دارد که درخواست (request) و پاسخ (response) را مدیریت می‌کند. این کنترلرها وظیفه اعتبارسنجی ورودی‌ها را داشته و سپس منطق اصلی تجاری را به لایه سرویس واگذار می‌کنند. لایه سرویس مسئول اجرای منطق خالص برنامه است (مثلاً محاسبه دستاوردها پس از یک تست).
- **پوشه prisma:** این پوشه شامل فایل schema.prisma است که به عنوان تنها منبع حقیقت<sup>۱</sup> برای تعریف مدل‌های داده و روابط بین آن‌ها عمل می‌کند. همچنین تمام فایل‌های مایگریشن (Migrations) که تاریخچه تغییرات ساختار پایگاه داده را ثبت می‌کنند، در این پوشه نگهداری می‌شوند.

#### ۴-۳-۲- پیاده‌سازی API ها و منطق سمت سرور

پیاده‌سازی API ها با استفاده از قابلیت‌های Fastify و پلاگین‌های آن انجام شده است. به عنوان مثال، برای پیاده‌سازی post/api/tests، ابتدا یک schema برای اعتبارسنجی بدنه درخواست تعریف می‌شود تا اطمینان حاصل شود که داده‌های ورودی (مانند wpm و accuracy) معتبر هستند. سپس، کنترلر مربوطه این داده‌ها را دریافت کرده و به سرویس TestService ارسال می‌کند. این سرویس، پس از انجام محاسبات لازم (مثلاً بررسی برای کسب دستاورد جدید)، نتیجه را از طریق لایه Prisma در پایگاه داده ذخیره می‌کند و در نهایت، پاسخ موفقیت‌آمیز به کاربر بازگردانده می‌شود. در این فرآیند، از پلاگین‌هایی مانند fastify/cors@ برای مدیریت درخواست‌های فرانت‌اند و fastify/sensible@ برای مدیریت یکپارچه خطاها استفاده شده است.

#### ۴-۳-۳- پیاده‌سازی احراز هویت و امنیت (JWT و Bcrypt)

امنیت به عنوان یکی از ارکان اصلی در این بخش پیاده‌سازی شده است:

- **هش کردن رمز عبور:** در هنگام ثبت‌نام کاربر، رمز عبور دریافتی هرگز به صورت متن ساده ذخیره نمی‌شود. در عوض، با استفاده از کتابخانه bcryptjs، یک "نمک" (salt) تصادفی به آن اضافه شده و سپس حاصل، به صورت یک‌طرفه هش می‌شود. تنها این رشته هش شده در پایگاه داده ذخیره می‌گردد.
- **فرآیند ورود و تولید توکن:** هنگام ورود، کاربر ایمیل و رمز عبور خود را ارسال می‌کند. سرور با استفاده از bcrypt.compare، رمز وارد شده را با هش ذخیره‌شده مقایسه می‌کند. در صورت تطابق، با استفاده از کتابخانه jsonwebtoken، یک توکن JWT جدید حاوی شناسه‌ی کاربر و زمان انقضا، با یک کلید مخفی (secret) امضا و برای کلاینت ارسال می‌شود. این توکن در تمام درخواست‌های بعدی برای احراز هویت کاربر به کار می‌رود.

<sup>۱</sup> Single Source of Truth

#### ۴-۳-۴- تعامل با پایگاه داده (استفاده از Prisma)

تمام تعاملات با پایگاه داده PostgreSQL از طریق **Prisma Client** انجام می‌شود. Prisma Client یک query builder کاملاً تایپ-سیف است که به صورت خودکار بر اساس فایل schema.prisma تولید می‌شود. این ابزار به توسعه‌دهنده اجازه می‌دهد تا با استفاده از توابع شهودی و زنجیره‌ای (مانند `prisma . user . findUnique (...)` یا `prisma . testResult . create (...)`) با پایگاه داده کار کند. این رویکرد نه تنها از خطاهای رایج در نوشتن کوئری‌های SQL جلوگیری می‌کند، بلکه خوانایی کد را به شدت افزایش داده و فرآیند توسعه را تسریع می‌بخشد.

#### ۴-۳-۵- پیاده‌سازی منطق و مسیرهای مدیریت

مسیرهای مربوط به پنل ادمین تحت پیشوند `api/admin/` پیاده‌سازی شدند. برای محافظت از این مسیرها، یک میدل‌ور<sup>۱</sup> سفارشی در Fastify نوشته شد. این میدل‌ور به عنوان یک قلاب<sup>۲</sup> قبل از اجرای کنترلر، توکن JWT درخواست را اعتبارسنجی کرده و اطمینان حاصل می‌کند که ادعای مربوط به `role` در محتوای توکن، برابر با `ADMIN` است. برای قابلیت انحصاری مدیر اصلی (تغییر نقش سایر کاربران)، در کنترلر مربوطه، یک بررسی اضافی انجام می‌شود تا اطمینان حاصل شود که شناسه‌ی کاربری که درخواست را ارسال کرده، با شناسه‌ی مدیر اصلی سایت که در متغیرهای محیطی (`env.`) تعریف شده، یکسان است.

#### ۴-۴- تست و اعتبارسنجی سامانه

برای اطمینان از صحت عملکرد کد، پایداری و جلوگیری از خطاهای ناخواسته در آینده (`Regression`)، تست‌نویسی به عنوان بخشی جدایی‌ناپذیر از فرآیند توسعه در نظر گرفته شده است. در این پروژه، از رویکرد **تست واحد**<sup>۳</sup> برای اعتبارسنجی کوچکترین واحدهای منطقی برنامه به صورت مجزا استفاده شده است.

#### ۴-۴-۱- تست واحد در فرانت‌اند با Vitest

در بخش فرانت‌اند، از فریم‌ورک **Vitest** برای نوشتن تست‌های واحد استفاده شده است. Vitest به دلیل سرعت بالا و یکپارچگی کامل با اکوسیستم Vite (که زیربنای Nuxt 3 است) انتخاب شده است. تست‌ها عمدتاً بر روی موارد زیر متمرکز شده‌اند:

<sup>۱</sup> Middleware

<sup>۲</sup> hook

<sup>۳</sup> Unit Testing

- **توابع کمکی<sup>۱</sup>:** توابع خالص موجود در پوشه `/utils`، مانند تابع محاسبه WPM، با ورودی‌های مشخص تست شده‌اند تا از صحت خروجی آن‌ها اطمینان حاصل شود.
- **منطق Store های Pinia:** اکشن‌ها<sup>۲</sup> و گترهای<sup>۳</sup> تعریف‌شده در انبار های Pinia به صورت مجزا تست شده‌اند. برای مثال، تست می‌شود که آیا پس از اجرای اکشن `login`، وضعیت کاربر به درستی تغییر می‌کند یا خیر.

#### ۴-۲- تست واحد در بک‌اند با Jest

در بخش بک‌اند، از فریم‌ورک **Jest** به همراه **ts-jest** برای تست کدهای تایپ‌اسکریپت استفاده شده است. تمرکز اصلی تست‌ها بر روی منطق تجاری موجود در لایه سرویس بوده است. از آنجایی که تست واحد باید به صورت ایزوله و بدون وابستگی‌های خارجی (مانند پایگاه داده) انجام شود، از تکنیک شبیه‌سازی<sup>۴</sup> به طور گسترده استفاده شده است. به عنوان مثال، برای تست یک سرویس که با پایگاه داده کار می‌کند، به جای اتصال به دیتابیس واقعی، **Prisma Client** شبیه‌سازی شده و پاسخ‌های از پیش تعیین‌شده‌ای را باز می‌گرداند. این کار به ما اجازه می‌دهد تا منطق خود سرویس را بدون درگیر شدن با لایه‌های دیگر، به طور دقیق اعتبارسنجی کنیم.

#### ۴-۵- چالش‌های پیاده‌سازی و راهکارها

در طول فرآیند توسعه هر پروژه نرم‌افزاری، بروز چالش‌های فنی و غیرمنتظره امری طبیعی است. این بخش به بررسی مهم‌ترین موانعی که در مسیر پیاده‌سازی این سامانه با آن‌ها مواجه شدیم و راهکارهای اتخاذ شده برای غلبه بر آن‌ها می‌پردازد.

- **چالش ۱: ارائه بازخورد آنی در رابط کاربری تست**
  - **چالش:** یکی از نیازمندی‌های اصلی، ارائه بازخورد فوری (هایلایت کردن کاراکتر صحیح/غلط و به‌روزرسانی زنده WPM) به کاربر در حین تایپ بود. پیاده‌سازی ساده این قابلیت می‌توانست با هر بار فشردن کلید، منجر به رندر مجدد کامپوننت‌های سنگین و ایجاد تأخیر<sup>۵</sup> در رابط کاربری شود.
  - **راهکار:** برای حل این مشکل، از بهینه‌سازی فرآیند رندر در **Vue** استفاده شد. به جای رندر مجدد کل پاراگراف، متن آزمون به مجموعه‌ای از کامپوننت‌های کوچک‌تر (یک کامپوننت برای هر کاراکتر) تقسیم شد. با این روش، تنها وضعیت کامپوننت کاراکتر فعلی تغییر می‌کند که این امر بار پردازشی را به شدت کاهش داده و تجربه‌ای کاملاً روان و بدون تأخیر را برای کاربر تضمین می‌کند.

<sup>1</sup> Utils functions

<sup>2</sup> actions

<sup>3</sup> getters

<sup>4</sup> Mocking

<sup>5</sup> Lag

## • چالش ۲: مدیریت وضعیت احراز هویت در (SSR) Nuxt 3

- **چالش:** به دلیل استفاده از رندر سمت سرور (SSR) در Nuxt 3، همگام‌سازی وضعیت لاگین کاربر بین سرور و کلاینت یک چالش بود. لازم بود که اگر کاربر یک توکن معتبر داشت، هم رندر اولیه در سرور و هم اپلیکیشن در سمت کلاینت، او را به عنوان کاربر لاگین شده بشناسند.
- **راهکار:** این چالش با استفاده از ترکیب پلاگین‌های Nuxt و **Pinia** حل شد. یک پلاگین در Nuxt نوشته شد که در اولین بارگذاری، توکن JWT را از کوکی‌های درخواست<sup>۱</sup> در سمت سرور می‌خواند. در صورت وجود توکن، اطلاعات اولیه کاربر از API دریافت شده و در store مربوط به کاربر در Pinia مقداردهی اولیه می‌شد. این store سپس به سمت کلاینت منتقل<sup>۲</sup> شده و وضعیت لاگین در هر دو محیط یکپارچه باقی می‌ماند.

## • چالش ۳: پیاده‌سازی پنل مدیریت (Admin Panel)

- **چالش:** با رشد سامانه، نیاز به یک بخش مدیریتی برای کنترل محتوا و کاربران احساس شد. مدیر سیستم باید قادر به افزودن، ویرایش یا حذف درس‌ها و پاراگراف‌های تست، و همچنین مشاهده و مدیریت کاربران باشد. چالش اصلی، ایجاد یک محیط امن و ایزوله برای این قابلیت‌ها بود که از دسترس کاربران عادی خارج باشد، بدون آنکه نیاز به ساخت یک اپلیکیشن کاملاً مجزا باشد.
- **راهکار:** این چالش از طریق یک راهکار یکپارچه در سه لایه حل شد:
  ۱. **سطح پایگاه داده:** یک فیلد role از نوع شمارشی (Enum) به مدل User در فایل schema.prisma اضافه شد که می‌توانست مقادیر USER یا ADMIN را بپذیرد.
  ۲. **سطح بک‌اند (API):** یک مجموعه جدید از API Endpoints تحت پیشوند `/api/admin/...` ایجاد شد (مانند `delete/post/api/admin/lessons` یا `api/admin/users/:id`). سپس، یک **میدل‌ور** سفارشی در Fastify نوشته شد که قبل از اجرای هر یک از این مسیرها، توکن JWT کاربر را بررسی کرده و اطمینان حاصل می‌کند که نقش (role) موجود در محتوای توکن، برابر با ADMIN است. در غیر این صورت، دسترسی با خطای 403 Forbidden رد می‌شود.
  ۳. **سطح فرانت‌اند:** به جای ساخت یک پنل جدا، یک بخش مدیریت در خود اپلیکیشن Nuxt و تحت مسیر `/admin` ایجاد شد. دسترسی به این مسیر و تمام زیرمجموعه‌های آن با استفاده از **نگهبان مسیر (Route Guard)** در Nuxt

<sup>۱</sup> Request Cookies

<sup>۲</sup> hydrate



محافظت می‌شود. این نگهبان، قبل از نمایش صفحه، نقش کاربر را از store پینیا خوانده و تنها در صورتی که کاربر ادمین باشد، اجازه ورود به این بخش را می‌دهد. این رویکرد ضمن تأمین امنیت، باعث استفاده مجدد از کامپوننت‌ها و حفظ یکپارچگی ظاهری سامانه شد.

#### • چالش ۴: پیاده‌سازی روابط چند-به-چند در پایگاه داده

- چالش: طراحی رابطه چند-به-چند بین کاربران و دستاوردها (Users) و (Achievements) در Prisma نیازمند درک دقیق نحوه تعریف جداول واسط (Join Tables) بود.
- راهکار: این رابطه با ایجاد یک مدل صریح به نام UserAchievement در فایل schema.prisma پیاده‌سازی شد. این مدل شامل دو کلید خارجی به جداول User و Achievement بود. سپس در منطق سرویس بک‌اند، تابعی نوشته شد که پس از هر آزمون، عملکرد کاربر را بررسی کرده و در صورتی که شرایط یک دستاورد جدید احراز شده بود، یک رکورد جدید در جدول UserAchievement برای اتصال آن کاربر به آن دستاورد ایجاد می‌کرد.

#### ۴-۶- جمع‌بندی

در این فصل، فرآیند دقیق تبدیل طرح‌های مفهومی فصل سوم به یک سامانه نرم‌افزاری کامل و کاربردی تشریح گردید. این فرآیند با پیاده‌سازی بخش فرانت‌اند با استفاده از فریم‌ورک **Nuxt 3** آغاز شد، که در آن ساختار ماژولار پروژه، کامپوننت‌های اصلی و نحوه پیاده‌سازی ماژول‌های کلیدی مانند آموزش، تست و گزارش‌گیری به تفصیل بیان شد. در ادامه، بخش بک‌اند که با **Fastify** توسعه یافته بود، مورد بررسی قرار گرفت و جزئیات پیاده‌سازی API ها، منطق سرور، سازوکارهای امنیتی و نحوه تعامل با پایگاه داده از طریق **Prisma** شرح داده شد. همچنین، استراتژی تست و اعتبارسنجی سامانه با استفاده از ابزارهای تست واحد مدرن بیان گردید و در نهایت، به مهم‌ترین چالش‌های فنی و راهکارهای عملی برای حل آن‌ها پرداخته شد. با اتمام این فصل، محصول نرم‌افزاری اولیه آماده ارزیابی نهایی است که در فصل آینده به آن پرداخته خواهد شد.

## فصل پنجم

### ارزیابی و نتایج

#### ۱-۵- مقدمه

پس از طی کردن مراحل تحلیل، طراحی (فصل سوم) و پیاده‌سازی (فصل چهارم)، پروژه به نقطه عطفی می‌رسد که در آن، محصول نرم‌افزاری تولید شده باید در ترازوی سنجش قرار گیرد. فصل حاضر به طور کامل به فرآیند ارزیابی سامانه اختصاص دارد؛ مرحله‌ای حیاتی که در آن، میزان موفقیت پروژه در دستیابی به اهداف اولیه و برآورده‌سازی نیازمندی‌های تعریف‌شده، به صورت عینی و روشمند اندازه‌گیری می‌شود. این ارزیابی صرفاً به معنای یافتن خطاها نیست، بلکه یک فرآیند جامع برای صحت‌گذاری<sup>۱</sup> بر روی کل سیستم است تا مشخص گردد آیا محصول نهایی نه تنها از نظر فنی صحیح عمل می‌کند، بلکه برای کاربر نهایی نیز مفید، قابل استفاده و رضایت‌بخش است.

در همین راستا، این فصل یک نقشه راه مشخص را دنبال می‌کند. ابتدا، متدولوژی و روش‌های به کار رفته برای ارزیابی سامانه به تفصیل تشریح خواهند شد. این متدولوژی بر یک رویکرد دوگانه استوار است: ارزیابی فنی برای سنجش معیارهای عملکردی و کمی، و ارزیابی کاربرپذیری برای تحلیل کیفیت تعامل انسان و کامپیوتر. در ادامه، نتایج کمی و کیفی به دست آمده از این آزمون‌ها به صورت دقیق ارائه و تحلیل می‌گردند. سپس، با نگاهی به آینده، طرح تجاری و امکان‌سنجی<sup>۲</sup> پروژه مورد بحث قرار گرفته و در نهایت، فصل با یک جمع‌بندی از تمام یافته‌ها و دستاوردهای این مرحله به پایان می‌رسد.

---

<sup>۱</sup> Validation

<sup>۲</sup> Feasibility Study

## ۲-۵- روش ارزیابی سامانه

به منظور دستیابی به یک ارزیابی جامع، قابل اعتماد و چندبعدی، یک استراتژی آزمون دومنظوره اتخاذ گردید. این استراتژی به ما اجازه می‌دهد تا هم سلامت فنی و درونی سیستم را بسنجیم و هم کیفیت تجربه بیرونی آن را از نگاه یک کاربر واقعی ارزیابی کنیم. این دو رویکرد مکمل یکدیگر بوده و تصویری ۳۶۰ درجه از وضعیت نهایی پروژه ارائه می‌دهند.

### ۱-۲-۵- ارزیابی فنی و عملکردی

هدف اصلی این بخش، ارزیابی ویژگی‌های غیرعملکردی سامانه، به ویژه کارایی (Performance) و پایداری آن تحت شرایط استاندارد است. این آزمون‌ها به صورت کنترل‌شده و با استفاده از ابزارهای تخصصی انجام شده‌اند تا داده‌های کمی و قابل استنادی به دست آید.

#### معیارهای ارزیابی:

عملکرد فرانت‌اند: برای سنجش عملکرد سمت کاربر، از معیارهای استاندارد Google Core Web Vitals و گزارش جامع Lighthouse استفاده شد. تمرکز اصلی بر روی موارد زیر بود:

Largest Contentful Paint (LCP): مدت زمان لازم برای بارگذاری و نمایش بزرگترین عنصر محتوایی در صفحه، که نشان‌دهنده سرعت بارگذاری درک‌شده توسط کاربر است.

امتیاز عملکرد (Performance Score): امتیاز کلی Lighthouse که سرعت بارگذاری و تعامل‌پذیری صفحه را می‌سنجد.

امتیاز دسترسی‌پذیری (Accessibility Score): میزان انطباق سایت با استانداردهای دسترسی‌پذیری برای کاربران دارای معلولیت.

عملکرد بک‌اند: برای ارزیابی کارایی سرور، زمان پاسخ‌دهی (Latency) دو Endpoint حیاتی که نماینده عملیات خواندن و نوشتن هستند، اندازه‌گیری شد:

post /api/tests: به عنوان نماینده یک عملیات نوشتن (Write-intensive).

get /api/tests/history: به عنوان نماینده یک عملیات خواندن (Read-intensive).

ابزارهای مورد استفاده:

Google Lighthouse: این ابزار متن‌باز که مستقیماً در ابزارهای توسعه‌دهنده مرورگر کروم (Chrome DevTools) تعبیه شده است، برای اجرای یک تحلیل کامل بر روی صفحات اصلی و تست سامانه و استخراج امتیازات فوق به کار گرفته شد.

Postman: از این پلتفرم محبوب برای تست API استفاده شد. با استفاده از قابلیت Collection Runner در Postman، برای هر یک از Endpoints هدف، ۱۰۰ درخواست به صورت متوالی ارسال گردید و میانگین زمان پاسخ‌دهی به میلی‌ثانیه ثبت شد.

## ۲-۲-۵- ارزیابی کاربردپذیری

عملکرد فنی بالا زمانی ارزشمند است که کاربران بتوانند به سادگی و به طور مؤثر از سامانه استفاده کنند. هدف از این بخش، ارزیابی رابط کاربری و تجربه کاربری (UI/UX) از طریق مشاهده تعامل کاربران واقعی با محصول بود.

### جامعه آماری:

برای اطمینان از اینکه بازخوردها طیف وسیعی از کاربران بالقوه را پوشش می‌دهند، یک گروه ۱۰ نفره با مشخصات ناهمگون به صورت هدفمند انتخاب و در این فرآیند مشارکت کردند. ترکیب این گروه به شرح زیر بود:

۵ دانشجو: ۳ نفر از این گروه در سطح کاملاً مبتدی و ۲ نفر در سطح متوسط تایپ قرار داشتند تا منحنی یادگیری و سادگی کار با بخش آموزش سنجیده شود.

۳ کارمند حرفه‌ای: این گروه شامل یک برنامه‌نویس، یک کارمند اداری و یک تولیدکننده محتوا بود تا کاربرد سامانه در سناریوهای شغلی واقعی ارزیابی گردد.

۲ کاربر عمومی: این دو نفر نماینده کاربران عادی بودند که بدون هدف شغلی خاص و صرفاً برای بهبود مهارت شخصی از سامانه استفاده می‌کنند.

شرح وظایف (Tasks) محول شده به کاربران:

از هر شرکت‌کننده خواسته شد تا یک سناریوی از پیش تعریف شده را که شامل وظایف اصلی سامانه بود، به ترتیب انجام دهد. هدف هر وظیفه در پرانتز مشخص شده است:

وظیفه ۱ (آنبوردینگ): یک حساب کاربری جدید در سامانه ایجاد کرده و با موفقیت وارد شوید.

وظیفه ۲ (یادگیری): دو درس ابتدایی از بخش "آموزش تایپ" را پیدا کرده و تکمیل کنید.

وظیفه ۳ (آزمون): یک آزمون تست سرعت در سطح "متوسط" و با طول پاراگراف "متوسط" را پیدا کرده و به پایان برسانید.

وظیفه ۴ (ناوبری و گزارش‌گیری): پس از اتمام آزمون، به صفحه پروفایل خود رفته و نمودار پیشرفت و نتیجه آخرین آزمون خود را پیدا کنید.

وظیفه ۵ (شخصی‌سازی): تم رنگی وبسایت را به رنگ دلخواه خود تغییر دهید.

روش جمع‌آوری بازخورد:

یک رویکرد ترکیبی برای جمع‌آوری داده‌های کمی و کیفی به کار گرفته شد:

پروتکل تفکر با صدای بلند<sup>۱</sup>: از شرکت‌کنندگان خواسته شد تا در حین انجام وظایف، افکار، سوالات و نقاط سردرگمی خود را با صدای بلند بیان کنند. این تعاملات توسط آزمون‌گر ثبت می‌شد تا درک عمیقی از

---

<sup>1</sup> Think Aloud Protocol

فرآیندهای ذهنی کاربر به دست آید.

پرسشنامه پس از تست<sup>۱</sup>: پس از اتمام تمام وظایف، یک پرسشنامه استاندارد در اختیار هر کاربر قرار گرفت. این پرسشنامه شامل دو بخش بود:

بخش کمی: شامل سوالاتی بر اساس مقیاس ۵ درجه‌ای لیکرت<sup>۲</sup> برای امتیازدهی به میزان سادگی هر یک از وظایف (از ۱=بسیار دشوار تا ۵=بسیار آسان).

بخش کیفی: شامل سوالات باز مانند "کدام بخش از کار با سامانه برای شما گیج‌کننده‌تر بود؟" و "چه قابلیت‌هایی را دوست دارید به سامانه اضافه شود؟" تا نظرات و پیشنهادهای خلاقانه کاربران جمع‌آوری گردد.

### ۵-۳- نتایج ارزیابی

این بخش به ارائه و تحلیل داده‌های کمی و کیفی به دست آمده از روش‌های ارزیابی که در بخش قبل تشریح شد، می‌پردازد. نتایج به تفکیک ارزیابی فنی و ارزیابی کاربردپذیری ارائه می‌گردند تا تصویری کامل از عملکرد و کیفیت سامانه حاصل شود.

### ۵-۳-۱- نتایج عملکرد فنی

آزمون‌های فنی با هدف سنجش کارایی و بهینگی سامانه در دو بخش فرانت‌اند و بک‌اند انجام شد.

- نتایج عملکرد فرانت‌اند (Lighthouse): ارزیابی صفحه اصلی و صفحه تست با استفاده از ابزار Lighthouse نتایج بسیار مطلوبی را نشان داد که حاکی از کیفیت بالای توسعه و رعایت استانداردهای مدرن وب است. نتایج کامل این ارزیابی در جدول ۵-۱ خلاصه شده است.

این امتیازات، به ویژه امتیاز بالای **Accessibility** و **Best Practices**، نشان‌دهنده یک معماری فرانت‌اند سالم و استاندارد است. امتیاز عملکرد ۹۴ نیز زمان بارگذاری سریع و تجربه کاربری بدون وقفه را برای کاربران تضمین می‌کند.

<sup>1</sup> Post-Test Questionnaire

<sup>2</sup> Likert Scale

- نتایج عملکرد بک‌اند (Postman): تست بار<sup>۱</sup> بر روی API های اصلی نیز نتایج رضایت‌بخشی را به همراه داشت. میانگین زمان پاسخ‌دهی برای ۱۰۰ درخواست متوالی به شرح زیر ثبت گردید:
  - `post/api/tests`: میانگین زمان پاسخ‌دهی ۶۸ میلی‌ثانیه.
  - `get/api/tests/history`: میانگین زمان پاسخ‌دهی ۴۲ میلی‌ثانیه.

این نتایج نشان می‌دهد که بک‌اند نوشته‌شده با **Fastify** و **Prisma** عملکردی بسیار سریع و کارآمد دارد. زمان پاسخ‌دهی زیر ۱۰۰ میلی‌ثانیه برای هر دو عملیات خواندن و نوشتن، تجربه‌ای کاملاً آبی و بدون تأخیر را برای کاربر نهایی فراهم می‌آورد.

معیار	امتیاز از ۱۰۰	تفسیر
Performance	۹۴	سرعت بارگذاری و تعامل‌پذیری بسیار بالا، تجربه کاربری روان.
Accessibility	۹۸	انطباق عالی با استانداردهای دسترسی‌پذیری برای افراد دارای معلولیت.
Best Practices	۱۰۰	رعایت کامل اصول مهندسی وب و امنیت در فرانت‌اند.
SEO	۹۵	بینه‌سازی مناسب برای موتورهای جستجو به لطف ساختار <code>nuxt3</code> .

جدول ۵-۱: نتایج ارزیابی عملکرد فرانت‌اند با ابزار **Lighthouse**

### ۵-۳-۲- تحلیل نتایج کاربردپذیری و بازخوردهای کاربران

نتایج به دست آمده از تست با ۱۰ کاربر واقعی، دیدگاه‌های ارزشمندی در مورد سادگی و کارایی رابط کاربری ارائه داد.

- تحلیل کمی:
  - نرخ موفقیت انجام وظایف<sup>۲</sup>: از مجموع ۱۰ شرکت‌کننده، ۹ نفر (۹۰٪) توانستند تمام ۶ وظیفه محول‌شده را با موفقیت و بدون نیاز به راهنمایی به پایان برسانند. یک کاربر در پیدا کردن گزینه "تغییر تم" دچار مشکل شد که نشان می‌دهد این قابلیت ممکن است نیاز به جایابی بهتری در رابط کاربری داشته باشد.

<sup>1</sup> Load Testing

<sup>2</sup> Task Success Rate

- امتیاز سادگی وظایف: میانگین امتیاز ثبت شده توسط کاربران برای میزان سادگی انجام وظایف، ۴.۷ از ۵ بود که نشان دهنده رضایت بالا و طراحی شهودی سامانه است. وظیفه "انجام تست سرعت" بالاترین امتیاز (۴.۹) و وظیفه "تغییر تم سایت" کمترین امتیاز (۴.۲) را کسب کردند. این نتایج در نمودار میله‌ای (شکل ۵-۲) به صورت بصری مقایسه شده‌اند.

## میانگین امتیاز سادگی وظایف



شکل ۵-۲: نمودار میانگین امتیاز سادگی وظایف از دید کاربران

- تحلیل کیفی: بازخوردهای کیفی جمع‌آوری شده از پرسشنامه‌ها و پروتکل "تفکر با صدای بلند" به دو دسته اصلی تقسیم می‌شوند:
  - بازخوردهای مثبت:
    - رابط کاربری تمیز و متمرکز: تقریباً تمام کاربران از سادگی و طراحی بدون حواس‌پرتی صفحه تست و درس تمجید کردند.
    - بازخورد آنی: هایلایت شدن آنی کاراکترهای صحیح و غلط به عنوان یکی از "انگیزه‌بخش‌ترین" ویژگی‌ها توسط چندین کاربر ذکر شد.
    - نمودار پیشرفت: کاربران بخش نمودار پیشرفت در پروفایل را برای مشاهده رشد مهارت خود "بسیار مفید و کاربردی" توصیف کردند.
  - انتقادات و پیشنهادها برای بهبود:
    - تنوع متون: پر تکرارترین پیشنهاد، افزودن متون تمرینی بیشتر و متنوع‌تر، به خصوص متون تخصصی (مثلاً برای برنامه‌نویسان) بود.
    - افزودن بازخورد صوتی: دو کاربر پیشنهاد دادند که گزینه‌ای برای فعال‌سازی صدای کلیک یا صدای خطا هنگام تایپ اضافه شود.



- قابلیت کشف بهتر شخصی سازی: همانطور که در نتایج کمی نیز مشهود بود، چند کاربر اشاره کردند که گزینه های شخصی سازی مانند تغییر تم می توانند در جایگاه برجسته تری قرار گیرند.

### ۵-۳-۳- نمایش محصول نهایی

برای ارائه یک تصویر عینی از سامانه پیاده سازی شده، در ادامه مجموعه ای از تصاویر کلیدی (شکل های ۳-۵ تا ۳-۱۵) از محیط نهایی برنامه آورده شده است. این تصاویر، طراحی رابط کاربری و قابلیت های اصلی سامانه را به نمایش می گذارند.

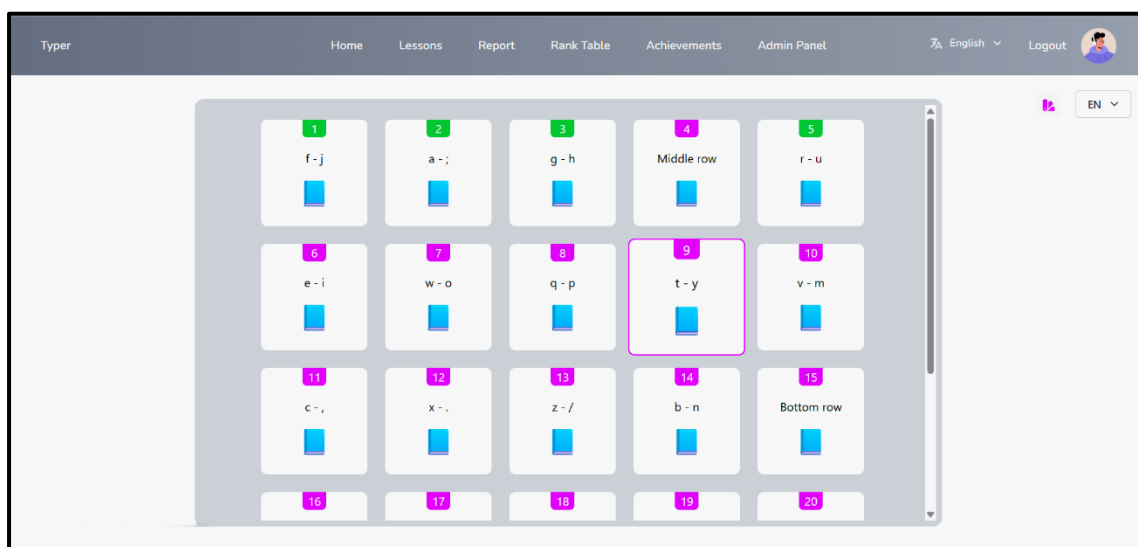
#### ۱. فرآیند ورود و ثبت نام کاربر

صفحه ثبت نام سامانه را با طراحی مدرن و فیلدهای ورودی مشخص نمایش می دهد. تمرکز در طراحی این صفحه بر سادگی و ایجاد یک تجربه روان برای آنبردینگ (Onboarding) کاربران جدید بوده است. شکل ۳-۵ نمایشی برای نمایش آن است.

شکل ۳-۵: نمایشی از صفحه ی ثبت نام کاربران سامانه

#### ۲. بخش آموزش و یادگیری

پس از ورود، کاربر سفر یادگیری خود را آغاز می کند.



شکل ۴-۵: نمایی از درس های آموزشی سامانه

شکل ۴-۵، صفحه اصلی بخش آموزش را نشان می‌دهد که در آن، تمام درس‌ها به صورت کارت‌های مجزا و شماره‌گذاری شده نمایش داده شده‌اند. این طراحی ماژولار به کاربر اجازه می‌دهد تا به راحتی مسیر یادگیری خود را دنبال کرده و به درس مورد نظر خود دسترسی پیدا کند. شکل ۴-۵، رابط کاربری صفحه انتخاب درس است.

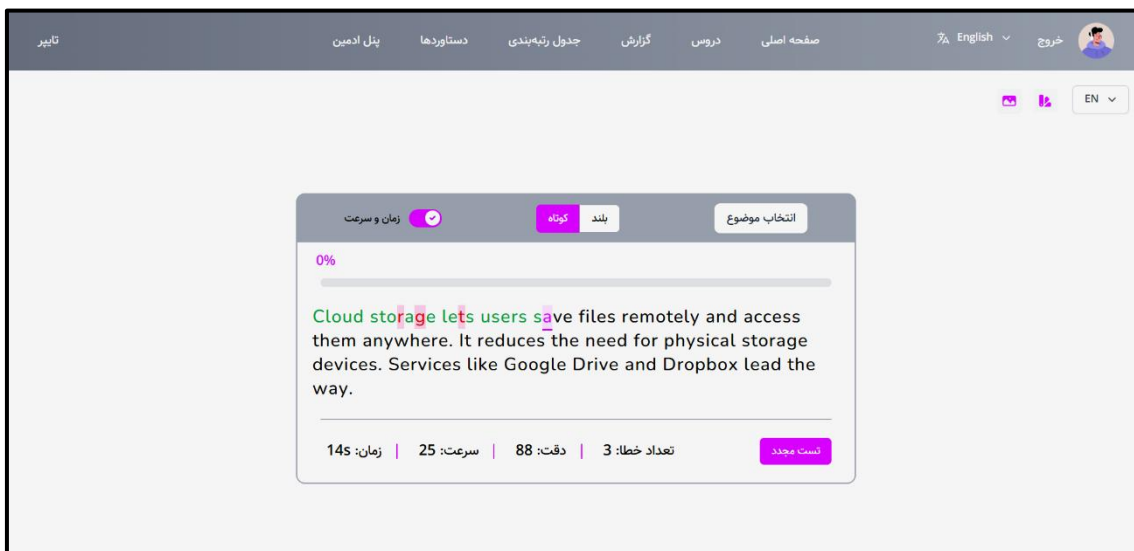


شکل ۵-۵: نمایی از یک درس آموزشی با کیبورد راهنمای تصویری انگشتان

شکل ۵-۵، محیط تعاملی یک درس را به نمایش می‌گذارد. ویژگی‌های کلیدی مانند هایلایت آنی کاراکتر فعلی (صورتی)، نمایش کاراکترهای صحیح (سبز)، نوار پیشرفت و کیبورد راهنمای تصویری که انگشت صحیح را نشان می‌دهد، در این نما به وضوح قابل مشاهده است.

### ۳. بخش تست سرعت و ارزیابی

پس از یادگیری، کاربر مهارت خود را می‌سنجد. شکل ۵-۶، مازول اصلی تست سرعت را در حین کار نشان می‌دهد. قابلیت‌های شخصی‌سازی مانند انتخاب طول پاراگراف و موضوع، وجود دارد. در حین تایپ، آمار زنده شامل زمان، سرعت و دقت به کاربر نمایش داده می‌شود تا از عملکرد لحظه‌ای خود آگاه باشد.



شکل ۵-۶: نمای صفحه انجام تست سرعت با هایلایت آنی کاراکترها و نمایشگرهای زنده

### ۴. بخش انگیزش و پیگیری پیشرفت

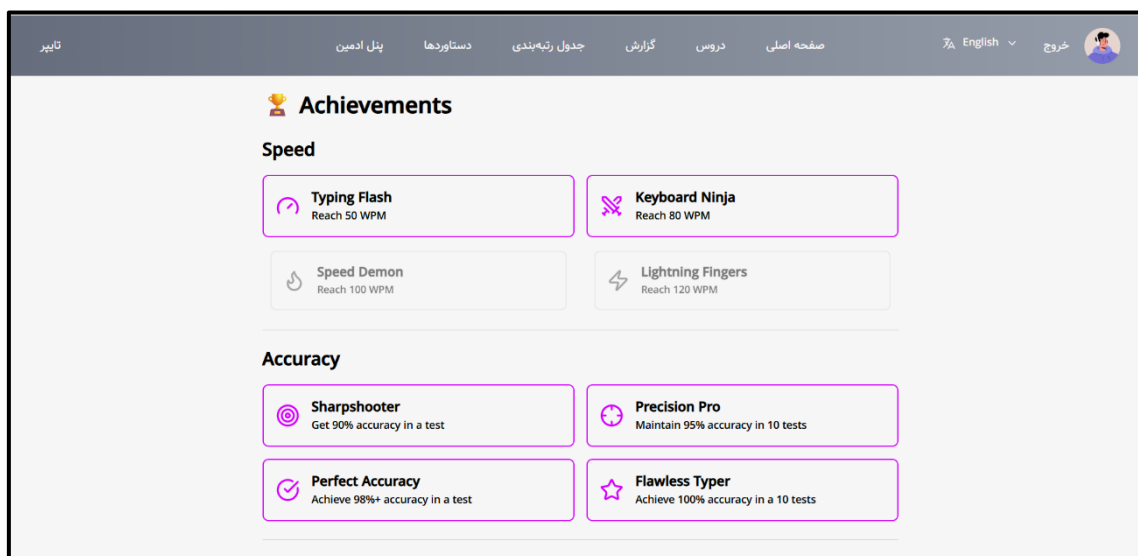
در این بخش، نتایج و دستاوردهای کاربر به نمایش گذاشته می‌شود. در شکل ۵-۷، بخش جدول رتبه‌بندی را نشان می‌دهد که با نمایش برترین رکوردهای ثبت‌شده، یک محیط رقابتی سالم و انگیزشی برای کاربران ایجاد می‌کند.

The screenshot shows a 'Top 20 Typists' leaderboard. The table has four columns: Rank, Profile, Username, and Speed Record. The top three typists are ali123 (76 WPM), sepehrshz (65 WPM), and matin (32 WPM).

Rank	Profile	Username	Speed Record
1		ali123	76 WPM
2		sepehrshz	65 WPM
3		matin	32 WPM

شکل ۵-۷: نمایی از جدول رتبه‌بندی بر اساس سرعت

در ادامه شکل ۵-۸، سیستم بازی‌وارسازی (Gamification) سامانه را به نمایش می‌گذارد. دستاوردها در دسته‌بندی‌های مختلف (مانند سرعت و دقت) تعریف شده‌اند و کسب هر یک از آن‌ها، به کاربر حس پیشرفت و موفقیت می‌دهد.

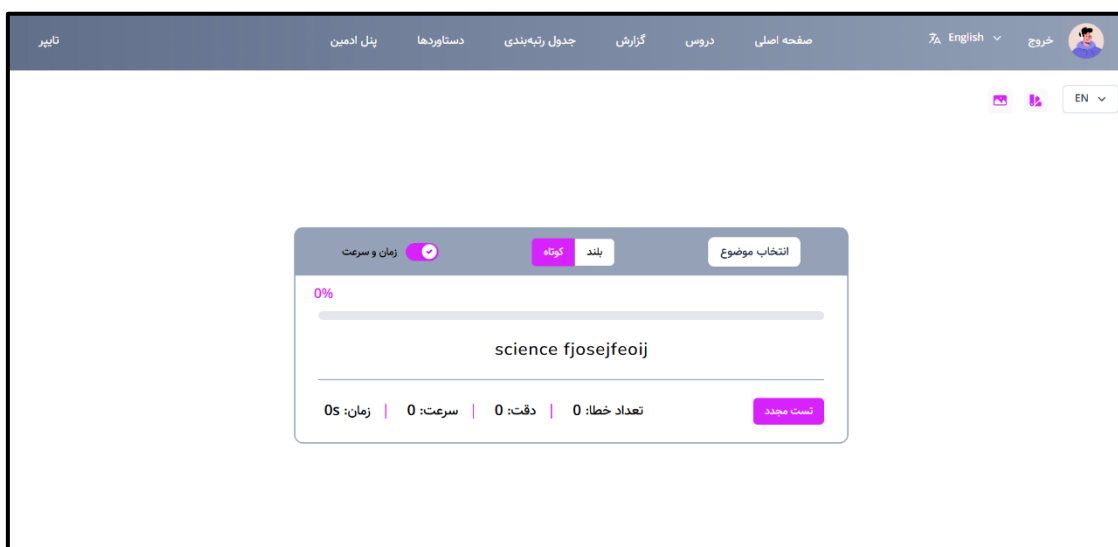


شکل ۵-۸: دستاورد های کاربران در سایت

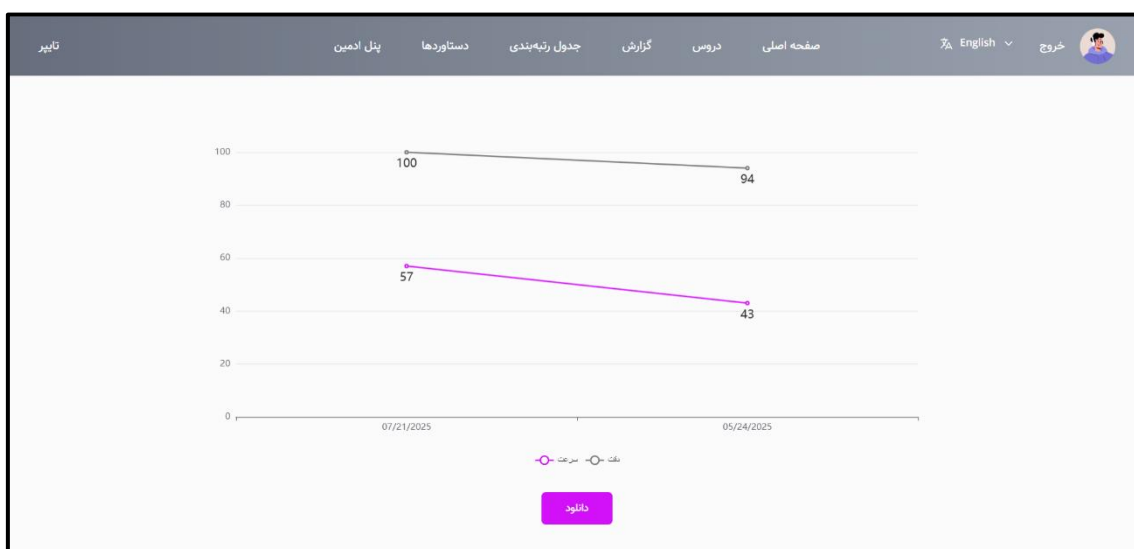
بخش کلیدی پیگیری پیشرفت در پروفایل کاربری را به نمایش می‌گذارد که یکی از مهم‌ترین ابزارهای تحلیلی سامانه برای کاربر است. این صفحه به دو بخش اصلی تقسیم شده است:

- **نمودار خطی روند پیشرفت:** در بالای صفحه، یک نمودار بصری و تعاملی قرار دارد که روند تغییرات دو معیار اصلی سرعت (WPM) و دقت را در طول زمان به تصویر می‌کشد. این نمودار به کاربر اجازه می‌دهد تا به سرعت، رشد و پیشرفت مهارت خود را مشاهده کرده و تأثیر تمرین‌های مستمر را به صورت عینی درک کند.
- **جدول تاریخچه نتایج:** در پایین نمودار، یک جدول کامل از تاریخچه تمام تست‌های انجام‌شده توسط کاربر قرار دارد. این جدول شامل جزئیات دقیقی از هر تست، مانند تاریخ، سرعت، دقت و تعداد خطاها می‌باشد و به کاربر این امکان را می‌دهد که عملکرد خود را در هر آزمون به صورت مجزا بازبینی و تحلیل کند.

شکل ۵-۹ و شکل ۵-۱۰، با تبدیل داده‌های عملکردی به اطلاعات بصری و قابل فهم، نقش مهمی در ایجاد انگیزه و ارائه یک مسیر یادگیری هدفمند برای کاربر ایفا می‌کند. شکل ۵-۱۱، این صفحه را به نمایش می‌گذارد.

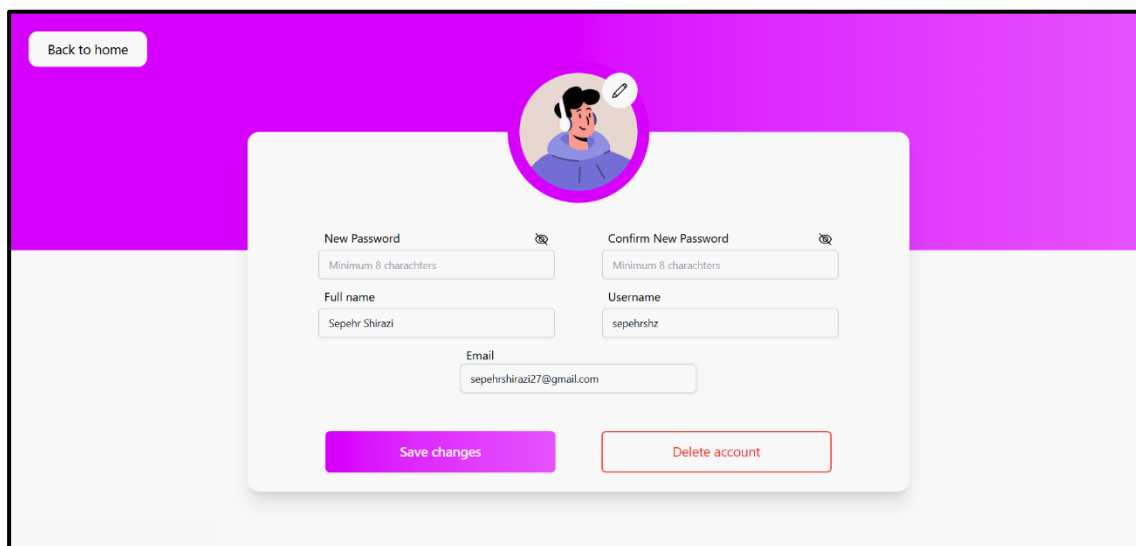


شکل ۵-۹: نتیجه ی تست کاربر

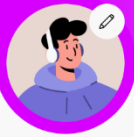



شکل ۵-۱۰: نمای صفحه پروفایل کاربر با نمایش جدول تاریخچه نتایج و نمودار خطی روند پیشرفت.


شکل ۵-۱۱، صفحه پروفایل کاربر را نشان می‌دهد که در آن، کاربر می‌تواند اطلاعات شخصی خود را ویرایش کرده، رمز عبور خود را تغییر دهد و به تنظیمات حساب خود دسترسی داشته باشد. (در صورت وجود نمودار پیشرفت، بهتر است تصویری از آن صفحه قرار داده شود).



Back to home



New Password  Minimum 8 characters

Confirm New Password  Minimum 8 characters

Full name Sepehr Shirazi

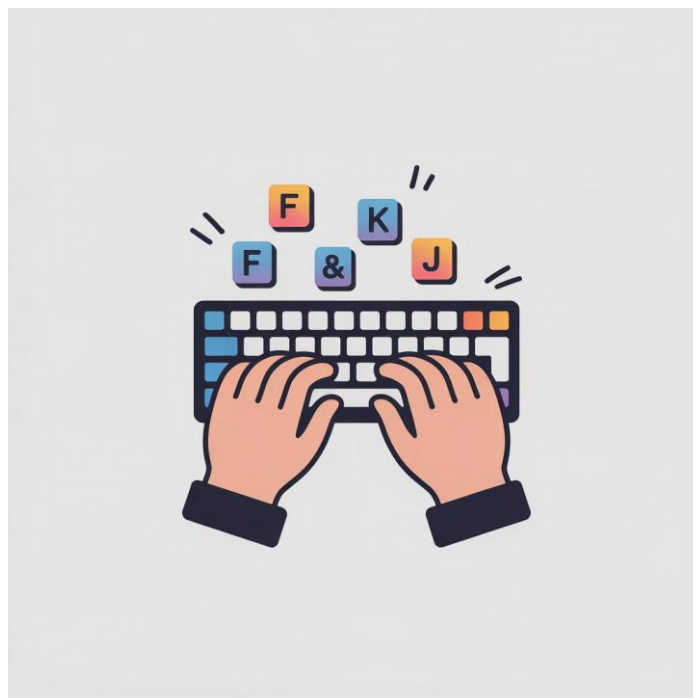
Username sepehrshz

Email sepehrshirazi27@gmail.com

Save changes Delete account

شکل ۵-۱۱: نمای صفحه پروفایل کاربر

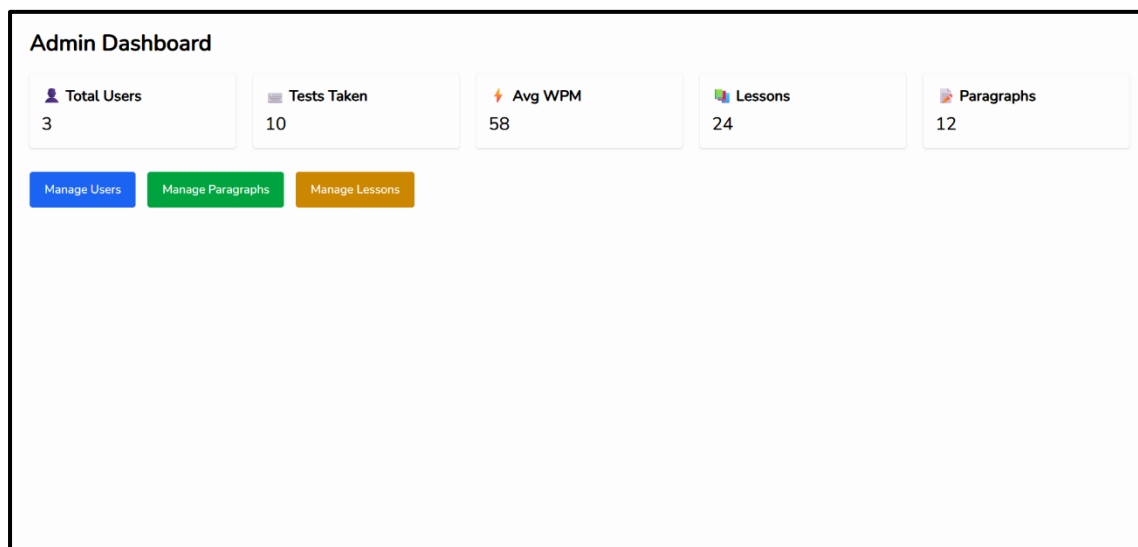
لوگوی رسمی سامانه را شکل ۵-۱۲، نمایش می‌دهد که هویت بصری و برند پروژه را تعریف می‌کند. طراحی این لوگو با الهام از مفاهیم سرعت، پیشرفت و یادگیری صورت گرفته است.



شکل ۵-۱۲: تصویر لوگوی سامانه

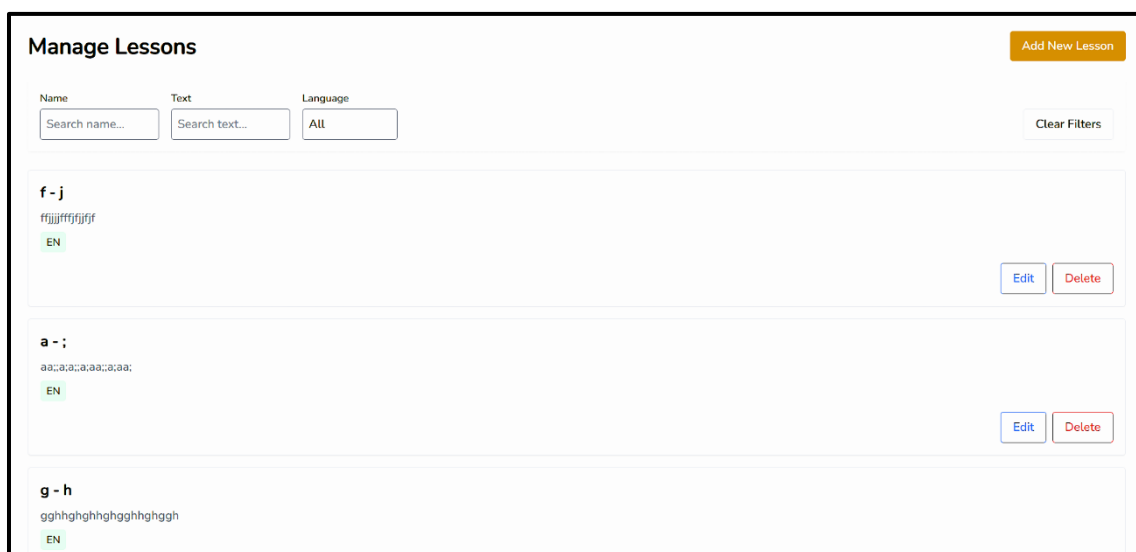
## ۶. بخش مدیریت سامانه (پنل ادمین)

این بخش، قابلیت‌های مدیریتی را که برای راهبری و نگهداری سامانه طراحی شده‌اند، به نمایش می‌گذارد. شکل ۵-۱۳، نمای اصلی پنل مدیریت را نشان می‌دهد که دسترسی به بخش‌های مختلف مدیریتی مانند مدیریت کاربران، درس‌ها و پاراگراف‌ها را فراهم می‌کند.



شکل ۵-۱۳: نمای صفحه ی داشبورد ادمین

شکل ۵-۱۴، قابلیت‌های کلیدی پنل ادمین مانند مشاهده لیست درس‌ها، افزودن درس جدید، ویرایش و حذف محتوای آموزشی را به صورت عملی نمایش می‌دهد. این بخش، پایداری و سهولت در نگهداری بلندمدت سامانه را تضمین می‌کند.



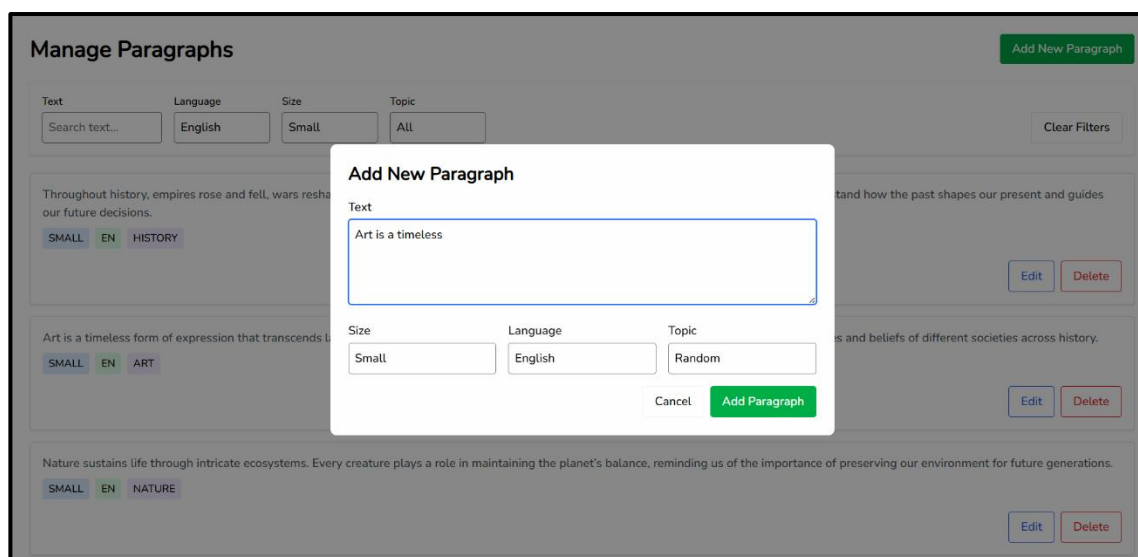
شکل ۵-۱۴: نمای صفحه مدیریت محتوای آموزشی

یکی از قابلیت‌های اصلی پنل مدیریت سامانه را به نمایش می‌گذارد که برای راهبری و نگهداری بلندمدت پروژه طراحی شده است. در این نما، مدیر سیستم به لیستی از تمام درس‌ها و پاراگراف‌های آموزشی دسترسی دارد و می‌تواند به راحتی عملیات مدیریتی **CRUD** را بر روی محتوا انجام دهد.

این صفحه به مدیر اجازه می‌دهد تا:

- درس جدیدی با محتوای متنی دلخواه به سامانه اضافه کند.
- محتوای یک درس موجود را ویرایش و به‌روزرسانی نماید.
- دروس غیرضروری را از سامانه حذف کند.

وجود این پنل مدیریتی، تضمین می‌کند که محتوای آموزشی سایت همواره تازه و به‌روز باقی بماند و سامانه بتواند بدون نیاز به دخالت مستقیم تیم توسعه، به رشد و تکامل خود ادامه دهد. در ادامه شکل ۵-۱۴، نمایی از آن را نمایش می‌دهد.



شکل ۵-۱۵: نمایی از ادیت یک درس از قابلیت‌های ادمین

#### ۴-۵- طرح تجاری<sup>۱</sup> و امکان‌سنجی

هر پروژه نرم‌افزاری موفق، علاوه بر برتری فنی، باید دارای یک چشم‌انداز برای کاربرد در دنیای واقعی و پایداری در بلندمدت باشد. این بخش به بررسی یک طرح تجاری بالقوه و امکان‌سنجی اولیه پروژه می‌پردازد تا مسیری از یک طرح دانشگاهی به یک محصول قابل عرضه در بازار را ترسیم کند.

<sup>۱</sup> Business Plan



#### ۱-۴-۵- مدل کسب‌وکار<sup>۱</sup>

مدل کسب‌وکار پیشنهادی برای این سامانه، یک رویکرد دومارحله‌ای است که با هدف جذب حداکثری کاربر و سپس درآمدزایی پایدار طراحی شده است.

##### فاز اول: ارائه رایگان برای جذب کاربر و جمع‌آوری بازخورد

در مرحله ابتدایی، این سایت به صورت رایگان در اختیار کاربران قرار می‌گیرد. هدف اصلی در این فاز، ساخت یک جامعه کاربری اولیه، اعتبارسنجی قابلیت‌های اصلی سامانه در مقیاس واقعی، و از همه مهم‌تر، جمع‌آوری و اصلاح بازخوردها و ایرادات احتمالی است. این رویکرد به بهبود مستمر محصول بر اساس نیازهای واقعی کاربران کمک کرده و اعتماد آن‌ها را جلب می‌نماید.

##### فاز دوم: مدل اشتراکی<sup>۲</sup> با ارائه امکانات پیشرفته

پس از رسیدن به پایداری و جذب تعداد قابل توجهی از کاربران، سامانه وارد فاز درآمدزایی می‌شود. در این مرحله، برخی امکانات سایت را به صورت رایگان باقی می‌مانند تا همچنان برای کاربران جدید جذاب باشد. اما بخش دیگری از امکانات تنها در دسترس کاربرانی که اشتراک ماهیانه سایت را تهیه کرده‌اند قرار می‌گیرد. این امکانات پیشرفته می‌توانند شامل موارد زیر باشند:

- تحلیل‌های آماری پیشرفته: ارائه نمودارهای تحلیلی عمیق‌تر در مورد خطاهای متداول کاربر، عملکرد بر روی هر کلید و مقایسه عملکرد در بازه‌های زمانی مختلف.
- درس‌ها و متون تمرینی تخصصی: افزودن بسته‌های آموزشی اختصاصی برای حوزه‌های خاص مانند برنامه‌نویسی (تایپ کد)، حقوقی (تایپ متون قانونی) یا پزشکی.
- تجربه کاربری بدون تبلیغات: حذف کامل تبلیغات برای کاربران ویژه.
- شخصی‌سازی بیشتر: ارائه تم‌ها و قابلیت‌های شخصی‌سازی انحصاری.

#### ۲-۴-۵- تحلیل بازار هدف

بازار هدف این سامانه شامل گروه‌های متنوعی است که مهارت تایپ سریع برای آن‌ها یک نیاز یا مزیت مهم محسوب می‌شود:

- دانشجویان و دانش‌آموزان: به عنوان بزرگترین بخش بازار، این گروه برای انجام تکالیف، نوشتن

<sup>۱</sup> Business Model

<sup>۲</sup> Freemium Model

مقالات و پایان‌نامه‌ها به این مهارت نیاز مبرم دارند.

- متخصصان حوزه فناوری و برنامه‌نویسان: برای این گروه، سرعت و دقت در تایپ کد به طور مستقیم بر بهره‌وری کاری تأثیرگذار است.
- تولیدکنندگان محتوا، نویسندگان و مترجمان: افرادی که شغل آن‌ها مستقیماً با نوشتن سروکار دارد و حجم بالایی از متن را روزانه تولید می‌کنند.
- کارمندان اداری و دفتری: برای انجام مکاتبات، گزارش‌نویسی و ورود داده‌ها.

### ۳-۴-۵- امکان‌سنجی اولیه

**امکان‌سنجی فنی:** همانطور که در فصل چهارم به تفصیل نشان داده شد، پیاده‌سازی موفق یک نمونه اولیه کاملاً کاربردی، امکان‌پذیری فنی پروژه را به اثبات رسانده است. پشته فناوری منتخب مدرن، مقیاس‌پذیر و قابل نگهداری بوده و بستر مناسبی برای توسعه‌های آتی را فراهم می‌آورد.

**امکان‌سنجی اقتصادی:** این پروژه دارای هزینه‌های اولیه پایینی است که عمدتاً شامل هزینه ثبت دامنه و میزبانی وب (هاستینگ) سرور می‌شود. با توجه به مدل کسب‌وکار Freemium، در صورت موفقیت در جذب یک جامعه کاربری بزرگ در فاز رایگان، تبدیل درصد کمی از این کاربران به مشترکین ویژه می‌تواند به راحتی هزینه‌های جاری را پوشش داده و پروژه را به سوددهی اقتصادی برساند. بنابراین، این طرح از نظر اقتصادی نیز دارای توجیه اولیه است.

### ۵-۵- جمع‌بندی

در این فصل، سامانه پیاده‌سازی شده از طریق یک فرآیند ارزیابی جامع و دومنظوره مورد سنجش قرار گرفت. نتایج حاصل از ارزیابی فنی، عملکرد بسیار مطلوب و بهینه سامانه را در دو بخش فرانت‌اند و بک‌اند به اثبات رساند. ارزیابی کاربردپذیری نیز با مشارکت کاربران واقعی نشان داد که سامانه از نظر سادگی و کارایی در سطح بالایی قرار دارد و توانسته است رضایت کاربران تست‌کننده را جلب نماید؛ در عین حال، بازخوردهای کیفی ارزشمندی برای بهبودهای آتی نیز از این فرآیند استخراج گردید. علاوه بر این، با ارائه یک طرح تجاری مبتنی بر مدل Freemium و تحلیل امکان‌سنجی اولیه، یک مسیر بالقوه برای رشد و پایداری پروژه در دنیای واقعی ترسیم شد. در مجموع، یافته‌های این فصل تأیید می‌کند که پروژه حاضر با موفقیت به اهداف اولیه خود دست یافته و یک محصول کارآمد، کاربرپسند و قابل اتکا حاصل شده است. این نتایج، زمینه را برای نتیجه‌گیری نهایی و ارائه پیشنهادات برای کارهای آتی در فصل بعد فراهم می‌کند.

## فصل ششم

### نتیجه‌گیری و پیشنهادات

این فصل به عنوان نقطه پایانی این گزارش، وظیفه جمع‌بندی، تأمل و نگاه به آینده را بر عهده دارد. اکنون زمان آن است که دستاوردهای پروژه را در یک تصویر کلی بازبینی کنیم، با نگاهی انتقادی به محدودیت‌های آن بنگریم و در نهایت، مسیرهای هیجان‌انگیزی را که برای توسعه و تکامل این سامانه در آینده وجود دارد، ترسیم نماییم. این فصل نه تنها خاتمه‌بخش این پژوهش، بلکه سرآغازی برای ایده‌های آتی است.

#### ۱-۶- نتیجه‌گیری کلی

این پروژه با هدف اصلی طراحی و پیاده‌سازی یک سامانه آنلاین و مدرن برای آموزش تایپ ده‌انگشتی آغاز گردید. همانطور که در فصل اول تشریح شد، انگیزه اصلی این پژوهش، شناسایی یک شکاف مشخص در بازار ابزارهای موجود بود؛ جایی که پلتفرم‌های فعلی یا بر سنجش صرف تمرکز داشتند و فاقد مسیر آموزشی بودند، یا در ارائه «بازخورد تحلیلی» و «تجربه کاربری شخصی‌سازی‌شده» دچار ضعف بودند. پروژه حاضر به صورت مشخص برای پر کردن همین شکاف‌ها و ارائه یک راهکار یکپارچه تعریف شد که سه رکن اصلی آموزش، تمرین و تحلیل را در یک محیط جذاب و کاربرمحور ادغام کند.

برای دستیابی به این هدف، در فصل سوم یک معماری مدرن کلاینت-سرور به عنوان شالوده فنی پروژه طراحی شد. این معماری با تفکیک کامل مسئولیت‌ها، به ما اجازه داد تا بخش فرانت‌اند را با استفاده از اکوسیستم و فریم‌ورک قدرتمند توسعه دهیم. این انتخاب، امکان پیاده‌سازی یک رابط کاربری واکنش‌گرا، سریع و با قابلیت رندر سمت سرور را فراهم آورد. در سوی دیگر، بخش بک‌اند با استفاده از پلتفرم Node.js و فریم‌ورک سبک و پرسرعت Fastify ساخته شد تا بتواند به درخواست‌ها با حداقل تأخیر پاسخ دهد. ارتباط با پایگاه داده PostgreSQL نیز از طریق یک لایه انتزاعی مدرن به نام Prisma ORM مدیریت شد تا ضمن تضمین امنیت،

فرآیند توسعه تسریع گردد.

فصل چهارم به تفصیل نشان داد که چگونه این طرح‌های مفهومی به کد عملی تبدیل شدند. ماژول‌های کلیدی سامانه، از جمله بخش آموزش گام‌به‌گام با راهنمای تصویری انگشتان، ماژول تست سرعت با قابلیت‌های پیشرفته شخصی‌سازی، و سیستم مدیریت کاربر با سازوکارهای امنیتی مبتنی بر JWT و Bcrypt، با موفقیت پیاده‌سازی شدند. همچنین، استراتژی تست واحد با استفاده از ابزارهای Vitest و Jest به کار گرفته شد تا از صحت و پایداری کدهای نوشته‌شده اطمینان حاصل شود.

نقطه اوج پروژه، مرحله ارزیابی در فصل پنجم بود. نتایج به دست آمده، موفقیت پروژه در دستیابی به اهدافش را به روشنی تأیید کرد. ارزیابی فنی با ابزارهایی مانند Lighthouse، امتیازات بسیار بالایی را در زمینه‌های عملکرد، دسترسی‌پذیری و رعایت بهترین شیوه‌ها ثبت کرد که نشان‌دهنده کیفیت بالای مهندسی محصول بود. از سوی دیگر، ارزیابی کاربرپذیری با حضور ۱۰ کاربر واقعی، نرخ موفقیت ۹۰ درصدی در انجام وظایف و میانگین امتیاز رضایت ۴.۷ از ۵ را به همراه داشت. این آمار نشان داد که سامانه نه تنها از نظر فنی کارآمد است، بلکه از دیدگاه کاربران نهایی نیز شهودی، ساده و رضایت‌بخش می‌باشد.

در نهایت، می‌توان نتیجه گرفت که این پروژه با موفقیت به تمام اهداف اولیه خود دست یافته است. محصول نهایی، یک پلتفرم کاربردی، پایدار و مدرن است که علاوه بر ارائه یک تجربه کاربری یکپارچه، دارای یک پنل مدیریت جامع برای تضمین سهولت در نگهداری و مدیریت محتوا در بلندمدت می‌باشد.

## ۲-۶- محدودیت‌های پروژه

نگاه واقع‌بینانه و انتقادی به هر پروژه علمی و مهندسی، مستلزم شناخت و پذیرش محدودیت‌های آن است. این محدودیت‌ها که اغلب ناشی از قیدهای زمانی، منابع و تمرکز بر اهداف اصلی پروژه هستند، به هیچ وجه از ارزش دستاوردهای آن نمی‌کاهند، بلکه زمینه را برای درک بهتر نتایج و تعریف کارهای آتی فراهم می‌کنند. مهم‌ترین محدودیت‌های این پروژه به شرح زیر است:

### محدودیت در گستره قابلیت‌ها (Scope Limitation):

عدم وجود اپلیکیشن موبایل بومی<sup>۱</sup>: این پروژه منحصرأً به عنوان یک وب اپلیکیشن توسعه یافته است. اگرچه طراحی آن واکنش‌گرا است و بر روی مرورگرهای موبایل قابل استفاده می‌باشد، اما فاقد یک اپلیکیشن بومی برای سیستم‌عامل‌های اندروید و iOS است. یک اپلیکیشن بومی می‌توانست با بهره‌گیری از قابلیت‌های سخت‌افزاری دستگاه و ارائه اعلان‌ها<sup>۲</sup>، تجربه کاربری عمیق‌تر و ماندگارتری را رقم بزند.

سیستم بازی‌وارسازی<sup>۳</sup> ساده: در حال حاضر، عنصر اصلی بازی‌وارسازی در سامانه به یک "جدول رتبه‌بندی" محدود می‌شود. سیستم‌های پیشرفته‌تر مانند اعطای نشان<sup>۴</sup>، سیستم امتیازدهی و سطوح کاربری<sup>۵</sup>، چالش‌های روزانه یا هفتگی، که می‌توانند به شدت بر انگیزه و تعامل بلندمدت کاربران تأثیرگذار باشند، در این فاز از پروژه پیاده‌سازی نشده‌اند.

محدودیت در محتوای آموزشی: متون و درس‌های موجود در سامانه به صورت ایستا و محدود تعریف شده‌اند. یک سیستم مدیریت محتوای پیشرفته (CMS) یا یک پنل ادمین که به مدیران سایت اجازه دهد به راحتی درس‌ها و آزمون‌های جدید اضافه کنند، در محدوده فعلی پروژه قرار نداشت.

### محدودیت در فرآیند ارزیابی:

مقیاس و جامعه آماری تست کاربردپذیری: آزمون کاربردپذیری با حضور ۱۰ کاربر انجام شد. اگرچه این تعداد برای شناسایی مشکلات اصلی و دریافت بازخوردهای کیفی ارزشمند کافی است، اما نتایج آن از نظر آماری قابل تعمیم به کل جامعه کاربران نیست. یک ارزیابی در مقیاس بزرگ‌تر می‌توانست دیدگاه‌های جامع‌تری را آشکار سازد.

عدم انجام مطالعه طولی<sup>۶</sup>: ارزیابی انجام‌شده، یک مطالعه مقطعی<sup>۷</sup> بود که تجربه اولیه کاربران با سامانه را می‌سنجید. برای اندازه‌گیری دقیق میزان تأثیر سامانه بر یادگیری و بهبود واقعی مهارت تایپ، نیاز به یک مطالعه طولی است که در آن، عملکرد گروهی از کاربران در یک بازه زمانی چند هفته‌ای یا چند ماهه رصد شود.

---

<sup>1</sup> Native

<sup>2</sup> Push Notifications

<sup>3</sup> Gamification

<sup>4</sup> Badges

<sup>5</sup> Levels

<sup>6</sup> Longitudinal Study

<sup>7</sup> Cross-sectional Study

### محدودیت‌های فنی:

عدم انجام تست بار<sup>۱</sup>: اگرچه عملکرد API ها با ارسال درخواست‌های متوالی تست شد، اما تست بار و استرس در مقیاس بزرگ (برای شبیه‌سازی هزاران کاربر همزمان) انجام نشده است. بنابراین، رفتار دقیق سیستم تحت فشارهای بسیار بالا به صورت عملی سنجیده نشده و مبتنی بر پیش‌بینی‌های معماری است.

### ۳-۶- کارهای آتی و پیشنهادات برای توسعه آینده

پروژه حاضر، با وجود موفقیت در دستیابی به اهداف اولیه، تنها یک نقطه شروع است. این سامانه یک زیربنای مستحکم و قابل توسعه فراهم آورده است که می‌توان بر روی آن، قابلیت‌های بسیار پیشرفته‌تر و هیجان‌انگیزتری را بنا نهاد. این بخش، با الهام از محدودیت‌های شناسایی شده و چشم‌انداز بلندمدت محصول، مسیرهایی را برای توسعه‌های آتی پیشنهاد می‌دهد:

توسعه اپلیکیشن‌های موبایل بومی: اولین و مهم‌ترین گام برای گسترش دامنه دسترسی، طراحی و توسعه اپلیکیشن‌های اختصاصی برای اندروید و iOS است. این کار نه تنها دسترسی را برای کاربران موبایل آسان‌تر می‌کند، بلکه امکان استفاده از قابلیت‌هایی مانند تمرین آفلاین و ارسال اعلان‌های یادآوری برای تشویق کاربران به تمرین منظم را نیز فراهم می‌آورد.

پایه‌سازی یک موتور بازی‌وارسازی پیشرفته: برای افزایش حداکثری انگیزه و ماندگاری کاربران، پیشنهاد می‌شود یک سیستم گیمیفیکیشن<sup>۲</sup> جامع طراحی شود. این سیستم می‌تواند شامل موارد زیر باشد:

سیستم امتیاز و سطح: کاربران با هر تمرین، امتیاز تجربه<sup>۳</sup> کسب کرده و سطح خود را ارتقا دهند.

نشان‌ها<sup>۴</sup> و دستاوردها<sup>۵</sup>: تعریف ده‌ها نشان مختلف برای دستاوردهای گوناگون (مانند "تایپ ۱۰۰۰ کلمه"، "یک هفته تمرین متوالی"، "دقت ۹۹٪").

گسترش پنل مدیریت: پنل ادمین فعلی را می‌توان با افزودن داشبوردهای تحلیلی و بصری گسترش داد. این داشبوردها می‌توانند آمارهای کلیدی مانند تعداد ثبت‌نام‌های روزانه، محبوب‌ترین درس‌ها، میانگین

<sup>۱</sup> Load Testing

<sup>۲</sup> Gamification

<sup>۳</sup> XP

<sup>۴</sup> Badges

<sup>۵</sup> Achievements

سرعت کاربران و گزارش‌های کلی از فعالیت سامانه را به مدیر نمایش دهند.

### ۳-۶-۲ چالش‌های روزانه و هفتگی: ارائه مأموریت‌های مشخص با پاداش‌های ویژه.

شخصی‌سازی هوشمند با استفاده از یادگیری ماشین: می‌توان با استفاده از الگوریتم‌های یادگیری ماشین، یک گام فراتر از شخصی‌سازی فعلی برداشت. سیستم می‌تواند با تحلیل عملکرد کاربر، خطاهای متداول او (مثلاً ضعف در تایپ با انگشت کوچک چپ یا کندی در فشردن یک کلید خاص) را شناسایی کرده و به صورت خودکار، درس‌ها و تمرین‌های اختصاصی برای رفع همان نقطه ضعف تولید کند. این رویکرد، فرآیند یادگیری را از یک مسیر ثابت به یک تجربه کاملاً پویا و انطباقی تبدیل می‌کند.

### گسترش محتوا و قابلیت‌های اجتماعی:

افزودن بسته‌های تمرینی تخصصی: ایجاد بخش‌های مجزا برای تمرین تایپ در حوزه‌های تخصصی مانند برنامه‌نویسی (با متون برگرفته از کدهای واقعی زبان‌های مختلف)، ادبیات، علوم و حقوق.

حالت رقابتی چندنفره<sup>۱</sup>: ایجاد اتاق‌هایی که در آن کاربران بتوانند به صورت زنده و همزمان با یکدیگر به رقابت پردازند و سرعت خود را مقایسه کنند.

ایجاد پنل مدیریت محتوا<sup>۲</sup>: برای سهولت در مدیریت و به‌روزرسانی محتوای آموزشی، پیشنهاد می‌شود یک پنل ادمین طراحی شود که به مدیران سایت اجازه دهد بدون نیاز به دخالت توسعه‌دهندگان، درس‌ها، آزمون‌ها و اخبار جدید را به سایت اضافه کنند.

تحلیل‌های آماری عمیق‌تر برای کاربر: می‌توان داشبورد کاربر را با ارائه گزارش‌های پیشرفته‌تر غنی‌تر کرد. برای مثال، نمایش یک نقشه حرارتی<sup>۳</sup> از کیبورد که نشان دهد کدام کلیدها بیشتر با خطا فشرده می‌شوند یا کندتر تایپ می‌شوند، می‌تواند بینش فوق‌العاده‌ای به کاربر بدهد.

با اجرای این پیشنهادات، سامانه فعلی می‌تواند از یک ابزار آموزشی کارآمد، به یک پلتفرم جامع، هوشمند و پیشرو در زمینه تقویت مهارت‌های دیجیتال تبدیل شود.

<sup>۱</sup> Multiplayer Mode

<sup>۲</sup> CMS

<sup>۳</sup> Heatmap

### پیوست ۱: لیست برنامه‌ها

در قسمت پیوست می‌توانید لیست برنامه‌های نوشته شده را بیاورید. البته برنامه‌ها و ضمائم آن را به صورت CD باید به گزارش نهایی ضمیمه کنید.



## منابع:

- [1] You, E. et al. (2024). *Vue.js Official Documentation*. Retrieved from <https://vuejs.org/guide/introduction.html>
- [2] Chopin, S. et al. (2024). *Nuxt 3 Official Documentation*. Retrieved from <https://nuxt.com/docs/getting-started/introduction>
- [3] Collina, M. et al. (2024). *Fastify Official Documentation*. Retrieved from <https://www.fastify.io/docs/latest/>
- [4] Schick, S. et al. (2024). *Prisma ORM Official Documentation*. Retrieved from <https://www.prisma.io/docs/>
- [5] The PostgreSQL Global Development Group. (2024). *PostgreSQL 16 Documentation*. Retrieved from <https://www.postgresql.org/docs/current/index.html>
- [6] Jones, M., Bradley, J., & Sakimura, N. (2015). *RFC 7519: JSON Web Token (JWT)*. Internet Engineering Task Force (IETF). Retrieved from <https://tools.ietf.org/html/rfc7519>
- [7] Wathan, A. et al. (2024). *Tailwind CSS Documentation*. Retrieved from <https://tailwindcss.com/docs/>
- [8] You, E. et al. (2024). *Pinia Official Documentation*. Retrieved from <https://pinia.vuejs.org/>
- [9] Atlassian. (n.d.). *Understanding RESTful APIs*. Retrieved from <https://www.atlassian.com/git/tutorials/what-is-rest>