

عنوان مقاله: توسعه فرانت اند، بک اند و فول استک چیست؟ شرح وظایف + تفاوت بین آنها

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: ۱۲ دی ۱۴۰۲

منبع: <https://nikamooz.com/the-difference-between-front-end-back-end-and-full-stack-developers>



تفاوت توسعه دهنده فرانت اند ، بک اند و فول استک چیست ؟ احتمالاً حداقل یک بار این سؤال در ذهن شما شکل گرفته است. این مشاغل حوزه کامپیوتر، تفاوت‌های کلیدی با یکدیگر دارند؛ بنابراین، برای ورود به آنها، بهتر است با نقاط تمایزشان آشنا باشید. در این مقاله، ابتدا به این پرسش پاسخ داده می‌شود که منظور از توسعه Frontend ، Backend و FullStack چیست، سپس تفاوت میان آنها و شرح وظایف مربوط به هر یک شرح داده خواهند شد.

منظور از توسعه فرانت اند ، بک اند و فول استک چیست ؟

توسعه فرانت اند (Frontend)، بک اند (Backend) و فول استک (Full Stack) به جوانب مختلف Web Development اشاره دارد که در هر یک، تمرکز و وظایف مشخصی تعریف می‌شوند. در این بخش، به بررسی این موارد می‌پردازیم:

توسعه Frontend

در توسعه فرانت اند یا همان سمت کلاینت (Client-Side) ، برنامه‌نویس روی اموری همچون ساخت [رابط کاربری](#) (User Interface) و [تجربه کاربری](#) (User Experience) تمرکز می‌کند. کاربران از طریق مرورگر وب، به صورت مستقیم با این بخش‌ها در تعامل دارند. CSS، جاوا اسکریپت (JavaScript)، زبان نشانه‌گذاری HTML و همچنین فریمورک‌های فرانت اند، همگی از تکنولوژی‌های مورد استفاده در توسعه Frontend به حساب می‌آیند. برای درک بهتر، ادامه این بخش را مطالعه کنید.



تکنولوژی های وب

توسعه دهندگان Frontend ترکیبی از تکنولوژی‌ها را به منظور ایجاد سمت Client اپلیکیشن استفاده می‌کنند. هسته این تکنولوژی‌ها عبارتند از:

- **HTML: زبان نشانه گذاری HTML** برای تعریف ساختار و محتوای صفحات وب استفاده می‌شود.
- **CSS:** با استفاده از CSS، استایل‌دهی صفحات وب، کنترل نمایش و Layout آن، امکان‌پذیر است. درحقیقت، CSS به منظور تعیین نحوه نمایش عناصر HTML کاربرد دارد.
- **جاوا اسکریپت:** این زبان، رفتار پویا و تعاملی را روی سمت کلاینت میسر می‌کند و برنامه‌نویس می‌تواند آن را با انواع کتابخانه‌ها و فریمورک‌ها همچون **ریکت** (React)، **انگولار** (Angular) و **Vue.js** به کار ببرد.

طراحی رابط کاربری (UI)

توسعه دهندگان فرانت اند با برقراری تعامل با طراحان UI/UX، در ایجاد رابط‌های کاربری جذاب، ریسپانسیو و کاربرپسند برای وب اپلیکیشن سهمیم خواهند بود. این افراد، طراحی Mockup ها و Wireframe ها را به رابط‌های وب مناسب و کاربردی تبدیل می‌کنند.

طراحی ریسپانسیو

اطمینان از اینکه **وب اپلیکیشن‌ها** (Web Applications) روی انواع دستگاه‌ها و سایزهای مختلف صفحه نمایش، قابل دسترس و ریسپانسیور هستند، از وظایف Frontend Developer است. با این کار، تجربه کاربری مطلوب خواهد بود.

سازگاری Cross-Browser

کدنویسی باید به گونه‌ای باشد که کدها بدون هیچ مشکلی روی مرورگرهای مختلف کار کنند تا بدین طریق، مشکلات مربوط به سازگاری رفع شوند و کاربر یک تجربه یکپارچه از آن وب اپلیکیشن یا وب سایت به‌دست آورد.

بهینه سازی کارایی

توسعه دهنده فرانت اند با پیاده‌سازی روش‌های خاصی، صفحات وب را بهینه‌سازی می‌کند. به‌عنوان مثال، اموری همچون کمینه‌سازی اندازه فایل‌ها، بهینه‌سازی تصاویر و استفاده از **بارگذاری ناهمگام** (Asynchronous Loading) برای Performance Optimization مفید هستند.

اسکریپت نویسی سمت کلاینت و کار با سیستم کنترل نسخه

به واسطه جاوا اسکریپت، Frontend Developer می‌تواند کنش‌گرایی و رفتار پویا را به صفحات وب بیافزاید؛ به این ترتیب، به ورودی کاربر رسیدگی شده و درخواست‌های ناهمگام از طریق API به بک اند ارسال می‌شوند. ضمن اینکه لازم است توسعه دهنده فرانت اند، با یک سیستم کنترل نسخه، مانند **گیت** (Git) کار کند تا بدین طریق بتواند تغییرات رخ داده شده در کدبیس (Codebase) را مدیریت و پیگیری کند.

کتابخانه ها و چارچوب های نرم افزاری Frontend

با بهره‌مندی از کتابخانه‌ها (Libraries) و فریمورک‌های سمت فرانت اند، فرآیند توسعه تسهیل پیدا می‌کند و کارایی بهبود می‌یابد. به همین دلیل، اغلب توسعه دهندگان فرانت اند از آن‌ها استفاده می‌کنند.

آزمایش (Test)

برنامه نویسی فرانت اند باید بتواند از روش‌هایی همچون [Unit Testing](#) و [Integration Testing](#) و [User Testing](#)، به منظور تشخیص و رفع مشکلات مربوط به کدهای سمت فرانت استفاده کند.

توسعه Backend

در توسعه بک اند یا سمت سرور (Server-Side)، برنامه‌نویس روی بخش Server-Side از یک Web Application کار می‌کند. این بخش از توسعه وب، به منظور رسیدگی به منطق پشت صحنه، پایگاه‌های داده، کانفیگ سرور و اطمینان از ارتباط مطلوب میان سرور و کلاینت به کار می‌رود. زبان‌های برنامه نویسی سمت سرور، فریمورک‌های Backend و Database ها به‌عنوان تکنولوژی‌های مورد استفاده در بک اند به شمار می‌روند. در ادامه، این موارد به‌صورت شفاف‌تر مورد بررسی قرار می‌گیرند.



زبان های برنامه نویسی Server Side

برنامه نویسان بک اند با استفاده از زبان های برنامه نویسی سمت سرور، کدهایی را می نویسند که روی سرور اجرا می شوند.

برخی از انواع زبان های برنامه نویسی سمت سرور عبارتند از:

- پایتون (Python): **زبان برنامه نویسی پایتون** به واسطه تطبیق پذیری و خوانایی بالا آن شناخته شده است.
- PHP: این زبان به وفور در توسعه وب مورد استفاده قرار می گیرد.
- روبی (Ruby): زبان روبی به دلیل سادگی و نحو (Syntax) آن مشهور است.
- جاوا (Java): این زبان تطبیق پذیری، در اپلیکیشن های سطح سازمانی به کار می رود.
- NodeJS (جاوا اسکریپت): شما می توانید با استفاده از زبان جاوا اسکریپت و NodeJS، توسعه سمت سرور را انجام دهید.

چارچوب های نرم افزاری سمت سرور

برنامه نویسان بک اند می توانند با کمک چارچوب های نرم افزاری سمت بک اند، فرآیند توسعه را تسهیل دهند. برخی از این فریمورک ها در ادامه لیست شده اند:

- اکسپرس (Express): شما می توانید از اکسپرس به عنوان یک چارچوب وب اپلیکیشن NodeJS بهره مند شوید و از انعطاف پذیری و مینیمال بودن آن استفاده کنید.
- جنگو (Django): این فریمورک سطح بالای پایتون، به گونه ای طراحی شده است که توسعه سریع را ترویج دهد.
- Ruby On Rails: این چارچوب نرم افزاری، برای زبان برنامه نویسی روبی توسعه داده شده است.

مدیریت پایگاه داده

توسعه دهندگان بک اند با پایگاه های داده کار می کنند تا به واسطه آن، بتوانند اموری همچون استخراج، دستکاری و ذخیره سازی داده ها را انجام دهند.

به صورت کلی، پایگاه های داده به دو گروه زیر طبقه بندی می شوند:

- پایگاه داده رابطه ای: در **پایگاه داده رابطه ای (RDBMS)**، داده ها در قالب جداول ساختاریافته به همراه روابط از قبل تعیین شده، سازماندهی می شوند. **SQL Server**، PostgreSQL و MySQL برخی از نمونه های آن هستند.
- پایگاه داده غیر رابطه ای (NoSQL): **پایگاه داده غیر رابطه ای**، به منظور ذخیره سازی و استخراج داده های بدون ساختار (Non-Structured) یا نیمه ساختاریافته (Semi-Structured) کاربرد دارند. Cassandra و MongoDB هر دو NoSQL به حساب می آیند.

توسعه API

به زبان ساده، با استفاده از API، نحوه تعامل کامپوننت‌ها مشخص می‌شوند. در حقیقت، برنامه نویسی بک اند با توسعه API، به بخش فرانت اند این امکان را می‌دهد که با سرور ارتباط برقرار کند. معمولاً RESTful API ها برای وب اپلیکیشن مورد استفاده قرار می‌گیرند. در عمل، توسعه API در یکپارچه‌سازی انواع سیستم‌ها و سرویس‌ها نقش کلیدی دارد و با تقویت هم‌کنش‌پذیری (Interoperability) و ترویج یک رویکرد ماژولار و مقیاس‌پذیر در توسعه نرم افزار، کاربرد غیرقابل انکاری از خود نشان می‌دهد.

احراز هویت (Authentication) و اعطای مجوز (Authorization)

توسعه دهندگان بک اند با پیاده‌سازی مکانیزم‌های احراز هویت (Authentication) و اعطای مجوز (Authorization)، تضمین می‌کند که فقط کاربران مجاز امکان دسترسی به بخش‌های خاصی از اپلیکیشن را داشته باشند. Developer Backend باید با روش‌های رایج احراز هویت، از جمله [احراز هویت مبتنی بر توکن](#) (Token-Based Authentication) و OAuth، آشنا باشد.

نگهداری و استقرار سرور

توسعه دهنده Backend وظیفه استقرار اپلیکیشن‌ها روی سرورها را برعهده دارد و باید از مقیاس‌پذیری و مدیریت کارآمد منابع سرور اطمینان پیدا کند. علاوه بر این، مواردی همچون نظارت بر کارایی سرور، رسیدگی به Patch های امنیتی و رفع مشکلات مربوط به کارایی سمت سرور، همگی از وظایف توسعه دهنده بک اند به شمار می‌روند.

منطق سمت سرور

توسعه دهندگان بک اند منطق تجاری اپلیکیشن را پیاده‌سازی می‌کنند؛ این کار، پردازش ورودی کاربر، رسیدگی به داده‌ها و اجرای محاسبات پیچیده را در برمی‌گیرد.

یکپارچه سازی با سرویس های شخص ثالث

معمولاً در توسعه بک اند، یکپارچه‌سازی با API ها یا سرویس‌های خارجی، به هدف بهبود کارایی اپلیکیشن انجام می‌شود. از این رو، برنامه نویسی بک اند باید توانایی Integrate کردن سرویس‌های شخص ثالث (Third-Party Services) را داشته باشد.

توسعه Full Stack

توسعه فول استک داشتن تخصص در هر دو بخش فرانت اند، بک اند و فناوری‌های آن‌ها را شامل می‌شود. این یعنی، فول استک دولوپر باید توانایی و مهارت کافی برای کار روی جوانب مختلف وب اپلیکیشن را داشته باشد. برای آشنایی با وظایف برنامه نویس فول استک، پیشنهاد می‌شود [مقاله Full Stack چیست؟ + ۷ وظیفه Full Stack Developer](#) را نیز مطالعه کنید. در توسعه فول استک، ترکیبی از تکنولوژی‌های فرانت اند و بک اند، براساس نیازمندی‌های پروژه استفاده می‌شوند. به همین دلیل، لازم است برنامه نویس فول استک در طیف وسیعی از انواع تکنولوژی‌ها تخصص داشته باشد تا بدین طریق، بتواند در جوانب مختلف فرآیند توسعه سهیم باشد. شما می‌توانید با رجوع به [راهنمای مسیر شغلی Full Stack Developer](#)، از نحوه استخدام در این شغل، آینده و بازار کار آن با خبر شوید.



مقایسه توسعه فرانت اند، بک اند و فول استک

در این بخش، بارزترین تفاوت بین توسعه فرانت اند، بک اند و فول استک را از نقطه نظرات مختلف بررسی خواهیم کرد.

- **نوع تخصص:** در توسعه فرانت اند، جنبه‌های بصری و تعاملی وب سایت به صورت تخصصی مورد توجه قرار می‌گیرند و در بخش بک اند، اموری همچون منطق سمت سرور، دیتابیس و API حائز اهمیت است. ضمن اینکه توسعه فول استک با هدف کار کردن روی تمامی اپلیکیشن، شامل فرانت اند و بک اند به کار می‌رود.
- **دامنه کاری:** توسعه بک اند و فرانت اند دارای یک دامنه یا همان Scope خاص است؛ در حالی که در توسعه فول استک، درک هر دوی این دو بخش و فعالیت در آن‌ها ضروری است.
- **نحوه تعامل:** معمولاً برنامه نویسان بک اند و فرانت اند، برای ساخت یک وب اپلیکیشن تعامل می‌کنند؛ اما فول استک دولوپر باید دائماً میان تسک‌های فرانت اند و بک اند جابه‌جا شود تا تعاملات میان تیم‌های کوچک‌تر تسهیل یابند.

تا این بخش از مقاله، به چستی فرانت اند، بک اند و فول استک و تفاوت‌های آن‌ها پرداخته شد. حال در ادامه، نقاط تمایز برنامه نویسان فرانت اند، بک اند و فول استک اشاره می‌شود تا شما با جزئیات وظایف آن‌ها آشنا شوید.

تفاوت بین توسعه دهنده فرانت اند، بک اند و فول استک چیست؟

برنامه نویسان فرانت اند، بک اند و فول استک، هر یک وظایف و تسک‌های منحصر به فردی در فرآیند توسعه وب دارا هستند.

شرح وظایف توسعه دهنده فرانت اند

موارد زیر توسط توسعه دهنده فرانت اند انجام می‌شوند:

- طراحی و پیاده‌سازی عناصر بصری یک وب سایت (از طریق HTML، CSS و جاوا اسکریپت)
- اطمینان از ریسپانسیو و کاربرپسند بودن طراحی‌ها روی انواع مرورگرها و دستگاه‌ها
- یکپارچه‌سازی سرویس‌های سمت بک اند از طریق فراخوانی API
- پیاده‌سازی Validation سمت کلاینت برای ورودی‌های کاربر
- تعامل و مشارکت با طراحان UI/UX برای ساخت رابط‌های جذاب و قابل درک

شرح وظایف توسعه دهنده بک اند

وظایف توسعه دهنده Backend به شرح زیر است:

- طراحی و ایجاد معماری سرور
- ساخت API برای برقراری ارتباط فرانت اند و بک اند
- مدیریت پایگاه داده و پیاده‌سازی روش‌هایی برای بهینه‌سازی
- رسیدگی به فرآیندهای امنیت سمت سرور، احراز هویت و اعطای مجوز
- پیاده‌سازی منطق سمت سرور برای پردازش داده‌ها و قوانین کسب و کار
- اطمینان از مقیاس‌پذیری و کارایی پردازش‌های سمت سرور

شرح وظایف توسعه دهنده فول استک

در فرآیند توسعه نرم افزار، توسعه دهنده فول استک وظایف زیر را برعهده دارد:

- مدیریت و پیاده‌سازی ویژگی‌ها (Features) روی هر دو بخش کلاینت و سرور
- درک دقیق تمامی پردازش‌های توسعه وب، از طراحی تا استقرار (Deployment)
- تعامل و مشارکت چندمنظوره با تیم‌ها و رسیدگی به تسک‌های مدیریت پروژه
- رفع و اشکال‌زدایی مشکلات مربوط به تمام بخش‌های اپلیکیشن
- یکپارچه‌سازی کامپوننت‌های فرانت اند و بک اند
- تخصص و خبرگی در طیف وسیعی از انواع تکنولوژی‌ها

جمع بندی: فرق توسعه فرانت اند ، بک اند و فول استک چیست ؟

توسعه فرانت اند ، بک اند و فول استک ، هر یک فرآیندها و وظایف مجزایی را شامل می‌شوند که همگی در کنار هم، به توسعه نرم افزار منجر می‌شوند. در این مقاله، به نیازمندی‌های لازم برای هر یک از بخش‌های Backend ، Frontend و Fullstack اشاره شدند و شرح وظایف برنامه‌نویسان مربوطه نیز به صورت فهرست‌وار مورد بررسی قرار گرفتند. حال شما می‌توانید با کمک این راهنما، تمایزهای میان این مشاغل را درک کنید و با روند توسعه به شکل شفاف آشنایی پیدا کنید.