



عنوان مقاله: Full Stack چیست؟ + ۷ وظیفه Full Stack developer

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: ۳ دی ۱۴۰۲

منبع: <https://nikamooz.com/what-is-full-stack/>

فول استک (Full Stack) یک رویکرد جامع در توسعه نرم افزار به حساب می‌آید که در آن، برنامه‌نویس به تکنولوژی‌های مربوط به سمت **بک اند (Backend)** و **فرانت اند (Frontend)** تسلط دارد. درحقیقت، توسعه فول استک به دلیل تطبیق‌پذیری بالای آن، به برنامه‌نویسان این امکان را می‌دهد تا بر روی جوانب گوناگون یک پروژه کار کنند و تمامی آن استک را به خوبی درک کنند. در این مقاله، ابتدا فول استک و وظایف Full Stack Developer شرح داده می‌شوند، سپس اجزای اصلی فول استک به همراه مزایا و جایگاه آن در بازار کار، مورد بررسی قرار می‌گیرند.

### فول استک دولوپر چه کسی است ؟

Full Stack Developer یک فرد حرفه‌ای و متخصص است که مهارت و دانش کافی برای کار کردن روی هر دو بخش یک **وب اپلیکیشن**، یعنی سمت کلاینت (Frontend) و سمت سرور (Backend) را دارد. این شغل به گونه‌ای است که فرد باید دانش عمیقی از تمام فرآیند توسعه نرم‌افزار داشته باشد؛ به طوری که بتواند طراحی رابط کاربری (UI) تا مدیریت پایگاه داده و منطق سرور را به راحتی انجام دهد. فول استک دولوپر به عنوان یک عضو سازگار و توانمند، در تمام جوانب گوناگون یک پروژه سهمیم است و شخص پراهمیتی در صنعت تکنولوژی قلمداد می‌شود.



## وظایف Full Stack Developer

اصلی‌ترین وظایف فول استک دولوپر به شرح زیر است:

### ۱- توسعه فرانت اند (Frontend Development)

- طراحی و پیاده‌سازی رابط کاربری (UI): این کار با استفاده از [زبان نشانه گذاری HTML](#)، جاوا اسکریپت (JavaScript) و CSS انجام می‌شود.
- چارچوب‌های نرم‌افزاری Frontend: برنامه نویسی فول استک باید از فریمورک‌های فرانت اند، همچون [ری اکت](#) (React)، [انگولار](#) (Angular) یا [Vue.js](#) برای ایجاد [تجربه کاربری](#) (UX) پویا و تعاملی استفاده کند.
- اطمینان از جذابیت طراحی: فول استک دولوپر باید بررسی کند که آیا طراحی سایت از نظر بصری و [واکنش گرا](#) (Responsive) با قواعد تجربه کاربری، هم‌تراز است یا خیر.
- طراحی واکنش‌گرا: ایجاد کردن طراحی‌هایی که برای دستگاه‌ها و صفحه نمایش‌های گوناگون بهینه باشند.
- سازگاری Cross-Browser: برنامه نویسی فول استک باید مطمئن شود که کدهای فرانت اند به صورت یکپارچه و صحیح، روی انواع مرورگر کار می‌کنند.

### ۲- توسعه بک اند (Backend Development)

- منطق سمت سرور: نوشتن منطق سمت سرور و پیاده‌سازی آن از طریق زبان‌های برنامه‌نویسی همچون [Node.js](#)، [پایتون](#) (Python)، [روبی](#) (Ruby)، جاوا یا PHP، از وظایف برنامه نویسی فول استک است.
- فریمورک‌های بک اند: یک Full Stack Developer باید فریمورک‌های Backend از جمله Express.js، [جنگو](#) (Django)، [اسپرینگ](#) (Spring) یا [Ruby On Rail](#) را به منظور ساختاردهی و سازماندهی کارآمد کدها به کار ببرد.
- مدیریت پایگاه داده: رسیدگی به استخراج و ذخیره‌سازی داده‌ها، پیاده‌سازی منطق کسب و کار روی بخش سمت سرور، طراحی اسکیمای (Schema) و بهینه‌سازی کوئری‌ها، از وظایف کلیدی برنامه نویسی فول استک به شمار می‌رود. توجه شود که طبق نیازمندی‌های پروژه، فرد باید بتواند با [پایگاه داده رابطه ای](#) (RDBMS) یا [دیتابیس غیررابطه‌ای](#) (NoSQL) کار کند.
- توسعه API: فول استک دولوپر با ایجاد و نگهداری API ها، ارتباط میان کامپوننت‌های فرانت اند و بک اند را تسهیل می‌بخشد. این شخص باید درک صحیحی از API های GraphQL یا RESTful داشته باشد و بتواند آن‌ها را در اپلیکیشن یکپارچه‌سازی کند.

### ۳- یکپارچه سازی فول استک (Full Stack Integration)

- توسعه سرتاسری: توسعه دهنده فول استک با یکپارچه‌سازی اجزای فرانت اند و بک اند، راه حل‌های کارآمد و سریع سرتاسری (End-To-End) را ایجاد می‌کند.
- یکپارچه‌سازی و جریان داده: اطمینان از بدون اشکال بودن جریان داده‌ها میان بک اند و فرانت اند، مدیریت سازگاری داده‌ها و همچنین رسیدگی به تبدیل‌های آن‌ها، همگی از وظایف برنامه نویسی فول استک است.

#### ۴- سیستم کنترل نسخه (Version Control) و مشارکت

- استفاده از سیستم کنترل نسخه: فول استک دولوپر با به کارگیری سیستم‌های کنترل نسخه، مانند **گیت (Git)**، تغییرات را پیگیری می‌کند، با سایر توسعه‌دهندگان مشارکت دارد و ریپازیتوری‌های کد را مدیریت خواهد کرد.
- **تعامل و مشارکت:** برنامه نویس Full Stack باید بتواند به صورت مؤثر با سایر اعضای تیم مانند طراح (Designer)، مدیر محصول (Product Manager) و سایر توسعه‌دهندگان کار کند تا یک نرم‌افزار باکیفیت به مشتری برساند.

#### ۵- استقرار و زیرساخت

- **استقرار (Deployment):** استقرار اپلیکیشن‌ها به پلتفرم‌های ابری و سرورها توسط Full Stack Developer انجام می‌شود و بدین طریق، صحیح بودن نحوه کانفیگ و بهینه‌سازی کارایی تضمین خواهد شد.
- **وب سرورها:** برنامه نویس فول استک باید بتواند وب سرورها، مانند Nginx یا آپاچی (Apache) را به درستی درک کند و پیکربندی آن‌ها برای اپلیکیشن‌های Host را انجام دهد.

#### ۶- آزمایش و رفع اشکالات

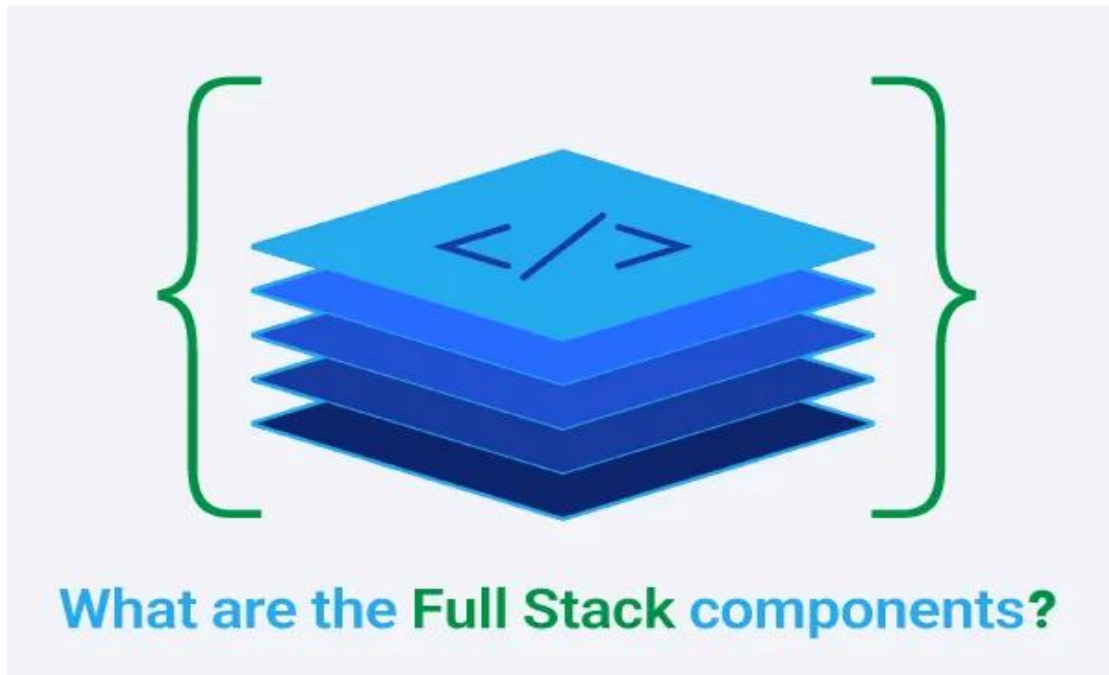
- **آزمایش (Test):** یکی از وظایف پراهمیت هر برنامه‌نویس، به‌ویژه فول استک دولوپر، انجام تست است. درحقیقت، باید Full Stack Developer روی هر دو بخش فرانت اند و بک اند تست اعمال کند تا از بابت پایداری و قابل اکتفا بودن اپلیکیشن، خیالمان راحت باشد.
- **رفع اشکالات (Debugging):** تشخیص مشکلات و خطاهایی که ممکن است درطول توسعه و تست ایجاد شوند و رفع آن‌ها، برعهده Full Stack Developer خواهد بود.

#### ۷- ملاحظات امنیتی

- **پیاده‌سازی SSL و TLS:** با انجام شدن این کار توسط Full Stack Developer، تضمین می‌شود انتقال داده‌ها به صورت امن روی یک شبکه صورت می‌گیرد.
- **احراز هویت (Authentication) و اعطای مجوز (Authorization):** برنامه نویس فول استک با پیاده‌سازی Access Control های امن، منابع اپلیکیشن و داده‌های کاربر را محافظت می‌کند.

## اجزای فول استک چیست؟

توسعه فول استک، مدیریت پایگاه داده، توسعه بخش‌های Backend و فرانت اند را دربرمی‌گیرد. ضمن اینکه در فول استک، یکپارچه‌سازی این اجزا و استقرار وب اپلیکیشن نیز انجام خواهد شد. مهم‌ترین اجزای فول استک عبارتند از:



## ۱- زبان‌های برنامه‌نویسی و فریم‌ورک‌ها

انتخاب یک زبان از میان [انواع زبان‌های برنامه‌نویسی](#)، به نیازمندی‌های پروژه و تخصص توسعه‌دهنده بستگی دارد. برای توسعه فرانت اند از CSS، جاوا اسکریپت و زبان نشانه‌گذاری HTML استفاده می‌شود و برای توسعه بک اند، پایتون، روبي، جاوا و سایر موارد به کار می‌روند. به منظور مدیریت پایگاه داده، مواردی همچون MySQL، اوراکل (Oracle) و MongoDB مورد استفاده قرار می‌گیرند. علاوه بر این، چارچوب‌های نرم‌افزاری یا همان فریم‌ورک‌ها، یک رویکرد ساختاریافته برای ایجاد و مدیریت وب اپلیکیشن‌ها به شما ارائه می‌دهند و معمولاً ماژول‌های از پیش ساخته‌شده، کتابخانه‌ها و ابزارهای کمکی مختلفی را دارا هستند.

## ۲- Best Practice ها و قوانین توسعه برای Full-Stack

- **اصل Separation Of Concerns:** این قانون کمک می‌کند تا کدها به صورت ساختارمند و قابل نگهداری حفظ شوند. این اصل بر این حکم می‌کند که هر کامپوننت اپلیکیشن باید تنها مسئول یک جنبه از اپلیکیشن باشد.
- **اصل DRY:** با استناد به اصل Don't Repeat Yourself، توسعه‌دهنده نباید کد یا منطق را مجدد تکرار کند تا یکپارچگی و قابلیت نگهداری کد بیس (Codebase) حفظ شود.
- **توسعه چابک (Agile Development):** به واسطه مدیریت پروژه به روش چابک (Agile)، رضایت مشتری حاصل می‌شود و نرم‌افزار، با کیفیت خاصی ارائه خواهد شد. پیشنهاد می‌شود [مقاله روش های اولویت بندی](#) **بک لاگ در اسکرام** را مطالعه کنید تا فرآیندهای مربوط به Backlog را بهتر درک کنید.
- **توسعه مبتنی بر تست (TDD):** در این رویکرد، برنامه‌نویس پیش از نوشتن کد، روی تست‌نویسی متمرکز می‌شود. TDD یا همان Test Driven Domain، به برنامه‌نویس کمک می‌کند تا خطاها را در مراحل اولیه توسعه تشخیص دهد و خیالش از بابت کارکرد مورد انتظار کدها راحت باشد.
- **کدنویسی تمیز و قابل نگهداری:** فول استک دولوپر باید روی نوشتن کدهای تمیز، خوانا و قابل نگهداری تمرکز کند و استانداردهای کدنویسی، Name Convention ها و Best Practice ها را دنبال کند.
- **کدنویسی ماژولار:** به عنوان برنامه نویس فول استک، لازم است که فرد، مهارت کدنویسی به صورت ماژولار را داشته باشد. به این ترتیب، اموری همچون تست، استفاده مجدد و نگهداری این کدها تسهیل می‌یابند و می‌توان ماژول‌ها را در بخش‌های گوناگون اپلیکیشن، مورد استفاده قرار داد.
- **استفاده از CI/CD:** روش «ادغام مداوم تحویل مداوم» (Continuous Integration and Deployment) یک رویکرد توسعه است که با کمک آن، فرآیندهای ساخت، تست و استقرار خودکار سازی می‌شوند. وجود CI/CD در تشخیص زودهنگام خطاها و اطمینان از قابل استقرار بودن کد مؤثر است.
- **استفاده از الگوهای طراحی (Design Patterns):** فول استک دولوپر می‌تواند با به‌کارگیری [دیزاین پترن ها](#)، مسائل رایج توسعه نرم‌افزار را حل کند. این کار، در ایجاد اپلیکیشن‌های نرم‌افزاری انعطاف‌پذیر، قابل نگهداری و مقیاس‌پذیر، مفید خواهد بود.
- **بهینه‌سازی کارایی:** Full Stack Developer می‌تواند با کمینه‌سازی اندازه فایل‌ها، کاهش تعداد درخواست‌ها (Requests) و استفاده از روش‌های Caching، کارایی وب اپلیکیشن‌ها را بهینه‌سازی کند.

## ۳- تست و کیفیت توسعه

در توسعه فول استک، تست‌نویسی و بررسی کیفیت، اهمیت فراوانی دارند. برای این کار، دو رویکرد [Unit Testing](#) و [Integration Testing](#) قابل استفاده هستند. در Unit Testing، هرکدام از کامپوننت‌های اپلیکیشن، به صورت جداگانه بررسی می‌شوند؛ بنابراین، مشخص خواهد شد که آیا هر یک از اجزای اپلیکیشن، به درستی کار می‌کنند یا خیر. در Integration Testing، تعاملات میان اجزای مختلف اپلیکیشن مورد بررسی و توجه قرار می‌گیرند و با انجام آن، Full Stack Developer از صحت کارکرد اجزای اپلیکیشن در کنار هم اطمینان می‌یابد.

در این بخش از مقاله، با اجزای فول استک آشنا شدید و حالا، در ادامه، به برخی از مزیت‌های بارز Full Stack اشاره خواهد شد.

## مزایای فول استک چیست؟

مزایای توسعه فول استک عبارتند از:

- درک End To End و توانایی کارکردن روی تمام استک اپلیکیشن
- تطبیق پذیری و سازگاری بالا
- توانایی حل مسئله به صورت کارآمد و همچنین، درک بهتر تجربه کاربری (UX)
- کاهش سربارهای مربوط به ارتباط و هماهنگی
- برعهده گرفتن مالکیت (Ownership) تمام پروژه یا Feature های آن
- مقرون به صرفه برای تیم‌های توسعه کوچک و استارت آپ‌ها
- پرتوتایپینگ سریع و همچنین، امکان سازگاری با تغییرات تکنولوژی



## آیا هنوز تقاضا برای توسعه دهنده Full Stack وجود دارد؟

با استناد به فرصت‌های شغلی موجود در داخل و خارج از کشور، تقاضا برای استخدام برنامه نویس فول استک یا همان Full Stack Developer بالا است. این افراد، یک عامل کلیدی و ارزشمند برای هر سازمان و کسب و کاری به شمار می‌روند و به واسطه دانش وسیع خود، این امکان را دارند که به صورت سازگار، انعطاف‌پذیر و چابک، با سایر اعضای تیم کار کنند. درحقیقت، برنامه نویسان فول استک می‌توانند براساس نیازهای کسب و کار، تمرکز خود را روی هر دو بخش وب اپلیکیشن، یعنی فرانت اند و بک اند، قرار دهند و با به‌کارگیری ترفندهایی برای بهینه‌سازی پایگاه داده، کارایی سیستم را افزایش دهند.

## جمع بندی: چگونه برنامه نویس فول استک شویم؟

شما می‌توانید با به‌دست آوردن مهارت‌های لازم برای Full Stack Developer ، برای استخدام در این حرفه اقدام کنید. البته باید توجه کرد که فول استک دولوپر شدن یک مسیر توأم با یادگیری و پیشرفت مداوم است و به منظور رسیدن به آن، صبر، تمرین و مواجهه با چالش‌های گوناگون، امری ضروری است. در این مطلب، به این پرسش پاسخ داده شد که فول استک چیست و چه اجزایی را شامل می‌شود. علاوه بر این، وظایف برنامه نویس Full Stack به‌طور کامل فهرست شدند. ادامه این مسیر برعهده شما خواهد بود؛ شما می‌توانید با فراگیری گام‌به‌گام و توسعه هرچه بیشتر دانش خود، برای استخدام شدن در این حرفه اقدام کنید.