پیاده سازی معماری نرم افزار

معماری نرم افزار در یک سیستم، نشان‌دهنده تصمیمات طراحی ساختار و رفتار کلی در آن سیستم است. معماری نرم افزار تعیین‌کننده کیفیت، عملکرد، قابلیت نگهداری و
معماری یک سیستم نرم افزاری معمولا محدود به یک .موفقیت کلی سیستم است
سبک معماری واحد نیست، بلکه ترکیبی از سبک‌هایی است که اغلب در سیستم
مورد استفاده قرار می‌گیرد.
معماری نرم افزار به عنوان طرح اولیه‌ای برای سیستم و پروژه در حال توسعه عمل می‌کند و می‌تواند بعداً از آن برای برون‌یابی استفاده گردد.

اجزا معماری نرم افزار:

جزییات پیاده سازی(ساختار پوشه مخزن)
تصمیمات طراحی پیاده سازی
فناوری های مورد استفاده
تصمیمات طراحی سیستم
تصمیمات زیر ساختی

مفاهیم
مدل کلاینت سرور:
کلاینت-سرور یک ساختار کاربردی توزیع شده است که وظایف یا حجم کاری را بین ارائه‌دهندگان یک منبع یا سرویس، به نام سرور، و درخواست‌کنندگان خدمات، به نام کلاینت، تقسیم می‌کند.
به زبان ساده، کلاینت یک برنامه کاربردی است که نوعی اطلاعات را درخواست می‌کند یا اقداماتی را انجام می‌دهد و سرور برنامه‌ای است که اطلاعات را ارسال می‌کند یا مطابق با فعالیت کاربر، اقداماتی را انجام می‌دهد. کلاینت‌ها معمولاً توسط برنامه‌های فرانت‌اند که روی وب یا برنامه‌های تلفن همراه اجرا می‌شوند، ارائه می‌شوند.

API: روشی است که این دو بخش برای برقراری ارتباط استفاده می‌کنند.
مجموعه‌ای از قوانین تعریف شده به حساب می‌آید که نحوه برقراری ارتباط برنامه‌ای قراردادی بین کلاینت و سرور است APIرا با برنامه دیگر مشخص می‌کند. در واقع را C خواهد بود. اگر B را بفرستید، پاسخ من همیشه Aکه مثلا بیان می‌کند: «اگر خواهد بود» و تا انتها به همین شکل ادامه پیدا Dبفرستید، پاسخ من همیشه APIمی‌کند.

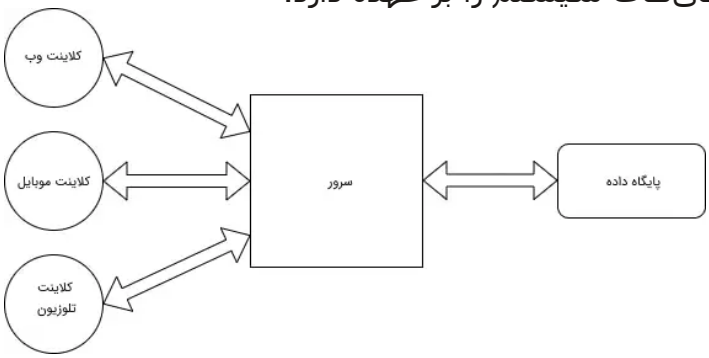
ماژولار بودن:
ماژولار بودن نرم افزار، تجزیه نرم افزار به بخش‌های کوچک‌تر با رابطهای استاندارد است. ما می‌خواهیم محصولاتی با قطعه کدهای قابل استفاده مجدد ایجاد کنیم، بنابراین فقط یک‌بار یک قابلیت عملکردی (فیچر | فانکشن) را پیاده‌سازی کرده و سپس از آن مکرراً استفاده می‌کنیم.

معماری لایه ای:

معماری لایه‌ای، کارکردها و مسئولیت‌ها را در پوشه‌ها و فایل‌های مختلف تقسیم‌بندی می‌کند و امکان برقراری ارتباط مستقیم فقط بین برخی از پوشه‌ها و فایل‌ها ممکن خواهد بود.

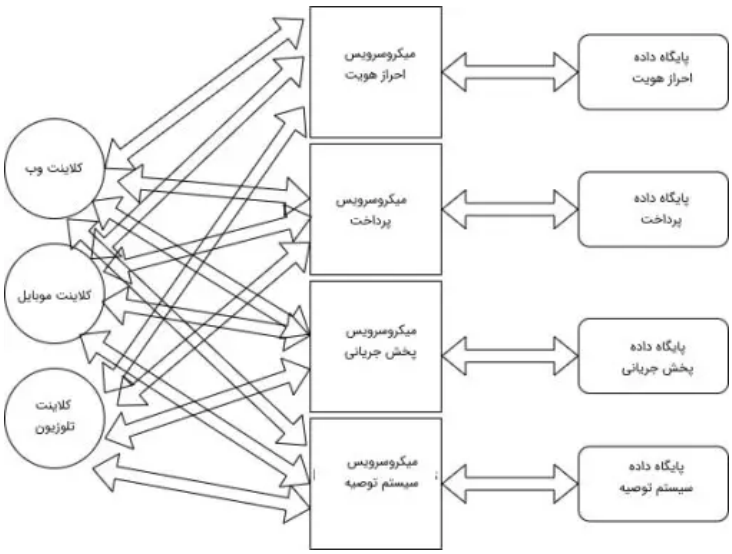
لایه برنامه
۱- راه اندازی و تنظیمات اولیه سرور
۲ – اتصال به مسیر ها
لایه مسیر ها
۱-تعریف مسیر ها
۲- اتصال به کنترلر
لایه کنترلر
۱-منطق عملیاتی
۲-اتصال به لایه مدل

- **معماری مونولیتیک:**
مونولیتیک یا یکپارچه نامیده می‌شود زیرا یک برنامه واحد در سمت سرور وجود دارد که مسئولیت تمام ویژگی‌های سیستم را بر عهده دارد.



مزیت اصلی طراحی مونولیتیک، سادگی آن است. عملکرد آن و تنظیمات مورد نیاز آن آسان است و به همین دلیل، شروع توسعه اکثر برنامه‌ها با این روش انجام می‌شود.

معماری میکروسرویس:
معماری میکروسرویس، تقسیم ویژگی‌های سمت سرور به چندین سرور کوچک است که تنها مسئول یک یا چند ویژگی خاص هستند. در معماری مونولیتیک فقط یک سرور واحد داشتیم که مسئولیت تمام ویژگی‌ها را برعهده داشت. پس از پیاده‌سازی میکروسرویس‌ها، چندین سرور کوچک خواهیم داشت که هر یک زیرمسئولیت‌هایی را بر عهده دارند.



- در معماری میکروسرویس لازم نیست کل قسمت بک‌اند به یکباره مقیاس‌بندی شود
- ویژگی‌ها وابستگی کمتری به یکدیگر دارند، به این معنی که ما می‌توانیم آن‌ها را به‌طور مستقل توسعه دهیم و اجرا کنیم.

الگو بک اند برای فرانت اند:
اجرای یک لایه میانی بین برنامه‌های فرانت‌اند و میکروسرویس‌ها. این لایه تمام درخواست‌های فرانت‌اند را دریافت می‌کند، آنها را به میکروسرویس مربوطه هدایت می‌کند، پاسخ میکروسرویس را دریافت می‌کند و سپس پاسخ را به برنامه فرانت‌اند مربوطه هدایت می‌کند. این معماری، الگوی «بک‌اند برای فرانت‌اند» (BFF | Back-end For Front-end) نام دارد.

معماری مبتنی بر رویداد:
این معماری بر اساس رویداد های اتفاق افتاده توسط یوزر مانند کلیک کردن و اسکرول هست در این معماری، اجزاء سیستم (کامپوننت‌ها) به جای ارتباط مستقیم، از طریق ارسال و دریافت رویدادها با هم تعامل می‌کنند.