

مدل MVC چیست؟

مخفف Model View Controller است که یک الگوی معماری نرم افزار محبوب برای توسعه برنامه های کاربردی، به خصوص برنامه های وب محسوب می شود.

هدف اصلی MVC جداسازی نگرانی ها و تقسیم یک برنامه بزرگ به بخش های کوچکتر و قابل مدیریت تر است که هر کدام وظایف مشخصی را بر عهده دارند. این امر منجر به افزایش وضوح، قابلیت نگهداری و قابلیت تست کد می شود.

اجزای اصلی MVC:

1. مدل (Model): نماینده داده ها و منطق کسب و کار برنامه است.
با پایگاه داده تعامل دارد و اطلاعات را بازیابی، ذخیره، به روز رسانی و حذف می کند.
به طور مستقیم با کاربر یا رابط کاربری ارتباط برقرار نمی کند.

2. نما (View):
مسئول نمایش داده ها به کاربر است. از طریق رابط کاربری با کاربر تعامل دارد.
اطلاعات را از مدل دریافت می کند و آن را به شکلی قابل فهم برای کاربر ارائه می دهد.
نباید حاوی هیچ گونه منطق کسب و کار باشد.

3. کنترل کننده (Controller):
واسطه ای بین مدل و نما است.
درخواست های کاربر را از طریق رابط کاربری دریافت می کند.
با مدل برای بازیابی، ذخیره یا به روز رسانی داده ها تعامل برقرار می کند.
داده های مورد نیاز را از مدل به نما ارسال می کند.
مسئول مدیریت رویدادها و اعمال منطق برنامه است.

نحوه عملکرد MVC:

1. کاربر با برنامه تعامل برقرار می‌کند (مثلاً با کلیک بر روی دکمه یا وارد کردن اطلاعات).

این اقدام یک **رویداد** را به **کنترل‌کننده** ارسال می‌کند.

1. **کنترل‌کننده** رویداد را پردازش می‌کند و با **مدل** برای بازیابی یا به‌روزرسانی داده‌های مورد نیاز تعامل برقرار می‌کند.

2. **مدل** داده‌های درخواستی را به **کنترل‌کننده** ارائه می‌دهد.

3. **کنترل‌کننده** داده‌ها را به **نما** ارسال می‌کند.

4. **نما** داده‌ها را به شکلی قابل فهم برای کاربر قالب‌بندی می‌کند و آن را در رابط کاربری نمایش می‌دهد.

مزایا استفاده از MVC:

- **وضوح:** جداسازی نگرانی‌ها کد را به بخش‌های کوچکتر و قابل درک‌تر تقسیم می‌کند که نگهداری و به‌روزرسانی آن را آسان‌تر می‌کند.
- **قابلیت نگهداری:** هر بخش از MVC وظایف مشخصی را بر عهده دارد که باعث می‌شود تغییرات در یک بخش بر بخش‌های دیگر تأثیر کمتری بگذارد.
- **قابلیت تست:** هر بخش از MVC را می‌توان به طور جداگانه تست کرد که به اطمینان از کیفیت کلی برنامه کمک می‌کند.
- **قابلیت انعطاف‌پذیری:** MVC یک الگوی معماری انعطاف‌پذیر است که می‌توان از آن برای انواع مختلف برنامه‌های کاربردی استفاده کرد.
- **معایب استفاده از MVC:**
- **پیچیدگی:** MVC می‌تواند برای برنامه‌های ساده پیچیده باشد.
- **منحنی یادگیری:** یادگیری نحوه استفاده از MVC نیاز به زمان و تلاش دارد.
- **مشکلات جداسازی:** جداسازی کامل مدل، نما و کنترل‌کننده همیشه آسان نیست.

مثال:

- فرض کنید در حال ساخت یک برنامه وب برای مدیریت لیست وظایف هستید.
- **مدل:** شامل کلاس‌هایی برای نشان دادن وظایف، لیست‌ها و کاربران خواهد بود.
- **نما:** شامل صفحات HTML، CSS و JavaScript برای نمایش لیست وظایف، فرم‌های اضافه کردن وظایف جدید و غیره خواهد بود.
- **کنترل‌کننده:** شامل کدهایی برای رسیدگی به درخواست‌های کاربر، مانند افزودن، ویرایش یا حذف وظایف خواهد بود.

فریم‌ورک‌های MVC:

- بسیاری از فریم‌ورک‌های محبوب برای توسعه برنامه‌های وب با استفاده از الگوی MVC وجود دارد، از جمله:

• Ruby on Rails (Ruby)

• Django (Python)

• ASP.NET MVC (C#)

• Spring MVC (Java)

نتیجه‌گیری:

- MVC یک الگوی معماری قدرتمند و انعطاف‌پذیر است که می‌تواند به شما در ساخت برنامه‌های کاربردی تمیز، قابل نگهداری و قابل تست کمک کند.