

به نام خدا

موضوع : مدل MVC

دانشجو : عرفان معمارپور

استاد : میثاق یاریان

درس : طراحی صفحات وب پیشرفته

MVC (Model View Controller) یک معماری برنامه نویسی است که نرم افزار را به سه بخش مجزا تقسیم می کند. این بخش ها عبارت اند از Model برای ارتباط با پایگاه داده، View برای ایجاد یک رابط کاربری برای نمایش و تغییر در داده های پایگاه داده و Controller برای پردازش درخواست های کاربر. هدف اصلی استفاده از الگوهای طراحی مثل MVC این است که عملکرد، منطق، رابط و کلا همه ی اجزای یک برنامه را طبق استاندارد های از پیش تعریف شده از هم جدا کنیم، تا چند نفر بتوانند همزمان روی یک پروژه بدون تداخل و سردرگمی کار بکنند.

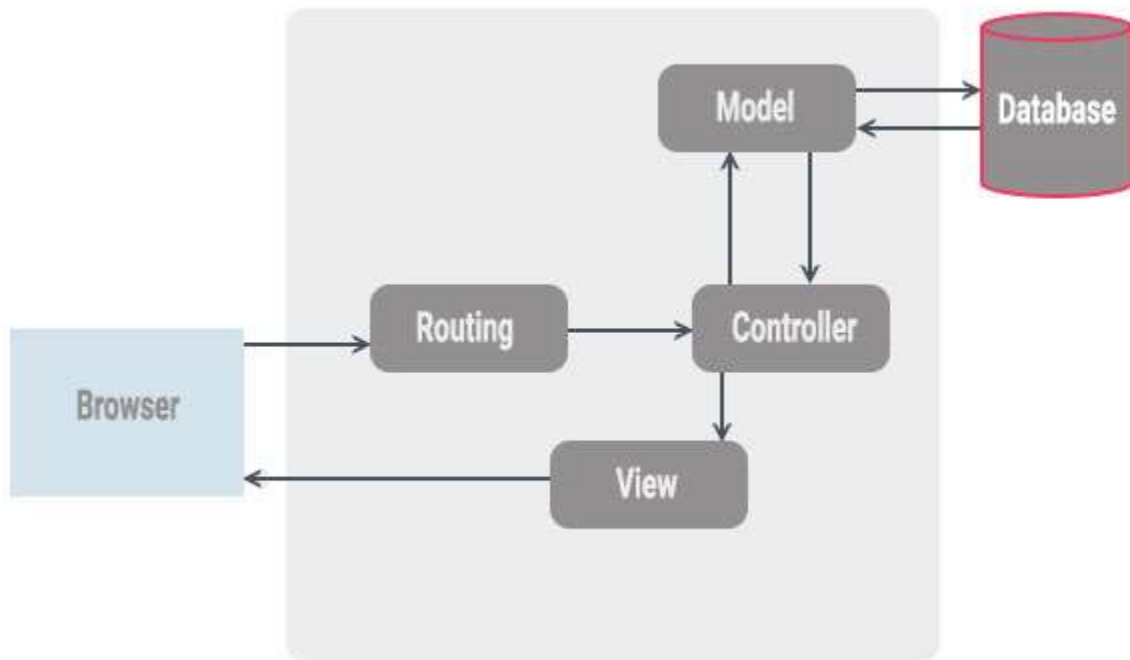
این معماری توسط Trygve Reenskaug اختراع شد. Trygve در ابتدا اسم این الگو را Model View Thing گذاشت اما بلافاصله آن را به مدل mvc تغییر داد. Trygve تلاش می کرد که مشکل مدل سازی سیستم های پیچیده را مانند: طراحی ساخت پل، نیروگاه تولید نفت و... حل کند.

۱- (مدل): قسمتی از برنامه کاربردی است که مسئول بازیابی داده از بانک اطلاعاتی، ذخیره آن، تبدیل آن به شی یا آبجکت ها و پیاده سازی منطق برنامه برای داده های دامنه ی مسئله است. در حقیقت بار اصلی معماری MVC بر عهده ی این بخش است. مثلاً یک آبجکت Product ممکن است اطلاعات را از بانک اطلاعاتی بازیابی کرده، بر روی آنها عملیاتی را انجام دهد و سرانجام نتیجه را در بانک اطلاعاتی یا در جدول Products ذخیره کند.

۲- (نمایشگر): اجزایی از برنامه است که واسط کاربری برنامه (UI) را می سازد. معمولاً این UI از داده های مدل ساخته می شود. در واقع نقطه پایان برنامه کاربردی است. به کاربر نتایج عملیات و بازیابی و نمایش داده از طریق برقراری ارتباط با دو بخش دیگر؛ یعنی مدل و کنترلگر را نشان می دهد. برای مثال، هنگامی که کاربر در فرم ورود به سیستم رمز عبور خود را وارد می کند، اکثر برنامه نویسان در همان فرم اقدام به چک کردن رمز عبور می کنند که این عمل مغایر با قوانین MVC است. در MVC هنگامی که کاربر رمز عبور را کرد، رمز عبور بدون هیچگونه اعمالی به بخش های دیگر فرستاده می شود و فقط یک نتیجه ساده یا خبر از بخش های دیگر دریافت می کند که از طریق آن اجازه ی ورود به برنامه داده می شود.

۳- (کنترلگر): اجزایی از برنامه هستند که مدیریت تعامل با کاربر را بر عهده دارند. می توان گفت که واسط بین مدل و نمایشگر می باشند؛ یعنی با مدل کار می کند و در انتها نمایشگری را برای نشان دادن واسط کاربری انتخاب می کند. ورودی کاربر را مدیریت کرده و به آنها پاسخ می دهد و با کاربر تعامل می کند. برای مثال، کنترلگر عبارت های پرس و جوی بانک اطلاعاتی را مدیریت کرده و آن ها را به مدل ارسال می

کند، وظیفه اجرای پرس و جوها با مدل است.



به عنوان مثال فرض کنید کاربر درخواست جست و جوی یک محصولی در سایت فروشگاهی را دارد، کاربر درخواست خود را به سرور میفرستد و سرور توسط مسیر یاب درخواست را به کنترلر مربوطه ارسال میکند، پس از بررسی و پردازش اطلاعات محصول، دیتا توسط مدل آورده میشود و ویوی متناسب با آن تولید میگردد و پاسخ مناسب به سمت کاربر ارسال میشود.

برخی از مزایای مدل MVC به صورت زیر میباشد:

استفاده نکردن از فرم های سروری.

پشتیبانی از زیرساخت های مسیریابی.

مدیریت آسان تر با مدل MVC.

پشتیبانی از طراحی و توسعه ابزار.

معایب مدل MVC به شرح زیر است:

پیچیدگی زیاد برای افراد مبتدی.

ناکارآمد بودن برای پروژه های کوچک.

ناکارآمد بودن در دسترسی داده ها.

استفاده از معماری MVC روند توسعه نرم افزار را بسیار اصولی خواهد کرد. زیرا لایه های مختلف برنامه را از یکدیگر جدا کرده است. و در آینده برای انجام تغییرات جزئی و کلی و یا حتی اضافه کردن یک ویژگی جدید به برنامه کار بسیار آسان تر و سریع تر از حالتی است که از معماری MVC در آن برنامه استفاده نشده باشد.