

خوب من برای اینکه پیام و به این هدف برسم اومدم و از به معماری معروف برای درست کردن سایت استفاده کردم. اون معماری mvc هست .

خوب mvc چیه ؟

خلاصه:

يك Design Pattern است.

Design Pattern چیست و چرا از Design Pattern استفاده می کنیم؟

Design Pattern ها الگو ها و روش های اثبات شده ای هستند که هنگام طراحی معماری Application باید مورد توجه قرار گیرند . Design Pattern ها نشانگر تجربه و دانش طراحی هستند که از آن ها با موفقیت استفاده کرده اند. استفاده از این Design Pattern ها به شما امکان استفاده از تجربیات ایشان را در جهت مقاصدتان می دهد. ضمناً استفاده از آنها ، زبان و قالب مشترکی را بین طراحان نرم افزار فراهم می سازد. Design Pattern های مختلفی وجود دارد که یکی از مهمترین آنها MVC است.

MVC چیست؟

کاربران ممکن است از طریق Browser با فرمت HTML یا Wireless Device با فرمت WML یا Web Service با فرمت XML و غیره به سرور، درخواست بفرستند. معماری MVC برای پاسخگویی به این نیاز ، یعنی فراهم آوردن امکان سرویس دهی به انواع کاربران با interface های گوناگون ، پدید آمده است. MVC یکی از معماری های مهم است و به Model-View-Controller architecture اشاره دارد. این معماری عملیات نمایش و مدیریت داده ها را به سه قسمت تقسیم می نماید.

Model

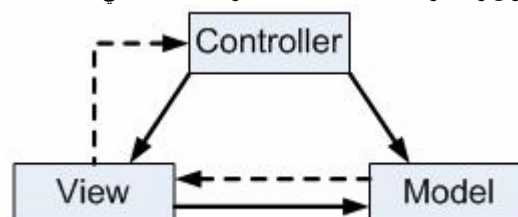
وظیفه ارتباط با منابع اطلاعاتی (Database) ها و غیره) را به عهده دارد. همچنین Business Logic در این قسمت پیاده سازی می شود.

View

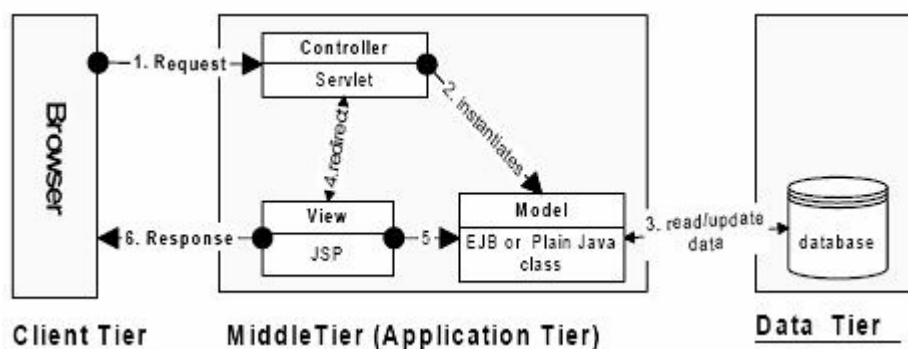
خروجی را با فرمت مناسب تولید می کند.

Controller

بعنوان واسط و هماهنگ کننده Model و View عمل می کند.



نحوه ارتباط بین این اجزا را از هنگام دریافت يك request تا صدور response در شکل زیر ملاحظه می نمایید.



و به طور کامل:

مقدمه

نیاز روز افزون به کامپیوتر و مکانیزه کردن و سپردن تقریبی تمامی امور به دست ماشین امری است که انکار ناشدنی است. در این بین تولید کنندگان نرم افزار نیز تلاش میکنند تا نرم افزاری تولید کنند تا بتواند اکثر نیاز های متقاضیان را به بهترین نحو ممکن تامین کند و در همین راستا در تلاش هستند که روند تولید نرم افزار را به سمتی بکشانند که ساختار استاندارد و تائید شده ای داشته باشد.

شاید اینطور بشه گفت که دوران کد نویسی به پایان رسیده و همه چیز به سمت زیر ساخت ها و بنیان نهادن چارچوب های استاندارد و پیروی از آن ها در امر تولید بهتر نرم افزار در حرکت است. اجازه دهید ببینیم خصوصیات یک نرم افزار خوب چیست ؟

نامبردن تمامی خصوصیات یک نرم افزار خوب در این مقال نمی گنجد اما تعداد محدود و مهمی از آنها عبارتند از:

قابل حمل بودن

قابل استفاده مجدد بودن

قابل تغییر بودن

بهینه بودن از لحاظ حافظه و زمان (زمان مهمتر از حافظه)

و.....

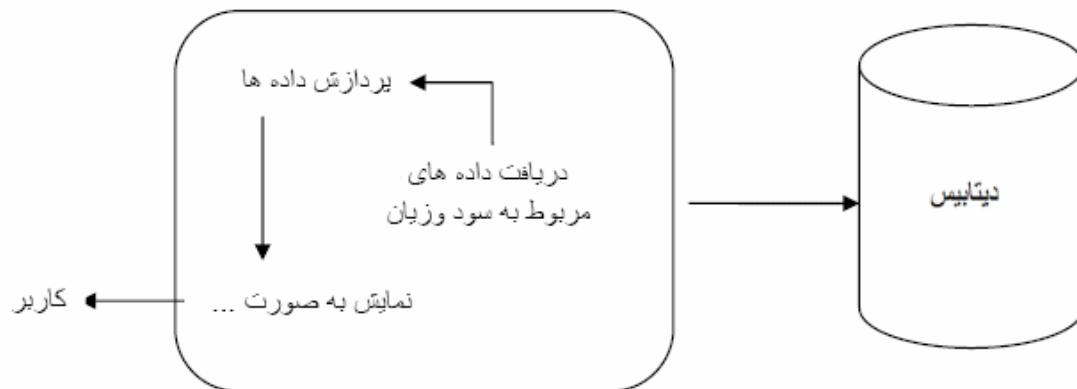
مسئله ؟

بهتر است وجود مسئله را با یک مثال نشان دهیم

فرض کنید نرم افزاری برای شرکتی نوشتید که به بخش آن مقدار سود و زیان شرکت را در سال های مختلف بر اساس ارقام بیان میکند. حال صاحب برنامه پس از مدتی از شما می خواهد برنامه را طوری تغییر دهید که همین اطلاعات را به گونه های مختلف دیگه ای مثلا نمودار های مختلف (میله ای ، دایره ای و ...) در اختیار داشته باشد و یا حتی بخواهد آنها را به فرمت خاصی و در فایل های خاصی ذخیره کند. در این مواقع چطور مشکل را حل میکنید؟

راه حل:

همانطور که گفته شد یکی از خصوصیات نرم افزار خوب قابل تغییر بودن آن میباشد. فرض کنید که برنامه را به این شکل طراحی کردید:



همانطور که در شکل نیز نشان داده شده است تمامی اعمال اعم از دریافت داده ها که مهمترین بخش است و همچنین پردازش آن ها همگی در یک فرم طراحی و پیاده سازی شده اند. و دقیقاً مشکل همین جا نمایان میشود. ارتباط مستقیم با منبع داده بر قرار کردن جدا از اینکه مشکلات امنیتی دارد که بحث در مورد آن خارج از این مقال است، باعث میشود که دست برنامه نویس را برای تغییرات آتی در برنامه ببندد. چون داده ها درون خود فرم از منبع داده و به صورت مستقیم خوانده میشود پس دسترسی به داده های خوانده شده وجود ندارد. یا حداقل متحمل سربار زیادی می باشد.

راه کار های نوین:

یکی از راه کار هایی که امروزه بیشتر شاهد استفاده آن هستیم، تولید نرم افزار بر اساس ساختار های لایه ای می باشد، بدین صورت که کل نرم افزار به تعداد لایه هایی تقسیم میشوند، هر لایه وظیفه خاص خود را دارد و لایه ها از نتایج لایه های دیگر استفاده میکنند.

تعداد این لایه ها بسته به نرم افزار و طراحی میتواند ۲، ۳، ۴ یا ۵ لایه یا حتی بیشتر باشد. اما استاندارد آن که بیشتر از بقیه هم استفاده میشود ۳ لایه هست و به روشی که بر اساس این تئوری پیاده سازی میشود اصطلاحاً ۳ Tire Programming گفته میشود.

در تئوری ۳ لایه، لایه ها عبارتند از

Data Access layer
Business Logic Layer
Presentation Layer

توضیحات بیشتر و نحوه پیاده سازی روش فوق را به عهده خود خواننده میگذارم.

پس از مقدمه ای مختصر وارد بحث اصلی یعنی (**MVC (Model View Controller** می شویم.
تذکر: قبل از پرداختن به مثال عملی ابتدا توضیحاتی در مورد معماری MVC می دهم برای آن دسته از عزیزانی که با این معماری آشنا نیستند، طبیعی است که افرادی که مباحث تئوری این معماری را میدانند مستقیماً میتوانند به بخش بعدی که به زودی آماده می شود، بروند.

MVC چیست ؟

MVC مخفف سه کلمه **Model View Controller** هست. در واقع MVC بر روی معماری های چند لایه ای جهت جداسازی قسمت های مختلف برنامه و به طور دقیق تر جدا کردن بخش ها منطقی برنامه اعم از دیتا، permission ها، چک کردن صحت داده ها و از لایه Presentation layer یا در واقع همان لایه ای که مستقیماً با کاربر نهایی (End user) در ارتباط است، قرار میگیرد.

پس بر اساس توضیحات فوق میتوانیم هر یک از بخش های معماری MVC یعنی Model و View و controller را به شکل زیر تعریف کنیم.

Model

در واقع بار اصلی معماری MVC بر عهده این بخش است. این بخش میتواند با داده ها در ارتباط باشد. الزاماً منظور از داده حتماً ارتباط با پایگاه های داده همچون MSSQL و Access و ... نیست، حتی منبع داده ها در بخش Model میتواند یک آرایه از اعداد و یا هر چیز دیگری باشد. همچنین Model وظیفه چک کردن داده ها جهت صحت درستی داده ها را هم در بر عهده دارد (در این زمینه همکاری بیشتری با بخش Controller دارد) و همینطور وظایف دیگری که در مثال های عملی که در آینده خواهیم زد بیشتر آشنا خواهید شد.

View

این بخش که در واقع همان بخش Presentation Layer در معماری ۳ لایه میباشد وظیفه بر قرار ی ارتباط با کاربر نهایی و گرفتن داده از کاربر

Controller

این بخش همانطور که از اسم آن مشخص است به بخش کنترل کننده می باشد ، و در واقع واسطی بین دو بخش Model و View میباشد.

حال ببینیم روند اجرای برنامه در معماری MVC به چه نحوی خواهد بود.
در معماری MVC روند کلی برنامه (جزئیات را در ادامه خواهید دید) به این شکل است که کاربر تقاضای خود را از طریق واسط های برنامه نویسی (نظیر Form ها و User Control ها و ..) از برنامه (از بخش View) درخواست می کند . بخش View درخواست ها را به بخش Controller فرستاده و این بخش با برقراری ارتباط با بخش Model درخواست های کاربر را پردازش کرده و پس از پایان پردازش زمانی که خروجی درخواست داده شده آماده گردید بخش Controller بخش View را آگاه میسازد تا خود را بر اساس تغییرات جدید که اصطلاحاً در معماری MVC به آن حال Model می گویند ، به روز سازد . در واقع چیزی که باعث میشود تا بخش Controller به بخش View اطلاع دهد که باید حالت جدید model را دریافت کند و خود را Update کند این است که بخش View باید قبلاً خودش را در بخش Model اصطلاحاً Register کرده باشد که البته عمل Register کردن توسط بخش Controller انجام میگیرد . نحوه register کردن بخش View به معماری آن محیط و همچنین زبانی که توسط آن برنامه را گسترش میدهید و همچنین قابلیت های آن زبان بستگی دارد.

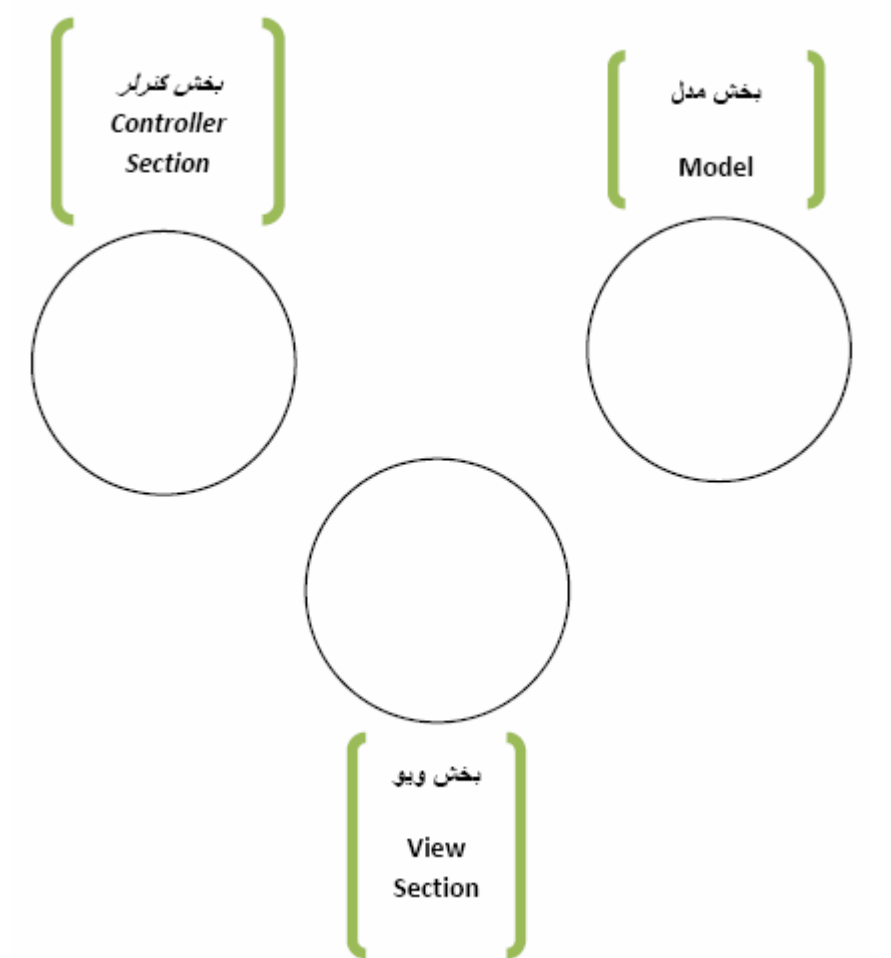
زبان C# در این زمینه قابلیت های زیادی دارد که میتوان معماری MVC را بر اساس آنها پیاده سازی کرد . به عنوان نمونه برای برای Register کردن بخش View میتوان از یک متد ساده مثلاً به نام RegisterMe که یک View را که معمولاً فرم میباشد را register میکند یا از Interface ها که انعطاف پذیری بالایی دارند و باعث قدرتمند تر کردن برنامه نویس جهت پیاده سازی اهداف میشوند و استفاده کرد.

همچنین از برای آگاه سازی بخش View جهت به روز شدن از روش های مختلفی از یک متد ساده تا استفاده کردن از Delegate ها و Event ها استفاده کرد.

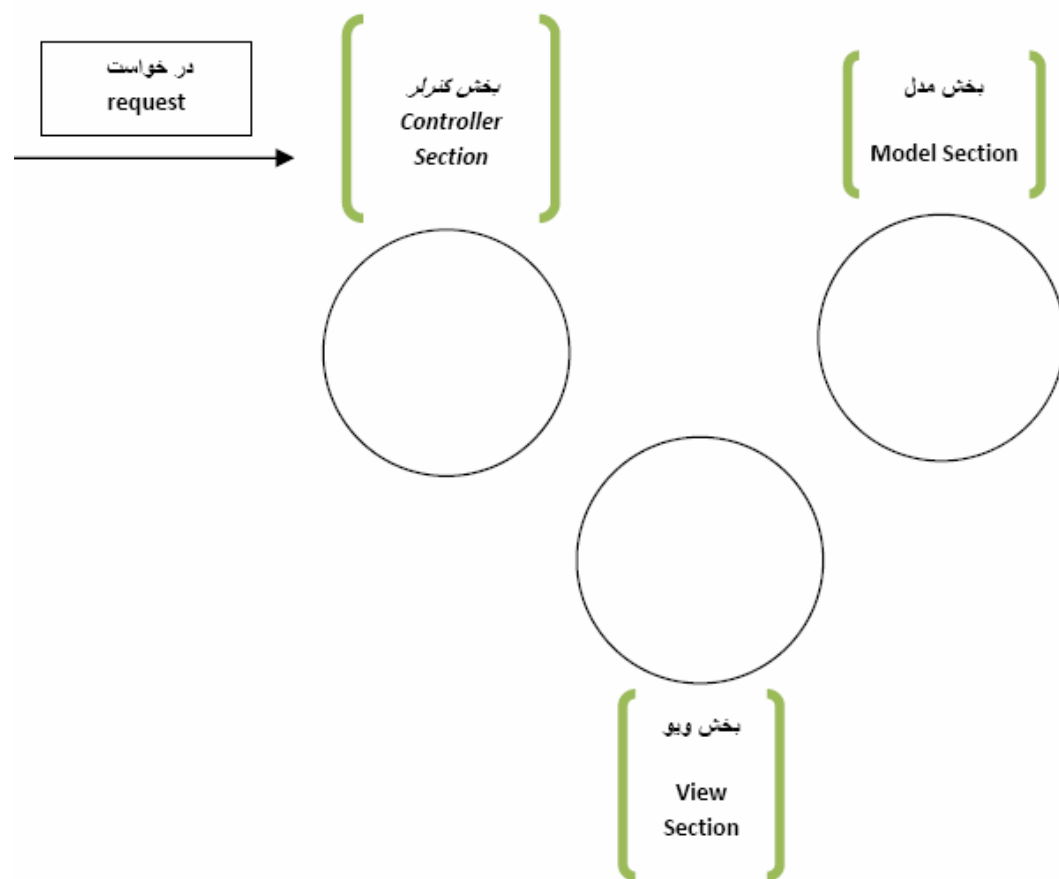
قصد دارم که مثال های متنوعی در این باره بزنم که سعی خواهم کرد ابتدا از روش های ساده شروع کنم و کم کم به سمت روش های حریجه ای تر حرکت کنم که همگی بتوانند از این معماری در برنامه های خودشان استفاده کنند.

حال ببینیم توضیحاتی که در مورد روند کلی یک برنامه که بر اساس معماری MVC نوشته میشود در قالب یک نمودار به چه شکلی خواهد بود.
نمودار را قدم به قدم رسم میکنم و ابتدا از ساده ترین نمودار شروع خواهم کرد.

گفتیم که معماری MVC بر اساس سه بخش یا سه لایه Model و View و Controller بنا میشود . و همه چیز تحت نظر این سه لایه باید انجام شود.

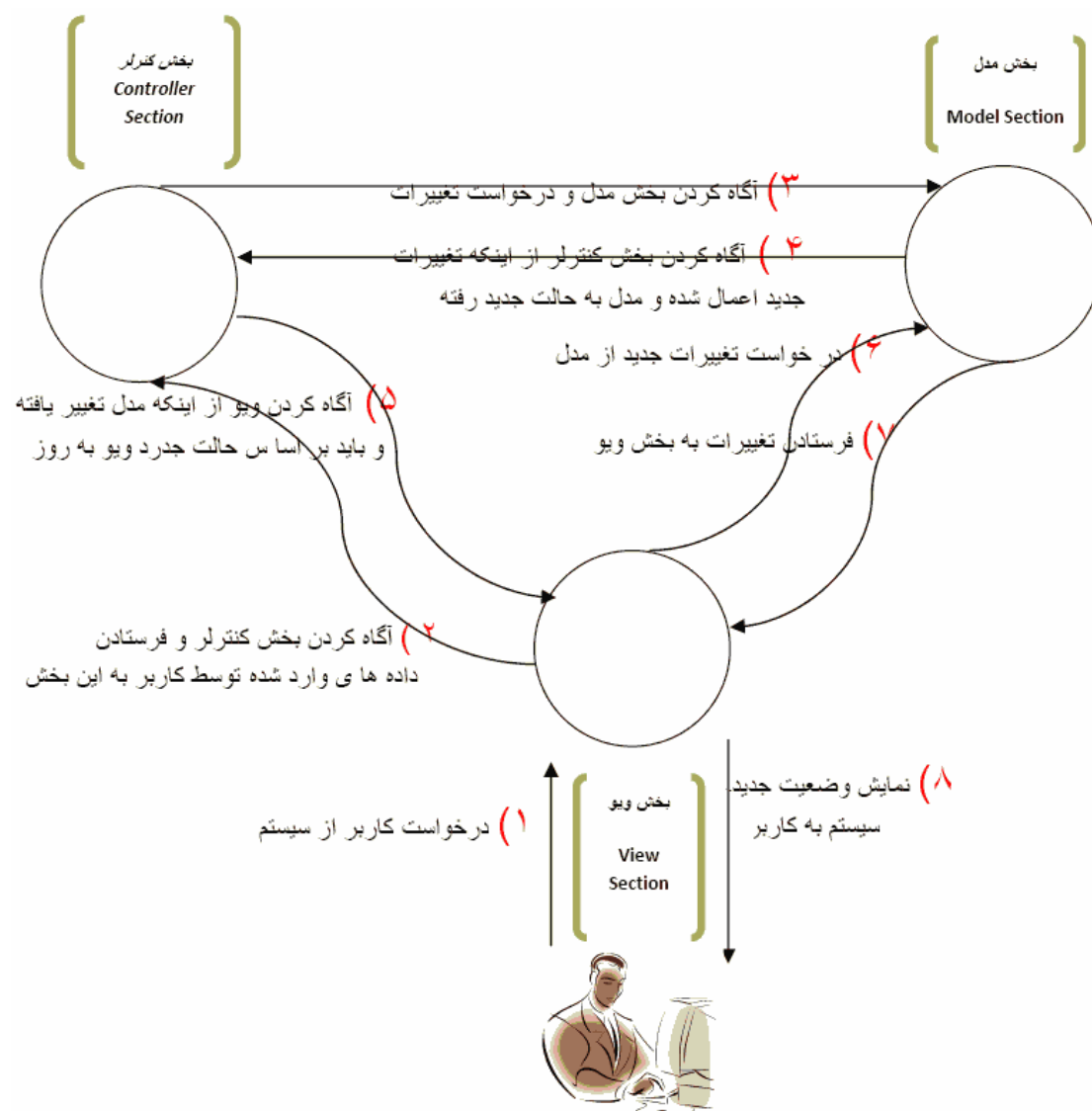


حال اقدام به برقراری ارتباط بین این سه بخش (لایه) بر اساس توضیحاتی که دادم میکنم . خیلی از مقاله ها شروع معماری را از بخش کنترلر میدانند در واقع درست هم میگویند چون در نهایت بخش کنترلر است وظیفه handle کردن ابزار های مختلف مثل موس و .. را دارد. پس میتوان نمودار فوق را به شکل زیر کامل کرد:



اما من در نمودار فوق و بر اسا توضیحاتی که دادم فرض میکنم که کاربر درخواست هایش را از طریق فرم ها وارد کرده و منتظر نتیجه اطلاعات میباشد.

مراحل را به ترتیب عملیت و با شماره روی شکل نشان دادم.



توضیحات شکل بر اساس روند جریان برنامه:

مرحله ۱ (در این مرحله درخواست ها از سمت کاربر نهایی به ویو فرستاده میشود . پس جهت جریان اطلاعات در این مرحله از سمت End user به سمت View می باشد).

مرحله ۲ (در این مرحله کاربر درخواست خود را از طریق بخش View اعمال میکند . همانطور که گفته شد ویو مسئولیتی در قبال داده های وارد شده معمولاً ندارد ، پس درخواست کاربر را به بخش کنترلر می فرستد . همچنین در همین مرحله ویو خودش را register می نماید تا بتواند اطلاعات جدید را دریافت نماید . جهت حرکت در قسمت ۱ از سمت View به سمت Controller میباشد).

مرحله ۳ (پس از اینکه درخواست کاربر از View به Controller رسید و Controller بخش View را رجیستر کرد ، درخواست را به مدل داده و از مدل میخواهد که بر اساس درخواست کاربر به حالت جدید رفته و خود را به روز نماید . جهت جریان اطلاعات در این مرحله از سمت Controller به سمت Model میباشد).

مرحله ۴ (پس از اینکه مدل به درخواست ها رسیدگی کرد ، و به حالت جدید رفت این رفتار را به کنترلر پیغام میدهد . (همچنین اگر خطایی در این بین اتفاق بیفتد نیز کنترلر مطلع و به طبع آن در مرحله بعدی ویو نیز مطلع میگردد). جهت جریان اطلاعات در این مرحله از سمت Model به سمت Controller میباشد).

مرحله ۵ (در این مرحله ، کنترلر موظف است که پیغامی به ویو بفرستد مبنی بر اینکه مدل تغییر کرده و به حالت جدید رفته است . این تغییر حالت الزاماً دریافت داده های جدید از دیتابیس و ... نیست بلکه هر نوع تغییری میتواند باشد ، مثلاً یک Exception رخ داده است و ... جهت جریان اطلاعات در این مرحله از سمت Controller به سمت View می باشد).

منابع:

<http://javaborder.radmanitd.com>

barnamenevis.org

نویسنده بخش دوم : مهدی کیانی

همچنین برای دیدن مکان اصلی این نوشته و بخش دوم آن به ادرس زیر مراجعه کنید:

<http://barnamenevis.org/forum/showthread.php?t=80731>

خوب حالا فهمیدیم mvc چیه ولی ایا اونقدر برنامه نویس شدیم که خودمون بیاییم و طبق mvc یک اسکلت درست کنیم روی اون برنامه بنویسیم؟

این جواب برای من نه هست و هنوز اونقدر مرد نشدم! ولی میتونم تو دنیای زیبای این سورس که همه چیزای با ارزشونو به اشتراک میزارن از زحمت دیگران استفاده کنم!

خوب با یکم جستجو به آموزش ساخت یک فریم ورک ساده بر طبق الگوی mvc دست پیدا کردم که اصل این آموزش در سایت phppro.org به ادرس زیر هست :

<http://www.phppro.org/tutorials/Model-View-Controller-MVC.html>

که خوشبختانه توسط یکی از دوستان عضو سایت برنامه نویس فارسی هم شده و میتونید از ادرس زیر مطالعه بفرمایید:

<http://www.barnamenevis.org/forum/showthread.php?t=۱۱۲۱۸۶>

ولی خوب اونو فقط برای کسانی گذاشتم که علاقه دارن بدونن چه طوری میشه اسکلتو ساخت ، تو این آموزش ما با داتلود این پروژه یاد میگیریم چه طوری کارامونو تو اسکلت اون انجام بدیم ، پس از ادرس زیر فایل زیرو که نتیجه آموزش بالاست داتلود کنید:

<http://www.phppro.org/downloads/mvc-۰.۰.۴.tar.gz>



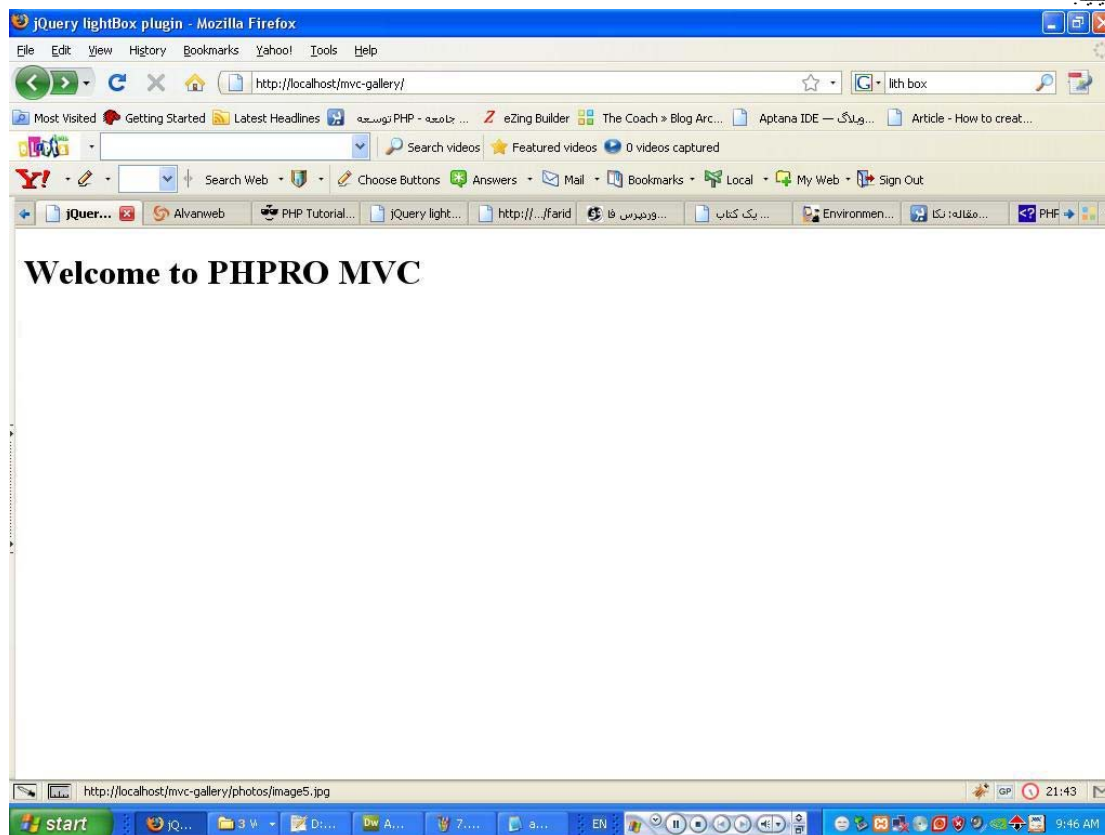
و حالا شما آماده اید تا کارو شروع کنید پس این فایل رو در پوشه htdocs با wwwroot شبیه ساز php از حالت فشرده در بیارین.

من از xampp استفاده میکنم چون نسبت به wamp تنظیماتش به هاست های اینترنتی شبیه تره و ادم موقع ایلود به مشکلات کمتری بر میخورم!

تازه به حالت قابل حمل هم داره که میتونید بدون نصب اونو اجرا کنید و من اونو تو فلشم زدم و هر جا میرم برای نشون دادن نمونه کارام دیگه لازم نیست شبیه سازو نصب کنم رو کامپیوترشون و بعد از توضیح ، روم نشه بخوام کارامو از کامپیوتر طرف پاک کنم . ((قابل توجه به لا قبا هایی مثل خودم که لپ تاب ندارن))

بعدش من اسم پوشه اونو به mvc-gallery تغییر دادم چون میخوام به عنوان اولین کار عملی به گالری پویا تحت این ساختار بسازم!

خوب حالا برای امتحان اینکه ببینیم همه چی درسته کافیه تو مرورگرمون ادرس سایتمونو بزنینم. باید صفحه ی زیرو ببینید:



و اما قبل از اینکه شروع به کار کنیم باید بدونیم چه گالری میخوایم داشته باشیم و با چه ساختاری!

میدونید ز موه عوض شده و ملت حرف از وب ۲ و حتی وب ۳ میزنن! خوب حالا اینا چین؟

منبع: <http://farhadi.ir/blog>

حتما تا به حال کلمه «وب ۲» زیاد به گوشتون خورده. همین وبی که الان توش زندگی میکنیم. قبل از اومدن وب ۲، وب یا وب ۱ به تدریج فرایند رشد و تکامل خودش رو طی کرد تا اینکه تغییر و تحولات به حدی رسید که دیگه وب نسبت به روزهای اول، کاربرد و معنایی متفاوت پیدا کرده بود کم کم واژه وب ۲ با تعاریف مختلفی ظهور پیدا کرد. طبیعی بود، چون وب چیزی نبود که یکدفعه از ورژن یک به دو ارتقا پیدا کرده باشه، تا بشه تعریف دقیقی ازش ارائه کرد بلکه مجموعه ای از تغییرات تدریجی موجب پیدایش وب ۲ شده بود. با گذشت زمان و گسترش و تثبیت وب ۲، واژه وب ۲ نیز از لحاظ مفهوم به یک ثبات نسبی دست پیدا کرد.

این روزها همانند اوایل پیدایش وب ۲ تعاریف مختلفی از وب ۳ ارائه میشه. البته با توجه به تغییرات وب تا به امروز و برنامه های آتی شرکتهای تاثیر گذار در وب، میشه تعریفی در حد امکان نزدیک به واقعیت از آینده وب یعنی وب ۳ ارائه کرد.

سیر تکاملی وب رو میشه از دیدگاه های مختلف مورد بررسی قرار داد.

دیدگاه اول: تولید محتوا

وب ۱: متشکل از سایتهایی است که محتوای آن توسط افرادی خاص و یا از منابعی خاص ارائه میشود. و کاربران وب فقط مصرف کننده هستند.
وب ۲: متشکل از سایتهایی است که محتوای آن توسط کاربران آن تولید میشود. و هر کاربر میتواند هم مصرف کننده و هم تولید کننده محتوا باشد.

دیدگاه دوم: ماهیت محتوا

وب ۱: سایتها حاوی اطلاعاتی هستند که فقط توسط انسان قابل خواندن است.
 وب ۲: سایتها حاوی اطلاعاتی هستند که محتوا از ظاهر جدا شده و ساختار محتوا نیز مشخص شده است. (ساختار محتوا با div و چیدمان با css)
 وب ۳: ماهیت محتوا از «اطلاعات» قابل خواندن توسط انسان به «دیتا» در قالب xml تغییر می یابد و امکان استفاده سایر application ها از داده ها و نمایش با استفاده از xslt فراهم میشود. به عبارت دیگر وب ۳ یک پایگاه داده عظیم خواهد بود. به همین دلیل آنرا «Data Web» نیز می نامند (rss , opml و...) .

دیدگاه سوم: از لحاظ ارتباطی

وب ۱: متشکل از سایتهایی است با ماهیت نسبتاً مستقل نسبت به یکدیگر.
 وب ۲: متشکل از سایتهایی که به وسیله شبکه های اشتراکی به صورت یکپارچه به یکدیگر پیوند خورده اند (del.icio.us) ، digg ، technorati ...
 وب ۳: متشکل از سایتهایی که از طریق داده های xml و webservice ها با سایتهای دیگر در ارتباط هستند و محتوا و کارایی سایر سایت ها را در خود جای داده اند (facebook) و google reader و...)

دیدگاه چهارم: رابط کاربری

وب ۱: تعامل کاربر با سایت بسیار محدود است. تمام پردازش ها سمت سرور انجام میگردد. فقط اطلاعات به مرورگر ارسال میشود.
 وب ۲: به کمک جاوااسکریپت و dhtml و ajax بخشی از پردازش اطلاعات در سمت کاربر صورت میگیرد، مرورگر با کاربر تعامل دارد و به event های کاربر پاسخ میدهد. در واقع سایت های وب تبدیل به application های تحت وب میشوند.
 وب ۳: برای استفاده از وب دیگر نیازی به مرورگر نیست. وب فقط شامل داده ها و webservice هاست و در واقع application ها فقط thin client هایی هستند بر روی کامپیوتر کاربران که با وب در تعامل هستند. از آنها میتوان حتی در حالت آفلاین نیز استفاده کرد (Adobe AIR) ، (Google Gears)

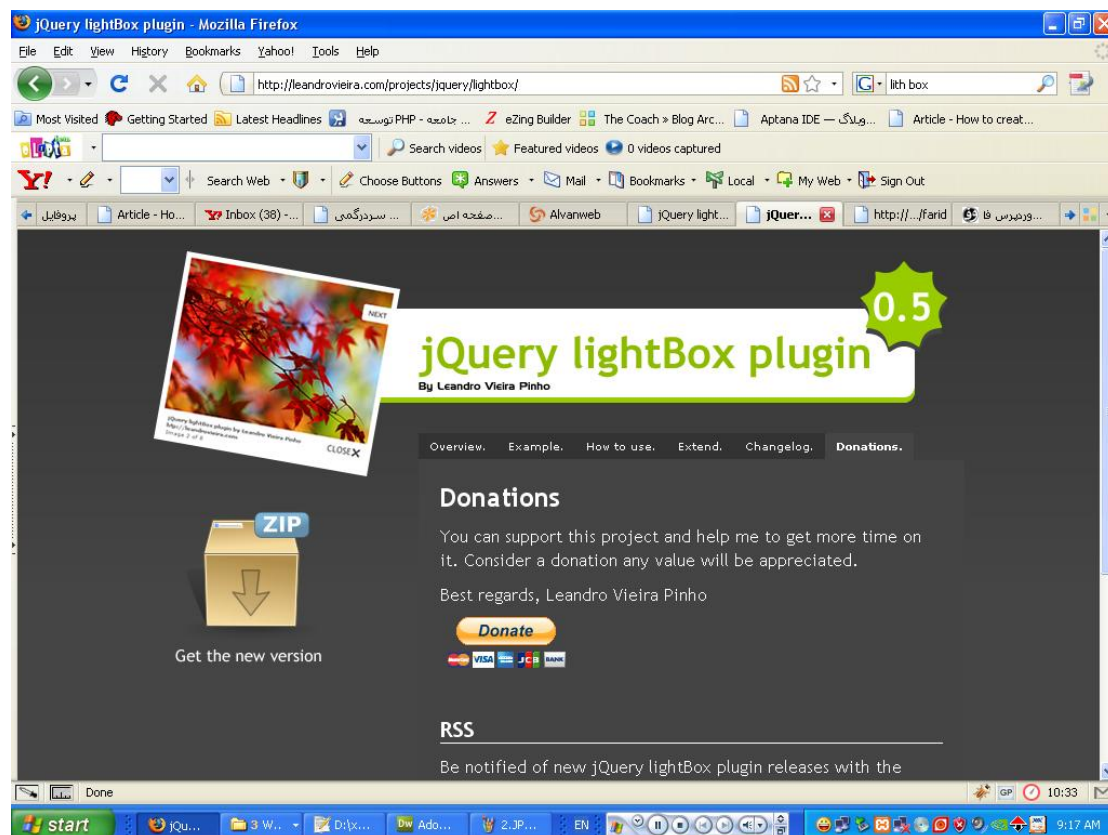
خوب حالا گوشی دستمون اومد که از همون اول باید مشخص کنیم میخوایم بر اساس استاندارد های کدوم نسل پروژمونو درس کنیم؟

میدونید وب دو خیلی خوبه ، خیییییلی کلاس داره ولی خوب مثلاً ما که به اونصورت چیزایی مثل اژاکس بلد نیستیم که

پس چه کنیم ؟ کلید حل مشکلات ما ، این سورسه

با کمی جستجو به jquery-lightbox رسیدیم که اونو میتونید از ادرس زیر دانلود کنید:

<http://leandrovieira.com/projects/jquery/lightbox/>



خوب حالا این چی هست ؟

اول اونو از حالت فشرده خارج کنید و بعد امتحانش کنید تا لذت ببرید که این گالری چقدر زیباست و اساس کار اون بر روی کتابخانه یا فریم ورک JQuery بنا شده که یک فریم ورک برای زبان جاوا اسکریپت و بسیار هم طرفدار پیدا کرده و میتونید با چند خط بجای چند خط جاوا اسکریپت به مراد دلتون برسین.

تو اینجا لازم نیس ما وارد این تکنولوژی بشیم و با استفاده از همین نمونه که دائلود کردیم کارمونو راه خواهیم انداخت.

خوب حالا همه چی این فایلو میبریم میریزیم تو پوشه mvc-gallery و کار واقیو شروع میکنیم :

اول باید بفهمیم چه طوری تو این سیستم mvc یک صفحه ی جدید بسازیم ؟ پس بیایید یه آزمایش کوچیک کنیم ، ما میخوایم یه صفحه ی درباره ی مای ساده ، تو این سیستم بسازیم که استاتیکه و هیچ ارتباطی با دیتابیس نداره و حتی کار خاصی هم نمیکنه

پس فقط کافیه تو پوشه views یه فایل به نام about.php بیسازیم ولی اگر بخوایم با دادن این ادرس

<http://localhost/mvc-gallery/about.php>

بیش دسترسی پیدا کنیم با پیغام خطا رو برو میشیم! چرا ؟ چون ما داریم طبق ساختار mvc کد میزنیم و دیگه برای اینطور صفحاتی که با دیتابیس کاری ندارن باید دو بخش **views** و **Controller** بسازیم پس در پوشه **Controller** ما باید فایل **index.php** رو طوری تغییر بدیم تا فایل **about.php** در پوشه **views** رو بشناسه!

خوب فایل **index.php** رو باز میکنیم تا ببینیم چیه و چه طوری باید تغییرش بدیم!

<?php

```

Class indexController Extends baseController {

public function index() {

    /*** set a template variable ***/

    $this->registry->template->welcome = 'Welcome to PHPRO MVC';

    /*** load the index template ***/

    $this->registry->template->show('index');

}

}

?>

```

خوب همونطور که میبینید در این فایل برای فایل index که در پوشه **views** قرار داره به فانکشن تعریف کرده پس ما هم باید قاعدتا به فانکشن به نام **about** بسازیم ولی توش چی باید بزاریم؟

اگه دقت کنید میبینید تو این فاکشن به به متغیر توسط روشهای شی گرایي مقدار 'Welcome to PHPRO MVC' داده شده و این متغیر همون متغیری هست که در فایل index.php در پوشه **views** تعریف شده پس میشه نتیجه گرفت که اون متغیر که در فایلمون در پوشه **views** هست بنوعی میانبری هست برای این متغیر!

اگه با برنامه نویسی شیء گرا تو php اصلا آشنا نیستید و میخواهید سریعتر به خورده آشنا بشین باهاش تا حداقل تو این جور جاها اگه چشتون به کدهاش خورد ، نترسین توصیه من:

<http://www.barnamenevis.org/forum/showthread.php?t=۸۳۱۰۳>

خوب پس من میام فایل index.php پوشه **Controller** رو اینطوری تغییر میدم:

```
<?php
```

```
Class indexController Extends baseController {
```

```

public function index() {

    /*** set a template variable ***/

    $this->registry->template->welcome = 'Welcome to PHPRO MVC';

    /*** load the index template ***/

    $this->registry->template->show('index');

}

public function about() {

    /*** set a template variable ***/

    $this->registry->template->about_us = 'hi ! my name is farid :-)';

    /*** load the index template ***/

    $this->registry->template->show('about');

}

}

?>

```

خوب الان باید برم تو فایل about.php پوشه **views** و متغیری رو که در کنترلر گفتیم نمایش بده کد بزنم تا echo کنه!

```
<h1><?php echo $about_us; ?></h1>
```

خوب پس فهمیدیم چه طوری مقادیر و باید از کنترلر یک صفحه به ویو اون صفحه پاس بدیم و در آخر پاس دادن کافیه در کنترلر بگیم ویو اون صفحه رو نشون بده:

```
$this->registry->template->show('about');
```

نکته خیلی مهم :

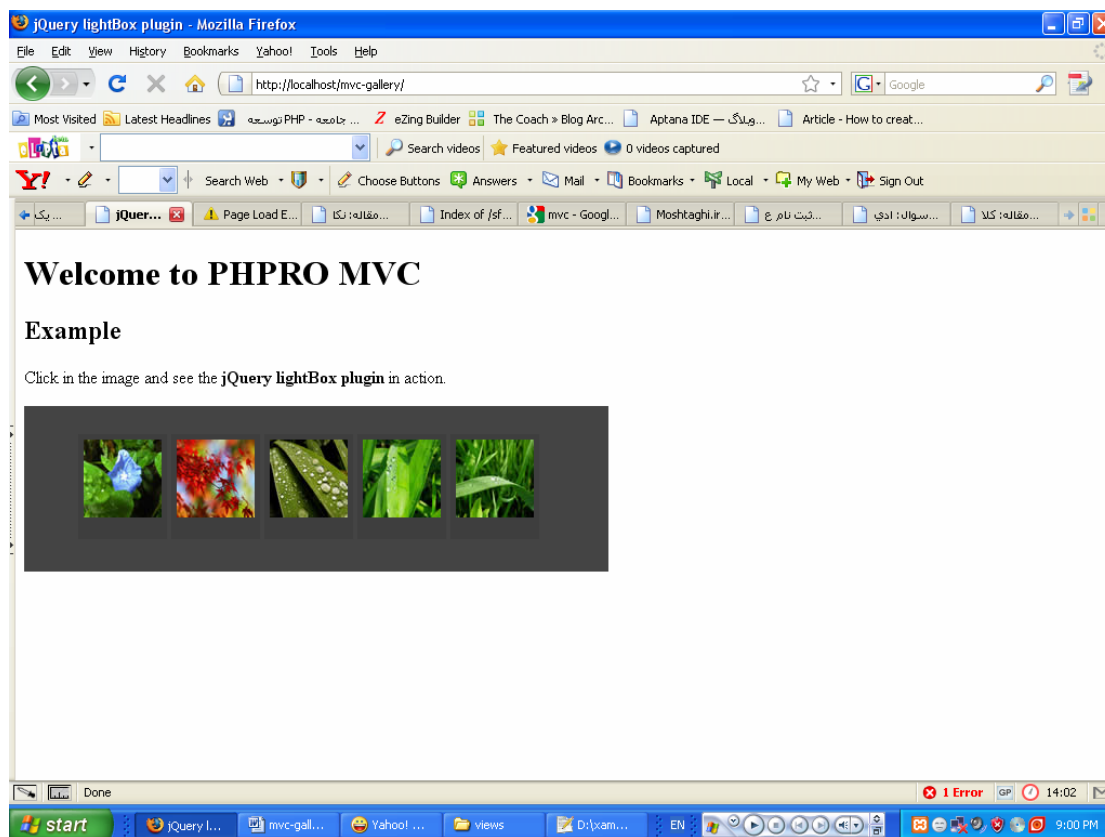
همین الان فهمیدیم که علامت **_** در این ساختار مفهوم خاص خودشو داره و نمیتونم همینجوری تو اسم فایلهاش ازش استفاده کنم مثلا **about_us.php** و اینکه این چه مفهومی داره تو مراحل بعدی میگم براتون.

خوب اگه مراحلو درست رفته باشید باید الان بتونید این صفحه رو ببینید:

<http://localhost/mvc-gallery/index/about>



نتیجه اخلاقی اینه که شما در این لحظه اولین صفحه تحت الگوی mvc خودتونو ساختید . تبریک میگم ، التماس دعا ☺
 خوب حالا که دونستیم تحت این معماری چه طوری صفحه بسازیم ! دیگه موقشه که واقعا رو پروژمون تمرکز کنیم ، در این مرحله شما فایل index.html گالریونو باز کرده و هر چی توش هست رو به فایل index.php پوشه views اضافه کنید و بعد اگر به ادرس <http://localhost/mvc-gallery/> برید به شکلی شبیه عکس پایین ببینید:



خوب حالا باید از مکانیزم نمایش عکسها سر در بیاریم تا بتونیم به دلخواه تغییرش بدیم ، اگه دقت کنید میبینید هر عکسی که نمایش داده میشه کدش به اینصورته:

```
<li>
    <a href="photos/image\ .jpg" title="Utilize a flexibilidade dos seletores da jQuery e crie um
    grupo de imagens como desejar. $('#gallery').lightBox();">
        
    </a>
</li>
```

همونطور که میبینید ما برای داشتن یه عکس در گالری باید در یک تگ دو تگ داشته باشیم که یکیش لینک ما به عکس با اندازه واقعی ای لینک شده و دومیش یه تگ عکس که در اون عکس کوچیک ما قرار داره!
خوب اگه ما بخوایم یک گالری داینامیک داشته باشیم باید این ساختارو به طور خود کار بسازیم که البته باید href و title و src رو از دیتابیسمون اضافه کنیم.
خوب پس الان باید یه دیتابیس طراحی کنم!
اسم اونو میزاد gallery و ساختارش هم :

```
-- phpMyAdmin SQL Dump
-- version 3.1.1
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Mar 06, 2009 at 05:16 PM
-- Server version: 5.1.30
-- PHP Version: 5.2.8

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

--
-- Database: `gallery`
--
```

```

-----

--
-- Table structure for table `gallery_groups`
--

CREATE TABLE IF NOT EXISTS `gallery_groups` (
  `g_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `g_name` varchar(100) COLLATE utf8_persian_ci NOT NULL,
  `g_hidden` tinyint(1) NOT NULL,
  PRIMARY KEY (`g_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_persian_ci AUTO_INCREMENT=2 ;

-----

--
-- Table structure for table `gallery_img`
--

CREATE TABLE IF NOT EXISTS `gallery_img` (
  `img_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `img_url` varchar(400) COLLATE utf8_persian_ci NOT NULL,
  `img_title` varchar(200) COLLATE utf8_persian_ci NOT NULL,
  `img_alt` varchar(200) COLLATE utf8_persian_ci NOT NULL,
  `g_id` int(10) unsigned NOT NULL,
  PRIMARY KEY (`img_id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_persian_ci AUTO_INCREMENT=2 ;

```

اینطوری و شما راحت میتونید با استفاده از sql بالا دیتابیسونو بسازید.
 کافیه با استفاده از phpmyadmin دیتابیسونو بسازید و بعد با استفاده از تب sql این دستوراتو توش اجرا کنید!
 در حقیقت ما این جا دو تا جدول داریم که کار جدول gallery_groups اینه که توش گروه بندی های مارو ذخیره میشه کرد تا بعدا بشه عکس ها رو تحت دسته بندی های خاص اینجا ، مشاهده کرد مثل عکس های خانوادگی و هنری و طبیعت و ...
 و جدول gallery_img که کار ذخیره سازی عکسهای ما و اطلاعات مربوط به اونها رو بر عهده داره
 خوب حالا که دیتابیسمون کامل شد باید بریم سراغ پوشه model تا فایل db.class.php رو که بر طبق معماری سه لایه وظیفه اتصال و کار کردن با دیتابیس رو بر عهده داره ویرایش کنیم.
 اما وقتی این فایلو باز میکنید دو تا مشکل اساسی برای ادمای مبتدی ای مثل من وجود داره !
 بر طبق اصول شیء گرایی ساخته شده و برای ارتباط با دیتابیس از PDO استفاده کرده!
 خوب حالا این pdo چیه؟

منبع : <http://ir.php.net/manual/en/intro.pdo.php>

The PHP Data Objects (PDO) extension defines a lightweight, consistent interface for accessing databases in PHP. Each database driver that implements the PDO interface can expose database-specific features as regular extension functions. Note that you cannot perform any database functions using the PDO extension by itself; you must use a [database-specific PDO driver](#) to access a database server.

PDO provides a *data-access* abstraction layer, which means that, regardless of which database you're using, you use the same functions to issue queries and fetch data. PDO does *not* provide a *database* abstraction; it doesn't rewrite SQL or emulate missing features. You should use a full-blown abstraction layer if you need that facility.

PDO ships with PHP 5.1, and is available as a PECL extension for PHP 5.0; PDO requires the new OO features in the core of PHP 5, and so will not run with earlier versions of PHP.

خلاصه مطلب خیلی چیز خوبی که از نسخه پنج پی اچ پی اضافه شده و میشه گفت به لایه ایه بین شما و دیتابیس که اگه ازش استفاده کنید اگه دیتابیسونو مثلا از my_sql به oracle تغییر دادید دیگه احتیاجی ندارید کدهاتونو عوض کنید و فقط کافیه تنظیمات اولیه رو تغییر بدین و میشه گفت به دیلماجه بین شما و انواع وسیعی از دیتابیس که اگه بلد باشین احتیاجی به اشنایی با تویع کار کردن هر کدوم از دیتابیس ندارید ! ولی اگه زبون این دیلماجه رو بلد باشین!!!

من که فقط توابع کار با my_sql رو بلدیم برای همین از روی بی سوادى مجبور شدم فایل db.class.php رو طورى تغییر بدم تا فقط بتونه با my_sql با اون توابعی که بلدم ارتباط برقرار کنه!
کد db.class.php اولیه به اینصورت:

```
<?php

class db{

    /** Declare instance */
    private static $instance = NULL;

    /**
     *
     * the constructor is set to private so
     * so nobody can create a new instance using new
     */
    private function __construct() {
        /** maybe set the db name here later */
    }

    /**
     *
     * Return DB instance or create initial connection
     */
    * @return object (PDO)
    *
    * @access public
    */
    public static function getInstance() {

        if (!self::$instance)
        {
            self::$instance = new PDO("mysql:host=localhost;dbname=periodic_table", 'username', 'password');;
            self::$instance->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        }
        return self::$instance;
    }

    /**
     *
     * Like the constructor, we make __clone private
     * so nobody can clone the instance
     */
}
```

```

*
*/
private function __clone(){
}

} /** end of class */

?>

```

که خیلی ترو تمیزو خوبه ولی من زیاد با شیء گرایى آشنا نیستم و خوب نتایجش همون فایل میشه به اینصورت:

```

<?php

class db{

/** Declare instance */
private static $instance = NULL;
        private static $host='localhost';
        private static $user='root';
        private static $pass="";
        private static $database_name='gallery';
        private static $link;

/**
 *
 * the constructor is set to private so
 * so nobody can create a new instance using new
 */
private function __construct() {
/** maybe set the db name here later */

}

public static function getInstance() {

if (!self::$instance)
{

/* self::$instance = new PDO("mysql:host=localhost;dbname=gallery", 'root', "");
self::$instance-> setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);*/

}
return self::$instance;
}

function query($str){
        self::$link=mysql_connect(self::$host,self::$user,self::$pass);
                mysql_select_db(self::$database_name);
if($result=mysql_query($str,self::$link))
return $result;
else

```

```

return "sorry !";
}

/**
 *
 * Like the constructor, we make __clone private
 * so nobody can clone the instance
 *
 */
private function __clone(){
}

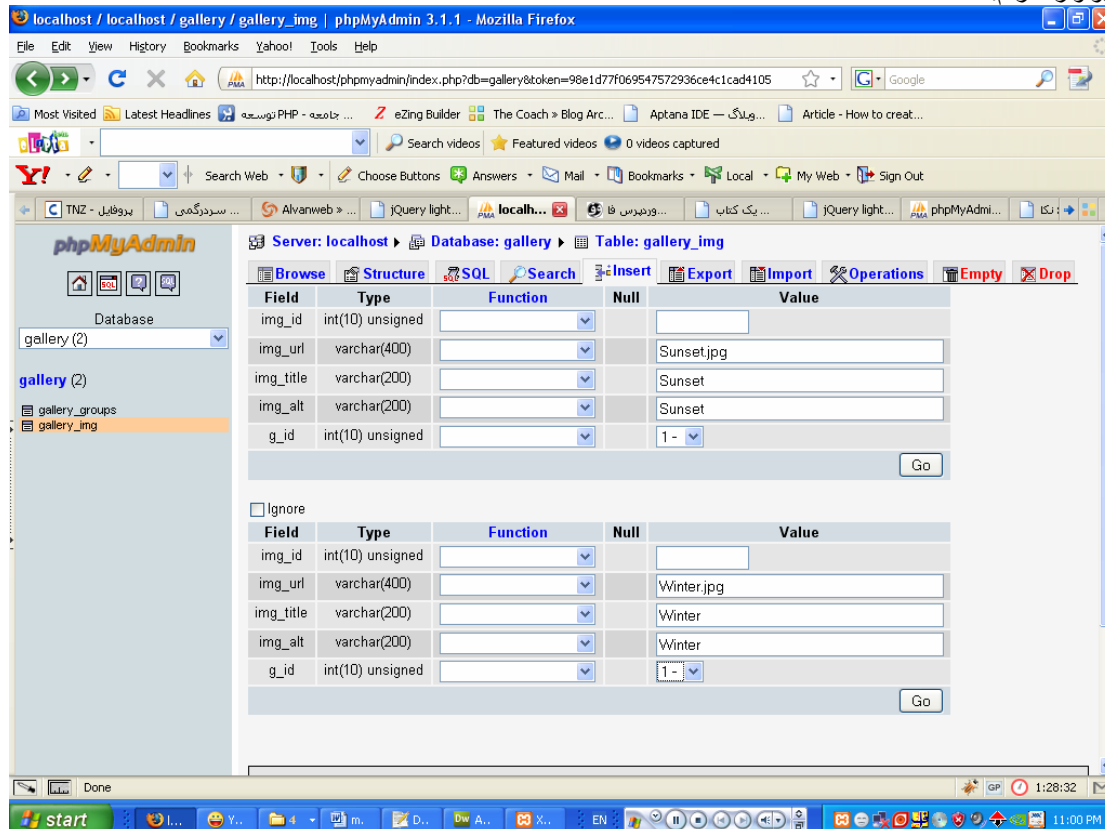
} /** end of class */

?>

```

خوب با اینکه میدونم افتضاحه ولی خوب ، کارو راه میندازه و در اون یه فانکشن به نام query راه انداختم که همون کارایی که قبلا برای گرفتن نتیجه از my_sql میکردیمو انجام میده و کافیه از فایلمون کوئری مورد نظرو بهش بدیم تا نتیجه رو برامون بفرسته!
حالا چه طوریشو در مراحل بعد میبینیم!

خوب حالا برای اینکه یه صفحه بسازیم تا از پایگاه دادمون اطلاعات بگیریم ، همون فایل index.php پوشه کنترل رو باز میکنیم تا تغییرش بدیم.
اما قبلش باید برای نمونه اطلاعاتی رو وارد بانک اطلاعاتیمون کنیم پس با استفاده از phpmyadmin به صورت زیر اطلاعاتی رو وارد کردم:



و سپس باید عکس های ویندوزمون در مسیر Documents\My Pictures\Sample Pictures که به همین نام هایی هستند که وارد دیتابیس کردیم رو در پوشه ی photos پروژمون کپی کنیم !

خوب تقریبا آماده ایم پس صفحه index.php پوشه کنترلر لایه رو بصورت زیر تغییر میدیم:

```
<?php

Class indexController Extends baseController {

public function index() {

//create gallery item from data base
echo('<div id="gallery">
    <ul>');

        $result=db::query('SELECT * from gallery_img');
        while($row=mysql_fetch_assoc($result))
        {

            echo('    <li>
                <a href="photos/'.$row["img_url"].'" title="'.$row["img_title"].'">
                    
                </a>
            </li>');
        }
echo('    </ul>
</div>');

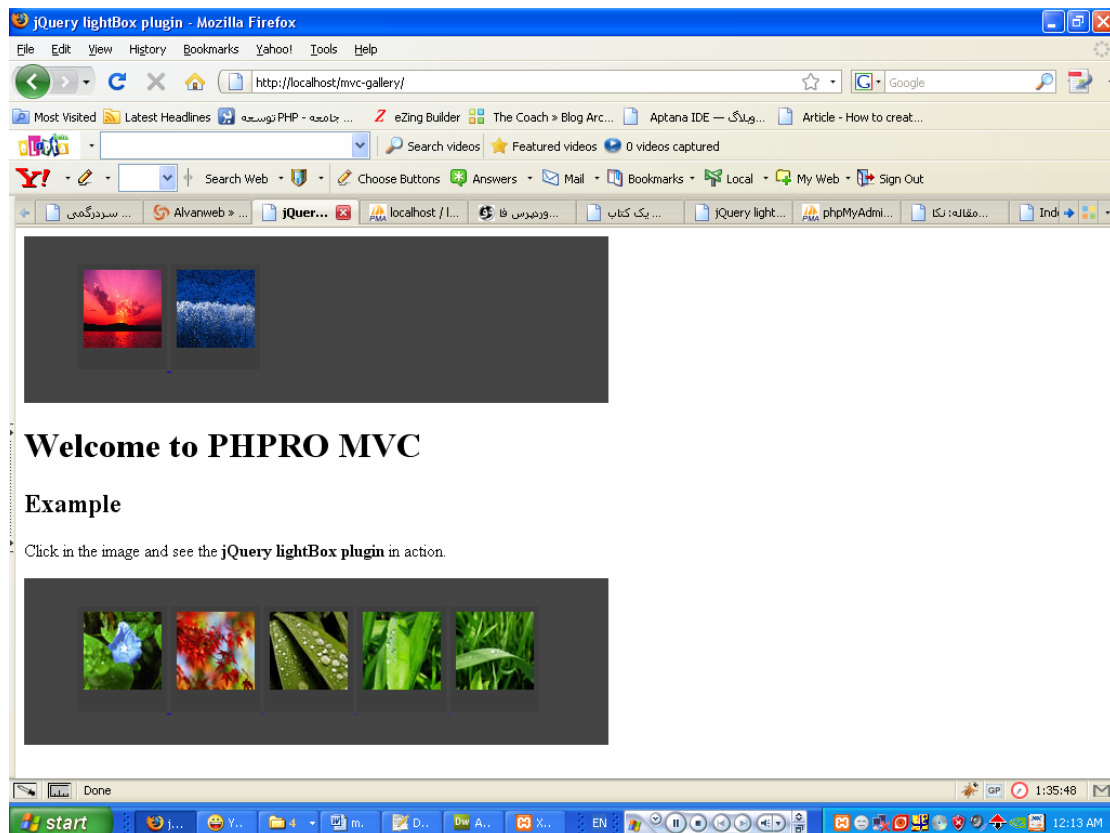
        /** set a template variable */
        $this->registry->template->welcome = 'Welcome to PHPRO MVC';
        /** load the index template */
        $this->registry->template->show('index');
    }

    public function about() {
        /** set a template variable */
        $this->registry->template->about_us = 'hi ! my name is farid :-)';
        /** load the index template */
        $this->registry->template->show('about');
    }

}

?>
```

و نتیجهش :



و نتیشو مشاهده میکنید!
به به ! اینم از اولین صفحه تحت این الگو که اطلاعاتشو از دیتابیس میگیره!
خوب بیاین ببینیم چی شد که این شد ؟

```
//create gallery item from data base
echo("<div id='gallery'>
    <ul>');

    $result=db::query('SELECT * from gallery_img');
    while($row=mysql_fetch_assoc($result))
    {

        echo('    <li>
        <a href="photos/'.$row["img_url"].'" title="'.$row["img_title"].'">
        
        </a>
        </li>');
    }
echo(' </ul>
</div>');
```

قلب ارتباط با دیتابیس و درست کردن ساختار گالریمون همین کده!
و کده:

```
$result=db::query('SELECT * from gallery_img');
```

کار ارجاع به کلاس db و دادن ارگومان به فانکشن query رو انجام میده و نتیجه رو تو result میریزه و سپس حلقه while کار رو ادامه میده و تک تک عکسهای موجود تو دیتابیسو به یک ایتm گالری تبدیل میکنه!

اما خوب زیا اصولی کار نکردیم!!!

اخره باید ایجاد ایتm در index.php در پوشه کنترل رو طوری میساختیم که اشیا رو به یه متغیر در پوشه view پاس بده پس :

```
<?php

Class indexController Extends baseController {

public function index() {

//create gallery item from data base
$item=<div id="gallery"> <ul>;
        $result=db::query('SELECT * from gallery_img');
        while($row=mysql_fetch_assoc($result))
        {

$item.=' <li>
        <a href="photos/".$row["img_url"]." title="'.$row["img_title"].">
        
        </a>
        </li>;
        }
$item.=' </ul>
</div>;

        /** set a template variable */
        $this->registry->template->welcome = $item;
        /** load the index template */
        $this->registry->template->show('index');
    }

    public function about() {
        /** set a template variable */
        $this->registry->template->about_us = 'hi ! my name is farid :-)';
        /** load the index template */
        $this->registry->template->show('about');
    }

}

?>
```

یادتونه تو به جایی گفتم تو این سیستم " _ " معنی خاص خودشو داره ؟ الان موقشه تا بفهمیم چی کار میکنه!
از نظر من سمت کابری سایت تقریبا تمومه! البته کلی چیز میز میشه اضافه کرد تا کامل و بالغ شه ، ولی هدف اون نیست!
به نظرم الان موقشه تا بباییم و بخش مدیریتیو بسازیم تا مدیر بتونه به مدیریت عکس های گالری و مجموعه ها و حذف و اضافه اونها بپردازه.

اما قبل از شروع به درست کردن این بخش باید به یک نکته توجه کنیم. هیچ متوجه شدید موقعه ای که اولین صفحه مونو که همون صفحه درباره ما بود ساختیم چه طوری بهش ادرس دادیم؟

<http://localhost/mvc-gallery/index/about>

چه قدر شگالش عجیبه نه؟

در واقع نه دارای پسوند php. و نه واقعا ما پوشه ای به نام index داریم پس این ادرس چه طوری به ادرس واقعیش تبدیل میشه؟ در واقع این یکی از کارهایی که فایلی به نام htaccess. برای پروژمون انجام میدی و در پوشه پروژمون قرار داری و میتونه کلی کارهای دیگه کنه:

RewriteEngine on

RewriteCond %{REQUEST_FILENAME} !-f

RewriteCond %{REQUEST_FILENAME} !-d

RewriteRule ^(.*)\$ index.php?rt=\$1 [L,QSA]

ErrorDocument 404 /error404.html

اما واقعا htaccess چیه؟ از اونجایی که توضیحش خیلی زیاده در اخر مقاله با ذکر منبع توضیح داده شده و برای این ارجاع ندادم به سایتش چون لازم دیدم، پی اچ پی کاری که میخواد به مقداری حرفه ای تر بشه حتما با این فایل و کارهایی که میتونه انجام بده آشنا بشه.

و اما بعد

این فایل رو به صورت زیر تغییر بدین تا کاملتر شه:

RewriteEngine on

RewriteCond %{REQUEST_FILENAME} !-f

RewriteCond %{REQUEST_FILENAME} !-d

RewriteRule ^(.*)\$ index.php?rt=\$1 [L,QSA]

ErrorDocument 404 /error404.html

<Files .htaccess>

order allow,deny

deny from all

</Files>

Options All -Indexes -ExecCGI -MultiViews

DirectoryIndex index.php index.html

AddDefaultCharset utf-8

<FilesMatch "\.(php|phphtml|html|xml)\$">

Header unset cache-control

</FilesMatch>

<IfModule sapi_apache2.c>

php_value register_globals 0

php_value session.auto_start 0

</IfModule>

اینکه دقیقاً این دستورات چه می‌کنن از حوصله این مقاله و سواد من خارجه ولی با کمی جستجو میتونید بفهمین!

خوب میریم سراغ بخش مدیریت!
قدیم من یه پوشه به نام admin میساختمو در اون فایلهای مدیریتمو قرار میدادم ولی اینجا این مفهوم یه نمه تغییر کرده! در مراحل بعد من فقط میگم چه فایلهایی رو میسازیم تا بعد به تحلیلشون بپردازیم:

در پوشه views فایل های زیر رو بسازید:

```
admin_index.php
admin_login.php
admin_photos.php
admin_photosRemover.php
admin_photosUpdate.php
admin_photosInsert.php
```

که محتوای همشون کد زیره:

```
<h1><?php echo $contents; ?></h1>
```

و حالا در پوشه controller فایل admin.php رو بسازید که محتوایش کد زیره:

```
<?php
Class adminController Extends BaseController {
public function index()
{
    // define all links of administrator
    $link = '<a href="admin/photos">admin_photos</a><br><a href="admin/photosInsert">photos_Insert</a>';

    $this->registry->template->contents = $link;
    $this->registry->template->show('admin_index');
}

public function login(){
    $this->registry->template->contents = 'This is login page';

    $this->registry->template->show('admin_login');
}

public function photos(){

    //show all of img

    $item='
    <table width="100%" border="1">
```

```

        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
';

$result=db::query('SELECT * from gallery_img inner join gallery_groups on
gallery_img.g_id= gallery_groups.g_id');

while($row=mysql_fetch_assoc($result))
{
    $item.="
    <tr>
    <td><a href='photosRemover?id=$row[img_id]' >delet</a></td>
    <td><a href='photosUpdate?id=$row[img_id]' >update</a></td>
    <td><img src='../photos/$row[img_url]' alt='$row[img_alt]' height='100' width='100' /></td>
    <td>$row[img_title]</td>
    <td>$row[g_name]</td>
    </tr>

        ";
    }

    $item.='</table>';

$this->registry->template->contents = $item;

    $this->registry->template->show('admin_photos');
}

public function photosRemover(){

    $result=db::query(sprintf("DELETE FROM gallery_img WHERE img_id=%s",$_GET['id']));
    if($result){
        $item='<meta http-equiv="Refresh" content="0;URL=photos" />';
    }else{
        $item='sorry ! there was a problem in delet item from gallery';
    }

    $this->registry->template->contents = $item;

```

```

    $this->registry->template->show('admin_photosRemover');
}

public function photosUpdate(){
    if(isset($_POST["g_id"])){

        $upload_file = $_FILES["img_url"]["name"];
        $hast = true;

        while($hast){
            if(file_exists('photos/'.$upload_file))
                $upload_file = "1" . $upload_file;
            else
                $hast = false;
        }

        if
(move_uploaded_file($_FILES["img_url"]["tmp_name"],'photos/'.$upload_file)){

            $result=db::query("UPDATE gallery_img SET
img_url='".$upload_file."',img_title='".$_POST['img_title']."',img_alt='".$_POST['img_alt']."' WHERE
img_id='".$_GET['id']");

            if($result){
                $item=<meta http-equiv="Refresh"
content="0;URL=photos" />';
            }else{

```

```

problem in update item from gallery';
    }

    }
    else{
        echo "sorry! upload is failed";
    }

}

} else{
    $item='
data">
    <form id="insertimg" name="insertimg" method="post" action="" enctype="multipart/form-
    <p align="center">
        <input name="g_id" type="hidden" id="g_id" value="1" />
    </p>
    <table width="100%" border="0">
        <tr>
            <td><input type="file" name="img_url" id="img_url" /></td>
            <td>&#1575;&#1583;&#1585;&#1587; &#1593;&#1705;&#1587;</td>
        </tr>
        <tr>
            <td><input type="text" name="img_title" id="img_title" /></td>
            <td>&#1593;&#1606;&#1608;&#1575;&#1606; &#1593;&#1705;&#1587;</td>
        </tr>
        <tr>
            <td><input type="text" name="img_alt" id="img_alt" /></td>
            <td>alt</td>
        </tr>
    </table>
    <p align="center">
        <label>
            <input type="submit" name="button" id="button"
value="&#1579;&#1576;&#1578;" />
        </label>
    </p>
    </form>
    ';
}

$this->registry->template->contents = $item;

```

```

    $this->registry->template->show('admin_photosUpdate');
}

public function photosInsert(){

    if(isset($_POST["g_id"])){

        $upload_file = $_FILES["img_url"]["name"];
        $hast = true;

        while($hast){

            if(file_exists('photos/'.$upload_file))
                $upload_file = "1" . $upload_file;
            else
                $hast = false;

        }

        if (move_uploaded_file($_FILES["img_url"]["tmp_name"],'photos/'.$upload_file)){

            $result=db::query("INSERT INTO `gallery`.`gallery_img`
(
    `img_id` ,
    `img_url` ,
    `img_title` ,
    `img_alt` ,
    `g_id`
)

```

```

VALUES (
NULL , ".$upload_file.", ".$_POST["img_title"].",
".$_POST["img_alt"].", '1')");

if($result){
    $item='<meta http-equiv="Refresh"
content="0;URL=photos" />';
} else{
    $item='sorry ! there was a problem in insert item
from gallery';
}

    }
else{
    echo "sorry! upload is failed";
}

} else{
    $item='
<form id="insertimg" name="insertimg" method="post" action="" enctype="multipart/form-
data">
    <p align="center">
        <input name="g_id" type="hidden" id="g_id" value="1" />
    </p>
    <table width="100%" border="0">
        <tr>
            <td><input type="file" name="img_url" id="img_url" /></td>
            <td>&#1575;&#1583;&#1585;&#1587; &#1593;&#1705;&#1587;</td>
        </tr>
        <tr>
            <td><input type="text" name="img_title" id="img_title" /></td>
            <td>&#1593;&#1606;&#1608;&#1575;&#1606; &#1593;&#1705;&#1587;</td>
        </tr>
        <tr>
            <td><input type="text" name="img_alt" id="img_alt" /></td>
            <td>alt</td>
        </tr>
    </table>
    <p align="center">
        <label>
            <input type="submit" name="button" id="button"
value="&#1579;&#1576;&#1578;" />
        </label>
    </p>
</form>
';

```

```

}

$this->registry->template->contents =$item;

        $this->registry->template->show('admin_photosInsert');

}

}
?>

```

خوب امیدوارم به کدهای بالا دقت کرده باشید ، اولین نکته اینه که حالا میدونیم _ برای چی بکار میره ! ببینید من تونستم یه گروه تعریف کنم به نام admin که اعضای اون با علامت _ مشخص میشه مثل admin_index.php که فایل اصلی گروه ماست یا admin_photos.php که مدیریت گالری ما رو به عهده داره و حالا که پوسته ها ساخته شده باید بخش کنترل این فایلها هم ساخته بشه!

پس اومدم فایل admin.php رو ساختم که یه نکته اصلی داره و اونم

Class adminController Extends baseController

که حتما باید همه ی اعضای گروهمون تو این کلاسمون باشن که این کلاسمون هم باید از کلاس baseController ارث بیره و اسمش هم باید Controлле + نام گروه ، باشه.

وقتی این کلاسو تو فایل پوشه کنترلتون ایجاد کردین ، باید برای تک تک اعضای گروهتون یه فانکشن تعریف کنید مثل:

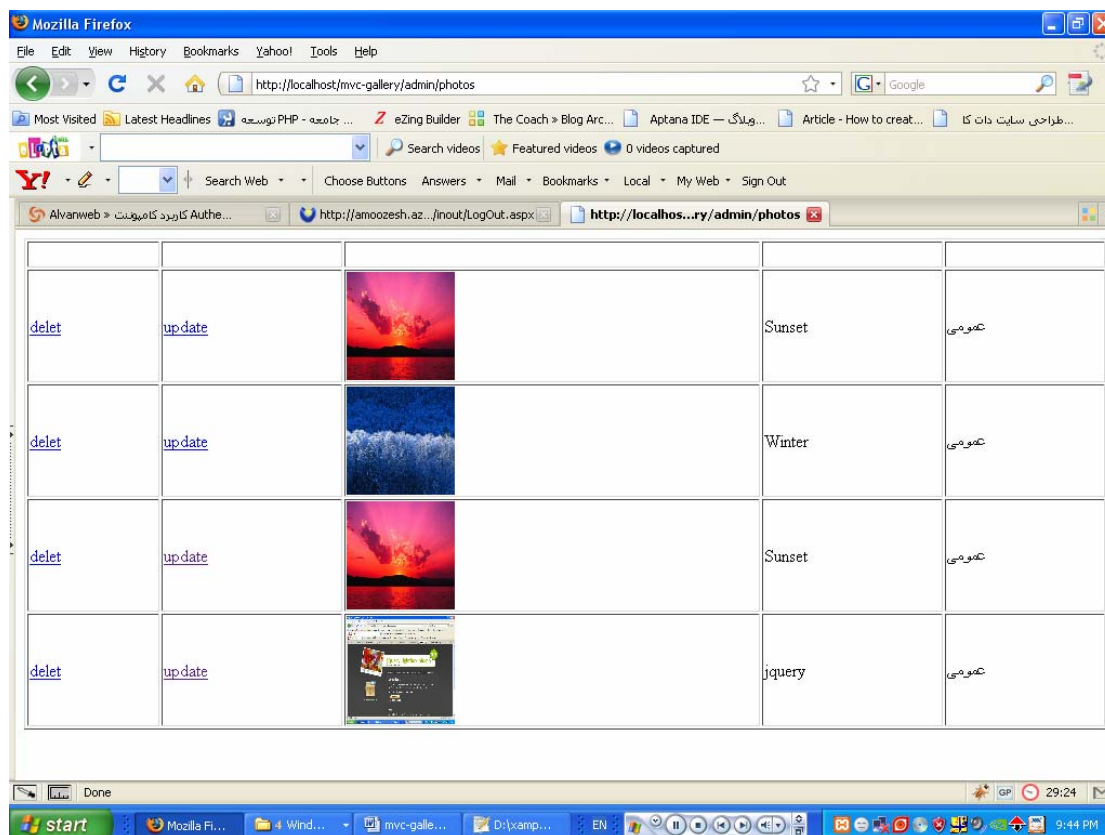
```
public function index()
```

خوب دیگه بقیه کدهای فانکشن ها ، پی اچ پی معمولیه و فرض بر اینه که شما میدونید!

مثلا شما میتونید با زدن ادرس زیر به صفحه ی اول کنترل پنلتون برید:

<http://localhost/mvc-gallery/admin>

و از اونجا با زدن لینک مدیریت عکس ها به بخش مدیریت گالری برید:



کار ما به آخر رسید !!! یه گالری داینامیک که دارای بخش مدیریت هست که با دیتابیس تعامل داره! البته در نظر داشته باشید که این یه پروژه آموزشیه و کلی چیز میز کم داره و کلی چیز میز دیگش باید اصلاح بشه مثلاً باید یه قسمت لوگین برای مدیریت اضافه شه یا مثلاً هنگام آپلود عکس باید یه عکس کوچیک از عکس ساخته شه تا اون عکس در پیش نمایش گالری نمایش داده شه نه همون عکسای بزرگ با محدودیت سایز چون باعث شدن سرعت میشه و ...

و میدونم که به خاطر بی سواد بودنم همیشه زیاد به متنی که نوشتم و کدهایی که زدم اعتماد کرد ، ولی سعی کردم اون چیزو که دوستم با بقیه به اشتراک بزارم. امیدوارم پر حرفیهایی من سرتونو نبرده باشه و تونسته باشم به اندازه ی توانم مفید بوده باشم ، سعی کردم مثل همیشه باشم و حتی اگه کتابی و رسمی نوشتم برای این بود که خودم از متنی رسمی که فقط ، اومدن هر کلمه رو ترجمه کردن حالم بهم میخوره. شما میتونید کل این پروژه رو همراه با فایل پایگاه دانش از ادرس زیر دریافت کنید:

<http://blog.pitm.net/downloads/mvc-gallery.zip>

در آخر با اینکه کار خاصی نکردم ولی اعتراف میکنم یکی از بزرگترین تلاشای من برای یاد دادن به بقیه بوده و دوس دارم اینو تقدیم به تمام کسانی که دوستشون دارم بکنم و تقدیم به کسی که ای کاش اونقدر بزرگ بودم که میتونستم دوش داشته باشم ، کسی که با اومدنش تمام دنیا بهشت میشه و ادما طعم واقعی عدالتو میچشن.

At last, he will come ...

موفقو پیروز باشید

۱۳۸۷/۱۱/۲۳

فرید احمدیان

بخش ۱: htaccess چیست ؟

يك فايل بسيار ساده با محيط تكست، چگونه ميتوان آنرا توليد كرد ؟
براي ساختن اين فايل تنها كافيست يك محيط تكست اديتور داشته باشيد . مثل notepad
اين فايل در كجا هست ؟
اين فايل در دايركتوري اصلي وبسایت شما قرار دارد و هر دايركتوري ديگري كه بخواهيد.
وظيفه اين فايل چيست ؟
تقريبا تمامي محدوديت هاي اعمايي بر آن دايركتوري را بر عهده دارد و ميتواند پيغام هاي خطا را مديريت كند و
چندين وظيفه ديگر كه اگر كمى در تنظيمات آن قصور و غفلت كنيد شايد سايت شما بار گذاري نشود.
تمامي هاست ها از اين نوع گونه فايل حمايت ميكنند ؟
اين بستگي به ميزبان شما دارد اصولا از اين فايل حمايت ميكنند . اما اگر هم ميخواهيد مطمئن شويد ميتوانيد از
وبمستر سرور خود پيرسيد.
قبل از شروع كار حتما يك نسخه بك آپ از فايلي كه هم اكنون در دايركتوري شما قرار دارد بگيريد بعضي ويسرور
ها اين فايل را دارند اما داخلش هيچي نيست كه ديگر نيازي به پشتيبان گيري نيست.
قبل از شروع آموزش بگويم كه اين آموزش مختص صاحبان وب سايت ها هست و كاربراني كه داراي سايت
اختصاصي نيستند نميتوانند استفاده چندانى از اين مطلب بكنند اما خوب براي اطلاعات عمومي بد نيست به
ادامه ان بپردازيد . فكر كنم تقريبا ۹۰% صاحبان وب اين فايل را ديده باشند و بدون هيچ شكى در مورد ان از كنار ان
گذشته باشند براي همين هم احتمال ميدهم يكي از موضوعاتي كه در موردش كمتر مقاله اي پيدا ميكنيد همين
مطلب باشد . به هر حال تصميم گرفته ام كمى در مورد اين فايل بظاهر بي ارزش صفر كيلوبايتي توضيحاتي بدهم
.
با استفاده از اين فايل ميتوانيد پيغام هاي خطايي كه ظاهر ميشود را تغيير دهيد يا اينكه صفحه را بدون استفاده از
redirector كنترل پنل redirect كنيد يا اينكه اجازه ليست گرفتن از دايركتوري خاصي از وبسايتهون رو محدود كنيد.
قبل از شروع كار مطمئن شويد كه اديتور شما خاصيت word wrap را غير فعال کرده است اگر نه خودتان آنرا
غيرفعال كنيد.
فايل htaccess. در دايركتوري اصلي وبسایت شما يا سابدايركتوري شما قرار دارد دستوراتي كه در اين فايل قرار
ميگيرد نه تنها دايركتوري جاري را تحت تاثير خود قرار ميدهد بلكه كلييه زيردايركتوري هاي خود را نيز در بر ميگيرد .
بنابراين اگر از اين فايل بروي دايركتوري اصلي خودتان استفاده كنيد تمامي سايت را در برميگيرد . در ضمن
ميتوانيد اين فايل را در هر دايركتوري كه خواستيد نيز ايجاد كنيد تا محدوده كنترل در دست خودتان باشد . اگر با
CSS كار کرده باشيد كار كردن با اين فايل هم تقريبا مشابه همان است و براي افراي كه با ان اشنايي دارند
اسانتيد است به هر حال هر فايلي كه در دايركتوري ويژه اي قرار دارد به نزديكترين فايل htaccess خود مرتبط
ميشود و ان htaccess ان فايل را كنترل ميكنند.
براي ساختن اين فايل هيچ نيازي به نرم افزار يا سخت افزار ويژه اي نداريد تنها به يك notepad يا هر پردازنده متني
ديگري نيازمنديد.
براي ساختن اين فايل كه نامي ندارد و فقط هشت حرف براي نوع فرمت ان در نظر گرفته شده است "htaccess".
وقتي ميخواهيد اين فايل را در پردازنده متني خود ذخيره كنيد نام فايل به اين صورت ذخيره ميشود htaccess.txt .
براي اينكه چنين اتفاقي نيفتد كافيست از منوي file گزينه save as... را انتخاب كنيد در file type گزينه all files را
انتخاب كنيد و سپس در file name تايپ كنيد "htaccess". دات اچ تي اكسس را درون دو كوتيشن بگذاريد اگر
نتوانستيد اين كار را بكنيد بعدا ميتوانيد از طريق كنترل پنل يا نرم افزار ftp خود نام فايل را عوض كنيد . در مورد
استفاده از نرم افزار هاي ftp بگويم كه همانطور كه ميدانيد اينگونه نرم افزار ها براي اسودگي اتصال به ويسرورتان
ميشايد كه فايل ها را در سه نوع upload ميكنند كه سه گزينه binary , ASCII , Auto , وقتي اين فايل را با ftp اپلود
ميكنيد يادتان باشد نرم افزار شما در حالت ASCII باشد.
در ضمن يك مطلب ديگر هم در اين مورد كه بايد CHMOD كه مخفف Change MODE ميباشد را براي اين فايل
عوض كنيد پرميشن اين فايل بايد ۶۴۴ باشد يا RW-R—R
كه اين عمل هم براي افزايش امنيت ميباشد كه خوشبختانه همگي اشنايي با ان را داريد اما شايد بعدا بيشتر در
مورد permission ها توضيح دادم.
اين فايل به شما اجازه ميدهد كه شما كاربران و ويزيتور هاي خودتان را از صفحه اي به صفحه ديگر منتقل كنيد
حتما شده است كه بخواهيد اين كار را انجام دهيد مثلا در مورد خود من شما از parsx.com به weblog.parsx.com
ميرويد البته من از اين روش استفاده نكرده ام اما دلايل ديگري نيز ميتواند عامل شود كه شما از اين ريدايركت
استفاده كنيد شايد كاربر شما اشتباها ادرس صفحه اي از وبسایت شما را وارد كند يا اينكه شايد لينكي در
وبسایت شما بنا به دليلي شكسته شده باشد و از بين رفته باشد و ان صفحه پاك شده باشد به جاي اينكه كاربر
شما هيچ چيزي نبيند ميتوانيد آنرا به صفحه اصلي سايت بسته به سليقه خودتان منتقلش كنيد.
براي مثال فرض كنيد كاربري با يك پيغام خطا مواجه شد براي آنكه او را به صفحه ديگري منتقل كنيد بايد به اين
صورت در فايل بنويسيد:

ErrorDocument errornumber /filename.html

به جای errornumber شماره خطا را مینویسید اصولا خطاي ۴۰۴ زیاد اتفاق میفتد بنابراین میتوانید شماره ۴۰۴ را بنویسید.
اگر خواستید تمامی پیغام خطاها را به فولدر خاصی ببرید به اینصورت عمل کنید:

ErrorDocument ۴۰۴ /errorpages/notfound.html

در مورد شماره پیغام خطاها هم کمی بگویم:
۴۰۰ این پیغام اصولا زمانی ظاهر میشود که کاربر ادرس اشتباهی رفته باشد
۴۰۱ وقتی کاربر به دایرکتوری و یا جایی میرود که اجازه ورود به آنرا ندارد
۴۰۳ وقتی فایل طوری تنظیم شده است که کاربر اجازه دسترسی به آن را ندارد
۴۰۴ not found

۵۰۰ internal server error اشکال عموما از اسکریپت های داخلیست.
باز هم پیغام خطا داریم که زیاد به درد نمیخورد.
کاربرد دیگر این فایل استفاده از رمز عبور است که البته پیکر بندی آن کمی سخت تر از تنظیمات پیام های خطاست.
ابتدا فایلی به این نام ایجاد کنید httpasswd. این فایل هم خصوصیاتی چون فایل htaccess. دارد این فایل حاوی نام های کاربری و کلمه های عبور . که بصورت زیر آنها را تعیین میکنید در این فایل

```
Username:password
Username:password
Username:password
```

حواستان باشد که هر کدام در يك خط باشند و همچنین اشتباهای فضای space در آخر خط ها وارد نکنید . در ضمن برای نام های کاربری که مشکلی پیش نمی آید اما برای کلمه های عبور برای افزایش امنیت باید از نرم افزارهایی که آنها را تبدیل میکنند استفاده کنید . که بعدا کمی در این مورد توضیح خواهم داد.
یادتان باشد که این فایل را در فولدر های بالایی www یا public_html بگذارید یعنی در فولدر. main
حال برای آنکه مشخص کنید کدام بخش از سایت یا کدام دایرکتوری شامل رمز عبور بشوند باید به این صورت عمل کنید در فایل htaccess خطوط زیر را اضافه کنید

```
AuthUserFile /home/pathto/.htpasswd
AuthType Basic
AuthName "Secret Place"
```

```
<LIMIT GET POST>
require valid-user
</LIMIT>
```

دستورات فوق باعث میشوند که در صورت نادرستی و عدم تطابق کلمه عبور با آنچه که در فایل مربوطه ثبت شده از ورود کاربر جلوگیری کند . بنابراین اگر کاربری خواست وارد دایرکتوری شود که بدینوسیله محافظت میشد يك منوی پاپ آپ ظاهر میشود که نام کاربری و رمز عبور را میطلبد.
اگر خواستید در يك فولدر تنها از يك فایل محافظت کنید و عملیات پسورد گذاری تنها بر روی يك فایل انجام شود دستورات زیر را در فایل وارد کنید.

```
<files "filename.cgi">
AuthUserFile /home/pathto/.htpasswd
AuthType Basic
AuthName "Secret Place"
require valid-user
</files>
```

برای محافظت از خود فایل htaccess. نیز میتوانید خط زیر را وارد کنید:

```
<files ".htaccess">
order allow,deny
deny from all
</files>
```

اگر هاست شما از این فایل پشتیبانی میکند اما از SSI پشتیبانی نمیکند کافیسست خطوط زیر را وارد کنید تا از SSI هم پشتیبانی شود.

```
AddType text/html .shtml
AddHandler server-parsed .shtml
Options Indexed FollowSymLinks Includes
```

خط اول بیان میکند که کلیه فایل ها با پسوند shtml دارا صحت هستند خط دوم يك handler اضافه میکند خط سوم باعث میشود سرور این فایل ها را اجرا کند. بنابراین شما خودتان بدون نیاز به درخواست از ادمین سرور توانستید SSI را فعال کنید. اگر هم دوست دارید فایل های SSI بصورت جداگانه از فایل های html نباشند میتوانید خطوط زیر را اضافه کنید:

```
AddType text/html .shtml .html .htm
AddHandler server-parsed .shtml .html .htm
Options Indexes FollowSymLinks Includes
```

این خطوط باعث میشوند که کلیه فایل ها با پسوند html نیز به سرور برای اجرا شدن منتقل شوند تا اگر کدی که برای SSI باشد داشته باشد را اجرا کند و سپس صفحه شروع به بارگذاری کند.

قطع دسترسی از طریق IP
با این روش شما میتوانید بازه خاصی از آدرس IP یا IP خاصی از فرد خاصی را اضافه کنید تا آن فرد صاحب IP نتواند وارد صفحه خاص یا کل وبسایت شود. البته در کنترل پنل ها اصولا این گزینه به صورت گرافیکی هست اما توجه کنید که اگر در آنجا به این کار اقدام کنید فرد مورد نظر از کل سایت محروم خواهد شد اما با استفاده از این عمل میتوانید IP خاصی را از صفحه خاصی محروم کنید. مثلا فرض کنید آدرس گوگل بت رو بگذاریم تا نتواند وارد صفحه خاصی شود.
بنابراین خطوط زیر را اضافه کنید:

```
Order allow,deny
Deny from ipaddress
Allow from all
```

به جای ipaddress میتوانید آدرس ایپی را بنویسید مثل : ۱۷۲,۱۶,۲۰۴,۱۰۰
خط اول میگوید که شما میخواهید چکار کنید و خط دوم میگوید که آدرس مشخص شده را ممنوع کند. البته میدانید که هر فرد با هر قطع ارتباط از اینترنت مخصوصا سرویس های تلفنی بعد از هر بار اتصال آدرس ایپی آنها عوض میشود یعنی ۳ رقم آخر که برای اینکه کل ۲۵۶ آدرس را ممنوع کنید کافیسست ارقام اخري را ننویسید. خط آخر هم به دیگران اجازه میدهد که وارد سایت شوند. در ضمن شما میتوانید نام دامنه را نیز ممنوع کنید به این صورت که به جای آدرس ای پی بنویسید parsx.com.
وقتی کاربر سعی در ورود به صفحه ویژه که ممنوع شده است را دارد پیغام خطای ۴۰۳ را مبیند. اگر میخواهید مثلا دایرکتوری خاصی را از دسترسی همه ممنوع کنید اما اسکریپت ها و دیگر اجزایی که به فایل نیاز دارند بتوانند به آن دسترسی پیدا کنند کافیسست خط deny from all را اضافه کنید.
عوض کردن ایندکس فولدري خاص
اکثر میزبانهای وب فایلی که وظیفه لود شدن در هنگام باز کردن دایرکتوری خاصی را دارد با نام index قرار میدهند که با استفاده از htaccess میتوانید آنرا عوض کنید. خط زیر را به فایل اضافه کنید:

```
DirectoryIndex filename.html
```

حتما دیده اید که اگر فایل index.html در دایرکتوری نبود سرور فایل دیگری مثلا index.php را اجرا میکند روش به اینصورت است که در دستور فوق بعد از filename.html با يك فضای خالي نام های دیگر را وارد میکنید. بنابراین سرور در دایرکتوری ویژه ابتدا به دنبال فایل اولی و سپس به دنبال فایل دومی و به همین ترتیب جستجو میکند. اگر هم هیچ يك از فایل ها پیدا نشد پیغام خطای ۴۰۴ ظاهر میشود.

Redirect توسط htaccess
حتما میدانید که راههای زیادی برای ریدایرکت کردن فایلی خاص به فایل دیگر وجود دارد اما بهترین کار همین است

<http://www.domainname.com/newdirRedirect/OldDir/OldFile.html>

بنابراین دستور به شرح زیر است

Redirect oldlocation newlocation

منتهي يادتان باشد که newlocation باید نام ادرس کامل سایت باشد.
مخفي کردن محتويات دایرکتوري
شاید درون فولدرتان فایلي نداشته باشید که لود شود بنابراین لیست تمامی فایلهاي فولدر شما نمایان میشود که این اصولاً از لحاظ امنیتی هم مناسب نیست و اگر یادتان باشد در مطلبي که در مورد ساخت سایت در ویندوز ۲۰۰۳ دادم گزینه ای داشتیم برای اینکه این فایلها لیست نشوند همچنین در کنترل پنل هم چنین گزینه ای داریم اما خوب آن هم به صورت کلی در سایت ظاهر نمیشود. به هر حال میتوانید خط زیر را وارد کنید:
Options -Indexes

یا خط زیر را وارد کنید

IndexIgnore *

اما عکس این عمل نیز میتواند اتفاق بیفتد یعنی به صورت پیشفرض در سرور تنظیم شده است که فایلها لیست نشوند اما شما میتوانید این گزینه را با خط زیر غیرفعال کنید:

Options +Indexes

اگر هم میخواهید فایلها لیست شوند اما گروه خاصی از آنها مثلاً عکس ها لیست نشوند به صورت زیر عمل کنید:

IndexIgnore *.gif *.jpg

بعضي سرور ها بعضي از فایلها را نمیشناسند با استفاده از این فایل میتوانید خط زیر را وارد کنید تا آنها را بشناسد
مثلاً برای فایل های فلش میتوانید دستور زیر را وارد کنید

AddType application/x-shockwave-flash swf

Addtype به فایل اعلان میکند که باید نام فایلي جديد اعلان شود
ایجاد ممانعت از لینک دادن دیگران به فایلهاي شما
در مباحث پیشین در مورد پهنای باند صحبت کرده ام و میدانید که هر فایلي که از سرور سایت شما بارگذاری میشود در پهنای باند شما موثر است حال فرض کنید شما فایلي مثلاً عکسي یا موسيقي برای دانلود در سایت گذاشته اید! خوب بالطبع شما با استفاده از فرمول و محاسباتي که در ذهن خود انجام داده اید و نسبت به تعداد کاربرانان سنجیده اید که با پهنای باندتان مطابقت داشته باشد اما کافیسیت سایت هاي ديگري از فایل شما مثلاً يك عکس شما استفاده کنند و عکس شما را در سایت خود قرار دهند و ادرس عکس ادرس همان عکسي باشد که در سایت شما قرار دارد بنابراین از پهنای باند شما برای بار گذاری عکس استفاده میشود فلذا انروزي که پهنای باند شما از حد خود میگذرد شما باید هزینه کنید که باز هم پهنای باند بخريد در صورتي که میتوانید با استفاده از این فایل از انجام چنین کاري ممانعت کنید! منتهي قبلش باید توجه داشته باشید که آیا هاست سرور شما از mod_rewrite پشتیباني میکند یا خير؟!
اگر نمیکند این گزینه برای شما فایده ای نخواهد داشت و نمیتوانید با این کار مبادرت ورزید.
به فولدري که آن فایل یا مجموعه عکس هاي شما قرار دارد بروید و در فایل htaccess آن دایرکتوري خطوط زیر را اضافه کنید

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?mydomain.com/.*$ [NC]
RewriteRule \.(gif|jpg)$ - [F]
```

به جای mydomain.com نام دامنه خودتان را بنویسید مثلا parsx.com دستور فوق باعث میشود فایل‌های با پسوند gif jpg نتوانند در سایت‌های دیگر نمایش داده شوند اما شما میتوانید کار دیگری نیز بکنید مثلا بگذارید دیگران از عکس‌های شما استفاده کنند و سپس طی یک حرکت ناجوانمردانه فایل htaccess را گونه‌ای تعویض کنید که تمامی عکس‌ها و فایل‌ها با عکس دیگری نمایش داده شوند مثلا عکسی بگذارید که آدرس سایت خودتان را بگذارید. مثلا freeservers چنین کاری کرده است. اگر به عکس سایت‌هایش لینک دهید نام سایت خودش نمایش داده میشود.

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?mydomain.com/.*$ [NC]
RewriteRule \.(gif|jpg)$ [R,L]http://www.mydomain.com/badimage.gif
```

به جای mydomain نام دامنه خودتان و آدرس عکس خودتان را به جای آدرس فوق بگذارید.
این بود تمامی ماجرایی htaccess

بخش ۲

با htaccess. قلب وبسایت خود را تحت کنترل بگیرید (قسمت اول)
اگر از وبسرور آپاچی برای وبسایت خود استفاده می‌کنید، دیگر نیازی نیست برای هر مشکل جزئی و کوچک، به وبمستر هوستینگ‌تان تماس بگیرید. با یادگیری قابلیت‌های مهم فایل htaccess، می‌توانید کنترل کامل وبسایت خود را برعهده بگیرید.

فایل htaccess. یک فایل ساده اسکی (ASCII) می‌باشد که می‌توانید آن را به وسیله نرم‌افزارهای متنی ساده نظیر Notepad یا SimpleText به وجود آورید. دقت داشته باشید که این فایل دارای هیچ نامی نمی‌باشد و فقط دارای پسوند است که پسوند آن، htaccess می‌باشد. برای درست کردن این فایل می‌توانید فایل متنی خود را که مثلا به صورت fantazio.txt می‌باشد، تغییر نام داده و آن را به صورت htaccess. درآورید.

تذکر: اگر از ویندوز استفاده می‌کنید و هنگام تغییر نام دادن فایل به htaccess، پیغام خطایی مبنی بر اینکه «می‌بایست برای فایل خود نامی را اختیار کنید» دریافت می‌کنید، فایل را بر روی سرور آپلود کرده و سپس در سرور آن را تغییر نام دهید. این کار به وسیله نرم‌افزارهای اف‌تی‌پی نظیر CuteFTP یا AbsoluteFTP بسیار ساده می‌باشد.

در هنگام استفاده از فایل htaccess، دانستن سه نکته از اهمیت به‌سزایی برخوردار است:

نکته اول: فایل htaccess. را می‌بایست به صورت ASCII آپلود کرده و مجوز دسترسی به آن را بر روی ۶۴۴ (یا-rw-r--r--) تنظیم نمایید.

نکته دوم: فایل htaccess. را می‌توانید به پوشه‌های مختلفی اعمال نمایید. به عنوان مثال اگر آن را بر روی root آپلود نمایید، تنظیمات آن به کل وبسایت اعمال خواهد شد و اگر آن را درون پوشه‌ای نظیر images/ آپلود نمایید، تنظیمات آن به پوشه images و زیرپوشه‌های آن اعمال خواهد شد.

نکته سوم: هر htaccess را می‌بایست درون یک خط تاپی نمایید. یعنی در انتهای هر دستور، می‌بایست یک بار کلید Enter را فشار دهید.

جلوگیری از لیست شدن محتویات یک پوشه

فرض کنید پوشه‌ای از تصاویر دارید و نمی‌خواهید که افراد غریبه، محتویات این پوشه را دیده و آن‌ها را تماشا کنند. برای این کار، سه راه وجود دارد:

۱- ساخت یک صفحه index و قرار دادن آن در پوشه مورد نظر

۲- استفاده از فایل htaccess.

۳- استفاده از فایل‌های HEADER و README

IndexIgnore *

این کد از لیست شدن تمامی فایل‌های پوشه مورد نظر، جلوگیری می‌کند. حال فرض کنید که درون پوشه‌ای انواع و اقسام فایل‌ها نظیر فایل‌های اچ‌تی‌ام‌ال و تصاویر دارید و می‌خواهید که فقط از لیست شدن تصاویر جلوگیری شود. بدین‌منظور می‌توانید از کد زیر استفاده کنید:

IndexIgnore *.gif *.jpg

در واقع این کد از لیست شدن تمامی فایل‌هایی که به gif و jpg. ختم می‌شوند، جلوگیری می‌کند.

حال فرض کنید که سرور شما، به طور پیش‌فرض، از لیست شدن پوشه‌ها جلوگیری کرده است و شما می‌خواهید آن را تغییر دهید. برای این کار می‌توانید از دستور زیر استفاده کنید:

Options +Indexes

عوض کردن صفحه پیش‌فرض وبسایت

به طور کلی، تمامی وب‌سرورها از روی قانون خاصی صفحات وب را نشان می‌دهند. برای آنکه صفحات وب تنها به یک پسوند منحصر به فرد نظیر .htm. ختم نمی‌شوند و انواع و اقسام پسوندهای مختلف، برای صفحات وب وجود دارد.

حال فرض کنید که وب‌سرور شما، ابتدا فایل index.html را نشان می‌دهد و در صورت نبود آن، فایل‌های index.htm، index.php و ... شما می‌خواهید این روند را تغییر داده و فایل index.php را در اولویت قرار دهید. بدین منظور می‌بایست از دستور DirectoryIndex استفاده نمایید:

DirectoryIndex index.php index.cgi index.pl default.htm

کد بالا ابتدا به دنبال فایل index.php گشته و در صورت پیدا کردن آن، آن را به عنوان صفحه وب پیش‌فرض نمایش می‌دهد. اگر فایل index.php پیدا نشد، سرور به دنبال فایل بعدی که index.cgi می‌باشد گشته و همین روال، ادامه پیدا می‌کند. دقت داشته باشید که شما حتماً مجبور به استفاده از index برای نام فایل نیستید و از هر اسم دیگری نیز، می‌توانید استفاده کنید.

Redirects

شاید تا به حال برایتان اتفاق افتاده باشد که به وبسایتی مراجعه کرده باشید و سپس متوجه شوید که آدرس وبسایت به طور اتوماتیک عوض شده و در واقع شما به آدرس دیگری ارجاع داده شده‌اید. برای این کار می‌توانید از دستور Redirect استفاده نمایید:

<http://yoursite.com/newdirectory/newfile.html> Redirect /olddirectory/oldfile.html

همچنین می‌توانید یک پوشه را به پوشه دیگر ارجاع دهید:

<http://yoursite.com/newdirectory/> Redirect /olddirectory

Hot Linkin

در جامعه وب‌مسترها، عبارت Hot Linkin یک عبارت نفرت‌برانگیز است که از آن به عنوان «نقره‌داغ وب‌مسترها» نیز یاد می‌شود. اما معنای آن چیست؟

فرض کنید دارای یک سایت عکاسی هستید که روزانه تعداد زیادی عکس در زمینه‌های مختلف را در آن آپلود می‌کنید و در اختیار عموم قرار می‌دهید. حال فرد دیگری از عکس‌های شما در وبسایت خود استفاده می‌کند. این کار نه تنها برای شما هیچ سودی نخواهد داشت، بلکه از پهنای باند شما (BW) خواهد کاست و در نتیجه به زحمت و هزینه خواهید افتاد.

```

RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?mydomain.com/.*$ [NC]
RewriteRule \.(gif|jpg)$ - [F]

```

یادتان نرود که mydomain.com را به آدرس وبسایت خود تغییر دهید. همچنین می‌توانید کاری کنید در صورت استفاده از عکس‌های سایت شما، به جای نمایش آن‌ها، عکس مورد نظر شما نشان داده شود. مثلاً اگر سایت A از عکس‌های سایت شما استفاده کرد، عکسی را نمایش دهید که تبلیغ سایت شما باشد. برای این کار می‌توانید از کد زیر استفاده کنید:

```

RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?mydomain.com/.*$ [NC]
[R,L]http://www.mydomain.com/nasty.gif RewriteRule \.(gif|jpg)$

```

آدرس عکس خود را در خط آخر کد تغییر دهید.

تذکر mod_rewrite: یکی از قابلیت‌های وحشت‌ناک به‌دردبخور آپاچی می‌باشد که توضیح مفصل آن، احتیاج به یک مقاله صدصفحه‌ای دارد.

صفحات خطا

به احتمال قوی با خطای معروف ۴۰۴ در اینترنت سروکار داشته‌اید. خطایی که اعصاب آدم را به هم می‌زند. این خطا در صورتی ظاهر می‌شود که صفحه مورد نظر شما (به هر دلیل)، در دسترس نباشد. یعنی در واقع صفحه‌ای وجود نداشته باشد و شما آن را از مرورگر بخواهید.

وبسایت‌های حرفه‌ای و آن‌هایی که خواهان جذب مخاطب برای وبسایت‌هایشان هستند، این صفحات خطا را (که هر فرد با دیدن آن‌ها سریعاً دکمه close را می‌زند) سفارشی می‌کنند. یعنی به جای آن خطای ۴۰۴ تکراری، یک صفحه خطای مناسب (در فرمت وبسایت) می‌گذارند و به مخاطب راهنمایی‌هایی می‌دهند.

معمولاً در این نوع صفحات خطا، بیوگرافی کوچکی از وبسایت و زمینه کاری آن نوشته شده و سپس مخاطب را (به وسیله لینک و یا اتوماتیک) به صفحه اصلی وبسایت ارجاع می‌دهند.

برای ساخت صفحات خطا (که تعداد آن‌ها کم هم نیست!)، و ارجاع مخاطبین در صورت برخورد به صفحات خطای مختلف به آن‌ها، می‌توانید از کد زیر استفاده کنید:

```
ErrorDocument code /directory/filename.ext
```

به عنوان مثال کد زیر، مخاطب را در صورت مواجه شدن با خطای ۴۰۴، به صفحه notfound.html ارجاع می‌دهد:

```
ErrorDocument ۴۰۴ /notfound.html
```

منبع: iranmodern.com

پروردگارا

به من ارش ده

دلیری ده

بینش ده

مرا فهم ده

تا بپذیرم آنچه را که نمیتوانم تغییر دهم

تا تغییر دهم آنچه را که میتوانم تغییر دهم

تا تفاوت این دو را بدانم

تا متوقع نباشم دنیا و مردم ان مطابق میل من رفتار کنند