

**وزارت تحصیلات عالی**  
**دانشگاه کابل**  
**دانشکده کمپیوتر ساینس**  
**دپارتمنت انجینیری نرم افزار**

**سیستم فروشات آنلاین جواهر فروشی**  
**(Jewelry Online Shopping System)**

تهیه کننده: فاطمه فروتن  
استاد راهنما: پوهنیار محمد رفیع بائز

۱۳۹۵ - ۲۰۱۶

## خلاصه مونوگراف

ترتیب کننده: فاطمه فروتن

استاد راهنما: پوهنیار محمد رفیع بائر

### سیستم فروشات آنلاین جواهرفروشی (۵۴ صفحه)

مونوگراف «سیستم فروشات آنلاین جواهرفروشی» از ۷ فصل تشکیل شده است که در فصل اول به طرح مشکل، پس زمینه و ضرورت، هدف از این مونوگراف، میتودولوژی (به طور مختصر) و محدودیت‌هایی که بر سر راه این مونوگراف وجود داشته است، پرداخته شده است. همچنین در فصل دوم که میتودولوژی می‌باشد به بررسی مفصل‌تر برنامه‌ها و زبان‌های برنامه‌نویسی که در این مونوگراف استفاده شده‌اند، مشخصات و فواید هر کدام پرداخته شده است.

فصل سوم، تجزیه و تحلیل سیستم یا Data Modeling می‌باشد که اولین مرحله (Phase) در طراحی و دیزاین یک سیستم دیتابیس به حساب می‌آید. در این فصل موجودیت‌ها (Entity) مونوگراف مطالعه شده‌اند و خصوصیت‌های (Properties) مورد نیاز هر یک از موجودیت‌ها تعیین گردیده است.

در فصل چهارم به دیزاین فیزیکی دیتابیس پرداخته شده است و برای هر کدام از Weak و Strong Entity در فصل‌های موجود در مونوگراف، جدول‌های مربوط با فیلدهای لازم در نظر گرفته شده است.

فصل پنجم، برنامه نویسی Client Side می‌باشد که به User Interface پرداخته شده است و توسط زبان‌های HTML، CSS و JavaScript صفحات انترنتی و فورم‌های مورد نیاز در هر صفحه برای استفاده توسط user طراحی و دیزاین شده است.

در فصل ششم که موضوع آن برنامه نویسی Server Side می‌باشد و آخرین فصل در قسمت توسعه (Development) مونوگراف «سیستم فروشات آنلاین جواهرفروشی» است، به مسایل اساسی

Back-End مانند ارتباط با دیتابیس (Database Connectivity)، الگوریتم‌ها و تکنیک‌های لازم برای Functionality‌ها و منطق‌های (Logic) موجود و... پرداخته شده است.

فصل هفتم که آخرین فصل این مونوگراف را تشکیل می‌دهد، جمع‌بندی و نتیجه‌گیری تمام سیستم، تکنیک‌ها و روش‌های به کار رفته در مراحل مختلف طراحی و دیزاین این مونوگراف می‌باشد که در پایان یک بار دیگر به ضرورت استفاده از برنامه‌های Open Source و فواید آن تاکید شده است.

تقدیم بہ

پدر مہربان و مادر فداکارم

و خواہر دلسوزم

و استاد کراتقدرم

لاد ل

## فهرست محتویات

فصل اول: معرفی	۱
طرح مشکل	۱
پس زمینه و ضرورت	۲
هدف مونوگراف	۳
میتودولوژی	۳
محدودیت‌ها	۵
فصل دوم: میتودولوژی	۶
سیستم دیتابیس MySQL	۸
برنامه MySQL WORKBENCH	۹
زبان برنامه‌نویسی PHP	۱۰
زبان HTML	۱۱
زبان CSS	۱۱
زبان JAVASCRIPT	۱۳
فریم ورک JQUERY	۱۴
تکنالوژی AJAX	۱۵
فصل سوم: تجزیه و تحلیل سیستم (MODELING)	۱۷
تجزیه و تحلیل موجودیت‌های سیستم	۱۷
موجودیت‌های اصلی STRONG ENTITIES	۱۸
موجودیت‌های فرعی WEAK ENTITIES	۲۰
دیاگرام روابط موجودیت‌ها	۲۲
فصل چهارم: دیزاین فیزیکی دیتابیس	۲۵
جدول مشتریان CUSTOMER	۲۵
جدول بخش‌های جواهرات CATEGORY	۲۶
جدول انواع جواهرات TYPES	۲۶

۲۷.....	جدول اجناس ITEM
۲۸.....	جدول مدیران USERS
۲۸.....	جدول فروشات SALE
۲۹.....	جدول جزییات فروشات SALE_DETAIL
۳۰.....	جدول تحویل اجناس DELIVERY
۳۱.....	فصل پنجم: برنامه نویسی CLIENT SIDE
۳۱.....	ساخت عناصر صفحات در HTML
۳۳.....	به وجود آوردن استایل و دیزاین صفحات در CSS
۳۵.....	اعتبارسنجی و مدیریت رویدادها توسط JAVASCRIPT
۳۸.....	ایجاد انیمیشن و گالری تصاویر در JQUERY
۴۰.....	درخواست های ناهمزمان و تکنالوژی WEB 2 توسط AJAX
۴۴.....	فصل ششم: برنامه نویسی SERVER SIDE
۴۴.....	ارتباط با دیتابیس DATABASE CONNECTIVITY
۴۵.....	ارسال QUERY به دیتابیس
۴۸.....	دریافت معلومات از دیتابیس
۴۹.....	مدیریت ورود استفاده کنندگان
۵۱.....	امنیت سیستم
۵۳.....	فصل هفتم: جمع بندی و نتیجه گیری
۵۴.....	منابع

## فهرست تصاویر

شکل ۱: برنامه MySQL Workbench	۹
شکل ۲: users Strong Entity	۱۸
شکل ۳: customer Strong Entity	۱۹
شکل ۴: category Strong Entity	۱۹
شکل ۵: types Strong Entity	۱۹
شکل ۶: sale Weak Entity	۲۰
شکل ۷: item Weak Entity	۲۰
شکل ۸: sale_detail Weak Entity	۲۱
شکل ۹: delivery Weak Entity	۲۱
شکل ۱۰: دیاگرام روابط موجودیت ها (ERD)	۲۲
شکل ۱۱: جدول customer	۲۶
شکل ۱۲: جدول category	۲۶
شکل ۱۳: جدول types	۲۷
شکل ۱۴: جدول item	۲۷
شکل ۱۵: جدول users	۲۸
شکل ۱۶: جدول sale	۲۹
شکل ۱۷: جدول sale_detail	۲۹
شکل ۱۸: جدول delivery	۳۰
شکل ۲۲: مقایسه مدل وب کلاسیک و Web 2	۴۱



## فصل اول: معرفی

مونوگراف «سیستم فروشات آنلاین جواهرفروشی» یک وبسایت دینامیک آنلاین میباشد که برای معرفی و به نمایش قرار دادن جواهرات و زیورآلات اصیل افغانستان به هموطنان عزیزمان و همچنان مردم سراسر دنیا میباشد.

در این وبسایت امکان خریداری آنلاین جواهرات از طریق سیستم Paypal وجود خواهد داشت. در این صورت مشتریانی که دارای Master Card باشند، میتوانند مستقیماً و به آسانی، بدون نیاز به مراجعه فیزیکی، اجناس مورد نظرشان را خریداری نمایند.

طرز فعالیت این سیستم طوری میباشد که هر جواهر فروشی (انتیک) میتواند بعد از ورود (Login) به قسمت مدیریت سیستم، اجناس خویش را با مشخصات کامل آن، تصویر جنس و قیمت فروش در وبسایت قرار بدهد.

همچنان امکانات لازم دیگر برای مدیریت اجناس مانند: تغییر دادن مشخصات جنس، حذف نمودن، اضافه نمودن تصویر، جستجو و ... نیز در بخش مدیریت سیستم قابل دسترس خواهد بود.

مشتریان و بازدیدکنندگان میتوانند از هر جا و از هر مکانی (Anytime, Anywhere) وبسایت جواهرفروشی را باز کرده و جواهرات مختلف را به همراه مشخصات، تصویر و قیمت آن ببینند، جستجو کنند و در صورت تمایل، از طریق سیستم پرداخت آنلاین Paypal خریداری نمایند.

## طرح مشکل

افغانستان، میهن عزیز ما دارای جواهرات و زیورآلات بسیار زیبا و منحصر به فردی مانند: زمرد، لعل، لاجورد و ... میباشد که متأسفانه به شکل درست به جهانیان شناسانده نشده است. و همین جواهرات بسیار با ارزش به شکل غیر قانونی از افغانستان خارج شده و به نام کشورهای دیگر در مارکیت جهانی به فروش میرسد. در حالی که اگر جواهرات کشور عزیزمان به جهانیان معرفی شود، نه تنها این سرمایه گرانبها به نام افغانستان شناخته میشود، بلکه منبع بسیار خوبی برای کسب درآمد و رشد اقتصاد و توسعه صنعت توریسم میشود.

همچنین هموطنان عزیز خودمان در داخل کشور، برای خریداری جواهرات مجبور هستند به جواهر فروشی‌ها به شکل حضوری و فیزیکی مراجعه کرده و زیورآلات دلخواه خویش را خریداری نمایند که در این صورت باعث اتلاف وقت و هدر رفتن انرژی‌شان میشود.

همچنان هموطنان عزیز ما مجبور هستند که پول نقد با خود همراه داشته باشند تا بتوانند هزینه خریداری جواهرات را پرداخت نمایند. اما در صورتی که فروش جواهرات به شکل آنلاین و همچنان پرداخت پول نیز به شکل آنلاین صورت بگیرد، خطر و ریسک به همراه داشتن پول نقد از بین میرود.

از طرف دیگر تنوع در بازار جواهرات و زیورآلات بسیار زیاد میباشد. در نتیجه اگر مشتری بخواهد شخصا از فروشگاه‌های مختلف یکی پس از دیگری بازدید نماید و انواع و مدل‌های مختلف را ببیند، باعث گیج شدن و سرگردانی مشتری خواهد شد.

## پس زمینه و ضرورت

از آنجایی که مشکلات مطرح شده در قسمت قبل مربوط به تمام جواهرفروشی‌ها میباشد و محدود به یک یا چند فروشگاه نمیشود، اشخاص و شرکت‌های دیگر اقدام به ساختن وبسایت‌هایی برای اعلانات و فروش جواهرات کرده‌اند.

اما معمولاً این وبسایت‌ها موضوع فروش را به طور عمده دنبال کرده‌اند. یعنی انواع اجناس مختلف از اجناس الکتریکی و الکترونیکی گرفته، تا خانه و زمین و موتر را برای فروش قرار داده‌اند و به طور تخصصی در زمینه فروش جواهرات و زیورآلات تلاش نکرده‌اند.

همچنین اکثر این وبسایت‌ها فقط به اعلانات و معرفی اجناس پرداخته‌اند و راه حلی برای فروش و پرداخت هزینه آن در نظر نگرفته‌اند و از این لحاظ میتوان گفت که فقط به شکل تبلیغاتی فعالیت می‌کنند.

بنا بر مشکلاتی که ذکر گردید، ضرورت احساس می‌شود که یک وبسایت امن و پیشرفته که به طور تخصصی در زمینه فروش جواهرات و زیورآلات فعالیت کند، ساخته شود تا هم جهانیان با جواهرات اصیل و با ارزش افغانستان آشنایی پیدا کنند و هم زمینه فروش و پرداخت آنلاین برای مشتریان فراهم گردد.

## هدف مونوگراف

هدف مونوگراف «سیستم فروشات آنلاین جواهرفروشی»، تجزیه، تحلیل، طراحی و دیزاین یک وبسایت دینامیک آنلاین میباشد که بتواند یک روش بسیار ساده، سریع و با سهولت را برای خریداری جواهرات و زیورآلات فراهم کرده و زمینه معامله بین فروشندگان و خریدار را به وجود بیاورد.

این هدف از طریق دیزاین یک وبسایت اینترنتی و لینک نمودن آن با سیستم پرداخت آنلاین Paypal، قابل دسترسی و تحقق می‌باشد. طوری که مالک وبسایت (فروشنده) بتواند جواهرات مورد نظر خویش را با مشخصات کامل، تصویر و قیمت آن در وبسایت قرار بدهد و بازدیدکنندگان (خریداران) بتوانند جواهرات مختلف را مشاهده نموده و یا مطابق میل و بودجه خود بین جواهرات جستجو نمایند.

در صورتی که مشتری جواهر یا زیورآلات دلخواه خویش را پیدا نماید، میتواند مستقیماً اقدام به خریداری آنلاین نماید. اما در صورتی که مشتری نتواند جواهر یا زیورآلات مورد نظر خویش را پیدا نماید، میتواند از طریق فورم Contact با مالک وبسایت ارتباط برقرار کرده و درخواست خویش را مبنی بر تقاضای جواهر مورد نظر ارسال نماید.

از آنجایی که این مونوگراف نه تنها فروش جواهرات را در افغانستان، بلکه در سراسر جهان هدف قرار داده است، این هدف از طریق به وجود آوردن یک وبسایت جهانی (آنلاین) قابل تحقق و دستیابی می‌باشد.

هدف جانبی دیگری که در این مونوگراف تعقیب می‌شود، چنانچه در قسمت بعدی (میتودولوژی) بیشتر بحث خواهد شد، ترویج فرهنگ استفاده از تکنالوژی‌ها و نرم‌افزارهای Open Source می‌باشد. برای دستیابی به این هدف، تمام نرم‌افزارها و تکنالوژی‌های به کار رفته در این مونوگراف Open Source خواهند بود.

## میتودولوژی

موضوع مونوگراف «سیستم فروشات آنلاین جواهرفروشی» در حقیقت یک دیتابیس اینترنتی یا Web-based Database می‌باشد که جهت طراحی و دیزاین آن نیاز به یک سیستم دیتابیس (RDBMS) مانند MySQL، SQL Server، Oracle و همچنین به یک پلتفرم برنامه‌نویسی تحت وب مانند JSP، ASP.NET و یا PHP نیاز می‌باشد.

همچنین برای نوشتن کدهای مربوطه، نیاز به یک برنامه Editor نیز می‌باشد. برنامه‌ای که بتواند زبان‌های مختلف برنامه‌نویسی تحت وب را به طور یکجایی پشتیبانی کند و ترجیحاً Syntax هر زبان را نیز تشخیص بدهد.

در این مونوگراف سعی شده است که نرم افزارها و تکنالوژی های Open Source استفاده شود. استفاده از برنامه های Open Source برای هر کشور فواید بسیار زیادی دارد که شاید مهم ترین آن ها از نگاه اقتصادی باشد. از آن جایی که برنامه های Open Source نیاز به خریداری ندارند، بنابراین هزینه بسیار زیادی که برای تهیه برنامه های لازم باید صرف گردد، می تواند صرفه جویی شود.

در همین راستا و برای نهادینه ساختن فرهنگ استفاده از برنامه های Open Source طوری که ذکر شد، تمام نرم افزارها و تکنالوژی هایی که در این مونوگراف استفاده شده اند، Open Source می باشند.

نرم افزارها و تکنالوژی هایی که در ساخت این مونوگراف به کار رفته اند را میتوان به دو بخش Server Side و Client Side تقسیم بندی نمود که هر کدام به ترتیب عبارتند از:

نرم افزارها و زبان های Server Side:

– سیستم دیتابیس MySQL

– نرم افزار MySQL Workbench

– زبان برنامه نویسی PHP

نرم افزارها و زبان های Client Side:

– HTML 5

– CSS 3

– JavaScript

– jQuery

– AJAX

– Bootstrap

همچنین جهت نوشتن کودها و اسکریپت های لازم، از برنامه Notepad++ که آن هم یک ادیتر Open Source می باشد، استفاده خواهد شد.

## محدودیت‌ها

در قسمت توسعه و دیزاین مونوگراف «سیستم فروشات آنلاین جواهرفروشی» مشکل و محدودیت اساسی، طریقه خریداری کردن آنلاین جواهرات میباشد. از آن جایی که تا هنوز فرهنگ خریداری و تجارت آنلاین در کشور عزیزمان افغانستان به میزان زیادی ترویج پیدا نکرده است، همه اشخاص دارای امکانات لازم برای خریداری آنلاین نمی‌باشند.

این بدان معنی است که همه اشخاص و بازدیدکنندگان وبسایت، دارای Master Card و یا کارت‌های مشابه نمی‌باشند و زمینه خریداری مستقیم و آنلاین برای همه مشتریان فراهم نمی‌باشد.

برای برطرف کردن این محدودیت میتوان امکانات Pay On Delivery را در سیستم به وجود آورد. یعنی اگر مشتری دارای Master Card نمی‌باشد و نمیتواند به طور آنلاین جنس مورد نظر خویش را خریداری نماید، میتواند برای خود یک Account ایجاد کرده و خودش را register نماید. در فورم Registration بر علاوه نام و مشخصات اولیه مشتری، شماره تلفون و آدرس وی ضروری می‌باشد.

در این صورت مشتری بعد از ثبت نام کردن در وبسایت، میتواند جنس دلخواه خویش را فرمایش بدهد. جنس فرمایش داده شده، به آدرس مشتری ارسال میشود و مشتری بعد از تحویل گرفتن جنس میتواند قیمت آن را پرداخت نماید.

با وجود داشتن این سهولت یعنی Pay On Delivery، تمام مشتریان و بازدیدکنندگان وبسایت میتوانند جواهرات و زیورآلات مورد نظر و متناسب سلیقه خویش را در وبسایت پیدا نموده و خریداری نمایند و پول آن را از طریق آنلاین و یا از طریق تحویل در خانه پرداخت نمایند.

## فصل دوم: میتودولوژی

موضوع مونوگراف «سیستم فروشات آنلاین جواهرفروشی» در حقیقت یک دیتابیس اینترنتی یا Web-based Database می باشد که جهت طراحی و دیزاین آن نیاز به یک سیستم دیتابیس (RDBMS) مانند: MySQL، SQL Server، Oracle و همچنین به یک پلتفرم برنامه نویسی تحت وب مانند: JSP، ASP.NET و یا PHP نیاز می باشد.

همچنین برای نوشتن کدهای مربوطه، نیاز به یک برنامه Editor نیز می باشد. برنامه ای که بتواند زبان های مختلف برنامه نویسی تحت وب را به طور یکجایی پشتیبانی کند و ترجیحا Syntax هر زبان را نیز تشخیص بدهد.

در این مونوگراف سعی شده است که تماما از برنامه های Open Source استفاده گردد. برنامه های Open Source برنامه هایی می باشند که به طور رایگان در اختیار تمام استفاده کنندگان قرار می گیرد و نیازی به خریداری لایسنس آن نمی باشد. همچنین از آن جایی که کد منبع برنامه (Source Code) در دسترس شخص استفاده کننده قرار می گیرد، استفاده کننده می تواند برنامه را مطابق نیازها و ضرورت های خود تغییر داده و حتی مجددا منتشر (Redistribute) نماید. و تمام این موضوعات بدون هیچ مشکل قانونی در قسمت Copyright و با آزادی کامل انجام می شود.

استفاده از برنامه های Open Source برای هر کشور فواید بسیار زیادی دارد که شاید مهم ترین آن از نگاه اقتصادی باشد. همان طور که بیان شد، برنامه های Open Source نیاز به خریداری ندارند و بنابراین هزینه بسیار زیادی که برای تهیه برنامه های لازم در یک نهاد باید صرف گردد، می تواند صرفه جویی شود.

تنها مشکلی که استفاده کنندگان برنامه های Open Source با آن مواجه می باشند (البته در صورتی که خودشان متخصص یا Expert نباشند) موضوع پشتیبانی (Support) می باشد. چرا که مرجع خاصی در قبال برنامه Open Source مسئول نمی باشد و مسئولیت استفاده از برنامه به دوش خود استفاده کننده می باشد.

البته این مشکل نیز از طریق آموزش و بالا بردن ظرفیت های مسکلی قابل حل می باشد. همچنین با به وجود آمدن مراکز تخصصی برای کمک و پشتیبانی از استفاده کنندگان برنامه های Open Source، این مشکل نیز قابل برطرف شدن می باشد.

در همین راستا و برای نهادهای ساختن فرهنگ استفاده از برنامه‌های Open Source طوری که تذکر داده شد، تمام برنامه‌هایی که در این مونوگراف استفاده شده‌اند، Open Source می‌باشند.

برنامه‌ها و زبان‌هایی که در ساخت این مونوگراف به کار رفته‌اند را میتوان به دو بخش Server Side و Client Side تقسیم‌بندی نمود که هر کدام به ترتیب عبارتند از:

برنامه‌ها و زبان‌های Server Side:

– دیتابیس MySQL v5.6

– برنامه MySQL Workbench 5.2 OSS

– زبان برنامه‌نویسی PHP v5.4

برنامه‌ها و زبان‌های Client Side:

– زبان HTML 5

– زبان CSS 3

– زبان JavaScript

– فریم‌ورک jQuery

– تکنالوژی AJAX

– فریم‌ورک Bootstrap

همچنین جهت نوشتن کودها و اسکریپت‌های لازم، از برنامه Notepad++ که آن هم یک ادیتر Open Source می‌باشد، استفاده شده است.

حال به معرفی و بررسی کوتاه هر یک از این زبان‌ها و برنامه‌های متذکره می‌پردازیم:

## سیستم دیتابیس MySQL

هسته دیتابیس (Core Database) این مونوگراف MySQL می باشد. این سیستم دیتابیس در ۲۳ می ۱۹۹۵ توسط کمپنی به نام MySQL AB که یک کمپنی سوئدنی بود، به وجود آمد و به زبان C++ نوشته شده است.

MySQL قابل اجرا و استفاده در تقریباً تمام سیستم عامل های (Operating System) مشهور دنیا، مانند Windows، Linux، UNIX و ... می باشد. یا به اصلاح دیگر، این سیستم دیتابیس Cross Platform است.

کمپنی MySQL AB در سال ۲۰۰۸ توسط کمپنی سان میکروسستمز (Sun Microsystems) خریداری شده و سپس در ۲۷ جنوری ۲۰۱۰ کمپنی Sun توسط کمپنی Oracle خریداری شد. بنابراین فعلاً مسئولیت انتشار و توسعه این سیستم دیتابیس در اختیار کمپنی Oracle می باشد.

از جمله کمپنی ها و وبسایت های مشهوری که از سیستم دیتابیس MySQL استفاده می کنند، میتوان به Facebook، Wikipedia، LinkedIn، Nokia، Flickr و ... اشاره نمود!

### فواید استفاده از سیستم دیتابیس MySQL:

– مقیاس پذیری و قابلیت انعطاف (Scalability and Flexibility)

– عملکرد بالا (High Performance)

– در دسترس بودن بالا (High Availability)

– پشتیبانی از تراکنش ها (Transaction Support)

– محافظت از دیتا (Data Encryption)

– آسان بودن مدیریت (Ease of Management)

– آزاد بودن برنامه (Open Source)

لینک دریافت دیتابیس MySQL:

<http://www.dev.mysql.com/downloads>



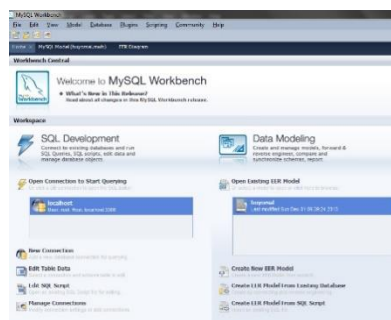
## برنامه MySQL Workbench

این برنامه از سه بخش اساسی تشکیل شده است که عبارتند از:

۱- SQL Development

۲- Data Modeling

۳- Server Administration



شکل ۱: برنامه MySQL Workbench

از طریق بخش اول این برنامه (SQL Development) میتوان به یک دیتابیس از قبل ساخته شده در سیستم دیتابیس MySQL وصل گردیده و فرمان‌های دیتابیس (SQL Query) را اجرا نمود. همچنین به کمک این قسمت از برنامه میتوان SQL Script را نیز اجرا نموده و حتی میتوان معلومات ذخیره شده را مورد دسترسی قرار داد و یا اشیا مختلف دیتابیس را مدیریت نمود.

از طریق بخش دوم برنامه (Data Modeling) میتوان به ساختن و مدیریت Model دیتابیس اقدام نموده و دیاگرام رابطه موجودیت‌ها (Entity Relationship Diagram) ERD را طراحی و دیزاین نمود.

از طریق بخش سوم برنامه Server Administration نیز میتوان به عیارسازی سیستم دیتابیس MySQL پرداخته و همچنین User Account ساخته و یا وضعیت دیتابیس را مشاهده نمود. همچنین میتوان با کمک این بخش اقدام به Export و Import دیتابیس نیز نمود.

لینک دریافت برنامه MySQL Workbench:

<http://wb.mysql.com/workbench>

## زبان برنامه‌نویسی PHP

این زبان که Server Side می‌باشد توسط شخصی به نام راسموس لردورف Rasmus Lerdorf در سال ۱۹۹۵ به وجود آمد. این زبان نیز Open Source بوده و یکی از محبوب‌ترین پلتفرم‌ها و زبان‌های برنامه‌نویسی تحت وب در دنیا می‌باشد. چنان‌که گوگل چندی پیش اعلام کرد که بیش از ۸۰٪ وب سایت‌های دنیا از PHP استفاده می‌کنند.

نمونه‌ای از کد نوشته شده به زبان PHP:

```
<?php
if(!isset($_SESSION)) {
    session_start();
}

if(isset($_SESSION["login"])) {
    unset($_SESSION["login"]);
    $_SESSION["create"] = NULL;
}

?>
```

سه مورد از خواص و فواید زبان PHP که باعث شهرت و محبوبیت بسیار زیاد آن شده است عبارتند از:

– قابلیت کار کردن در پلتفرم‌های مختلف مانند: ویندوز، لینوکس، یونیکس و... (Cross Platform)

– منبع باز و رایگان بودن (Open Source)

– قابلیت ترکیب بسیار بالا و انعطاف پذیری زیاد با کدهای زبان HTML (HTML Embedded)

از آن جایی که زبان PHP یک زبان Server Side می‌باشد، برای اجرا نمودن کدهای نوشته شده به این زبان باید به یک Web Server مانند Apache و یا IIS دسترسی داشت. همچنین باید PHP Engine (یا PHP Parser) را که یک Software می‌باشد در کنار Web Server نصب نمود تا بتوان نتیجه اجرای کدهای PHP را مشاهده کرد.

لینک دریافت PHP:

<http://www.php.net>

## زبان HTML

زبان HTML توسط تیم برنرزی Tim Berners-Lee رییس فعلی سازمان جهانی وب W3C در دهه ۹۰ میلادی به وجود آمد. این زبان که Syntax آن بسیار ساده و آسان می باشد، در حقیقت یک Markup Language یا زبان نشانه گذاری است و امکانات برنامه نویسی ندارد.

عناصر یک صفحه اینترنتی مانند جدول، پاراگراف، تصویر، فورم و ... توسط کدهای این زبان و با استفاده از علامت های < و > به وجود می آیند که به عنوان Tag یاد می شود. به طور مثال برای ساختن یک جدول که دارای دو Column و دو Row باشد چنین کدهایی نوشته می شود:

```
<table>
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
  </tr>
  <tr>
    <td>Cell 3</td>
    <td>Cell 4</td>
  </tr>
</table>
```

زبان HTML اولین زبانی بود که برای ساختن صفحات اینترنتی و در نتیجه وب سایت ها به وجود آمد و با گذشتن سال های زیاد، اکنون این زبان به ورژن پنجم خود رسیده است که تکنالوژی های بسیار جدید و مفیدی را مانند: پشتیبانی مستقیم از صدا و فیلم، تشخیص موقعیت جغرافیایی و... معرفی نموده است.

## زبان CSS

CSS زبانی می باشد که برای چگونگی نشان دادن (ظاهر و فورمت) اجزای تشکیل دهنده یک متن یا یک صفحه اینترنتی استفاده می شود. از این زبان برای دیزاین و استایل دادن به صفحات وب سایت هایی که بر پایه HTML و XHTML ساخته شده اند، استفاده می شود.

اساساً زبان CSS برای جدا کردن تگ‌های HTML از فورمت ظاهری آن‌ها به وجود آمده است. فورمت‌های ظاهری می‌توانند شامل رنگ، فونت، موقعیت و ظاهر هر عنصر در صفحه باشند. این تفکیک و جداسازی خوانایی کدهای HTML را افزایش داده است و نیز کنترل بیشتری را بر روی مشخصه‌های ظاهری کدها و مستندات به وجود آورده است.

توسط زبان CSS می‌توان فورمت و ظاهر چندین صفحه را به طور یک‌جایی تغییر داد و نیازی به دوباره نوشتن همان کدها برای صفحات مشابه نمی‌باشد.

نمونه‌ای از کدهای نوشته شده به زبان CSS:

```
div#category {
    clear: none;
    float: left;
    margin-left: 0;
    width: 100%;
    display: block;
    font-size:14px;
}
div#category ul li {
    margin-top:10px;
    float:none !important;
}
div#visits {
    clear: none;
    float: left;
    margin-left: 0;
    width: 100%;
    display: block;
    font-size:14px;
}
```

## زبان JavaScript

این زبان که Client Side می‌باشد و از امکانات برنامه‌نویسی برخوردار است، در سال ۱۹۹۵ توسط شخصی به نام برندن ایک Brendan Eich که در آن زمان برای کمپنی Netscape کار می‌کرد، به وجود آمد.

زبان JavaScript نقش بسیار مهمی در وبسایت‌ها دارد و از آن جایی که تنها زبان برنامه‌نویسی Client Side می‌باشد، میتوان گفت استفاده از آن در هر وبسایتی اجتناب ناپذیر است.

توسط این زبان میتوان خواص و امکانات Dynamic را به زبان HTML که فوق‌العاده Static می‌باشد اضافه نمود و صفحات اینترنتی بر مبنای DHTML را به وجود آورد.

زبان JavaScript نه تنها به خودی خود در وبسایت‌ها مورد استعمال بسیار فراوانی دارد، بلکه امروزه با به میان آمدن کتابخانه‌ها و فریم‌ورک‌هایی مانند jQuery استفاده از آن در صفحات اینترنتی بیشتر و بیشتر شده است.

همچنین طوری که در ادامه بیان می‌شود، تکنالوژی AJAX نیز مبتنی بر همین زبان است و توسط کدهای نوشته شده به این زبان، پیاده‌سازی و تطبیق می‌گردد.

لازم به ذکر است که این زبان و دو زبان قبلی که به آن اشاره گردید (HTML و CSS) هر سه نیازی به نصب نمودن ندارند و کد نوشته شده به این زبان‌ها از طریق هر Browser مانند Chrome، Firefox، IE و... قابل اجرا می‌باشد. به همین دلیل است که به این زبان‌ها Client Side گفته می‌شود.

نمونه‌ای از کد نوشته شده به زبان JavaScript:

```
function jQuery (window, undefined) {
var document = window.document,
    navigator = window.navigator,
    location = window.location;
var jQuery = (function() {
var jQuery = function (selector, context) {
return new jQuery.fn.init( selector, context, rootjQuery );
}
}
```

## فریم ورک jQuery

جی کوئری jQuery یک کتابخانه جاوا اسکریپت با اندازه کوچک (تقریباً 70KB) و قابل استفاده در تمام براوزرها می باشد که برای ساده کردن نوشتن اسکریپت های Client Side در HTML طراحی شده است.

جی کوئری که امروزه محبوب ترین کتابخانه جاوا اسکریپت در حال حاضر می باشد، توسط شخصی به نام John Resig به وجود آمده است. جی کوئری Open Source و رایگان بوده و تحت دو لایسنس GPL و MIT منتشر می شود.

با استفاده از کتابخانه jQuery می توان حرکات انیمیشن ایجاد کرد و رویدادهای صفحه را کنترل نموده و به وسیله آن حتی می توان برنامه های مبتنی بر AJAX را ایجاد و توسعه داد.

جی کوئری همچنین این اختیار را به برنامه نویسان می دهد تا Plugin هایی برای کتابخانه جاوا اسکریپت ایجاد کنند که به کارگیری همه این امکانات کمک می کند صفحات وب قدرتمند و داینامیک داشته باشیم.

نمونه ای از کود نوشته شده در jQuery:

```
$(document).ready(function({

    $(".delete").click(function(){

        var element = $(this);

        var del_id = element.attr("id");

        var info = 'id=' + del_id;

        if(confirm("Are you sure?")) {

            $.ajax({ type: "GET", url: "delete.php", data:
            info, success: function(){ } });

            $(this).parents(".record").css({ backgroundColor:
            "red" }, "fast").animate({ opacity: "hide" }, "slow"); }

            return false;

        });

    });
```

لینک دریافت jQuery:

<http://jquery.com>

## تکنالوژی AJAX

واژه Ajax را برای اولین بار شخصی به نام Jesse James Garrett در فبروری سال ۲۰۰۵ در مقاله‌ای تحت عنوان Ajax: A New Approach to Web Applications استفاده کرد. و از همین جا بود که یک تحول بسیار بزرگ در دنیای وب اتفاق افتاد و Web 2 به میان آمد.

به طور بسیار خلاصه، تکنالوژی و یا تکنیک AJAX کمک می‌کند تا فقط قسمتی از صفحه اینترنتی که نیاز به تغییر دارد، تغییر کند و احتیاجی نباشد که تمام صفحه Reload شود.

همین پیشرفت به ظاهر کوچک باعث شد که Web Application ها بیشتر از پیش به Desktop Application ها نزدیک شوند و باعث شد تا وبسایت‌ها بتوانند عکس‌العمل‌های سریع‌تر به رفتارهای یوزر از خود نشان بدهند. و دیگر نیازی نیست که زمان‌های زیادی در پیش چشم استفاده‌کننده فقط صفحه سفید قرار بگیرد!

تکنالوژی یا تکنیک AJAX (Asynchronous JavaScript And XML) همان طور که از نام آن پیداست، یک زبان برنامه‌نویسی یا برنامه کمپیوتری نمی‌باشد. بلکه فقط روشی برای نوشتن کودها به زبان JavaScript و تبادل اطلاعات با Server به روش ناهمزمان یا Asynchronous می‌باشد.

به همین دلیل، برای استفاده و اجرا کردن AJAX باز هم نیاز به نصب کردن هیچ گونه برنامه نمی‌باشد و فقط با نوشتن کودهای AJAX در زبان JavaScript میتوان از این تکنالوژی استفاده نمود.

نمونه‌ای از کود نوشته شده AJAX به زبان JavaScript:

```
function loadXMLDoc() {
    var xmlhttp;
    if (window.XMLHttpRequest) {
        xmlhttp=new XMLHttpRequest();
    } else {
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function() {
        if(xmlhttp.readyState==4 && xmlhttp.status==200) {
            document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
        }
    }
}
```

```
    }  
    xmlhttp.open("GET","ajax_info.txt",true);  
    xmlhttp.send();  
}
```



## فصل سوم: تجزیه و تحلیل سیستم (Modeling)

### تجزیه و تحلیل موجودیت‌های سیستم

در طراحی و دیزاین سیستم فروشات آنلاین جواهرفروشی چهار موجودیت اصلی (Strong Entity) وجود دارد:

۱- مدیران Users

۲- مشتریان Customers

۳- بخش‌ها Categories

۴- انواع جواهرات Types

موجودیت مشتریان دارای یک موجودیت فرعی (Weak Entity) می‌باشد که عبارت است از:

۱- فروشات Sale

موجودیت فرعی (Weak Entity) موجودیت‌های Category و Types به طور مشترک عبارت است از:

۱- اجناس Items

موجودیت اصلی (Strong Entity) مدیران یا Users یک موجودیت Stand Alone میباشد که با هیچ یک از

موجودیت‌های دیگر رابطه برقرار نمی‌کند.

## موجودیت‌های اصلی Strong Entities

حال به تشریح هر کدام از موجودیت‌های اصلی می‌پردازیم:

### ۱- Users

این موجودیت اصلی برای ذخیره نمودن مشخصات مدیران سیستم در نظر گرفته شده است. در حقیقت این موجودیت Profile مدیران را نیز تشکیل می‌دهد. همچنین از همین موجودیت برای پروسه Authentication نیز استفاده صورت می‌گیرد. بدین معنی که Credential هر مدیر که عبارت از Username و Password وی می‌باشد، در همین موجودیت ذخیره می‌گردد.

users
user_id INT(11)
firstname VARCHAR(32)
lastname VARCHAR(32)
gender TINYINT(1)
photo VARCHAR(128)
phone VARCHAR(16)
email VARCHAR(128)
address VARCHAR(255)
username VARCHAR(32)
password VARCHAR(64)
user_level TINYINT(1)

شکل ۲: users Strong Entity

### ۲- Customers

این موجودیت برای ذخیره کردن مشخصات مشتریان و خریداران می‌باشد. شماره تلفون و آدرس ایمیل مشتری برای برقراری تماس و ارتباط ذخیره میشود و همچنین آدرس فیزیکی هر مشتری نیز برای انتقال جنس به درب خانه و سهولت Pay-On-Delivery می‌باشد.

customer
customer_id INT(11)
customer_name VARCHAR(64)
phone VARCHAR(16)
email VARCHAR(128)
address VARCHAR(255)
gender TINYINT(1)
username VARCHAR(32)
password VARCHAR(64)
verified TINYINT(1)

شکل ۳: customer Strong Entity

### Category \_۳

از این موجودیت اصلی برای ذخیره کردن انواع بخش‌های جواهرات استفاده میشود. مانند: گردنبند، گوشواره، انگشتر و ... تا در آینده هم مشتریان بتوانند به آسانی جواهرات را جستجو کنند و هم مدیران سیستم بتوانند اجناس را با سهولت بیشتر مدیریت کنند.

category
category_id INT(11)
category_name VARCHAR(64)

شکل ۴: category Strong Entity

### Types \_۴

در این موجودیت اصلی، انواع جواهرات و زیورآلات از نظر جنسیت‌شان مانند: الماس، طلا، نقره و ... ذخیره میشود. این موجودیت نیز به مشتریان کمک میکند تا بهتر بتوانند جواهرات مورد نظرشان را پیدا کنند و از طرف دیگر به مدیران نیز کمک میکند تا اجناس را بهتر بتوانند مدیریت کنند و گزارش‌های دقیق و مفیدی از سیستم به دست بیاورند.

types
type_id INT(11)
type_name VARCHAR(64)

شکل ۵: types Strong Entity

## موجودیت‌های فرعی Weak Entities

حال به تشریح هر کدام از موجودیت‌های فرعی می‌پردازیم:

### موجودیت‌های فرعی موجودیت Customer:

۱-۱: Sale

این موجودیت فرعی برای ذخیره کردن فروشات جواهرات میباشد. شماره بل، شماره مشتری و تاریخ فروش که از جمله مهمترین معلومات در مورد فروشات است، در این موجودیت ذخیره میشود.

sale
bill_id INT(11)
customer_id INT(11) (FK)
sale_date DATE

شکل ۶: sale Weak Entity

### موجودیت‌های فرعی موجودیت‌های Category و Types:

۱-۲: Item

در این موجودیت، معلومات درباره جواهرات مختلف ذخیره میشود. مانند: نام جواهر، قیمت، عکس، ساخت کدام کشور، وزن و...

item
item_id INT(11)
item_name VARCHAR(128)
price INT(11)
picture VARCHAR(128)
madein VARCHAR(64)
weight FLOAT
type_id INT(11) (FK)
category_id INT(11) (FK)

شکل ۷: item Weak Entity

**موجودیت‌های فرعی موجودیت Sale:****۳-۱: Sale\_Detail**

در این موجودیت جزییات مربوط به فروشات ذخیره میشود. این که در یک بل کدام اجناس فروخته شده است، چه تعداد و به چه قیمتی فروخته شده است.

sale_detail
detail_id INT(11)
bill_id INT(11) (FK)
item_id INT(11) (FK)
quantity INT(11)
unitprice INT(11)
totalprice INT(11)

شکل ۸: sale\_detail Weak Entity

**۳-۲: Delivery**

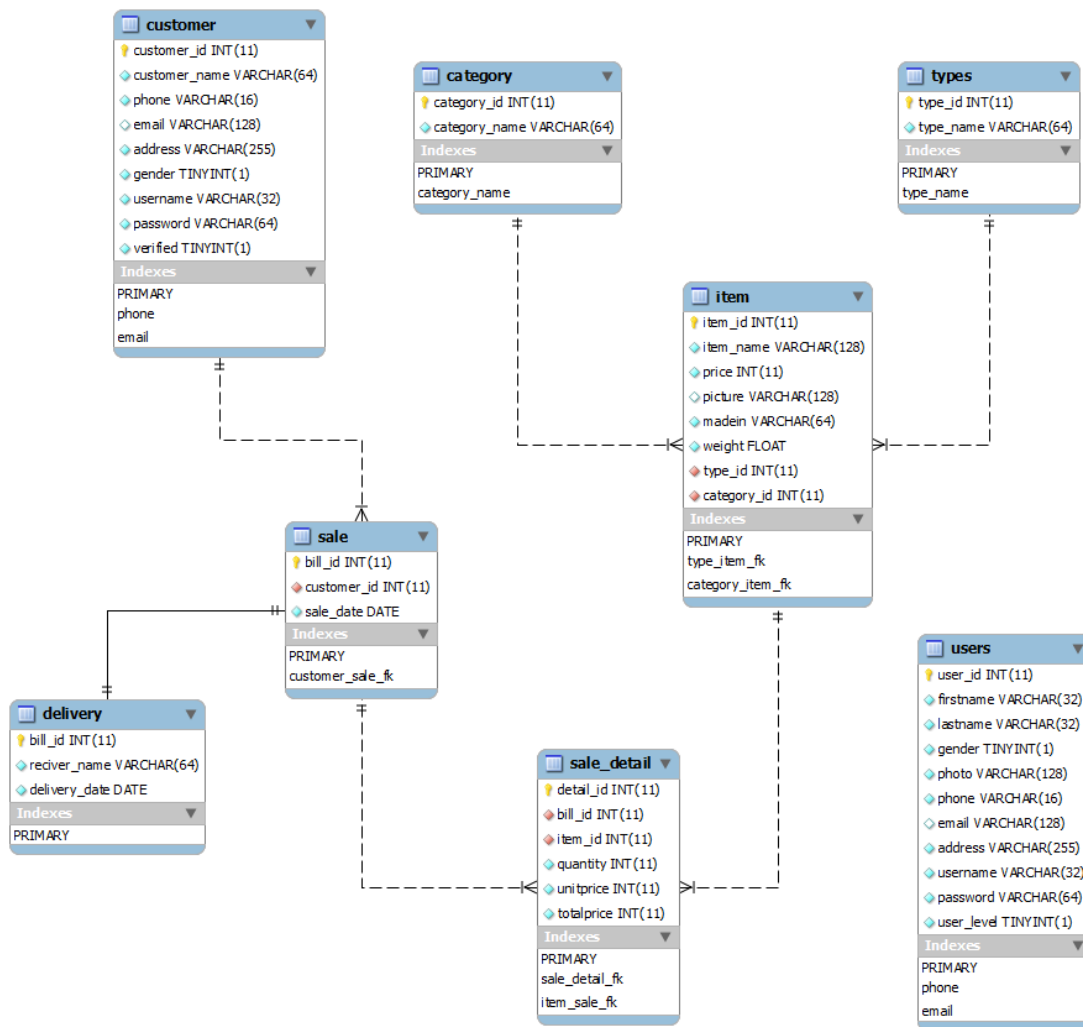
از این موجودیت برای اطمینان حاصل کردن از تحویل اجناس فروخته شده به مشتری، استفاده میشود. طوری که نام شخص تحویل گیرنده و تاریخ آن در سیستم ذخیره میشود. به این ترتیب مدیران میتوانند تشخیص بدهند که کدام بل‌ها به مشتری تحویل داده شده است و کدام بل‌ها تا هنوز تحویل داده نشده است.

delivery
bill_id INT(11) (FK)
reciver_name VARCHAR(64)
delivery_date DATE

شکل ۹: delivery Weak Entity

## دیاگرام روابط موجودیت‌ها

در این قسمت برای نمایش و تشریح روابط موجود بین Entityهای مختلف، ERD یا Entity Relationship Diagram «سیستم فروشات آنلاین جواهرفروشی» دیزاین شده است که آن را بررسی و مطالعه می‌کنیم.



شکل ۱۰: دیاگرام روابط موجودیت‌ها (ERD)

همان طور که در قسمت قبل نیز بیان شد، چهار موجودیت اصلی (Users, Customer, Category, Types) در این سیستم وجود دارند که در دیاگرام دیده می‌شوند و بیشترین ارتباطات (Relationship) مربوط به موجودیت Sale می‌شود.

موجودیت اصلی customer دارای یک موجودیت فرعی یعنی موجودیت sale می‌باشد و موجودیت‌های اصلی category و types به طور مشترک دارای یک موجودیت فرعی یعنی item می‌باشد. همچنین موجودیت users مستقل بوده و با دیگر موجودیت‌ها رابطه برقرار نمی‌کند.

### روابط موجودیت customer:

رابطه بین customer و sale از نوع 1:N و Mandatory می‌باشد چرا که یک مشتری می‌تواند بارها خریداری نماید و مشخص بودن این که فروشات به کدام مشتری بوده است، اجباری است.

### روابط موجودیت‌های category و types:

رابطه بین category و item از نوع 1:N و Mandatory می‌باشد، چرا که در یک بخش چندین جنس (جواهر) قرار گرفته می‌تواند و مشخص بودن این که هر جواهر مربوط کدام بخش میشود، حتمی است.

رابطه بین types و item نیز از نوع 1:N و Mandatory است، چرا که چندین جواهر از یک نوع بوده می‌تواند و مشخص بودن نوعیت جواهر ضروری است.

### روابط موجودیت users:

موجودیت اصلی (Strong Entity) مدیران یعنی users، دارای هیچ رابطه‌ای با سایر موجودیت‌ها نمی‌باشد. این موجودیت مستقل بوده و فقط برای Authentication مدیران استفاده میشود.

### روابط موجودیت item:

رابطه بین item و sale\_detail از نوع 1:N و Mandatory می‌باشد، چرا که یک جنس (جواهر) چندین مرتبه به فروش رسیده می‌تواند و در بل فروخته شده، مشخص بودن جنس کاملاً ضروری است.

**روابط موجودیت sale:**

رابطه بین sale و sale\_detail از نوع 1:N و Mandatory می باشد، چرا که در یک مرتبه فروش، یعنی در یک بل، چندین جنس (جواهر) به فروش رسیده میتواند. و مشخص کردن این که کدام جنس (جواهر) به فروش رسیده است، حتمی میباشد.

رابطه بین sale و delivery از نوع 1:1 و Mandatory میباشد، چرا که جنس فروخته شده فقط یک بار به مشتری تحویل داده میشود و در وقت تحویل دادن جنس به مشتری، مشخص بودن شماره بل کاملاً ضروری است.



## فصل چهارم: دیزاین فزیک دیتابیس

مهم‌ترین بخش یک سیستم معلوماتی، دیتابیس آن می‌باشد. در واقع تمام اعمال و اتفاقاتی که در یک Web Application انجام می‌شود بر مبنای دیتابیس آن می‌باشد و دیتابیسی که با در نظر گرفتن تمام شرایط و نیازمندی‌ها طراحی و دیزاین شود، کار برنامه‌نویسی و توسعه Interface را بسیار منظم و منسجم می‌نماید.

همان طور که در فصل اول (میتودولوژی) بیان شد، دیتابیس این مونوگراف با MySQL طراحی و ساخته شده است، که اکنون به شرح هر یک از جداول (Table) موجود در آن می‌پردازیم.

### جدول مشتریان customer

این جدول مشخصات مهم هر مشتری را در خود ذخیره می‌کند. هر مشتری دارای یک آی‌دی Unique می‌باشد که Primary Key این جدول را تشکیل می‌دهد. برای امکان برقراری تماس و تحویل جنس فروخته شده به مشتری، شماره تلفن، آدرس ایمیل و آدرس فزیک مشتری در این جدول ذخیره می‌شود که شماره تلفن و ایمیل Unique می‌باشد.

همچنین Credential هر مشتری برای Login کردن به وبسایت، در فیلدهای username و password ذخیره می‌شود. فیلد verified برای اطمینان حاصل کردن از واقعی بودن حساب (User Account) مشتری است. از آن جایی که هر شخص می‌تواند از طریق وبسایت خودش را راجستر نماید، باید راهی برای تشخیص حقیقی بودن و عدم جعلی (Fake) بودن مشتری پیدا کرد. در این حالت هر مشتری باید بعد از راجستر کردن خودش، یک مرتبه به فروشگاه به طور حضوری مراجعه کند تا حقیقی بودن وی برای ما ثابت شود تا در آینده بتوانیم اجناس فروخته شده را به درب خانه مشتری تحویل بدهیم.

customer	
customer_id	INT (11)
customer_name	VARCHAR(64)
phone	VARCHAR(16)
email	VARCHAR(128)
address	VARCHAR(255)
gender	TINYINT(1)
username	VARCHAR(32)
password	VARCHAR(64)
verified	TINYINT(1)
Indexes	
PRIMARY	
phone	
email	

شکل ۱۱: جدول customer

## جدول بخش های جواهرات category

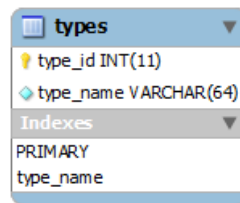
این جدول که از جمله موجودیت‌های اصلی می‌باشد، برای ذخیره کردن انواع بخش‌های جواهرات از قبیل: گردنبند، گوشواره، انگشتر و ... استفاده می‌شود. که دارای دو فیلد category\_id و category\_name می‌باشد. فیلد category\_id رول Primary Key را در این جدول بازی میکند و فیلد category\_name نیز unique می‌باشد.

category	
category_id	INT (11)
category_name	VARCHAR(64)
Indexes	
PRIMARY	
category_name	

شکل ۱۲: جدول category

## جدول انواع جواهرات types

در این جدول انواع جواهرات از نگاه جنسیت ذخیره می‌شود، مانند: الماس، طلا، نقره و ... که دارای دو فیلد type\_id و type\_name می‌باشد که فیلد type\_id کلید اصلی (Primary key) این جدول می‌باشد و فیلد type\_name غیرتکراری Unique تعیین شده است.



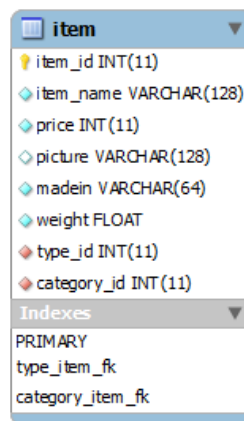
types	
type_id	INT(11)
type_name	VARCHAR(64)
Indexes	
PRIMARY	type_name

شکل ۱۳: جدول *types*

## جدول اجناس *item*

این جدول، برای ذخیره کردن جواهرات و مشخصات آنها استفاده میشود. هر جنس (جواهر) دارای یک شماره منحصر به فرد، یعنی فیلد *item\_id* است که Primary key جدول میباشد.

برای مشخص کردن این که هر جواهر مربوط کدام بخش و از کدام نوع میباشد، دو فیلد *type\_id* و *category\_id* به حیث Foreign Key در این جدول در نظر گرفته شده است.



item	
item_id	INT(11)
item_name	VARCHAR(128)
price	INT(11)
picture	VARCHAR(128)
madein	VARCHAR(64)
weight	FLOAT
type_id	INT(11)
category_id	INT(11)
Indexes	
PRIMARY	item_id
type_item_fk	type_id
category_item_fk	category_id

شکل ۱۴: جدول *item*

## جدول مدیران users

این جدول برای ذخیره نمودن Profile مدیران و همچنان برای ذخیره و چک نمودن Credential آنها به کار می‌رود. هر user باید دارای یک آی‌دی unique باشد که فیلد user\_id به همین منظور ساخته شده و منحیت Primary Key معرفی شده است. فیلد user\_level نیز به منظور Authorization در نظر گرفته شده است که میتوان دو نوع مختلف مدیر (صلاحیت محدود، صلاحیت کامل) تعریف کرد.

users	
user_id	INT(11)
firstname	VARCHAR(32)
lastname	VARCHAR(32)
gender	TINYINT(1)
photo	VARCHAR(128)
phone	VARCHAR(16)
email	VARCHAR(128)
address	VARCHAR(255)
username	VARCHAR(32)
password	VARCHAR(64)
user_level	TINYINT(1)
Indexes	
PRIMARY	
phone	
email	

شکل ۱۵: جدول users

## جدول فروشات sale

این جدول برای ذخیره نمودن فروشات جواهرات استفاده میشود که با جدول customer رابطه دارد. بنابراین فیلد customer\_id به حیث Foreign Key معرفی شده است. همچنین هر Bill که به فروش میرسد دارای شماره Unique و تاریخ فروش میباشد.

sale
bill_id INT(11)
customer_id INT(11)
sale_date DATE
Indexes
PRIMARY
customer_sale_fk

شکل ۱۶: جدول sale

### جدول جزییات فروشات sale\_detail

در این جدول جزییات مربوط به فروشات ذخیره میشود که با جدول sale رابطه 1:N دارد. به همین دلیل فیلد bill\_id که کلید اصلی جدول sale است، در این جدول به حیث Foreign key معرفی شده است که نمیتواند همزمان Primary key باشد. در نتیجه فیلد دیگری به نام detail\_id که Surrogate Key میباشد، در این جدول معرفی شده است تا نقش Primary key را در این جدول داشته باشد.

کلید خارجی (Foreign key) دیگری که در این جدول وجود دارد، item\_id میباشد که رابطه بین این جدول و جدول item را به وجود میآورد. همچنین تعداد جنس فروخته شده، قیمت آن و قیمت مجموعی نیز در این جدول ذخیره میشود.

sale_detail
detail_id INT(11)
bill_id INT(11)
item_id INT(11)
quantity INT(11)
unitprice INT(11)
totalprice INT(11)
Indexes
PRIMARY
sale_detail_fk
item_sale_fk

شکل ۱۷: جدول sale\_detail

## جدول تحویل اجناس delivery

این جدول برای اطمینان حاصل کردن از تحویل جنس فروخته شده به مشتری میباشد و با جدول فروشات (sale) رابطه‌ای از نوع 1:1 دارد. چرا که جنس فروخته شده فقط یک بار به مشتری تحویل داده میشود. به همین دلیل فیلد bill\_id منحص Foreign key در این جدول قرار گرفته است که همزمان منحص Primary key این جدول نیز معرفی شده است تا رابطه از یک نوع One to One را به وجود بیاورد. فیلدهای دیگر این جدول، نام شخص تحویل گیرنده و تاریخ تحویل دادن میباشد.

delivery	
bill_id	INT(11)
reciver_name	VARCHAR(64)
delivery_date	DATE
Indexes	
PRIMARY	

شکل ۱۸: جدول delivery

## فصل پنجم: برنامه نویسی Client Side

### ساخت عناصر صفحات در HTML

هر صفحه اینترنتی (Web Page) در حقیقت یک فایل HTML از نوع متنی یا Text می باشد که از عناصر یا elementها تشکیل شده و برای ایجاد عناصر از تگ ها (Tag) استفاده می شود. با کمک تگ های اجتملی عناصر یا Elements ساخته می شوند.

در زبان HTML حدود ۸۰ عنصر تعریف شده است. تگ های HTML به وسیله دو کاراکتر < و > ساخته می شوند که این تگ ها معمولاً به صورت زوج (Pair) ظاهر می شوند.

تگ های HTML را می توان با حروف کوچک (lower case) و یا بزرگ (upper case) نوشته کرد و این تگ ها -case-sensitive نمی باشند. برای مثال دو تگ <b> و <B> معادل هم هستند اما سازمان جهانی وب توصیه می کند که از حروف کوچک استفاده شود.

### ویژگی های تگ (Tag Attributes)

تگ ها می توانند حاوی معلومات اضافی دیگری هم باشند. به این اطلاعات ویژگی یا Attribute گفته می شود و وظیفه آنها بیان معلومات اضافه تر درباره یک عنصر (Element) می باشد. مثلاً در مورد تگ <body> ویژگی به نام bgcolor وجود دارد که رنگ زمینه (background) را تعیین می کند.

ویژگی ها به صورت کلی به شکل name="value" نوشته می شوند و همیشه به تگ شروع یک عنصر (Opening Tag) اضافه می شوند و در صورتی که ویژگی های یک عنصر مشخص نشوند، از مقدارهای قراردادی یا default آنها استفاده خواهد شد. مثلاً در تگ body اگر ویژگی bgcolor نوشته نشود از رنگ سفید برای زمینه صفحه استفاده خواهد شد.

### عنوان ها یا سرخط ها (Headings)

عنوان ها با کمک تگ های <h1> تا <h6> تعریف می شوند <h1> معرف بزرگ ترین سرخط و <h6> معرف کوچک ترین سرخط است. براورز به هنگام نمایش یک عنوان به صورت اتوماتیک یک خط خالی قبل و بعد از هر عنوان اضافه خواهد کرد.

## پاراگراف‌ها (Paragraphs)

پاراگراف‌ها با کمک تگ `<p>` معرفی می‌شوند. براورزها به هنگام نمایش یک پاراگراف به صورت اتوماتیک یک خط خالی قبل و بعد از آن نمایش خواهند داد.

## خط جدید (Line Breaks)

برای رفتن سر خط جدید از تگ `<br>` استفاده می‌شود. در این حالت یک پاراگراف جدید ساخته نمی‌شود. تگ `<br>` از نوع تگ‌های خالی بوده و دارای تگ انتهایی (Closing Tag) نمی‌باشد.

## توضیحات (Comments)

برای نوشتن شرح و توضیحات در مورد کدهای HTML باید از تگ خاصی استفاده شود. برای این کار باید توضیحات در بین علامت‌های `<!-->` و `!<` قرار بگیرد. براورز تگ‌های comment را در نظر نگرفته و محتوی آن‌ها را نمایش نخواهد داد و فقط شرح و توضیحات برای برنامه‌نویس و دیگر افرادی که احتمالاً در آینده با کد HTML کار خواهند کرد، مفید خواهد بود.

## لینک‌ها (Links)

صفحات HTML با کمک لینک‌ها به یکدیگر متصل (Link) می‌شوند. با کمک لینک‌ها میتوان از یک صفحه به هر صفحه دیگر در وب متصل شد. این کار با کمک عنصری به نام A یا Anchor میسر می‌گردد.

برای به وجود آوردن لینک به صفحات دیگر از تگ `<a>` استفاده می‌شود. لینک‌ها می‌توانند به بخش دیگری از همان صفحه، صفحات دیگر وب، تصاویر، فایل‌های صوتی یا حتی فیلم‌ها و... اشاره کنند (در واقع به هر آدرس اینترنتی).

شکل کلی یک لینک به قرار زیر است:

```
<a href="url">Some Text</a>
```

با کمک ویژگی target امکان تعیین مقصد لینک جدید فراهم می‌شود. اگر خواسته باشیم که براورز لینک را در صفحه جدیدی باز کند باید از ویژگی target و مقدار `"_blank"` برای آن استفاده شود.

## جداول (Tables)

نمایش و قرار دادن اطلاعاتی که به صورت row و column هستند مانند جدولی از اعداد و ارقام تنها با استفاده از تگ‌های `<p>`، `<br>` و `&nbsp;` بسیار مشکل می‌باشد و به همین دلیل زبان HTML تگ‌های خاصی را برای ساختن اطلاعاتی که



ذاتا به شکل جدولی هستند در نظر گرفته است. تگ‌های `<table>`، `<tr>` و `<tr>` از مهم‌ترین تگ‌هایی هستند که برای ساختن و تعریف نمودن جدول‌ها به کار می‌روند.

### لیست‌ها (Lists)

در زبان HTML دو نوع لیست مرتب (ordered) و نامرتب (unordered) وجود دارد که هر کدام به ترتیب از طریق تگ‌های `<ol>` و `<ul>` به وجود می‌آیند. همچنین برای ساختن آیتم برای هر کدام از این لیست‌ها از تگ `<li>` استفاده می‌شود.

### فرم‌ها (forms)

از طریق استفاده از عنصر `<form>` و چند تگ مرتبط امکان دریافت اطلاعات از بازدیدکنندگان و یا تبادل اطلاعات بین صفحات مختلف وجود دارد. با کمک این گروه از تگ‌ها میتوان باکس‌های ورودی اطلاعات متنی (text fields)، چک‌باکس‌ها (check-boxes)، دکمه‌های رادیویی (radio-buttons) و ... را نمایش داده و همچنین دکمه‌های ارسال (submit) و یا reset را به وجود آورد.

### تصاویر (Images)

برای نمایش دادن تصاویر از تگ `<img>` و خاصیت src این تگ استفاده می‌شود. تگ `<img>` از نوع تگ‌های Single است، بدین معنا که فقط دارای یک یا چند ویژگی و attribute بوده و دارای تگ انتهایی یا `</img>` نمی‌باشد. از دیگر ویژگی‌های مهم تگ `<img>` ویژگی‌های width و height جهت تعیین عرض و ارتفاع نمایش تصویر است.

## به وجود آوردن استایل و دیزاین صفحات در CSS

خاصیت مهم زبان HTML قالب‌بندی ساختار یک صفحه می‌باشد و قابلیت کنترل استایل و دیزاین صفحات را ندارد. برای تعیین استایل و ایجاد افکت‌های بیشتر از زبان CSS استفاده می‌شود. همچنین CSS این امکان را ایجاد می‌کند تا دیزاینر بتواند چندین مشخصه در طراحی صفحه را به یک‌باره تعیین کند.

به عنوان مثال خصوصیات تمامی تگ‌های h1 موجود در صفحه اعم از اندازه، فونت و رنگ را مشخص نماید و یا مشخصات ظاهری چندین صفحه را در یک فایل CSS تعیین نماید.

## روش‌های به کارگیری CSS

سه روش برای به کارگیری CSS وجود دارد:

۱- **Inline**: در این روش کدهای CSS در ادامه نام تگ HTML و به عنوان مقدار ویژگی style در داخل Opening Tag قرار می‌گیرد. این روش اگر چه ساده می‌باشد اما با این روش نمی‌توان یک style را برای یک مجموعه تگ در آن واحد اعمال کرد و در نتیجه باید استایل‌ها را تکرار نمود.

۲- **Internal**: در این روش برای صفحات جداگانه‌ای که متن طولانی دارند بسیار مناسب می‌باشد. به کمک این روش می‌توان عین style را برای تمام عناصر مشابه در یک صفحه تعیین نمود. در این حالت باید استایل‌ها را در داخل تگ <style> در قسمت <head> صفحه قرار داد.

۳- **External**: در صورتی که بخواهیم از یک style در چندین صفحه استفاده کنیم، این روش مناسب‌ترین روش می‌باشد. در این حالت می‌توان برای تمام صفحات وب ظاهر یکسانی را به وجود آورد. یعنی به جای این که برای هر صفحه از style‌های داخلی استفاده کنیم، می‌توانیم تمام صفحات را با یک style خارجی به یک‌بارگی تحت تاثیر قرار بدهیم.

در دو روش دوم (طریقه Internal و External)، ابتدا باید عنصر و یا عناصری را که می‌خواهیم برای آن‌ها استایل تعیین کنیم را انتخاب کنیم. به این قسمت از کد CSS به اصطلاح Selector گفته می‌شود. سپس در ادامه خواص مورد نظر را به همراه مقدارهای هر کدام داخل علامت‌های { } قرار داده و هر خاصیت و مقدار آن را از خاصیت و مقدار بعدی توسط کاراکتر ; جدا می‌نماییم.

## استفاده از ویژگی‌های Class و ID

با استفاده از کلاس می‌توانیم برای تگ‌های HTML استایل‌های متفاوتی را تعریف کنیم. برای نمونه فرض کنیم می‌خواهیم در صفحه دو نوع پاراگراف داشته باشیم: یکی با رنگ سیاه و دیگری با رنگ سرخ. برای مشخص کردن این دو نوع پاراگراف می‌توانیم از سلکتور کلاس استفاده کنیم. ابتدا استایل را به صورت زیر می‌نویسیم:

```
p.black {color: black}
```

```
p.red {color: red}
```

در مرحله بعد باید برای هر پاراگرافی که می‌خواهیم به رنگ سیاه باشد از class="black" استفاده کنیم. و برای پاراگراف‌هایی به رنگ سرخ از class="red" استفاده نماییم.

روش دیگری که برای تعریف استایل وجود دارد، استفاده از ID است. هر عنصر در یک صفحه اینترنتی باید دارای یک ID غیرتکراری باشد. بدین معنی که دو عنصر یا Tag نمی‌توانند دارای عین ID باشند. سلکتور ID در زبان CSS با علامت # مشخص می‌شود. به طور مثال: با استایل زیر همه عناصری که دارای id="border" باشند، با یک چوکات سبزرنگ نمایش داده می‌شوند:

```
#border {border: 1px solid green ;}
```

## اعتبارسنجی و مدیریت رویدادها توسط JavaScript

### مدیریت رویدادها

کدهای جاوا اسکریپت را نیز میتوان مانند زبان CSS در داخل صفحات (Internal) و یا خارج از صفحه اینترنتی و به شکل یک فایل مستقل (External) نوشته و ذخیره نمود.

اگر از روش دوم (External) استفاده شود میتوان اسکریپت‌های نوشته شده را در همه صفحات وبسایت استعمال مجدد (Reuse) کرد که تاثیر بسیار زیادی در کارکرد (Performance) وبسایت داشته و همچنان انرژی و وقت زیادی برای دیزاینر صرفه‌جویی خواهد شد.

اگر از روش اول (Internal) استفاده شود، کدهای جاوا اسکریپت منحصراً محتویات (Content) تگ <script> در قسمت <head> صفحه قرار می‌گیرد. اما اگر کدهای جاوا اسکریپت به روش دوم (External) در یک فایل Separate نوشته شده باشد، باز هم در قسمت <head> صفحه و از طریق تگ <script> لینک می‌شود اما با استفاده از ویژگی src="url" و همچنان به حیث محتویات چیزی قرار نمی‌گیرد.

در حالی که کدهای جاوا اسکریپت را میتوان برای اجرا کردن فرمان‌های دلخواه از طرف Programmer استفاده نمود، همچنان میتوان اسکریپت‌ها را به عنوان کنترل کننده حوادث Event Handler نیز قرار داد. در این صورت، کدهای جاوا اسکریپت دقیقاً زمانی اجرا می‌شود که حادثه خاصی از طرف User اتفاق افتاده باشد.

کنترل کننده‌های حوادث به براورز می‌فهمانند در برخورد با یک حادثه مشخص، چگونه باید رفتار کند. مانند کلیک شدن دکمه‌ای از ماوس، ارسال شدن یک فورم، قرار گرفتن نشانه‌گر (Cursor) ماوس بر روی عنصر خاص و .... که از جمله مهم‌ترین حوادث در جاوا اسکریپت می‌باشند.

اسکریپتی را که خواسته باشیم در عکس‌العمل به یک حادثه نشان دهیم، میتوانیم به دو طریقه نوشته کنیم:

۱- به صورت Inline در ادامه ویژگی Event مربوطه در تگ HTML مورد نیاز

۲- به صورت یک Function در قسمت <head> صفحه

مثال برای روش اول که کودهای جاوا اسکریپت به عنوان مقدار (value) حادثه مربوطه نوشته می‌شود:

```

```

که در صورت کلیک شدن بالای تصویر مذکور یک Dialog Box ظاهر شده و پیام Are you sure? را نمایش می‌دهد.

مثال روش دوم که اسکریپت‌ها به عنوان یک فانکشن در <head> صفحه معرفی می‌شود:

```
<html>
<head>
<script type="text/javascript">
    Function greet() {
        var username = prompt("Please enter your name");
        alert("Welcome " + username);
    }
</script>
</head>
<body onLoad="greet();" >
</body>
</html>
```

در این مثال با باز شدن صفحه (حادثه onLoad) فانکشنی به نام greet اجرا (Call) می‌شود که سپس از طریق این فانکشن یک Dialog Box ظاهر شده و از شخص می‌خواهد که نام خود را تایپ نماید. بعد از تایپ شدن نام شخص، پیام خوش آمدید به همراه نام همان شخص ظاهر می‌گردد.

## اعتبارسنجی فرم‌ها

همان طور که در تشریح زبان HTML گفته شد، فرم‌ها یکی از قدرتمندترین قابلیت‌های زبان HTML می‌باشند و همچنان یکی از مهم‌ترین عناصر یک صفحه اینترنتی. چرا که فقط از طریق همین عنصر میتوان دیتا را از یوزر دریافت کرد.

اما همین عنصر فوق العاده مهم و کاربردی، نقطه ضعف امنیتی هم به دنبال دارد. این که یوزر میتواند در یک فیلد متنی (Text Field) هر مقداری را وارد نماید، حتی دستورات و کدهای مخرب. بنابراین باید تمام فورم‌هایی که در یک وبسایت قرار می‌گیرند، ابتدا اعتبارسنجی (Validation) گردیده و سپس به طرف Server فرستاده شوند.

همان طور که در قسمت HTML بیان گردید، برای ارسال هر فورم، باید یک دکمه از نوع Submit به وجود بیاید که وظیفه این Button از قبل مشخص شده است: ارسال دیتای تمام عناصر فورم به آدرس مشخص شده توسط Programmer در ویژگی action تگ فورم.

به همین لحاظ چنانچه توسط کدهای جاوا اسکریپت اعتبارسنجی هم صورت بگیرد، با این که میتوان معتبر بودن یا نبودن دیتا را تعیین کرد، اما نمیتوان از فرستاده شدن فورم جلوگیری نمود.

بنابراین باید در پروسه Validation ابتدا از فرستاده شدن خودکار فورم جلوگیری کرد. سپس بعد از این که صحت و اعتبار دیتا تضمین گردید، اقدام به ارسال فورم نمود.

برای این کار، دو روش کلی وجود دارد:

۱- استفاده از Submit Button اما کنترل فرستاده شدن و یا نشدن فورم با استفاده از حادته onSubmit

۲- استفاده از Blank Button و ارسال فورم به طور Manual از طریق میتود submit().

روش اول:

در این روش Submit Button فورم به حالت خود باقی می‌ماند. اما در حادته onSubmit باید یک فانکشن اجرا شود که پروسه اعتبارسنجی را انجام بدهد و در ختم این پروسه، در صورتی که معتبر بودن دیتا ثابت شود، مقدار منطقی True از طریق این فانکشن بازگشت (return) داده شود.

روش دوم:

در این روش Submit Button فورم را با یک Button معمولی (`<input type="button">`) جابه‌جا می‌نماییم. در این حالت، از آن جایی که فورم دارای دکمه ارسال نمی‌باشد، اصلاً فورم ارسال نمی‌شود! بنابراین از حادته onClick بر روی همین دکمه استفاده کرده و یک فانکشن اعتبارسنجی را مانند روش قبلی اجرا می‌نماییم.

سپس در صورتی که اعتبار دیتا ثابت گردید، با استفاده از میتود submit(). اقدام به ارسال فورم به طور Manual می‌نماییم.

## ایجاد انیمیشن و گالری تصاویر در jQuery

همان طور که در فصل دوم (میتودولوژی) نیز بیان شد، jQuery یک کتابخانه بسیار مفید برای جاوااسکریپت است که بسیار ساده و کارآمد است و مشکل جاوا اسکریپت را برای تطابق با برازرهای مختلف برطرف نموده است.

همچنین jQuery میتودهایی برای کار با AJAX نیز فراهم نموده و در این زمینه نیز کار را بسیار ساده کرده است. در جی کوئری میتوان از خصوصیت فراخوانی زنجیره‌ای (Chaining Call) میتودها استفاده نمود و این امر باعث می‌شود که چندین کد jQuery فقط در یک خط قرار بگیرد و در نتیجه کود بسیار مختصر گردد.

از آنجایی که jQuery یک کتابخانه کودهای نوشته شده جاوا اسکریپت است، بنابراین در ابتدا برای استفاده از آن باید فایل این کتابخانه را در صفحه مورد نظر لینک کرد که این کار با استفاده از تگ `<script>` و ویژگی `src="url"` در قسمت `<head>` صفحه انجام می‌شود.

در jQuery ابتدا باید یک یا چندین عنصر را انتخاب کرده (Select) و سپس میتوان عملیاتی (Action) را بر آن انجام داد. برای انتخاب کردن عناصر در jQuery از همان قوانین و قواعد زبان CSS استفاده می‌شود. به عنوان مثال:

```
$ ("p#msg").hide();
```

قطعه کود بالا، پس از انتخاب کردن عنصر پاراگرافی که دارای `id="msg"` باشد، آن را مخفی (hide) می‌نماید.

در jQuery نیز میتوان کودها را طوری نوشت که در مقابل حوادث (Event) اجرا شده و آن‌ها را مدیریت (Handle) نماید. به طور نمونه:

```
$ ("img.trigger").click(function() {
    $(this).hide(1000);
});
```

در این مثال، با واقع شدن (Fire) حادثه Click بالای عنصر تصویر (Image) که دارای کلاس trigger باشد، آن تصویر مخفی می‌شود. اما همان طور که مشاهده می‌شود این کار در مدت ۱ ثانیه انجام می‌شود (واحد زمان در جاوا اسکریپت میلی ثانیه است) که در نتیجه یک Animation بسیار ابتدایی و ساده را به وجود می‌آورد.

توصیه می‌شود تمام کودهایی که در jQuery نوشته می‌شوند در واقعه ready شی document قرار بگیرند. به این شکل:

```
$(document).ready(function() {

    // jQuery Codes

});
```

چرا که این کار باعث جلوگیری از مشکلات ناخواسته مختلفی می‌گردد. مانند:

– مخفی (Hide) کردن عنصری که هنوز Load نشده است میتواند باعث یک خطا Error شود.

– تغییر دادن اندازه (Size) تصویری که هنوز به طور کامل Load نشده است نیز میتواند باعث بروز مشکلاتی در براوررها گردد.

کتابخانه jQuery دارای میتودهای بسیاری در قسمت Effect می‌باشد. مانند: `hide()`, `show()`, `toggle()`, `slide()`, `fade()`, `animate()` و ... که مشابه همین اعمال را میتوان از طریق کودهای جاوا اسکریپت نیز انجام داد. اما مهم‌ترین خاصیت این میتودها در jQuery امکانات جانبی دیگری مانند `Timing`, `Chaining` و `Callback` می‌باشد که با سهولت بسیار زیادی در jQuery قابل استفاده می‌باشند.

همان طور که در مثال قبلی نیز ذکر شد، میتوان به عنوان پارامتر هر کدام از میتودهای فوق الذکر، یک مقدار عددی که عبارت از زمان بر حسب میلی ثانیه می‌باشد، ارسال کرد. در نتیجه میتود مورد نظر به آهستگی در مدت زمانی مشخص شده انجام می‌شود و باعث به وجود آمدن Animation می‌شود.

خاصیت Chaining در جی کوثری کمک میکند تا بتوان چندین میتود را در یک خط به طور بسیار کوتاه و مختصر تایپ کرده و همچنان به ترتیب و یکی بعد از دیگری آن‌ها را اجرا نمود. به طور نمونه:

```
$("#box").fadeIn(1000).animate({left:"100px"},"slow").fadeOut(1000);
```

در این مثال، عنصری با آی‌دی box به آرامی در مدت ۱ ثانیه ظاهر گردیده و سپس به طرف چپ به اندازه 100 پیکسل به آهستگی حرکت نموده و سپس به آرامی در مدت ۱ ثانیه مخفی شده و از روی صفحه محو می‌گردد!

خاصیت **Callback** نیز مشکل بسیار بزرگی را در قسمت اجرا کدهای جاوا اسکریپت و یا **jQuery** که پی در پی نوشته شده باشند، حل می‌نماید. به طور مثال اگر چنین کودی نوشته شده باشد:

```
$("button").click(function() {
    $("p").hide(1000);
    alert("The paragraph is now hidden");
});
```

شاید تصور گردد که بعد از مخفی شدن عنصر پاراگراف پیام «The paragraph is now hidden» ظاهر شود. اما متأسفانه اصلاً چنین نیست! بلکه از آن جایی که عنصر پاراگراف در مدت زمانی ۱ ثانیه مخفی می‌شود، در همان ابتدای مخفی شدن پاراگراف، پیام متذکره ظاهر می‌گردد. و این در حالی است که هنوز پاراگراف به طور کامل مخفی نشده است!

علت این امر آن است که براورز بعد از اجرا کردن میتود **hide()** به سراغ فرمان بعدی می‌رود و آن را اجرا می‌کند و نتیجه کار، اجرا شدن هر دو فرمان در یک وقت است که اصلاً قابل قبول نیست و توقع نمی‌رفت.

خاصیت **Callback** در **jQuery** برای رفع این مشکل معرفی شده است. چنان‌که از نام این خاصیت نیز پیداست، این ویژگی به مفهوم اجرا کردن **(Call)** یک فانکشن دیگر بعد از اتمام عمل فعلی می‌باشد. **(Call Back)** در نتیجه با استفاده از این خاصیت مفید میتوان مثال قبل را به این شکل اصلاح و بازنویسی کرد:

```
$("button").click(function() {
    $("p").hide(1000, doAlert);
});
Function doAlert() {
    alert("The paragraph is now hidden");
}
```

## درخواست های ناهمزمان و تکنالوژی Web 2 توسط AJAX

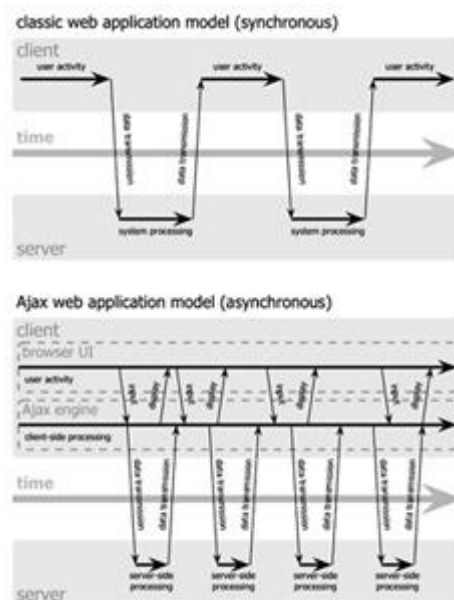
**AJAX** تکنالوژی است که به کمک آن می‌توان اینترفیس یک برنامه تحت وب را طوری دیزاین نمود که وقتی یوزر بالای یک دکمه و یا یک لینک کلیک می‌کند، همه عملیات ارسال دیتا و دریافت نتایج در پشت صحنه **(Background)** انجام شود و فقط آن قسمت از اینترفیس که قرار است معلومات جدید را نمایش بدهد تغییر کند، بدون این‌که تمام صفحه دوباره **Load** شود.



در این تکنالوژی، هر عملی که از سوی یوزر به طور معمول باعث تولید یک تقاضای HTTP می‌شود به جای ارسال مستقیم به وب‌سرور، باعث اجرا شدن یک فرمان جاوا اسکریپتی و هدایت آن به انجین AJAX می‌شود. هر نوع پاسخی به یوزر از سوی وب‌سرور (مانند کنترل صحت دیتا وارد شده در یک فورم و یا حتی برخی از انواع هدایت یوزر در داخل وب‌سایت) نیازی به ارسال یک صفحه جدید به سمت یوزر ندارد و تنها همان قسمتی که باید تغییر کند، تغییر می‌کند و نه بیشتر!

در Web کلاسیک، وقتی یوزر فورمی را خانه‌پوری می‌کند و به سرور ارسال می‌نماید، وب‌سرور با لود کردن مجدد صفحه (refresh) و نمایش یک پیام و یا نتیجه Process اطلاعات، به وی پاسخ می‌دهد و به همین دلیل، هم وقت وب‌سرور برای ارسال کل محتویات آن صفحه گرفته می‌شود و هم یوزر باید برای دریافت کامل آن صفحه منتظر بماند که نتیجه این مدل، کم شدن کارایی (Performance) وب‌سرور، مصرف Bandwidth زیادتر و به هدر رفتن وقت و هزینه زیاتر می‌باشد. اما استفاده از تکنالوژی و یا تکنیک AJAX این مشکلات را به طور قابل ملاحظه‌ای کاهش می‌دهد.

در دیگرام زیر به مقایسه مدل قبلی وب (Synchronous) و مدل جدید وب که با معرفی شدن تکنالوژی AJAX به میان آمده است (Asynchronous) می‌پردازیم:



شکل ۱۹: مقایسه مدل وب کلاسیک و Web 2

همان طور که در شکل دیده می‌شود، اگر تعامل یوزر با یک وب‌سایت را در محور زمان ترسیم کنیم، چه تفاوتی میان پروسه ارسال و دریافت دیتا در وب‌سایت‌های کلاسیک و وب‌سایت‌های مدرن مبتنی بر تکنالوژی AJAX وجود دارد؟

اگر به گراف دوم دقت شود، دیده می‌شود که هنگامی که اینترفیس یک برنامه وب از **AJAX** استفاده می‌کند، ارتباط میان یوزر و اینترفیس وبسایت هرگز قطع نمی‌شود. یوزر همیشه وبسایت را در دسترس و پیش روی خود می‌بیند و این در حالی است که انجین **AJAX** در پشت صحنه عملیات ارسال و دریافت دیتا را مدیریت می‌کند.

بنابراین و با در نظرداشتن گراف فوق اگر انجین **AJAX** برای پاسخ دادن به یوزر نیازمند گرفتن دیتا از سمت وبسرور باشد و یا اگر قرار باشد اصل دیتا به وبسرور ارسال شود و یا هم اگر لازم باشد کدهای اضافه‌ای برای نمایش تغییرات اینترفیس لود شوند، و یا اگر نیاز به دست آوردن اطلاعات از دیتابیس باشد، همه این کارها و بیشتر از این، به طور ناهمزمان (**Asynchronous**) و با استفاده از شی **XHR (XML HTTP Request)** بدون اینکه وقفه‌ای در ارتباط میان یوزر و اینترفیس وبسایت به وجود بیاید، توسط انجین **AJAX** انجام خواهد شد.

همان طور که قبلاً نیز اشاره شد، **AJAX** یک تکنالوژی و یا به اصطلاح ساده‌تر یک تکنیک می‌باشد و یک زبان برنامه‌نویسی یا برنامه (**Software**) نمی‌باشد. بنابراین برای تطبیق و استفاده از این تکنالوژی با استفاده از اسکریپت‌های نوشته شده در زبان **JavaScript** ابتدا باید انجین **AJAX** را که عبارت از شی **XHR** می‌باشد، به وجود آورد.

این کار را با در نظرداشتن مشکل عدم سازگاری بین براورهای مختلف، میتوان به این شکل انجام داد:

```
var xmlhttp;
if (window.XMLHttpRequest)
{ // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
}
else
{ // code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
```

همان طور که در کد دیده می‌شود، برای تمام براورزها اعم از **Firefox**، **Chrome** و ... و حتی **IE** بالاتر از ورژن ۷ میتوان این کار را از طریق **Object** از پیش تعریف شده داخلی خود براورز انجام داد. اما در صورتی که براورز **Internet Explorer** ورژن ۵ و یا ۶ باشد، در این صورت باید به کمک تکنالوژی **ActiveX** مایکروسافت شی **XHR** را به وجود آورد.

پس از به وجود آوردن انجین **AJAX** میتوان درخواست‌های ناهمزمان (**Asynchronous Request**) را از طریق دو روش **GET** و **POST** به وبسرور ارسال نمود که برای این کار از میتود **open()** و **send()** باید استفاده شود.

میتود **open()** دارای سه پارامتر می‌باشد که به ترتیب عبارتند از: **url**، **method**، **async** که پارامتر اول یکی از دو روش **GET** یا **POST** بوده میتواند و پارامتر دوم آدرس فایل در وبسرور می‌باشد که ارتباط با آن برقرار می‌شود. پارامتر سوم نیز یک

مقدار Boolean می باشد که True ارتباط ناهمزمان (Asynchronous) و مقدار False ارتباط همزمان (Synchronous) را تعیین می کند.

سپس از طریق میتود `send()` که فقط یک پارامتر از نوع `String` قبول می کند، میتوان دیتا مربوطه را به وب سرور ارسال نمود. همچنین برای دریافت `Response` و استفاده از آن، باید از رویداد `onreadystatechange` استفاده کرده و خوب است که برای جلوگیری از اشتباهات ناخواسته کد وضعیت پروتوکول `HTTP` یا `HTTP Status Code` را چک کرده و سپس به نمایش دادن جواب و یا به وجود آوردن تغییرات در صفحه پرداخت.

در مثالی که در ادامه ذکر خواهد شد، ابتدا انجین `AJAX` به وجود آمده و سپس یک درخواست `Asynchronous` از طریق روش `GET` با هیچ دیتایی از جانب `Client`، فرستاده شده است. سپس از طریق رویداد `onreadystatechange` و بعد از چک کردن کد وضعیت `HTTP` و `readyState`، نتیجه درخواست در عنصری با `ID` مانند `"some"` قرار خواهد گرفت. نتیجه کار این که بعد از اجرا شدن تمام این کدها، معلومات جدید به حیث محتوای یک عنصر `Load` می شود بدون این که نیاز باشد تمام صفحه دوباره از وب سرور درخواست شده و صفحه `Refresh` شود.

```
var xmlhttp;
if (window.XMLHttpRequest) {
    xmlhttp=new XMLHttpRequest();
}
else {
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange=function() {
    if (xmlhttp.readyState==4 && xmlhttp.status==200) {
        document.getElementById("some").innerHTML=xmlhttp.responseText;
    }
}
xmlhttp.open("GET","url",true);
xmlhttp.send();
```

## فصل ششم: برنامه نویسی Server Side

### ارتباط با دیتابیس Database Connectivity

همان طور که در قسمت میتودولوژی هم ذکر گردید، برای برنامه نویسی Server Side این مونوگراف، از زبان برنامه نویسی PHP استفاده شده است که یک زبان Open Source، Cross Platform و HTML Embedded می باشد.

برای برقراری ارتباط با دیتابیس از طریق PHP از یک اکستنشن این زبان به نام PDO یا PHP Data Object استفاده می شود که در حقیقت PDO یک انترفیس بسیار ساده، واضح و سازگار می باشد که اکثر دیتابیسی های مشهور جهانی را نیز پشتیبانی می کند. مانند: MySQL، Oracle، SQL Server، PostgreSQL، Informix، SQLite و ... .

PDO در واقع یک لایه انتزاعی (Abstraction Layer) برای برنامه های نوشته شده به زبان PHP می باشد که یک API یا Application Programming Interface بسیار سازگار را ارائه می کند که نسبت به سیستم دیتابیس که به آن متصل می شود، مستقل می باشد.

در زبان PHP برای کار کردن با هر سیستم دیتابیس، فانکشن هایی از قبل ساخته و تعریف شده اند. از آن جایی که دیتابیس استفاده شده در این مونوگراف MySQL می باشد، فقط به آن عده از توابع که برای برقراری ارتباط و تبادل دیتا با MySQL می باشد، می پردازیم.

#### mysql\_connect

این فانکشن همان طور که از نام آن پیداست برای متصل شدن به دیتابیس MySQL به کار می رود که دارای سه Parameter می باشد. پارامتر اول عبارت از نام سرور، پارامتر دوم username و پارامتر سوم password برای اتصال به سرور دیتابیس می باشد.

```
mysql_connect("host","username","password");
```

این فانکشن در صورت وصل شدن به سرور مربوطه، یک Resource ID بازگشت می دهد که از آن برای ارسال فرمان ها (Queries) در آینده استفاده خواهد شد.

## mysql\_select\_db

این فانکشن برای تعیین کردن (یا انتخاب کردن) دیتابیس مورد نظر در یک DBMS Server می باشد. از آن جایی که در یک سرور دیتابیس امکان دارد که چندین دیتابیس وجود داشته باشد، ابتدا و قبل از اجرا کردن هر فرمان، باید دیتابیس مورد نظر را Select کرد.

این فانکشن فقط یک پارامتر ورودی از نوع String دارد که عبارت از نام دیتابیس دلخواه می باشد.

```
mysql_select_db("database-name")
```

## mysql\_query

این فانکشن که بسیار پر استفاده می باشد و به دفعات بسیار زیاد در برنامه های وب که به زبان PHP نوشته شده باشند، استفاده می شود، برای اجرا نمودن فرمان های دیتابیس (SQL Query) می باشد. این فانکشن دارای دو پارامتر می باشد که پارامتر اول Command Text و پارامتر دوم عبارت از Resource ID می باشد که در فانکشن mysql\_connect یادآوری گردید. از قابلیت های بسیار جالب و مفید این تابع آن است که میتواند انواع مختلف فرمان ها به زبان SQL را بالای دیتابیس اجرا نماید. از کوئری SELECT و INSERT و UPDATE و DELETE گرفته تا کوئری هایی مانند CREATE TABLE و ALTER TABLE و حتی کوئری های DROP!

```
mysql_query("sql-command", connection)
```

## ارسال Query به دیتابیس

همان طور که بیان گردید، برای اجرا نمودن یک Query ابتدا دو مرحله ضروری را باید طی نمود. اول، ایجاد و برقراری ارتباط از طریق فانکشن mysql\_connect و سپس انتخاب دیتابیس مورد نظر از طریق فانکشن mysql\_select\_db و بعد از طی هر دو مرحله میتوان با استفاده از فانکشن mysql\_query فرمان دلخواه را بالای دیتابیس اجرا نمود.

در این قسمت به نحوه ارسال و اجرا نمودن چهار Query که اساس تمام عملیه ها و Functionality های هر سیستم دیتابیس را تشکیل می دهد که عبارت از CRUD یا Create, Read, Update, Delete می باشد، به طور مختصر می پردازیم.

## عملیه Create

برای به وجود آوردن و ذخیره نمودن دیتا در یک دیتابیس از فرمان Insert استفاده صورت می‌گیرد که Syntax عمومی آن به این شکل است:

```
INSERT INTO tablename (col 1, col 2, col n) VALUES (value 1, value 2, value n)
```

که به این طریق میتوان فقط یک Record در هر فرمان به جدول مورد نظر اضافه نمود. اما اگر خواسته باشیم که چندین

Record را به طور همزمان در یک جدول ذخیره نماییم میتوانیم از همین کوئری اما به شکل Multiple Insertion

استفاده نماییم:

```
INSERT INTO tablename (col 1, col 2, col n) VALUES (value a1, value a2, value an), (value b1, value b2, value bn), ...
```

بدین معنی که کافی است مقدارهای هر Record را در یک قوس ( ) قرار داده و آن‌ها را توسط علامت Comma از هم جدا نماییم.

مثالی از کوئری INSERT:

```
INSERT INTO users (user_id, username, password) VALUES (NULL, 'admin', 'pass')
```

## عملیه Read

برای خواندن و به دست آوردن معلومات (Fetch نمودن دیتا) باید از کوئری SELECT استفاده شود. این کوئری تنها کوئری است که در زبان SQL دارای اشکال و حالات‌های بسیار مختلف می‌باشد. چرا که انتخاب نمودن معلومات یک جدول حالات بسیار مختلفی را تقاضا می‌کند.

به هر حال Syntax عمومی فرمان SELECT به این شکل می‌باشد:

```
SELECT col1, col2, col n FROM tablename WHERE col operator value
```

## عملیه Update

برای تغییر دادن یا Edit کردن دیتای موجود در یک جدول، از کوئری UPDATE استفاده می‌شود. نکته قابل توجه در مورد این کوئری و کوئری بعدی (یعنی DELETE) آن است که نباید در صورت عدم نیاز، بدون Criteria استفاده شوند. چرا که تاثیر این

Query ها بالای تمام Record های یک جدول گذاشته می شود و در صورتی که به طور مثال فرمان حذف اجرا کرده باشیم، ناخواسته تمام Record های جدول حذف خواهند شد!

شکل عمومی فرمان UPDATE:

```
UPDATE tablename SET col1 = value1, col2 = value2, col n = value n
WHERE col operator value
```

## عملیه DELETE

جهت حذف نمودن معلومات از یک جدول به کوئری DELETE نیاز داریم که یکی از ساده ترین فرمان های موجود در زبان SQL می باشد. چرا که از شکل ها و حالت های مختلف برخوردار نمی باشد. شکل عمومی این کوئری:

```
DELETE FROM tablename WHERE col operator value
```

در تمام مثال های فوق منظور از operator هر یک از علامت های <, >, =, <=, >=, <> می باشد که در برنامه نویسی نیز کاربرد دارند. همچنین در زبان SQL علامت های دیگری نیز مانند IN, LIKE و ... نیز وجود دارد که هر کدام دارای مورد استعمال خاصی می باشند.

در مجموع قسمتی که بعد از کلمه کلیدی WHERE قرار میگیرد تحت عنوان Criteria (شرط یا معیار) شناخته می شود که معادل ساختار شرطی IF در برنامه نویسی می باشد.

در پایان بعد از این که کوئری عملیه مربوطه را تشکیل و ترتیب نمودیم، میتوان آن را مستقیماً به عنوان پارامتر اول فانکشن mysql\_query قرار بدهیم و یا هم با استفاده از یک متحول (Variable) فرمان SQL را ذخیره نموده و این متحول را به عنوان پارامتر اول Pass بدهیم که نتیجه هر دو روش کاملاً یکسان است. به این شکل:

```
mysql_query("SELECT * FROM users", $con)
```

یا:

```
$sqlcom = "SELECT * FROM users";
mysql_query($sqlcom, $con)
```

فانکشن `mysql_query` بعد از اجرا شدن، یک `Recordset` بازگشت می‌دهد که اگر فرمان اجرا شده، فرمانی مانند `DELETE` و یا `UPDATE` باشد، مقدار بازگشت داده شده یک `Boolean Value` می‌باشد که نشان‌دهنده موفقیت (`True`) و یا عدم موفقیت (`False`) این تابع است.

اما اگر فرمان اجرا شده، فرمان خواندن و به دست آوردن دیتا، یعنی `SELECT` باشد، در `Recordset` که توسط این تابع بازگشت داده می‌شود، `Record`های انتخاب شده، ذخیره می‌گردد که با استفاده از تابع دیگری که در قسمت بعدی معرفی خواهد شد، قابل دسترسی و استفاده می‌باشد.

## دریافت معلومات از دیتابیس

همان طور که اشاره شد، اگر فرمان `SELECT` توسط فانکشن `mysql_query` اجرا شود، حاصل یک `Recordset` یا مجموعه‌ای از ریکاردها است. اما نکته اینجاست که `Recordset` در حقیقت در زبان `PHP` یک `Resource` می‌باشد و دیتای قابل استفاده و مشاهده نیست.

به همین دلیل ابتدا باید توسط یک فانکشن دیگر، ریکاردهای ذخیره شده در این منبع را مورد دسترسی قرار داد، سپس میتوان از آن‌ها استفاده کرده و یا آن‌ها را نمایش داد. این فانکشن `mysql_fetch_assoc` می‌باشد که فقط دارای یک پارامتر می‌باشد و آن هم عبارت از `Resource ID` می‌باشد که از طریق فانکشن `mysql_query` بازگشت داده شده است.

به این شکل:

```
$result = mysql_query("SELECT * FROM users", $con);
$rows = mysql_fetch_assoc($result);
```

در مثال فوق، ابتدا فرمان `SELECT` اجرا شده که باعث انتخاب شدن تمام فیلدها و `record`های جدول `users` می‌شود، با استفاده از `connection` به نام `$con` که قبلاً از طریق فانکشن `mysql_connect` باید ساخته شده باشد.

سپس `recordset` که فانکشن `mysql_query` بازگشت داده است، در یک متحول به نام `$result` ذخیره گردیده است تا به عنوان پارامتر ورودی تابع `mysql_fetch_assoc` استفاده شود.

اما چیزی که این تابع (`mysql_fetch_assoc`) بازگشت می‌دهد، یک `Array` از نوع `Associative` می‌باشد. این نوع `Array` فقط در زبان `PHP` وجود دارد و عبارت از `Array` می‌باشد که `index`های آن حروفی می‌باشد. یعنی به جای استفاده از فقط اعداد منحنیث ایندکس، میتوان از `Character`ها نیز استفاده کرد.



بعد از این قسمت دیگر کار با دیتابیس تمام شده و است و ادامه کار مربوط به برنامه نویسی می شود! حال یک Array در اختیار داریم که دیتای مربوط به یک record در آن ذخیره شده است. اما همان طور که گفته شد، دیتای فقط یک record! چرا که فانکشن mysql\_fetch\_assoc در هر بار اجرا شدن، فقط یک row را می خواند.

این بدان معنی است که در بار اول، معلومات row اول و در بار دوم اجرا شدن، معلومات row دوم را می خواند. یعنی هر بار، یک row. بنابراین اگر خواسته باشیم معلومات تمام row ها را به دست بیاوریم، باید از یک ساختار Loop مانند for یا while در برنامه نویسی استفاده کنیم تا بتوانیم به تمام record ها دسترسی داشته باشیم.

از آن جایی که در اکثر موارد تعداد row های انتخاب شده نامعلوم می باشد، بهتر است از حلقه while استفاده کنیم:

```
while (mysql_fetch_assoc($result)) {
    echo $rows["username"];
}
```

اما نکته دیگری که وجود دارد، شرط انتهایی این حلقه می باشد چرا که بدون یک شرط صحیح، خطر بی نهایت (Infinite) شدن حلقه وجود دارد. یکی دیگر از خواص مفید تابع mysql\_fetch\_assoc این است که در صورتی که اجرا شود و record دیگری برای خواندن وجود نداشته باشد، مقدار منطقی false را بر می گرداند که در نتیجه باعث خاتمه یافتن حلقه while می شود.

## مدیریت ورود استفاده کنندگان

به عنوان مثالی از یک عملیه کامل Read با استفاده از کوئری SELECT در یک سیستم دیتابیس، پروسه ورود (Login) اعضای وبسایت (User) را مطالعه و بررسی می نماییم.

ابتدا به یک فورم Login ضرورت می باشد که از طریق تگ های HTML آن را به وجود می آوریم. چیزی شبیه این:

```
<form method="post">
<input type="text" name="username">
<input type="password" name="password">
<input type="submit" value="Login">
</form>
```

همان طور که مشاهده می شود روش ارسال دیتای این فورم POST انتخاب شده است چرا که معلومات این فورم حساس (Sensitive) می باشد و روش POST روش مطمئن تری (Secure) می باشد.

بعد از این که شخص `username` و `password` خود را در فیلدهای مربوطه وارد کرد و فرم را ارسال (Submit) نمود، این معلومات توسط براورز برای سرور به طور مخفیانه ارسال می‌شود.

در طرف سرور باید قبلاً `username` و `password` شخص در یکی از جدول‌های دیتابیس مانند `users` ذخیره شده باشد. در این صورت، کافی است که یک کوئری `SELECT` با مقدارهای به دست آمده از فرم برای دیتابیس فرستاده شود و نتیجه آن را بررسی کرد.

```
$username = $_POST["username"];
$password = $_POST["password"];
$result = mysql_query("SELECT * FROM users WHERE
username = '$username' AND password = '$password'", $con);
```

بعد از این که کوئری فوق اجرا گردد، از آنجایی که فرمان `SELECT` است در `recordset` بازگشت داده شده یا هیچ `record` (در صورتی که `username` یا `password` نادرست باشد) و یا هم یک `record` که متعلق به `user` است ذخیره می‌گردد.

سپس میتوان از تابع `mysql_fetch_assoc` برای ادامه پروسه Login استفاده نمود. اما از آنجایی که به احتمال زیاد به معلومات انتخاب شده نیازی نمی‌باشد، پس میتوان از فانکشن دیگری به نام `mysql_num_rows` استفاده نمود که تعداد `record`های موجود در یک `recordset` را شمارش (count) می‌کند.

منطق پروسه Login به این شرح است که اگر تعداد `record`های به دست آمده در `recordset` فقط و فقط یک `record` باشد، پس میتوانیم نتیجه بگیریم که `username` و `password` وارد شده درست بوده و متعلق به یک یوزر است که قبلاً در دیتابیس ثبت گردیده است.

اما اگر تعداد `record`ها صفر باشد، در این صورت احتمال دارد که `username` یا `password` غلط باشد و یا هم این که اصلاً چنین یوزری با `username` وارد شده، در سیستم وجود ندارد.

```
$no_row = mysql_num_rows($rows);
```

پس میتوانیم چنین کودی داشته باشیم:

```
if($no_row == 1) {
    // do something when login is correct
}
else { echo "Incorrect Username or Password!";
}
```

## امنیت سیستم

یکی از مروج‌ترین و خطرناک‌ترین حملاتی که تمام Web Application ها را تهدید می‌کند، حمله‌ای به نام SQL Injection می‌باشد که تمام Web Application هایی که مصئون نشده باشند، در مقابل این روش آسیب‌پذیر می‌باشند.

منشا این آسیب‌پذیری به یکی از عناصر اساسی در تمام وب‌سایت‌ها برمی‌گردد: فورم‌ها و Text Field هایی که یوزر از طریق آن‌ها قادر به وارد کردن دیتای متنی می‌باشد.

زمانی که یک Text Field به هر منظوری از طرف دیزاینر در یک صفحه اینترنتی قرار می‌گیرد، در حقیقت یک رخنه امنیتی را به روی هکرها باز می‌کند. چرا که یوزر نه تنها میتواند در آن Text Field حروف قابل قبول و آن چیزی که دیزاینر توقع داشته را وارد کند، بلکه میتواند هر چیز دیگری مثل فرمان‌ها و قطعه کدهای مخرب را نیز وارد نماید.

همان مثال قبل (فورم Login) را در نظر می‌گیریم. دو Text Field یکی برای وارد کردن username و یکی هم برای password در نظر گرفته شده است. دیتایی که در این دو باکس از طریق یوزر تایپ می‌شوند با استفاده از روش POST به سرور ارسال شده و سپس توسط کدهای PHP مستقیماً در دو متحول ذخیره گردیده است.

سپس همین دو متحول در کوئری SELECT باز هم به طور مستقیم و بدون هیچ گونه فلتري استفاده شده است. در نتیجه اگر یوزری که قصد و نیت خوبی ندارد، به جای وارد کردن username درست و واقعی، حروفی مانند:

```
someone ` OR 1 = 1 -
```

را وارد نماید، چه اتفاقی خواهد افتاد؟

اصلی کوئری به این شکل می‌باشد:

```
SELECT * FROM users WHERE username = '$username' AND password = '$password'
```

که بعد از وارد کردن username و password و جایگزینی آن در کوئری، به این شکل درخواهد آمد:

```
SELECT * FROM users WHERE username='someone'OR 1=1 --AND password = '$password'
```

این کوئری توسط دیتابیس به این شکل تفسیر خواهد شد: تمام فیلدهای جدول users را انتخاب کند به شرطی که فیلد username برابر با 'someone' باشد. با فرض این که چنین یوزری وجود ندارد، در ادامه اضافه شده است که یا اگر ۱ برابر

با ۱ باشد که این شرط به طرز غیر قابل انکاری! همیشه درست می‌باشد. پس تا این قسمت، دو شرط توسط دیتابیس چک شده است.

شرط اول که نتیجه آن false است و شرط دوم که همیشه نتیجه آن true می‌باشد. Logical Operator هم که بین این دو شرط استفاده شده است، OR می‌باشد که در صورتی که یکی از Operand ها True باشد، نتیجه آن True خواهد شد.

و سپس بعد از این دو شرط علامت -- قرار گرفته است که به معنای Comment در زبان SQL می‌باشد. در نتیجه متباقی فرمان SQL اصلا توسط دیتابیس در نظر گرفته نمی‌شود و حاصل کار انتخاب تمام record های جدول users است! بدون این که شخص اصلا username و یا password درستی را وارد نموده باشد!

همان طور که مشاهده می‌شود، اساس این روش حمله (SQL Injection) بسیار ساده اما بسیار خطرناک می‌باشد و آسیب‌پذیری در مقابل این حمله زمانی به وجود می‌آید که یوزر بتواند دیتایی را در یک فیلد متنی تایپ نماید. بنابراین واضح است که تقریباً تمامی وب‌سایت‌ها و Web Application ها در مقابل این حمله آسیب‌پذیر می‌باشند.

اما همان طور که منشا این آسیب‌پذیری و روش حمله به آن، بسیار ساده و ابتدایی است، جلوگیری از آن نیز تقریباً کار ساده و آسانی می‌باشد. کافی است که یوزر نتواند حروف ویژه دارای معنای خاص در دیتابیس مانند Double, Single Quotation Dash و... را وارد نماید.

بنابراین باید همیشه تمام حروف و کلماتی را که یوزر وارد می‌کند و از فورم‌ها جمع‌آوری می‌شود، ابتدا فیلتر نموده و هیچ‌گاه آن را به طور مستقیم به کار نبرد!

به همین منظور در زبان PHP فانکشن مخصوصی به نام mysql\_real\_escape\_string معرفی شده است که برای جلوگیری از حملات SQL Injection می‌باشد.

این تابع یک پارامتر از نوع String دریافت کرده و سپس آن را تجزیه کرده و حروف غیر مجازی را که نباید به دیتابیس فرستاده شود، توسط اضافه کردن Escaping Character که Backward Slash (\) می‌باشد، خنثی نموده و نتیجه کار را بازگشت می‌دهد.

پس میتوان در کودهای قبل با استفاده از این تابع، چنین اصلاحاتی را به وجود آورد و وب‌سایت را در مقابل این نوع حمله مصئون (Secure) ساخت:

```
$username = mysql_real_escape_string($_POST["username"]);
$password = mysql_real_escape_string($_POST["password"]);
```

## فصل هفتم: جمع بندی و نتیجه گیری

در مونوگراف «سیستم فروشات آنلاین جواهرفروشی» به طراحی و دیزاین یک دیتابیس انترتی (Web-based Database) پرداختیم و با استفاده از سیستم دیتابیس MySQL و زبان برنامه نویسی PHP که هر دو Open Source می باشند و با کمک زبان های دیگری مانند HTML، CSS و JavaScript، همچنان کتابخانه jQuery و تکنالوژی AJAX و فریم ورک Bootstrap موفق به انجام این کار شدیم.

موضوع مونوگراف «سیستم فروشات آنلاین جواهرفروشی» در حقیقت دیزاین یک وبسایت دینامیک برای فروشات می باشد که اشخاص مختلف از سراسر دنیا می توانند جواهرات و زیورآلات زیبا و اصیل افغانی را ببینند و همچنین مشتریان میتوانند به صورت آنلاین جواهرات مورد نظر خویش را خریداری نموده و هزینه آن را پرداخت نمایند.

برای ساخت این سیستم می توانستیم از هر کدام یک از سه پلتفورم PHP، ASP.NET و یا JSP استفاده کنیم که ترجیح دادیم از پلتفورم PHP استفاده نماییم. چرا که PHP کاملاً Open Source، رایگان، Cross Platform و بسیار زیاد HTML Embedded می باشد. و تمام این خصوصیات باعث می شود که با خیال آسوده و ذهن آرام، با دستی باز و انعطاف پذیری بالا به طراحی و دیزاین یک سیستم پرداخت.

استفاده از برنامه های Open Source منافع بسیار زیادی برای هر کشور و هر ارگان دارد که شاید مهم ترین آن فواید اقتصادی باشد. چرا که دیگر نیازی به خرید هر Software و درگیری با مسایل حقوقی مانند License و مسئله Copyright نمی باشد.

و از طرف دیگر، گذشته از مسایل اقتصادی، برنامه های Open Source آزادی را به برنامه نویسان و هر کسی که از کمپیوتر استفاده می کند، هدیه می دهد. آزادی که بسیاری از کمپنی های بزرگ جهانی برای حفظ منافع اقتصادی شان می خواهند از مردم دنیا سلب کنند.

## منابع

- 1- Achour Mehdi, Betz Friedhelm, Dovgal Antony (2012), PHP Manual, the PHP Documentation Group
- 2- McFarland David Sawyer (2012), JavaScript & jQuery the Missing Manual, OReilly
- 3- Goldstein Alexis, Lazaris Louis, Weyl Estelle (2011), HTML5 & CSS3 For The Real World, SitePoint
- 4- Pollock John (2010), JavaScript A Beginner's Guide, McGraw-Hill.Osborne
- 5- Curioso Andrew, Bradford Ronald, Galbraith Patrick (2010), Expert PHP and MySQL, Wiley Publishing Inc
- 6- Darie Cristian, Balanescu Emilian (2008), Beginning PHP and MySQL E-Commerce, Apress
- 7- Zervaas Quentin (2008), Practical Web 2.0 Application with PHP, Apress
- 8- Bulger Brad, Greenspan Jay, David Wall (2004), MySQL/PHP Database Applicatons, Wiley Publishing Inc