**بسم الله الرّحمن الرّحيم**

****

**دانشکده مهندسی کامپيوتر**

**پروپوزال پروژه ی مهندسی نرم افزار 2**

**:(F)اعضای گروه**

**محمد دولتشاه**

**ايرج احمدي**

**علي اتقايي**

**عزيزاحمد نورزايي**

**پويا محصوري**

**فهرست مطالب**

**1. مقدمه 3**

**1.1. خلاصه 3**

**1.2. دامنه 4**

**1.3. ابعاد محصول 4**

**1.4. مسائل حقوقي 7**

**2. ويژگي هاي اصلي 8**

**2.1. موارد فني 8**

**2.2. زمانبندي 9**

**2.3. ابزارهاي مورد استفاده 13**

**2.4. محدوديت ها 31**

**2.5. پيش فرض 33**

**2.6. متدولوژي 34**

**2.7. معماري مورد استفاده 41**

**2.8. انواع روش هاي تست نرم افزار 49**

**3. نحوه ي انجام کارهاي گروهي 52**

**3.1. تقسيم وظايف 52**

**3.2. مديريت کد نويسي 52**

**4. افراد و نقش ها 53**

**5. پشتيباني 54**

**1. مقدمه**

**1.1. خلاصه**

همواره انتخاب کردن يا به دست آوردن يک ايده ي خوب براي طراحي يک مشکل اساسي براي شرکت ها يا موسسات بوده وهست .

اگر امکاني براي اين شرکتها فراهم مي شد که به جاي سفارش دادن لوگو ( يا طرح سايت يا تبليغات …) به يک فرد به صورت انحصاري  , بتوانند از ميان چند ايده بهترين را انتخاب کنند اين مشکل برطرف ميشد.

در واقع هدف پروژه ما استفاده از طراحان فري لنسر براي طراحي لوگو , سايت , اپليکيشن , تبليغات , لباس , بسته بندي , هنر , مجله و کتاب به صورت جمع سپاري  براي خدمت رساني به نيازمندان اين خدمات است.

اين کار به صورت طراحي و پياده سازي يک سايت امکان پذير است به اين صورت که طراحان freelancer  ( کارمزدي )ميتوانند در سايت ثبت نام کنند و مهارت تخصص خود را بيان کنند , سپس کارفرماياني که به طرحي براي موسسه خود نياز دارند پروژه مورد نياز خود را در سايت تعريف مي کنند تا پيمانکاران سايت که همان طراحان باشند براي طرح مورد نظر يک طرح اوليه ثبت کنند سپس کارفرما ميتواند از بين چندين طرح موجود بهترين ( يا چندين طرح در صورت بزرگ بودن پروژه ) را براي ادامه کار انتخاب کند .

بعد از انجام کامل پروژه مبلغ تعيين شده براي پروژه را براي طراح ( يا طراحان ) آزاد کند.

در واقع عمل طراحي به صورت جمع سپاري انجام خواهد شد.

**1.2. دامنه**

دامنه ي افراد مرتبط با اين سيستم محدود به قشر خاصي از جامعه نمي شود و تمام افرادي که براي پيشبرد کار و پروژه ي خود نياز به يک طرح گرافيکي دارند و همچنين تمام افرادي که توانايي خلق ايده ها ي زيبا با توجه به نيازهاي کارفرمايان و پياده سازي آن در قالب يک طرح گرافيکي را دارند, با اين سيستم در ارتباط خواهند بود.

به طور مثال: گرافيست ها, شرکت هاي تازه تاسيس که نياز به داشتن يک لوگو دارند, شرکت ها و نهاد هايي که نياز به طراحي وب سايت دارند, توليدي هاي البسه که نياز به ايجاد طرح براي توليدات خود هستند و ... .

**1.3. ابعاد محصول**

**زمان:**

در بخش زمانبندی توضيح داده شده است.

**تعداد افراد :**

تعداد افرادي که براي قسمت پياده سازي پروژه با پروژه درگير هستند شامل 4 نفر از

اعضاي اصلي گروه است که اين 4 عضو وظيفه پياده سازي اصلي فايلهاي زيرين و سرور

سايد را به عهده دارند. همچنين پياده سازي سطحي و معقول رابط گرافيکي که براي يک

کاربر پيشرفته قابل درک باشد.

بعد از پياده سازي اصلي دو نفر طراح به گروه اضافه خواهند شد که در مجموع 6 نفر کار

طراحي و پياده سازي ظاهر حرفه اي و مورد قبول نرم افزار را به عهده مي گيرند.

اين مرحله شامل بهينه کردن کاربر پسند بودن نرم افزار است.

بعد از بيرون آمدن ورژن نهايي نرم افزار پروژه به يک تيم پشتيباني نياز دارد که تعداد

افراد اين تيم به بزرگ شدن بازار نرم افزار بستگي دارد .

افرادي که براي پشتيباني در نظر گرفته شده است براي هر هزار کاربر و يا هر 200

کاربر فعال نرم افزار 50 ساعت نفر در هفته ميباشد .

**تکنولوژي :**

تکنولوژي به کار رفته در نرم افزار استفاده از يک زبان برنامه نويسي مفسر و فريم ورک وب تحت آن زبان ميباشد.

زبان مورد نظرPython2.7 و فريم ورک Django1.5 ميباشد.

جنگو (به انگليسي: Django) يک چارچوب نرم‌افزاري تحت وب آزاد و متن‌باز است که به زبان پايتون نوشته شده است و از معماري مدل-نما-کنترل‌گر (به انگليسي: Model–view–controller) پيروي مي‌کند. جنگو اولين بار براي مديريت صفحات خبري وبسايت شرکت Lawrence Journal-World توسعه داده شد و سپس در سال ۲۰۰۵ تحت اجازه‌نامه بي‌اس‌دي منتشر شد.

هدف اصلي جنگو ساخت آسان سايت‌هاي پيچيده و وابسته به ديتابيس است و بر پايهٔ قابليت استفادهٔ مجدد و قابل اتصال بودن اجزاي مختلف، توسعهٔ سريع و اصل خودت تکرار نکن (DRY) طراحي شده است. جنگو سراسر از پايتون استفاده مي‌کند، حتي براي تنظيمات، فايل‌ها و مدل‌هاي اطلاعات.

براي سطح ديتابيس از تکنولوژي ديتابيس هاي رابطه اي استفاده مي شود که زبان مورد

استفاده ما زبان SQLاست.

براي سطح کاربري از تکنولوژي HTML 5 و CSS3 و jQuery 11 و

Bootstrap 3 استفاده مي شود.

**1.4. مسائل حقوقي**

همانطور که گفته شد نرم افزار ما تحت ليسانس جنگو منتشر مي شود که خود تحت اجازه‌نامه بي‌اس‌دي منتشر شده است.

علاوه براين مساله, تمام حقوق کپي رايت نرم افزار متعلق به اعضاي گروه پياده ساز مي باشد.

**2. ويژگي هاي اصلي**

**2.1. موارد فني**

امکانات اضافه اي که قرار است در پروژه قرار داده شود شامل موارد زير است:

* امکان چت کردن کارفرما با پيمانکاران پروژه به صورت مجزا
* امکان به اشتراک گذاري فايل بين کارفرما و پيمانکاران
* امکان اديت پروفايل کاربري
* امکان آپلود رزومه پيمانکاران
* امکان رتبه دادن به کارفرما و پيمانکاران
* امکان شکايت کردن  از طرفين پروژه در صورت ناراضي بودن از کار
* امکان تقسيم کار و مديريت زمان و هزينه  توسط کارفرما
* امکان مشاهده روند اجرايي پروژه به صورت آماري
* فرستادن ايميل پروژه هاي جديد به صورت هفتگي به پيمانکاران سايت بر اساس مهارت هاي وارد شده
* امکان شکايت کردن  از طرفين پروژه در صورت ناراضي بودن از کار
* امکان امتياز بندي براي کافرمايان و پيمانکاران در صورت دعوت کردن افراد ديگر يا به اشتراک گذاري سايت
* اضافه کردن مهارت ها يا رشته ي تحصيلي و فعاليت هاي کاري

**2.2. زمانبندي**

طبق بررسي ها وصحبت هاي انجام شده با اعضاي گروه پياده سازي و تيم تحقيقاتي در

نهايت تيم به اين نتيجه رسيد که کليه عمليات پياده سازي پروژه ظرف مدت 3 ماه انجام

شود که اين 3 ماه شامل موارد زير است :

1) ماه اول شامل طرح ريزي براي فاز اول پروژه شامل طراحي کامل پروژه من جمله

طراحي مدل ها و کلاس هاي اجرايي پروژه , طراحي ديتابيس و معماري اصلي پروژه ,

تعيين روند کلي اجرايي اجزاي پروژه , مشخص کردن عملکرد تمامي فانکشن ها و

جزئيات کلاس ها , تعيين تمامي ويژگي ها و مشخصات اکشن هاي کاربرها , تعيين روند

اجرايي مراحل انجام پروژه توسط کارفرما يا پيمانکار , تعيين تواناييهاي کاربرهاي پروژه

به صورت دقيق و با تمام جزئيات

2) ماه دوم شامل پياده سازي سطح پايين موارد مطرح شده در معماري است .

مثل پياده سازي ديتابيس و مدل ها وکلاس هاي entity , پياده سازي فايلهاي اصلي سرور

شامل فايلهاي اجرايي کنترلرها ,پياده سازي فايلهاي مربوط به backend , پياده سازي

فايلهاي اجرايي سرور , پياده سازي تمامي کلاسها وفرم هاي لازم , مشخص کردن تمام

تنظيمات برنامه شامل تنظيمات سرور و شبکه , پياده سازي فرم ها و ظاهر گرافيکي ساده

براي آنها

3) ماه سوم پياده سازي شامل پياده سازي جزييات متکي بر ظاهر پسند بودن پروژه است

که شامل پياده سازي دقيق تر جزييات اکشن ها براي کاربرها , پياده سازي امکانات فرعي

براي سيستم , پياده سازي جزييات و کاربر پسند کردن پروژه , بررسي دوباره و تست

فانکشن ها و اکشن هاي پروژه ,

بعد از عمليات پياده سازي پروژه مدت 1 ماه به عمليات تست پروژه اختصاص داده ميشود

که شامل تست کردن تمامي امکانات پروژه شامل تست منطقي تمام فايلهاي پروژه , تست

اکشن ها و عملکرد هاي اجرايي کاربران , بازبيني دوباره اشکالهاي ريز پروژه يا همان

debug کردن پروژه , بررسي اشکال هاي نحوي پروژه به همراه رفع مفاهيم گنگ و

مبهم, ست کردن exception handeling هاي ست نشده در مرحله هاي قبل ,

سپردن پروژه به کاربران براي تست و نظر دهي دوباره

بعد از کامل شدن مراحل ذکر شده ميتوان پروژه را به صورت ورژن بتا پروژه به بازار

معرفي کرد که خود شامل تهيه امکانات سخت افزاري و شبکه اي براي پروژه و ارائه

تبليغات براي پروژه مي شود .

اين مرحله زمان زيادي لازم دارد تا محصول بتواند جاي خود را در بازار هدف ثابت کند

که زمان در نظر گرفته براي آن 3 ماه ميباشد که تبليغات و بهينه سازي براي موتورهاي

جست و جو هم شامل آن مي باشد.

در اين مرحله عمليات تست در اسکيل بزرگتر انجام ميشود و کابران بازار هدف با ارتباط

برقرار کردن با نرم افزار ميتوانند اشکالات نرم افزار را مشخص کرده يا با نظر گرفتن

از آنها ميتوان نرم افزار را ارتقا داد.

بعد از ورژن بتا نوبت به پياده سازي ورژن آلفا ميرسد.

اين قسمت از پياده سازي مدت 1 ماه طول کشيده خواهد شد . که با استفاده از اشکالات

مشخص شده در ورژن بتا و نظرات کاربران اين پروسه انجام خواهد پذيرفت .

در پايان اين مرحله محصول نهايي و بدون نقص به بازار وارد خواهد شد .

بعد از بيرون آمدن ورژن نهايي مرحله پشتيباني نرم افزار شروع مي شود .

اين مرحله شامل در ارتباط بودن با کاربران نرم افزار از طريق راه هاي ارتباطي شامل

چت, ايميل يا تماس تلفني مي باشد.

اين مرحله نيازمند يک تيم پشتيباني مستقل است که عمليات کمک و راهنمايي يا اطلاع

رساني به کاربران را بر عهده ميگيرند.

زمان اجراي اين مرحله بستگي به زمان اجرايي خود نرم افزار دارد که ممکن است تا

چنديدن سال هم طول بکشد در واقع تا موقعي که نرم افزار وجود دارد تيم پشتيبان هم بايد

وجود داشته باشد.

**2.3. ابزارهاي مورد استفاده**

1. استفاده از يک ويرايشگر کد براي نوشتن سورس کد پروژه و اصلاح آن (Eclipse)
2. فريم ورک جنگو (Django Framework)
3. استفاده از پايگاه داده (MySQL)

**سامانه مديريت محتوا**

سامانه مديريت محتوا به انگليسي: Content Management System و به اختصار CMS مجموعه‌اي از رويه‌ها براي مديريت روند کار در يک محيط مشارکتي است. اين رويه‌ها مي‌توانند دستي يا رايانه‌اي باشند. اين رويه‌ها طراحي مي‌شوند که:

* افراد زيادي بتوانند مشارکت کنند و داده‌هاي مختلفي را به اشتراک بگذارند
* داده‌ها داراي کنترل دسترسي براساس نقش کاربران باشند. نقش کاربران تعيين مي‌کند که چه کاربري چه چيزي را مي‌تواند ويرايش کند يا ببيند
* ذخيره‌سازي و بازيابي داده‌ها را آسان کند
* ورودي‌هاي مشابه تکراري را کاهش دهد
* گزارش‌نويسي را بهبود بخشد
* ارتباط بين کاربران را بهبود بخشد

در يک CMS ، داده مي‌تواند هر چيزي باشد، مانند اسناد، فيلم، عکس، شماره تلفن، داده‌هاي علمي و غيره.

CMS ها بيشتر براي ذخيره، کنترل، بازبيني، توسعه معنايي و انتشار اسناد به‌کار مي‌رود.

Django نوعي CMS است.

**Django چيست ؟**

[](http://www.mspsoft.com/wp-content/uploads/2011/08/django-logo-positive1.png)

**جنگو** به انگليسي Django يک چارچوب نرم‌افزاري تحت وب [آزاد و متن‌باز](http://fa.wikipedia.org/wiki/%D8%A2%D8%B2%D8%A7%D8%AF_%D9%88_%D9%85%D8%AA%D9%86%E2%80%8C%D8%A8%D8%A7%D8%B2) است که به زبان پايتون نوشته شده است و از معماري (مدل-نما-کنترل‌گر) به انگليسي Model–view–controller پيروي مي‌کند.

**جنگو** اولين بار براي مديريت صفحات خبري وبسايت شرکت Lawrence Journal-World توسعه داده شد و سپس در سال ۲۰۰۵ تحت اجازه‌نامه بي‌اس‌دي منتشر شد. نام جنگو ازجنگو راينهارت، نوازنده گيتار جاز گرفته شده است.

هدف اصلي جنگو ساخت آسان سايت‌هاي پيچيده و وابسته به ديتابيس است و بر پايه قابليت استفاده مجدد و **قابل اتصال بودن** اجزاي مختلف، توسعه سريع و اصل خودت تکرار نکن (DRY) طراحي شده است. جنگو سراسر از پايتون استفاده مي‌کند، حتي براي تنظيمات، فايل‌ها و مدل‌هاي اطلاعات ، يکي از نکات قوت اين سيستم توانايي آن در ساخت سايت ها و برنامه هاي کاربردي وب با ساختار پايگاه داده پيچيده مي باشد .

 اين فريم ورک سطح بالا به زبان پايتون براي وب مي باشد که امکان طراحي و ايجاد بسيار سريع و آسان برنامه هاي تحت وب را فراهم مي کند .

**پروانه بي‌اس‌دي** به انگليسي BSD License مخفف پروانه نرم‌افزار توزيع برکلي است. از جمله پراستفاده‌ترين مجوزهاي نرم‌افزارهاي آزاد است.اين مجوز اولين بار براي توزيع برکلي به طور خلاصه نسخه‌اي از يونيکس است استفاده شده‌ است.

**جنگو يک فريم ورک سطح بالا با زبان پايتون مي باشد**

جنگو باعث سهولت ايجاد وب سايت هايي دايناميک مي شود . بسياري از موارد برنامه نويسي را بصورت خودکار فراهم کرده و در اختيار برنامه نويس قرار مي دهد . همچنين روش هايي ميانبر و واسط براي اجراي اعمال مختلف را دارا مي باشد . پس برنامه نويس نيازي به نوشتن کد هايي اضافي و مشخص را ندارد .

يک وب سايت داياناميک و پويا وبسايتي هست که شامل کد هاي ساده Html نبوده بلکه بصورت پويا و توسط نرم افزار

(web application) تگ هاي اچ تي ام ال و در حقيقت صفحات سايت از روي کد ها و دستورات نوشته شده توسط برنامه نويس ايجاد مي گردد . براي مثال ممکن است بر اساس در خواست کاربر اطلاعاتي بصورت دايناميک از پايگاه داده دريافت و نمايش داده شود .

**نکات کليدي**

جنگو مجموعه اي از کتابخانه هاي نوشته شده در زبان برنامه نويسي پايتون است، براي گسترش سايت مورد نظر با جنگو بايد کد پايتون زد که از کتابخانه هاي jango استفاده مي کند.

در يک کلام : جنگو = نوشتن برنامه به زبان پايتون + کار با کتابخانه هاي جنگو

در هر زمان دو نسخه از جنگو در دسترس است :

1. انتشار آخرين نسخه رسمي (تست شده و ثابت از جنگو)
2. نسخه .trunk (حاوي آخرين تغييرات و خصوصيات)

بسته به کار مورد نظر هر يک از نسخه ها انتخاب مي شود اما ما در اين پروژه از آخرين نسخه رسمي استفاده مي کنيم.

در سيستم عامل ويندوز فايل Django2.0.zip را از حالت فشرده خارج کرده و در محيط cmd دايرکتوري که با نام ... Django - 2.0 شروع مي شود مي نويسيم :

Python setup.py install

فايل هاي جنگو درون دايرکتوري site – packages نصب مي شوند . يک ديکشنري جايي که در آن جا کتابخانه هاي third – party را جستجو کند معمولا در جايي مثل آدرس زير قرار دارد :

./usr/lib/python2.7/site – packages

**ويژگي هاي يک فريم ورک خوب عبارتند از** :

**مکانيسمي براي نگاشت URL درخواست شده به کدهايي که وظيفه پاسخ دادن به آن درخواست را دارند**

به عبارت ديگر امکان تعيين اينکه کدام کد يا تابع مسئول جواب دادن به آدرس درخواست شده مي باشد . براي مثال مي توان از فريم ورک خواست تا براي آدرسهايي شبيه به /users/joe/ کدي را اجرا کند که اطلاعات کاربران را نمايش مي دهد .

**تسهيل نمايش ، اعتبار سنجي و نمايش مجدد فرم هاي html**

فرم هاي اچ تي ام ال يکي از مهترين روش ها براي دريافت اطلاعات از يک کاربر وب مي باشد . پس يک فريم ورک خوب بايد داراي روش هايي مناسب براي نمايش و کار با اين فرم ها باشد . (براي مثال امکان مشخص کردن خطاهاي رخ داده)

**تبديل اطلاعات وارد شده توسط کاربر به ساختاري مناسب براي مديريت و استفاده**

براي مثال تبديل اطلاعات وارد شده در يک فرم اچ تي ام ال به ساختاري متناسب با زبان مورد استفاده

**امکان جداسازي محتوا از ظاهر نمايشي**

در اکثر زبانهاي موجود برنامه نويسي کد هاي اچ تي ام ال و در حقيقت ظاهر يا template سايت با کد ها و محتواي سايت آميخته مي گردد که باعث سختي و ايجاد مشکلاتي هنگام تغييرات بعدي و نگهداري مي گردد . با استفاده از اين روش ظاهر نمايشي سايت بصورت جداگانه در فايلي خاص ذخيره مي گردد . اکنون با اعمال تغيير در هر کدام نيازي به ويرايش ديگري نيست و اين دو مجوديتي مستقل مي باشد .

**امکان دروني سازي آسان با لايه اطلاعات ذخيره شده (storage)**

در حالت معمول هر برنامه نوشته شده با جنگو داراي ۳ لايه مهم مي باشد بخش نمايشي (template)، محتوا يا کد هاي کنترلي و بخش ذخيره دائمي اطلاعات(براي مثال پايگاه داده) . البته بسته به نوع برنامه و خواست برنامه نويس مي توان اين لايه ها را ادغام کرده يا ناديده گرفت .

**امکان استفاده موثر از سطح بالايي از تجريد و انتزاع**

جنگو در موارد زيادي با استفاده از مفهوم انتزاع سهولت بسياري را فراهم کرده است . براي مثال براي کار با تکنولوژي چون ftp يا http با يک مفهوم انتزاعي و سطح بالا روبرو خواهيد بود که با استفاده از روابط و توابع متعدد برنامه نويسي را بسيار آسان و قدرتمند مي کند . البته امکان نفوذ و استفاده از سطوح پايين تر نيز فراهم است .

**حذف موارد و مشکلات معمول در برنامه نويسي وب**

حذف و رفع موارد و مشکلات معمول در برنامه نويسي سمت وب . براي مثال حذف پسوند فايل از آدرس هاي وب (.aspx) يا (.php)  
فريم ورک django تمامي موارد فوق را دارا مي باشد . اين فريم ورک با استفاده از زبان زيبا ، قدرتمند و سطح بالاي پايتون نوشته شده است . براي ايجاد وب سايت برنامه نويس کدهاي پايتوني مي نويسد که امکان استفاده از ويژگي هاي پايتون و کتابخانه هاي اين زبان را دارد .

**تاثير پايتون بر اين فريم ورک**

کل اين فريم ورک با استفاده از زبان پايتون پياده سازي شده است . پس بسياري از ويژگي هاي خود را از زبان پايتون به ارث برده است . با استفاده از اين فريم ورک امکان ايجاد وب سايت هايي حرفه اي و پيچيده در کمترين زمان ممکنه و همچنين با در نظر گرفتن مورادي چون امنيت و سرعت و … فراهم مي باشد . در ادامه برخي از ويژگي هاي قابل تاکيد زبان پايتون که تاثير مهمي بر اين فريم ورک دارند مطرح مي گردد .

* پايتون زباني تفسيري بوده و براي اجرا نيازي به کامپايل ندارد . در برنامه نويسي وب با استفاده از اين ويژگي بعد از تغيير کد يا ايجاد آن نتايج کار بلافاصله قابل مشاهده مي باشد .
* انواع داده در پايتون دايناميک مي باشد . پس شما نگراني از تعريف نوع متغيير نداريد .
* ساينتکس زبان پايتون کوتاه و در عين حال واضح و قابل فهم مي باشد . اين بدين معني هست که براي انجام کار هاي مشابه کدي بسيار کمتر لازم مي باشد . براي مثال معمولا هر خط پايتون معادل ۱۰ خط در جاوا (Java) مي باشد !!
* پايتون روشهايي قدرتمند براي meta-programming در اختيار قرار مي دهد . اين ويژگي امکان ويرايش با اضافه کردن رفتار ها و توابع را با اشيا در زمان اجراي برنامه فراهم مي کند .

جدايي از ويژگي هاي فوق جنگو خود روشهايي براي افزايش سرعت و سهولت برنامه نويسي همزمان با رعايت کامل موارد امنيتي و کارايي را دارا مي باشد . که در ادامه با برخي از اين ويژگي ها بيشتر آشنا خواهيم شد.

**ايجاد وب سايتي با طرح و ساختار درست و تميز**

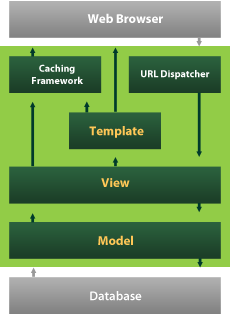
جنگو با استفاده از قوانين از پيش تعيين شده و روش هايي مختلف باعث ايجاد برنامه هايي با سرح و ساختار درست و تميز مي گردد . بنابراين نگهداري و توسعه برنامه هاي نوشته شده با اين فريم ورک بسيار آسان تر خواهد بود .

جنگو از ايجاد برنامه هايي با آميختگي و روابط نادرست جلوگيري مي کند . با استفاده از اين ويژگي، برنامه به بخش هايي مستقل و مجرا تقسيم مي شود که هر يک بدون نياز به اعمال تغييرات در بقيه قسمت ها قابل تغيير بوده با استفاده از روابط (API) واضح و بهينه با يکديگر در ارتباط هستند . براي مثال سيستم ظاهر نمايشي (templates) هيچ گونه اطلاعاتي از سيستم پايگاه داده و ذخيره اطلاعات يا لايه مربوطه به پاسخگويي به درخواست هاي کاربران يا نهان سازي اطلاعات براي استفاده هاي بعدي (cache) ندارد . هر کدام از اين بخش ها بصورت جداگانه بوده و در صورت نياز با استفاده از رابط هايي با يکديگر تبادل اطلاعات کرده يا درخواستي را ارسال مي کنند .

جنگو از ساختار MVC يا Model-View-Controller تبعيت مي کند . پس کد هاي مربوط به کار با داده ها (model) و بخش کنترلي (controller) يا bussiness logic و بخش مربوط به رابط کاربر (view) از هم جدا مي باشد . براي روشن شدن هرچه بيشتر موضوع اين ويژگي با استفاده از يک مثال شرح داده مي شود .

**آميختگي ظاهر نمايشي با کد برنامه**

در تکنولوژي MVC که جنگو نيز از آن بهره مي برد ظاهر نمايشي (يعني تگ هاي اچ تي ام ال) يا template سايت در فايلي جدا ذخيره مي گردد . بخش کنترلي نيز به عنوان ماژول پايتون ايجاد و ذخيره مي گردد . اکنون برنامه نويس با بخش کنترلي و طراح با بخش html سروکار خواهد داشت .

[](http://www.mspsoft.com/wp-content/uploads/2011/08/Django_overview21.png)

**آميختگي کد مربوط به پايگاه داده با کدهاي کنترلي**

علاوه بر مشکل فوق اين مورد باعث افزونگي نيز مي گردد . اگر براي مثال شما نام پايگاه داده را تغيير دهيد بايد دستور sql را تغيير دهيد و ممکن است اين تغيير در بخش هاي مختلفي صورت گيرد . با استفاده از MVC يک لايه واحد و انتزاعي از محل ذخيره سازي اطلاعات ايجاد مي گردد که مسئول انجام کليه درخواست هاي مربوط به دسترسي به داده ها مي باشد . در حالتي که شما از پايگاه داده استفاده کنيد نيازي به نوشتن دستورات sql نبوده بلکه با استفاده از دستورات پايتون با اطلاعات و داده ها سروکار داريد . پس براي مثال اگر نام يک فيلد تغيير کند کافيست در يک جا نام آن را تغيير دهيد .

**کد نوشته شده با آدرس اينترنتي (URL) گره خورده است**

براي مثال اگر اين فايل php در مسيري چون foo/index.php/ باشد اين فايل براي تمامي درخواست هاي اين مسير اجرا خواهد شد . حال اگر بخواهيم همين فايل درخواست هاي مسير /bar/ را هم اجرا کند به مشکل بر مي خوريم ! در فريم ورک جنگو مسير اينترنتي غير وابسته به مسير فايل روي هارد ديسک سرور و کد مربوطه مي باشد . به عبارت ديگر با روشي خاص هر مسير را به کدي دلخواه در مسير دلخواه نگاشت مي کنيم . اکنون درخواست هاي مربوطه به آن مسير خاص به کد مربوطه ارجاع مي گردد .

**نوشتن پارامتر هاي مربوط به اتصال پايگاه داده و نوع پايگاه داده در فايل اصلي برنامه**

نوشتن اطلاعات مربوط به اتصال پايگاه داده در داخل کد اصلي برنامه صحيح نمي باشد . چون اين اطلاعات تنظيمات برنامه مي باشد نه برنامه نويسي منطقي برنامه . جنگو يک محل واحد براي ذخيره اطلاعات و تنظيمات برنامه دارد . پس براحتي مي توان تغييراتي چون تغيير نوع پايگاه داده را انجام داد .

موراد ذکر شده در اين کد PHP تنها مثالي براي بررسي موارد و مشکلات معمول بود . بطور حتم با زبان PHP و البته با کمي زحمت بيشتر و نوشتن بسياري از بخش ها بصورت دستي مي توان موارد فوق را پياده سازي کرد .

**بخش هاي مختلف Django**

**هسته مرکزي و اصلي**

هسته اصلي و مرکزي جنگو از سه بخش اصلي تشکيل شده است .

 همچنين بخش مرکزي جنگو شامل موارد ديگري مي باشد که عبرتند از :

* يک وب سرور مستقل و کوچک که براي تست برنامه هنگام توسعه آن استفاده مي گردد .
* يک سيستم براي معتبر سازي و serialization فرم هاي اچ تي ام ال .
* يک فريم ورک نهان سازي اطلاعات براي استفاده مجدد يا کش (cache) که روش هاي مختلف کش را در اختيار قرار مي دهد .
* حمايت از ابزار هاي مياني (middleware) که امکان اجراي توابع و دستورات مورد نظر را در بخش هاي مختلف پردازش يک درخواست فراهم مي کند .
* يک توزيع کننده (dispatcher) دروني که به بهش اي مختلف يک برنامه وب امکان دريافت سيگنا لها و رويداد هاي مختلف را مي دهد .
* سيستم بين الملل سازي (internationalization) که حتي اماکن ترجمه بخش هاي مختلف جنگو به زبان هاي مختلف را فراهم مي کند .
* يک سيستم براي تسلسل و سري سازي (serialization) که امکان کار با انواع داده هاي مبتني بر XML و JSON و در حالت کلي آژاکس (AJAX) را فراهم مي کند .
* سيستمي براي توسعه قابليت هاي موتور قالب (template)

**ساير بخش هاي جنگو**

بسته جنگو همچنين شامل ابزار ها و برنامه هاي جانبي مختلفي مي باشد که در داخل بسته contrib قرار دارند . برخي آز اين ابزار ها عبارتند از :

* يک سيستم تصديق و شناسايي کاربر (authentication) قابل توسعه
* يک رابط مديريتي دايناميک و پويا
* ابزار هايي براي ايجاد RSS و Atom
* سيستم نظر دهي (commenting) قابل انعطاف و توسعه
* ابزار هايي براي توليد Google Sitemaps
* ابزار هاي امنيتي براي جلوگيري از حملات cross-site request forgery
* کتابخانه هاي قالب (tamplate) که امکان استفاده از زبان هاي نشانه گذاري سبک و کوچک چون Textile و Markdown را فراهم مي آورد .

**سرور هاي قابل استفاده**

Django با استفاده از ماژول mod\_python بخوبي بر روي وب سرور آپاچي (apache) اجراي مي گردد . همچنين بر روي تمامي سروي هايي که از WSGI حمايت کنند (اکثر سرور هاي موجود) قابل اجرا مي باشد . همچنين جنگو توانايي راه اندازي سرور FastCGI را دارا مي باشد که بدين تريتب مي توان از جنگو در داخل هر وب سروري که از FastCGI پشتيباني مي کند استفاده کرد .

سرورهاي زير بطور رسمي پشتيباني مي شود و با استفاده از جنگو قابل اسفتاده مي باشد .

* PostgreSQL
* MySQL
* SQLite
* Oracle

ابزار و بخش هاي لازم براي پشتيباني از سرور SQL مايکروسافت نيز در حال توسعه مي باشد . )در مراحل آزمايشي قرار دارد (

**راه اندازي پايگاه داده :**

جنگو از چهار پايگاه داده زير پشتيباني مي کند :

PostgreSQL

SQLite 3

MySQL

Oracle

تمامي اين پايگاه داده ها با هسته جنگو به خوبي کار مي کنند. تنها در مورد پشتيباني از GIS جنگو پايگاه داده PostgreSQL پر قدرت از بقيه کار مي کند.

اين پايگاه داده از نظر سرعت و خصوصيت و هزينه و پايداري تعادل خوبي برقرار کرده است. در اين پروژه از اين پايگاه داده يا به احتمال قوي از پايگاه داده MySQL استفاده مي کنيم.

**جنگو با PostgreSQL :**

1- پکيج Psycopg يا نسخه 2 را از سايت زير نصب مي کنيم، نسخه 2 داراي نصبي آسان تر فعاليت بيشتر در توسعه و جديدتر است :

<http://www.djangoproject.com/r/python-pgsq/>

پايگاه داده براي نسخه ويندوز باينري هاي کامپايل شده در آدرس زير موجودند :

<http://www.djangoproject.com/r/python-pgsql/windows/>

**جنگو با MySQL :**

با MySQL 4.0 به بالا کار مي کند. نسخه 3.x subquery هاي تو در تو و تعدادي از جملات SQL را پشتيباني نمي کند، همچنين بايد پکيج MySQLdb را از سايت زير دريافت کرد :

<http://www.djangoproject/r/python-mysql/>

**سرور جنگو :**

براي بخش سرور اين پروژه از سرور خود جنگو به نام runserver استفاده مي کنيم.

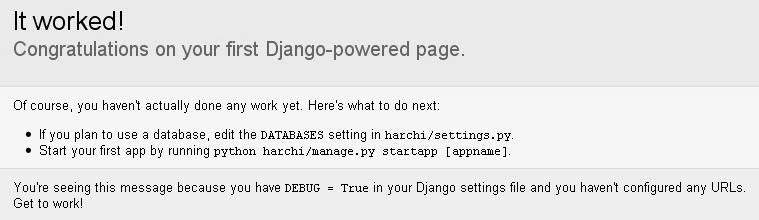
يک سرور سبک و داخلي است و سايت را با آن توسعه مي دهيم. بدون راه اندازي و تنظيمات سرور (Apache) سرعت سايت را افزايش داده و اگر در کد تغييري ايجاد کنيم به طور اتومات آن تغيير در سرور به روز رساني مي شود.

براي اجرا کردن سرور به دايرکتوري پروژه رفته (cd mysite) :

python manage.py runserver

دستور بالا سرور را به صورت داخلي روي پورت 8000 راه اندازي مي کند. تنها بر روي رايانه شخصي قابل دسترسي است.

اگر مسير رو به رو را بنويسيد در مرورگر عکس زير را مي بينيد : http://127.0.0.1:8000



**تغيير هاست يا پورت :**

براي اشتراک گذاري سايت با ديگر اعضا IP آدرس 0.0.0.0 به سرور مي گويد به هر رابط شبکه گوش کند.

python manage.py runserver 0.0.0.0:8000

**2.4. محدوديت ها**

سايت مورد پياده سازي قابليت دارا بودن امکانات و بخشهاي زيادي را دارد, اما به علت اهميت سرعت سايت و محدوديت منابع, تصميم گرفته شد که تعدادي از مهمترين امکانات مورد نياز پياده سازي شود.

با توجه به اينکه اقشار مختلف مردم با سطوح متفاوتي از آشنايي با اينترنت و وب مي توانند کاربر اين سايت باشند, طراحي سايت بايد به گونه اي باشد که درک بخشهاي مختلف سايت و استفاده از آن براي همه ممکن و آسان باشد.

به علت ذخيره سازي اطلاعات کاربران, انجام تراکنشهاي مالي در طول انجام پروژه ها در سايت و همچنين حفظ حق مالکيت پروژه هاي انجام شده, از نظر امنيتي نيز محدوديت هايي بوجود خواهد آمد که در طراحي سايت بايد مدنظر قرار بگيرد.

با توجه به امکان انجام گفتگو بين کاربران سايت(از طريق چت روم) , براي حفظ شئون اسلامي و رعايت قوانين جمهوري اسلامي ايران, بايد نظارت دقيقي بر اين قسمت از سايت وجود داشته باشد.

همچنين با توجه به هدف سايت, کاربران محدود به افرادي خواهند بود که توانايي انجام کارهاي گرافيکي داشته باشند و همچنين قسمت هاي مختلف سايت بايد محدود باشند. براي مثال قسمت ارائه ي پروژه بايد محدود به فيلد هايي باشد که بيانگر مشخصات يک پروژه ي گرافيکي باشد.

**2.5. پيش فرض**

اولين گام آماده کردن کامپيوتر براي کار، قبل از شروع پروژه است.  
قبل از اين که وب سايت خودتان را بسازيد، مي بايست کامپيوتر خود را براي کار آماده نماييد که شامل برنامه هايي است که يا در سيستم شما موجود مي باشد و يا اين که مي بايست آن ها را نصب نماييد.  
برنامه هاي زيادي در بازار موجود مي باشد ولي در ابتدا از برنامه هاي رايگان و قابل دسترس استفاده مي کنيم.  
وسايل و برنامه هاي اصلي که شما نياز داريد:

1. سيستم عامل تحت ويندوز (7 يا 8)
2. نرم افزار نمايش فايل هاي فشرده شده (Winrar or 7Zip)
3. يک مرورگر (web browser) براي ديدن صفحات وب در آن (Mozilla Firefox or Google Chrome)
4. وسيلۀ ابتدايي براي ويرايش فايل هاي متني که در واقع صفحه وب شما را شامل مي شود (Notepad++ يا هر TextEditor ديگري )
5. اما همه عناصر که متن نيستند شما براي ويرايش عکس هم به برنامه هايي نياز داريد که مي توانيد از برنامه هاي زير استفاده کنيد:  
   Adobe photoshop  
   Adobe fireworks  
   Picasa
6. نسخه نصب شده python 2.7

با توجه به اين که قرار است پروژه سايت با استفاده از فريم ورک Django پياده سازي شود (در قسمت ابزار هاي مورد استفاده به تفصيل در مورد فريم ورک مي پردازيم) اين فريم ورک با استفاده از زبان پايتون استوار شده است پس يکي از پيش فرض هاي سيستم ما زبان برنامه نويسي Python مي باشد.

اولين گام در نصب فريم ورک اطمينان حاصل کردن از نصب پايتون است.

پايتون نسخه 3.0 نيز منتشر شده است اما جنگو هنوز از آن پشتيباني نمي کند بنابراين نسخه مورد نظر ما براي اين فريم ورک python 2.7 مي باشد.

**2.6. متدولوژي**

متدولوژي مورد استفاده براي اين پروژه, روش آبشاري به همراه تغييرات جزئي به منظور تطبيق بهتر با مراحل ذاتي طراحي وب سايت خواهد بود.

دلايل استفاده:

1. اين نوع سايت قبلاً چند بار پياده سازي شده است.

2. نيازمندي ها مشخص هستند و تقريباً هيچ نياز مبهمي وجود ندارد.

3. منابع کافي و تجارب قبلي به صورت رايگان موجود هستند.

4.تکنولوژي هاي مورد نياز براي ساخت اين سايت مشخص هستند.

مراحل تعيين شده برای انجام پروژه با در نظر گرفتن قوانين مدل آبشاری در ادامه توضيح داده شده است:

**2.6.1. استخراج و تحليل نيازمندي ها**

اولين گام شناخت نيازمندي هاي کاربران است. شناسايي کاربران نهايي سيستم, گامي بسيار مهم به شمار مي آيد.

در اين فاز هدف کلي سايت و نحوه ي تعامل کاربران با آن مورد بررسي قرار مي گيرد. در اين مرحله شناخت صحيح کاربران سيستم از اهميت بالايي برخوردار است.

با استفاده از روش هاي موجود و رسم دياگرام هاي مختلف و تحليل دقيق نيازها, بايد نيازمندي ها ي کاربران استخراج و نحوه ي تعامل کاربران با يکديگر مشخص شود.

اين مرحله از اهميت بالايي برخوردار است و بايد زمان مناسبي براي آن اختصاص داده شود. به طوري که نه وقت زيادي صرف آن شود و نه به خاطر عجله و تخصيص زمان کم, برخي از نيازها به درستي شناسايي نشوند.

**2.6.2. تعيين مشخصات سايت**

در اين مرحله با استفاده از نتايج تحليل نيازمندي ها که در بخش قبل بدست آمد, روي طرح کلي سايت و قسمتهاي مختلف آن تصميم گيري مي شود.

در واقع در اين مرحله با استفاده از گزارش بدست آمده در مرحله ي تحليل نيازمندي ها و همچنين در نظر گرفتن زمان و هزينه و در مجموع منابع در دسترس, يک طرح کلي از سايت ايجاد مي شود.

در اين مرحله, يک پروپوزال کامل, درباره ي بخشهاي مختلف سايت و نحوه ي برطرف کردن نياز هاي کاربران توسط اين بخش ها تهيه خواهد شد.

**2.6.3. تعيين ساختار کلي**

در اين مرحله به توجه به قسمت هاي مختلف سايت در مرحله ي قبل تعيين شد, ساختار کلي کدها مشخص خواهد شد.

براي مثال, طراحي مدل ها و کلاس هاي اجرايي پروژه , طراحي ديتابيس, مشخص کردن عملکرد تمامي فانکشن ها و جزئيات کلاس ها و ... .

**2.6.4. نوشتن کدها**

بعد از تعيين ساختار کلي و سطح بالاي کدها, بخش اصلي ساخت سايت يعني پياده سازي کدها آغاز مي شود. در اين مرحله تمام کدهاي سايت از جمله کدهاي html, css, javascript, python و همچنين ساخت پايگاه داده و کوئري هاي آن انجام خواهد شد.

نوشتن کدها بايد طوري صورت پذيرد که منطبق با ساختار گرافيکي سايت باشد و آنرا دچار تغيير نکند.

البته به خاطر گستردگي اين بخش, مي توان آنرا به چند بخش کوچکتر تقسيم بندي نمود. براي مثال:

نوشتن کدهاي سمت کاربر

ساخت پايگاه داده

نوشتن کدهاي کنترلي براي ارتباط کاربر و پايگاه داده

...

**2.6.5. طراحي گرافيکي**

در اين مرحله, با توجه به قسمتهاي مختلف سايت که در مرحله ي قبل تعيين شد, قالب و گرافيک صفحات سايت طراحي مي شود.

براي ايجاد قالب سايت موارد مختلفي از جمله نوع کاربران سايت, نوع فعاليت سايت و ... بايد درنظر گرفته شود, تا طراحي بخش هاي مختلف سايت از جمله لوگو, منوها, هدر و همچنين رنگ بندي قسمتهاي مختلف بدرستي و با توجه به هدف سايت انجام شود.

2.6.6. **افزودن محتوا**

در اين مرحله محتواي مرتبط با سايت به قالب طراحي شده اضافه مي گردد.رعايت نکات گرامري نکته ي مهمي براي درک قسمتهاي مختلف سايت توسط کاربران مي باشد.

نوع نگاشت محتواي سايت براي موتورهاي جستجو بسيار حائز اهميت است. موتورهاي جستجو به کلمات، جمله بندي و بسياري موارد ديگر از محتوا اهميت ويژه اي مي دهند.

در اين مرحله دو نوع محتوا به سايت اضافه خواهد شد: محتواي متني و محتواي چند رسانه اي با محتواي متني به راحتي مي توان اطلاعات اوليه مورد نياز مشتريان و مخاطبان را مرتفع نمود.

در استفاده از محتواي چند رسانه اي مي بايست ابتدا همه آنها را جهت بالا بردن سرعت نمايش در وب سايت بهينه کنيم.

**2.6.7. تست سايت**

در اين مرحله وب سايت طراحي شده از ابعاد مختلف و با استفاده از ابزارهاي مناسب تست خواهد شد و کارهاي لازم براي افزايش کارايي سيستم انجام خواهد شد.

براي مثال با توجه به سرعت اينترنت در ايران, سرعت لود قسمتهاي گرافيکي سايت, از اهميت بالايي برخوردار خواهد بود.

همچنين وب سايت از نظر امنيت مثلاً امنيت دسترسي به پايگاه داده بايد مورد آزمايش قرار بگيرد.

همچنين در اين مرحله سعي خواهد شد مواردي که اضافي هستند و وجود آنها تاثير چنداني در بهبود عملکرد سايت ندارد حذف شود.

همچنين کمبود ها و نقص هاي سايت شناسايي خواهند شد و تا حد امکان برطرف خواهند شد.

ابزارهاي مختلفي براي تست سايت از جمله ابزارهاي تست آنلاين براي اين منظور موجود مي باشد.

**2.6.8. SEO و ارتقا، سايت**

در اين مرحله با استفاده از روش هاي افزايش رتبه, از جمله استفاده ي مناسب از تگ هاي متا, استفاده از شبکه هاي اجتماعي و ساير روشهاي مناسب, سعي در معرفي هرچه بهتر سايت به کاربران اينترنت خواهد شد.

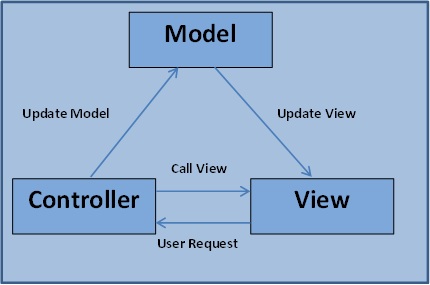
**2.6.9. آپديت و پشتيباني سايت**

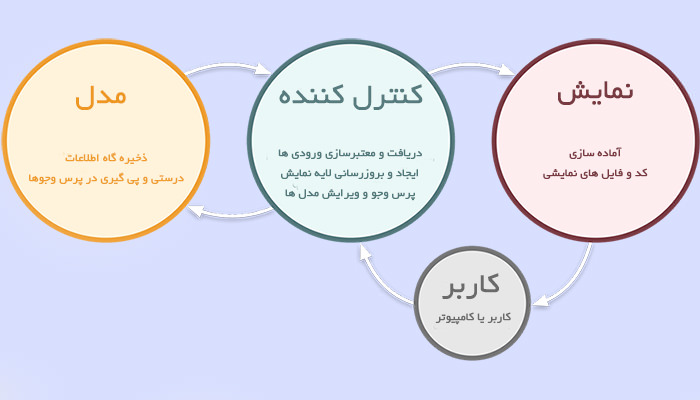
يک وب سايت همواره نياز به بروزرساني و تغيير دارد تا با شرايط و نيازهاي کاربران وفق پيدا کند. بعد از ساخت سايت و شروع به استفاده از آن, نياز داريم تا همواره کارايي سايت را بهبود دهيم و باگ هاي احتمالي را کشف و برطرف کنيم.

**2.7. معماري مورد استفاده**

* معماري نرم افزار چيست ؟
* معماري نرم افزار ، ساختار يک نرم افزار را بصورت کلي و سطح بالا بيان ميکند. يعني در انتخاب معماري ، جزئيات ريز و نحوه ي پياده سازي مورد بررسي قرار نمي گيرد.
* هر معماري شامل يک سري اجزاء (Component) بوده که اين اجزاء از طريق يک سري اتصالات (Connector) با هم در ارتباطند. هر کدام از اجزاء درون معماري مي تواند يک وظيفه ي مشخص داشته باشد.
* هر نرم افزاري که پياده سازي شده ، از يک معماري خاص تبعيت کرده است. حتي اگر طراح آن متوجه معماري آن نباشد و يا جزو معماري هاي شناخته شده نباشد!
* برخي از معماري هاي شناخته شده عبارتند از :
* - معماري 3 لايه
* - معماري Sense Compute Actuator
* - معماري Client - Server
* - معماري Peer To Peer
* - معماري MVC
* و ...
* يک نرم افزار مي تواند در آن واحد از چند معماري بصورت همزمان استفاده کند!

**بررسي معماري نرم افزاري MVC –**



****

نياز روز افزون به کامپيوتر و مکانيزه کردن و سپردن تقريبي تمامي امور به دست ماشين امري انکار ناشدني است. در اين بين توليد کنندگان نرم افزار نيز تلاش مي کنند تا نرم افزاري توليد کنند که بتواند اکثر نياز هاي متقاضيان را به بهترين نحو ممکن تامين کند؛ در همين راستا در تلاش هستند که روند توليد نرم افزار را به سمتي بکشانند که ساختار استاندارد و تائيد شده اي داشته باشد.

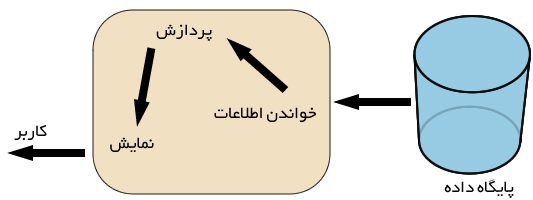
شايد اينطور بتوان گفت که دوران کد نويسي به پايان رسيده و همه چيز به سمت زير ساخت ها و بنيان نهادن چارچوب هاي استاندارد و پيروي از آن ها در امر توليدِ بهتر نرم افزار در حرکت است.

سوالي که اکنون پيش مي آيد اين است که خصوصيات يک نرم افزار خوب چيست؟

* قابل حمل بودن
* قابل استفاده مجدد بودن
* قابل تغيير بودن
* بهينه بودن از لحاط حافظه و زمان

فرض کنيد نرم افزاري براي شرکتي نوشتيد که يک بخش از آن مقدار سود و زيان شرکت را در سال هاي مختلف بر اساس ارقام بيان مي‌کند. حال مدير شرکت پس از مدتي ازشما مي خواهد برنامه را طوري تغيير دهيد که همين اطلاعات را به گونه هاي مختلف ديگري مانند نمودارهاي مختلف (ميله اي، دايره اي و …) در اختيار داشته باشد و يا حتي بخواهد آنها را به فرمت خاصي و در فايل هاي خاصي ذخيره کند؛ در اين مواقع چطور مشکل را حل مي کنيد؟

همانطور که در تصوير1 نيز نشان داده شده است تمامي اعمال، از دريافت داده ها که مهمترين بخش است و همچنين پردازش آن‌ها تا نمايش آن‌ها به کاربر همگي در يک فرم طراحي و پياده سازي شده اند و دقيقا مشکل همين جا نمايان مي‌شود. ارتباط مستقيم با منبع داده بر قرار کردن، جدا از اينکه مشکلات امنيتي دارد که صحبت در مورد آن خارج از اين مبحث است، باعث مي‌شود که دست برنامه نويس را براي تغييرات آتي دربرنامه ببندد. چون داده ها درون خود فرم از منبع داده و به صورت مستقيم خوانده مي‌شوند پس دسترسي به داده هاي خوانده شده ديگر وجود ندارد يا حداقل متحمل سربار زيادي مي باشد.

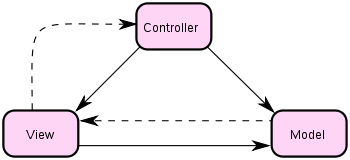


**تصوير 1: خواندن و نمايش اطلاعات از پايگاه داده**

يکي از راه کار هايي که امروزه بيشتر شاهد استفاده از آن هستيم، توليد نرم افزار بر اساس ساختار هاي لايه اي مي باشد. بدين صورت که کل نرم افزار به تعداد لايه هايي تقسيم مي‌شود، هر لايه وظيفه خاص خود را دارد و لايه ها از نتايج لايه هاي ديگر استفاده مي کنند و از نظر ساختار جدا از يکديگر هستند.

تعداد اين لايه ها بسته به نرم افزار و طراحي مي تواند 2، 3، 4 يا 5 لايه يا حتي بيشتر باشد. اما استاندارد آن که بيشتر از بقيه هم استفاده مي‌شود 3 لايه است و به روشي که بر اساس اين تئوري پياده سازي مي‌شود اصطلاحاTire Programming 3 گفته مي‌شود.

MVC مخفف سه کلمه Model، View و Controller است. در واقع هدف از معماري نرم افزاري MVC ايجاد معماري چند لايه اي جهت جداسازي قسمت هاي مختلف برنامه است. به طور دقيق‌تر MVC باعث جدا شدن بخش هاي منطقي برنامه که شامل اطلاعات، سطح دسترسي ها، چک کردن صحت داده ها از لايه نمايش (presentation) يا در واقع همان لايه اي که مستقيما با کاربر نهايي (end-User) در ارتباط است، مي‌شود. (تصوير 2)



**تصوير 2: معماري سه لايه MVC**

پس بر اساس توضيحات فوق ميتوان هر يک از بخش هاي معماري MVC يعني Model ، View و Controller را معرفي کرد.

**معرفي مدل، نما و کنترلگر**

معماري MVC به نوعي از معماري در مبحث برنامه نويسي گفته مي شود که تاکيد بسيار به لايه بندي سيستم دارد. در اين جلسه و در ادامع بررسي معماري نرم افزاري MVC مي خواهيم سه بخش اصلي در معماري نرم افزاري MVC را بررسي مختصري داشته باشيم.

**مدل**

بار اصلي معماري MVC بر عهده اين بخش است. مدل ميتواند با داده ها در ارتباط باشد. الزاما منظور از داده حتما ارتباط با پايگاه هاي داده همچون MYSQLو MSSQL نيست، حتي منبع داده ها در بخش Modelميتواند يک آرايه از اعداد و يا هر چيز ديگري باشد. Model وظيفه چک کردن داده ها جهت صحت درستي آنها را هم بر عهده دارد.

**نما**

نما وظيفه‌ي بر قراري ارتباط با کاربر نهايي و گرفتن داده از کاربر و نمايش داده هاي آماده به کاربر از طريق برقراري ارتباط با دو بخش ديگر يعني Model و Controller را دارد. در واقع نکته مهمي که در بخش View بايد آن را مد نظر داشت اين است که اين لايه مسئول کنترل صحت داده هاي وارد شده از طريق کاربر و همچنين مسئول صحت داده هاي نشان داده شده به کاربر نيست و در واقع اين بخش با داده هاي خام کار مي‌کند.

به عنوان يک مثال ساده، بسياري از برنامه نويسان براي زماني که در فرم Login برنامه، کاربر کلمه عبور خود را وارد مي‌کند، در همان فرم Login اقدام به چک کردن پسورد مبني بر صحت آن و ديگر موارد مورد نظرشان مي کنند؛ که اين عمل در معماري MVC قابل قبول نيست! در واقع براي حل اين مسئله در معماري MVC، در فرم Login، هنگامي که کاربر کلمه عبور را وارد کرد و دکمه Login را زد، کلمه عبور داده شده بدون هيچ گونه اعمالي مانندرمزگذاري **(encrypt)** کردن و . . . به بخش هاي ديگر فرستاده مي شود و فقط يک نتيجه ساده مبني بر اين که کاربر اجازه ورود دارد يا خير را از بخش هاي ديگر دريافت مي‌کند که بر اساس آن اجازه ورود کاربر به برنامه داده مي شود.

**کنترلگر**

در اين لايه درخواست هاي کاربر پردازش مي‌شود و به قسمت هاي مربوطه هدايت مي‌شود. اين لايه، واسطه‌ي ارتباط بين لايه‌ي نما و مدل است، بدين صورت که درخواست هاي کاربر از طريق اين لايه به مدل ارسال مي‌شود.

**روند اجراي برنامه در MVC**

در معماري MVC روند کلي برنامه (تصوير زير) به اين شکل است که کاربر تقاضاي خود را از طريق واسط هاي برنامه نويسي نظير Formها، Linkها و … از بخش View درخواست مي‌کند. بخشView درخواست ها را به بخش Controller فرستاده و اين بخش با برقراري ارتباط با بخش Model در خواست هاي کاربر را پردازش کرده و پس از پايان پردازش زماني که خروجي درخواست داده شده آماده گرديد، بخش Controller بخش View را آگاه مي سازد تا خود را بر اساس تغييرات جديد به روز سازد.



**2.8. انواع روش هاي تست نرم افزار**

**2.8.1. تست عملکرد (WHITE BOX TESTING):**

تست نرم افزار از نظر درستي عملکرد , در اين تست کامپوننتها و فرآيندهاي مورد تست قرار مي گيرند .

اين استراتژي با تست کدهاي نرم افزاري و ساختمان داخلي نرم افزار سر و کار دارد .

در اين روش کدهاي بايد به گونه اي اجرا و بررسي شود که مطمئن شويم سطر به سطر کد هاي برنامه حداقل يکبار اجرا شده است .

در کد برنامه بايد تمام عبارتهاي شرطي(if else ,switch case) را تک تک مورد بررسي قرار دارد بدين صورت که در يک عبارت if else هم قسمت if و هم قسمت else هر کدام به صورت مجزا يکبار اجرا شود، حلقه ها در نرم افزار نقش اساسي دارند چون مي توانند با اشتباه جزئي مقدار زيادي از منابع را مصرف کرد و براي مثال شرط خروج از حلقه به اشتباه هيچ وقت true نشود براي نمونه حلقه را براي ورودي بزرگتر تست کنيد .

**2.8.2. تست استرس (STRESS TESTING) :**

در اين تست برنامه در مقابل بار سنگيني مانند مقادير عددي پيچيده , مقادير زياد ورودي و مقادير زياد پرس و جو امتحان ميشود.

فشار همزمان چند ايستگاه مختلف براي تست حافظه، از ديگر نمونه هاي اين نوع تست مي باشد.هدف از اين تست طراحي محيطي است که مخرب تر از محيطي که برنامه در دنياي واقعي و در شرايط نرمال با آن روبرو مي شد .

**2.8.3. تست اکتشافي :**

در اين روش فرض مي شود شخصي که هيچ گونه دانشي در خصوص نرم افزار ندارد در حال کار با سيستم است و به روشهاي غير معمول نرم افزار را مورد تست قرار مي دهد.

اين روش يک روش جالب و قوي براي تست نرم افزار است.

**2.8.4. تست تطبيق پذيري با محيط :**

در اين تست، نرم افزار در سيستم عاملهاي مختلف يا مرورگرهاي مختلف و کالاً زير ساختهاي مختلف تست مي شود وميزان تطبيق پذيري آن با ساير محيط ها مشخص مي شود .

**2.8.5. تست امنيت :**

اين نوع تست مي تواند شامل تست محرمانه بودن،جامعيت،تصديق هويت،مجوز کاربري

مي باشد .

**2.8.6. تست واحد (Unit Test):**

اين تست معمولا توسط خود برنامه نويسان انجام مي گيرد و تک تک ماژولها را در حين پياده سازي تست مي کنند.

**2.8.7. آزمون تحمل خرابي :**

اين تست مي تواند شامل موارد زير باشد :

اختلال در محيط سخت افزاري(قطع ناگهاني برق – خرابي ديسکهاي دستگاه سرويس دهنده يا ايستگاه کاري – قطع اتصالات شبکه داخلي)

روش هاي مختلفي وجود دارد که بتوان يک نرم افزار را مورد ارزيابي عملکرد قرار داد به عنوان مثال کار پيوسته با سيستم بدون خارج شدن از نرم افزار ( حداقل 72 ساعت)، استفاده فعال همزمان از نرم افزار توسط چند کاربر، بررسي کارکرد نرم افزار با استفاده از حداکثر گنجايش بانک اطلاعاتي به اين منظور جدول بايد حاوي حداکثر تعداد پيش بيني شده رکورد باشد(مانند تست سيستم در مواجه با بيش از چند ميليون رکورد، بازخورد سيستم در هنگام قطع و اختلال در خطوط انتقال داده يا آماده نبودن تجهيزات جانبي.

در پايان تست،هر آزمونگر ضمن توليد مستند گزارش انجام آزمون (Test Result) نتيجه آزمون رادر يک TR درج مي نمايد. از ديگر نکات مهم تست نحوه برخورد سيستم ها در هنکام بروز خطا و کنترل محيط اجرايي در جهت حفظ کارايي کل سيستم مي باشد.

**2.9. نحوه ي تست نرم افزار**

بر اساس صحبت ها و هماهنگي هاي انجام شده اعضاي گروه به اين نتيجه رسيدند که از بين

روشهاي موجود روش تست واحد (unit) را انتخاب بنمايند.

در نتيجه اجزاي نرم افزاري پروژه در عين حال برنامه نويسي تست خواهند شد.

**3. نحوه ي انجام کارهاي گروهي**

**3.1. تقسيم وظايف**

سايت در نظر گرفته شده براي کار گروهي سايت Trello.com است که تمامي اعضاي

گروه در آن عضو شده اند و تقسيم وظايف در آن انجام خواهد گرفت.

آدرس Board اختصاصي گروه در لينک زير است :

<https://trello.com/b/SdnfzsML/se2>

**3.2. مديريت کد نويسي**

براي مديريت کدهاي برنامه نويسي و کد نويسي گروهي از سايت Github.com استفاده

شده است.

که آدرس آن در زير ذکر شده است:

<https://github.com/KinGoverm/SE2>

**4. افراد و نقش ها**

در اين پروژه افرادي که نقش دارند شامل سه دسته هستند. مدير ، کارفرما و پيمانکار که هرکدام نقش ها و امکانات خاص خود را دارند.

کارفرما با شناسه و رمز خود وارد مي شود و سفارش و قيمت و مدت زمان کار مورد نظرش را در سيستم وارد مي کند.و در سيستم ثبت مي شود و به بقيه استفاده کنندگان توسط صفحه ي عمومي ابلاغ مي شود.

پيمانکار با شناسه و رمز خود وارد سيستم مي شود و از بين سفارش هاي ثبت شده کاري را براي خودش انتخاب مي کند با احتساب شرايط مالي و زماني و ... . بدين ترتيب طرحي را در سيستم بارگذاري مي کند.

کارفرما با بررسي طرح هاي اوليه ثبت شده آنهايي را که مي خواهد با مبلغ پيشنهادي انتخاب مي کند و در صورت بروز اختلافات در مورد هزينه ها و پرداخت پيش پرداخت و پرداخت کامل و مواردي اينچنين هريک از استفاده کنندگان با انتخاب گزينه ي شکايت مدير سايت را مطلع مي سازد و مدير با بررسي شکايت ها موضوع را بررسي خواهد کرد.در اين سيستم نحوه ي تعامل کارفرما و مشتري به منظور پرهيز از ايجاد اختلافات و اختلالات بدين گونه است که کارفرما پس از بررسي طرح هاي اوليه به آن طرح هايي که مورد پسندش واقع شده مبلغي پيشنهادي عنوان مي کند و اگر پيمانکار نيز قبول کرد کارفرما موظف به پرداخت درصدي از آن مبلغ به سايت مي سپارد و پيمانکار از جهت پشتوانه ي مالي نگراني نخواهد داشت و چون اطلاعات و تراکنش ها و قرارداد ها در نزد سايت محفوظ است امکان پيگيري هاي احيانا قانوني نيز فراهم است.

پس از آنکه پيمانکار طرح نهايي را پياده سازي نمود و خواستار بارگذاري آن در سايت شد کارفرما بايد همه ي مبلغ قرارداد را بپردازد و طرح يا همان سفارش کار را دريافت کند.در آخر در هنگام تصفيه ي حساب سهم پيمانکار و کارفرما و سايت به درستي و طبق قرارداد و قوانين تقسيم مي گردد.

در همه ي اين مراحل براي مدير سايت امکان بررسي ها و تنظيمات و حل و فصل کردن اشتباهات فراهم است.

در حين کار کردن با سيستم به طريقي که گفته شده امکان گفتگو نيز فراهم است و کارفرما و پيمانکار مي توانند با هم به توافق بيشتري برسند و تبادل اطلاعات به منظور پيشرفت کار کنند.

**5. پشتيباني**

پشتيباني نرم افزاري سيستم تحت وب ما شامل برطرف کردن اختلالات مالي در استفاده از سيستم و در صورت وقوع چنين رخدادي با بررسي آن مورد را برطرف مي سازيم.بررسي شکايات مطرح شده توسط استفاده کنندگان توسط مديريت سايت صورت مي گيرد.در صورت بروز اختلالات در سيستم اتصال به بانک اطلاعاتي و موارد اينچنين مديريت سايت موظف به پيگيري مورد مي شود و حق استفاده کننده را مي ستاند.

در حين استفاده از سيستم در صورت بروز اشکال در سيستم يا سوال در مورد قوانين و اطلاعات هر کدام از استفاده کنندگان اعم از کارفرما يا پيمانکار مي توانند با استفاده از سيستم گفتگوي آنلاين موجود در سايت با بخش پشتيباني در ارتباط باشند و پشتيبان با ارائه ي پاسخ هاي مفهوم و مربوط به اين وظيفه عمل مي نمايد.

در صورت بروز اشکال در سيستم سرور وب سايت بخش پشتيباني و مديريت سايت وظيفه ي برطرف کردن اختلال را دارد و در اسرع وقت اين کار را با به کارگيري بهترين راه حل هاي کاربردي انجام مي دهد.

بخش ديگر پشتيباني اين وب سايت ، درگاه پرداخت اينترنتي بانک معتبر جمهوري اسلامي ايران مي باشد که با ارائه ي خدمات خود تبادلات مالي را تسهيل مي کنند.

در صورت بروز اشکال در تراکنش بانکي بخش پشتيباني با بانک مربوط تماس گرفته و مسئله را پيگيري مي کند.

**پايان**