



نام پروژه:

طراحی وبسایت محاسبه مخارج دانشجویی

استاد راهنما:

جناب آقای مهندس سید مجتبی بنایی

مقدمه

امروزه با گسترش فراوان اینترنت داشتن یک وب سایت برای اهداف مختلف یک ضرورت و باید انکار نشدنی است. که در راستای این ضرورت ها و با توجه به مشکلاتی که دانشجویان برای حساب و کتاب خوابگاه های دانشجویی دارند ما تصمیم به تولید نرم افزاری برای رفع این مشکل گرفتیم. و امیدواریم که این نرم افزار جهت رفع این مشکل مفید واقع شود.

اهداف

- ارائه یک وبسایت برای آسانتر شدن محاسبه مخارج دانشجویی در خوابگاه ها و خانه های دانشجویی به صورت آنلاین
- جلوگیری از ،گم شدن اطلاعات مخارج
- رفع معضلات محاسبات مخارج دانشجویی
- انجام محاسبات به صورت دقیق

بخش های مختلف سایت

- دارای قسمتی جهت درج اطلاعاتی ها
- قسمتی جهت ساخت یوزر
- ثبت مخارج توسط اعضای هر گروه تعریف شده توسط مدیر آن گروه
- قسمتی برای ارتباط با مدیر سایت
- تعریف و دسته بندی انواع هزینه ها
- گزارش گیری های جامع از وضعیت دخل و خرج ماهیانه و مقایسه آن در ماههای مختلف
- ثبت ، نگهداری و ارائه گزارش از وضعیت حساب وکتاب
- دفترچه یادداشت و دفتر نگهداری و یادآوری قرار ملاقات ها و برنامه های مهم
- مطالب بسیار مفید جهت کسب ثروت و موفقیت در کسب وکار، و تجارت
- و امکانات دیگر ...

مشخصات فنی

پایه اصلی زبان برنامه نویسی نرم افزار فوق ، زبان قدرتمند python و فریم ورک tornado به همراه زبانهای html و javascript و css می باشد که از پایگاه داده ی مای اسکیوال استفاده میکند. برای طراحی این سایت از نرم افزار pycharm که نرم افزاری قدرتمند برای برنامه نویسی تحت وب می باشد استفاده شده است .

- همچنین نکات زیر در طراحی مورد توجه قرار خواهد گرفت :

- بارگزاری سریع
- سازگاری با مرورگرهای معروف و متفاوت

- ظاهر گرافیکی ساده و زیبا
- امنیت
- محیط کاربری ساده برای مدیریت سایت

مشخصات و ویژگی های زبان پایتون:

پایتون (به انگلیسی: Python) یک زبان برنامه نویسی همه منظوره، [۱] سطح بالا، [۱] شیء گرا و مفسر است که توسط خودو فان روسوم (به هلندی: Guido van Rossum) در سال ۱۹۹۱ در کشور هلند طراحی شد.

فلسفهٔ ایجاد آن تاکید بر دو هدف اصلی خوانایی بالای برنامه های نوشته شده [۲] و کوتاهی و بازدهی نسبی بالای آن است. [۳] کلمات کلیدی و اصلی این زبان به صورت حداقلی تهیه شده اند و در مقابل کتابخانه هایی که در اختیار کاربر است بسیار وسیع هستند.

بر خلاف برخی زبان های برنامه نویسی رایج دیگر که بلاک های کد در آکولاد تعریف می شوند (به ویژه زبان هایی که از گرامر زبان سی پیروی می کنند) در زبان پایتون از نویسه فاصله و جلو بردن متن برنامه برای مشخص کردن بلاک های کد استفاده می شود. به این معنی که تعدادی یکسان از نویسه فاصله در ابتدای سطرهای هر بلاک قرار می گیرند، و این تعداد در بلاک های کد درونی تر افزایش می یابد. بدین ترتیب بلاک های کد به صورت خودکار ظاهری مرتب دارند.

پایتون مدل های مختلف برنامه نویسی (از جمله شیء گرا و برنامه نویسی دستوری و تابع محور) را پشتیبانی می کند و برای مشخص کردن نوع متغیرها از یک سامانهٔ پویا استفاده می کند.

این زبان از زبان های برنامه نویسی مفسر بوده و به صورت کامل یک زبان شیء گرا است که در ویژگی ها با زبانهای تفسیری پرل، روبی، اسکیم، اسمال تاک و تی سی ال مشابهت دارد و از مدیریت خودکار حافظه استفاده می کند.

پایتون پروژه ای آزاد و متن باز توسعه یافته است و توسط بنیاد نرم افزار پایتون مدیریت می گردد.

مشخصات و ویژگی های فریم ورک ترنادو:

با کمک تورنادو شما می توانید یک وب سرور ایجاد کنید (Tornado Web Server) همانند آپاچی، که سپس بتوانید روی آن وب سایت نوشته شده خود با کد های پایتون که از کتابخانه قدرتمند تورنادو نیز قدرت می گیرند را اجرا کنید.

همچنین تورنادو این قابلیت و قدرت را دارد که همزمان از ده ها و هزاران کانکشن بطور طولانی مدت پشتیبانی کند پس با خیال اسوده می توانید Tornado Web Server خود را راه اندازی کنید.

تورنادو تقریباً به چهار قسمت اصلی می تواند تقسیم شود:

۱- یک فریم ورک وب (که شامل می شود RequestHandler برای ایجاد اپلیکیشن های وب و پشتیبانی کلاس های مختلف)

۲- می توان سمت سرور و کلاینت آنرا با HTTP پیاده سازی کرد، HTTPServer و AsyncHTTPClient

۳- کتابخانه شبکه ناهمگام (IOLoop و IOStream) که می تواند برای پیاده سازی دیگر پروتکل ها استفاده شود

۴- کتابخانه coroutin (tornado.gen)

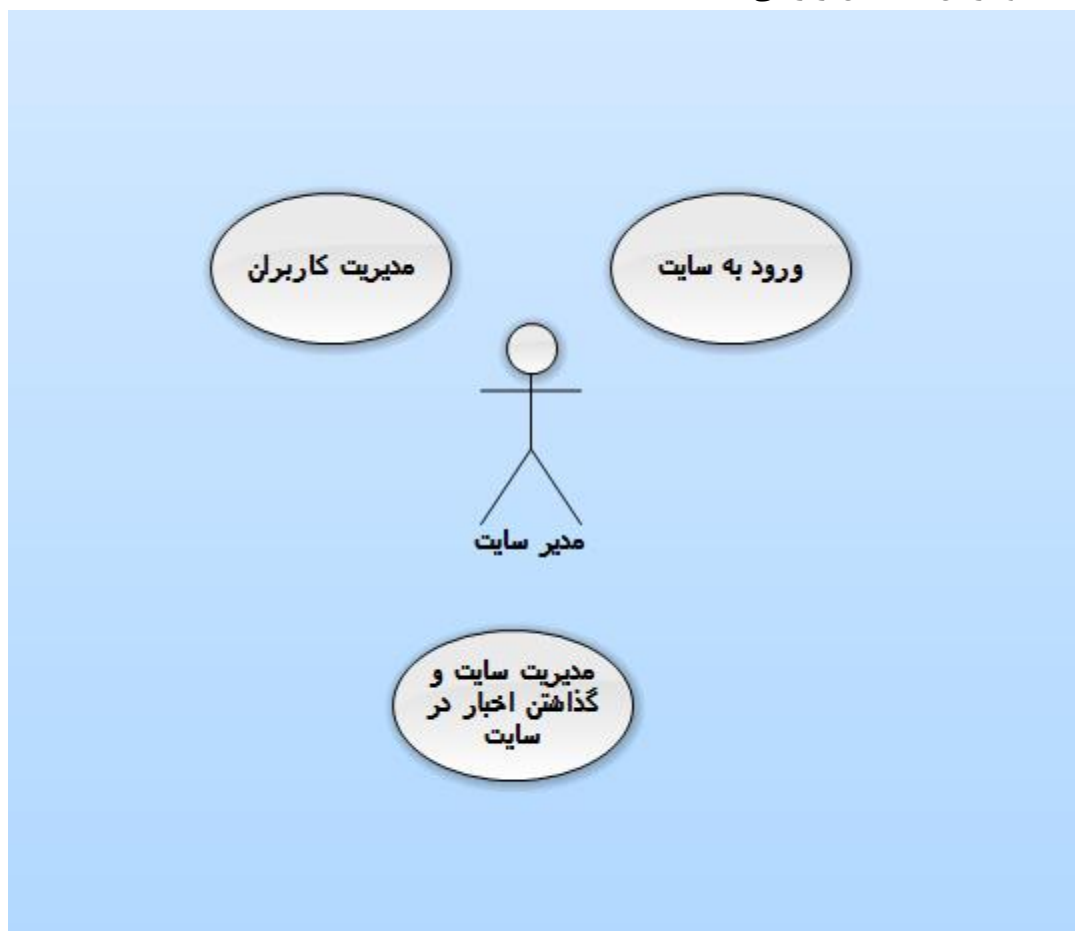
معرفی سایت:

نمودار یوزکیس ها و اکتور های سایت:

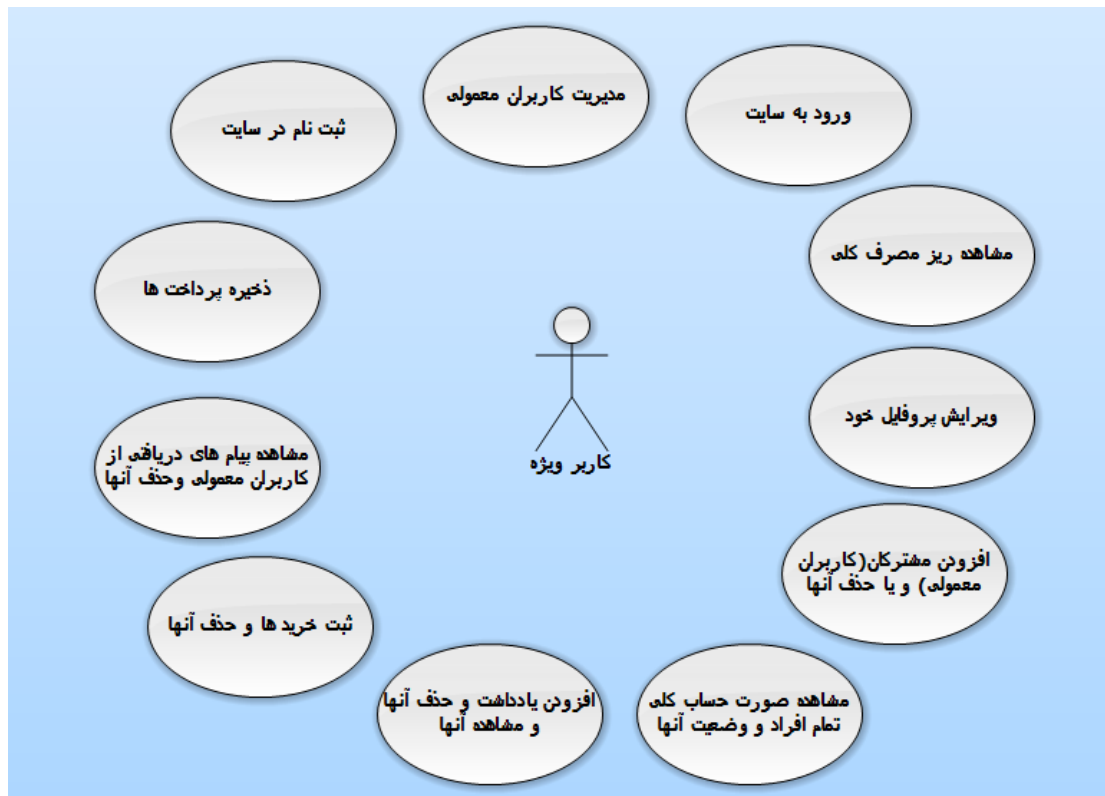
این سایت شامل سه دسته کاربر می باشد:

۱- مدیر سایت

که نمودار آن به شکل زیر می باشد:

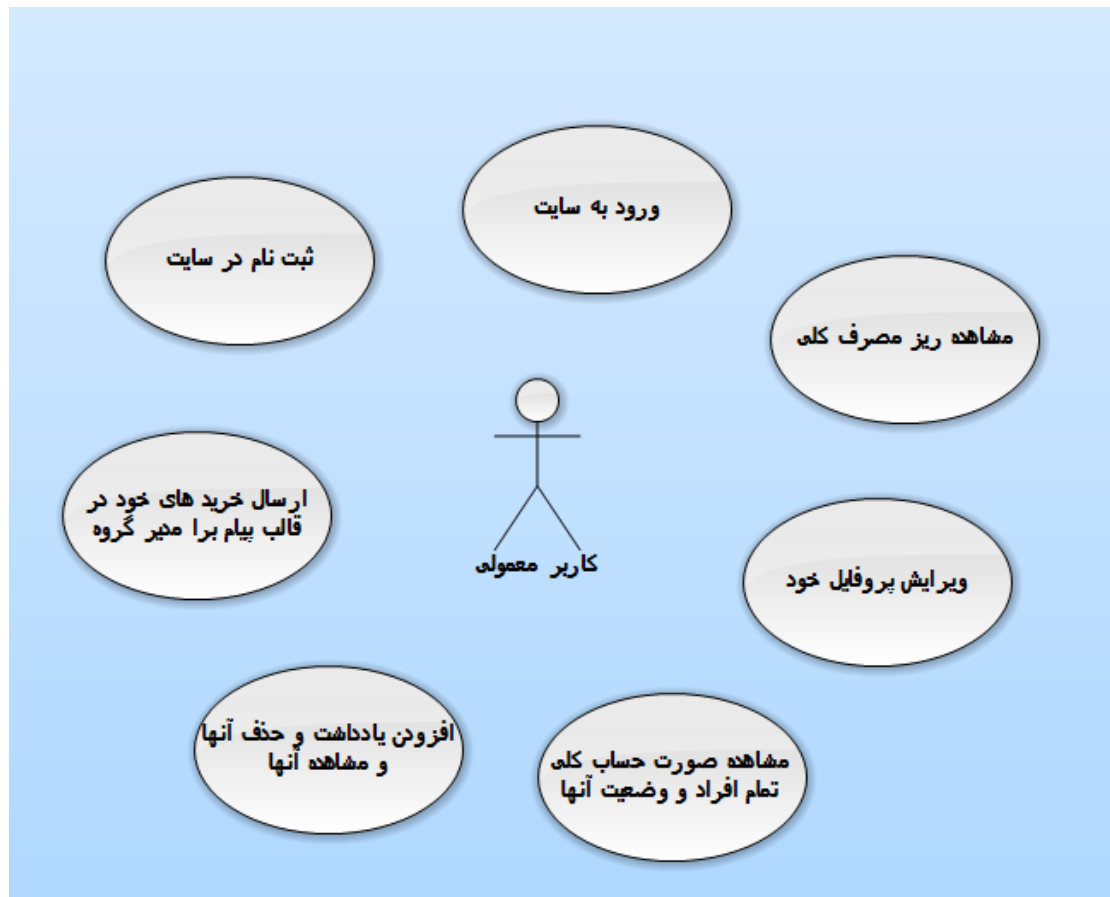


۲- کاربر ویژه که یک اکانت خریداری کرده و یک گروه می سازد شامل مشترکان (کاربران معمولی) که نمودار آن به شکل زیر می باشد:



۳-کاربر معمولی

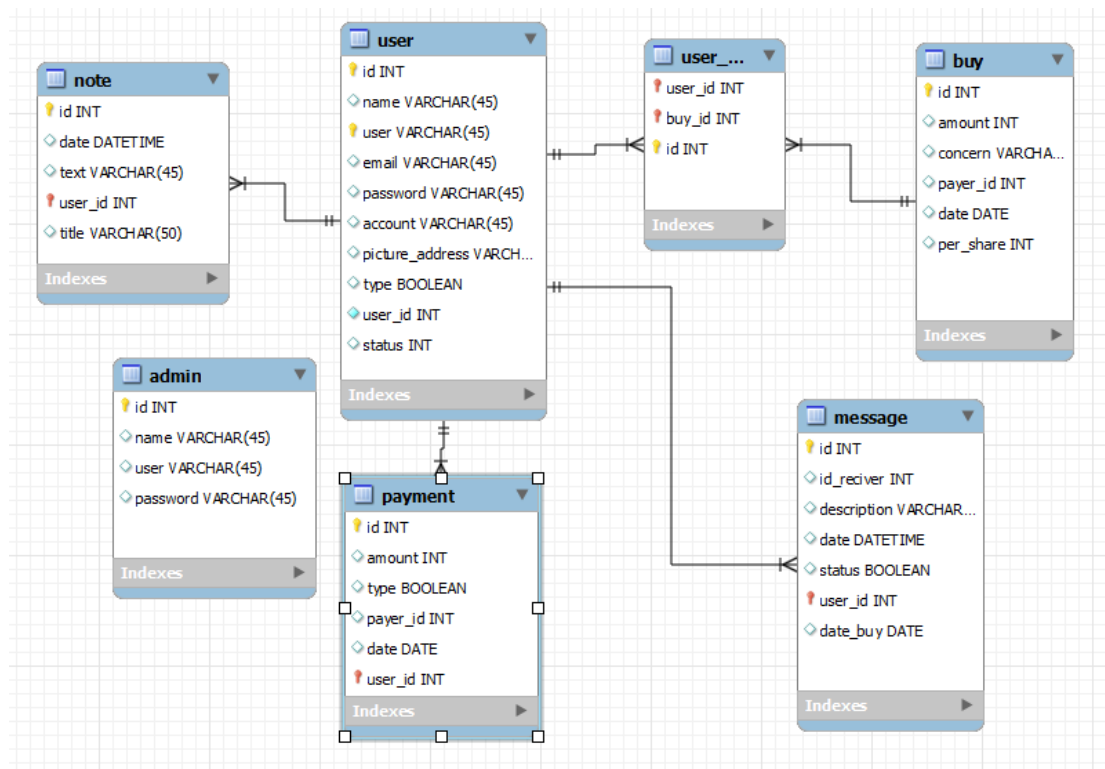
که نمودار آن به شکل زیر می باشد:



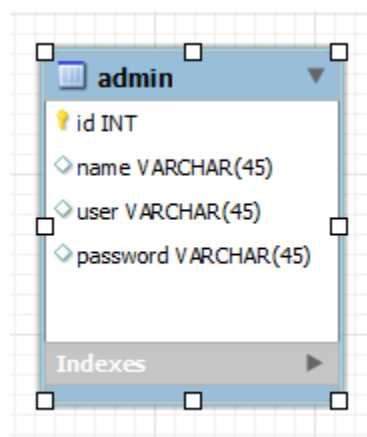
دیتابیس سایت

در این پروژه از دیتابیس مای اسکیوال استفاده شده است که شامل ۷ جدول میباشد و جداول آن به صورت زیر می باشد.

شمای کلی دیتابیس



۱- جدول ادمین که شامل چهار فیلد می باشد. که فیلد id آن primary key می باشد.
 در فیلد name آن که از نوع کاراکتر (varchar) نام ادمین سایت یا مدیر سایت ذخیره میشود. و در
 فیلد user که از نوع کاراکتر (varchar) می باشد نام کاربری مدیر ذخیره میشود و در فیلد
 password که آن هم نیز از نوع کاراکتر میباشد یعنی پسورد هم میتواند عدد باشد و هم میتواند
 کاراکتر باشد، پسورد مدیر را ذخیره می کنیم.



۲- جدول یوزر که مخصوص کاربران سایت می باشد و شامل ۱۰ فیلد می باشد. که فیلد user , id
 آن primary key می باشند.
 در فیلد name این جدول اسامی کاربران ویژه (مدیران گروه ها) و اسامی کاربران معمولی ذخیره
 میشود که از نوع varchar میباشد.

در فیلد user، نام کاربری کاربران ذخیره میشود و این نیز از نوع varchar می باشد.

در فیلد email، آدرس ایمیل کاربران را ذخیره میکنیم که این فیلد نیز از نوع varchar میباشد.

در فیلد password، پسورد کاربران را ذخیره میکنیم که در کدنویسی به گونه ای در نظر گرفته شده است که حتما باید ۶ کاراکتر باشد این فیلد نیز از نوع varchar میباشد.

در فیلد account مقادیری که کاربر در خریدها مشارکت داشته است ذخیره میشود و به ازای هر خرید جدیدی که افزوده میشود این مقدار آپدیت میشود. که از نوع Int میباشد.

در فیلد picture_address، آدرس عکس کاربر ذخیره میشود که از نوع varchar میباشد.

در فیلد type نوع کاربر مشخص میشود که آیا کاربر معمولی است یا کاربر ویژه (مدیر یک گروه) که این فیلد از نوع Boolean میباشد که برای مدیران گروه true در نظر گرفته میشود و برای بقیه ی کاربران مقدار false در نظر گرفته میشود.

فیلد status در اصل وضعیت کاربر را مشخص میکند که آیا این کاربر فعال میباشد و یا اینکه توسط مدیر گروه غیر فعال گشته است. این فیلد از نوع int در نظر گرفته شده است که برای مدیران گروه همیشه عدد ۳ در نظر گرفته میشود زیرا مدیر گروه خود را نمیتواند غیر فعال کند و برای دیگر کاربران در صورت فعال بودن عدد ۱ و در صورت غیر فعال بودن عدد صفر در نظر گرفته میشود.

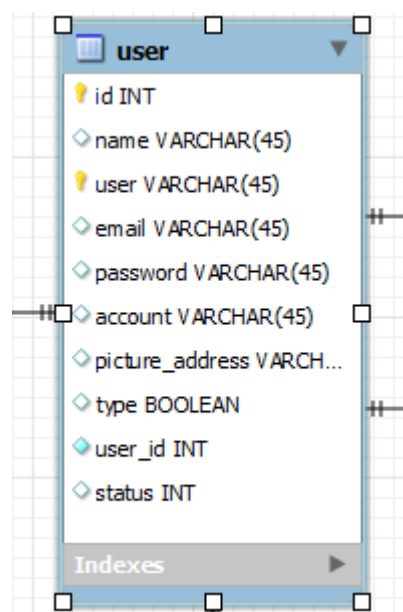
رابطه های این جدول به صورت زیر می باشد:

۱- رابطه ی چند به چند با جدول buy

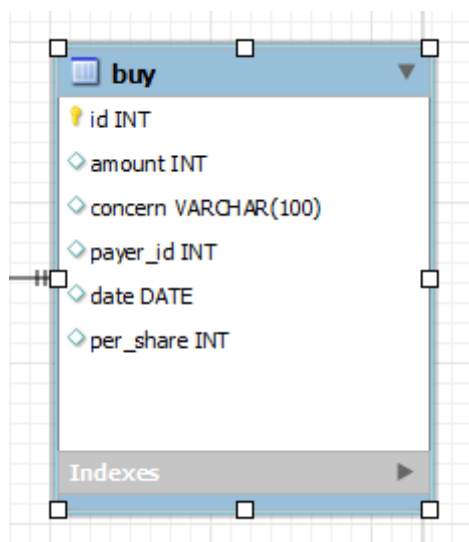
۲- رابطه ی یک به چند با جدول note

۳- رابطه ی یک به چند با جدول message

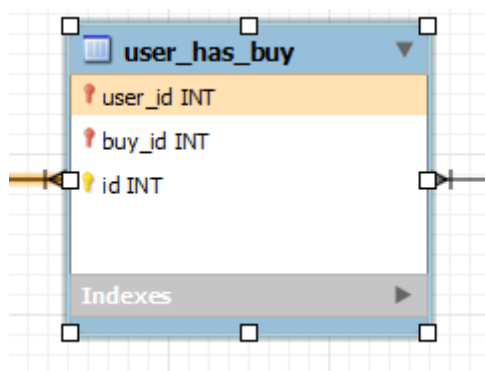
۴- رابطه ی یک به چند با جدول payment



۳-جدول buy (خرید) که مخصوص ثبت خرید ها می باشد و شامل ۶ فیلد می باشد. و فیلد id در آن primary key می باشد. و یک رابطه ی چند به چند با جدول user دارد. فیلد amount در این جدول برای مقدار خرید در نظر گرفته شده است که از نوع int میباشد. فیلد concern در این جدول برای کالایی که خرید شده است در نظر گرفته شده است که از نوع varchar می باشد. فیلد payer_id برای خریدار این خرید در نظر گرفته شده است و در اصل id کسی است که خرید را انجام داده است. که از نوع int میباشد. فیلد date که تاریخ انجام خرید در آن ذخیره میشود که از نوع date میباشد. فیلد per_share در واقع یک فیلد کمکی است چون در قسمت حذف خرید برای کم کردن مقداری که از آن خرید پای افرادی که در آن خرید بوده اند حساب شده است استفاده میشود و در اصل مقدار پولی است که از هر خرید سهم افرادی است که در آن خرید بوده اند. که از نوع int میباشد.



۴-جدول user_has_buy که رابطه ی چند به چند بین جدول های user , buy را مشخص میکند:

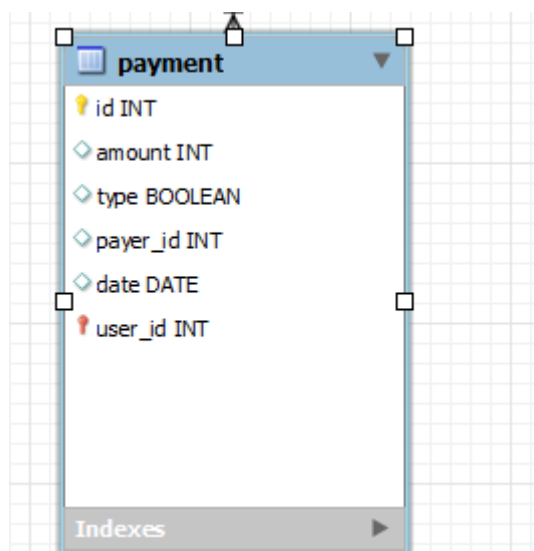


۵-جدول payment (پرداخت) که برای ثبت پرداخت ها و دریافت ها در نظر گرفته شده است. و شامل ۶ فیلد می باشد و فیلد id آن primary key می باشد و فیلد user_id آن foreign key می باشد از جدول user بدلیل رابطه ی یک به چند جدول user با آن. فیلد amount در این جدول برای ذخیره مقدار پول پرداخت یا دریافت است که از مشترک گرفته میشود و از نوع int میباشد.

فیلد type در این جدول مشخص کننده نوع عملیات که ایا دریافت بوده است یا پرداخت را مشخص میکند. که از نوع Boolean میباشد.

فیلد payer_id، مشخص کننده فردی است که عملیات پرداخت یا دریافت پول را انجام داده است. که در اصل id فردی است که عملیات را انجام داده است. که از نوع int میباشد.

فیلد date هم که مشخص کننده تاریخ انجام پرداخت یا دریافت میباشد.



۶-جدول message (پیام) که شامل ۷ فیلد می باشد. و فیلد id آن primary key می باشد و فیلد user_id آن foreign key می باشد از جدول user

فیلد `id_reciver` برای ذخیره `id` فردی است که قرار است پیام را دریافت کند (که در اصل مدیر همان گروهی است که کاربر عضو آن است). که از نوع `int` می باشد.

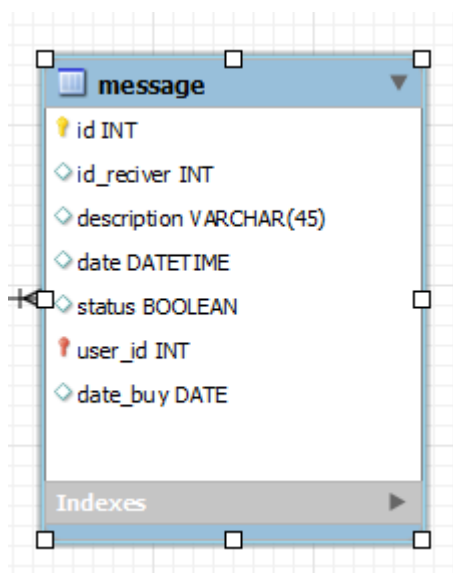
فیلد `description` برای ذخیره متن همان پیام میباشد که حداکثر ۲۵۵ کاراکتر را میتواند ذخیره کند و از نوع `varchar` میباشد.

فیلد `date` هم که مربوط به تاریخ ثبت آن پیام میباشد و از نوع `date` میباشد.

فیلد `status` که در جدول مشخص کننده وضعیت یک پیام میباشد که اگر `true` باشد به معنای ثبت نشدن آن پیام میباشد و اگر `false` باشد به معنای آن است که این پیام ثبت شده است. که این فیلد از نوع `Boolean` میباشد.

فیلد `user_id` که این فیلد مشخص کننده فرستادننده یک پیام میباشد و `id` کسی است که پیام را میفرستد و از نوع `int` میباشد.

فیلد `date_buy` که در آن تاریخ خرید که انجام شده است را ثبت میکنیم و این فیلد از نوع `date` میباشد.



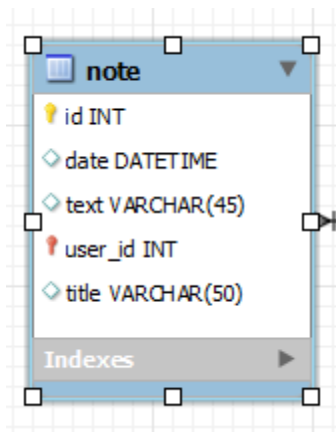
۷-جدول `not` (یادداشت) که برای ثبت یادداشت های سایت می باشد و شامل ۵ فیلد می باشد. که فیلد `id` آن `primary key` می باشد و فیلد `user_in` آن `foreign key` میباشد از جدول `user`

فیلد `date` برای ذخیره زمان یادآوری یادداشت به کاربر در نظر گرفته شده است.

فیلد `text` برای ذخیره متن یادداشت در نظر گرفته شده است که از نوع `varchar` میباشد.

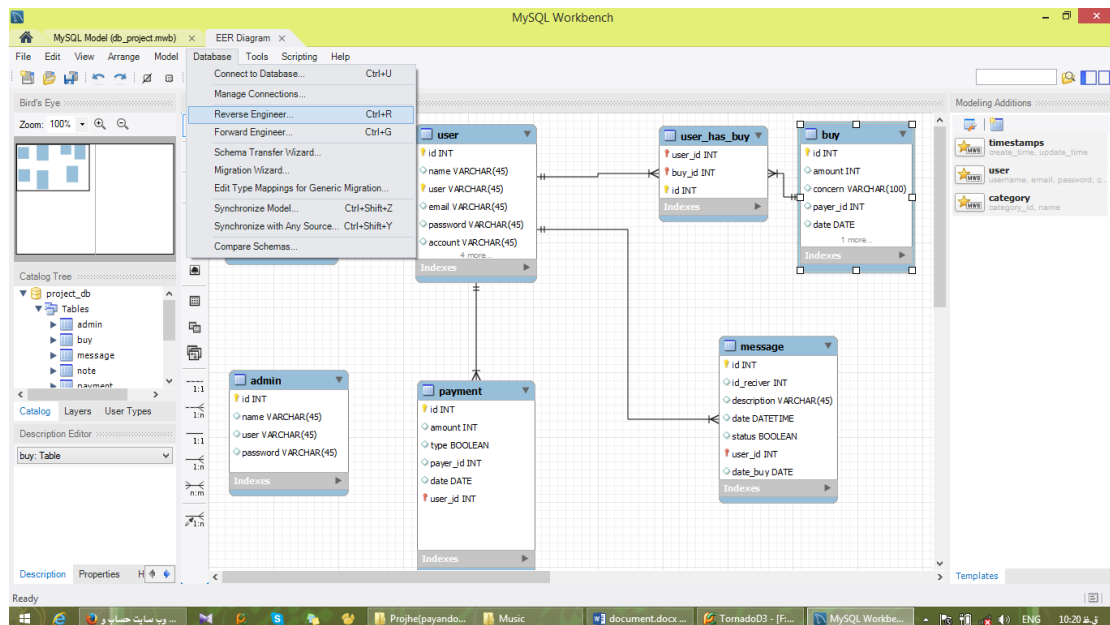
فیلد `user_id` که مشخص کننده کاربری است که یادداشت مربوط به آن میباشد.

فیلد `title` که مشخص کننده موضوع یک یادداشت میباشد و از نوع `varchar` میباشد.

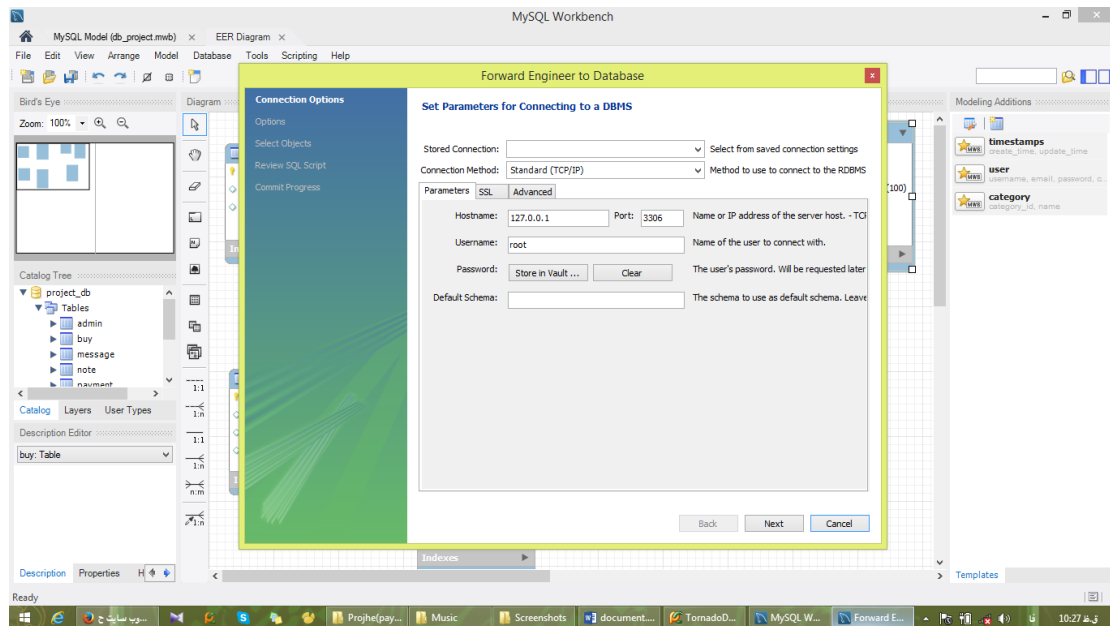


پس از طراحی کامل دیتابیس باید آن را به سرور محلی متصل کرد. نحوه ی انجام این کار را مرحله به مرحله با تصویر در زیر نشان میدهم:

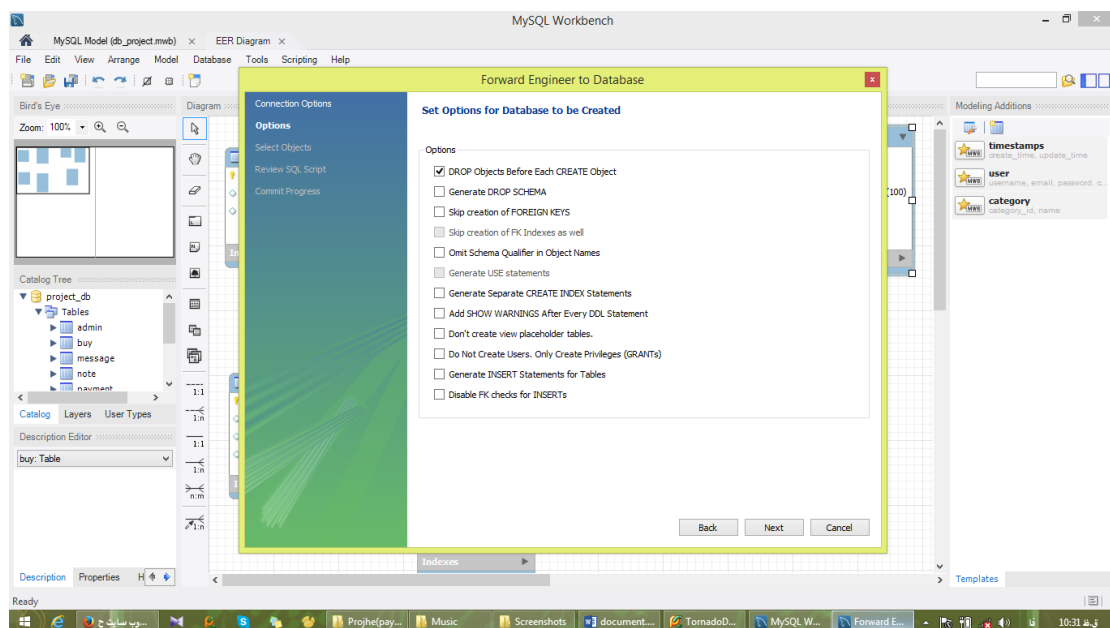
ابتدا از تب database در منوی بالا گزینه forward Engineer را انتخاب میکنیم:



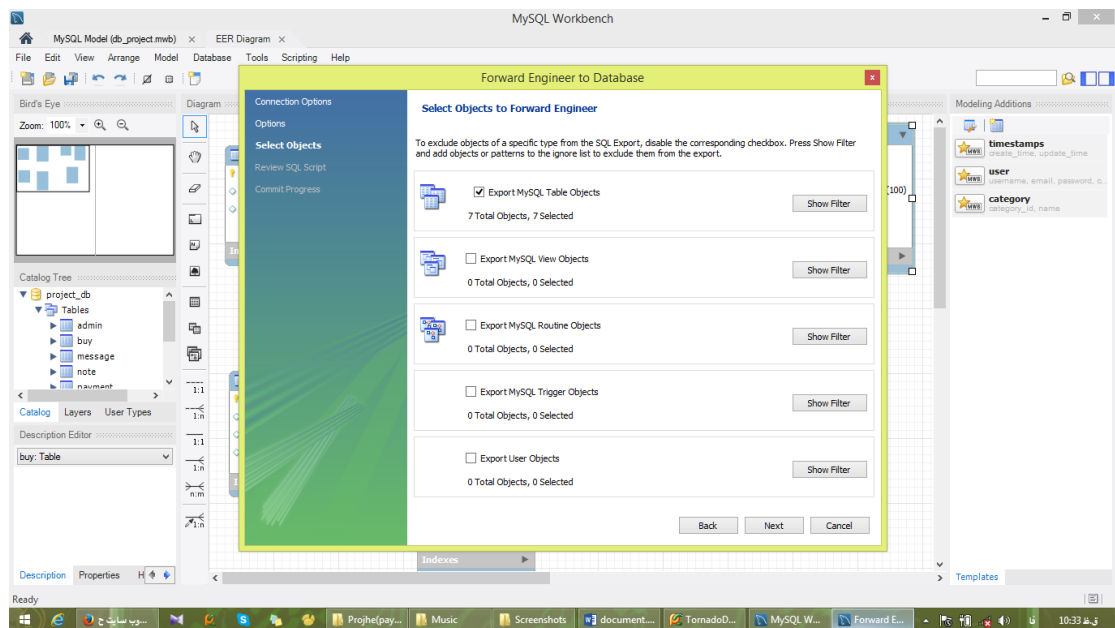
سپس صفحه ای باز میشود که مربوط به تنظیمات آن میباشد که در آنجا باید Hotname را 127.0.0.1 وارد کنیم و port را ۳۳۰۶ بزنیم و در قسمت Username کلمه ی root را وارد کنیم که البته این مقادیر و پارامترها به صورت پیش فرض قرار دارند فقط ما باید چک کنیم که حتما این مقادیر باشند:



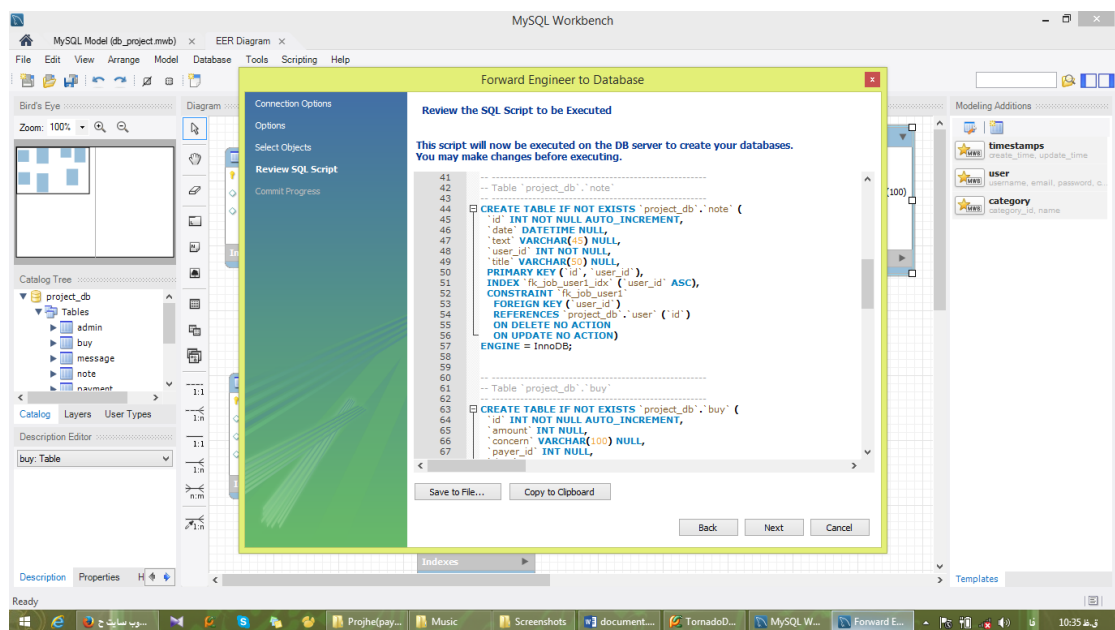
سپس گزینه ی Next را میزنیم که صفحه ای به شکل زیر ظاهر میشود که اگر دیتابیس قبلا با این نام در سرور محلی قرار داشته باشد گزینه ی drop object before each create object را میزنیم تا دیتابیس قبلی را پاک کرده و دیتابیس جدید را به جای آن قرار دهد.



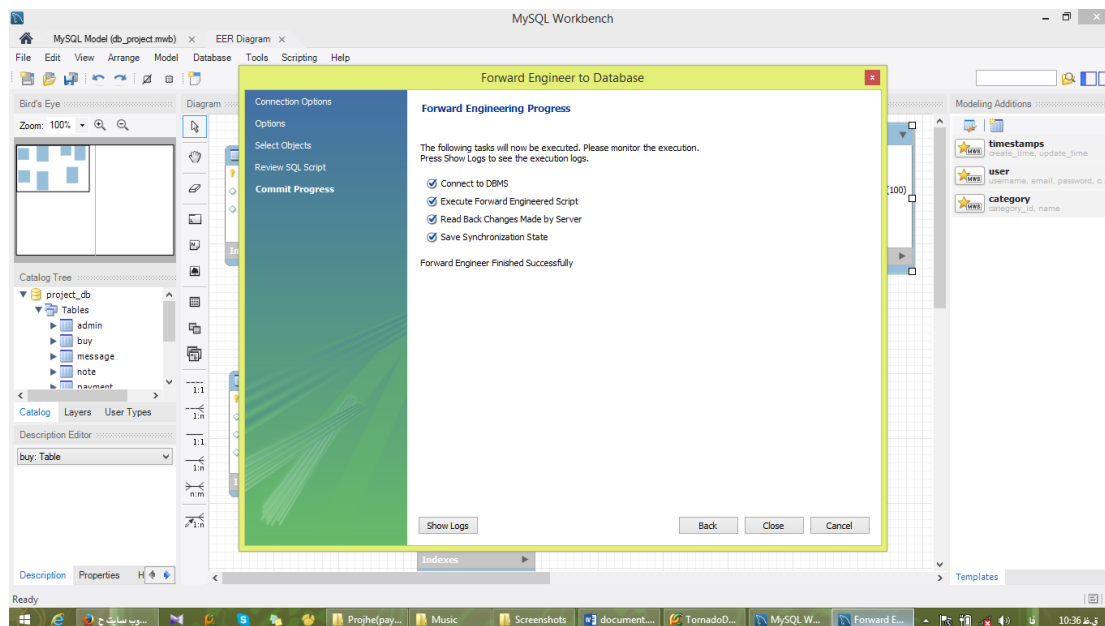
سپس صفحه ای مانند شکل زیر ظاهر میشود که باید چک کنیم که گزینه ی اول تیک خورده باشد:



سپس صفحه ی زیر به کاربر نمایش داده میشود که میخواهد جداول را در سرور محلی بسازد:



سپس صفحه ی زیر به کاربر نمایش داده میشود که دیگر کار تمام است و باید گزینه ی close را بزنیم:



نحوه ی تعریف کردن دیتابیس در ترنادو:

در ترنادو دیتابیس را در فایل `models.py` تعریف میکنیم. یک نمونه جدول را در زیر به عنوان نمونه در زیر تعریف میکنیم: (که این نوع تعریف کردن کلاس ها و جداول مخصوص کتابخانه `peewee` میباشد)

در مثال زیر `Payment` نام جدول مورد نظر میباشد.

هر کدام از فیلد ها هم به طور جداگانه داخل آن تعریف میشوند به صورت زیر و نوع آنها هم مشخص میشود.

اینجا باید `primarykey` ها و `foreignkey` ها همان طور که در مثال زیر آمده است تعریف شوند.

مثلا در مثال زیر فیلد `User` یک کلید خارجی از جدول `User` میباشد و فیلد `id` جدول `User` هم میباشد.

```
class Payment (MySQLModel):
    id = peewee.PrimaryKeyField()
    amount = peewee.IntegerField()
    type = peewee.BooleanField()
    payer_id = peewee.IntegerField()
    date = peewee.DateField()
    User = peewee.ForeignKeyField(rel_model=User,
    to_field=User.id)
```


کتابخانه peewee :

Peewee یکی از کتابخانه های کار با دیتابیس در پایتون میباشد که برای استفاده از آن ابتدا باید آنرا در قسمت setting پای چارم نصب کرده و در هر قسمت که خواستیم از آن استفاده کنیم ابتدا آنرا import میکنیم سپس از آن استفاده میکنیم. با دستور زیر peewee را import میکنیم:

```
from peewee import fn
```

برای وارد کردن داده در کتابخانه peewee باید مانند زیر داده را وارد کنیم:

```
User.create(
    id=id_user,
    name=name,
    user=user,
    email=email,
    password=password,
    picture_address=address_image,
    account=0,
    type=type_user,
    User=id_admin,
    status=status
)
```

User نام جدولی است که قرار است در آن جدول داده وارد کنیم و یکی یکی فیلدهای را که در آن جدول قرار دارند مقداردهی میکنیم.
برای حذف کردن داده از جداول دیتابیس در کتابخانه peewee مانند زیر عمل میکنیم:

```
User().select().where(User.id == user_id).get().delete_instance()
```

که در این مثال User نام جدولی است که میخواهیم از آن داده حذف کنیم و برای گذاشتن شرط از where استفاده میکنیم و داخل پرانتز آن هر تعداد شرطی که داشته باشیم مینویسیم.
برای update کردن یک جدول در peewee مانند زیر عمل میکنیم:

```
User.update(status=status).where(User.id == user_id).execute()
```

در مثال بالا user نام جدولی است که قرار است در آن جدول عمل آپدیت را انجام دهیم.
فیلدی که قرار است آپدیت شود در پرانتز () update نوشته میشود و همان مقداری که قرار است به جای مقدار قبلی قرار بگیرد نیز جلوی علامت مساوی نوشته میشود.

اگر بخواهیم شرطی هم اعمال کنیم از `where` استفاده میکنیم و داخل پرانتز هر تعداد شرطی که داشته باشیم مینویسیم و در پایان `.execute()` را مینویسیم برای اینکه عملیات آپدیت انجام شود.

توابع تجمعی در peewee :

یکی از توابع تجمعی که در این پروژه استفاده شده است تابع `sum` میباشد که یک مثال از آن در زیر برای شما آورده ایم:

```
Payment= Payment.select(
    fn.sum(Payment.amount)).where(
    Payment.payer_id == user_id,
    Payment.type == True)
```

در این مثال ما میخواستیم جمع پرداخت های یک مشترک را از دیتابیس بخوانیم و در قسمت صورت حساب به کاربران نمایش دهیم.
برای این کار از جدول `payment`، جمع مقادیر پرداخت ها را با دستور `بالا بدست می آوریم` و هر شرطی هم که داشته باشیم اعمال میکنیم.

Join در peewee :

در این پروژه در قسمت ریز مصرف برای نمایش تمامی خرید ها و افرادی که در آن خریده ها بوده اند و دیگر مشخصات خرید از `join` استفاده کردیم:

```
find = Buy.select(
    Buy.id.alias('buy_id'),
    Buy.amount,
    Buy.concern,
    Buy.date,
    User.name,
    User.id
).join(User_has_buy).switch(
    User_has_buy).join(User).where(
    User.User == self.user['id_admin'])
```

در مثال بالا ما سه جدول `buy` و `user_has_buy` و `user` را با هم `join` کردیم که از جدول `buy`، مقدار و تاریخ و اجناس خرید شده و `id` را با نام `buy_id` میخواستیم داشته باشیم و از جدول `user`، نام و `id` را میخواستیم داشته باشیم.
جدول `buy` با جدول `User_has_buy` روی فیلد مشترک `Buy`، `join` شده اند.
جدول `user` با جدول `User_has_buy` روی فیلد مشترک `User`، `join` شده اند.

توضیح مختصری در مورد نحوه ی کار کردن ترنادو

به طور کلی ترنادو یک ساختار سه لایه ای دارد شامل:

۱-قسمت **Handler** که در این قسمت توابع ترنادو نوشته میشود.

۲-قسمت **Template** که در این کل صفحات **html** سایت تعریف میشوند.

۳-قسمت **static** که مربوط به فایل های **CSS** و **JS** و تصاویر مربوط به سایت قرار دارند.

سه فایل مهم در ترنادو:

فایل های **models** و **url** و **start** که فایل **models** جداول دیتابیس را در آن تعریف میکنیم و در فایل **url** یو آر ال مربوط به هر تابع را تعریف میکنیم و فایل **start** مربوط به تنظیمات ترنادو میباشد و تابع **main** در آن تعریف میشوند.

نحوه ی بالا آوردن یک صفحه در ترنادو و کارهای لازم:

برای این منظور ابتدا باید یک فایل **ch.py** در قسمت **templates** تعریف کنیم. سپس یک

url در فایل **url** به صورت زیر تعریف میکنیم:

```
(r'/admin/example', Handler.exampleHandler, "admin_example ")
```

سپس یک تابع در قسمت **handlers** برای این یو آر ال تعریف میکنیم به صورت زیر:

```
class example(TornadoRequestBase):
```

```
    def get(self, *args, **kwargs):
```

```
        self.render('user/add_buy.html')
```

که در این تابع صفحه ای که باید نمایش داده شود به تابع داده میشود.

این یک مثال ساده برای نمایش یک صفحه در ترنادو بود.

نحوه ی ارسال یک متغیر به یک صفحه و استفاده از آن در صفحه:

برای ارسال متغیر آنرا موقعی که صفحه را رندر میکنیم به آن صفحه میفرستیم مثلا در مثال زیر

متغیری برای یک صفحه ارسال میکنیم:

```
def get(self, *args, **kwargs):
```

```
    self.render('user/note.html', note=note)
```

حال در صفحه ی مورد نظر از این متغیر استفاده میکنیم.

مثال بالا همان صفحه ی یادداشت ها میباشد که کاربر وقتی به آن صفحه میرود باید لیست تمام

یادداشت ها را ببیند.

برای این منظور ابتدا ما باید یک سلکت در دیتابیس بنویسیم و کل یادداشت های آن کاربر را بخوانیم

و سپس آن را در یک متغیر به صفحه ی مورد نظر ارسال کنیم:

```
note = Note().select().where(
```

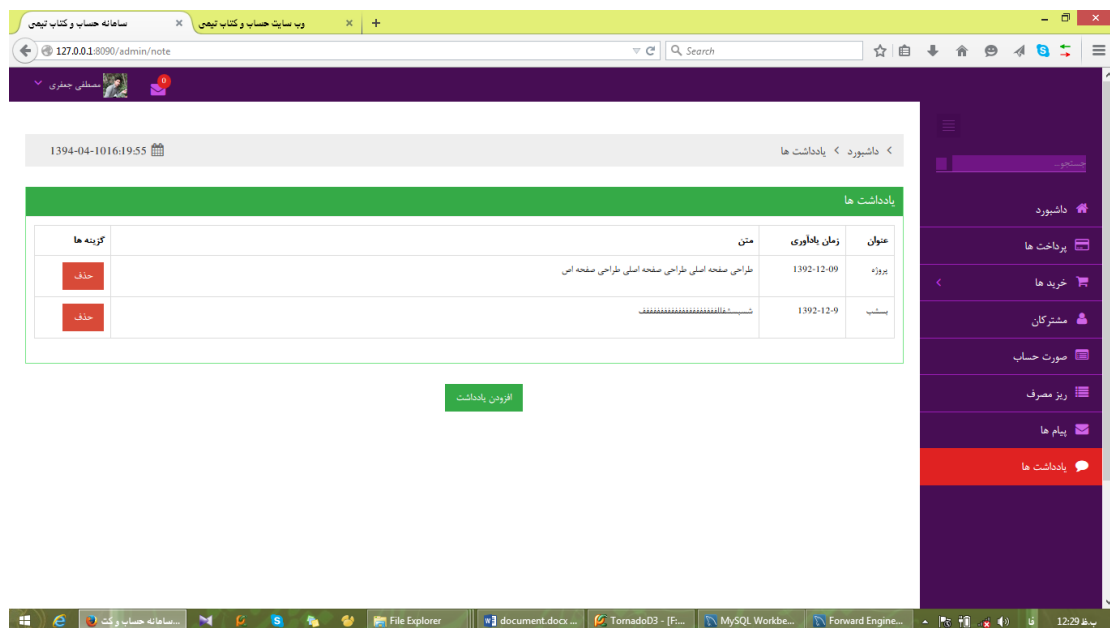
```
Note.User == self.session.get('id'))
```

حال از این متغیر در صفحه ی یادداشت ها استفاده میکنیم مانند زیر:

```
<tbody>
{% for item in note %}
<tr id="note_{{ item.id }}">
    <td>{{ item.title }}</td>
    <td>{{ item.date }}</td>
    <td>{{ item.text }}</td>
    <td><button onclick="delete_note('{{ item.id }}')"
class="btn red"> حذف </button></td>
</tr>
{% end for %}
```

که با این قطعه کد ما میتوانیم یکی یکی یادداشت ها را از دیتابیس بخوانیم و به کاربر نمایش دهیم.

که مانند شکل زیر یادداشت ها به کاربر نمایش داده میشود:



برخی از کتابخانه هایی که در پروژه استفاده شده است:

۱- Os.path:

این کتابخانه برای آپلود تصویر استفاده شده است. یک مثال از آن برای شما در زیر می آوریم که در قسمت ثبت نام از آن استفاده شده است:

```
file = self.request.files['image'][0]
original_fname = file['filename']
address_image = str(uuid.uuid4()) + ".jpg"
file2 = "static/upload/user_images/"
```

```
if not os.path.exists(file2): os.makedirs(file2)
output_file = open("static/upload/user_images/" +
address_image, 'wb')
output_file.write(file['body'])
```

در مثال بالا ابتدا فایل تصویر را میخوانیم و سپس برای این که تصویر با نام تکراری ذخیره نکنیم یک نام رندوم برای آن انتخاب میکنیم سپس در همان پوشه ای که میخواهیم آنرا ذخیره میکنیم.

:uuid-۲

از این کتابخانه برای تولید یک رشته ی رندوم با طول زیاد استفاده میکنیم.

:Jdatetime-۳

از این کتابخانه برای بدست آوردن تاریخ و ساعت شمسی استفاده میکنیم.

:Pycket.session-۴

از این کتابخانه برای استفاده از حافظه موقت استفاده میگردیم.

استفاده از زبان فارسی در توابع و فایل های هندلر:

اگر از زبان فارسی در فایل های هندلر استفاده کنیم خطا میدهد برای برطرف کردن این خطا باید دو خط کد در بالا آن فایل بنویسیم که آن دو خط کد در زیر نوشته شده اند:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

معرفی قسمت های مختلف سایت

صفحه اصلی:

صفحه اصلی سایت در اصل یک صفحه میباشد که این صفحه به بخش های مختلفی از جمله:

۱-خانه / ۲-درباره ما / ۳-ارتباط با ما / ۴-راهنمای سایت / ۵-ثبت نام / میباشد.

۱- خانه:

قسمت خانه فقط شامل فرم لاگین میباشد شامل فرم لاگین میباشد.

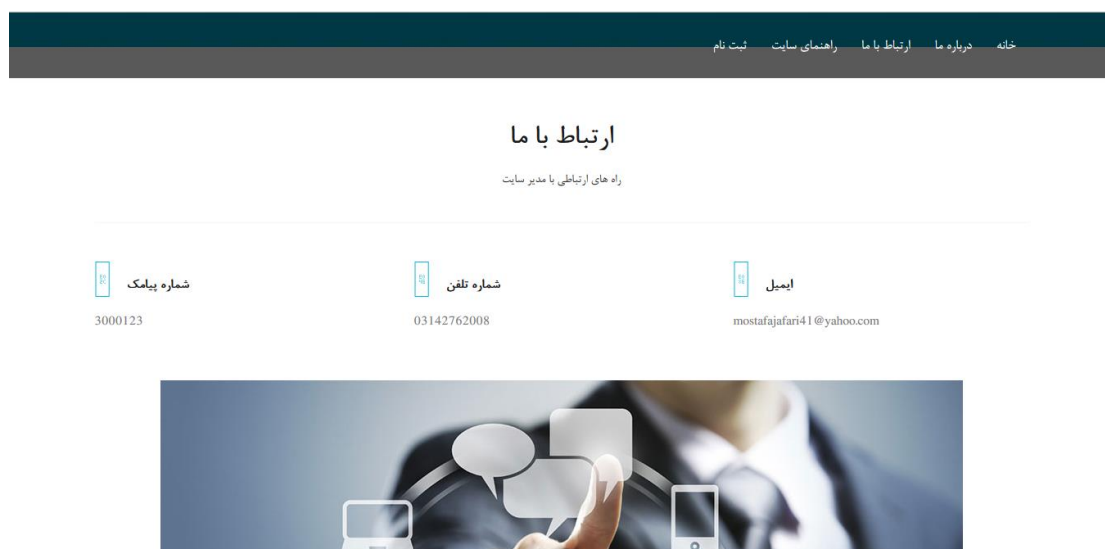
۲- درباره ما:

این قسمت اطلاعاتی درباره خود سایت و امکانات و ابزارهایی که در اختیار کاربران قرار میدهد به کاربران و بازدید کنندگان میدهد:



۳- ارتباط با ما:

این قسمت از سایت اطلاعاتی برای برقراری ارتباط کاربران با مدیر سایت میدهد شامل : تلفن و ایمیل و موبایل



۴- راهنمای سایت:

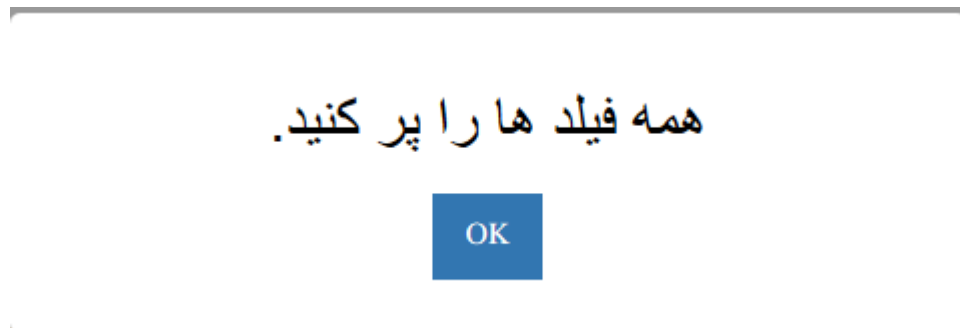
این قسمت از سایت راهنمایی جهت نحوه ی کارکردن با سایت در اختیار کاربران قرار میدهد به منظور سهولت استفاده از سایت و کاهش سردرگمی کاربران و همچنین جنبه ی تبلیغاتی هم میتواند برای سایت داشته باشد

۵- قسمت ثبت نام:

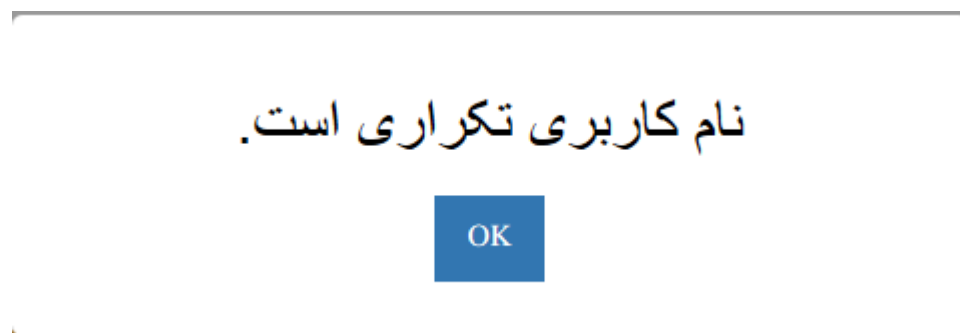
The image is a composite of two parts. The background is a blurred photograph of colorful wooden beads (red, blue, white) on strings. On the left, there is a white rectangular overlay containing Persian text and two buttons. The text reads 'وب سایت حساب و کتاب تیمی' (Team Accounting Website) and 'با نرم افزار حساب و کتاب تیمی دیگر نگران حساب و کتاب های گروهی خود نباش' (Don't worry about your group's accounting with our team accounting software). Below the text are two buttons: 'صفحه اصلی' (Home) in blue and 'ورود' (Login) in grey. On the right, there is another white rectangular overlay containing a registration form. The form has a title 'ثبت نام کنید' (Register Now) and five input fields with Persian labels: 'نام و نام خانوادگی...' (Full Name...), 'ایمیل...' (Email...), 'نام کاربری...' (Username...), 'پسورد...' (Password...), and 'تکرار پسورد...' (Repeat Password...). Below the form is a text label 'No file selected.' and a button labeled '...Browse' with the Persian text 'انتخاب تصویر' (Select Image). At the bottom of the form area is a blue button labeled 'ادامه' (Continue).

در این قسمت کاربر پس از وارد کردن اطلاعات ثبت نام میکند:

در این قسمت اگر کاربر یکی از فیلدها را خالی بگذارد و یا پسورد طولش از حد مورد نظر کمتر باشد و یا عکس آپلود نکرده باشد به کاربر پیام میدهد و خطای مورد نظر را اعلام میکند به شکل زیر:



و یا اگر نام کاربری تکراری وارد کند باز هم پیامی به کاربر نمایش داده میشود که نام کاربری تکراری است به شکل زیر:



پس از اینکه کاربر با موفقیت ثبت نام کرد یک پیام به کاربر نمایش داده میشود که ثبت نام با موفقیت انجام شده است به صورت زیر:



تایید!

ثبت نام با موفقیت انجام شد.

OK

۶- قسمت لاگین:

در این قسمت کاربر پس از وارد کردن یوزر و پس وادر سامانه میشود:



شمای کلی داشبورد کاربر وقتی که لاگین می کند به شکل زیر است:



۷- قسمت پرداخت ها:

در این قسمت کاربر میتواند پول های دریافتی یا پرداختی توسط مشتریان را ثبت کند:

مستطقی جعفری

3

1394-04-01 19:26:14

داشبورد

پرداخت ها

پرداخت ها

مبلغ را به تومان وارد کنید.

مبلغ

تاریخ

مشترک

دریافت از مشترک

پرداخت به مشترک

ذخیره پرداخت

لغو

داشبورد

پرداخت ها

خرید ها

مشترکان

صورتحساب

ریز مصرف

پیام ها

یادداشت ها

فرم پرداخت ها شامل چهار فیلد میباشد که در فیلد اول باید مبلغ پرداختی را به تومان وارد کنیم و در قسمت تاریخ، تاریخ را وارد کرده و در قسمت مشترک که تمام مشترکان در یک گروه در آن میباشند مشتری که پرداخت را انجام داده است انتخاب میکنیم و در قسمت عملیات نوع عملیات که آیا پول از مشترک دریافت شده و یا پول پرداخت شده را مشخص میکنیم مثلاً اگر مشترک علی احمدی مبلغ ۱۰۰۰۰ تومان را پرداخت کرده است مبلغ را وارد کرده و نوع عملیات را پرداخت انتخاب میکنیم و سپس گزینه ی ذخیره پرداخت را میزنیم. اگر پرداخت با موفقیت انجام شود یک پیام به کاربر نمایش داده میشود که پرداخت با موفقیت با موفقیت انجام شد.

۸- قسمت افزودن خرید:

کاربر در این قسمت خرید های خانه را وارد میکند. این قسمت به گونه ای طراحی شده است که جزئیات خرید مانند: خریدار، مقدار پول خرید، افرادی که در آن شریک بوده اند ثبت میشود.

ثبت خرید جدید

خریدار

مبلغ را به تومان وارد کنید.

تاریخ

بابت

سعيد شاد

روح اله ملكي

مستطقی جعفری

كمال شوربانی

مستطقی زارعی

ذخیره خرید

لغو

داشبورد

پرداخت ها

خرید ها

خرید جدید

حذف خرید

مشترکان

صورتحساب

ریز مصرف

پیام ها

یادداشت ها

فرم ثبت خرید جدید شامل پنج فیلد میباشد که فیلد اول یک منوی آپشنال است که نام همه ی مشترکین در آن میباشد که هرکس خرید را انجام داده باشد نام همان فرد را انتخاب میکنیم ، فیلد بعدی مبلغ خرید میباشد که باید به عدد و تومان وارد شود و فیلد بعدی تاریخ خرید را وارد میکنیم و فیلد بعدی جنسی است که خریده شده است را وارد میکنیم و در فیلد بعدی افرادی که در آن خرید سهمیه بوده اند را تیک میزنیم و انتخاب میکنیم سپس دکمه ذخیره خرید را میزنیم و اگر خطایی مانند خالی بودن فیلد ها و یا مبلغ حتما به عدد وارد شده باشد و پیام موفقیت به کاربر نمایش داده میشود.

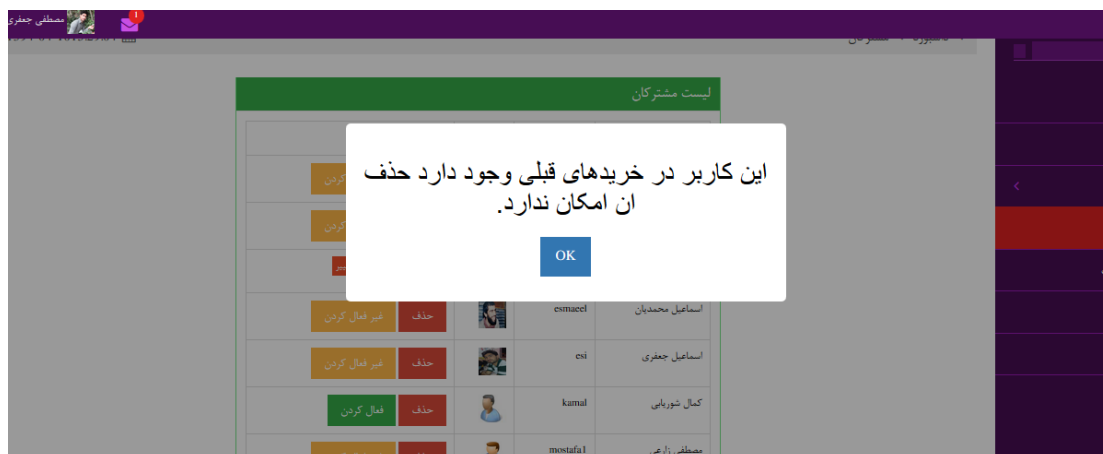
۹- قسمت مشترکان:

در این قسمت کاربر ویژه میتواند کاربران دیگر(هم خانه ای ها) را اضافه کند و در صورت لزوم آنها را حذف کند و یا اکانت آنها را غیر فعال کند:

نام	نام کاربری	تصویر	گزینه ها
سعید شاد	saeed		حذف غیر فعال کردن
روح اله ملکی	roohy		حذف غیر فعال کردن
مصطفی جعفری	mostafa		حذف غیر فعال کردن
اسماعیل محمدیان	esmaeil		حذف فعال کردن
کمال شوریانی	kamal		حذف غیر فعال کردن
مصطفی زارعی	mostafa1		حذف غیر فعال کردن

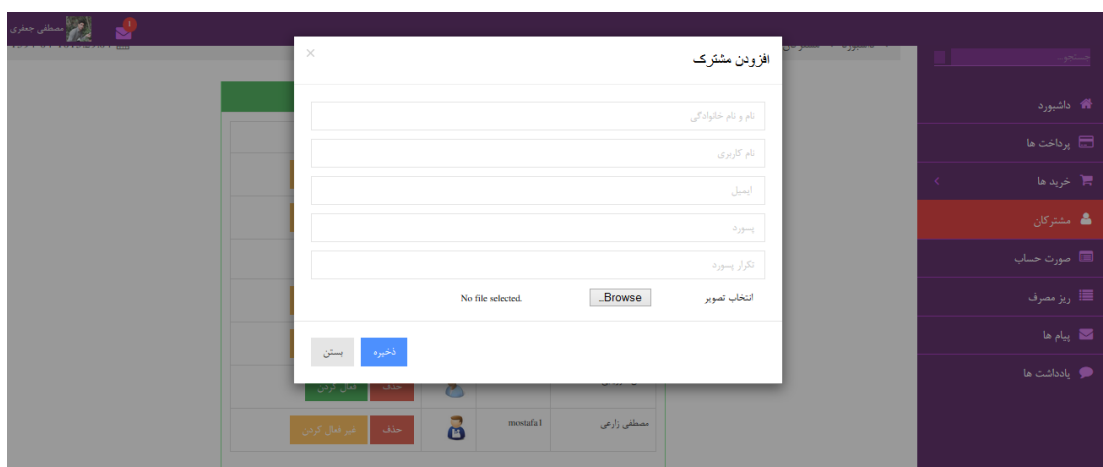
افزودن مشترک

در قسمت مشترکان همان طور که در تصویر بالا مشاهده میکنید لیستی از مشترکین به کاربر نمایش داده میشود که هرکدام دارای دو دکمه ی حذف و غیر فعال کردن میباشد که کاربری که مدیر گروه میباشد غیر قابل حذف کردن و غیر قابل غیر فعال کردن میباشد. حذف یک کاربر در صورتی امکان دارد که نام آن کاربر در خرید ها پرداخت های قبلی وجود نداشته باشد. در صورتی که کاربر در خرید های قبلی وجود داشته باشد و گزینه ی حذف زده شود یک پیام مانند شکل زیر به او نمایش داده میشود:

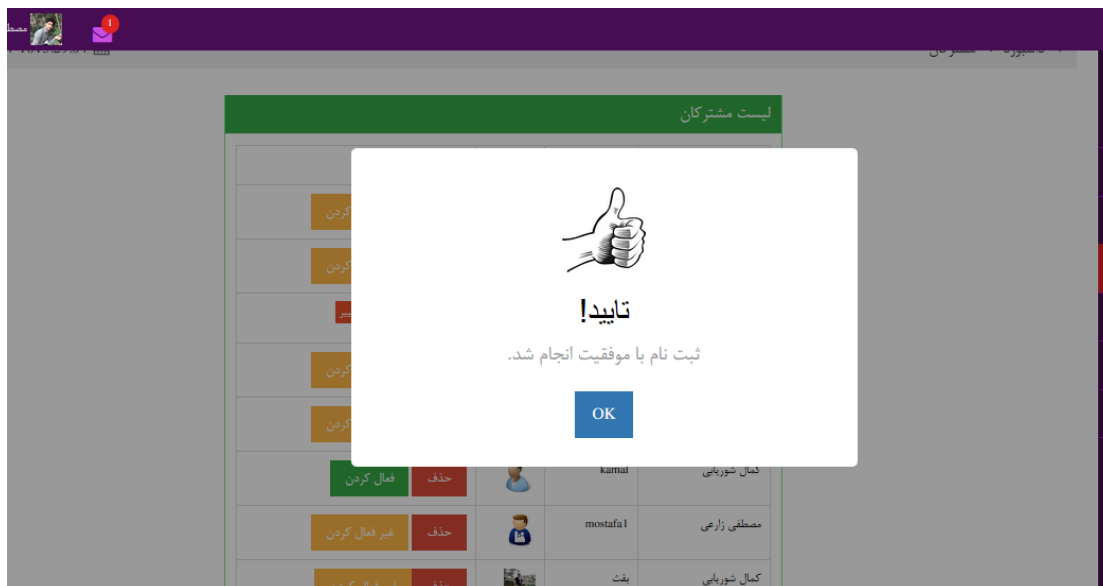


گزینه ی غیر فعال کردن برای این در نظر گرفته شده است که اگر یک کاربر در خرید های قبلی وجود داشته است و دیگر عضو یک گروه یا یک خانه نیست آن را غیر فعال میکنیم که در فرم های خرید و پرداخت نام آن کاربر نمایش داده نشود.و اگر دوباره بخواهیم آن را فعال کنیم با زدن گزینه ی فعال کردن این کار را انجام میدهیم.

در صورتی که بخواهیم یک کاربر جدید را اضافه کنیم دکمه ی افزودن مشترک را میزنیم و یک پنجره مودال باز میشود به شکل زیر:

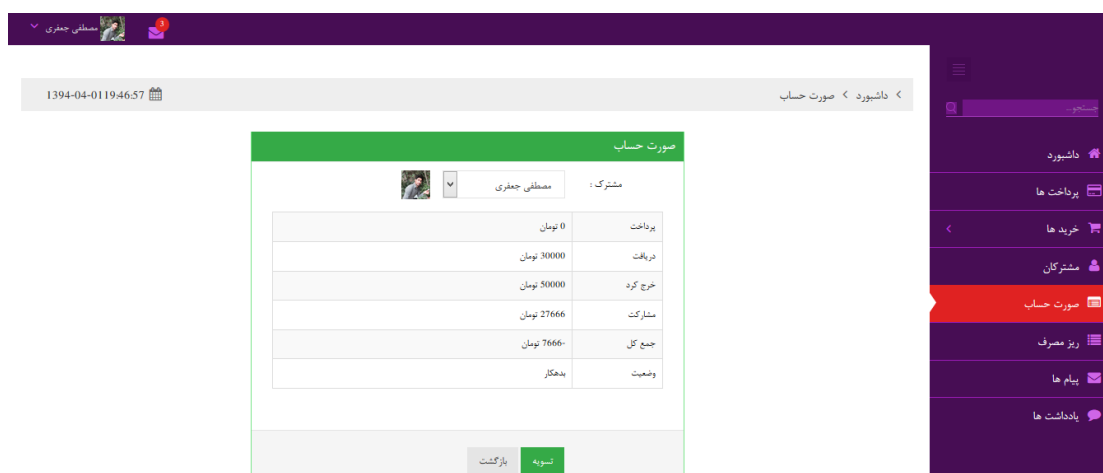


در این قسمت مشخصات کاربر را وارد کرده و دکمه ی ذخیره را میزنیم و در صورت نداشتن خطاهای از قبیل خالی بودن فیلدها و یا انتخاب نکردن تصویر و یا تکراری بودن نام کاربری پیام موفقیت آمیز بودن ثبت کاربر به صورت زیر نمایش داده میشود و کاربر به لیست مشترکان در همان لحظه اضافه میشود.



• ۱ - قسمت صورت حساب:

در این قسمت صورت حساب های تمام افراد را نمایش داده میشود و وضعیت بدهکار یا بستانکار بودن آنها را مشخص میکند:



برای دیدن صورت حساب هر فرد ابتدا نام آن فرد را در منوی بالا انتخاب میکنیم و با کلیک بر روی نام همان فرد تصویر آن مشترک در بالا کنار همان منو نمایش داده میشود و صورت حساب وی شامل یک جدول که دارای یک قسمت پرداخت میباشد که مشخص کننده ی مقدار پولی است که آن مشترک تا به حال به صورت دستی داده است غیر از خرید های که انجام داده است. قسمت دریافت در این جدول مقدار پولهای دریافتی آن مشترک را نمایش میدهد و قسمت مشارکت در این جدول مقدار خرید های که تا به حال آن مشترک انجام داده است را نمایش میدهد و آن خریدهای پای این مشترک حساب شده است و جمع کل در اصل مقدار پرداخت و مقدار خرج

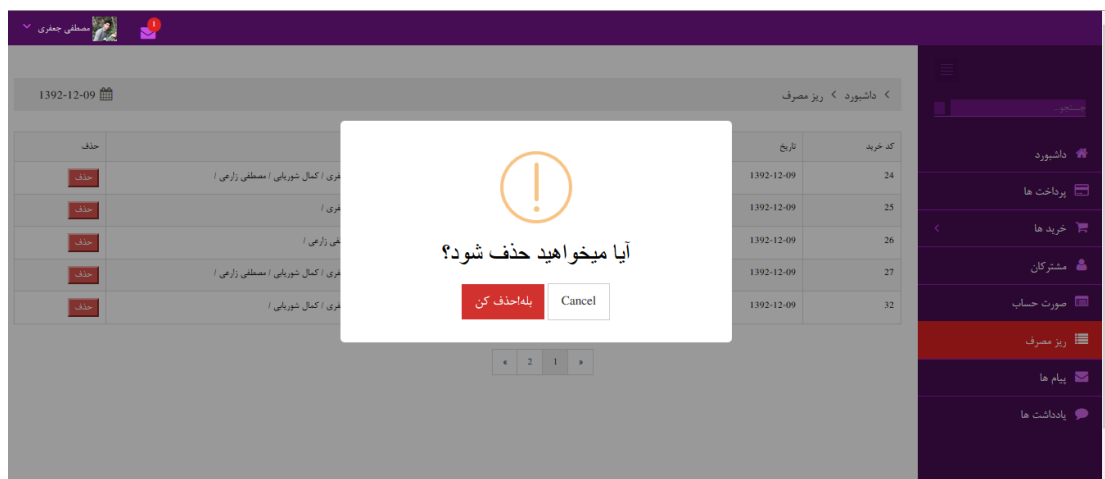
کرد مشترک را با هم جمع کرده است و مقدار مشارکت و مقدار دریافتی را از آن کم کرده است که اگر این مقدار عددی مثبت باشد وضعیت مشترک بستانکار میباشد ولی اگر این عدد منفی باشد وضعیت مشترک بدهکار میباشد و اگر آن عدد صفر باشد وضعیت مشترک تسویه می باشد. یک دکمه ی تسویه در قسمت صورت حساب در نظر گرفته شده است همان طور که در تصویر میبینید که با زدن این دکمه کاربر به صفحه ی پرداخت های هدایت میباشد که در آنجا اگر وضعیتش بدهکار باشد با پرداخت مقدار جمع کل حسابش تسویه میشود و اگر وضعیتش بستانکار باشد با دریافت مبلغ جمع کل وضعیتش تسویه میشود و حسابش تسویه میشود.

۱۱- قسمت ریز مصرف:

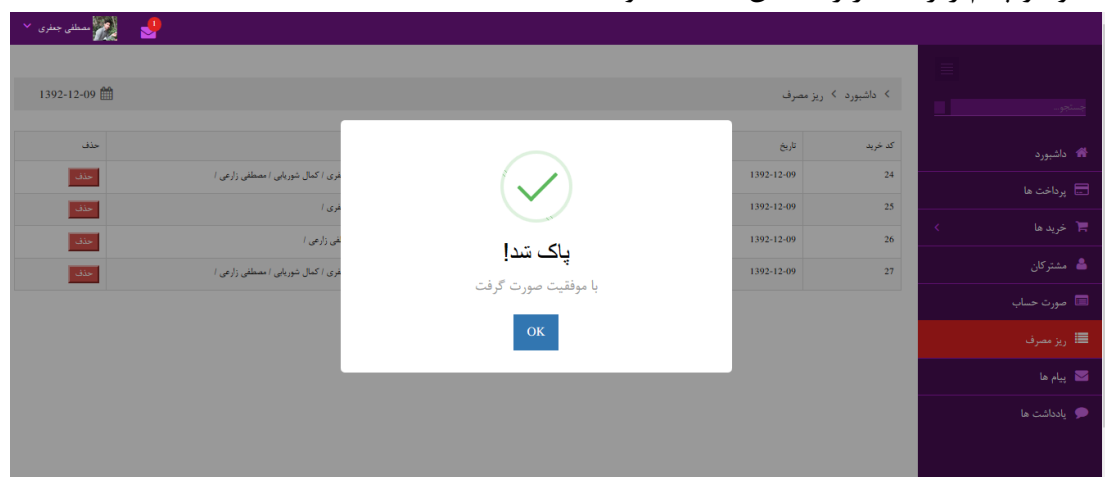
در این قسمت کاربر میتواند لیست تمام خرید های ثبت شده را ببیند و در صورت لزوم هر کدام را خواست بتواند حذف کند:

کد خرید	تاریخ	کالا	مبلغ	حذف
24	1392-12-09	نان و تخم مرغ	5000	حذف
25	1392-12-09	شیر موز و کیک	20000	حذف
26	1392-12-09	ماکارونی و رب	6000	حذف
27	1392-12-09	آب تصفیه و کزابه ماشین	8000	حذف
30	1392-12-09	ساکتویچ	6000	حذف

در این صفحه کاربر لیست تمام خرید ها را در یک جدول مشاهده میکند که این جدول شامل شش ستون میباشد که ستون اول کد خرید میباشد و ستون دوم تاریخ خرید را مشخص میکند و ستون بعدی اجناس خریده شده را نمایش میدهد و ستون بعدی مبلغ خرید را نمایش میدهد و ستون بعدی افرادی که در آن خرید بوده اند را نمایش میدهد و ستون بعدی یک دکمه به نام حذف دارد که در صورت زدن آن دکمه یک پیام به شکل زیر به کاربر نمایش داده میشود:



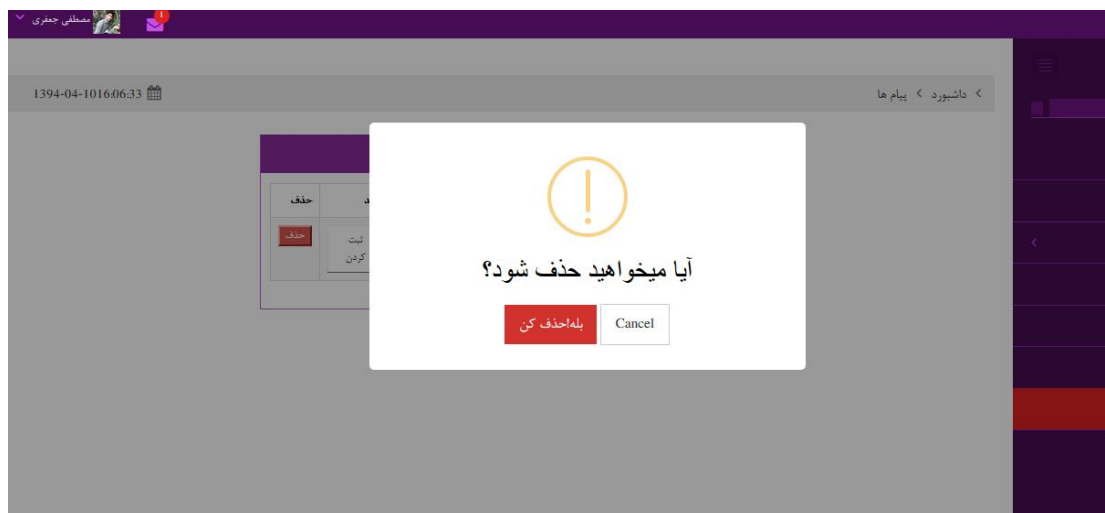
که اگر ما گزینه ی کنسل را بزنیم کاری انجام نمیشود و اگر گزینه ی بله را بزنیم آن خرید حذف میشود و پیام زیر به کاربر نمایش داده میشود:



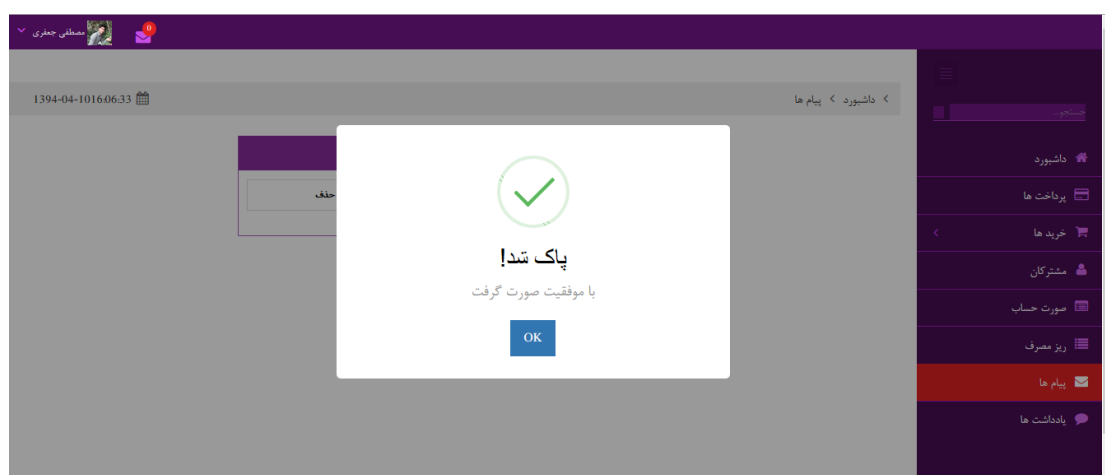
وقتی گزینه حذف را میزنیم آن خرید کلاً از حساب افراد پاک میشود و مبالغی که به خاطر آن خرید به حساب افراد اضافه شده بود از حساب آنها کم میشود.

۱۲ – قسمت پیام ها:

در این قسمت کاربر میتواند پیام های دریافتی خود را ببیند و آنها را مدیریت کند:



که با زدن دکمه ی کنسل عملیات کنسل میشود و با زدن دکمه ی بله آن پیام حذف میشود و یک پیغام به شکل زیر به کاربر نمایش داده میشود:



۱۳- قسمت یادداشت ها:

در این قسمت کاربر میتواند لیست یادداشت های خود را مشاهده کند و یادداشت های خود را ثبت کند:

افزودن یادداشت

عنوان

زمان یادآوری

متن

بستن ذخیره

در قسمت یادداشت ها تمام یادداشت ها در یک جدول چهارستونه به کاربر نمایش داده میشود که ستون اول نام و یا عنوان ان یادداشت را مشخص میکند و ستون بعدی زمان یادآوری آن یادداشت را مشخص میکند و ستون بعدی متن آن یادداشت را نمایش میدهد و ستون بعدی شامل یک دکمه ی حذف میباشد که در صورت تمایل کاربر میتواند آن یادداشت را حذف کند. یک دکمه ی سبز رنگ با نام افزودن یادداشت در پایین لیست یادداشت ها همان طور که در شکل میبینید قرار دارد که با کلیک کردن روی آن یک پنجره مودال به صورت زیر به کاربر نمایش داده میشود:

افزودن یادداشت

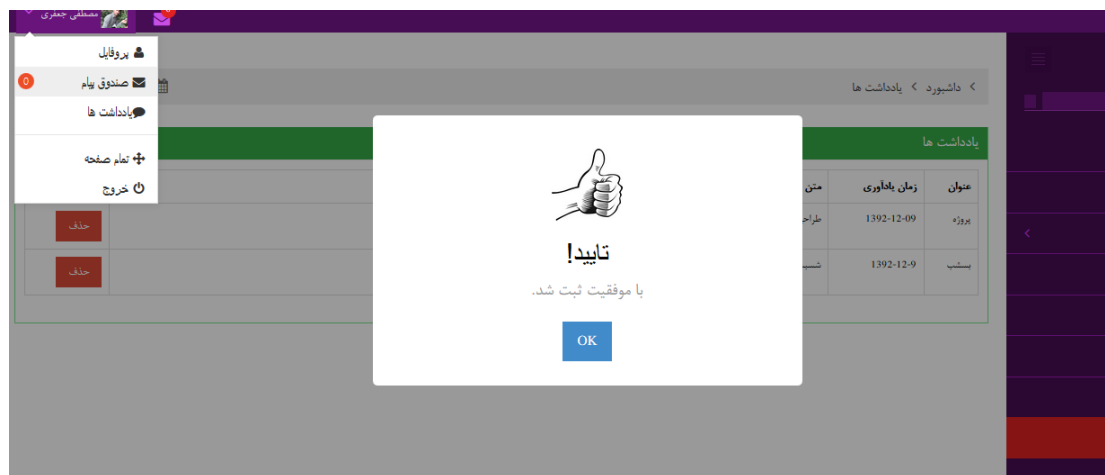
عنوان

زمان یادآوری

متن

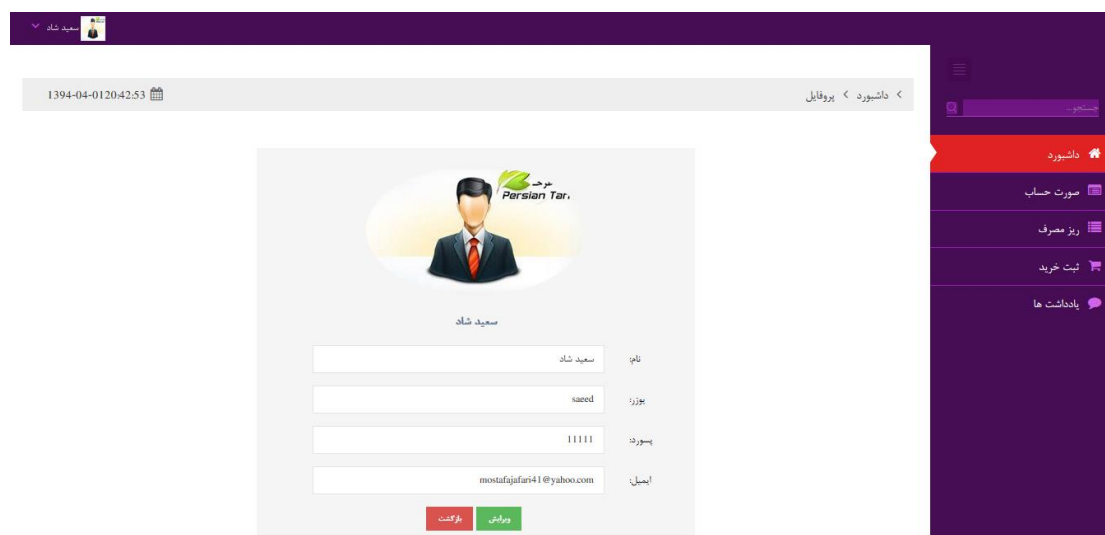
بستن ذخیره

که کاربر باید در قسمت عنوان،موضوع و یا نام آن یادداشت خود را وارد کند و در قسمت زمان یادآوری تاریخی که باید به کاربر یادآوری شود را وارد کند و در قسمت متن،متن آن یادداشت را وارد میکند و سپس دکمه ی ذخیره را میزند و یک پیغام که یادداشت با موفقیت ثبت شد مانند شکل زیر به کاربر نمایش داده میشود و همان لحظه به لیست یادداشت ها اضافه میشود:

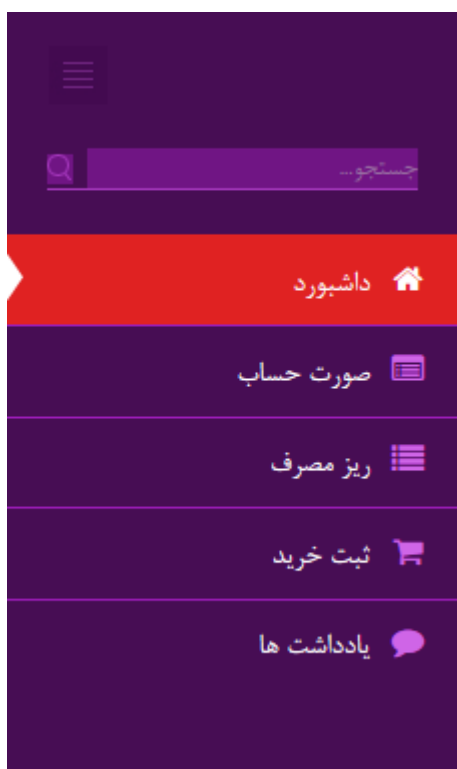


۱۴- قسمت پروفایل:

در قسمت پروفایل کاربر میتواند اطلاعات خود را شامل نام و نام کاربری و پسورد و تصویر و ایمیل را مشاهده و در صورت لزوم ویرایش کند:

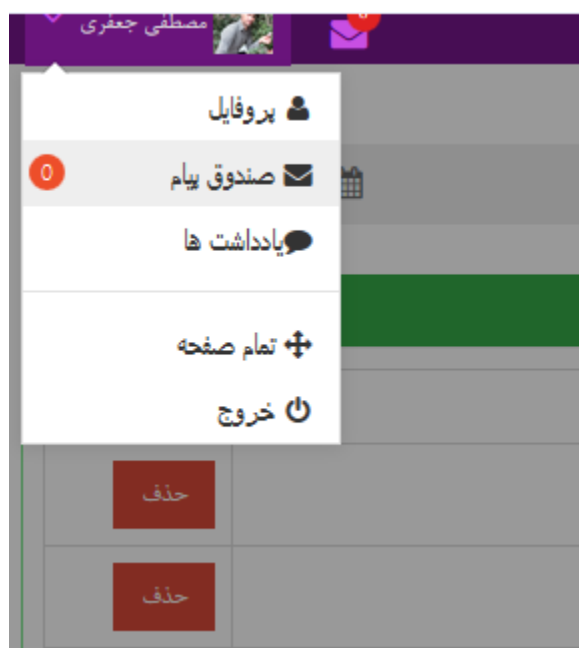


۱۵- کاربر معمولی در قسمت داشبورد خود همان طور که در شکل زیر میبینید فقط قادر به مشاهده صورت حساب افراد گروه و ریز مصرف خرید ها و ارسال خرید های خود به مدیر گروه در قالب پیام و همچنین قسمت یادداشت ها هم برای آن در نظر گرفته شده است.



۱۶- در قسمت منوی بالا:

کنار نام و تصویر کاربر یک منویی وجود دارد که در صورت رفتن ماوس روی نام طرف باز میشود که به شکل زیر میباشد:



این منو شامل پنج گزینه ی :پرو فایل و صندوق پیام و یادداشت ها و تمام صفحه و خروج میباشد. که با کلیک روی گزینه پرو فایل کاربر به صفحه پرو فایل خودش هدایت میشود و با کلیک کردن روی گزینه ی صندوق پیام به صفحه ی پیام ها هدایت میشود و با کلیک کردن روی گزینه یادداشت ها به صفحه یادداشت ها هدایت میشود و با زدن گزینه تمام صفحه شکل نمایش دادن صفحه تغییر میکند و منوهای اضافی مربوط به مرورگر دیگر نمایش داده نمیشوند و با کلیک کردن روی گزینه خروج کاربر از داشبورد خود خارج میشود.

۱۷- قسمت اخبارها و اطلاعاتیه ها:

در این قسمت مدیر سایت میتواند اخبار سایت را قرار دهد و اگر تصویر خاصی برای آن در نظر دارد روی سایت قرار دهد:

[خانه](#) [درباره ما](#) [ارتباط با ما](#) [راهنمای سایت](#) [ثبت نام](#)

چگونه پولدار شویم!

لازم نیست حتما مرد و یا یک خانم شغل و موفق باشید نکاتی که می خوانید نکاتی ساده اما پایه ی اصلی پولدار شدن برای هر کسی است.

ذخیره کردن را آغاز کنید:

این بدان معنا نیست که پولهای اضافی خود را جمع و ذخیره کنید بلکه قانون اصلی این تخریدن لازم اضافه و جلوگیری از ولخرجی است. یعنی زمانی که به بیرون از خانه میروید و در محل عبور خود به فروشگاه کفش و یا عطر فروشی میرسید آن را ندید گرفته و از کنارش بی توجه عبور کنید. نکته ی ساده و البته مهم دیگر ذخیره کردن پولهای خورد است برای خود قلکی تهیه کرده و مختص پولهای با ارزش پایین تر قرار دهید بعد از مدتی خواهید دی که چقدر کمک کننده است و یا اگر پول درشت دارید تا حد امکان آن را خورد نکنید تا بستر خرج کردن آن فراهم نشود.

برای خود بودجه تعیین کنید:

اول با آخر همراه هر زمانی که شما پول مشخصی را دریافت میکنید همان موقع برایش برنامه ریزی کنید و برای خود حتی نقطه ی تشویق هم بگذارید که اگر بهتر از حد انتظار در دخل و خرج تان عمل کردید می توانید به تیک های موفقیت خود اضافه کنید.



نتایج و تجربیاتی که در پایان انجام این پروژه من بدست آوردم:

در طول انجام این پروژه من به مشکلات زیادی برخورد کردم که هر کدام از آنها تجربیات زیادی برای من داشتند که دو نمونه از مشکلاتی که خیلی بزرگتر بودند و از نظر خودم مهمترین را برای شما ذکر میکنم:

۱- یکی از مشکلات طراحی بد دیتابیس بود که باعث شد من در وسط کار پس از برخورد کردن با مشکل مجبور شوم دیتابیس را دوباره طراحی کنم و بعضی از کدنویسی ها را دوباره انجام دهم و به این نتیجه رسیدم که در طراحی دیتابیس باید همه جوانب را در نظر گرفت مثلاً اینکه اگر قرار باشد جوین انجام دهیم آیا امکان دارد هر اطلاعاتی که خواستیم را از جداول بخوانیم. و به این نتیجه رسیدم که یکی از قسمت های مهم طراحی وبسایت دیتابیس آن میباشد.

۲- یکی دیگر از مشکلاتی که به آن برخورد کردم نداشتن بکلاگ برای پروژه یا وبسایت بود که بعضی از جاها خودم دچار سردرگمی میشدم که الان باید کجای پروژه را انجام دهم و یا چه قسمت های دیگری از پروژه مانده است.

و به این نتیجه رسیدم که اگر قرار است یک پروژه کاری انجام دهم در آینده ابتدا باید بکلاگ محصول را آماده کنم و سپس از روی آن کارم را شروع کنم.

به پایان آمد این دفتر

حکایت بهمنان باقی است

پایان