

به نام حق

گزارش پروژه پایانترم مبانی برنامه نویسی

اعضای گروه : علی احمدی - امیرعلی امیدوار - علیرضا دهقان - شهاب الدین کیانی

مقدمه:

در دنیای امروزی تکنولوژی در گوشه گوشه‌ی زندگی ما قابل مشاهده است و در همه امور روزمره ما پیدا می‌شود. یکی از پیچیده ترین اموری که ما در امور روزمره به آن برخورد می‌کنیم، تحلیل هزینه‌های جمعی است که امکان خطای انسانی در بسیار متداول است. یکی مثال خوب از تحلیل هزینه‌ی جمعی مدیریت ساختمان است و ما در این پروژه در تلاش هستیم تا عملکرد مدیر ساختمان را بهبود ببخشیم و وی را در امور مدیریتی یاری کنیم.

اهداف پروژه :

هدف از پروژه ی مذکور ساخت برنامه ای است که به مدیر ساختمان در اداره ی امور مالی کمک کند

برنامه دستورات را از طریق کنسول دریافت میکند و خروجی را از کنسول نمایش میدهد

از شروع اجرای برنامه، کاربر میتواند دستورات خود را از طریق کنسول وارد کند این روند تا زمانی که کاربر دستور

exit را وارد نکند ادامه پیدا می کند

این برنامه دارای دو پایگاه داده است :

DATAAss.csv (1

DATAAss_notdivided.csv (2

که پایگاه داده ی شماره دو تراکنش هایی را که کاربر وارد می کند را شامل می شود

و پایگاه داده ی شماره یک تراکنش های تقسیم شده برحسب واحد ها را

پروسه‌ی برنامه:

برای شروع برنامه کاربر باید مسیر فایل "root"، داده‌ی تقسیم نشده "File_not_divided"، داده‌ی تقسیم شده "File_divided" و داده‌ی واحد ها "File_information" را تنظیم کند.

با آغاز برنامه کاربر با چهار گزینه ی :

1.Append

2.Report

3.Help

4.Exit

رو به رو می شود که به طور خلاصه :

کاربر با انتخاب گزینه ی اول (وارد کردن عدد 1) وارد قسمت دادن داده های اولیه یا همان رکورد های حساب داری به برنامه می شود.

در قسمت اول کاربر اطلاعات را (به صورت یک جمله ی کامل) با فرمت استرینگ به برنامه میدهد
این جمله به ترتیب از چپ به راست شامل موارد زیر میباشد:

append time amount category subCategory responsibleUnit relatedUnits div

description

که برنامه با توابعی همچون append_input و portal_userstr_to_div و ...

این اطلاعات رو از جمله استخراج کرده و در دو لیست متفاوت با عنوان "تقسیم شده" و "تقسیم نشده" میسازد

و برنامه دائما با این لیست ها کار میکند و اطلاعات رو با استفاده از توابع مختلفی اعم از save

در فایل های سی اس وی (دو پایگاه داده ای که در ابتدا ذکر شد) ذخیره میکند و برنامه با استفاده از این فایل ها

خواسته ی کاربر را برآورده میکند

اما در قسمت دوم (گزارش گیری) کاربر با انتخاب کردن یکی از گزینه های زیر خروجی مورد نظر را دریافت خواهد کرد

1.taraz_vahed

2.tarikh_tarakonesh

3.tahlil_hazine

4.tajmi

5.takhmin_hazine

کاربر با انتخاب رپورت وارد قسمت گزارش گیری برنامه می شود قسمت گزارش گیری خود شامل پنج قسمت است

قسمت یک : (taraz_vahed()) این تابع شامل دو حالت 0 و 1 است. در حالت 0 تابع تراز مالی هر کدام از واحد ها را به صورت جداگانه بر میگرداند و در کنسول پرینت می کند. در حالت 1 همه ی تراز مالی واحد ها رو جمع کرده تا موجودی ساختمان رو به دست بیاوریم و جلوتر از این اطلاعات استفاده کنیم

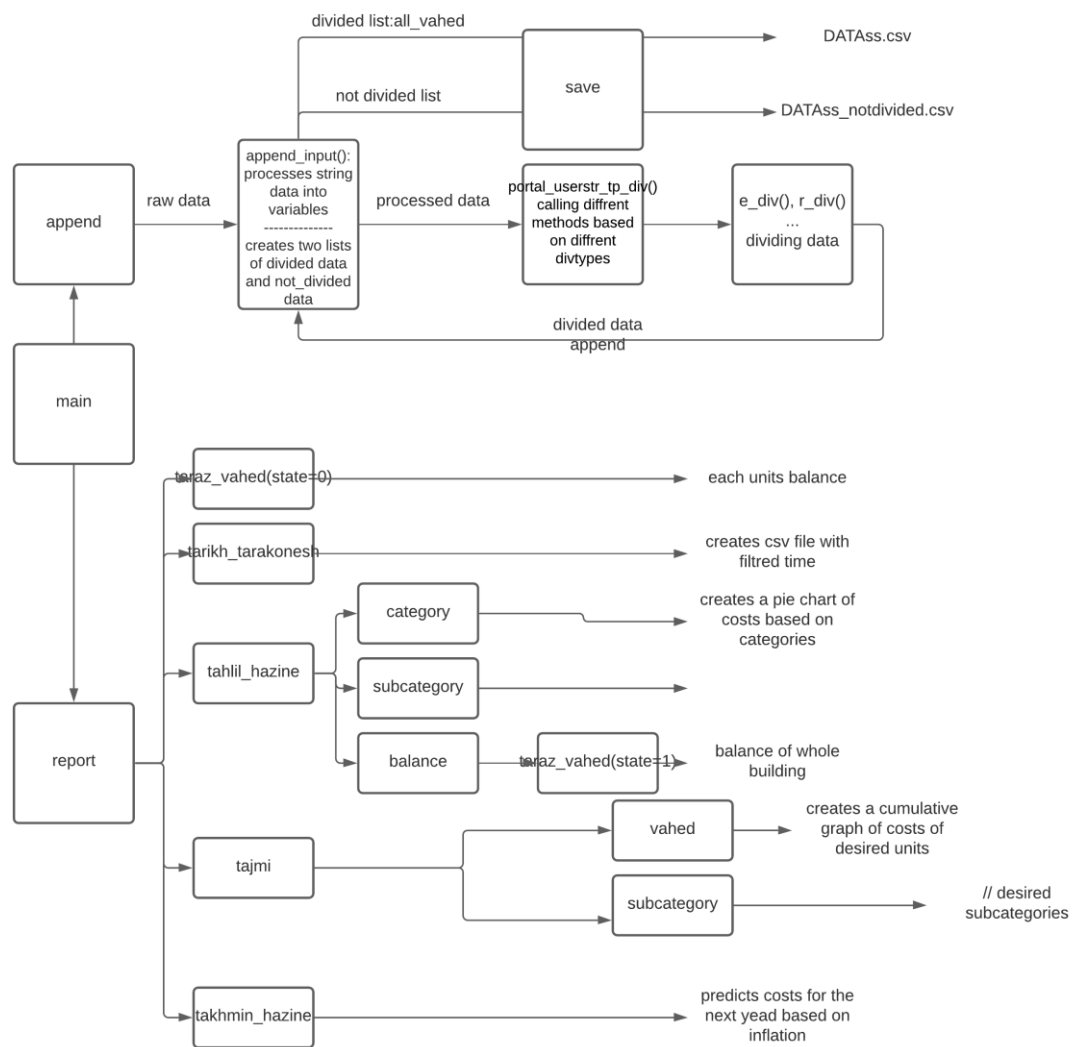
قسمت دوم : (tarikh_tarakonesh()) تراکنش ها رو بر اساس تاریخ تعیین شده توسط کاربر فیلتر کرده و در یک فایل سی اس وی ذخیره میکند.

قسمت سوم (tahlil_hazine class) : شامل سه بخش
بخش یک (category()) تحلیل هزینه ها براساس دسته ها، خروجی نمودار دایره ای هزینه ها تقسیم شده بر حسب دسته ها است.
بخش دو: (subcategory()) مانند بخش یک.
بخش سه: (balance()) با استفاده از حالت 1 موجودی ساختمان را حساب میکنند.

قسمت چهار: (tajmi class) شامل دو بخش:
بخش یک : (vahed()) با در نظر گرفتن فیلترهای ذکر شده کاربر روی زمان، دسته، زیر دسته، و تعداد واحد ها، نمودار تجمیعی هزینه ی واحد های ذکر شده را بر حسب زمان رسم می کند.
بخش دو : (subcategory()) : مانند بخش یک

قسمت پنج (takhmin_hazine) : تخمین شارژ برای سال بعد با محاسبه ی هزینه های امسال و در نظر گرفتن نرخ تورم (برای سال 99 نرخ تورم رسمی موجود نیست و فقط نرخ تورم 0 در نظر گرفته شده است)

توضیحات بالا را (گزینه ی یک و دو) در دیاگرام زیر میتوانید به طور ملموس تر مشاهده کنید:



کاربر با انتخاب گزینه ی سوم (کمک) با نحوه ی درست یا فرمت درست ورودی دادن به برنامه آشنا میشود همچنین با محدوده ی توانایی های برنامه برای مثال ، به این معنی که ، چه نوع تقسیم بندی هایی را با چه اسم هایی باید صدا بزند تا بتواند استفاده کند.

گزینه ی آخر (exit) هم که واضح است و نیازی به توضیح ندارد.

چالش‌ها:

به دلیل ناتوانی (pandas library) در پردازش تاریخ‌های شمسی یکی از چالش‌های تیم ما برطرف کردن این مشکل بود که با ایده زدن‌های گوناگون و متعدد به راه حل زیر دست یافتیم:

```
df[['year', 'month', 'day']] = pd.DataFrame([ x.split("/") for x in df['time'].tolist() ])
df['day'] = df['day'].str.lstrip('0')
df['month'] = df['month'].str.lstrip('0')
df['value']=(df['day'].astype('int32')+df['month'].astype('int32')*100+df['year'].astype('int32')*10000) #to determine
df=df.loc[(df['value'] >int(start[2])+int(start[1])*100+int(start[0])*10000) #pandas doesnt support shamsi calender b
          & (df['value'] <int(end[2])+int(end[1])*100+int(end[0])*10000) ] #nanoseconds as time units and uses 64bi
df.set_index('value', inplace=True) #so we used a system to filter for time(
df.sort_index()

df.drop('year', axis=1, inplace=True)
df.drop('month', axis=1, inplace=True)
df.drop('day', axis=1, inplace=True)
```

راجب توابع (reformat , reformat) : از اونجایی که تیم ما با یک مدل دیگری و متفاوت با فرمت داده‌هایی که شما در روز‌های پایانی پروژه به ما تحویل دادید کار کرده بودیم و کد زده بودیم مجبور شدیم که با درسی‌هایی ن‌چندان کم در مقابل این تغییر منعطف عمل کنیم.

```
#####
def reformat(file_not_divided,file_divided):
#reformat file divide shade
mm=pd.read_excel(file_divided)
df=pd.DataFrame(mm)
df.columns=['vahed','amount','time','category','subCategory','div','description','responsibleUnit']
#taviz bazi az stoona ha
df.vahed=df.time
df.time=df.category
df.amount=df.responsibleUnit
df.category=df.subCategory
df.responsibleUnit=''
df.description=''
df.time=df.time.str.replace('-', '/')
df.vahed=df.vahed.str.replace('id','vahed')
df.subCategory=df['div'].astype(str)
df=df.drop('div',1)
df['div']=''
#tavize category ha
a=['ghabz','nezafat','asansoor','hazine_parking','tamirat','sharj','other']
b=['ab','bargh','gas','avarez']
aa=['Ghabz','nezafat','asanson','parking','tamirat','sharj','other'] # sharj nabud too dataset jadid
bb=['water','bargh','gaz','avarez'] # ghabz avarez ham dare!
###
for i in range(len(a)):
    df['category']=df['category'].str.replace(aa[i],a[i])

for i in range(len(b)):
    df['subCategory']=df['subCategory'].str.replace(bb[i],b[i])
df=df.drop('description',1)
df=df.drop('responsibleUnit',1)

df['description']=''
df['responsibleUnit']=''
```

```

file_divided_new=file_divided.replace('xlsx','csv') #avaz kardane format be csv chon bishtare barname ba csv kar mikone

df.to_csv(file_divided_new,index=False)
df = pd.read_csv(file_divided_new, #bazi error be khatere reformat pish miomad ke in haleshoon mikone
lineterminator='\n')
df.to_csv(file_divided_new,index=False)

reformat file divide nashode
mm=pd.read_excel(file_not_divided)
df=pd.DataFrame(mm)
#taghire bazi sotoonha
df=df.drop(list(df.columns)[0],1)
df.date=df.date.str.replace('-', '/')
df=df.drop('id',1)
df.zirdaste,df.mablagh=df.mablagh,df.zirdaste
df.daste,df.zirdaste=df.zirdaste,df.daste
df.rename(columns = {'daste':'amount'}, inplace = True)
df.rename(columns = {'zirdaste':'category'}, inplace = True)
df.rename(columns={'mablagh':'subcategory'},inplace=True)
df['resunit']=''
#taghir bazi category ha
a=['ghabz','nezafat','asansoor','hazine_parking','tamirat','sharj','other']
b=['ab','bargh','gas','avarez']
aa=['ghabz','nezafat','asansoor','parking','tamirat','sharj','other'] # sharj nabud too excel
bb=['water','bargh','gaz','avarez'] # ghabz avarez ham dare!
###
for i in range(len(a)):
    df['category']=df['category'].str.replace(aa[i],a[i])

for i in range(len(b)):
    df['subcategory']=df['subcategory'].str.replace(bb[i],b[i])

```

```

def reformat(x): #taghir format related unit
    ans=''
    for i in eval(x):
        ans+=i[2:]+','
    ans=ans[:-1]
    return ans

df['relateunit']=df.name.apply(reformat)
df=df.drop('name',1)
df['divtype']=''
df['des']=''

file_not_divided_new=file_not_divided.replace('xlsx','csv') #avaz kardane format be csv chon bishtare barname ba csv kar i

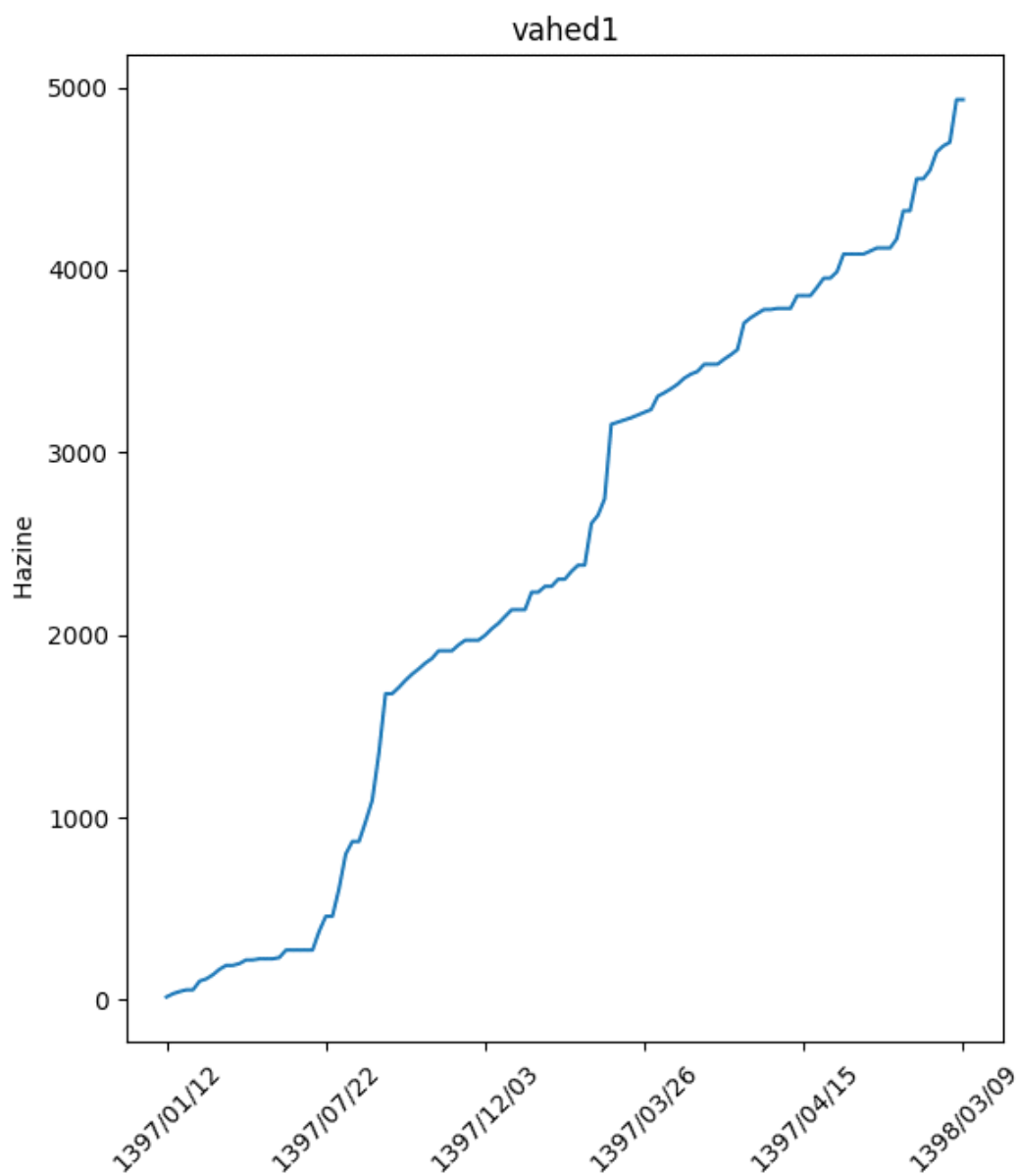
df.to_csv(file_not_divided_new,index=False)
df = pd.read_csv(file_not_divided_new, #bazi error be khatere reformat pish miomad ke in haleshoon mikone
lineterminator='\n')
df.to_csv(file_not_divided_new,index=False)
return file_not_divided_new,file_divided_new #barae taghire masire file ha be csv sakhte shode

```

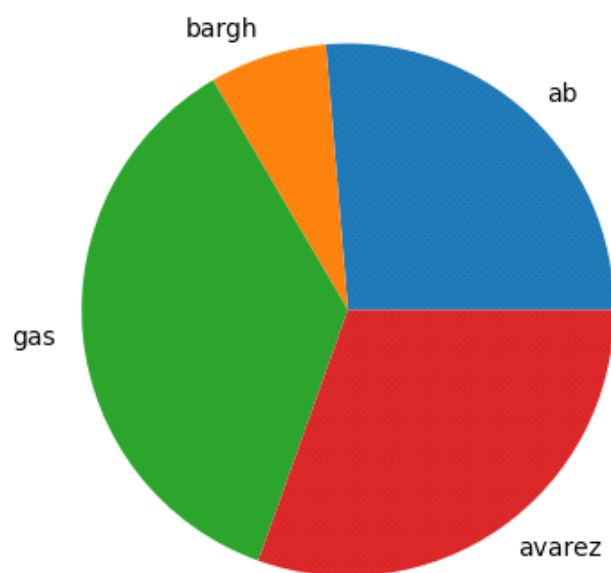
جلوهای بصری:

برای زیبایی کار و همان طور که در داک خواسته شده بود با استفاده از داده های تحویل گرفته شده دو نمودار را در زیر مشاهده می کنید :

نمودار تجمیعی هزینه های واحد "یک" در زمان های مشخص شده :



تحليل هزینه بر حسب زیر دسته :



نحوه‌ی تقسیم بندی:

و اما در نهایت گزارش تقسیم کار ، روند پیشروی تیم ما اینگونه بود که نخست ، قسمت (append) و داده گیری و نحوه تقسیم بندی هزینه ها و سیو کردن آنها روی فایل سی اس وی و.... توسط شهاب الدین کیانی و علیرضا دهقان انجام شد

قسمت گزارش گیری (report) و تعیین سهم بخش ها و پیش بینی هزینه ی پرداختی هر واحد در سال آینده با استفاده از نرخ تورم که فعالیتی اختیاری بود و.... توسط امیرعلی امیدوار و علی احمدی انجام شد

اما بعد از نوشتن کد های اولیه کل تیم (هر چهار نفر) روی هر دو قسمت نظارت کرد و تغییراتی اعمال کردیم و باگ های موجود رو دیباگ کردیم ، به این صورت که هر چهار نفر با تغییرات اعمالی روی کد ها ، قسمت اول (append) و قسمت دوم (report) را نهایی کرده است.

مسئولیت چک کردن نهایی برنامه برای تحویل و گزارش نویسی هم به عهده ی علیرضا دهقان و امیرعلی امیدوار و شهاب الدین کیانی بوده است.

تشکر از توجه شما

یا حق