گزارش کار پروژه

مبانی برنامه سازی کامپیوتری پایتون و کار با جدول داده

استاد درس **علیرضا کدیور**

اعضای گروه **سپهر حرفی ، مهدیه شهابی پور ، حنانه ملکوتی**

هدف اصلی این پروژه کمک به اداره یک ساختمان است.در این پروژه با کمک برنامه نویسی در پایتون گزارشی از مخارج و روند هزینه ها نمایش داده میشود.مدیر ساختمان میتواند با دادن دستور های مختلف به این برنامه گزارش هایی به صورت زیر دریافت کند:

- تراز مالی
- صورت حساب
- سهم بخش ها
- روند هزینه ها
- پیش بینی هزینه ها

فواید این برنامه سریعتر شدن محاسبات دخل و خرج یک ساختمان است که با دادن یک دستور گزارش را فورا به کاربر می دهد.

خواسته ها و توابع در فایلی جدا به نا orders وجود دارد و فایل اصلی به نام bpProject است که به وسیله whileدستور هارا اجرا میکند.

این برنامه قابلیت اضافه کردن داده های جدید توسط کاربر و یا گرفتن دستور های مختلف را دارد که به یک مدیر ساختمان کمک میکند تا وضعیت ساختمان را دریابد.فقط کافیست برنامه را runکند و دستور دلخواه خود وارد کند.و برنامه تا زمانی که کاربر دستور exitرا وارد نکند کار میکند.

همچنین اگر کاربر دستوری خارج از فرمت مورد نظر برنامه بدهد، با ارور ' Oops... ! Please type your order in the right format!' مواجه میشود.

اولین چالش ما برای این پروژه فهم آن بود.که شامل تشخیص ورودی ها و خروجی ها و همجنین درک هر یک از دستور ها خواسته شده در فایل بود.

دومین چالش ما تقسیم کار بین اعضای گروه بود.که این تقسیم بندی به صورت زیر انجام شد:

- سپهر حرفی:گرفتن ورودی گرفتن از کاربر، تابع تقسیم بندی هزینه ها برای هر واحد،تابع روند هزینه ها
- مهدیه شهابی پور:تابع ترازمالی،تابع پیشبینی هزینه ها
- حنانه ملکوتی:تابع صورت حساب،تابع گزارش و سهم بخش ها

چالش های دیگری نیز وجود داشت که در بخش مربوط به آن تابع به آن اشاره شده است.

مرحله کد زدن با شروع از نحوه ورودی گرفتن از کاربر شروع شد و کد توابع توسط اعضای گروه زده شد و توسط while در فایل اصلی با نوشتن شروط از این توابع استفاده شده است.

برای درک بهتر اطلاعات خواسته شده نمونه هایی از گزارش ها برای ساختمان ها نیز بررسی کرده ایم تا بتوانیم گزارشی نزدیک به واقعیت را نمایش دهیم و در آخر با داده های فرضی که به ما داده شد برنامه خود را چک کردیم.

در ادامه به توضیح هر یک از توابع نوشته شده میپردازیم.

داده گیری از کاربر

کاربر با وارد کردن دستورappend می‌تواند وارد بخش داده گرفتن بشود . در ابتدا از کاربر تاریخ پرسیده میشود که می‌تواند now را وارد کردن و سپس با گرفتن واحد و مبلغ و نحوه دسته بندی و گروه و زیرگروه این بخش به پایان می‌رسد.

در آخر داده ها در فایل data 2 سیو شده و همچنین با محاسبه سهم هر واحد به طور جداگانه (با توجه به دسته بندی مربوطه) داده ها در data 3 سیو میشوند که بعدا قابل دسترسی باشند .

در زیر می‌توانید نمونه ورودی این قسمت را ببینید :

```
please type your order!>? append
enter date (you can put now!)>? now
Category?>? Ghabz
SubCategory?(type none if there isnt any!)>? Water
Total Amount?>? 450
please type the units that are related with space!(like : id1 id2 id3)>? id1 id3 id4 id7
kind of div?(--d , --a , --p , --r , --e)>? --d
```

نمونه ورودی برای داده وارد کردن

155	155	1399-11-19	Ghabz	Water	450	['id1', 'id3', 'id4', 'id7']

داده وارد شده در جدول Data 2

1000	155	id3	1399-11-19	Ghabz	Water	450	150.00000
1001	155	id4	1399-11-19	Ghabz	Water	450	150.00000
1002	155	id1	1399-11-19	Ghabz	Water	450	75.00000
1003	155	id7	1399-11-19	Ghabz	Water	450	75.00000

داده وارد شده در جدول **Data 3**

تابع تقسیم‌بندی

در این تابع که برای بخشی از گرفتن داده نوشته شده است هزینه ها را بر حسب تقسیم‌بندی مورد نظر کاربر تقسیم میکنیم که میتواند بر حسب تعداد افراد و مساحت و پارکینگ و طبقه یا به طول کاملا مساوی تقسیم بشود (البته گزینه ای هم وجود دارد که کاربر درصد هر واحد را به صورت یک دنباله با – وارد کند برای مثال : ۳۰-۲۰-۵۰) ولی قابلیت دیگری که این تابع دارد این است که کاربر حالت دیفالت را انتخاب کند که در این صورت در تابع بر حسب دسته بندی داده ای که کاربر وارد کرده است هزینه ها بین واحد های مورد نظر تقسیم میشوند که برای دسته بندی قبض (به غیر از عوارض که به صورت مساوی تقسیم میشود) این تقسیم بندی بر حسب تعداد افراد می‌باشد (زیرا هزینه برق و آب و گاز به نسبت افراد زیاد یا کم میشود .) همچنین برای دسته بندی نظافت از تقسیم بندی بر اساس مساحت و برای تعمیرات از تقسیم بندی به صورت مساوی و برای آسانسور تقسیم بندی بر اساس طبقه واحد های مربوط انجام میشود زیرا هر چه طبقه بیشتر باشد بیشتر از آسانسور استفاده خواهد شد.

صورت حساب

در این بخش کاربر با دادن دستور bill به برنامه درخواست صورت حساب میکند.

به دنبال آن دو زمان را به برنامه میدهد سپس از کاربر پرسیده میشود که صورت حساب را برای چه واحد هایی میخواهد.کاربر با مشخص کردن واحد های مد نظرش(که می تواند کل واحد ها باشد با دستور all)،هزینه ها در آن بازه زمانی برای هرواحد نمایش داده میشود.این هزینه ها شامل هزینه های دسته بندی ها و زیر گروه های آنها است.

و در آخر از کاربر پرسیده میشود که آیا میخواهد صورت حساب را ذخیره کند یا خیر.اگر جواب بله بود صورت حساب به صورت یک فایل csv ذخیره میشود.

در ادامه یک مثال از خروجی این تابع نمایش داده شده است.

صورت حساب در بازه 1397/01/01 تا 1398/12/29 برای واحد های 1و2و3و5:

```
please type your order!bill 1397-01-01 1398-12-29

which units?(also you can put all!)id1 id2 id3 id5
```

نمونه ورودی

```
     asansor  nezafat  other  parking  tamirat  Water  avarez  bargh   gaz
id1      772     1318    120       36      244    260     583     96  1504
id2      772     1146    225       36      244    265     643     56   680
id3      772     1248     64       36      244    857     257    160   828
id5      772     1405    153       18      244   1312    1240    237  1042

do you want to save the bill? (yes , no)yes
```

نمونه خروجی

قسمت چالشی این بخش کارکردن با تاریخ شمسی بود،که با استفاده از jdatetime انجام داده ام.

همچنین در این بخش صورتحساب برای هر واحد نوشته شده است که حتی میتوانستیم بدون تفکیک واحد بنویسیم.

سهم بخش ها

در ایـن بخـش کـاربر بـا دادن دسـتور report درخواسـت گزارشـی از سـهم هزینـه هـا و اعـلام وضـعیت سـاختمان میکند.

ابتدا جدولی برای سهم هزینه هر یک از زیر گـروه هـای یـک دسـته بنـدی بـه صـورت نسـبت هزینـه آن بـه هزینـه کـل دسته بندی داده میشود و در ادامه جـدولی دیگـر بـرای سـهم هزینـه هـر یـک از دسـته بنـدی هـا بـه صـورت نسـبت آن ها به کل هزینه نمایش داده میشود.

در ادامـه موجـودی سـاختمان نشـان داده میشود.وضـعیت کلـی سـاختمان بـه صـورت قرمـز،زرد ویـا سـبز مشـخص میشـود.که ایـن رنـگ هـا بـا توجـه بـه میـزان بـدهکار بـود واحـد هـا کـه در برنامـه مشـخص شـده اسـت نمـایش داده میشود.

شرط استفاده شده برای وضعیت ساختمان این است که اگر میزان بدهی بیش از نصف شارژ باشد وضعیت قرمز

اگر بین نصف و بیست درصد شارژ باشد زرد و در بقیه موارد سبز است.

در ادامه نمونه ای از عملکرد این تابع برای داده های فرضی یک ساختمان نمایش می دهیم.

```
please type your order!report
```

نمونه ورودی

```
this is the ratio of each SubCategory  :
SubCategory      Water     avarez     bargh        gaz
Ratio(%)     26.955343  30.201342  7.072793  35.770521
this is the ratio of each category :
Category      Ghabz    asansor    nezafat     other   parking  tamirat
Ratio(%)  53.452915  13.345981  23.613315  3.154536  0.981373  5.45188
the Existence of the building is :
-57980
the Statuse of the building is:
Red
```

نمونه خروجی

محاسبه تراز مالی به تفکیک واحد(ها)

در این بخش داده های جدول را بر اساس بازه زمانی ورودی فیلتر کرده و سپس مقادیر هزینه ها در تمام آن مدت را برای هر واحد جداگانه حساب کرده و از شارژ کم میشود و موجودی را اعلام کرده و وضعیت هر واحد در صورتی که مبلغ نهایی منفی باشد بدهکار و درصورتی که مثبت باشد بستانکار اعلام میشود.

```
please type your order!balancesheet 1397-01-01 1398-12-29
```

نمونه ورودی

	Unit	Amount	Condition
0	id1	-5361	debtor
1	id10	-5602	debtor
2	id2	-4408	debtor
3	id3	-5342	debtor
4	id4	-5785	debtor
5	id5	-7513	debtor
6	id6	-4963	debtor
7	id7	-5004	debtor
8	id8	-7192	debtor
9	id9	-5890	debtor

نمونه خروجی

بزرگترین چالش در این توابع وجود ابهام در تفاوت صورت حساب و تراز مالی بود و در بخش پیش بینی استفاده از اختلاف تاریخ های جدول (برای نمودار و خطی سازی آن)از یک تاریخ دلخواه که در اینجا1397/01/01 در نظر گرفته شده است و عدم امکان استفاده از دیتا ها در روز های ۳۱ ام بعضی از ماه ها به علت استفاده از متد های مربوط به تاریخ میلادی بود .

پیش بینی هزینه پرداختی هر واحد در سال آینده

در این بخش برای پیش بینی هزینه های ثابت در سال آینده ابتدا میانگین هزینه های سالانه هر دسته را برای هر واحد حساب کرده و سپس این مقدار میانگین را از مقدار تمام دسته های هر واحد در سال های مشابه کم کرده که میزان اختلاف از میانگین یا همان تاثیر تورم بر هزینه ها را داشته باشیم. این میزان تفاوت ها را در ستونی به نام 'Inflation' به جدول اضافه میکنیم.

برای محاسبه این مقادیر در سال های بعد ابتدا به نمودار جدول مربوط به هزینه های هر واحد برای هر دسته خطی فیت کرده و معادله آن خط را می یابیم و مقدار آن را در 182 روز بعد از انتهای بازه‌ی تاریخ ورودی پیدا می کنیم(چون خط است میانگین این مقادیر در سال آتی برابر مقدار نقطه میانی این بازه در خط است که معادل 182 روز است.).

همین روند خط فیت کردن را برای ستون 'Inflation' تکرار کرده و مقادیر تاثیر تورم برای هردسته برای هر واحد را در سال آینده محاسبه میکنیم.

درنهایت مقادیر تاثیر تورمی که برای هردسته حساب شده است با مقادیری که برای سال آینده پیش بینی شده است جمع میشود و به عنوان مبلغ نهایی اعلام میشود.

```
please type your order!constantprices 1397-01-01 1398-12-29
```

نمونه ورودی

	Year	Unit	Category	SubCategory	Amount	Inflation	Final Amount
0	1400	id1	Ghabz	Water	117.622613	-1.070183	116.552431
1	1400	id1	Ghabz	avarez	744.289349	0.000000	744.289349
2	1400	id1	Ghabz	bargh	22.748175	2.523715	25.271890
3	1400	id1	Ghabz	gaz	210.537820	-7.385596	203.152225
4	1400	id1	asansor	###	360.144868	-4.703908	355.440961
..	...	...	...	...	...	...	...
85	1400	id9	asansor	###	360.144868	-4.703908	355.440961
86	1400	id9	nezafat	###	72.705402	-6.285733	66.419669
87	1400	id9	other	###	27.881558	-6.248659	21.632899
88	1400	id9	parking	###	12.000000	-5.099364	6.900636
89	1400	id9	tamirat	###	36.939006	-6.631757	30.307248

نمونه خروجی

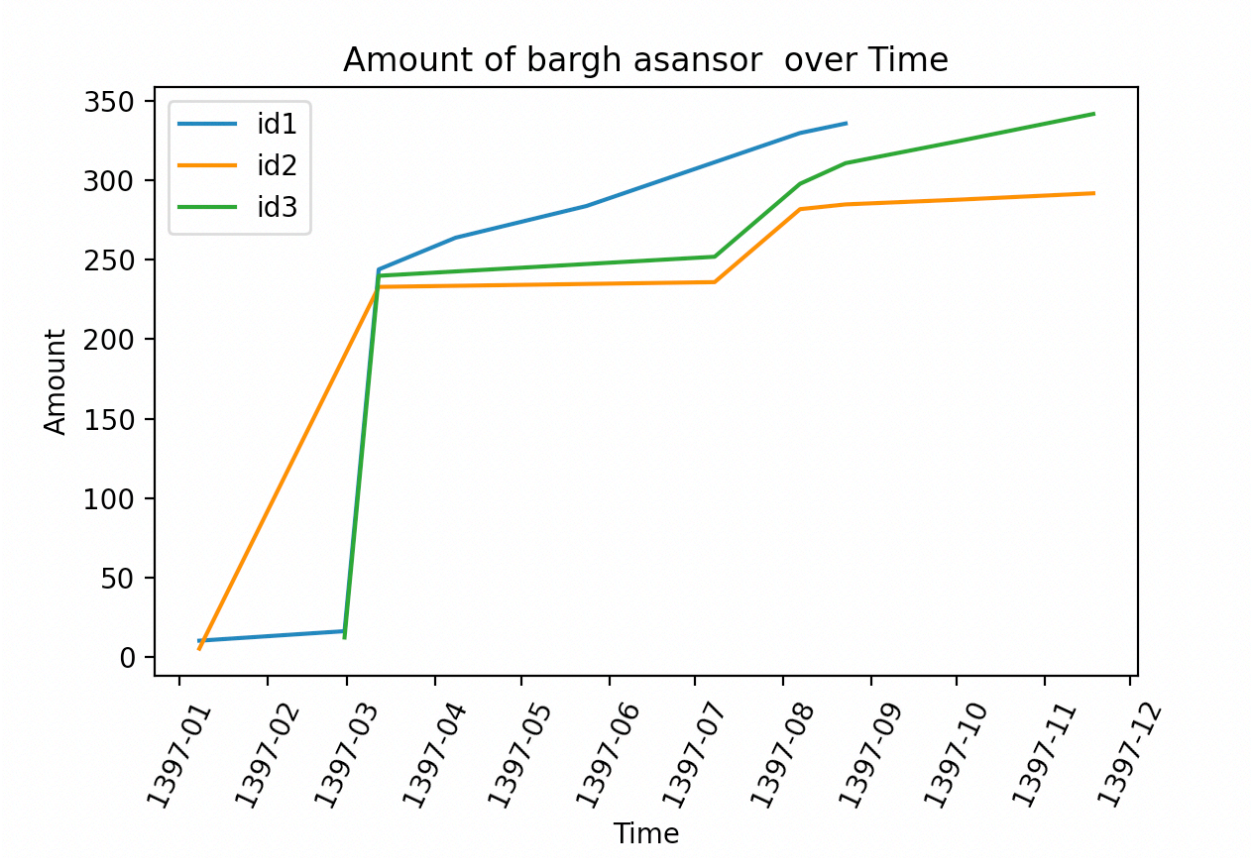
روند هزینه ها

الف – نمودار تجمعی هزینه های واحد های مختلف :

در این قسمت کاربر با وارد کردن دستور plot units و دادن دو تاریخ برای بازه ای که در آن نمودار کشیده میشود میتواند تابع را اجرا کند . سپس تابع با پرسیدن واحد ها و همچنین زیردسته‌بندی ها یا دسته بندی های مطلوب نمودار را رسم می‌نماید که می‌توانید در زیر یک نمونه از ورودی و خروجی را ببینید :

```
please type your order!>? plot units 1397-1-1 1398-1-1
please enter which units ? (you can write all!)>? id1 id2 id3
please enter which SubCategories or Categories? (you can write all!)>? bargh asansor
```

ورودی الف



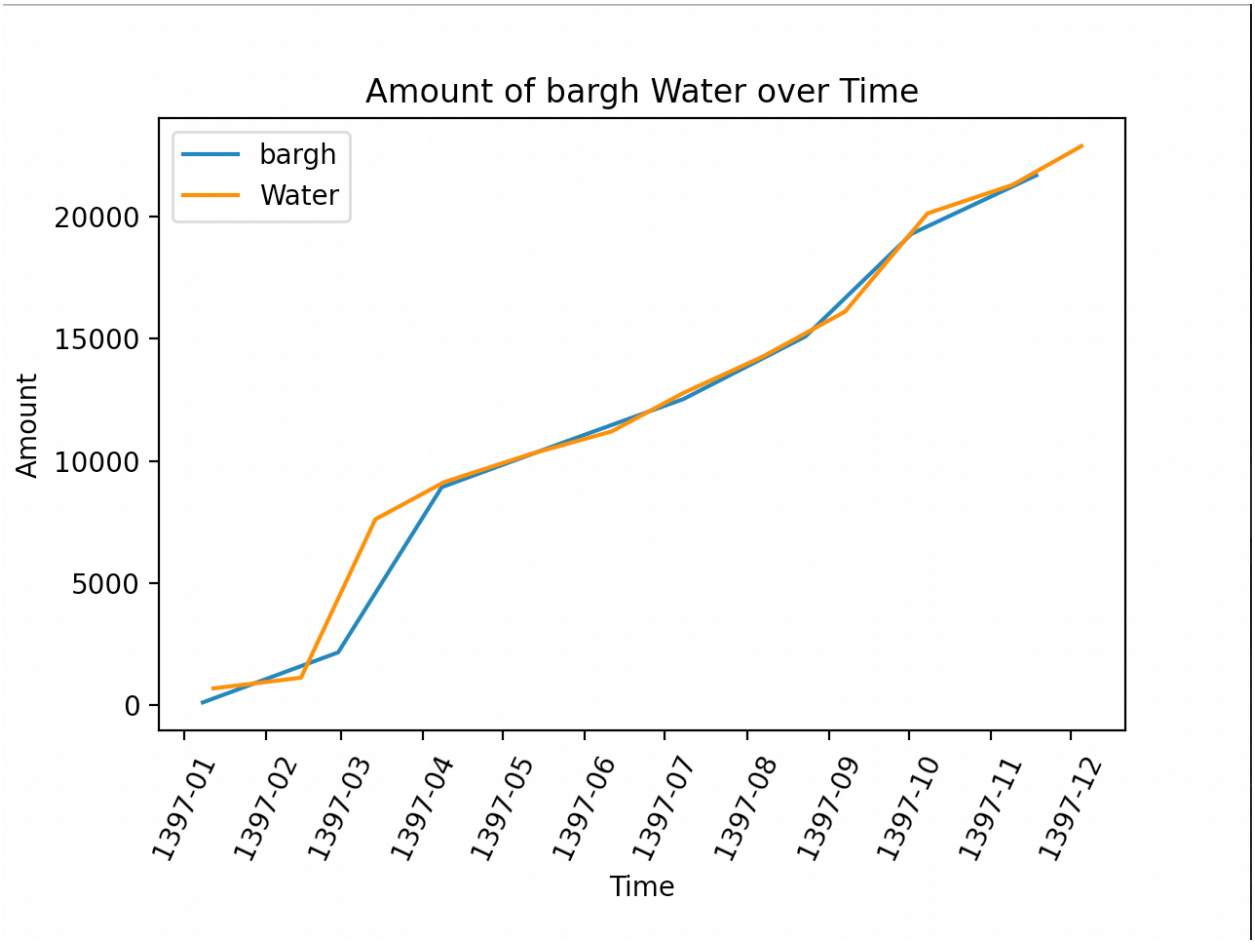
خروجی الف

ب – نمودار تجمعی هزینه های مربوط به زیرگروه های مختلف :

در این قسمت کاربر با وارد کردن دستور plot و دادن دو تاریخ برای بازه ای که در آن نمودار کشیده میشود میتواند تابع را اجرا کند . سپس تابع با پرسیدن زیر دسته بندی های مطلوب نمودار را رسم می‌نماید که
میتوانید در زیر یک نمونه از ورودی و خروجی را ببینید :

```
please type your order!>? plot 1397-1-1 1398-1-1
type SubCategories you want like: bargh Water gaz ... >? bargh Water
```

ورودی ب



خروجی ب

بزرگ ترین چالش در بخش رسم نمودار نمایش تاریخ های شمسی در محور افقی بود که این کار را با دستورات date2num و num2date کتابخانه matplotlib.dates انجام دادم.