



پروژه پایانی درس مبانی برنامه نویسی

استاد درس: علیرضا کدیور

راهنما پروژه: آرین احدی نیا

دانشجویان:

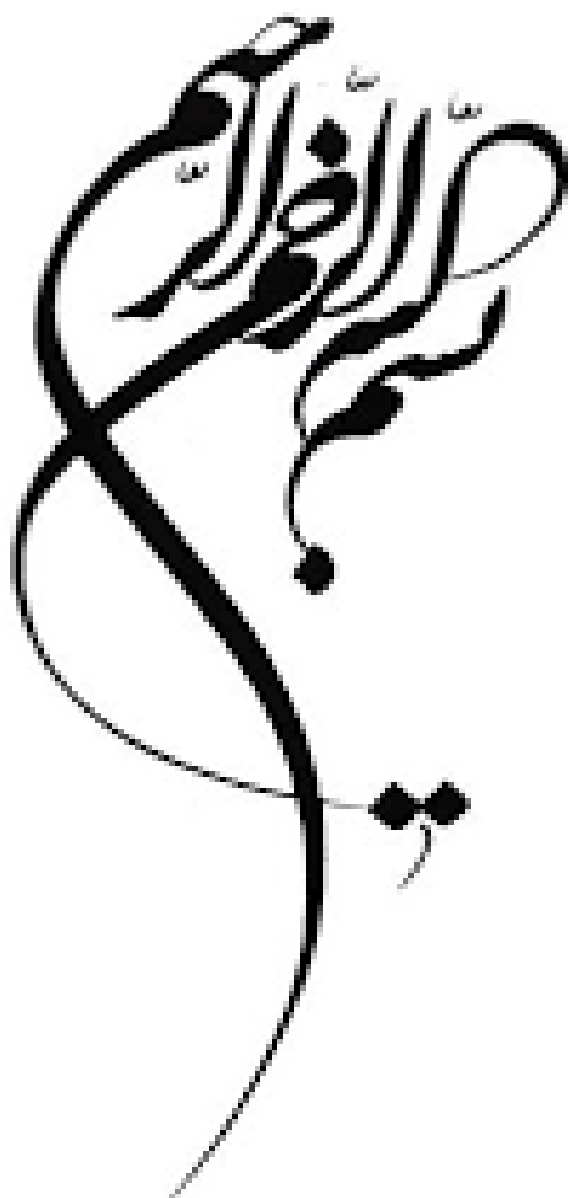
کیانا خسرویان

زهره کربلایی هادی

فاطمه ایرجی

فاطمه صادقی

پاییز ۱۳۹۹



مقدمه:

با گسترش شهر ها، تغییرات قابل توجهی در شکل سکونت افرادی که در شهر های بزرگ زندگی می کنند ایجاد شده است. تعداد طبقات ساختمان ها افزایش یافته است و با این افزایش چشمگیر، نیاز ها و در پی آن مشکلات نیز رو به زیاد شدن هستند. گسترش این نیاز ها از یک سو و همچنین افزایش جمعیت شهری از سویی دیگر، مشکلات جدیدی در حوزه زندگی آپارتمانی بوجود آمده است .

جوامع انسانی برای اینکه به مشکل بر نخورند به دو چیز نیاز دارند: قانون و مدیریت برای اجرای آن قانون. مجتمع های مسکونی نیز نمونه کوچکی از جوامع انسانی اند و به همین دلیل است که هر مجتمع باید برای خودش یک مدیر ساختمان یا چند مدیر تحت عنوان هیئت مدیره انتخاب کند. مدیر ساختمان باید از اصول اولیه مدیریت، قوانین اصلی بودجه بندی، تامین مالی، سازماندهی، برنامه ریزی، تناقض ها و مشکلات قانونی آگاهی کسب کند.

تکنولوژی های نرم افزاری به همراه فضای اینترنت راهکار های مختلف و متنوعی برای چنین مشکلاتی ارائه کرده اند. با گسترش روزافزون استفاده از اینترنت و تلفن های هوشمند، مدیریت ساختمان همانند دیگر عرصه های زندگی، به سمت استفاده بیشتر از تکنولوژی رفته است. امروزه با کمک نرم افزار های برنامه نویسی از جمله پایتون می توان برنامه ای نوشت تا به عنوان راه حلی مدرن برای جلوگیری از بروز مشکلات و اختلاف نظر ها در بین ساکنین یک ساختمان یا مجتمع مسکونی، به کمک مدیر یا مدیران ساختمان بیاید تا به سادگی و بدون نیاز به صرف زمان و یا انرژی زیاد، امور مجتمع ساختمانی خود را مدیریت کنند. این برنامه مدیریت ساختمان، امکان مدیریت شارژ ساختمان و بر طرف نمودن مشکلات بوجود آمده در ساختمان که امروزه همه ی افراد کم و بیش با آن ها در ارتباط هستند را فراهم می نماید. طراحی دقیق و مدرن این برنامه باعث ارتقا فرهنگ زندگی در آپارتمان و حفظ آرامش بین ساکنین و ایجاد هماهنگی در بر طرف نمودن نیاز هایی که برای مدیریت آپارتمان لازم می باشد از جمله تقسیم و جمع آوری هزینه های شارژ، قبوض و... می شود.

فهرست

۳	مقدمه:
۵	توضیح کد های نوشته شده
۵	• دریافت زمان
۵	• تراز مالی تراکنش
۵	• دسته بندی تراکنش
۶	• زیر دسته بندی
۶	• دریافت واحد مسئول
۶	• واحد مربوط
۷	• تابع نحوه تقسیم بندی هزینه div
۸	• جدول صورت حساب برای بازه ای معین
۸	• سهم بخش ها
۹	• روند هزینه ها
۱۱	• پیش بینی هزینه های پرداختی هر واحد در سال آینده (برآورد شارژ)
۱۲	نمونه خروجی های بصری
۱۵	گزارش تقسیم کار
۱۶	توصیف روند انجام کار
۱۶	چالش های تیم در روند اجرای پروژه
۱۶	بیان ایده های خالقانه و کارهای امتیازی
۱۶	نتیجه

توضیح کد های نوشته شده:

در این قسمت به ذکر و توضیح نکات مهم و قابل توجه در پروژه می پردازیم.

در قسمت اول، برنامه با اطلاعات وارد شده توسط کاربر یک دیکشنری و سپس یک dataframe می سازد و آن را به صورت CSV ذخیره میکند تا در بخش های بعدی برنامه از آن استفاده شود. همچنین لازم به ذکر است که کاربر در هنگام ورودی دادن به برنامه می تواند با وارد کردن عدد ۰ برای اصلاح خطاهای احتمالی به مرحله قبل برگردد.

• دریافت زمان (خط ۳۲-۳۰):

تابعی به نام time نوشته شده است که کاربر به دلخواه میتواند زمان را به صورت دستی وارد کند یا به طور خودکار آن را دریافت کند. در این بخش با توجه به توضیحات داده شده در کد اگر کاربر ورودی ۰ را به تابع بدهد می تواند تاریخ (سال، ماه، روز) را به طور دستی وارد کند و در غیر این صورت تاریخ شمسی به طور خودکار نمایش داده میشود.

• تراز مالی تراکنش (خط ۳۷-۳۳):

در ابتدا شماره ی واحد مورد نظر و بازه ی زمانی از کاربر گرفته میشود. سپس در تابع financial_balance تاریخ ها به ترتیب مرتب شده و با توجه به اطلاعات بین دو زمان مقدار مبلغ کل محاسبه شده و در خروجی نمایش داده میشود. با استفاده از این تابع کل بدهی های واحد ها به مدیر گرفته میشود و میتوانیم اعلام وضعیت بگیریم.

• دسته بندی تراکنش (خط ۳۹-۳۸):

تابع category_input کاربر را راهنمایی می کند تا دسته بندی را معرفی کند. کاربر میتواند عدد دسته بندی (۲۱) و یا اسم دسته بندی (۲۴) را انتخاب کند. همچنین گزینه بازگشت به عقب وجود دارد که "۰" را باز میگرداند تا در برنامه اصلی به ورودی قبلی یعنی مقدار هزینه باز گردد و کاربر آن را تغییر دهد. در مواجهه با ورودی اشتباه برنامه اطلاع می دهد و ورودی جدید از کاربر میخواهد. در آخر نام دسته بندی مورد نظر را برمیگرداند.

- زیر دسته بندی (۴۱-۴۰):

کارکرد این تابع (subcategory_input) مانند تابع category_input است با این تفاوت که می سنجد آیا دسته بندی وارد شده دارای زیر دسته بندی است یا نه (۴۳-۵۹). در صورت وجود نداشتن nan باز میگرداند و اگر زیر دسته بندی وجود داشت مانند دسته بندی ها و اعداد مربوط را گزارش میکند تا کاربر عدد یا اسم مربوط را وارد کند. گزینه بازگشت به عقب دارد و به ورودی اشتباه نیز رسیدگی می کند. در آخر نام زیر دسته بندی یا nan را بر میگرداند.

- دریافت واحد مسئول (۴۳-۴۲):

در این بخش، کلیدهای دیکشنری اطلاعات ساکنین به عنوان گزینه های قابل انتخاب برای کاربر در کنسول ارائه می شود. کاربر میتواند هر یک از گزینه های (دیفالت: مدیر ساختمان) یا شماره ردیف واحد مورد نظر را انتخاب کند. سپس این شماره واحد در متغیری با عنوان Responsible Apartment ذخیره می شود تا در ارائه گزارش و خروجی های برنامه استفاده شود. همانطور که پیش تر اشاره شد، تابعی تعریف شده تا شماره کلید دیکشنری اطلاعات را دریافت کند و بعنوان یک لیست ذخیره نماید. سپس در تابعی دیگر، هریک از اعضای این لیست + مدیرساختمان در یک ردیف و با یک شماره گزینه به کاربر نمایش داده می شوند تا واحد مسئول را انتخاب کند. در این بخش گزینه ای تعبیه شده که امکان بازگشت به عقب و اصلاح انتخاب را به کاربر می دهد.

- واحد مربوط (۴۵-۴۴):

ابتدا از کاربر خواسته میشود که انتخاب کند همه ی واحدها ، واحد مربوط هستند یا اگر واحد یا واحدهای دیگری واحد مربوط هستند، از کاربر گرفته شود و در خروجی نمایش داده شود ، همچنین اگر عدد ورودی کاربر از تعداد واحد های ساختمان بیشتر باشد ، کد خطا میدهد و دوباره از کاربر ورودی گرفته میشود.

^۱ دقت شود که کلید های دیکشنری باید شماره ی واحد و مقادیر هر کلید باید به صورت یک لیست تعریف شود که اعضای آن به ترتیب نام ساکن،تعداد افراد واحد،طبقه ی واحد،متراژ واحد و تعداد پارکینگ هاست. به طور مثال $d=\{1:['ahmadi',3,1,125,2],...\}$ یک دیکشنری است که فقط یک عضو آن نمایش داده شده و نشان میدهد واحد ۱ مربوط به خانواده سه نفره احمدی است که در طبقه اول سکونت دارند و متراژ خانه ۱۲۵ متر و ۲ پارکینگ دارند.

• تابع نحوه تقسیم بندی هزینه div (خط ۴۶ تا ۶۸):

ورودی این تابع دیکشنری اطلاعات ساکنین , یک عدد (مبلغی که باید بین واحد ها تقسیم شود) , یک عدد طبیعی بین ۱ تا ۷ که با توجه به توضیحات داده شده در کد و جدول ۱ هر کدام به یک نوع تقسیم بندی اشاره دارند و نام زیردسته بندی برای تقسیم بندی در حالت پیشفرض است که باید به صورت string وارد شود.

نوع تقسیم بندی در حالت پیشفرض در جدول ۲ مشخص شده است. یکی از روش های تقسیم بندی ورود دستی سهم هر واحد به صورت رشته ای از اعداد متوالی (جدا شده با یک فاصله) می باشد. در واقع کاربر میتواند سهم هر واحد را به صورت رشته ای از اعداد وارد کند (با انتخاب عدد ۷). لازم به ذکر است اعداد بر حسب درصد وارد می شود و اگر جمع مقادیر وارد شده مساوی صد نباشد، خطایی با توضیح مربوط نمایش داده خواهد شد. سپس این درصدها در دیکشنری جدیدی بعنوان مقدار به هریک از واحدها (کلیدها) اطلاق خواهد شد تا در محاسبات بعدی استفاده شود.

عدد ۱	تقسیم بندی بر اساس تعداد افراد هر واحد
عدد ۲	تقسیم بندی بر اساس شماره طبقه هر واحد
عدد ۳	تقسیم بندی بر اساس متراژ هر واحد
عدد ۴	تقسیم بندی بر اساس تعداد پارکینگ هر واحد
عدد ۵	تقسیم بندی به طور مساوی
عدد ۶	تقسیم بندی بر اساس پیشفرض
عدد ۷	تقسیم بندی بر اساس درصد های وارد شده توسط کاربر

جدول ۱- نحوه ورودی دادن به تابع div

نام زیر دسته بندی	نحوه ورودی دادن به تابع	نحوه تقسیم بندی
قبض برق	'BillsElectricity'	تقسیم بندی به طور مساوی
قبض گاز	'BillsGas'	تقسیم بندی به طور مساوی
عوارض شهرداری	'BillsTax'	تقسیم بندی به طور مساوی
قبض آب	'BillsWater'	تقسیم بندی بر اساس تعداد نفرات
شارژ	'Chargenan'	تقسیم بندی به طور مساوی
نظافت	'Cleaningnan'	تقسیم بندی بر اساس تعداد پارکینگ هر واحد
آسانسور	'Elevatornan'	تقسیم بندی بر اساس شماره طبقه هر واحد
تعمیرات	'Repairmentnan'	تقسیم بندی به طور مساوی

جدول ۲_ نحوه تقسیم بندی به طور پیشفرض

- **جدول صورت حساب برای بازه ای معین:**

کاربر دو تاریخ با فرم مطلوب بعنوان ابتدا و انتهای بازه وارد می کند. سپس با ابزارهای کتابخانه پانداس، همه تراکنش های موجود بین این دو بازه از دیتا فریم مربوط استخراج شده و در قالب یک فایل CSV ارائه می شود.

- **سهم بخش ها :**

در این بخش هدف از نوشتن تابع اول محاسبه هزینه های زیرگروه های یک دسته بندی در آن دسته بندی است . با توجه به اینکه در اطلاعات داده شده تنها دسته "قبض" دارای زیر دسته هایی شامل (گاز/آب/برق/عوارض) بود تابعی نوشته شد تا یک ورودی (dataframe) تشکیل شده توسط برنامه با اطلاعات وارد شده توسط کاربر (دریافت کند.

در این تابع نسبت کل مبلغی که برای یک قبض (آب / برق/گاز یا عوارض) پرداخت شده به کل مبلغی که برای قبوض پرداخت شده است محاسبه میشود. خروجی نیز یک دیکشنری است که کلید های آن اسم زیر دسته بندی و مقادیر آن نسبت مبلغ پرداخت شده برای هر زیردسته بندی به مبلغ پرداخت شده برای کل قبوض است. همچنین برای درک بهتر مدیریت نمودار میله ای رسم میشود تا بتوان به مقایسه مبالغ هزینه شده در این دسته پرداخت . در بخش "نمونه های خروجی بصری" بیشتر به نمودار ها خواهیم پرداخت.

در تابع دوم نوشته شده در این بخش سهم هزینه های هر دسته بندی نسبت به کل محاسبه میشود. نحوه ورودی دادن به این تابع مانند تابع قبلی است و تنها کافی است آدرس فایل مربوطه را به برنامه بدهید.

در این تابع ابتدا یک لیست به نام "a" شامل اسامی تمامی دسته ها و یک لیست به نام "b" شامل نسبت مبلغ هر کدام از دسته ها به کل مبلغ (به همان ترتیب اعضای لیست a) است . که خروجی تابع را میتوانید به شکل یک دیکشنری ببینید. علاوه بر این باتوجه به اطلاعات ورودی نمودار دایره ای رسم میشود که در بخش "نمونه های خروجی بصری" بیشتر به آن خواهیم پرداخت.

• ورودی روند هزینه ها (خط ۹۳-۱۷):

تابع `expsenseflow_input` کاربر را راهنمایی میکند تا ورودی ها را برای فیلتر هایی که روی نمودار اعمال می شوند وارد کند. بر اساس طرح پروژه صلاح بر آن دیده شد که انواع روند هزینه ها در یک پروسه قابل اجرا باشند. پیشبینی میشود این امر بر احاطه راحت تر کاربر بر برنامه کمک میکند. برنامه به حروف بزرگ و کوچک حساس است. این تابع به دو صورت قابل استفاده می باشد.

۱. معرفی کردن فیلتر ها با روش تعاملی (۶۱-۵۸):

در این روش کاربر حیط ای که در آن فیلتر اعمال می شود را انتخاب میکند و سپس بر اساس مثال و توضیحات ارائه شده محدودیت های فیلتر را معرفی می کند. کاربر میتواند هر مقدار که میخواهد انواع فیلتر ها را اعمال کند و یا تغییر دهد. تا آنجا که در توان بوده سعی شده ورودی های غیر مرتبط و اشتباه کاربر رسیدگی شوند و باعث بسته شدن و اشکال در برنامه نشود.

```

C:\ProgramData\Anaconda3\python.exe -i "C:\Users\kiana\Documents\project1\cumulativeacco...
please enter the number of Filtering mode or type declared filtering structure:
1. Based on categories
2. based on building number
3. based on dates
1
please enter desired categories in one line, separating them with space:
(eg. Elevator Bills:Water*Gas)
Bills:Gas Parking

You have chosen the following filters:
categories of accounts that will be reported: ['Bills:Gas', 'Parking']
building units whom the accounts refer to: Allunits
time period which the accounts have taken place on: Alltimes
if you want to add or change a filter please enter 0. otherwise enter any character to continue:
0
please enter the number of Filtering mode or type declared filtering structure:
1. Based on categories
2. based on building number
3. based on dates
3
please enter desired time period with '-'. separate the dates with space:
(eg. 1399-04-05 1399-05-05)
1399-03-04 1400-04-04

You have chosen the following filters:
categories of accounts that will be reported: ['Bills:Gas', 'Parking']
building units whom the accounts refer to: Allunits
time period which the accounts have taken place on: ['1399-03-04', '1400-04-04']
if you want to add or change a filter please enter 0. otherwise enter any character to continue:
-

C:\ProgramData\Anaconda3\python.exe -i "C:\Users\kiana\Documents\project1\cumulativeacco...
0
please enter the number of Filtering mode or type declared filtering structure:
1. Based on categories
2. based on building number
3. based on dates
2,3
the input does not match the defined structure/value. please try again.

You have chosen the following filters:
categories of accounts that will be reported: ['Bills:Gas', 'Parking']
building units whom the accounts refer to: Allunits
time period which the accounts have taken place on: ['1399-03-04', '1400-04-04']
if you want to add or change a filter please enter 0. otherwise enter any character to continue:
0
please enter the number of Filtering mode or type declared filtering structure:
1. Based on categories
2. based on building number
3. based on dates
no
the input does not match the defined structure/value. please try again.

You have chosen the following filters:
categories of accounts that will be reported: ['Bills:Gas', 'Parking']
building units whom the accounts refer to: Allunits
time period which the accounts have taken place on: ['1399-03-04', '1400-04-04']
if you want to add or change a filter please enter 0. otherwise enter any character to continue:
0
please enter the number of Filtering mode or type declared filtering structure:
1. Based on categories
2. based on building number
3. based on dates
the input does not match the defined structure/value. please try again.

```

بعد از آن که محدودیت ها معرفی شدند برنامه از تمام فیلتر های معرفی شده گزارش میدهد و کاربر بسته به نیاز خود تایید میکند یا نمی کند. توجه داشته باشید که ورودی اشتباه تاثیری در داده های درست گرفته شده ندارد و برنامه اجازه وارد شدن آنها به فیلتر های تایید شده را نمی دهد.

۲. معرفی کردن فیلتر ها با روش رشته دستور (۵۷-۴۹):

در این روش کاربر رشته فیلتر ها را در یک خط به برنامه اعلام می کند. ترتیب آن به این صورت است:

filterby categories units firstDate,secondDate skip(optional) indexby(optional)

command	Eg.	Contain all
filterby	filterby	
Categories جدا کردن دسته ها با ", " Subcategories تعریف آن با نوشتن نام دسته و : و گذاشتن* قبل از نام زیردسته	Elevator,Bills:*Water*Gas,Repairment	Allcategories
Units جدا کردن طبقات با ", "	۱,۲,۳	Allunits
Dates جدا کردن دو زمان با ", " و مشخص کردن تاریخ با "-"	۱۳۹۹-۰۸-۰۹,۱۴۰۰-۰۱-۰۱	Alltimes
skip اختیاری	skip	
Indexby انتخاب یکی از موارد زیر RelatedUnit Category All	RelatedUnit	

دستور skip برای رد کردن قسمت گرفتن تاییدیه فیلتر ها از کاربر است. اگر این دستور اجرا شود برنامه دیگر ورودی از کاربر نمی خواهد و گزارش آخرین وضعیت فیلتر ها را نمی دهد. توجه داشته باشید با ورود این دستور دستور بعدی index نیز باید وارد شود .

تابع Cumulative_filter :

دیتا فریم را بر اساس فیلتر ها فیلتر می کند. برای فیلتر بر اساس زمان رشته زمان ها را میگیرد "-" را حذف میکند و اعداد باقی مانده رشته را با هم مقایسه میکند. برای فیلتر دسته بندی وجود دسته بندی و زیر دسته بندی در لیست مورد نظر را چک می کند ولی ارتباط آن دو را بررسی نمی کند.

در کشیدن نمودار تابع هزینه های تجمعی هر واحد/دسته بندی را حساب میکند (۱۸۰-۱۹۵) و در نمودار قرار می دهد (۱۹۷-۲۱۱)

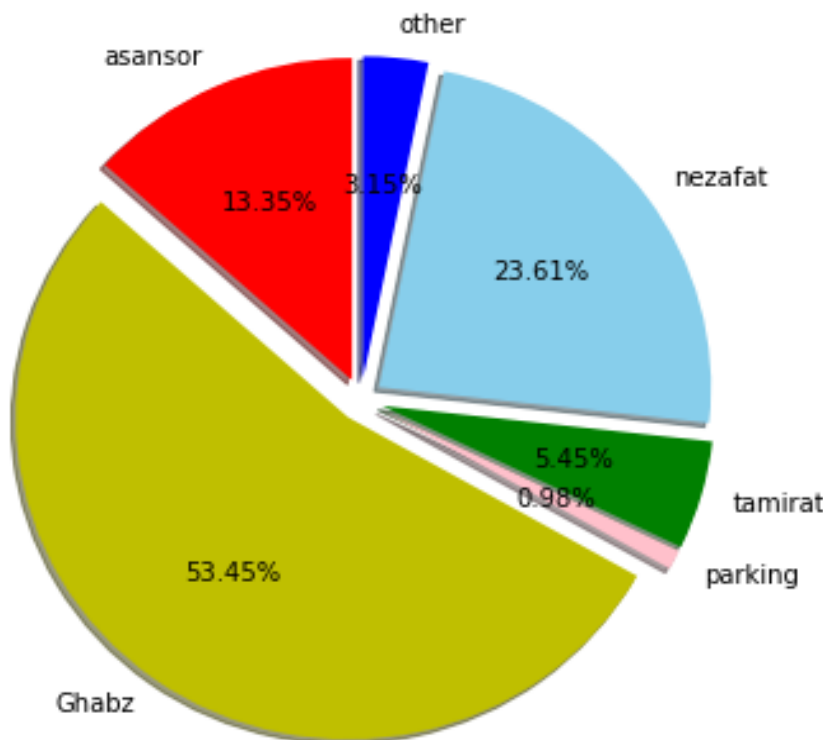
در بخش "نمونه های خروجی بصری" بیشتر به نمودار ها خواهیم پرداخت.

• پیش بینی هزینه های پرداختی هر واحد در سال آینده (برآورد شارژ):

با گرفتن شماره واحد ساختمان ، تابع charge_estimate حداکثری از تاریخ ها را بدست آورده و با بررسی یک سال گذشته و جمع مبالغ آن و تقسیم بر عدد دوازده ، یک مقدار تقریبی بدست می آید که با احتساب نرخ تورم حدودی ۲۰ درصد در پایان مبلغ را در ۱,۲ ضرب میکنیم.

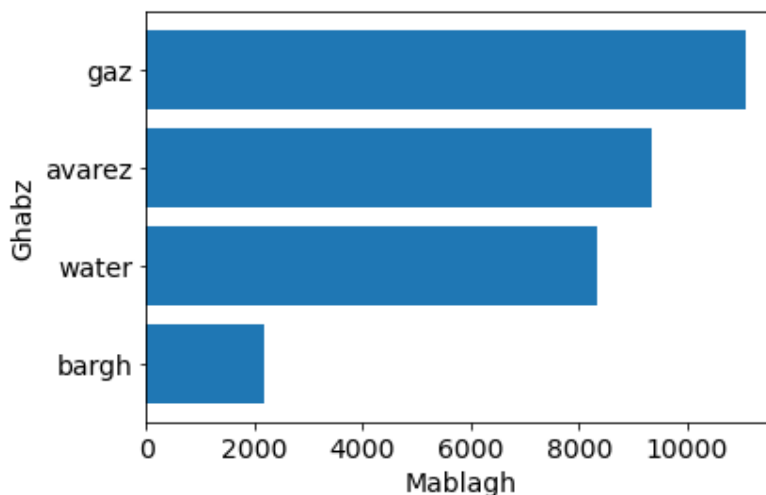
نمونه خروجی های بصری:

در قسمتی از این پروژه به بررسی سهم داده ها (سهم هزینه های زیرگروه های یک دسته بندی در آن دسته بندی و سهم هزینه های دسته بندی نسبت به کل) پرداخته شده است. علاوه بر این استفاده از نمودار ها سبب میشود تا مدیریت از میزان هزینه های ساختمان در بازه زمانی مشخص درک بهتری پیدا کند و نسبت به رفع هزینه های اضافی اقدام نماید .



در نمودار ۱ میتوانید میزان سهم هزینه های یک ساختمان فرضی نسبت به کل هزینه ها را در یک نمودار دایره ای ببینید . این نمودار نشان میدهد که حدود نصف هزینه های این ساختمان مربوط به پرداخت قبوض است و پس از آن هزینه های نظافت و آسانسور در صدر قرار دارند .

نمودار ۱_ سهم هزینه های هر دسته بندی نسبت به کل



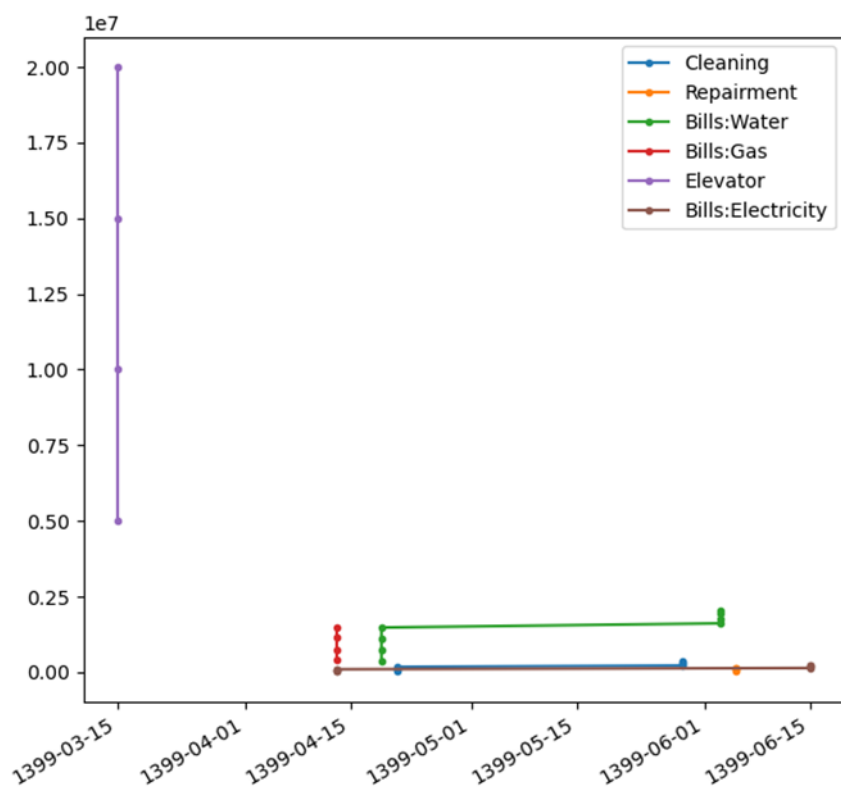
در نمودار ۲ سهم هزینه های زیر گروه های یک دسته بندی (قبض) در آن دسته بندی مربوط به یک ساختمان فرضی نشان داده شده است. که به مدیریت کمک میکند تا درک بهتری از میزان مصرف (گاز/آب/ برق/عوارض) داشته باشد.

نمودار ۲_ سهم هزینه های زیر دسته بندی در آن دسته بندی

در قسمت دیگری از این پروژه به بررسی روند هزینه ها پرداخته شده است که این امکان را به مدیریت میدهد تا روند هزینه ها در بازه زمانی دلخواهی بررسی و یا مقایسه کند. همچنین کاربر میتواند فیلترهایی روی این نمودار اعمال کند. به طور مثال هزینه های مربوط به چند زیرگروه خاص یا چند دسته بندی خاص یا کل هزینه ها را برای چند واحد خاص در یک بازه خاص ببیند.

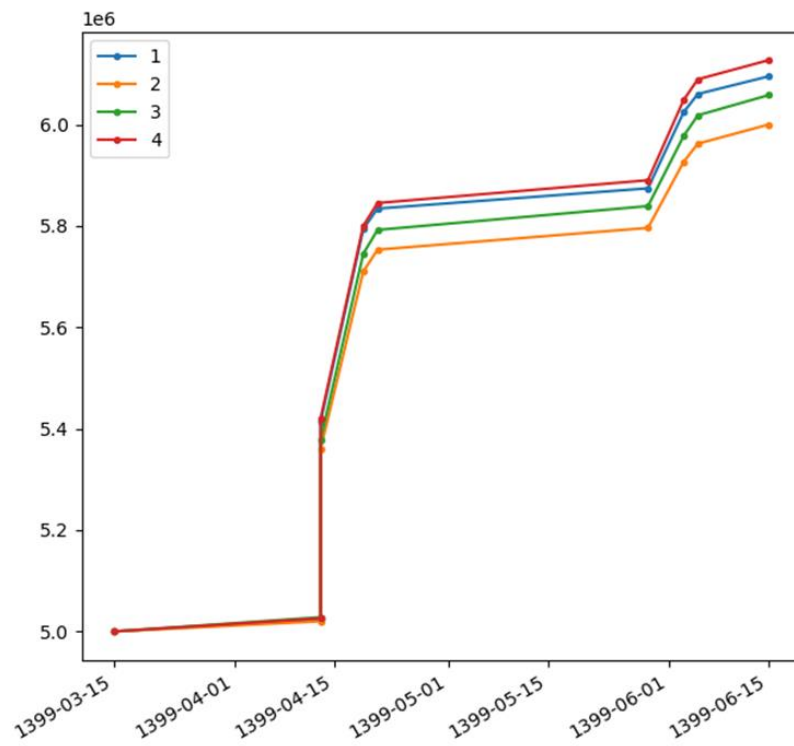
نمودارهای زیر هزینه های یک آپارتمان ۴ طبقه را به طرق مختلف نشان میدهد.

نمودار بر اساس دسته بندی:



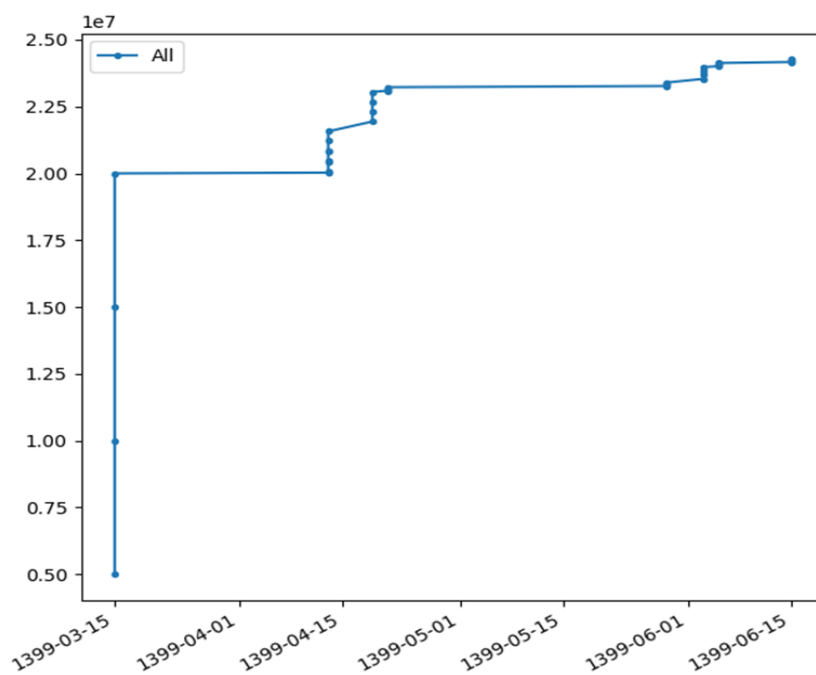
نمودار ۳_نمودار بر اساس دسته بندی

نمودار بر اساس شماره ساختمان:



نمودار ۴_نمودار بر اساس شماره ساختمان

نمودار همه ی هزینه ها:



نمودار ۵_نمودار همه ی هزینه ها

گزارش تقسیم کار:

فاطمه ایرجی:

- ✓ دریافت واحد مسئول (مدیر ساختمان / دریافت شماره واحد)
- ✓ جدول صورت حساب برای بازه ای معین (CSV)
- ✓ دریافت سهم هر واحد به صورت رشته ای از اعداد
- ✓ دریافت بازه وضعیت بدهی
- ✓ توضیح کد های نوشته شده ی خود در گزارش انجام پروژه

کیانا خسرویان:

- ✓ تابع دریافت دسته بندی و زیردسته بندی (ارائه لیست)
- ✓ توضیحات شرح مختصری از تراکنش
- ✓ روند هزینه ها (نمودار تجمعی)
- ✓ پیاده سازی پروژه روی Json
- ✓ توضیح کد های نوشته شده ی خود در گزارش انجام پروژه

فاطمه صادقی:

- ✓ تراز مالی تراکنش
- ✓ پیش بینی هزینه ی پرداختی هر واحد در سال آینده (برآورد شارژ)
- ✓ واحد مربوط
- ✓ توضیح کد های نوشته شده ی خود در گزارش انجام پروژه

زهره کربلایی هادی:

- ✓ تابع دریافت زمان (تعریف ساختار برای ورود دستی و خودکار)
- ✓ تابع گرفتن نام ساکنان واحد های مربوط از دیکشنری اطلاعات ساختمان
- ✓ تابع نحوه تقسیم بندی
- ✓ توابع مربوط به سهم بخش ها
- ✓ توضیح کد های نوشته شده ی خود در گزارش انجام پروژه
- ✓ نوشتن گزارش انجام پروژه

توصیف روند انجام کار:

در ابتدا تقسیم کار به صورت ذکر شده صورت گرفت و هردانشجو به طور جداگانه کد های خود را می نوشت و اگر در حین انجام کار با مشکلی مواجه میشد با کمک دیگر اعضای گروه به حل مشکل میپرداخت و در نهایت کدهای خود را در مخزن GitHub منتشر میکرد.

در پایان هر هفته جلساتی در Google Meet با حضور تمام اعضای گروه برگزار می شد تا اعضای گروه در جریان تمام روند پروژه قرار بگیرند و به حل مشکلات و چالش های یکدیگر در حین انجام پروژه می پرداختند.

چالش های تیم در روند اجرای پروژه:

✚ نامفهوم بودن خواسته های پروژه

بیان ایده های خالقانه و کارهای امتیازی:

- (۱) تحویل کد بر روی مخزن GitHub
- (۲) استفاده از ماژول Khayyam برای بیان تاریخ به صورت شمسی در تابع time
- (۳) تخصیص گزینه "بازگشت" برای مراحل دریافت ورودی از کاربر (جهت اصلاح خطاهای احتمالی)
- (۴) پیاده سازی پروژه روی Json

نتیجه:

بدون شک امروزه، زندگی آپارتمان نشینی چالش های خاص خود را به همراه دارد. مدیریت ساختمان از مهم ترین چالش هایی است که به طور قطع به یکی از دغدغه های اصلی جوامع در این چند سال تبدیل شده است. در اکثر مواقع منشا این مشکلات بر گرفته از بی نظمی در اداره امور ساختمان است. در این جور مواقع بهترین راه حل استفاده از نرم افزار مدیریت ساختمان است. این برنامه مدیریت ساختمان به منظور سهولت و شفافیت در مدیریت ساختمان ها و مجتمع های مسکونی و تجاری پیاده سازی شده است. با استفاده از این برنامه شما می توانید روش های سنتی مدیریت را پشت سر بگذارید و به طور خودکار شارژ و یا سایر هزینه های ساختمان را بر اساس مترáž، تعداد افراد، تعداد پارکینگ و غیره تعیین کنید.