

## بسمه تعالی

این برنامه به کاربر کمک می کند که راحت تر و سریع تر محاسبات مالی را انجام داده و یک نمایش کلی از تراکنش های مالی صورت گرفته به ما نشان می دهد.

ما در این گزارش به نحوه انجام ، تقسیم پروژه ، اینکه هر فرد از گروه کدام قسمت پروژه را انجام داده و نحوه کارکرد برنامه اشاره می کنیم.

ما تیم خود را برای هماهنگی راحت تر به دو گروه تقسیم کردیم .

گروه یک : محمدحسین یزدانی فر و یارسا پهلوانی

این گروه به بررسی و سامان دهی داده های ورودی پرداخت. کار این گروه نوشتن قسمتی از برنامه بود که ورودی ها را از کاربر گرفته و آن ها را قالب CSV ذخیره می کند.

گروه دو: محمدحسین حسین زاده فرجام و امیرحسین رحمتی

این گروه به تحلیل داده هایی که توسط گروه یک به صورت CSV ذخیره شده است می پردازد و نمایش کلی، از این داده به کاربر نشان می دهد.

### ❖ بخش اول

در پخش اول پروژه که مربوط به تیم یک است، به علت تعدد ورودی ها تصمیم گرفته شد که یک ساختار یا frame work برای ورودی طراحی کنیم.

ساختار بدین شکل است :

#### ○ **پیکشوری نام سوال ها (question\_list) :**

نوشتن عنوان هر ورودی و در صورتی که ورودی شامل تعداد محدودی جواب بود می توان برای جواب های آن ورودی یک لیست برای انتخاب قرار داد.

#### ○ **پیکشوری سوال بعدی (question\_next) :**

به ازای هر سوال انتخاب می کنیم که بعد از آن چه سوالی پرسیده شود و برای سوالاتی که شامل یک لیست برای انتخاب هستند می توان برای هر کدام از آن جواب ها یک سوال بعدی انتخاب کرد.

#### ○ **پیکشوری تابع سوالات (question\_function) :**

می توان برای هر سوال دستوراتی قرار داد که بعد از گرفتن جواب آن سوال، آن دستورات اجرا شود. دستورات هر سوال به صورت یک رشته ذخیره می شود که در ادامه با استفاده از تابع `exec1` آن ها را اجرا می کنیم.

---

`exec1` : تابعی مشابه `eval` می باشد اما محدودیت کمتری نسبت به آن دارد و می توان دستورات پایتون را با آن اجرا کرد.

#### ○ متغیر سوال اول (q) :

مشخص می کند که اولین سوال کدام باشد.

#### ○ پردازش :

پردازش تکه کدی است که با استفاده از سوالات تعریف شده آن ها را به کاربر نمایش می دهد و ورودی های کاربر را پردازش می کند.

#### ❖ بخش دوم :

در بخش دوم که مربوط به تیم دو است، تابع های قسمت گزارش گیری، یک به یک طراحی شده اند. که این تابع ها شامل :

#### ○ تابع تراز مالی Taraz Mali :

داخل این تابع از کتابخانه pandas و متد های Groupby و Aggregate استفاده شده است.

#### ○ تابع صورت حساب Soorat Hesab :

در این تابع از کتابخانه pandas استفاده شده است و داده بر اساس زمان مرتب شده و داده های قرار گرفته در بازه ی انتخابی کاربر به عنوان خروجی نمایش داده می شود.

#### ○ دو تابع سهم بخش ها Sahm Bakhshha :

داخل این تابع از کتابخانه pandas و متد های Groupby و Aggregate استفاده شده است. تفاوت دو تابع مذکور این است که یکی از تابع ها مبنای فیلترینگ را Category و دیگری Sub Category می گذارد.

#### ○ تابع موجودی Mojodi :

در این تابع از کتابخانه pandas و تابع sum استفاده شده است.

#### ○ تابع نمودار تجمعی Comsum Plot :

در این تابع از دو کتابخانه pandas و matplotlib.pyplot استفاده می شود. تابع با توجه به متغیر های ورودی داده های اولیه را فیلترینگ می کند. سپس با متد های Groupby و Aggregate مقادیر تجمعی آن ها را بدست می آورد و در مرحله بعد نمودار مربوطه را رسم می کند.

#### ○ تابع نمودار تجمعی 2 Comsum Plot 2 :

در این تابع از دو کتابخانه pandas و matplotlib.pyplot استفاده می شود. تابع با توجه به متغیر های ورودی داده های اولیه را فیلترینگ می کند. سپس با متد های Groupby و Aggregate مقادیر تجمعی آن ها را بدست می آورد و در مرحله بعد با پیمایش روی متغیر ورودی Category چند نمودار را رسم می نماید.

### ○ تابع پیش بینی Forecast :

این تابع از دو کتابخانه pandas و date time استفاده می کند و با بدست آوردن بازه ی بین تاریخ اولین و آخرین تراکنش و بدست آوردن جمع مقادیر هزینه ها و با در اختیار داشتن نرخ تورم در چند سال اخیر مقدار شارژ پیش بینی شده را نمایش می دهد.

کار چالشی متصل کردن تمامی تابع ها و ساختن دستورا های متناسب با تابع ها در کد اصلی هم توسط هر دو تیم صورت گرفت.

Enter Admin Id:

- 1 - Append
- 2 - Mojoodi
- 3 - Taraz Mali
- 4 - Soorat Hesab
- 5 - Sahm Bakhshha
- 6 - Cumsum Plot
- 7 - Forecast
- 8 - Exit

Enter Query:

نحوه کارکرد برنامه و نمونه خروجی بصری برای داده های تحویل داده شده توسط تیم دآوری :

- در ابتدا برنامه ثبت کننده تراکنش را تحت عنوان Admin ID<sup>2</sup> ثبت می کن.
- پس از گرفتن ثبت کننده برنامه چند دستور پیشنهادی را به کاربر ارائه می دهد، که کاربر با انتخاب عدد مربوطه به آن دستور دسترسی پیدا می کند.

### دستور اول : Append

- دستور Append جهت اضافه کردن هر یک از تراکنش های مالی از جمله تراکنش های مثبت ( شارژ ) و تراکنش های منفی ( هزینه ها ) می باشد
- پس از انتخاب دستور append برنامه زمان انجام تراکنش مذکور را از کاربر دریافت می کند که برنامه می تواند ورودی را به دو صورت از کاربر دریافت کند.

#### 1. صورت اول :

کاربر زمان انجام تراکنش مربوطه را به صورت دستی، به عنوان مثال : **1399-11-21**<sup>3</sup> وارد کند.

#### 2. صورت دوم :

کاربر با وارد کردن عبارت **now** در برنامه تاریخ دقیق روز را به عنوان تاریخ تراکنش ثبت کند.

Enter Time: now

- 
- 1 - Ghabz
  - 2 - Asansor
  - 3 - Nezafat
  - 4 - Parking
  - 5 - Tamirat
  - 6 - Other
  - 7 - Sharj

Enter Category:

---

<sup>2</sup> Admin ID : می تواند هر کدام از ساکنین ساختمان یا هر کاربر دیگه ای باشد.  
<sup>3</sup> تمام تاریخ های ثبت شده در برنامه به شکل تاریخ جلالی است.

▪ پس از وارد کردن تاریخ تراکنش زمان انتخاب نوع تراکنش است.

### 1. تراکنش های منفی:

تراکنش های منفی شامل تراکنش هایی است که مربوط به هزینه های ساختمان می باشد، که شامل:

- 1 - Water
- 2 - Gaz
- 3 - Bargh
- 4 - Avarez

Enter Sub Category:

- i. قبض : که تراکنش های تحت عنوان قبض به چهار زیردسته ی آب، گاز، برق و عوارض تقسیم می شود.
- ii. آسانسور
- iii. نظافت
- iv. پارکینگ
- v. تعمیرات
- vi. دیگر هزینه ها

پس از انتخاب نوع تراکنش (در صورت نیاز، انتخاب زیردسته تراکنش) زمان وارد کردن هزینه ی مربوطه تحت عنوان amount می باشد

Enter Amount: 500

Enter Related Unit: 2

id4

id5

پس از ثبت هزینه ی تراکنش زمان وارد کردن تعداد واحد های مربوط به تراکنش تحت عنوان related unit می باشد. برنامه با توجه به عدد ورودی در مرحله ی قبل به تعداد واحد های مربوطه در هر خط یک واحد مربوطه را از کاربر دریافت می کند. (پس از نوشتن هر واحد برای ثبت شدن ورودی در برنامه از کلید Enter استفاده کنید و سپس ورودی بعدی را وارد کنید.

در مرحله بعد زمان انتخاب نحوه تقسیم بندی هزینه ی تراکنش بین واحد ها است که به چهار روش به دلخواه کاربر می تواند انجام شود.

- 1 - Equal
- 2 - Number Of Residents
- 3 - Size Of Area
- 4 - Number Of Parkings

Enter Div: 2

Enter Description: ok

1. Equal : برنامه هزینه را به طور کاملاً مساوی بین واحد ها تقسیم می کند.
2. Number of residents : برنامه هزینه مربوطه را با توجه به تعداد ساکنین هر واحد تقسیم می کند.
3. Size of area : برنامه هزینه را با توجه به مساحت واحد مربوطه تقسیم می کند.
4. Number of parkings : برنامه هزینه را با توجه به تعداد پارکینگ های هر واحد تقسیم می کند.

کاربر میتواند توضیحات مربوطه به تراکنش را تحت عنوان description در برنامه وارد کند. فرایند ثبت یک تراکنش به صورت کامل در این مرحله به پایان می رسد و برنامه به صورت خودکار در صفحه ی اولیه انتخاب دستور های قرار می گیرد.

## 2. تراکنش های مثبت :

تنها تراکنش مثبت تعریف شده برای برنامه تحت عنوان charge در انتهای لیست تراکنش های قبلی قرار گرفته است.

با انتخاب عنوان charge مبلغ تراکنش باید توسط کاربر تحت عنوان amount of charge در برنامه ثبت شود. سپس برنامه واحد پرداخت کننده charge را تحت عنوان unit دریافت می کند.

در انتها مانند تراکنش های دیگر برای این نوع تراکنش نیز کاربر می تواند توضیحات مربوطه را تحت عنوان description در برنامه وارد کند و پس از ثبت کامل تراکنش، به صفحه اولیه بازمی گردد

```
1 - Append
2 - Mojoodi
3 - Taraz Mali
4 - Soorat Hesab
5 - Sahm Bakhshha
6 - Cumsum Plot
7 - Forecast
8 - Exit
```

Enter Query: 2

→ -59080

### دستور دوم : Mojodi

- با انتخاب عدد مربوط به دستور موجودی برنامه با در نظر گرفتن تمام تراکنش های مثبت و منفی، موجودی صندوق ساختمان را ارائه می دهد. ( این عدد می تواند منفی باشد که به معنای آن است که واحد ها به دلیل تراکنش های منفی صورت گرفته به ساختمان بدهکار می باشند)

```
1 - Append
2 - Mojoodi
3 - Taraz Mali
4 - Soorat Hesab
5 - Sahm Bakhshha
6 - Cumsum Plot
7 - Forecast
8 - Exit
```

Enter Query: 3

-----

Enter Id For Taraz Mali: id5

→ -8731.1818181818

### دستور سوم : تراز مالی Taraz Mali

- دستور Taraz Mali شباهت زیادی به دستور Mojodi دارد با این تفاوت که مقدار بستانکاری هر واحد را به صورت جدا نمایش می دهد که اگر عدد نشان داده شده مثبت باشد، واحد مربوطه به اندازه ی مقدار نشان داده شده بیشتر از هزینه های واحد تحت عنوان شارژ به ساختمان پرداخت کرده است.
- پس از انتخاب عدد مربوط به دستور Taraz Mali ، کاربر باید واحدی که خواستار تراز مالی آن است را تحت عنوان Id For Taraz Mali در برنامه وارد کند. در مرحله بعد برنامه تراز مالی واحد مربوطه را به کاربر ارائه می دهد.

#### دستور چهارم : صورت حساب Soorat Hesab

- دستور صورت حساب تمامی راکنش های انجام شده بین دو تاریخ مشخص را در قالب یک فایل csv تحت عنوان Soorat Hesab به کاربر ارائه می دهد.
- کاربر در دو بخش باید دو ورودی Start Time و End Time را در برنامه وارد کند.

```
1 - Append
2 - Mojoodi
3 - Taraz Mali
4 - Soorat Hesab
5 - Sahm Bakhshha
6 - Cumsum Plot
7 - Forecast
8 - Exit

Enter Query: 4

-----

Enter Start Time: 1397-01-01

-----

Enter End Time: 1399-01-01
```

#### دستور پنجم : سهم بخش ها Sahm Bakhshha

- این دستور شامل دو بخش می شود که کاربر پس از انتخاب این دستور باید بخش مربوطه را مشخص کند.

```
1 - Append
2 - Mojoodi
3 - Taraz Mali
4 - Soorat Hesab
5 - Sahm Bakhshha
6 - Cumsum Plot
7 - Forecast
8 - Exit
```

Enter Query: 5

1. **By Category** : در این بخش برنامه سهم هر یک از دسته ها را در قالب یک عدد بین 0 و 1 به کاربر نمایش می دهد.

```
1 - By Category
2 - By Sub Category
```

Enter Sahm Bakhshha: 1

2. **By Sub Category** : در این بخش برنامه سهم هر یک از زیردسته های تراکنش قبض را در قالب یک عدد بین 0 و 1 به کاربر نمایش می دهد.

	category	part
0	Ghabz	0.537735
1	asansor	0.129658
2	nezafat	0.239460
3	other	0.030647
4	parking	0.009534
5	tamirat	0.052966

## دستور ششم : نمودار مقادیر تجمعی Cumsum Plot

- در این بخش ارزش داده ها به صورت تجمعی سنجیده می شود، بدین صورت که هزینه های انجام شده در هر تاریخ جمع هزینه های تراکنش های آن روز و روز های قبل است. نمودار های تجمعی ارائه شده توسط برنامه در دو قالب Normal و Comprable قرار دارند.

### 1. Normal :

در این نوع رسم نمودار برنامه داده های تجمعی واحد ها و دسته های خاص گرفته شده از کاربر را در قالب یک نمودار خطی به کاربر نمایش می دهد.

پس از انتخاب دستور Normal کاربر باید دو تاریخ را تحت عنوان های Start Time و End Time به برنامه ارائه دهد، که نمودار مربوطه بین بازه ی گرفته شده رسم خواهد شد.

در قدم بعدی کاربر باید تعداد واحد های مد نظر برای نمودار را تحت عنوان Related Ids در برنامه وارد کند سپس با توجه به عدد وارد شده در هر یک از خطوط بعدی واحد های مد نظر را در برنامه وارد کند.

در مرحله بعد کاربر باید تعداد دسته های مد نظر برای نمودار را تحت عنوان Related Category در برنامه وارد کند سپس با توجه به عدد وارد شده در هر یک از خطوط بعدی دسته های مد نظر را در برنامه وارد کند.

### 2. Comprable :

در این نوع رسم نمودار، برنامه داده های تجمعی دسته های خاص گرفته شده از کاربر را در قالب چند نمودار خطی با یکدیگر مقایسه کرده و به نمایش می گذارد.

پس از انتخاب دستور Comprable کاربر باید دو تاریخ را تحت عنوان های Start Time و End Time به برنامه ارائه دهد، که نمودار مربوطه بین بازه ی گرفته شده رسم خواهد شد.

در مرحله بعد کاربر باید تعداد دسته های مد نظر برای نمودار را تحت عنوان Related Category در برنامه وارد کند سپس با توجه به عدد وارد شده در هر یک از خطوط بعدی دسته های مد نظر را در برنامه وارد کند.

```
1 - Append
2 - Mojoodi
3 - Taraz Mali
4 - Soorat Hesab
5 - Sahm Bakhshha
6 - Cumsum Plot
7 - Forecast
8 - Exit
```

```
Enter Query: 6
```

```
-----
1 - Normal
2 - Comparable
```

```
Enter Cumsum Plot: 2
```

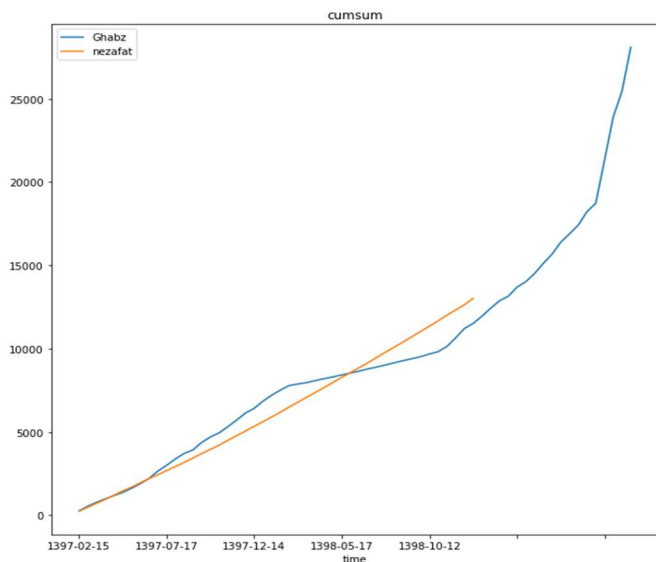
```
-----
Enter Start Time: 1397-01-01
```

```
-----
Enter End Time: 1399-01-01
```

```
-----
Enter Related Category: 2
```

```
Ghabz
```

```
nezafat
```



نمودار مقابل با توجه به داده های ارائه شده توسط تیم داوری می باشد. ( خروجی بصری )

- 1 - Append
- 2 - Mojoodi
- 3 - Taraz Mali
- 4 - Soorat Hesab
- 5 - Sahm Bakhshha
- 6 - Cumsum Plot
- 7 - Forecast
- 8 - Exit

Enter Query: 7

→ 207.34

#### دستور هفتم : پیش بینی Forecast

این دستور با توجه به تمامی تراکنش های منفی صورت گرفته، عددی را با توجه به بازه ی انجام تراکنش ها و تورم اقتصادی موجود در کشور تحت عنوان شارژ پیشنهادی ارائه می دهد.

نکته : دستور پیش بینی در بازه های بلند مدت ( حداقل 6 ماه ) ارزش خود را نشان می دهد و در بازه های کوتاه مدت فاقد اعتبار علمی است.

#### رعایت اصول زیبا و خلاصه نویسی گد :

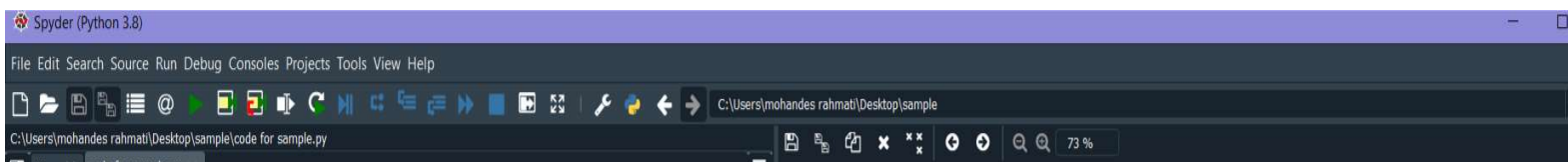
1. رعایت فاصله ها : در هر بخش فاصله ها دقیقا به شکل استاندارد یعنی ۴ اسپیس از بخش مربوطه قبلی فاصله دارند.
2. رعایت اصول استفاده از اپریتور ها : در هر بخش که از اپریتور ها استفاده شده اپرند ها با یک اسپیس از اپریتور جدا شده اند.
3. رعایت محدودیت تعداد کراکتر های هر خط : در هیچ یک از سطر های کد تعداد کراکتر ها از ۷۹ کراکتر تجاوز نمیکند.



## نکات الزامی برای صحیح اجرا شدن برنامه :

در ابتدا فایل **csv** ای با نام **init.csv** در محل کد پایتون قرار دارد که شامل اطلاعات اولیه ساختمان که در ذیل آمده می باشد و باید قبل از اجرا کد تکمیل شود. (این فایل حتما باید به صورت دستی تکمیل شود و در صورت عدم وجود **init.csv** یا تکمیل نبودن آن برنامه با مشکل روبرو می شود)

- نام واحد ها (name) : نام هر واحد میباشد که به صورت " id+شماره واحد " نوشته می شود. مثال: id13
- طبقه ( floor ) : عدد طبقه هر واحد است.
- تعداد ساکنین (residents) : یک عدد که نشان دهنده تعداد ساکنین می باشد.
- مساحت (area) : مقدار مساحت هر واحد.
- پارکینگ (parkings) : تعداد پارکینگ های اختصاص یافته به هر واحد.



در صورت استفاده از نرم افزار اسپایدر باید حتما مسیر ذخیره سازی و باز شدن داده ها را به صورت دستی در قسمت **directory** که بالای صفحه در قسمت راست وجود دارد وارد کنید.