**منو و زمان اجرای برنامه**

از وقتی که کاربر ورود یا ثبت نام را انجام میدهد، فایل menu.py اجرا می شود و شروع به محاسبه زمان استفاده از اپ می کند.

import time

from rich.console import Console

from rich.text import Text

from colors import blue, red, green, yellow, purple, cyan, magenta, white, black, gray

if \_\_name\_\_ == "\_\_main\_\_":

    startTime = time.time()  *# Start menu and app usage time recording*

    endTime = time.time()

Console().print(Text("Total time:", style=yellow), Text(

    f"{endTime - startTime:.2f}", style=magenta), Text("seconds", style=yellow))

کتابخانه time را وارد کرده و در هر جا که نیاز به ذخیره زمان باشد، متد time.time() را فرا میخوانیم. یکبار در آغاز اجرای برنامه و یکبار در آخر که دیگر استفاده از برنامه تمام می شود تا زمان استفاده از برنامه را نشان دهد.

[منبع](https://stackoverflow.com/questions/1557571/how-do-i-get-time-of-a-python-programs-execution#:~:text=Just%20use%20the%20timeit%20module,Python%202%20and%20Python%203.&text=Deprecated%20since%20version%203.3%3A%20The%20behavior%20of%20this%20function%20depends,have%20a%20well%2Ddefined%20behavior.)

**استفاده از رنگ ها**

برای رنگی و نمایش زیبای خروجی ها از چت چی پی تی خواستم که به من چند راهکار معرفی کند و از بین آن ها کتابخانه rich بهترین گزینه به نظر آمد. یک فایل به نام colors.py ساختیم.

from rich.style import Style

blue = Style(color="blue")

red = Style(color="red")

green = Style(color="green")

yellow = Style(color="yellow")

purple = Style(color="purple")

cyan = Style(color="cyan")

magenta = Style(color="magenta")

white = Style(color="white")

black = Style(color="black")

gray = Style(color="grey70")

[منبع 1](https://rich.readthedocs.io/en/stable/index.html)

[منبع 2](https://chat.openai.com/)

**کلاس MainMenu**

در فایل menu.pyمنوی ما 7 آپشن دارد که یک لیست از آن ها میسازیم. همچنین انتخاب کاربر را نیز تعریف می کنیم و به صورت پیش فرض آن را 0 می گذاریم که اولین ایندکس است.

class MainMenu():

    def \_\_init\_\_(self):

*self*.options = [

            "Register Income",

            "Register Expense",

            "Categories",

            "Search",

            "Reporting",

            "Settings",

            "Exit"

        ]

*self*.selectedOption = 0

**تابع display\_menu**

در این تابع که از نامش مشخص است، باید منو را نمایش بدهیم. آن چیزی که انتخاب کاربر است را به صورت پیش فرض پررنگ تر نشان میدهیم. همچنین راهنمایی های لازم را برای کاربر که می خواهد ورودی بگذارد را می نویسیم.در مورد ورودی و منو در تابع بعدی توضیح میدهم.در صورت وجود اخطار نیز آن را برای مدت کوتاهی نمایش میدهیم (توضیحات در تابع بعدی)

    def display\_menu(self, errorMessage=None):

        Console().clear()

        Console().print(Text("Main Menu:", style=cyan))

        Console().print(Text("Press arrow keys and enter button or press the number of your choice (range 1-7). Press 'q' to exit.", style=gray))

        if errorMessage:

            Console().print(Text(f"Error: {errorMessage}", style=red))

        for i, option in enumerate(*self*.options):

            if i == *self*.selectedOption:

                Console().print(Text(f"-> {i + 1}- {option}", style=blue))

            else:

                Console().print(Text(f"   {i + 1}- {option}", style=cyan))

**تابع handle\_input**

خروجی این تابع یک عضو لیست options خواهد بود که ایندکس اعضا آن عدد بین 0 تا 6 هست که همان 7 انتخاب منو می باشد. از کتابخانه msvcrt استفاده می کنیم. چون علاوه بر عدد، از arrow key کیبورد و کد اسکی آن ها استفاده می کنیم.

    def handle\_input(self):

        try:

            key = msvcrt.getch()

            if key == b'\xe0':  *# Arrow keys are preceded by '\xe0'. if this conditon not checked, we will get an invalid input error when using arrow keys*

                keyCode = ord(msvcrt.getch())

                if keyCode == 72:  *# Up arrow key*

*self*.selectedOption = (

*self*.selectedOption - 1) % len(*self*.options)

                elif keyCode == 80:  *# Down arrow key*

*self*.selectedOption = (

*self*.selectedOption + 1) % len(*self*.options)

            elif key == b'\r':  *# Enter key*

                time.sleep(1)

                return *self*.options[*self*.selectedOption]

            elif key == b'q':  *# 'q' key*

*self*.exit()

            elif key.isdigit():  *# If a digit is pressed*

                optionNum = int(key)

                if 1 <= optionNum <= len(*self*.options):

                    return *self*.options[optionNum - 1]

                else:

                    raise ValueError("Number out of range")

            else:

                raise ValueError("Invalid input")

        except ValueError as e:

*self*.display\_menu(str(e))

            time.sleep(3)  *# Pause to show the error message and then get input again if our input value was not valid.*

        except Exception as e:

            Console().print(

                Text(f"An unexpected error occurred: {e}", style=red))

            return None

در ادامه 7 عملیات اصلی منو را نیز باید در این تابع باید بگذاریم. (بخش های بعدی پروژه)

[منبع 1](https://docs.python.org/3/library/msvcrt.html)

[منبع 2](https://www.w3schools.com/python/python_try_except.asp)

**تابع exit**

برای بستن برنامه نو نمایش میزان استفاده او از اپ نوشته شده. فعلا فقط برای حالتی که کاربر دکمه q را بزند کار میکند و اگر کاربر خودش 7 را انتخاب کند . فقط انتخب او را نشان میدهد. هنگام زدن 7 بخش اصلی منو، این را اضافه می کنیم.

    def exit(self):

        endTime = time.time()

        Console().print(Text("Total time:", style=yellow), Text(

            f"{endTime - startTime:.2f}", style=purple), Text("seconds", style=yellow))

        Console().print(Text("Exiting ...", style=magenta))

        time.sleep(1)

        sys.exit()

[منبع 1](https://www.freecodecamp.org/news/python-exit-how-to-use-an-exit-function-in-python-to-stop-a-program/)

**تغییرات منو و زمان اجرای برنامه**

if \_\_name\_\_ == "\_\_main\_\_":

    startTime = time.time()  *# Start menu and app usage time recording*

    menu = MainMenu()

    while True:

        menu.display\_menu()

        selectedOptionName = menu.handle\_input()

        if selectedOptionName:

            Console().print(Text(f"Selected option: {

                selectedOptionName}", style=purple))

            break

    endTime = time.time()

    Console().print(Text("Total time:", style=yellow), Text(

        f"{endTime - startTime:.2f}", style=purple), Text("seconds", style=yellow))

این بخش با پیشرفت پروژه دچار تغییر خواهد شد. فعلا انتخب کاربر را صرفا نشان می دهد و برنامه را میبندد و زمان استفاده او را نشان میدهد.

یک نگاه به کتابخانه هایی که تا الان استفاده شده.

import time

from rich.console import Console

from rich.text import Text

from colors import blue, red, green, yellow, purple, cyan, magenta, white, black, gray

import sys

import msvcrt

**ثبت درآمد و هزینه**

چون این دو مثل هم هستند، از یک کلاس برای ساختن آن ها استفاده می کنیم.