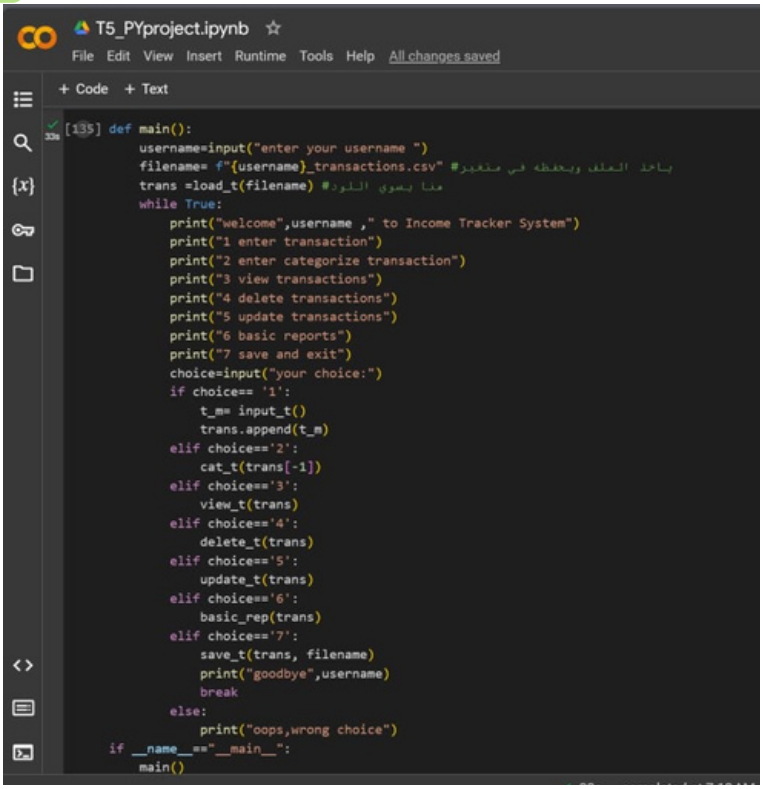
# INCOME TRACKER

MADAWI BIN SHAWIAH
&
KHAFUQ ALTHEYAB

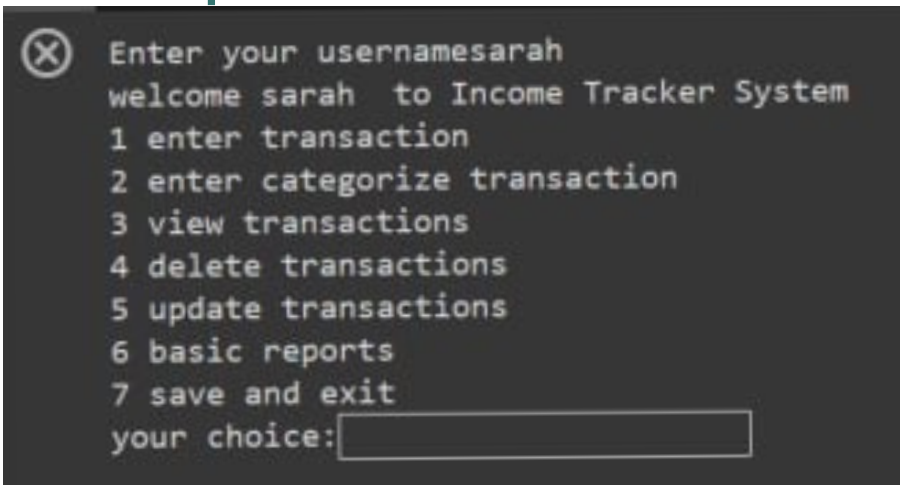
**MAIN FUN:** To call the functions

## Code:

```
def main():
 username=input("enter your username ")
 filename= f"{username}_transactions.csv" #ياخذ الملف ويحفظه في متغير
 trans =load_t(filename) #هنا يسوي اللود
 while True:
  print("welcome",username ," to Income Tracker System")
  print("1 enter transaction")
  print("2 enter categorize transaction")
  print("3 view transactions")
  print("4 delete transactions")
  print("5 update transactions")
  print("6 basic reports")
  print("7 save and exit")
  choice=input("your choice:")
  if choice== '1':
   t_m= input_t()
   trans.append(t_m)
  elif choice=='2':
   cat_t(trans[-1])
  elif choice=='3':
   view_t(trans)
  elif choice=='4':
   delete_t(trans)
  elif choice=='5':
   update_t(trans)
  elif choice=='6':
   basic_rep(trans)
  elif choice=='7':
   save_t(trans, filename)
   print("goodbye",username)
   break
  else:
   print("oops,wrong choice")
if name=="main":
 main()
```



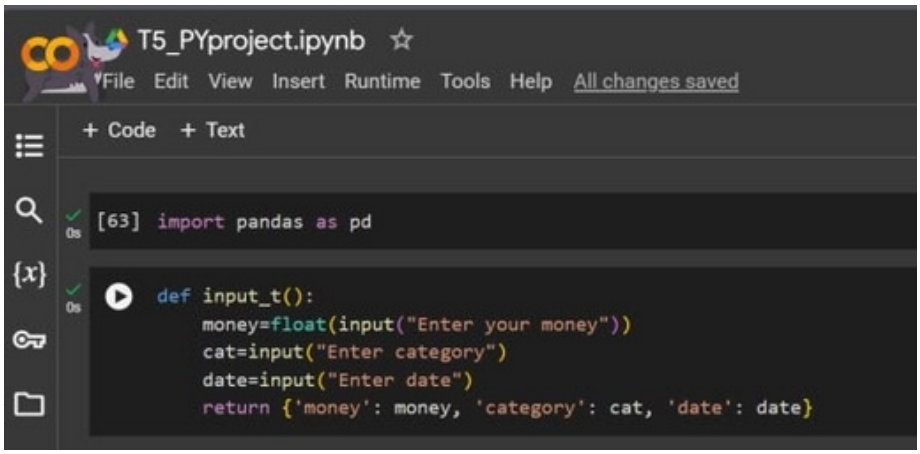

## Output:



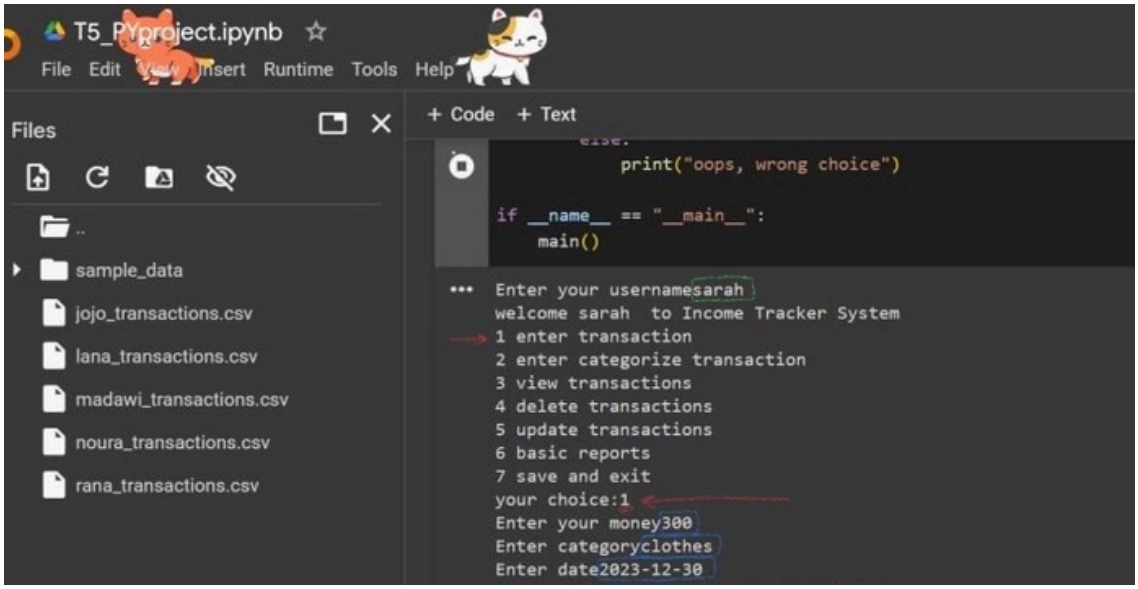
## INCOM ENTRY FEATURE:

## Code:

```
def input_t():
  money=float(input("Enter your money"))
  cat=input("Enter category")
  date=input("Enter date")
  return {'money': money, 'category': cat, 'date': date}
```




## Output:

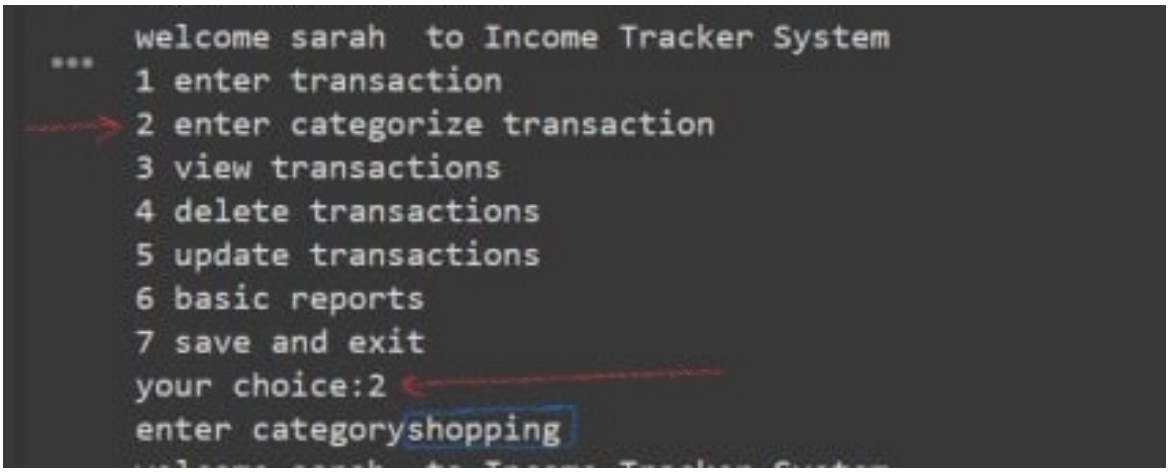


## CATEGORISATION FEATURE:

## Code:

```
def cat_t(tran):
 cate=input("enter category") #استقبال كاتقري
 tran['category']=cate #اضافة
```

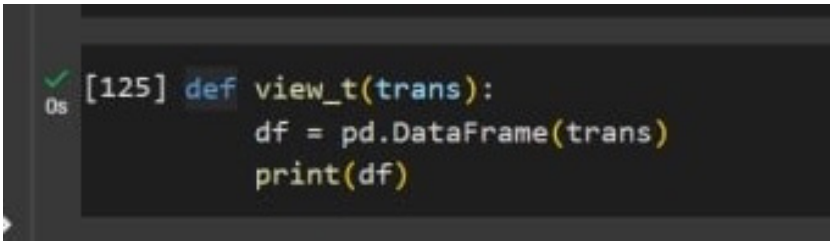

## Output:



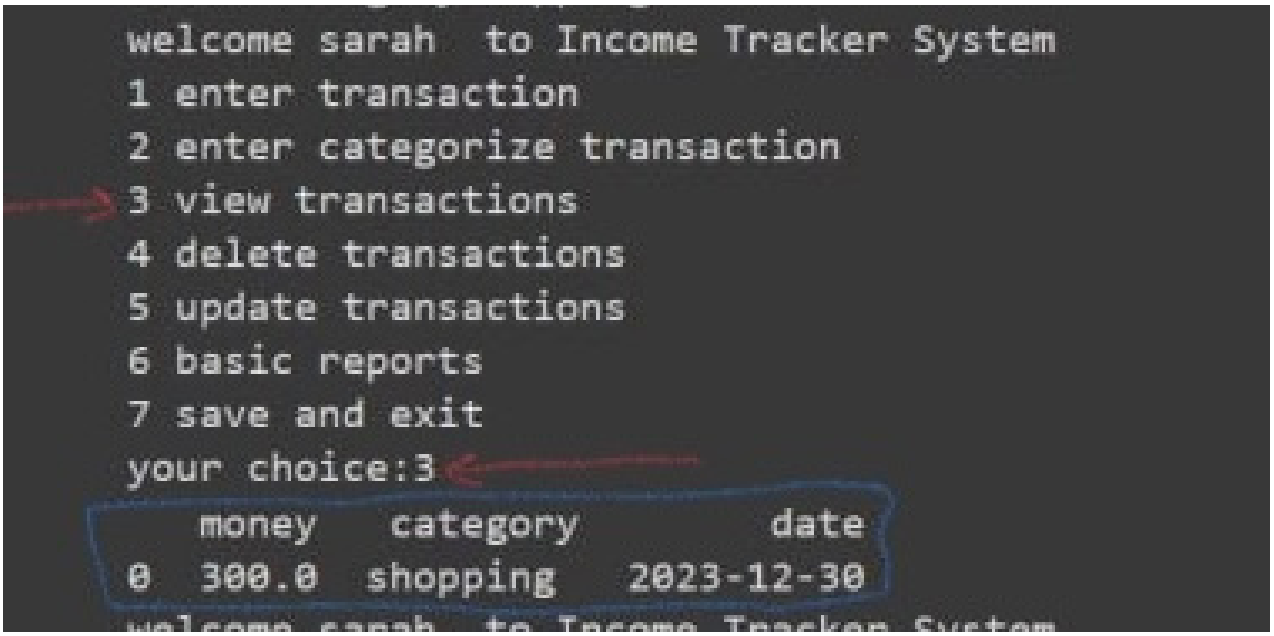
## VIEW FEATURE:

## Code:

```
def view_t(trans):
  df = pd.DataFrame(trans)
  print(df)
```



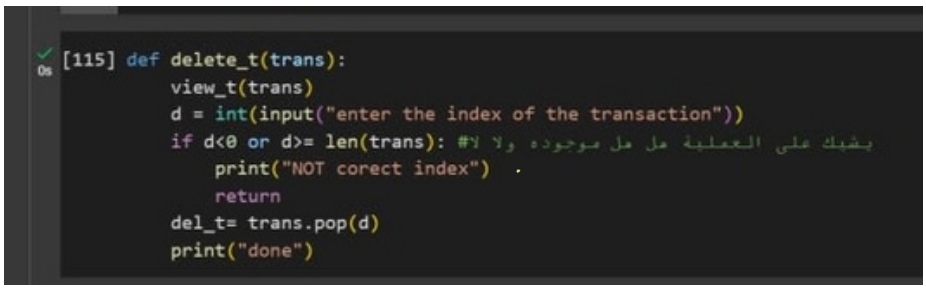

## Output:



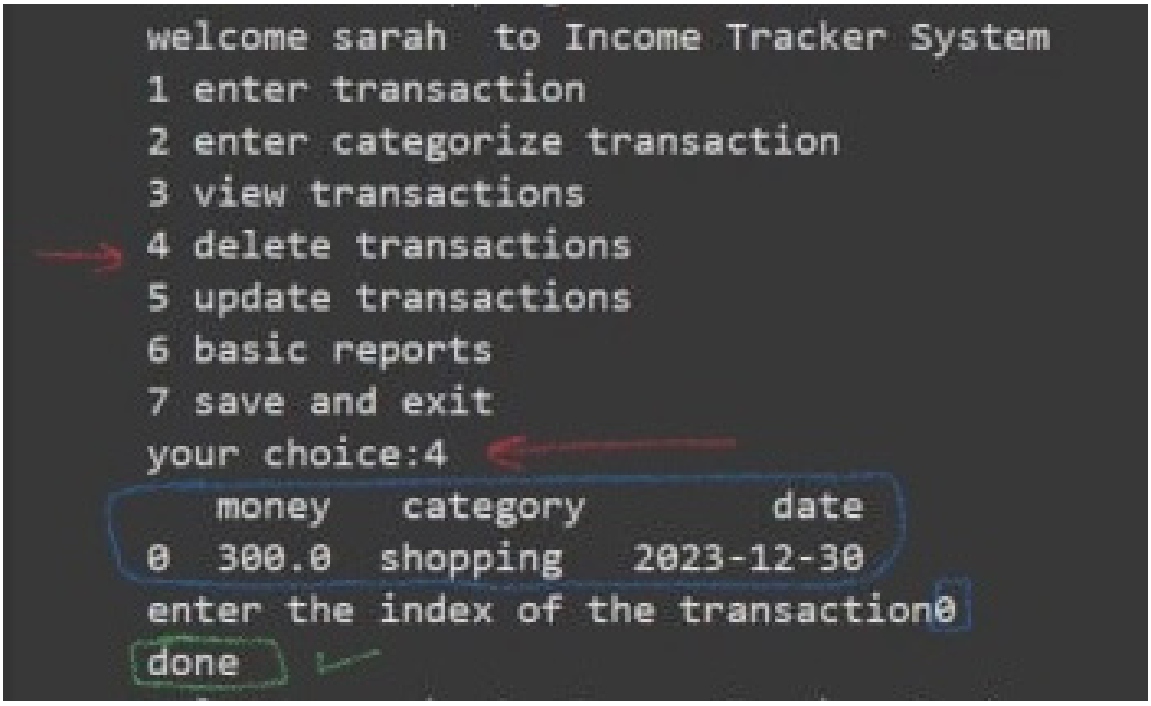
## DELETE FEATURE:

## Code:

```
def delete_t(trans):
  view_t(trans)
  d = int(input("enter the index of the transaction"))
  if d<0 or d>= len(trans): #يشيك على العملية هل هل موجوده ولا لا
    print("NOT corect index")
    return
  del_t= trans.pop(d)
  print("done")
```



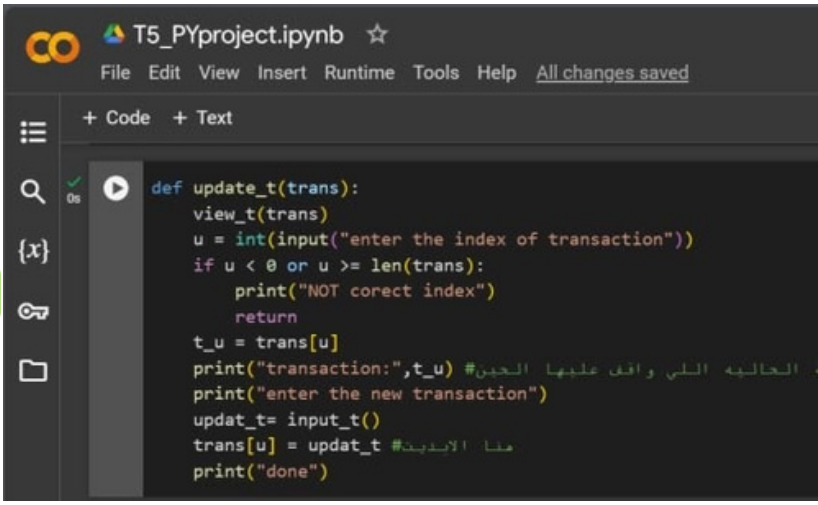

## Output:

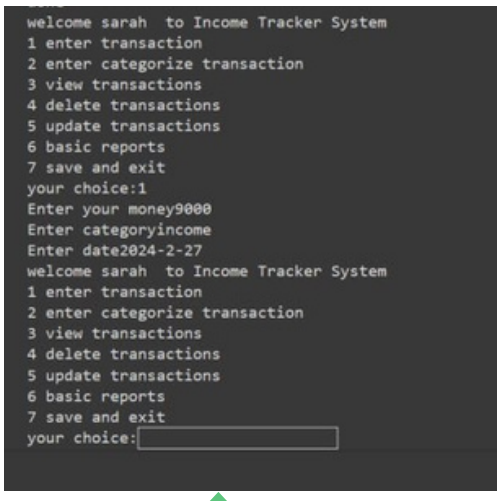


## UPDATE FEATURE:

# Code:

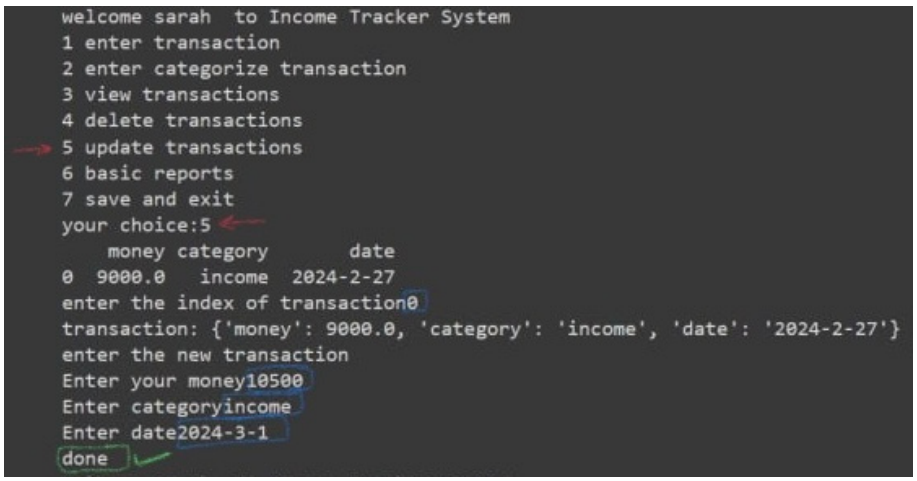
```
def update_t(trans):
  view_t(trans)
  u = int(input("enter the index of transaction"))
  if u < 0 or u >= len(trans):
    print("NOT corect index")
    return
  t_u = trans[u]
  print("transaction:",t_u)
  print("enter the new transaction")
  updat_t= input_t()
  trans[u] = updat_t #هنا الابديت
  print("done")
```




A new transaction has been added to apply the update feature, because all previous transactions have been deleted

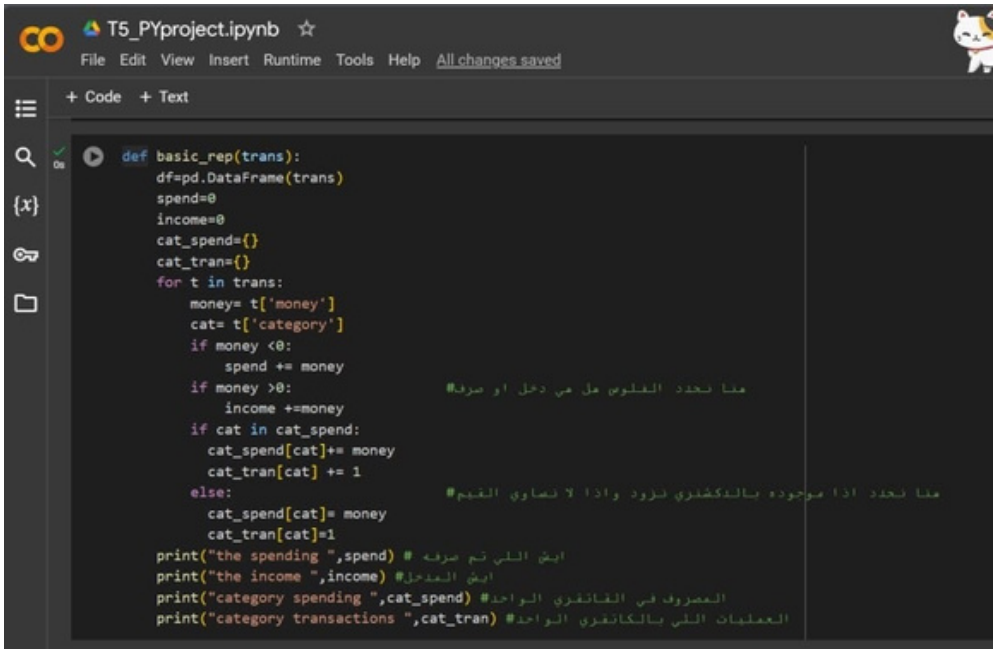
# Output:



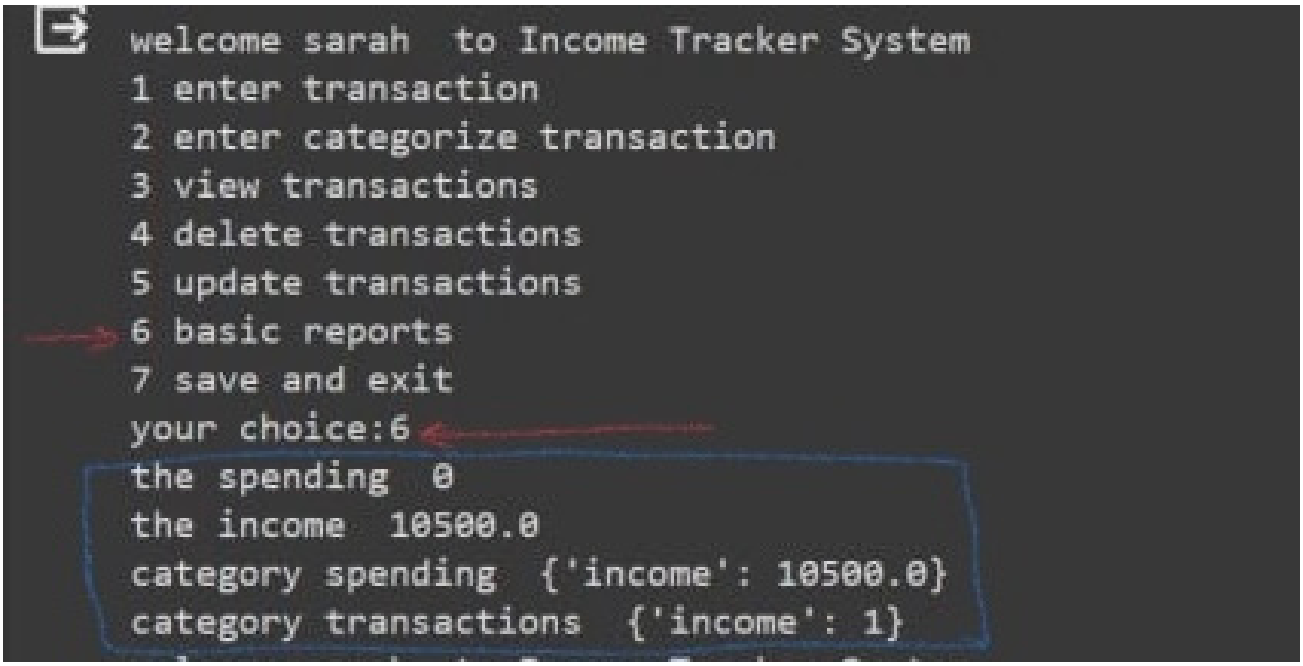
## BASIC REPORTS FEATURE:

# Code:

```
def basic_rep(trans):
 df=pd.DataFrame(trans)
 spend=0
 income=0
 cat_spend={}
 cat_tran={}
 for t in trans:
  money= t['money']
  cat= t['category']
  if money <0:
   spend += money
  if money >0:
   income +=money
  if cat in cat_spend:
   cat_spend[cat]+= money
   cat_tran[cat] += 1
  else:
   cat_spend[cat]= money
   cat_tran[cat]=1
 print("the spending ",spend)
 print("the income ",income)
 print("category spending ",cat_spend)
 print("category transactions ",cat_tran)
```




# Output:



# DATA PERSISTENCE FEATURE:

```
def save_t(trans, filename):
  df = pd.DataFrame(trans)
  df.to_csv(filename)
```

```
def load_t(filename):
  try:
    df = pd.read_csv(filename)
    trans = df.to_dict('records')
    return trans
  except FileNotFoundError:
    return []
```