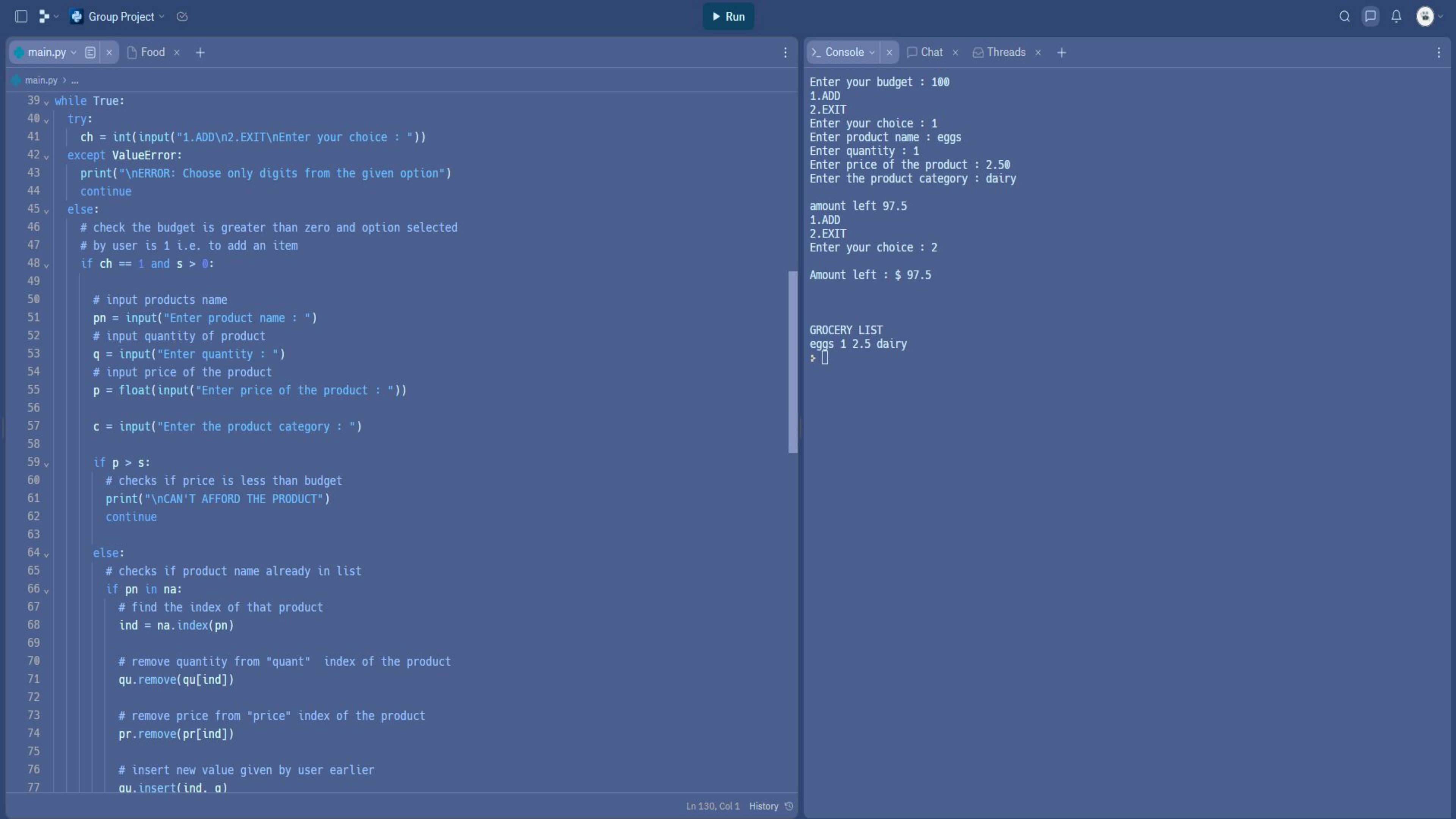
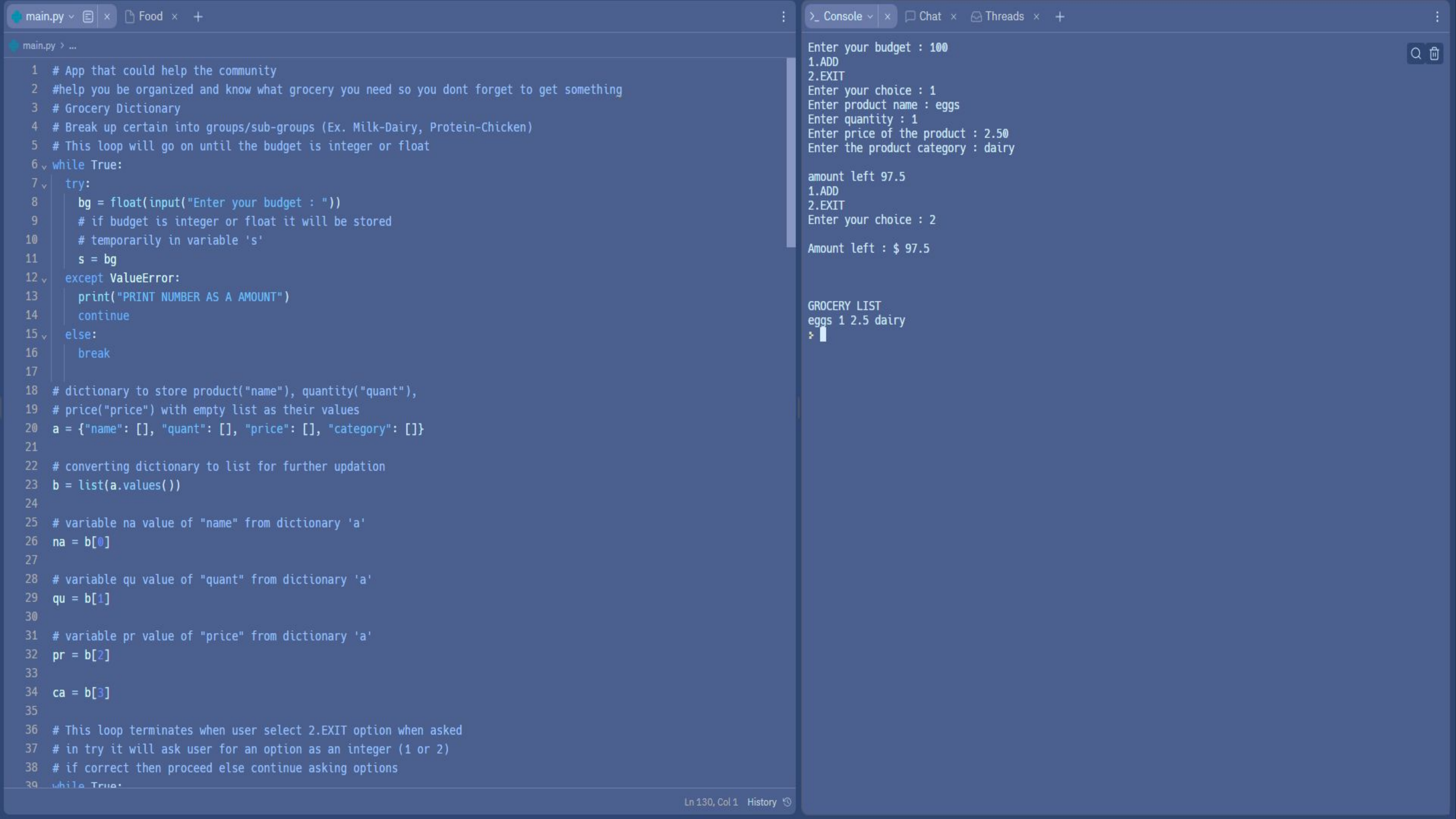
Grocer-eady

[redacted]



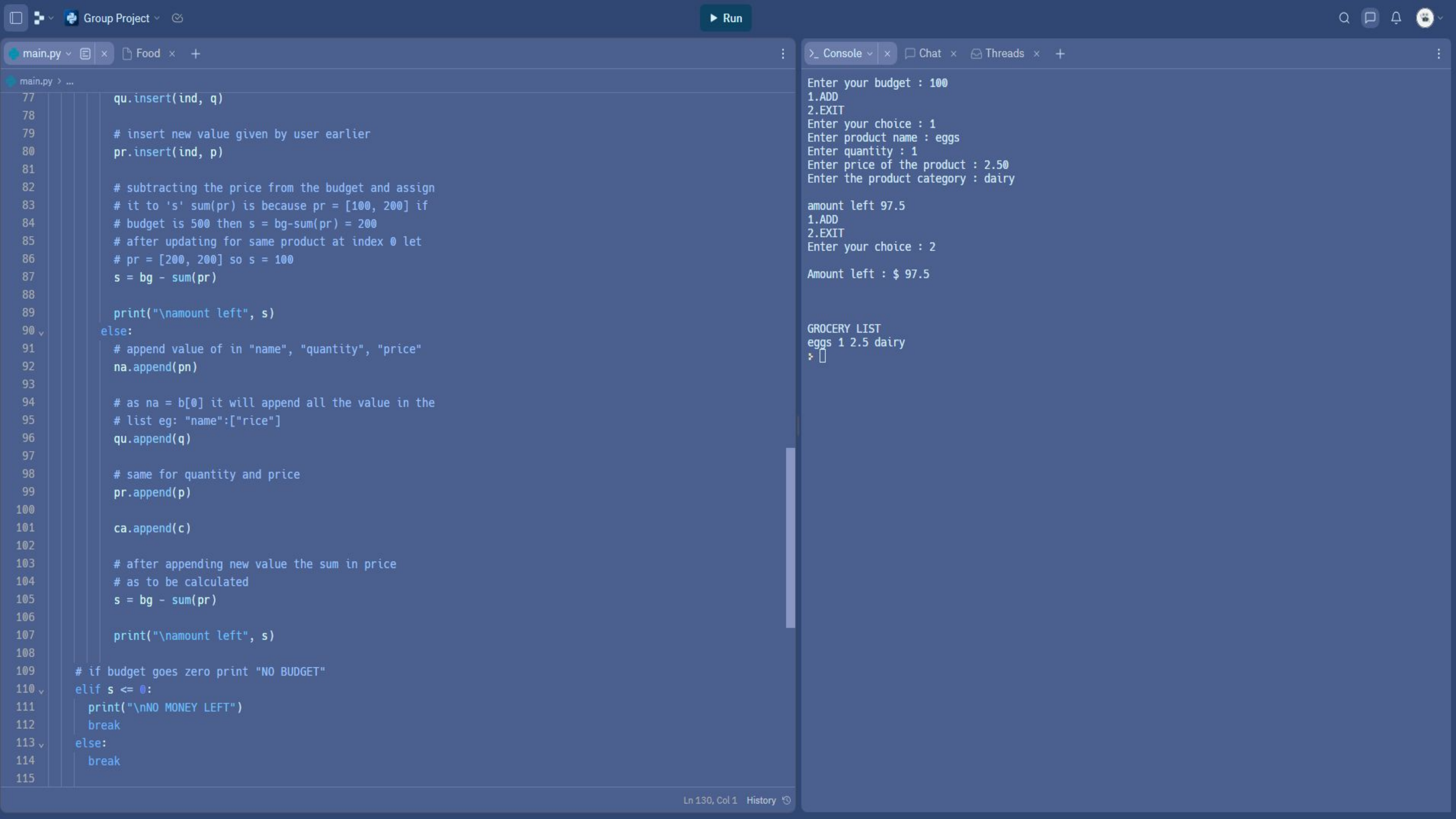
```python
while True:
  try:
    ch = int(input("1.ADD\n2.EXIT\nEnter your choice : "))
  except ValueError:
    print("\nERROR: Choose only digits from the given option")
    continue
  else:
    # check the budget is greater than zero and option selected
    # by user is 1 i.e. to add an item
    if ch == 1 and s > 0:

      # input products name
      pn = input("Enter product name : ")
      # input quantity of product
      q = input("Enter quantity : ")
      # input price of the product
      p = float(input("Enter price of the product : "))

      c = input("Enter the product category : ")

      if p > s:
        # checks if price is less than budget
        print("\nCAN'T AFFORD THE PRODUCT")
        continue

      else:
        # checks if product name already in list
        if pn in na:
          # find the index of that product
          ind = na.index(pn)

          # remove quantity from "quant"  index of the product
          qu.remove(qu[ind])

          # remove price from "price" index of the product
          pr.remove(pr[ind])

          # insert new value given by user earlier
          qu.insert(ind, q)
```

```
Enter your budget : 100
1.ADD
2.EXIT
Enter your choice : 1
Enter product name : eggs
Enter quantity : 1
Enter price of the product : 2.50
Enter the product category : dairy

amount left 97.5
1.ADD
2.EXIT
Enter your choice : 2

Amount left : $ 97.5



GROCERY LIST
eggs 1 2.5 dairy
> 
```



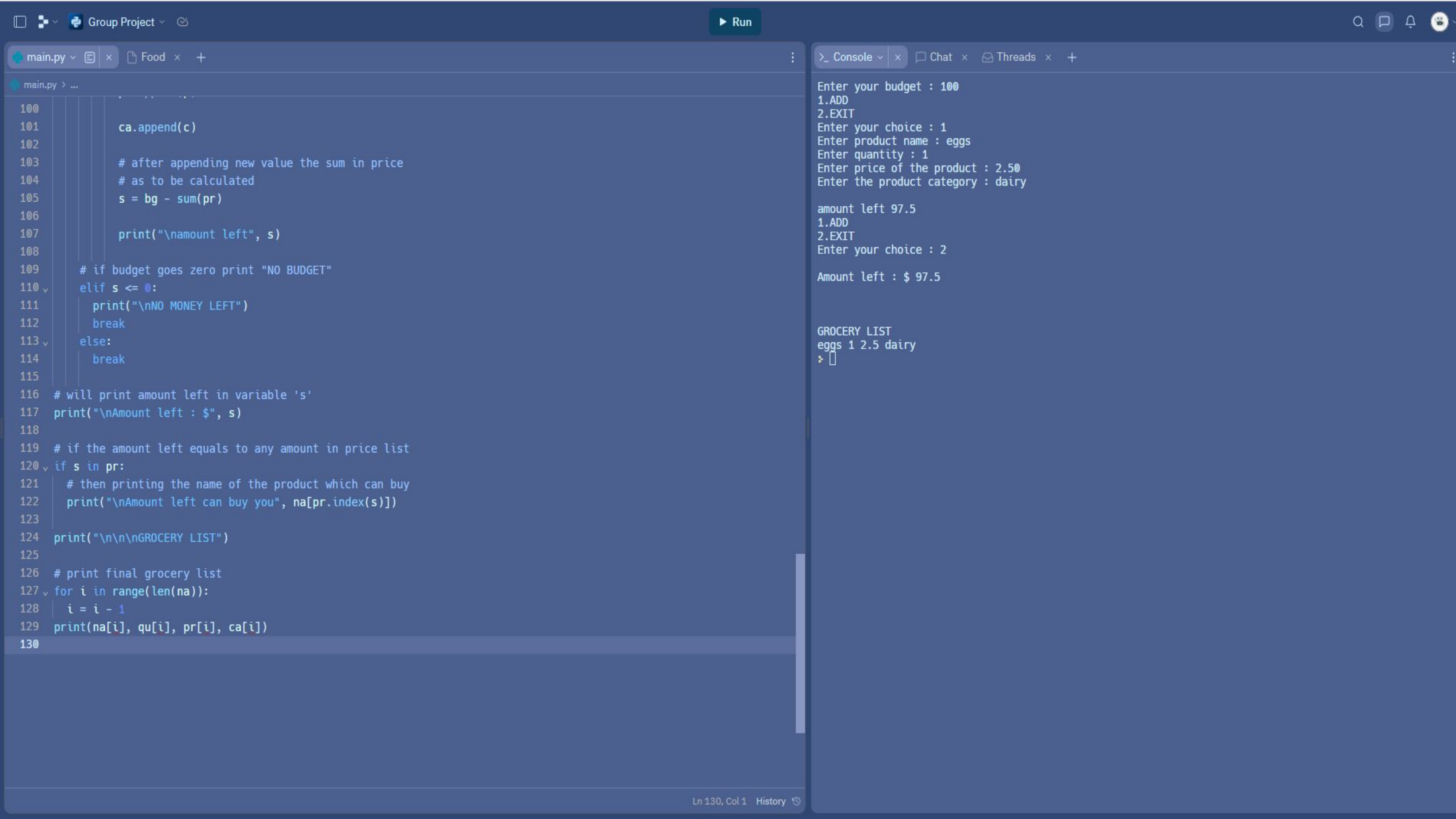
```python
# App that could help the community
#help you be organized and know what grocery you need so you dont forget to get something
# Grocery Dictionary
# Break up certain into groups/sub-groups (Ex. Milk-Dairy, Protein-Chicken)
# This loop will go on until the budget is integer or float
while True:
  try:
    bg = float(input("Enter your budget : "))
    # if budget is integer or float it will be stored
    # temporarily in variable 's'
    s = bg
  except ValueError:
    print("PRINT NUMBER AS A AMOUNT")
    continue
  else:
    break

# dictionary to store product("name"), quantity("quant"),
# price("price") with empty list as their values
a = {"name": [], "quant": [], "price": [], "category": []}

# converting dictionary to list for further updation
b = list(a.values())

# variable na value of "name" from dictionary 'a'
na = b[0]

# variable qu value of "quant" from dictionary 'a'
qu = b[1]

# variable pr value of "price" from dictionary 'a'
pr = b[2]

ca = b[3]

# This loop terminates when user select 2.EXIT option when asked
# in try it will ask user for an option as an integer (1 or 2)
# if correct then proceed else continue asking options
while True:
```

```
Enter your budget : 100
1.ADD
2.EXIT
Enter your choice : 1
Enter product name : eggs
Enter quantity : 1
Enter price of the product : 2.50
Enter the product category : dairy

amount left 97.5
1.ADD
2.EXIT
Enter your choice : 2

Amount left : $ 97.5



GROCERY LIST
eggs 1 2.5 dairy
> 
```



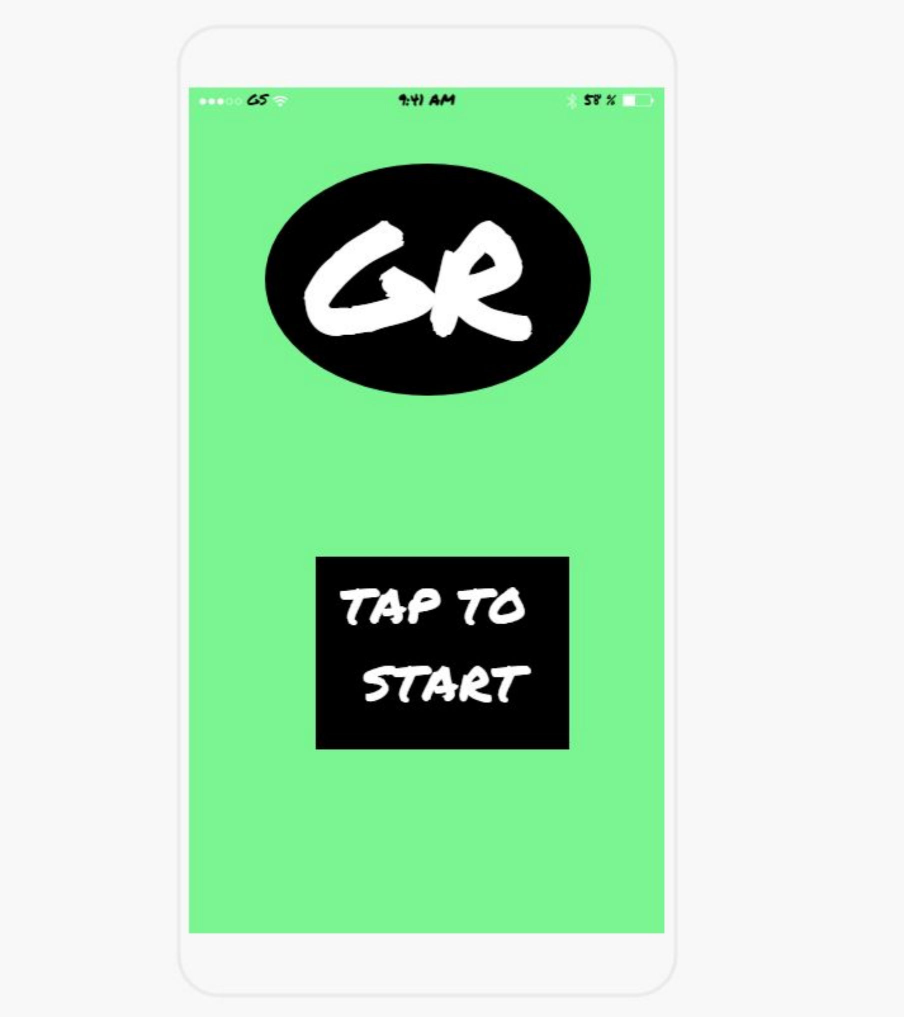
```python
          qu.insert(ind, q)

          # insert new value given by user earlier
          pr.insert(ind, p)

          # subtracting the price from the budget and assign
          # it to 's' sum(pr) is because pr = [100, 200] if
          # budget is 500 then s = bg-sum(pr) = 200
          # after updating for same product at index 0 let
          # pr = [200, 200] so s = 100
          s = bg - sum(pr)

          print("\namount left", s)
        else:
          # append value of in "name", "quantity", "price"
          na.append(pn)

          # as na = b[0] it will append all the value in the
          # list eg: "name":["rice"]
          qu.append(q)

          # same for quantity and price
          pr.append(p)

          ca.append(c)

          # after appending new value the sum in price
          # as to be calculated
          s = bg - sum(pr)

          print("\namount left", s)

    # if budget goes zero print "NO BUDGET"
    elif s <= 0:
      print("\nNO MONEY LEFT")
      break
    else:
      break

```

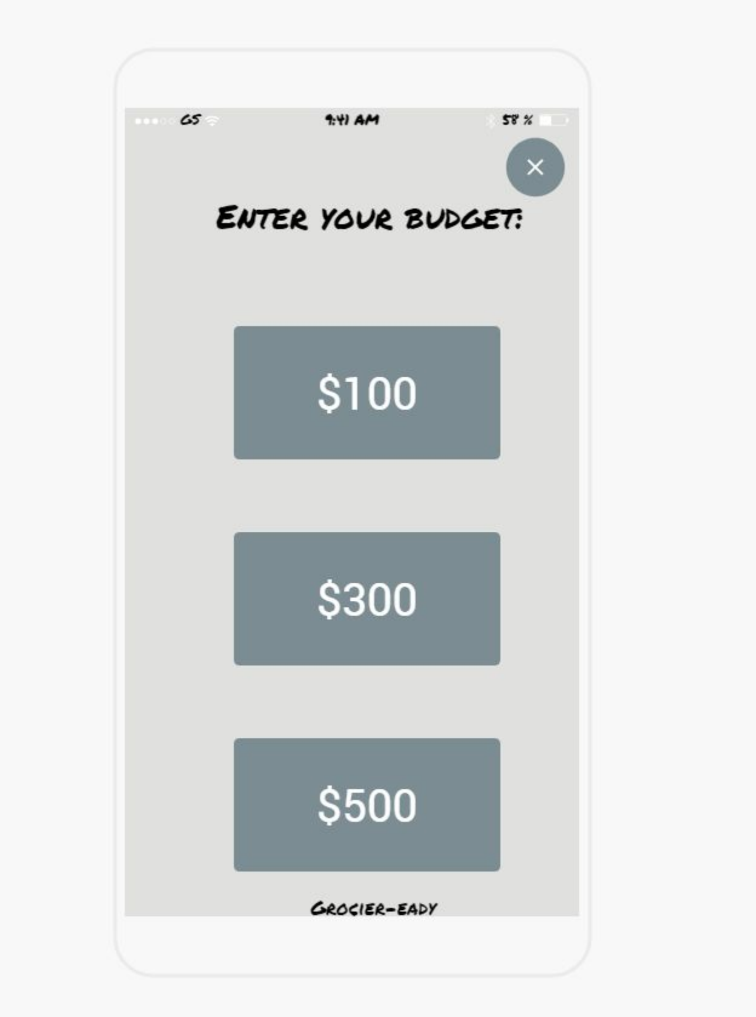
```
Enter your budget : 100
1.ADD
2.EXIT
Enter your choice : 1
Enter product name : eggs
Enter quantity : 1
Enter price of the product : 2.50
Enter the product category : dairy

amount left 97.5
1.ADD
2.EXIT
Enter your choice : 2

Amount left : $ 97.5



GROCERY LIST
eggs 1 2.5 dairy
> 
```

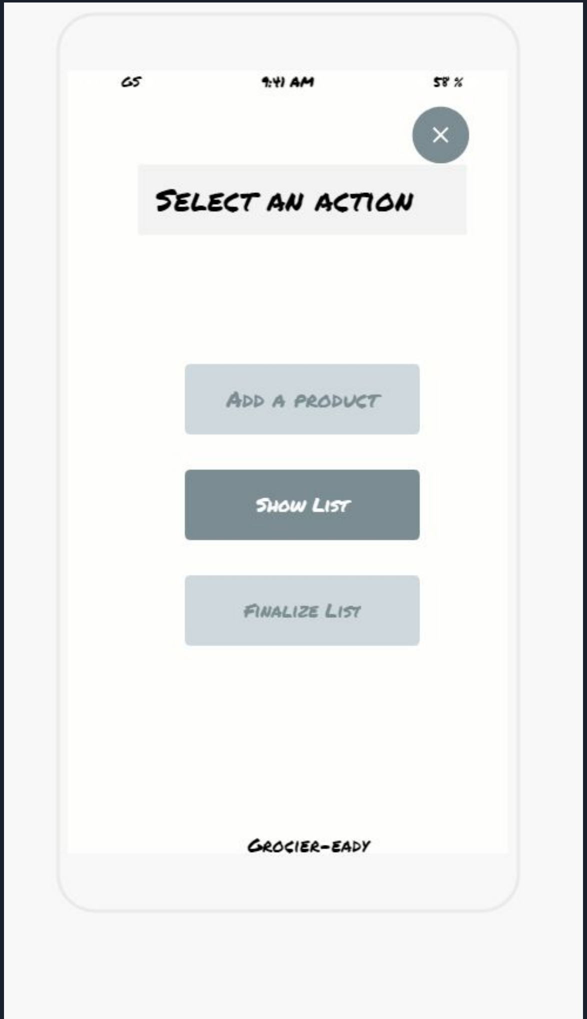


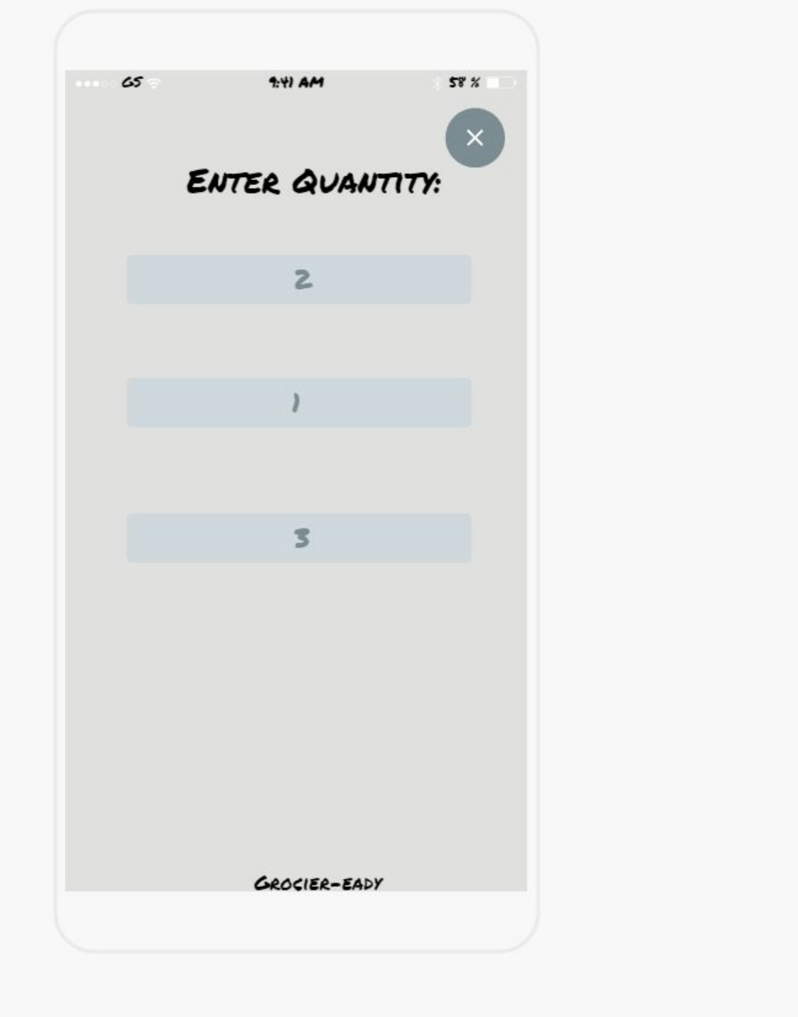
```python

          ca.append(c)

          # after appending new value the sum in price
          # as to be calculated
          s = bg - sum(pr)

          print("\namount left", s)

    # if budget goes zero print "NO BUDGET"
    elif s <= 0:
      print("\nNO MONEY LEFT")
      break
    else:
      break

# will print amount left in variable 's'
print("\nAmount left : $", s)

# if the amount left equals to any amount in price list
if s in pr:
  # then printing the name of the product which can buy
  print("\nAmount left can buy you", na[pr.index(s)])

print("\n\n\nGROCERY LIST")

# print final grocery list
for i in range(len(na)):
  i = i - 1
print(na[i], qu[i], pr[i], ca[i])

```

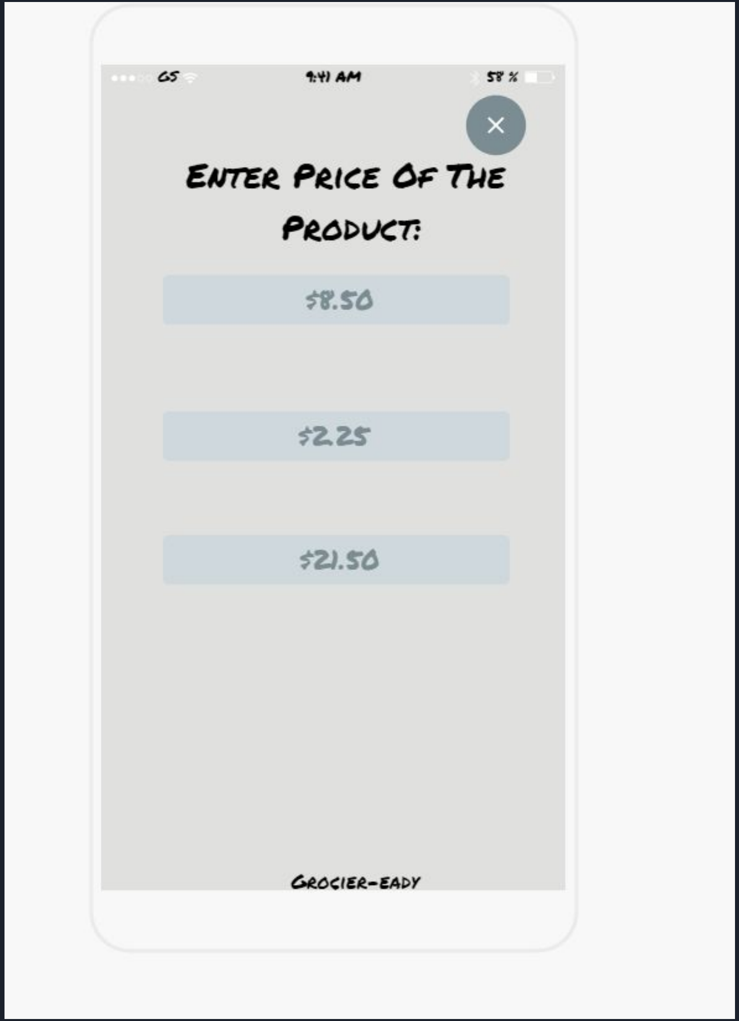
```
Enter your budget : 100
1.ADD
2.EXIT
Enter your choice : 1
Enter product name : eggs
Enter quantity : 1
Enter price of the product : 2.50
Enter the product category : dairy

amount left 97.5
1.ADD
2.EXIT
Enter your choice : 2

Amount left : $ 97.5



GROCERY LIST
eggs 1 2.5 dairy
> 
```


●●●○○ GS
9:41 AM
58 %
GR
TAP TO START

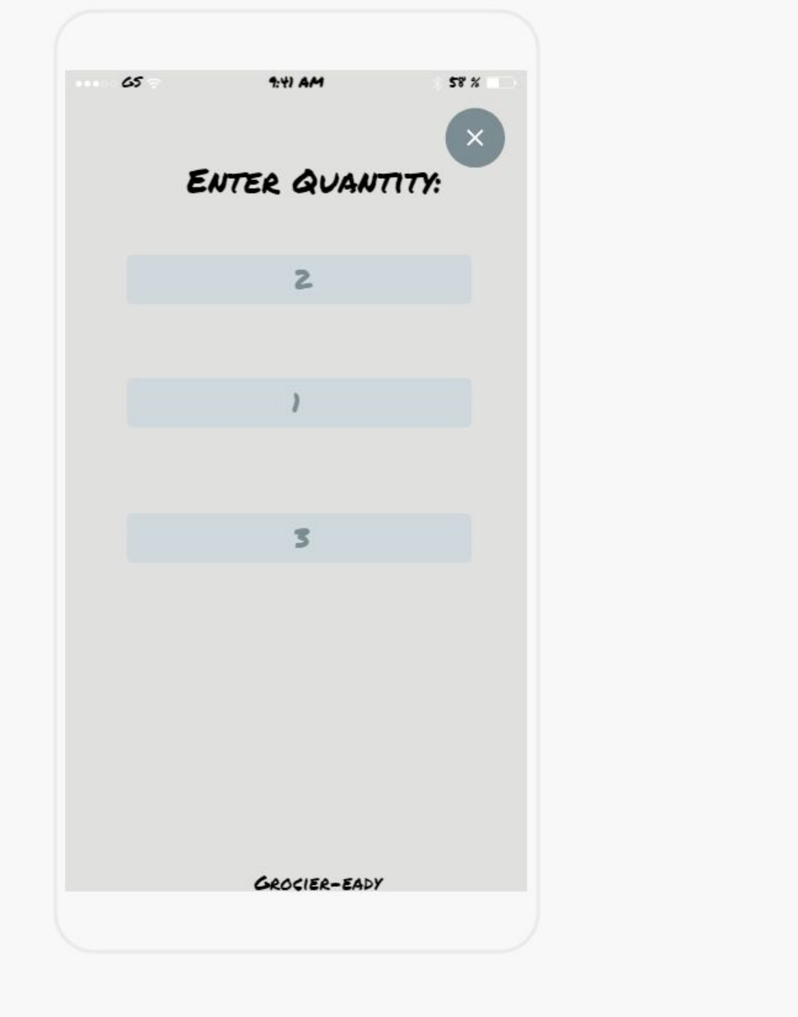

●●●○○ GS
9:41 AM
58 %
×
ENTER YOUR BUDGET:
$100
$300
$500
GROCIER-EADY


GS
9:41 AM
58 %
×
Select an action
Add a product
Show List
Finalize List
Grocier-eady

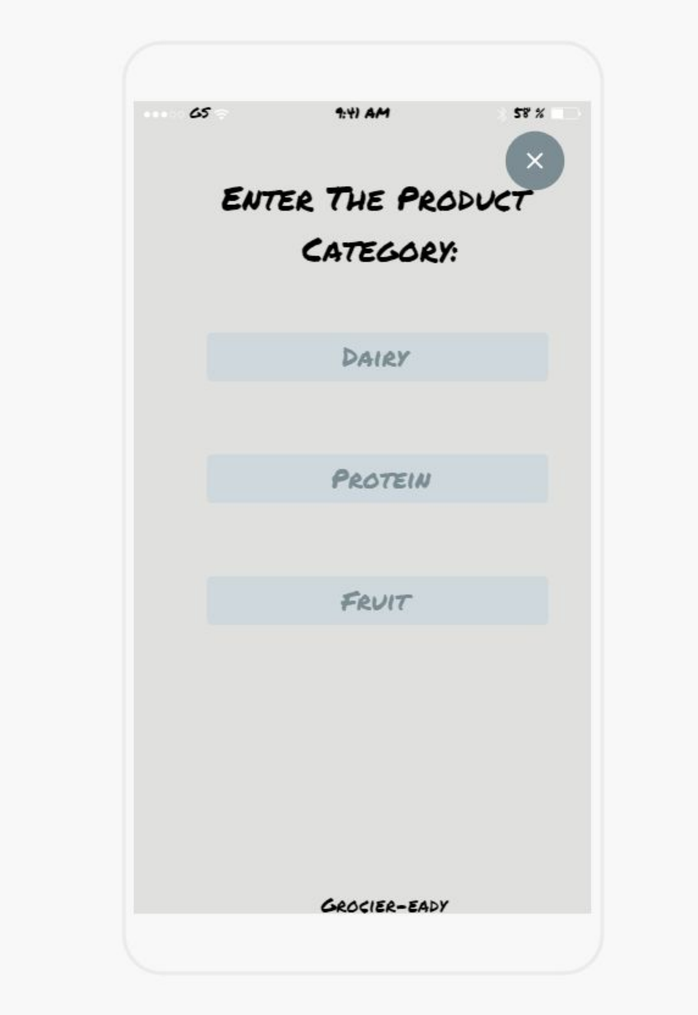

GS
9:41 AM
58 %
×
Enter Quantity:
2
1
3
Grocier-eady



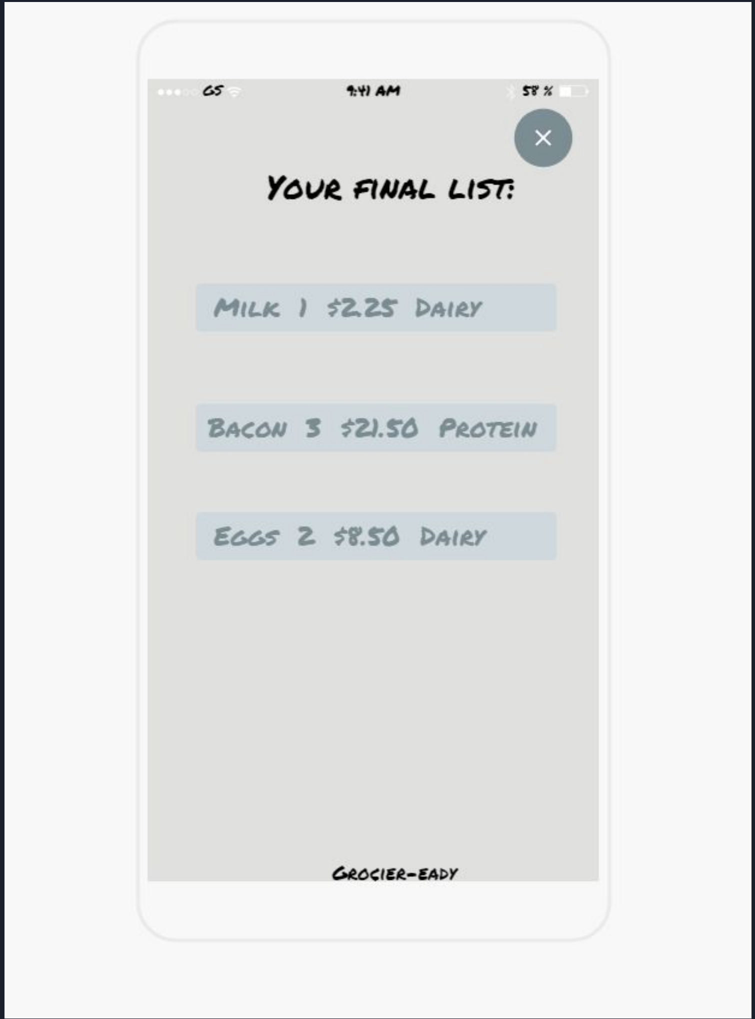
Enter Price Of The Product:



Enter Quantity:



Enter The Product Category:



Your final list:

Milk	1	$2.25	Dairy
Bacon	3	$21.50	Protein
Eggs	2	$8.50	Dairy

App Description

The app helps you remember what you need to get from the store because people are always forgetting one thing from the store and it is annoying that you remember it when you get home and have to go back to the store and get it. But that changes with our app.

App Difficulties

A problem when making the app was making a break function because the app didn’t stop when wanted it to stop and also adding a category section was a problem because it was showing the text for the category section.