**Development of a Grocery Receipt Generator Using C# Console Applicaꢀon**

*GROUP MEMBERS*

*JULIE WAMBUI -BSCIT-01-0035/2024*

*DEBORAH GATHUKU -BSCIT-01-0036/2024*

*EUNICE ONGIRI-BSCIT-05-0814/2023*

*KIMANI MUKUNDI-BSCIT-01-0051/2024*

*KIOKO BERNARD-BSCIT-01-0044/2024*

**Introducꢀon**

This document explains the step-by-step process followed to create a simple Grocery Receipt

Generator using C#. The system is a console-based applicaꢀon developed in Visual Studio Code.

It uses a text ﬁle for reading grocery data and another one for outpuꢁng the receipt

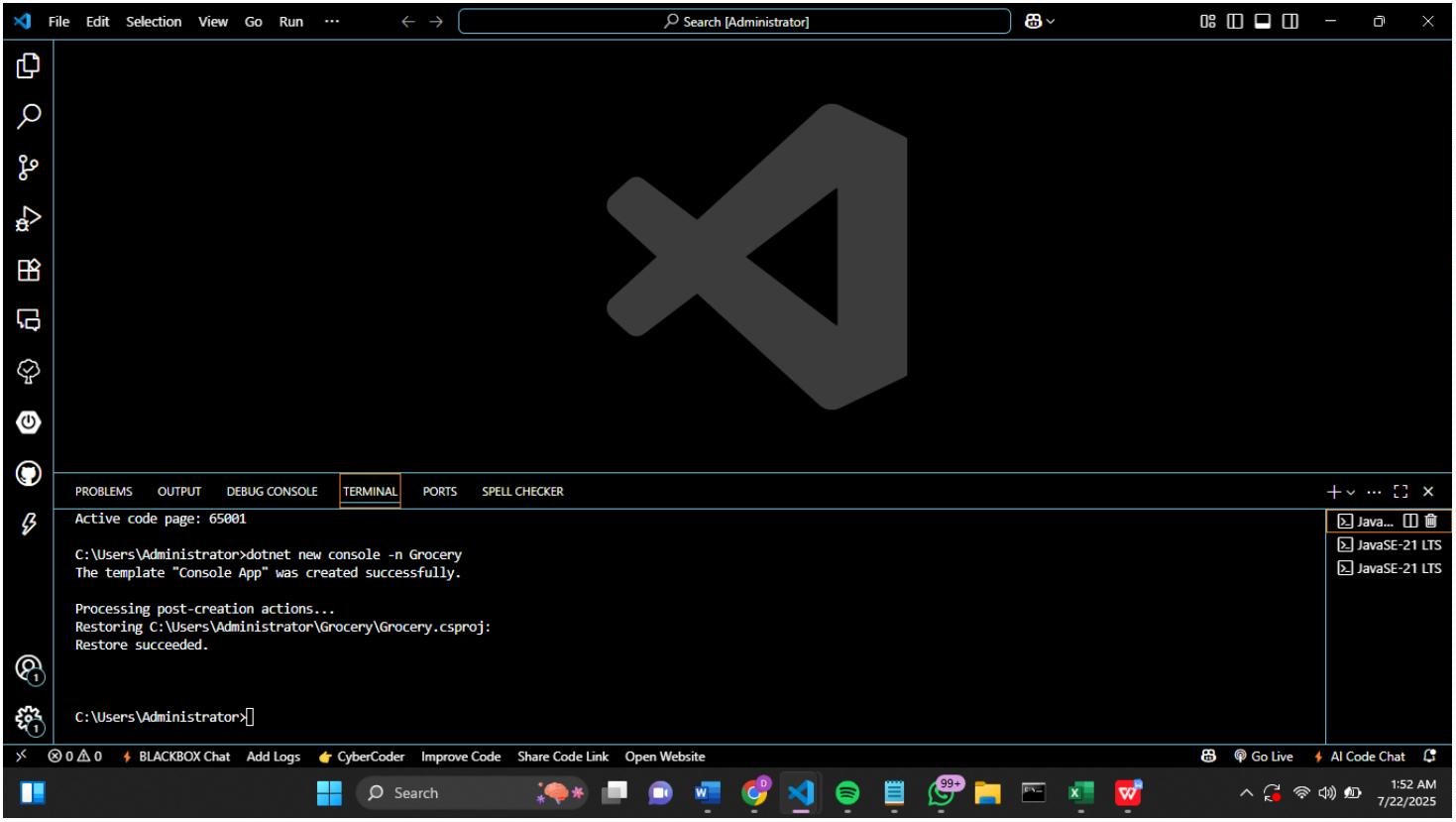
**Seꢁng Up the Project**

Created a new console app using the following command:

dotnet new console -n Grocery

Aﬅer running the command, a folder named GroceryApp was automaꢀcally created containing

the starter ﬁles.



**Opening the Project in Visual Studio Code**

We opened the new project using:

cd GroceryApp

This opened the project in Visual Studio Code where We could now edit the Program.cs ﬁle.

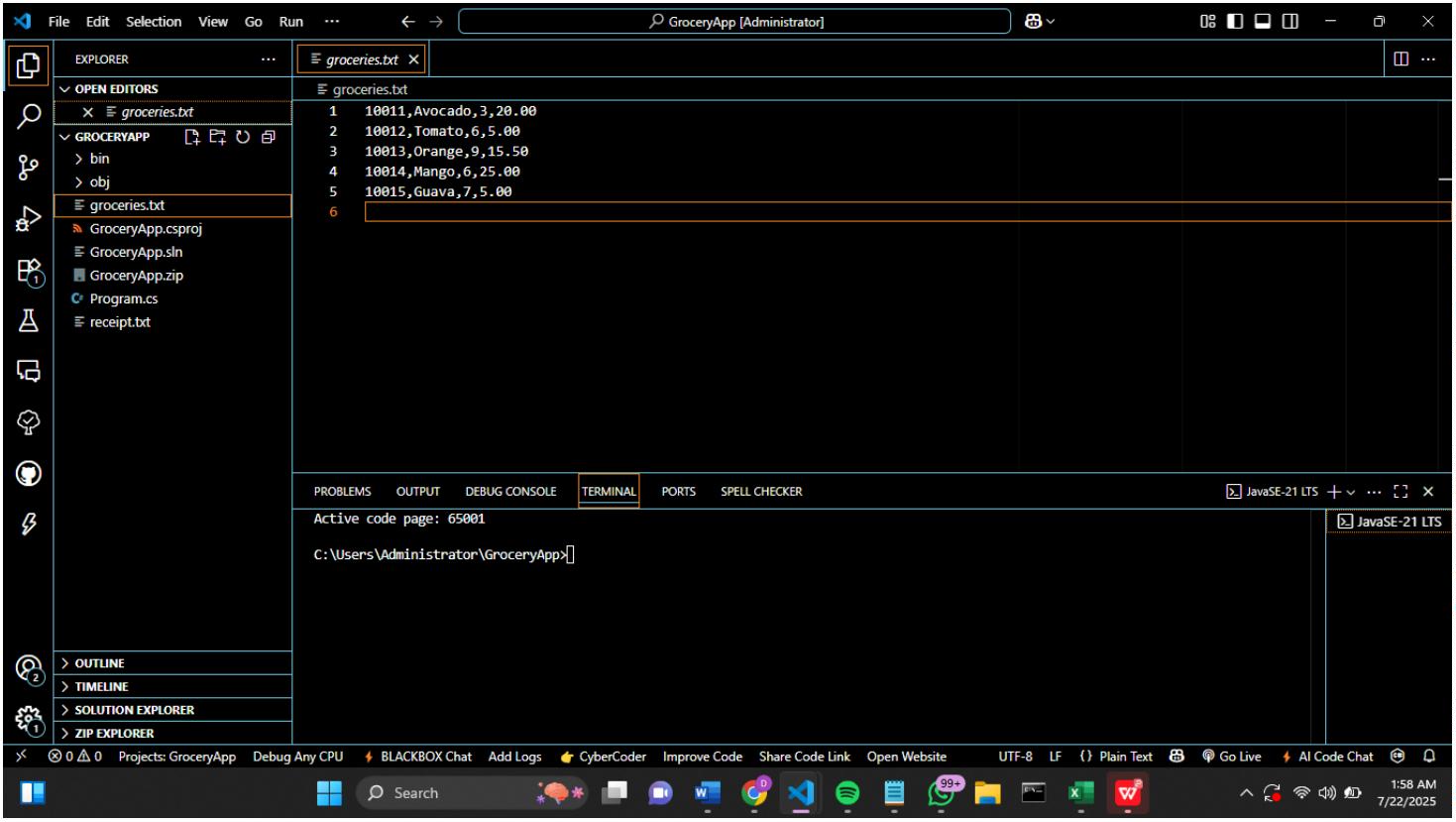
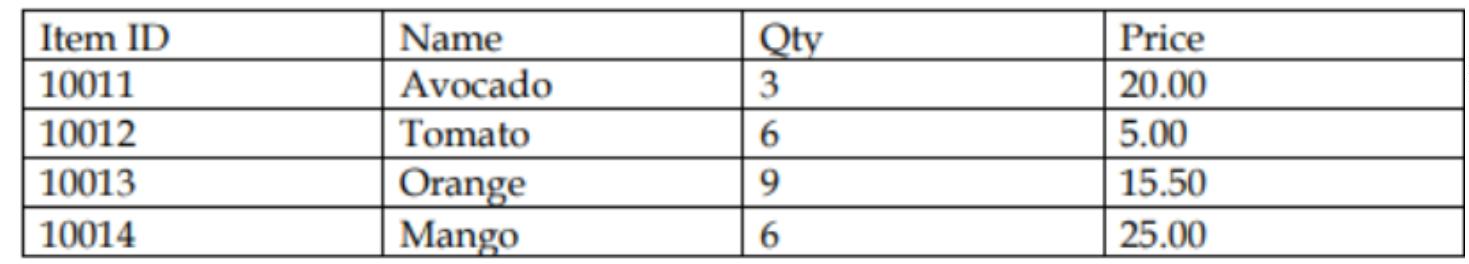
**Step 3: Creaꢀng the Input File**

We created a ﬁle named groceries.txt inside the project folder with the informaꢀon below.

**Wriꢀng the Code**

We modiﬁed the Program.cs ﬁle to:

1. Ask the user for the input ﬁle.



2

3

4

5

6

. Read and process data from the ﬁle.

. Calculate the total for each item.

. Add a 16% VAT.

. Print a formaꢂed receipt.

. Write the same receipt to an output ﬁle.

7

8

9

1

1

1

1

1

1

1

1

1

1

2

2

2

2

2

2

2

2

2

2

3

3

3

3

3

. using System;

. using System.IO;

. using System.Collections.Generic;

0.

1.class Item

2.{

3.

4.

5.

6.

7.

8.}

9.

public string ID;

public string Name;

public int Quantity;

public double Price;

public double Total => Quantity \* Price;

0.class Program

1.{

2.

3.

4.

5.

6.

7.

8.

9.

0.

1.

2.

3.

4.

static double VAT\_RATE = 0.16;

static void Main()

{

Console.WriteLine("====== Grocery Receipt Generator ======");

// 1. Ask for input file

Console.Write("Enter input file name (e.g. groceries.txt): ");

string fileName = Console.ReadLine();

if (!File.Exists(fileName))

{

Console.WriteLine("File not found. Please make sure the file

exists.");

3

3

3

3

3

4

4

5.

6.

7.

8.

9.

0.

1.

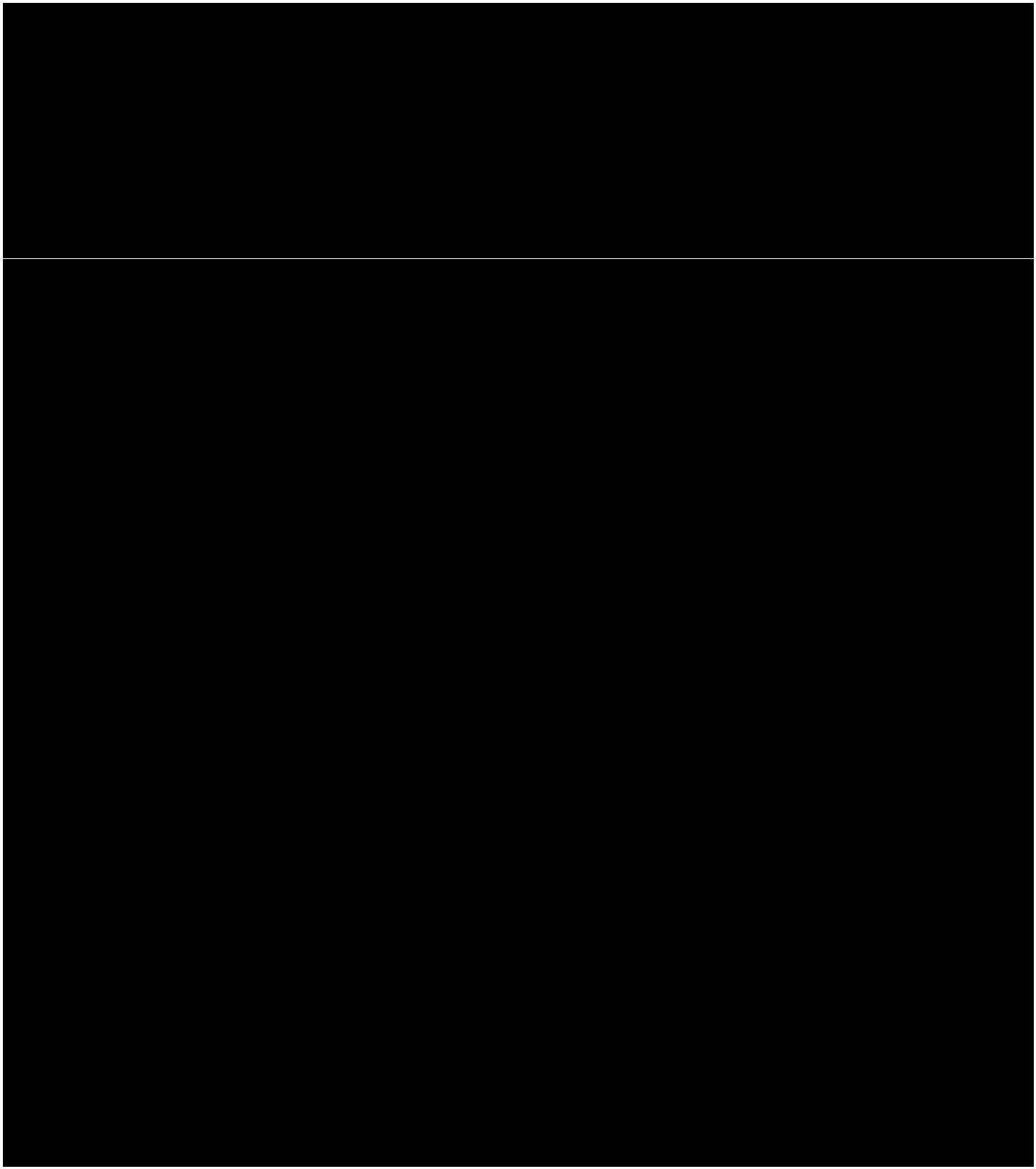
return;

}

// 2. Read and process data

List<Item> items = ReadItemsFromFile(fileName);

if (items.Count == 0)



4

4

4

4

4

4

4

4

5

5

5

5

5

5

5

5

5

5

6

6

6

6

6

6

6

6

6

6

7

7

7

7

7

7

7

7

7

7

8

8

8

8

8

8

8

2.

3.

4.

5.

6.

7.

8.

9.

0.

1.

2.

3.

4.

5.

6.

7.

8.

9.

0.

1.

2.

3.

4.

5.

6.

7.

8.

9.

0.

1.

2.

3.

4.

5.

6.

7.

8.

9.

0.

1.

2.

3.

4.

5.

6.

{

}

Console.WriteLine("No items found in file.");

return;

// 3. Calculate totals

double subTotal = 0;

foreach (Item item in items)

{

subTotal += item.Total;

}

double tax = subTotal \* VAT\_RATE;

double grandTotal = subTotal + tax;

// 4. Print receipt

PrintReceipt(items, subTotal, tax, grandTotal);

// 5. Write to file

WriteReceiptToFile(items, subTotal, tax, grandTotal);

Console.WriteLine("\nReceipt written to 'receipt.txt'.");

}

static List<Item> ReadItemsFromFile(string fileName)

{

List<Item> itemList = new List<Item>();

string[] lines = File.ReadAllLines(fileName);

foreach (string line in lines)

{

string[] parts = line.Split(',');

if (parts.Length == 4)

{

Item item = new Item();

item.ID = parts[0];

item.Name = parts[1];

item.Quantity = int.Parse(parts[2]);

item.Price = double.Parse(parts[3]);

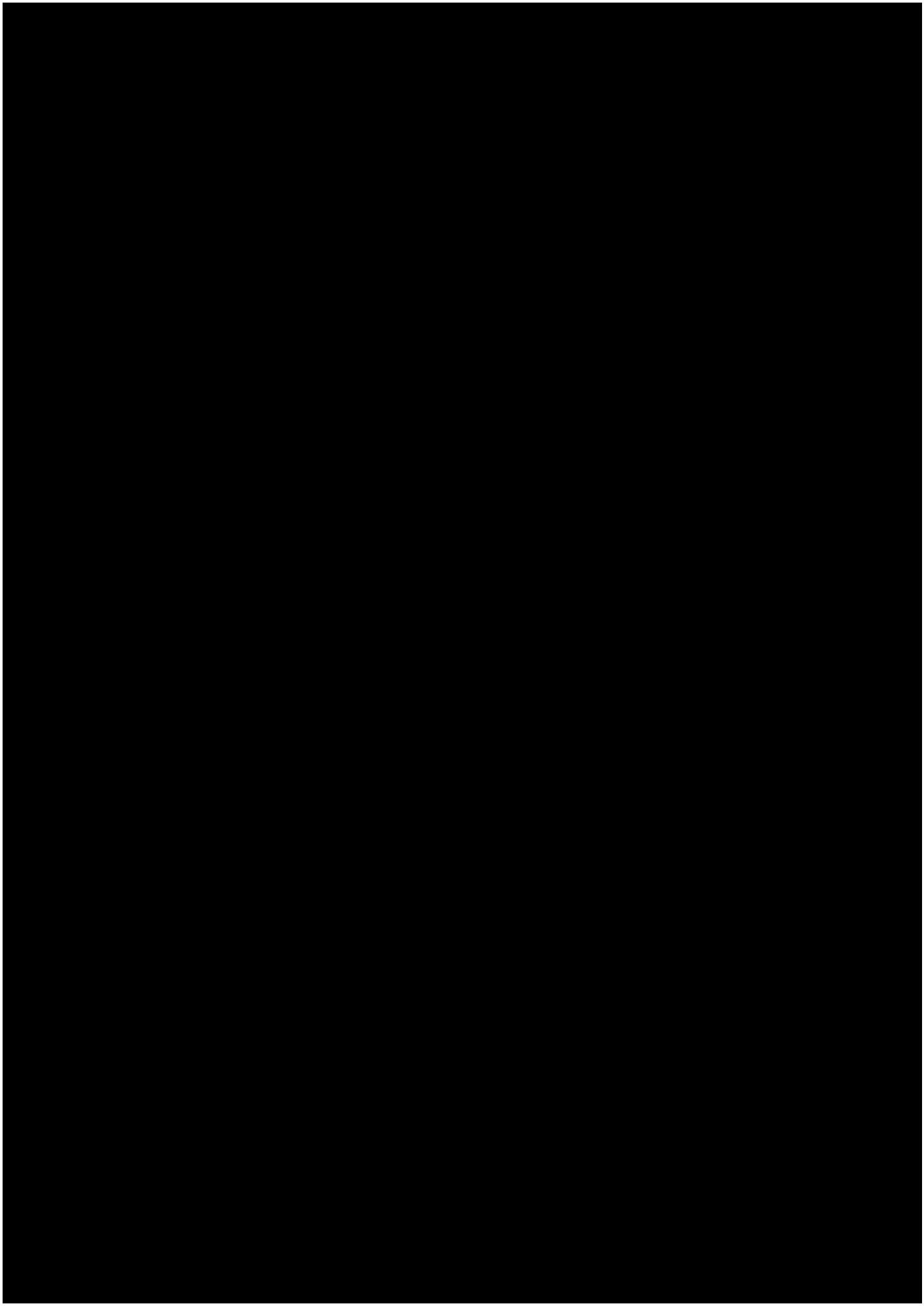
itemList.Add(item);

}

}

return itemList;

}



8

8

7.

8.

static void PrintReceipt(List<Item> items, double subTotal, double

tax, double grandTotal)

8

9

9

9.

0.

1.

{

Console.WriteLine("\n---------- Shopping Receipt ----------");

Console.WriteLine("{0,-10}{1,-10}{2,8}{3,10}{4,12}", "Item ID",

"

Name", "Qty", "Price", "Total");

9

9

9

9

2.

3.

4.

5.

foreach (Item item in items)

{

Console.WriteLine("{0,-10}{1,-10}{2,8}{3,10:F2}{4,12:F2}",

item.ID, item.Name, item.Quantity, item.Price, item.Total);

}

9

9

9

9

1

1

1

1

1

6.

7.

8.

9.

Console.WriteLine("---------------------------------------");

Console.WriteLine($"Sub-total:

Console.WriteLine($"VAT (16%):

KES {subTotal:F2}");

KES {tax:F2}");

00.

01.

02.

03.

04.

Console.WriteLine($"Grand Total:

KES {grandTotal:F2}");

}

static void WriteReceiptToFile(List<Item> items, double

subTotal, double tax, double grandTotal)

1

1

05.

06.

{

using (StreamWriter writer = new

StreamWriter("receipt.txt"))

1

1

07.

08.

"

{

writer.WriteLine("---------- Shopping Receipt ----------

);

1

09.

"

writer.WriteLine("{0,-10}{1,-10}{2,8}{3,10}{4,12}",

Item ID", "Name", "Qty", "Price", "Total");

1

1

1

1

10.

11.

12.

13.

1

foreach (Item item in items)

{

writer.WriteLine("{0,-10}{1,-

0}{2,8}{3,10:F2}{4,12:F2}", item.ID, item.Name, item.Quantity,

item.Price, item.Total);

1

1

1

14.

}

15.

16.

-

writer.WriteLine("--------------------------------------

");

1

1

1

17.

18.

19.

{

writer.WriteLine($"Sub-total:

writer.WriteLine($"VAT (16%):

writer.WriteLine($"Grand Total:

KES {subTotal:F2}");

KES {tax:F2}");

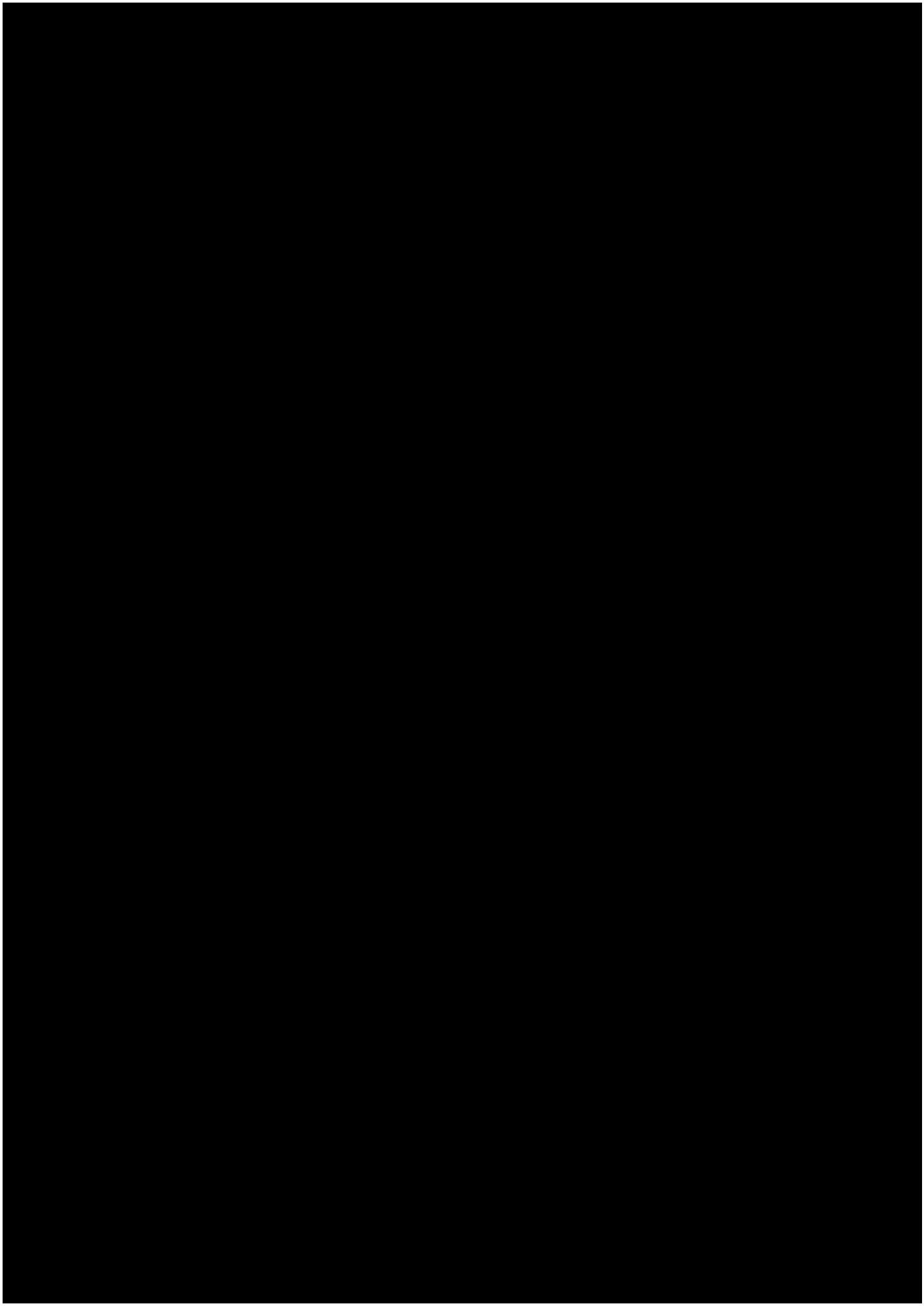
KES

grandTotal:F2}");

}

1

20.



1

1

1

21.

22.

23.

}

}

**Running the Applicaꢀon**

We ran the applicaꢀon using:

dotnet run

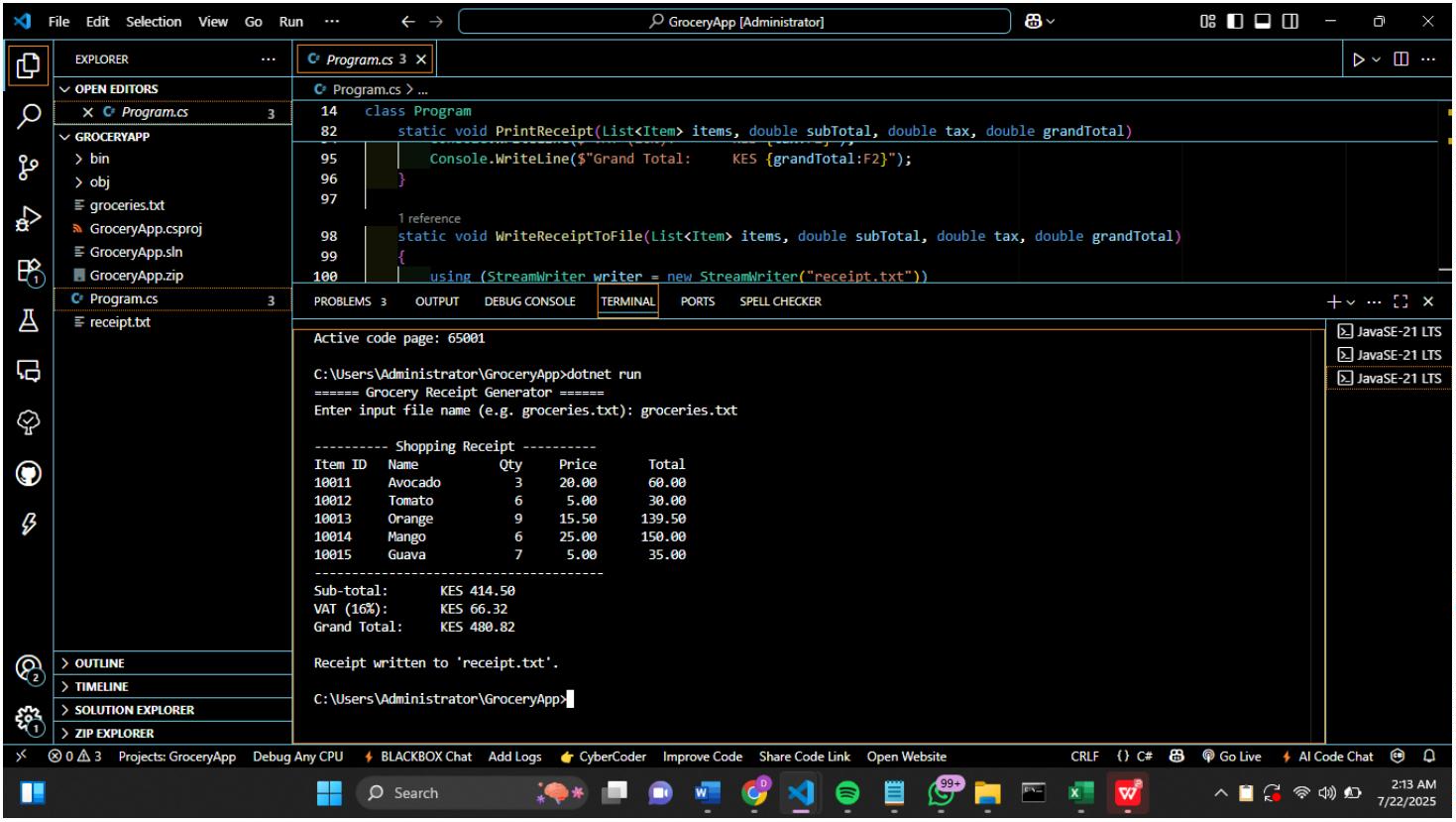
We entered the input ﬁle name, and the app read the contents, did the calculaꢀons, and

printed a receipt to the console.

**Viewing the Output File**

The same receipt was also saved to receipt.txt.

This ﬁle was located in the same folder as the applicaꢀon.



**Conclusion**

This was a simple grocery receipt applicaꢀon developed using basic C# concepts such as ﬁle I/O,

loops, lists, and user input. It helped me understand how to manipulate data from ﬁles and

output both to screen and ﬁles using console applicaꢀons.

