Sivan Moodley

ST10084225

PROG6221 POE

Read ME

Introduction

I was tasked with designing and developing a budget planning console app for a friend at campus, who needs assistance planning their spending for the month. The app requires the following information from the user:

- Their gross monthly income before deductions.
- Their estimated taxes for the month.
- Their estimated grocery expenses.
- Their estimated water and lights expenses.
- Their estimated travel cost (including fuel) expenses.
- Their cell phone and telephone billing expenses.
- Any other expenses they might have for the month.

Along with this the app needs to know if the user will be renting accommodation or buying a property with a home loan. if their renting, it only requires the monthly rental fee, however,

if their buying property however, the app will require the following information from the user:

- The purchase price of the house.
- The total deposit.
- The annual interest rate of the loan.
- The number of months given to repay the loan.

After this the app will require the user too choose between buying a car or continuing with their budget plan. If the user chooses to buy a car, they will need to enter the following information:

- The model and make of the car.
- The purchase price of the car.

- The total deposit.
- The interest rate.
- The estimated insurance premium.

Using this information, the app will produce the following:

1.) A budget report containing:

- Their gross salary before tax.

- Their estimated taxes for the month.

- Their total income after taxes.

- A list of their monthly expenses containing:

- Their estimated grocery expenses for the month.

- Their estimated water and lights expenses for the month.

- Their estimated travel costs (including fuel) expenses for the month.

- Their estimated cell phone and telephone billing expenses for the month.

- Any other expenses they might have for the month.

- The rental fee of their accommodation for the month.

- Their estimated home loan repayment fee for the month.

- Their estimated car repayment fee for the month.

- The amount of their salary is left over after their monthly expenses.

2.) Their gross salary and the values they've estimated for their home loan, the app will give an estimate as too whether the user is likely to be approved for the loan or not, if the users monthly home loan repayment cost more than a third of their gross salary, the app will notify the user that the home loan is not likely to be approved and if they've used more than 75% of their salary for the month.

Software Specs

1.) Visual Studio 2022 Version 17.1.0

2.) Sublime Text Stable Channel, Build 4126

3.) GitHub Version 3.5.0

4.) Microsoft Office Pro 2019

5.) Adobe Photoshop 2022

F.A.Q

1.) What information will I need to use this app?

All you'll need is your:

- Gross monthly income before deductions.
- Estimated taxes for the month.
- Estimated grocery expenses for the month.
- Estimated water and lights expenses for the month.
- Estimated travel cost (including fuel) expenses for the month.
- Estimated cell phone and telephone billing expenses for the month.
- Any other expenses you might have for the month.
- If you are renting, the app requires the monthly rental fee.
- If you are buying property, the app will require, the purchase price of the house, the total deposit, the annual interest rate of the loan, and the number of months given to repay the loan.

-If you want to buy a car, you will require, the make and model of the car the purchase price of the car, the total deposit, the interest rate of the loan, and the number of months given to repay the loan.

2.) What number notation should I use while entering the information?

- You can use comma notation (0,00) or you can use full stop notation (0.00)

3.) How does the app estimate whether my home loan is likely to be approved?

- The app simply checks if your monthly home loan repayments cost more than a third of your gross salary, if so, the app will notify you that your home loan is unlikely to be approved.

4.) What do I do if I enter invalid information?

- If you enter any invalid information, the app will recognize this and prompt you to enter the information again.

5.) What do I do if I don't know the estimated value of one of my expenses?

- Simply enter the value as 0 for the time being. You can always come back to the budget planner again once you calculate the correct amount.

System Requirements

- Windows 7 or newer / Linux(Debian x64 or Arch x64).
- Intel Celeron 2000 series or newer/ AMD Athlon 2000 series or newer.
- 1 GB RAM.
- 250 MB HDD storage minimum.
- Keyboard and Mouse.

Developer Info

Name: Sivan Moodley

Student number: ST10084225

Cell phone number: 0815750712

Email: sivan.moodley02@gmail.com

System Used For Testing

CPU: AMD Ryzen 7 3700X

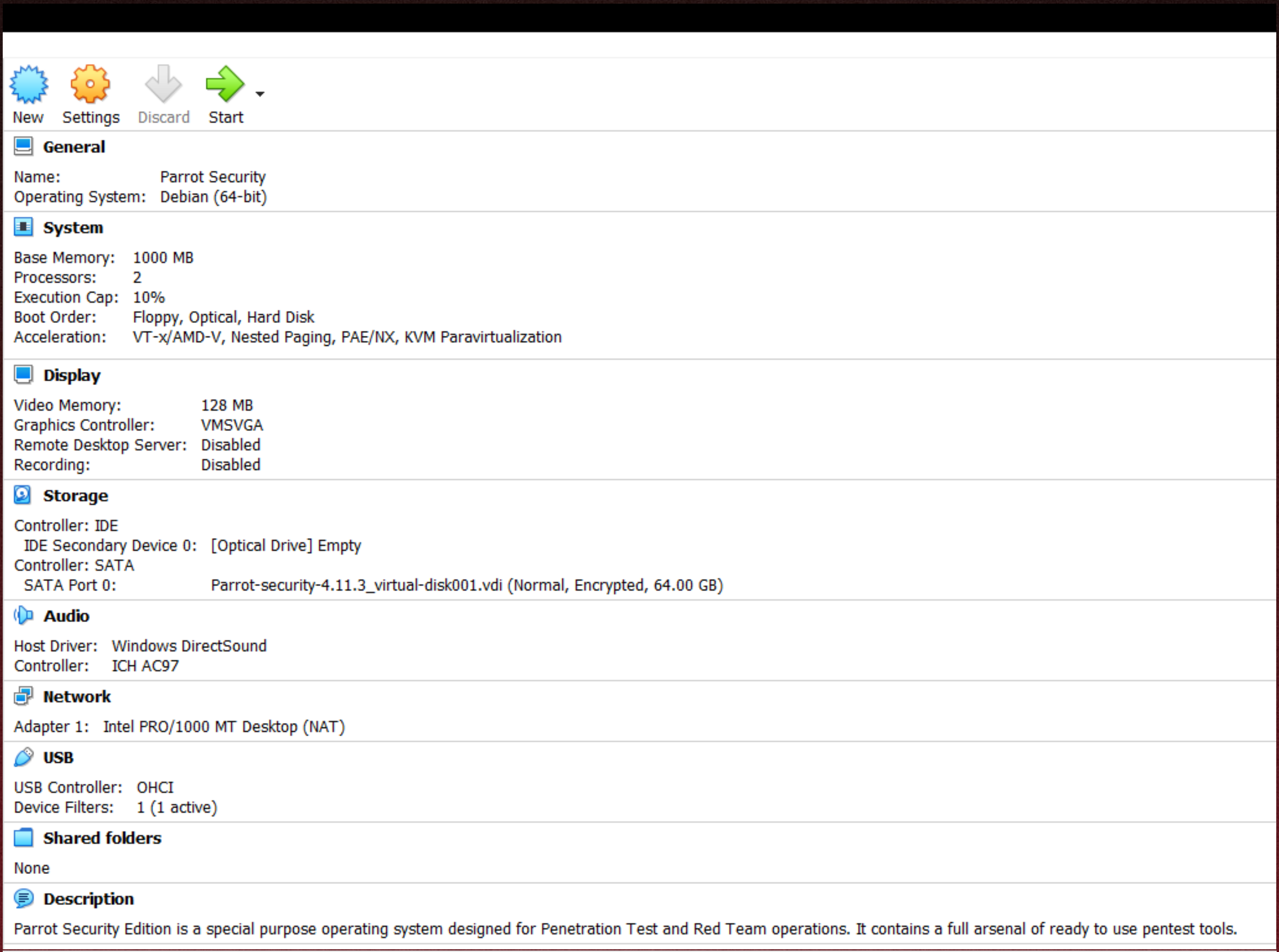
GPU: NVIDIA GeForce RTX 2060 Super

RAM: 16GB 2666Mhz DDR4

Storage: 500GB Nvme m.2 SSD

Testing Process For System Requirements

I created a virtual machine with the following specs:



General

Name: Parrot Security
Operating System: Debian (64-bit)

System

Base Memory: 1000 MB
Processors: 2
Execution Cap: 10%
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Display

Video Memory: 128 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Parrot-security-4.11.3_virtual-disk001.vdi (Normal, Encrypted, 64.00 GB)

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB

USB Controller: OHCI
Device Filters: 1 (1 active)

Shared folders

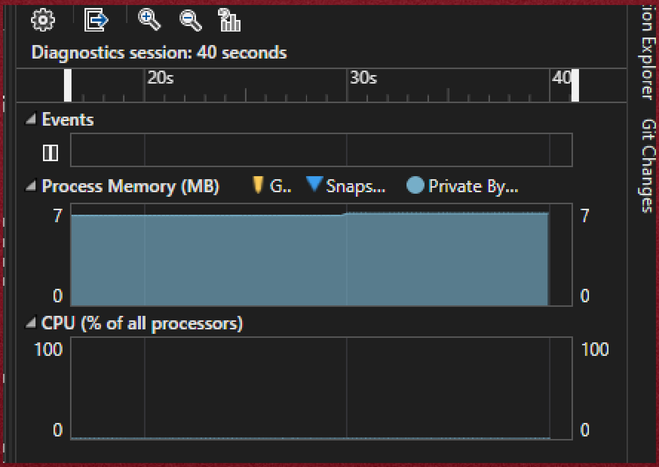
None

Description

Parrot Security Edition is a special purpose operating system designed for Penetration Test and Red Team operations. It contains a full arsenal of ready to use pentest tools.

Then ran the program on a Linux distribution, this was the lowest I could bring the specs too without it beginning to affect the program.

I chose Linux because it provides the least bloated OS, so the performance stats I captured were as accurate as possible with the hardware I have on hand.



Code Attribution

1.) How To Use Abstract Classes: https://www.w3schools.com/cs/cs_abstract.php (Accessed: 05/09/2022)

2.) How To Create and Read and Write from Array Lists: https://www.tutorialsteacher.com/csharp/csharp-arraylist (Accessed: 05/09/2022)

3.) Efficient While Loops: https://www.w3schools.com/cs/cs_while_loop.php (Accessed: 05/09/2022)

4.) Efficient For Loops: https://www.w3schools.com/cs/cs_for_loop.php (Accessed: 05/10/2022)

5.) Multithreading and Making Threads Sleep: https://docs.microsoft.com/en-us/dotnet/api/system.threading.thread.sleep?view=net-6.0 (Accessed: 05/10/2022)

6.) C# Delegates:

https://www.geeksforgeeks.org/c-sharp-delegates (Accessed: 06/01/2022)

7.) C# Exception Handling Best Practices:

https://stackify.com/csharp-exception-handling-best-practices/#:~:text=The%20C%23%20try%20and%20catch,error%2C%20or%20crash%20the%20application. (Accessed: 06/01/2022)

GitHub

1.) Repository: https://github.com/Noisyninja703/PROG6221_POE_Part1_ST10084225/tree/PROG6211_POE_Part_2

2.) Kanban: https://github.com/Noisyninja703/PROG6221_POE_Part1_ST10084225/projects/3

NOTE: There is a second branch named PROG6221_POE_Part_2

This is where the files for my part 2 are stored.

ASCII Dev Tag

Developed By
Sivan Moodley
ST10084225