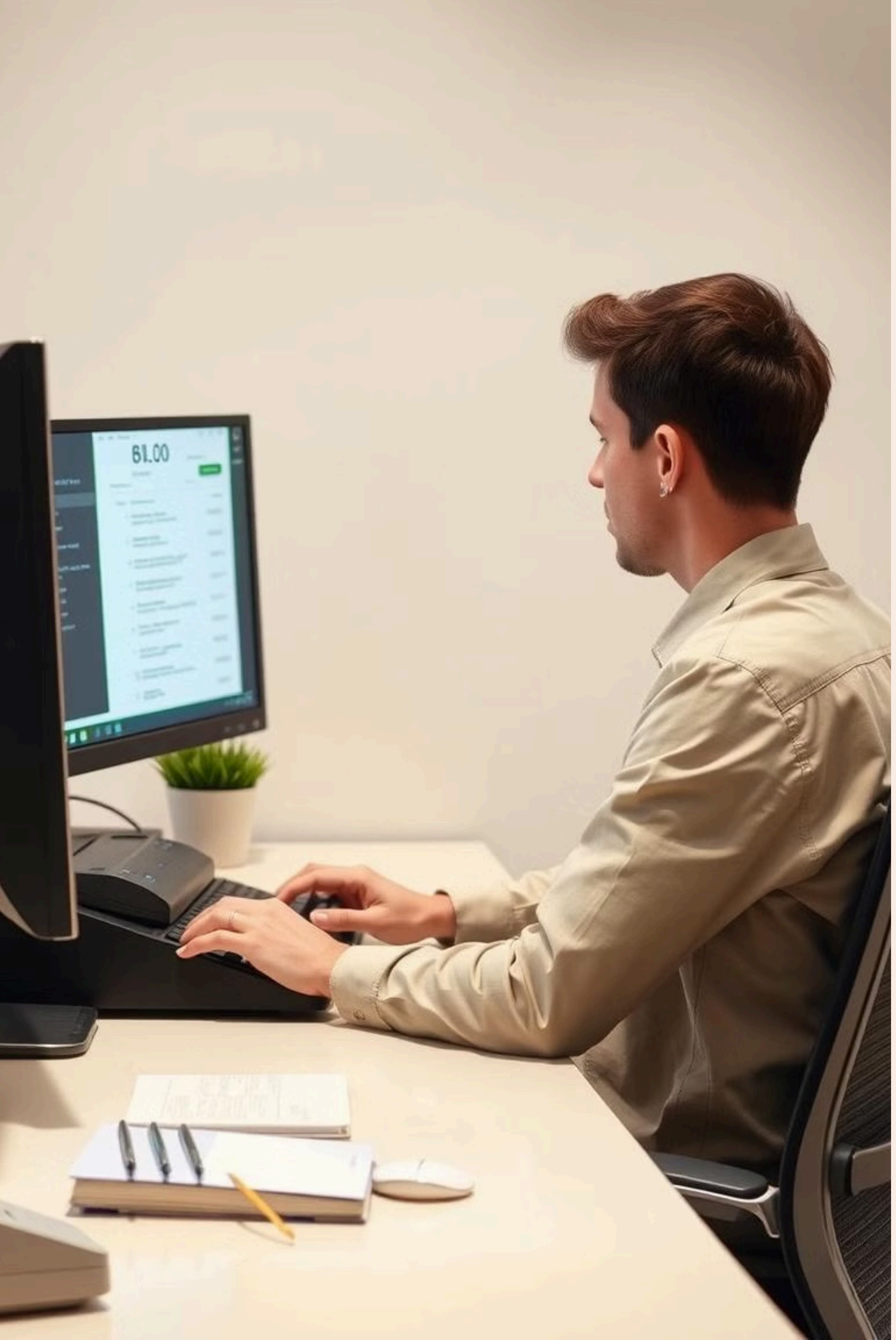


Linux Project: Bill Calculator Codes

This presentation will guide you through the development of a bill calculator project using Linux. This project will demonstrate basic Linux programming concepts and offer practical applications for managing finances.

by Sneha Dhole

Introduction to the Project

Project Overview

This project aims to create a bill calculator program using the Linux environment, which will enable users to calculate the total cost of their bills.

Programming Language

We will be using Python as the programming language for its simplicity and versatility.

Understanding the Requirements

1 **User Input**

The bill calculator must accept user input for the bill amount, including the ability to enter multiple items and their respective costs.

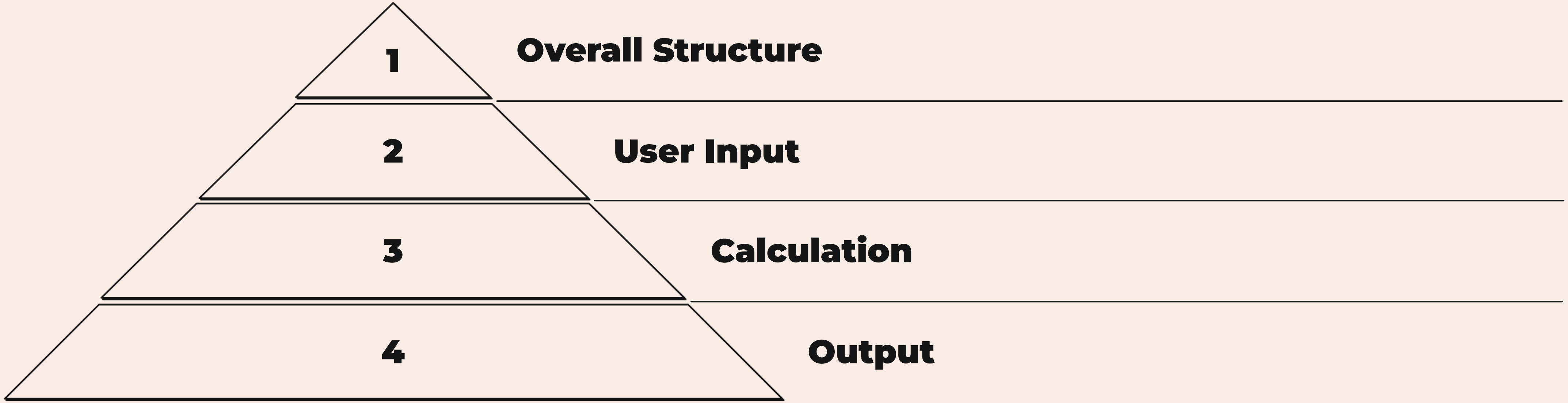
2 **Calculation**

The program should calculate the total bill amount based on the inputted items and their prices.

3 **Output**

The calculator must display the calculated total bill amount to the user.

Designing the Bill Calculator



1 Overall Structure

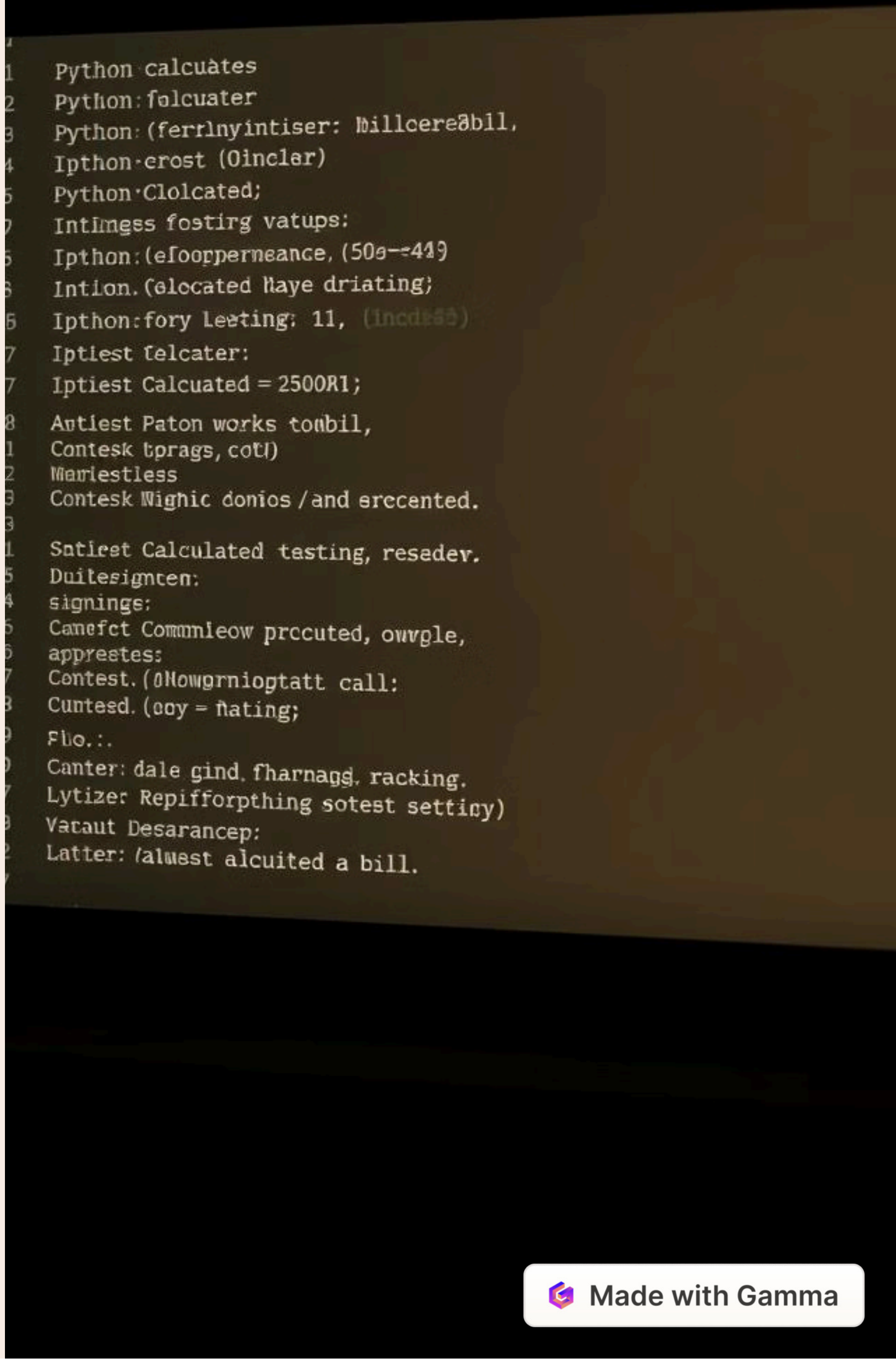
2 User Input

3 Calculation

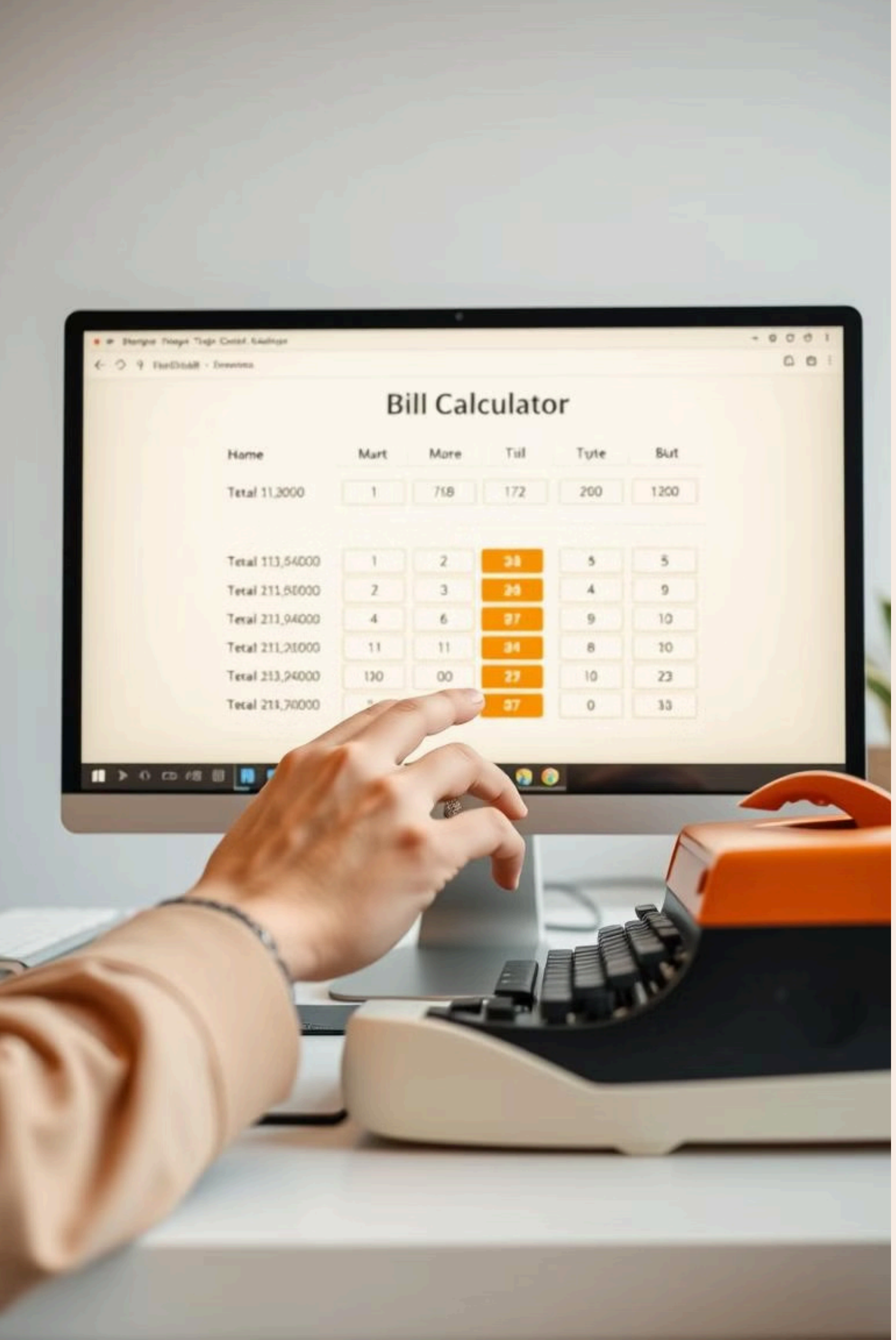
4 Output

Implementing the Bill Calculator

```
# Import necessary modules
import sys

# Function to calculate the total bill amount
def calculate_bill():
  total = 0
  while True:
    item = input("Enter item (or 'done' to finish): ")
    if item.lower() == "done":
      break
    price = float(input("Enter price: "))
    total += price
  return total

# Main function
if __name__ == "__main__":
  bill_amount = calculate_bill()
  print(f"Total bill amount: {bill_amount}")
```



```
Python calcuàtes
Python: fəlcuater
Python: (ferrlnyintiser: Millcereǝbil.
Ipthon·crost (0inclər)
Python·Clolcated;
Intimgss fostirg vatups:
Ipthon: (eſoorpermeance, (50ɘ-=449
Intlon. (əlocated Maye driating;
Ipthon: fory Leeting: 11, (incdɘss)
Iptiest ſelcater:
Iptiest Calcuated = 2500R1;
Antiest Paton works tombil,
Contesk tprags, cotl)
Mariestless
Contesk Mighic donios / and erccented.

Sntirst Calculated tasting, resedev.
Duitesigmten:
signings:
Canefct Commnleow prccuted, owrgle,
apprestes:
Contest. (0Mowgrniogtatt call:
Cuntesd. (ooy = ñating;
Flio.:.
Canter: dale gind. fharnagg. racking.
Lytizer Repifforpthing sotest settiny)
Vacaut Desarancep:
Latter: /almest alcuited a bill.
```



Testing the Bill Calculator

Unit Tests

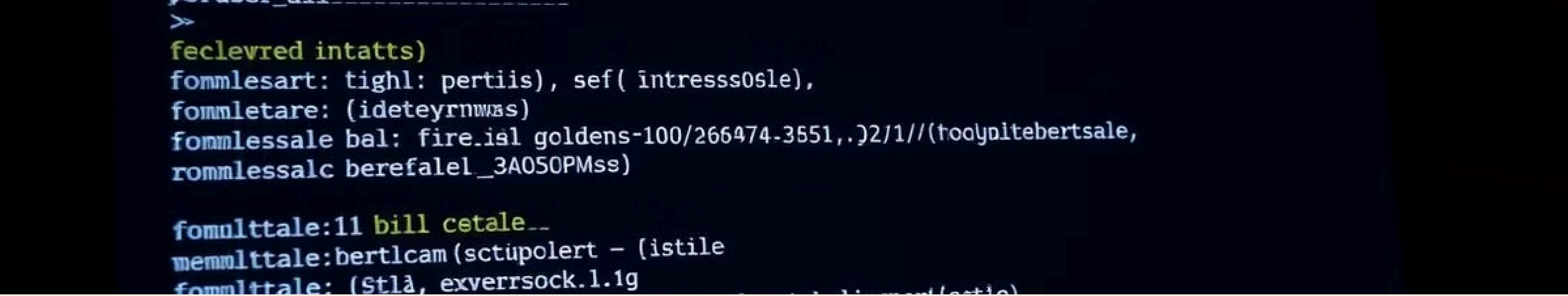
Test individual functions of the calculator to ensure they perform as expected.

Integration Tests

Test the entire bill calculator program to confirm it functions correctly with various user inputs.

Optimizing the Bill Calculator

1. **Efficiency**
2. **Error Handling**
3. **User Interface**



```
>>
feclevred intatts)
fommlesart: tighl: pertiis), sef( intresssOsle),
fommletare: (ideteyrnwss)
fommlessale bal: fire.isl goldens-100/266474-3551,.)2/1//(tooyaltebertsale,
rommlessalc berefalel_3AOSOPMss)

fomnlttale:11 bill cetale..
memmlttale:bertlcam(sctupolert – (istile
fommlttale: (Stlå, exverrsock.1.1g
```

Deploying the Bill Calculator

1

Packaging

Create a package for easy distribution and installation.

2

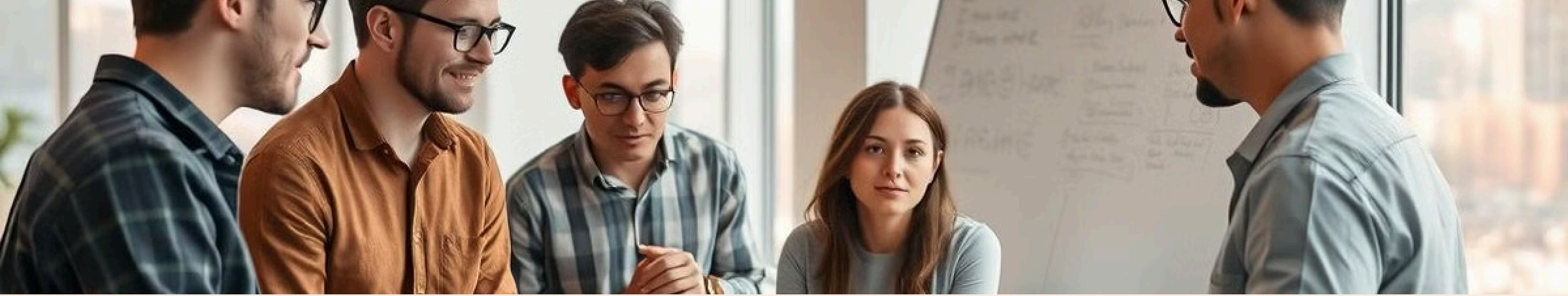
Installation

Install the bill calculator on the target system.

3

Execution

Run the bill calculator application on the target system.



Conclusion and Next Steps

This project has demonstrated the development process of a bill calculator using Linux. As next steps, consider expanding the functionality to include features like saving bills, generating reports, or integrating with online payment systems.