LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

CS4051NI Fundamentals of Computing

60% Individual Coursework

2023/24 Spring

Student Name: Nikita Bhandari

London Met ID: 23047392

College ID: NP01NT4A23092

Assignment Due Date: Tuesday, July 23, 2024

Assignment Submission Date: Sunday, July 28, 2024

Word Count: 242

Project File Links:

YouTube Link:	Keep Unlisted YouTube URL of your Project Here
Google Drive Link:	Keep Google Drive URL of your Project Here with Anyone in Organization can View Option Enabled

I confirm that I understand my coursework needs to be submitted online via MySecondTeacher under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and marks of zero will be awarded.

Table of Contents

List of Figures

Table of Figures

1. Introduction

The individual coursework of the module Fundamentals of Computing is all about creating a Land Rental System for a private company named Techno Property Nepal. This company has lands in various locations around Nepal where the clients can rent the land according to their needs and desires.

In this coursework, the files have been properly divided into 4 parts i.e. read.py, write.py, main.py, and operation.py. The main.py is the entry point where the designing part is kept, and lenders are provided with different options if they want to rent, return, or exit. The read.py file reads the text file named “Coursework.txt” and here a 2d list is also created. The operation file has two functions, “rent_land ()” and “return_land ()” where operations are done. The write.py file also has two functions i.e. “write_invoice” and “write_return_invoice”. An invoice is generated for renting the land as well as in returning the land both in the console and to the respected .txt file. In this report when the customer can’t return the land in the given time certain fines are applied according to the months taken. If the customer returns the land before the month, there will be no fine, but if the land returned month exceeds the actual month, the fine is imposed.

1.1 Goals and Objectives

The goal of this coursework is:

- Develop a user-friendly system where the available lands are displayed.
- Provide different kitta, City, direction, anna, and price so the borrower can have multiple choices while renting.
- Provide an invoice after the land is rented and after the land is returned and vice versa which helps in accurate record keeping.
- Automatically update the status of land i.e. when land is rented change the “available” to “not available” and when the land is returned change “not available” to “available”.

- Implement security measures to ensure compliance with regulatory requirements for land rental transactions and safeguard sensitive financial information.

2. Algorithm

An algorithm is a simple step-by-step for problem solution.

Step 1: Run the program.
Step 2: Display details of the company and a welcome message from Techno Property.
Step 3: Read the data from the coursework.txt file and show details,
(kitta, address, direction, anna, price, and availability)
Step 4: Start a loop for the main menu.
Step 5: Inside, display various options to the user:
Option 1: Rent land.
Option 2:Return land and show the table of available and non-available.
Option 3: Exit the process.
Step 6: Inquire the user about their option.
Step 7: If the user chooses option 1:
Request the user their details (such as name, number, kitta number, anna and duration)

If the user number is more or less than 10 digits or string, ask the user to enter 10 digits number only and numeric value.

If the user's kitta number is not valid inform the user and return otherwise proceed to the next step.

If the anna is not available ask them to enter all the available and correct anna.

Request that the user enter the number of months.

If all conditions are correct calculate the total cost based on price and anna.
Provide the invoice in the console and the text file.
Change the availability of the rented land to “Not Available”.

Ask the customer for consent to proceed:

If so, carry out step one again and create an invoice with `aggregate_ total in single txt file.

If the option is no, exit from the function and return.

Step 8: If the user chooses option 2:

Ask the user their details (such as name, number, kitta number of lands they want to return, months taken, and months returned)

If the entered kitta number from the customer is not valid or not rented inform the user and return otherwise proceed to the next step.

If all the conditions are correct calculate the sum of the total price and then calculate the fine accordingly.

Provide the invoice in the console and also the text file.

Change the availability status of the rented land to “Available”.

Print a message about updated availability.

Ask the customer if they want to continue the process:

If yes, then simply repeat the process.

If no, then end the function and return.

Step 9: If the user chooses option 3:

Print a message that the program is exited.

Step:10 If the customer inputs any invalid option or any negative value then, display a message and go back to step 4.

3. Flowchart

A flowchart is a graphical representation of a process or algorithm that uses different shapes and symbols to depict the steps involved in completing a task or solving a problem.

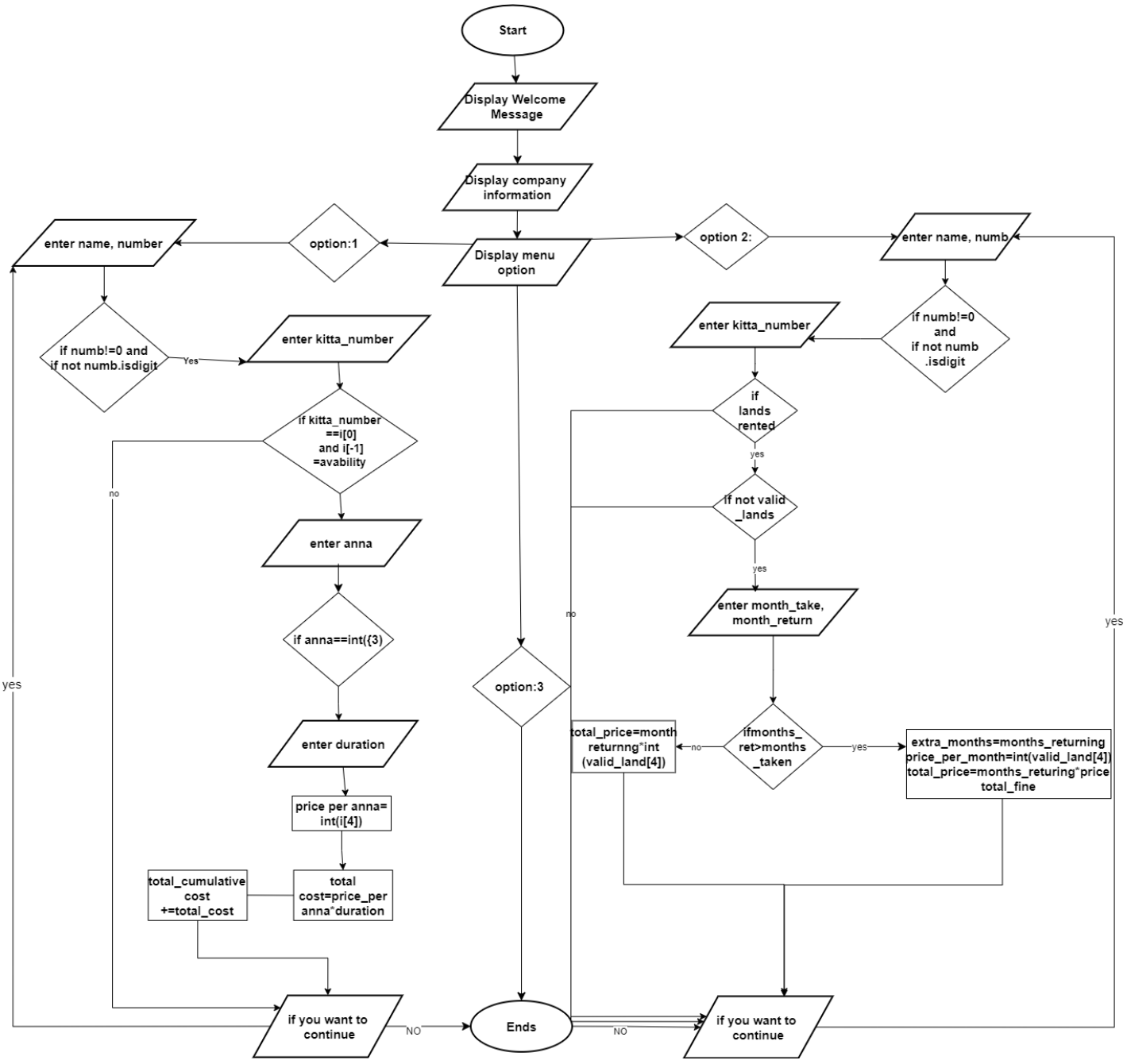


Figure 1: Diagram Of Flowchart

4. Pseudocode

4.1 For read_file.py()

BEGIN

Open “Coursework.txt” file **in** read mode **and** assign it **to** variable f

Initialize an **empty** list temp

FOR EACH line **IN** file f **THEN**

Remove the newline character **from** line **and** assign it **to** variable ff

Split the line **by** comma **as** separator

Add list to temp

END FOR

RETURN temp

END opening **FUNCTION**

END

4.2 For write.py()

Import datetime module

BEGIN

Def **FUNCTION** write_invoice(name, numb, kitta_number, anna, duration, total_cost, rented_date):

```
    Display("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    Display("\t\tInvoice:")
    Display("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    Display("Time of the rent: Rs.", rented_date)
    Display("Name of the borrower:", name)
    Display("Number of the borrower:", numb)
    Display("Taken Kitta Number:", kitta_number)
    Display("Number of Anna taken:", anna)
    Display("Duration (in months):", duration)
    Display("Total Cost: Rs.", total_cost)
    Display("Invoice has been generated. Thanks for choosing us")
    Display("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

    invoice_date = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

    with open("invoice.txt", "w") as file:
        file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
        file.write("\tInvoice is generated here:\n")
        file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
        file.write("Rented invoice date:" + rented_date + "\n")
        file.write("Name of the borrower: " + name + "\n")
        file.write("Number of the borrower:"+ str(numb) + "\n")
        file.write("Taken Kitta Number: " + kitta_number + "\n")
```

```
        file.write("Anna taken: " + str(anna) + "\n")
        file.write("Duration (in months): " + str(duration) + "\n")
        file.write("Total Cost: Rs. " + str(total_cost) + "\n")
        file.write("~~~~~~~~~~~~Thanks for choosing us~~~~~~~~~~~~~~~\n")
        file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
```

Def **FUNCTION** write_return_invoice(name, numb, kitta_number, months_taken, months_returning, total_fine, total_amount,returned_date):

```
    Display("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    Display("\t\tReturn Invoice:")
    Display("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    Display("Time of the rent returned: Rs.", returned_date)
    Display("Name:", name)
    Display("Number of the borrower:", numb)
    Display("Kitta Number:", kitta_number)
    Display("Months Taken:", months_taken)
    Display("Months Returning For:", months_returning)
    Display("Total Fine: Rs.", total_fine)
    Display("Total Amount: Rs.", total_amount)
    Display("Extra Months: ", months_returning - months_taken)
    Display("Return invoice has been generated successfully!")
    Display("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

    with open("return_invoice.txt", "w") as file:
        file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
        file.write("Return Invoice is generated here:\n")
        file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
        file.write("Rented invoice date:" + returned_date + "\n")
        file.write("Name: " + name + "\n")
        file.write("Number of the borrower:"+ str(numb) + "\n")
        file.write("Kitta Number: " + kitta_number + "\n")
```

```
    file.write("Months Taken: " + str(months_taken) + "\n")
    file.write("Months Returning For: " + str(months_returning) + "\n")
    file.write("Total Fine: Rs. " + str(total_fine) + "\n")
    file.write("Total Amount: Rs. " + str(total_amount) + "\n")
    file.write("Extra Months: " + str(months_returning - months_taken) + "\n")
    file.write("~~~~~~~~~~~~Thanks for choosing us~~~~~~~~~~~~~~~\n")
    file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
```

END

4.3 For operation.py()

BEGIN

```
# Import necessary functions and modules
from write_file import write_invoice, write_return_invoice
from read_file import opening
import datetime

def rent_land():
    Display rental process instructions
    Get customer name
    Get customer phone number (10 digits)
    Validate phone number format
    Read land data from file
    Read rented lands from file

    # Loop for renting lands
    While customer wants to rent land
        Get kitta number from customer
        Find land data with matching kitta number and availability "Available"
        If land found
            Get the desired anna
            Get rental duration in months
            Record rented date
            Calculate total cost
            Update land availability
            Write invoice for renting
            Print success message
            Ask user if they want to continue renting

        ELSE:
```

DISPLAY error message

IF no, break out of the loop

Write updated land data to file

Return list of rented lands

def return_land():

Display return process instructions

Get customer name

Get customer phone number

Record return date

Read rented lands from file

IF no rented lands:

Print message and return

WHILE True:

Get kitta number to return

Check if the kitta number is valid and currently rented

IF valid and rented:

Get months taken and months returning

Calculate total amount and fine

Write return invoice

Update land availability

Remove kitta number from rented lands

Print success message

Print updated land availability

Ask user if they want to continue returning

```
        # If kitta number is invalid or not rented
        ELSE:
            Print error message


        If no, break out of the loop

IF __name__ == "__main__":
    Ask user for operation choice (rent or return)
    IF rent:
        Call rent_land function
    IF return:
        Call return_land function
END
```

4.4 main.py()

BEGIN

Import rent_land and return_land functions **from** operation **module**
Import opening **function from** read_file **module**

Def function design():
Print header design
End function design

Def **function** table():
Call opening **function to** retrieve data
Print table header
FOR each row in data **THEN**
IF row has enough elements **THEN**
Print formatted row data
END IF
END FOR
End function table

Call design **function**
Call table **function**

Def **function** main():
Print menu options
WHILE True THEN
Get user's option
IF option is greater than or equal to 0
DISPLAY invalid option message

```
            Continue
        IF option is 1 THEN
            Call rent_land function
        ELSE IF option is 2 THEN
            Call table function
            Call return_land function
        ELSE IF option is 3 THEN
            Print exit message
            Exit loop
        ELSE
            Print invalid option message
        END IF
    END WHILE
End function main

Call main function
END
```

5. Data Structure

A data structure is in which data are managed, stored, and organized in a computer that enables efficient access and manipulation. There are various types of data structure lists, tuples, dictionaries, sets, strings, arrays.

In my coursework, I have implemented 2d listsA 2D list, which is also called a two-dimensional array, is pretty popular in Python. It's like a table or matrix. To access elements in a 2D list, you need two indices: one for the row and one for the column.

```
File  Edit  Format  Run  Options  Window  Help
def opening():
    f = open("Coursework.txt", "r")
    temp = []
    for each in f:
        ff = each.replace("\n", "")
        temp.append(ff.split(","))
    return temp
```

Figure 2: Code for data structure

In the read_file.py, a 2D list named 'temp' has been employed to systematically store the data retrieved from the file "Coursework.txt". Each sublist within this data structure represents a distinct row of information, with individual values, originally separated by commas in the file, now represented by the elements within each sublist.I have also used the 2d list in operation.py named “land_data”.

In rent_land() the 2d list land_data is used for checking the availability status as it matches the kitta_number with the kitta_number in land_data. After that when the lands get rented the availability status is changed to “Not Available”. Once the land gets rented invoice is generated by retrieving the details from “land_data” like price and the duration.

In the return_land () the 2d list land_data is used to check the validity of the kitta_number if it's “Not Available” or not. The after that when the land is returned the availability status of the land is updated to “Available” within “land_data”. Then finally the 2d list here calculates the total amount that needs to be paid by the borrower. Here the information from the 2d list is uded like cost i.e. price per anna.

Finally, in the main.py 2d list named “data” is used in the function table() to display the land data from the file.

6. Program

In this part of the report, there will a description of the overall program and how the program is implemented will be shown.

6.1 Implementation

- **Read.py**

In this code, the contents of a file named "Coursework.txt" is read and convert into a 2D list. In this list, each sublist represents a row of data from the file, while each element within a sublist represents a value separated by commas.

- **Operation.py**

The "rent_land()" function allows users to rent land by providing their name, phone number, and land details. It validates the phone number, updates the rented land information, and generates an invoice.

The "return_land()" function facilitates the return of rented land, calculates fines, updates land availability, and generates a return invoice while also updating the files with the returned land information.

- **Write,py**

This part is for making invoices for renting and returning land. The write_invoice() function is for creating an invoice for renting land, and the write_return_invoice() function is used to make an invoice for returning rented land. These invoices have info like fines, total amount, and extra months.

- **Main.py**

This section provides a command-line interface with detailed info about Techno Property Nepal, such as the parcel number, address, directions, area, price, and availability.

6.2 Renting process

In this process when the user chooses option 1. They are first asked about the name and then the number. In the number, the user should only input a 10-digit number and it shouldn't be a string data type but rather an integer.

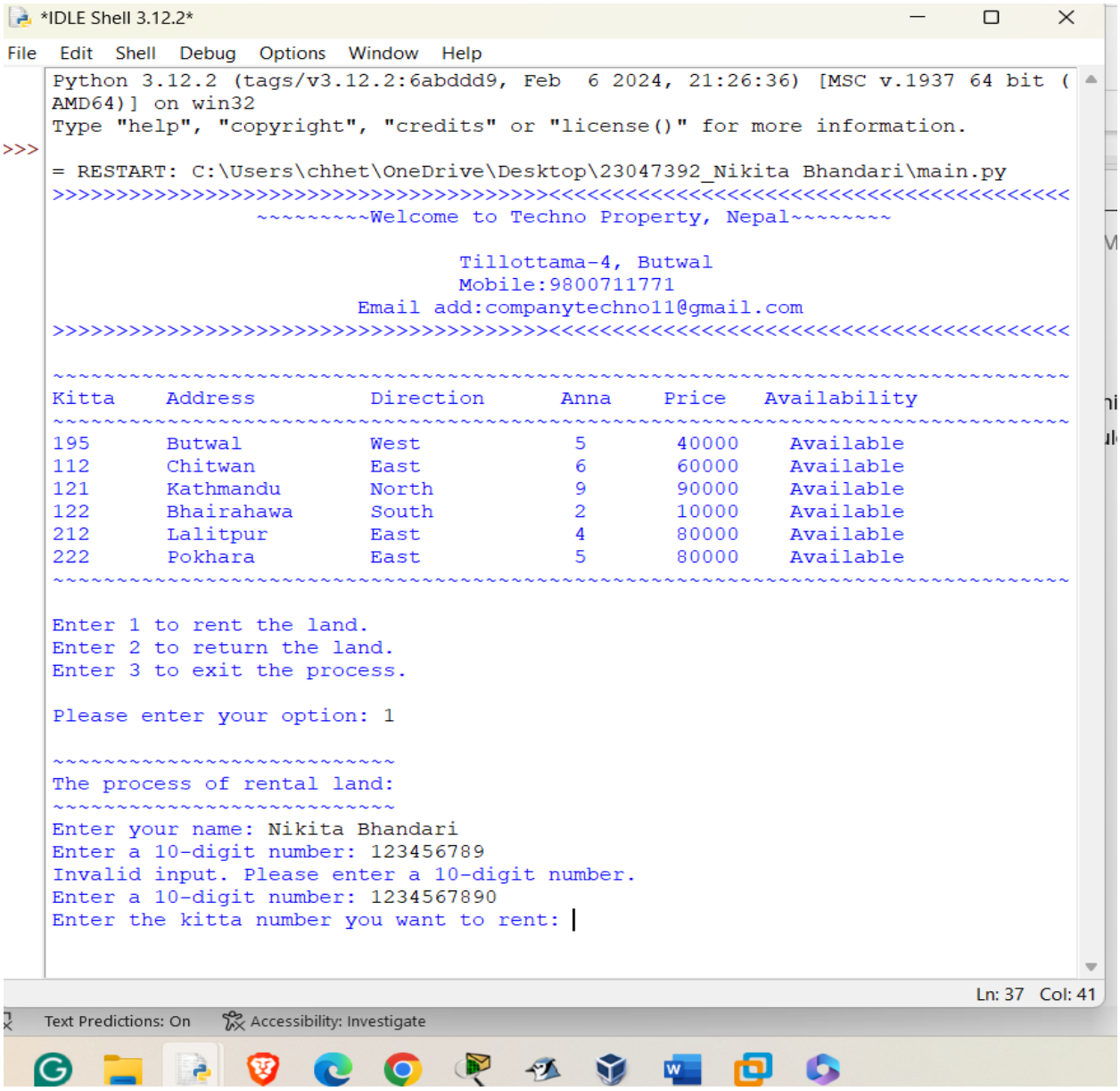


Figure 3: Implementation of number

Now, when the wrong kitta number is rented it says the kitta number is not found and asks user if they want to continue. If yes then Anna will be asked then if the wrong Anna is inputted then the user will have to input all the available Anna. When the correct kitta number and anna are inputted duration(months) is asked and then the invoice is generated with the current date and time.

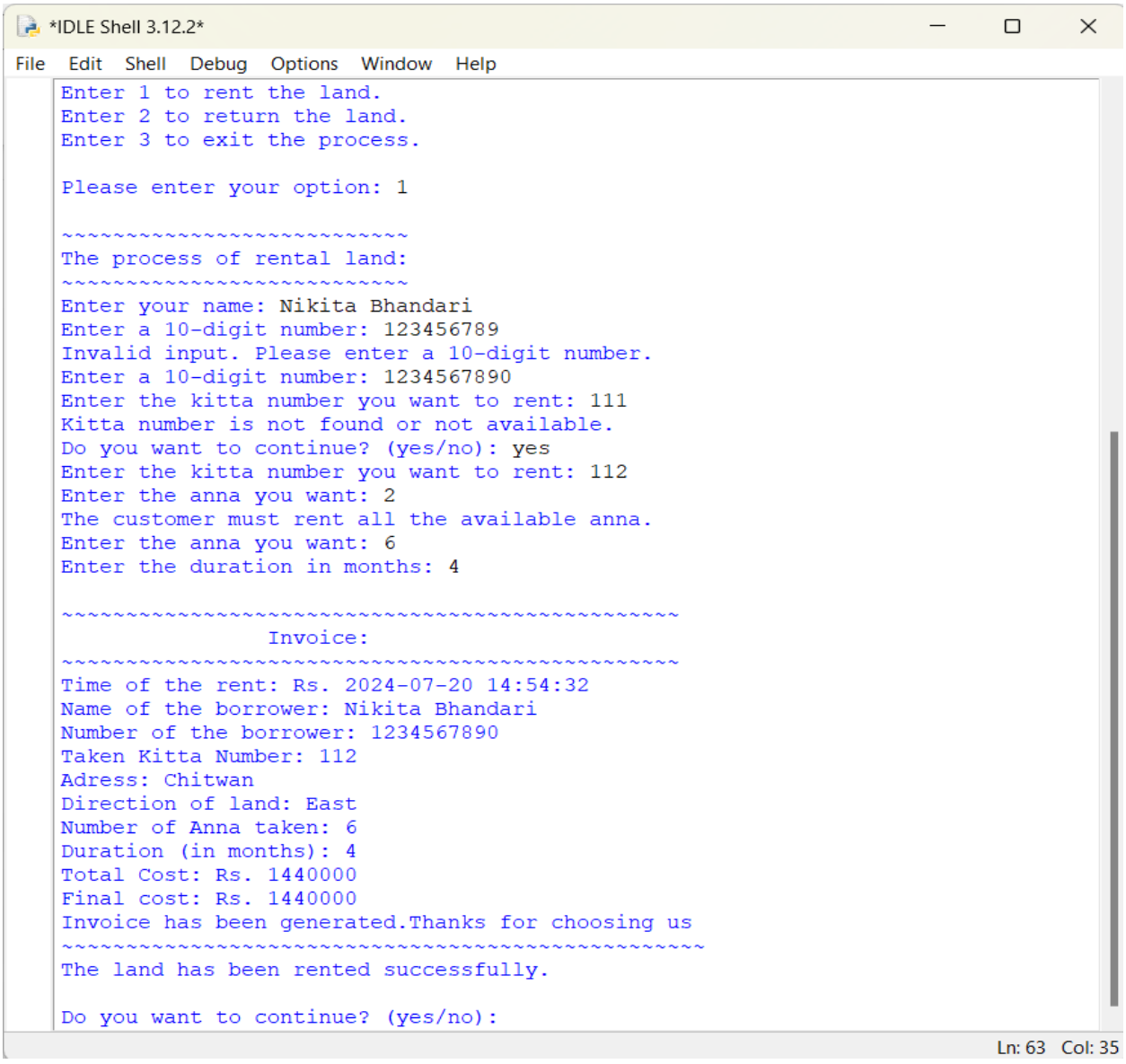


Figure 4: Renting Process 1

After this, the user is asked if they want to continue or not, and if yes the rental process is continued and a invoice is generated. This invoice contains the final which is the cost of both rented lands.

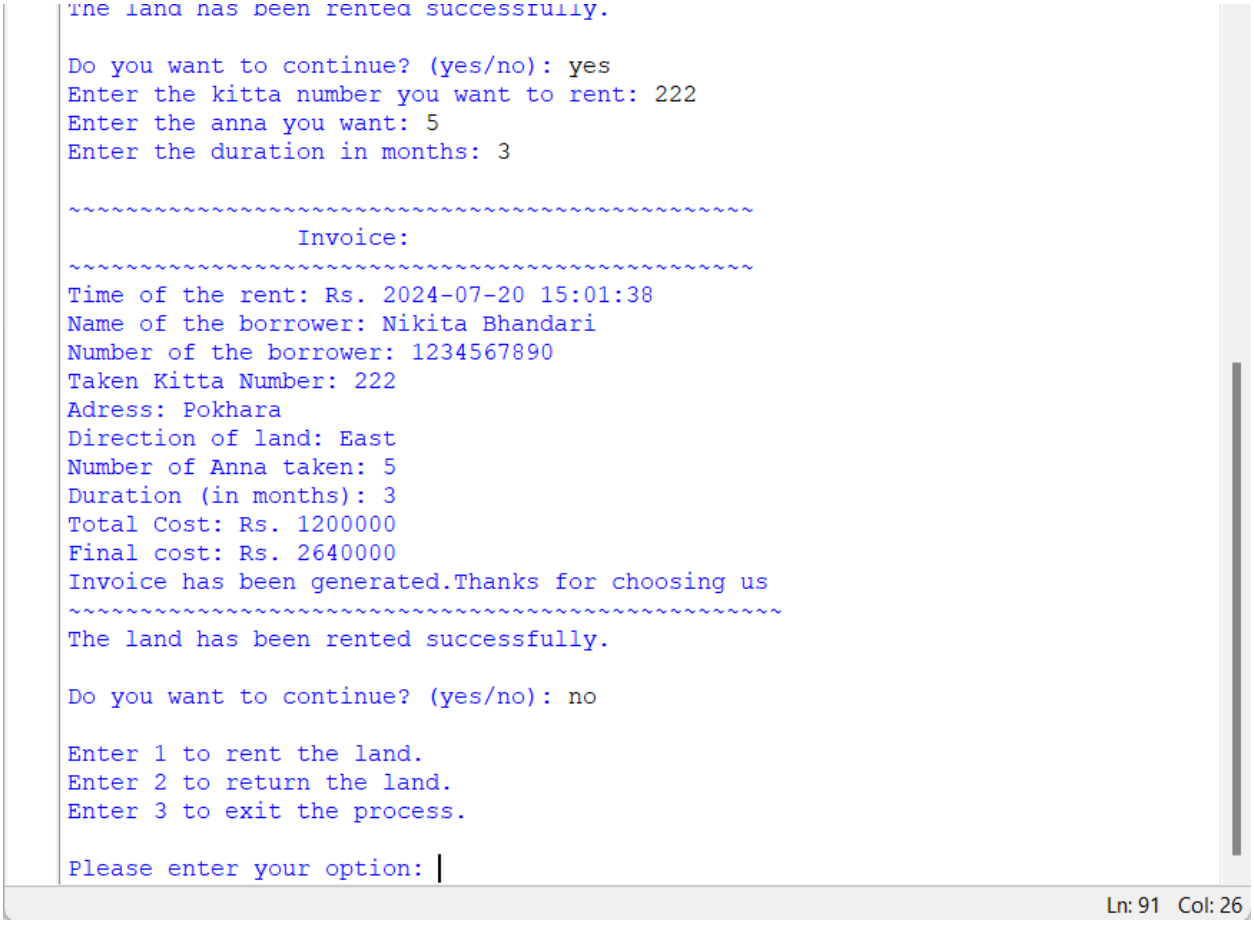


Figure 5: Renting Process 2

Here I will present the formulae for total cost. Here the total_cumulative_cost means Final Cost.

```
price_per_anna = int(i[4])
total_cost = price_per_anna * anna * duration
total_cumulative_cost += total_cost
```

Figure 6: Formulae for total cost

The invoice after renting 2 lands at the same time.

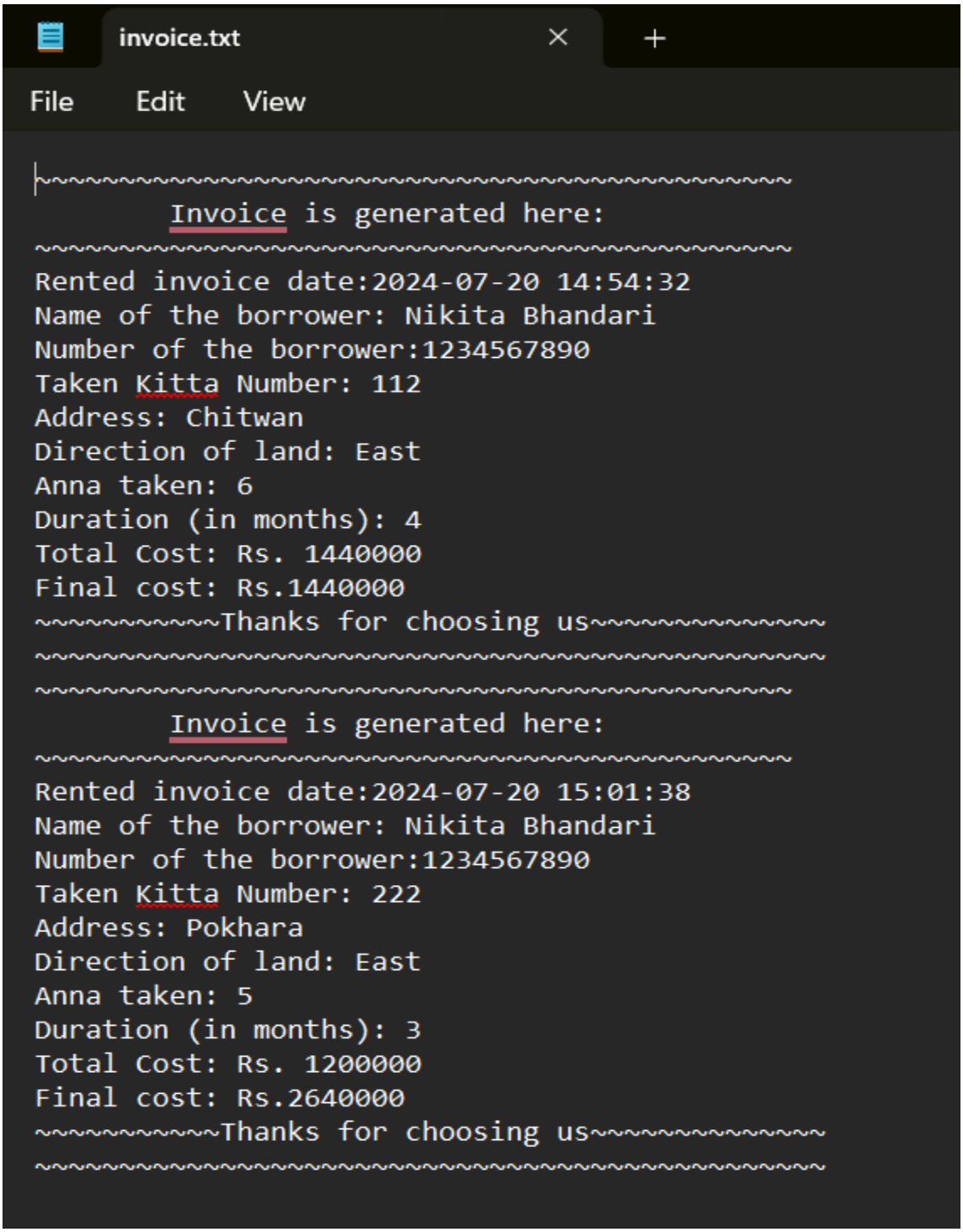


Figure 8: Invoice of rent

After the land is rented the availability is change from available to Not available.

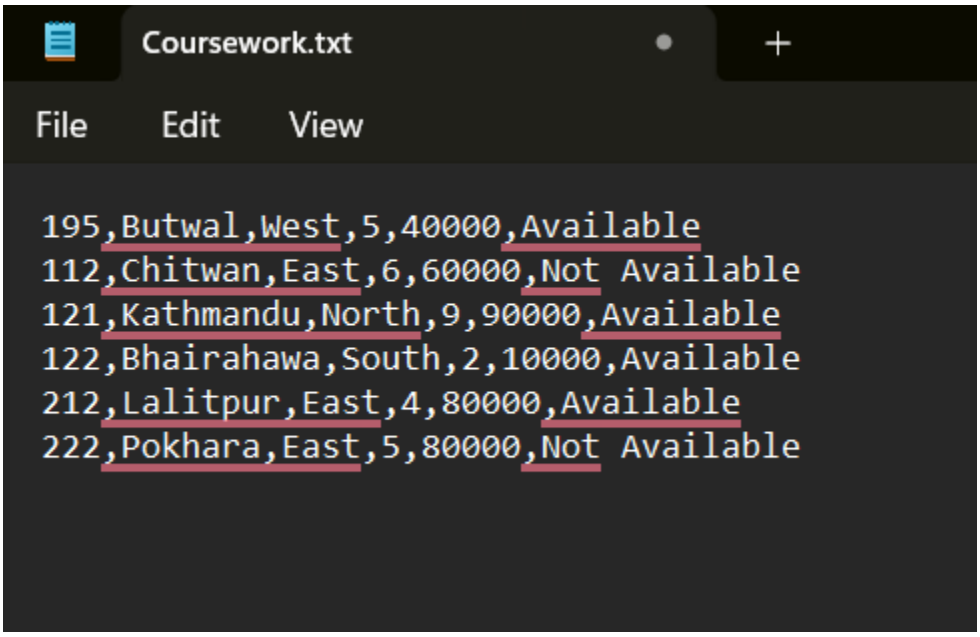


Figure 10: Change in availability

6.3 Returning Process

Here, when the land 195 been rented it changes its availability status to not available. Then, the customer is asked to enter name, number and kitta number. If the kitta number is invalid the customer is asked to enter the valid kitta number. Then the months of land taken and returned is asked.

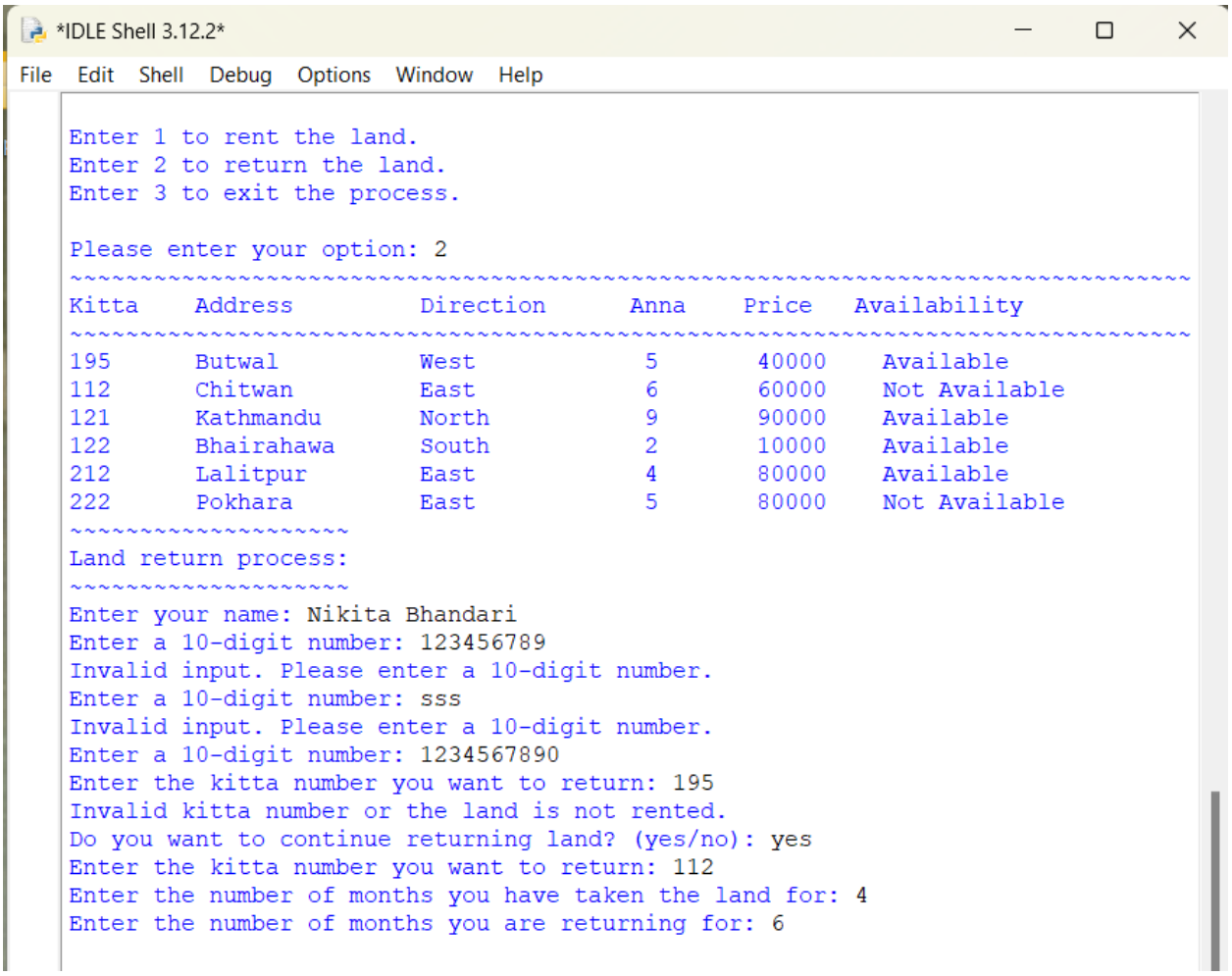


Figure 11: Phone number

Here months returned exceed the months taken so a fine is generated. After that invoice is created as well as the availability is changed to available. The invoice has unique date and time. Here the months where land is returned is more than the months taken so the fine is calculated, and the total cost is calculated.

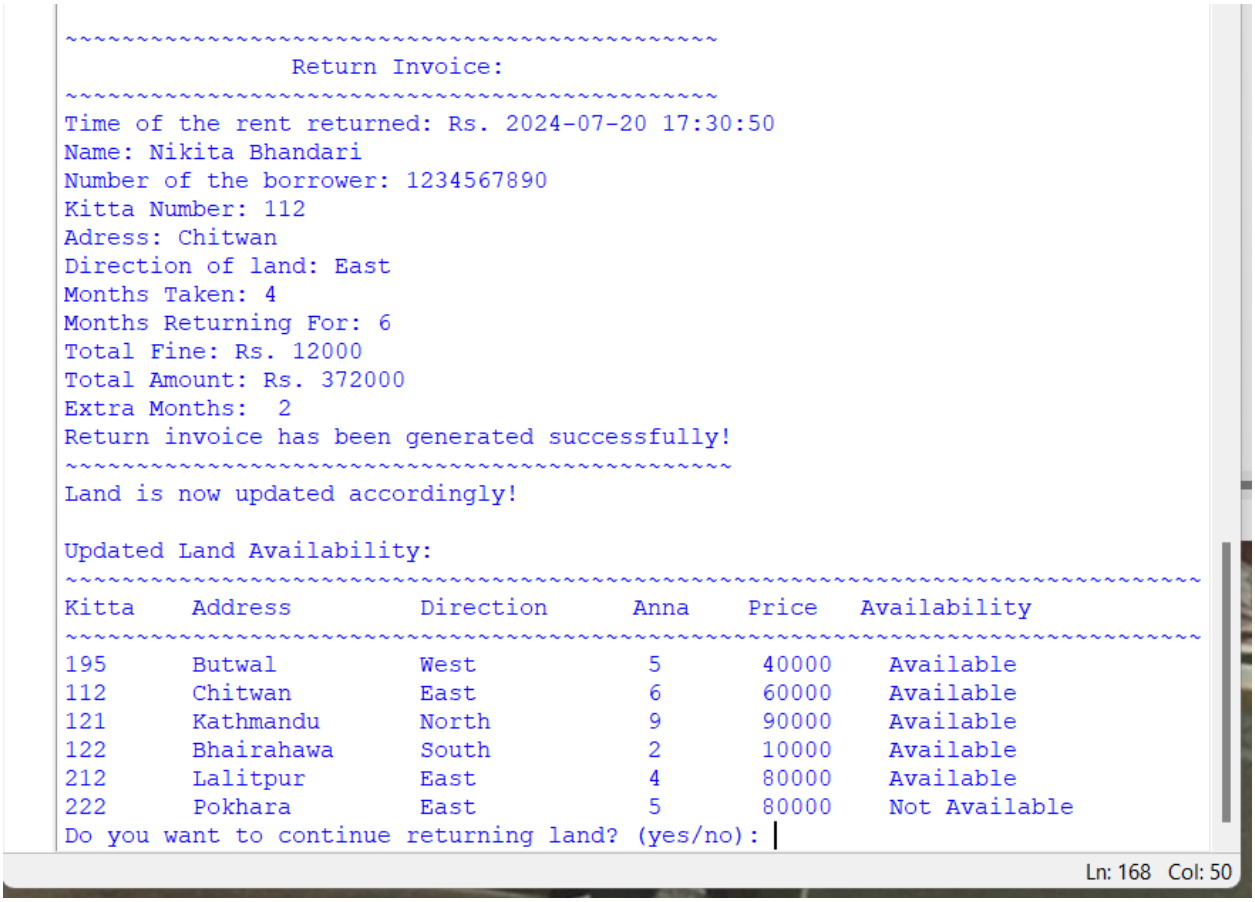


Figure 12: Returning Process 1

Formula for fine if land is returned after the actual time:

```
# Calculate fine, total amount, and extra months
months_taken = int(input("Enter the number of months you have taken the land for: "))
months_returning = int(input("Enter the number of months you are returning for: "))

if months_returning >= months_taken:
    #fine if the land is returned after the rented months
    extra_months = months_returning - months_taken
    price_per_month = int(valid_land[4])
    total_price = months_returning * price_per_month
    fine_percentage = 0.10  # 10% fine per month
    total_fine = round(extra_months * price_per_month * fine_percentage)  # 10% fine for each extra month
    total_amount = total_price + total_fine
```

Figure 13: Formulae for fine 1

Now, when the customer is asked whether they want to continue or not if the answer is yes then the same process is continued and the invoice is created.

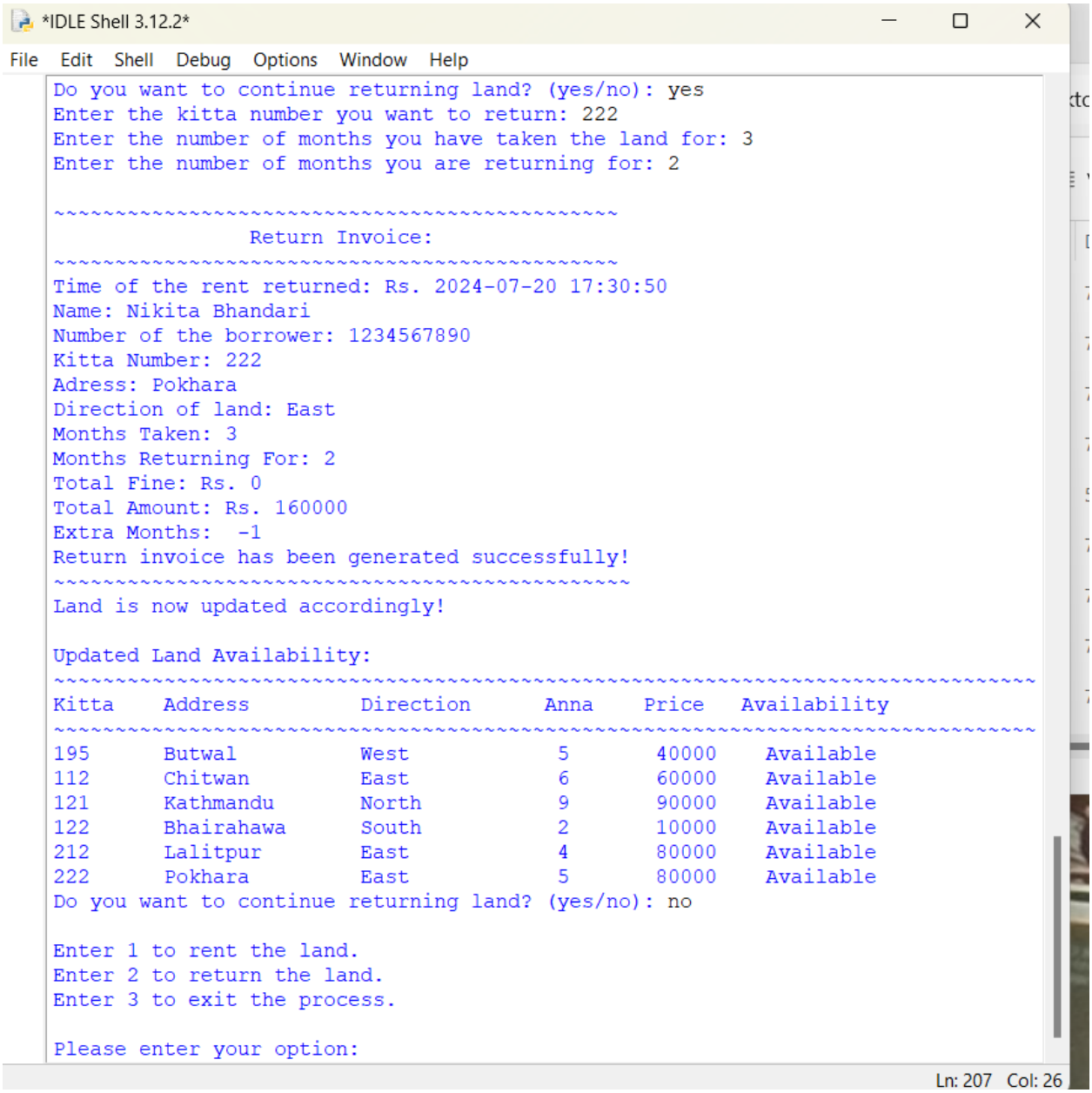
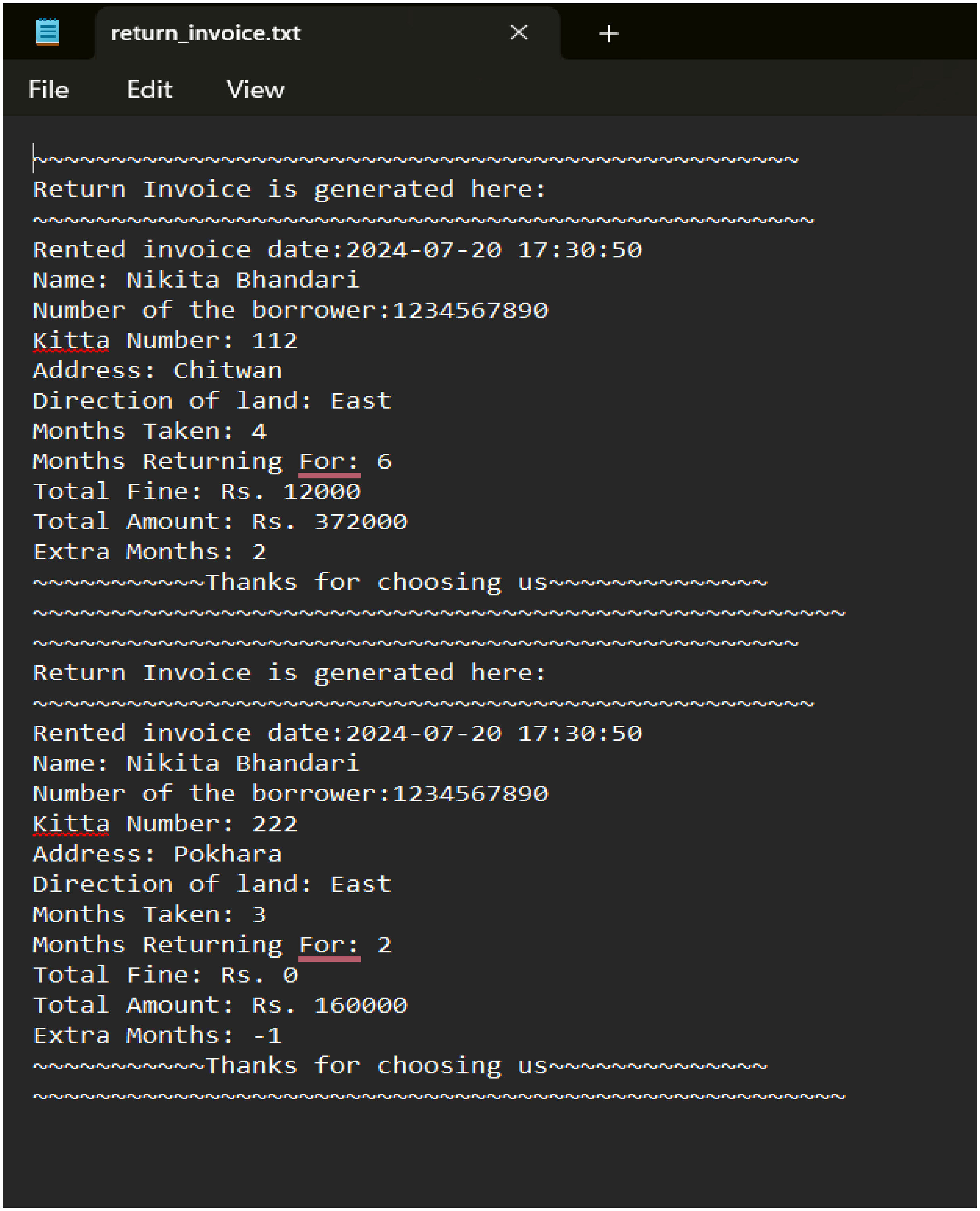


Figure 14: Returning Process 2

Formula for fine if land is returned before fine:

```
#fine if returned before the rented months
total_price = months_returning * int(valid_land[4])  # Price per anna * months taken
total_fine = 0  # No fine if returned before rented months
total_amount = total_price
```

Figure 15: Fine for land return 2



return_invoice.txt

File Edit View

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Return Invoice is generated here:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Rented invoice date:2024-07-20 17:30:50
Name: Nikita Bhandari
Number of the borrower:1234567890
Kitta Number: 112
Address: Chitwan
Direction of land: East
Months Taken: 4
Months Returning For: 6
Total Fine: Rs. 12000
Total Amount: Rs. 372000
Extra Months: 2
~~~~~~~~~~~Thanks for choosing us~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Return Invoice is generated here:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Rented invoice date:2024-07-20 17:30:50
Name: Nikita Bhandari
Number of the borrower:1234567890
Kitta Number: 222
Address: Pokhara
Direction of land: East
Months Taken: 3
Months Returning For: 2
Total Fine: Rs. 0
Total Amount: Rs. 160000
Extra Months: -1
~~~~~~~~~~~Thanks for choosing us~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Figure 16: Invoice for land returned

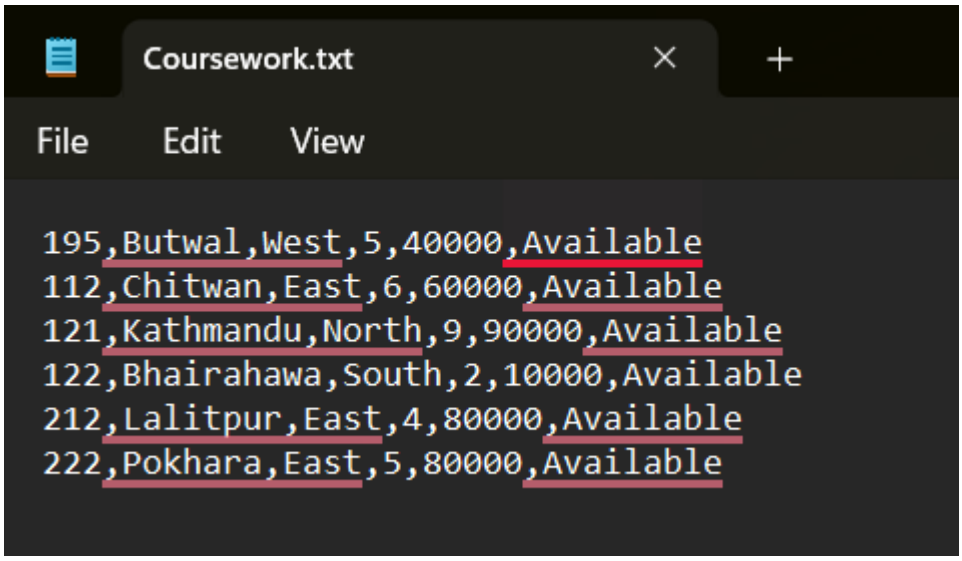


Figure 17: Availability check in txt

6.4 Exit Process

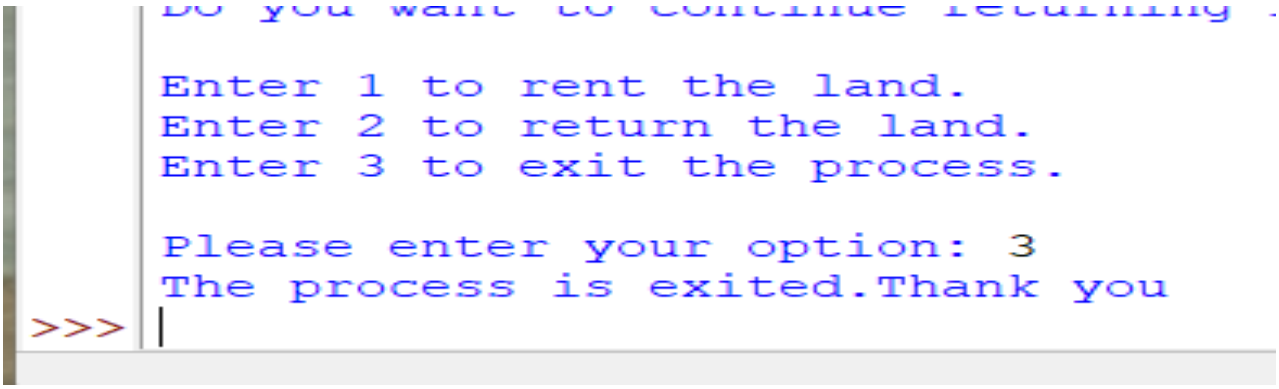


Figure 18: Exit Process

7. Testing

7.1 Test 1

Objective	Show the implementation of the try, except by showing an error message.
Action	The program was opened, and the incorrect number was entered, followed by the correct one.
Expected Result	The program should run if the value is correct and show an error when the wrong value is put.
Actual Result	The program continued when valid input is entered and throws exception when not.
Conclusion	The test was successful.

Table 1: Test 1 for try except

```
#Try except for number
while True:
    numb = input("Enter a 10-digit number: ")
    try:
        if not numb.isdigit() or len(numb)!= 10:
            raise ValueError
        break
    except ValueError:
        print("Invalid input. Please enter a 10-digit number.")
        continue
```

Figure 19: For Try Except

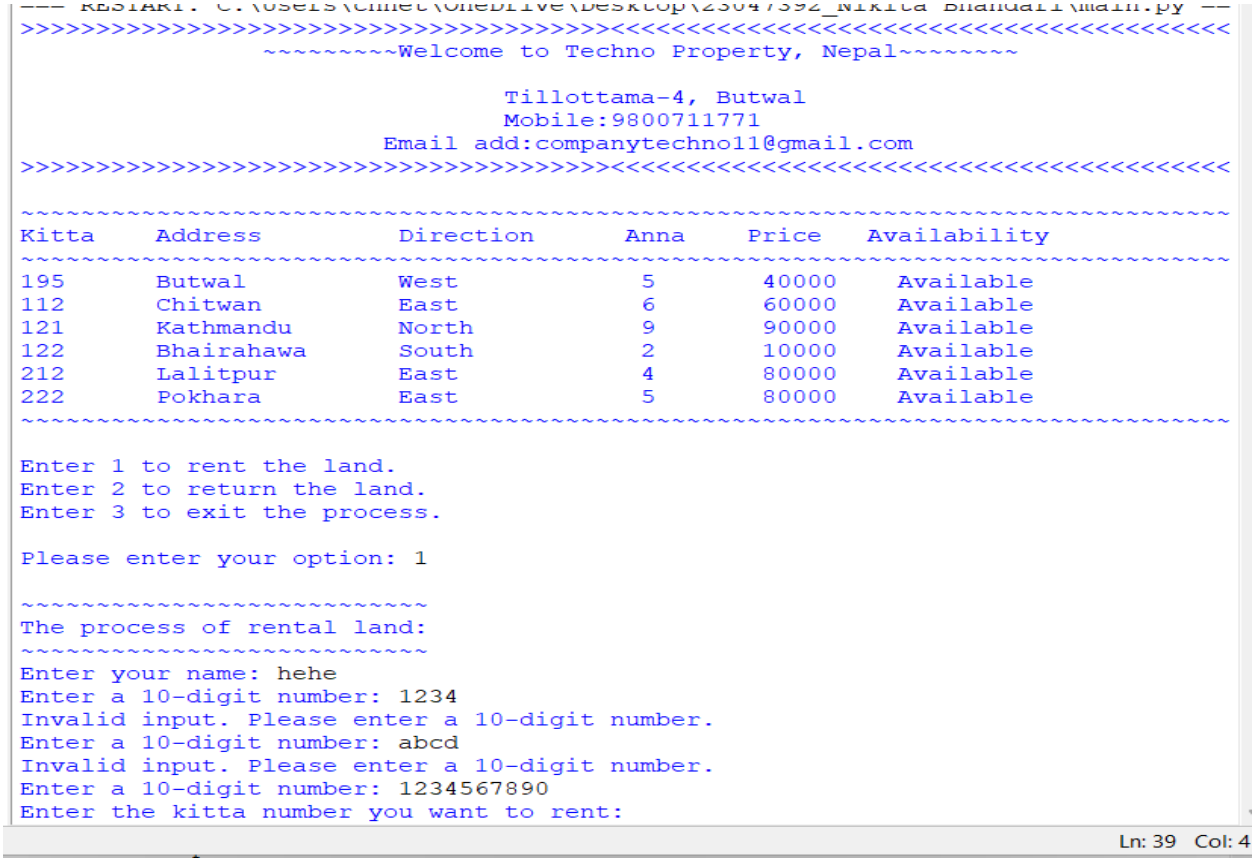


Figure 20: Implementation in phone number

7.2 Test 2

Objective	To Select rent and return land and check if wrong value or negative is inputted it warns the user.
Action	The program was opened, and then the incorrect correct value was entered and then the correct one.
Expected Result	The program should run if the value is correct and show an error when the wrong value is put.
Actual Result	The program generates an invalid message when the wrong option is entered.
Conclusion	The test was successful.

Table 2:Test 2 for wrong input

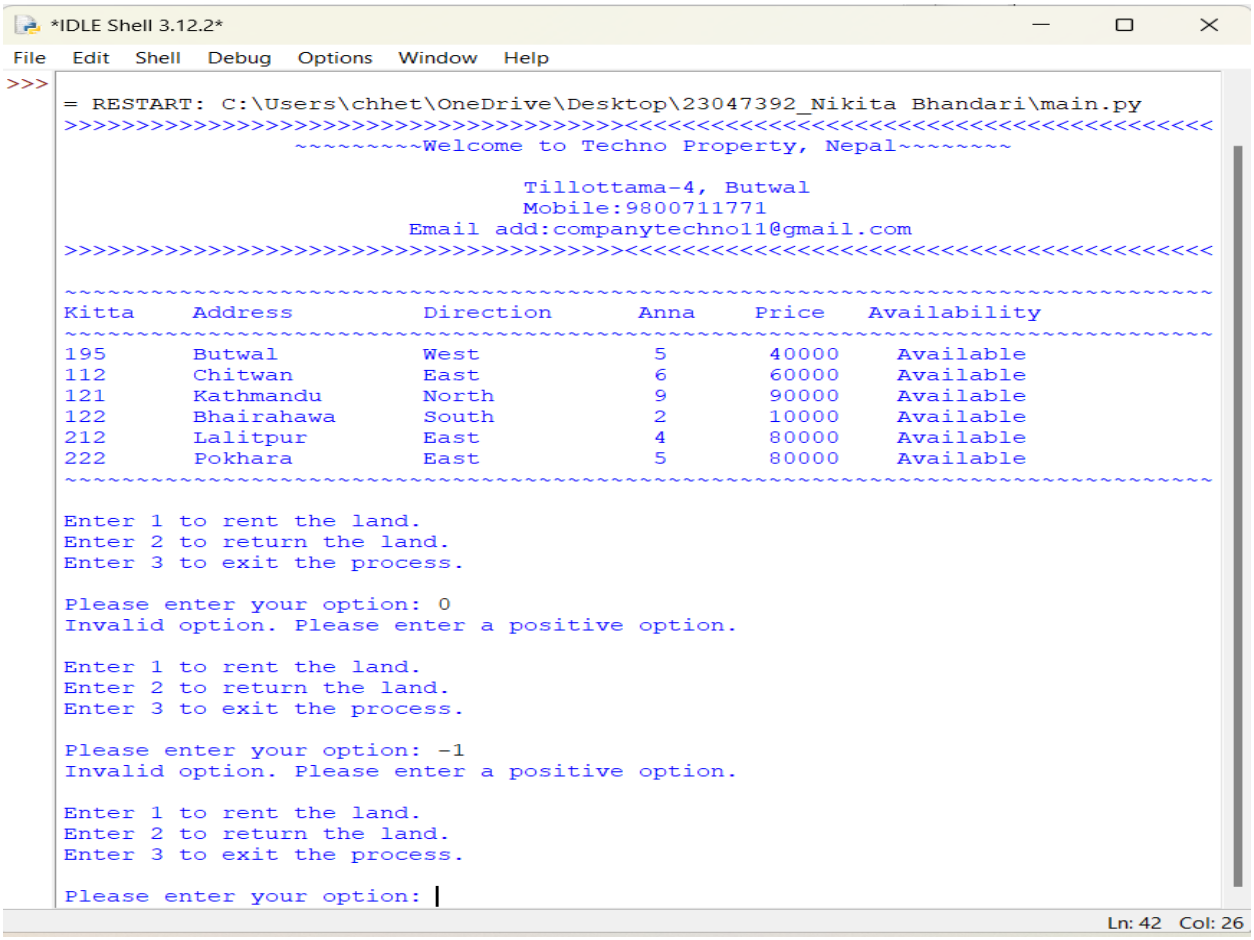


Figure 21:Test 2 for enter wrong input

7.3 Test 3

Objective	To show file generation of renting multiple lands in the shell as well as a text file.
Action	The program was opened, then name, numb, kitta_number, anna,duration is entered.
Expected Result	The invoice should be generated while renting multiple lands in both shell and txt file.
Actual Result	The land should be rented 2 times
Conclusion	The test was successful.

Table 3: Test 3 for continuous rent of land

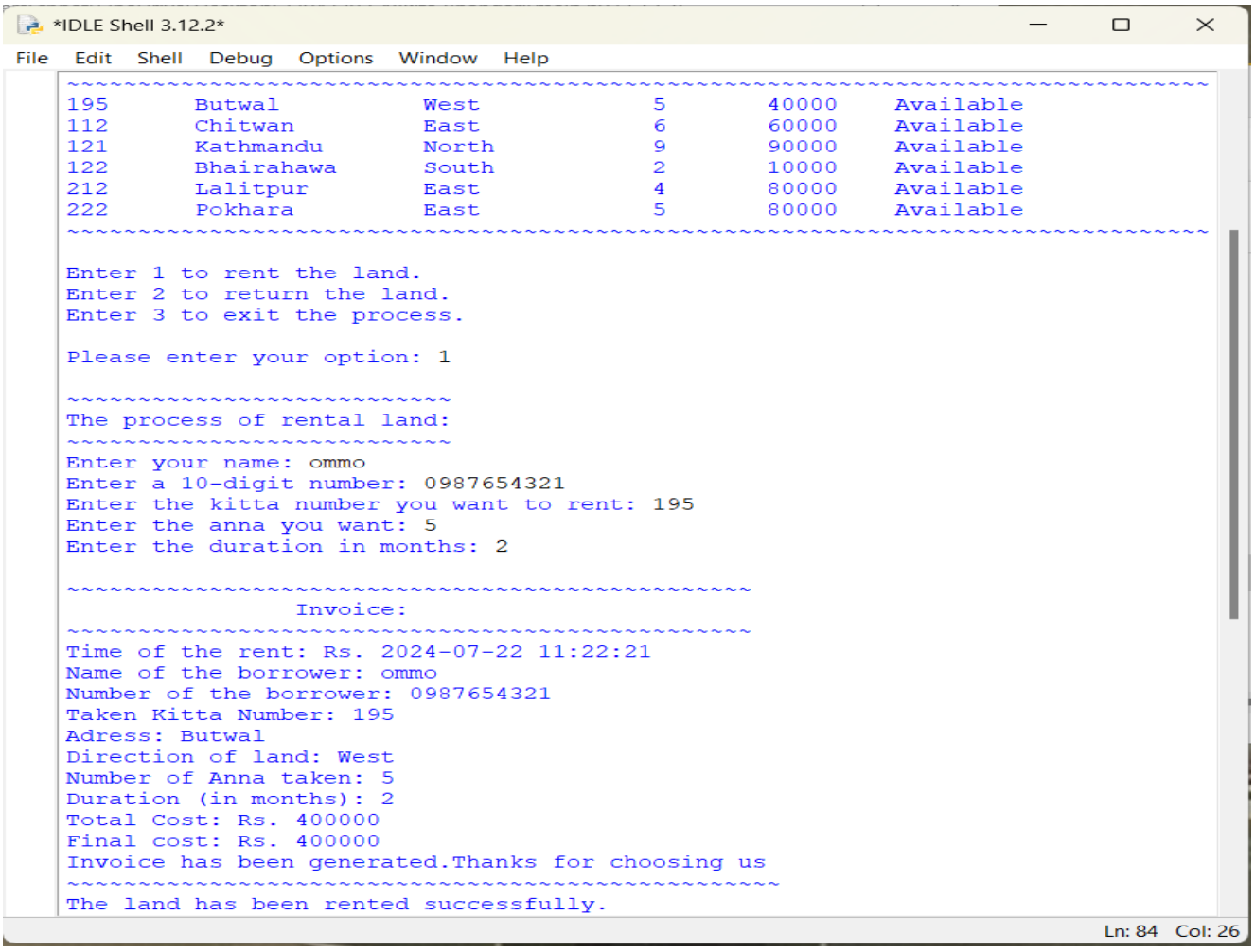


Figure 22: Test 3.1 Multiple land rent

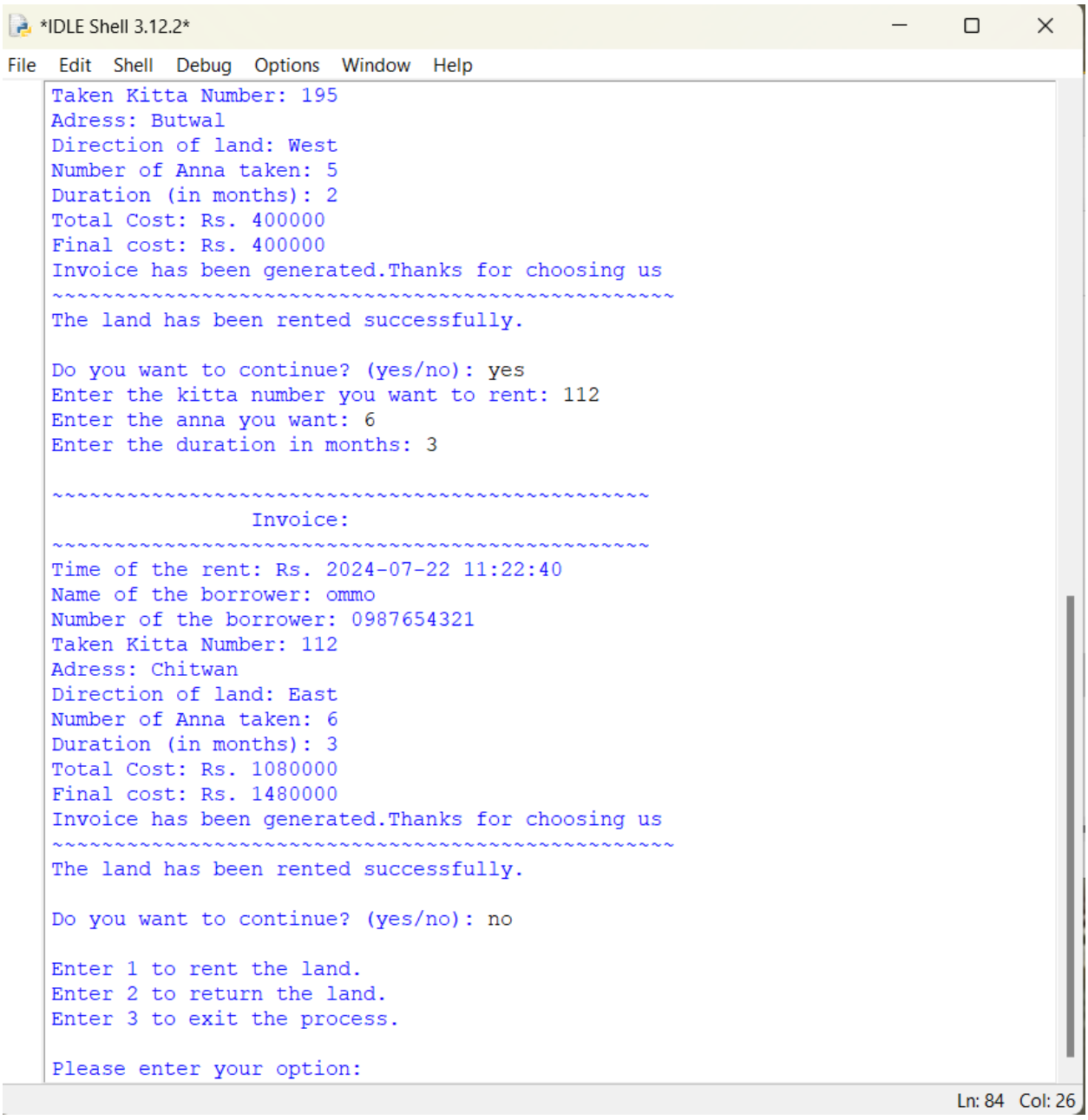


Figure 23: Test 3.2 Multiple land rent

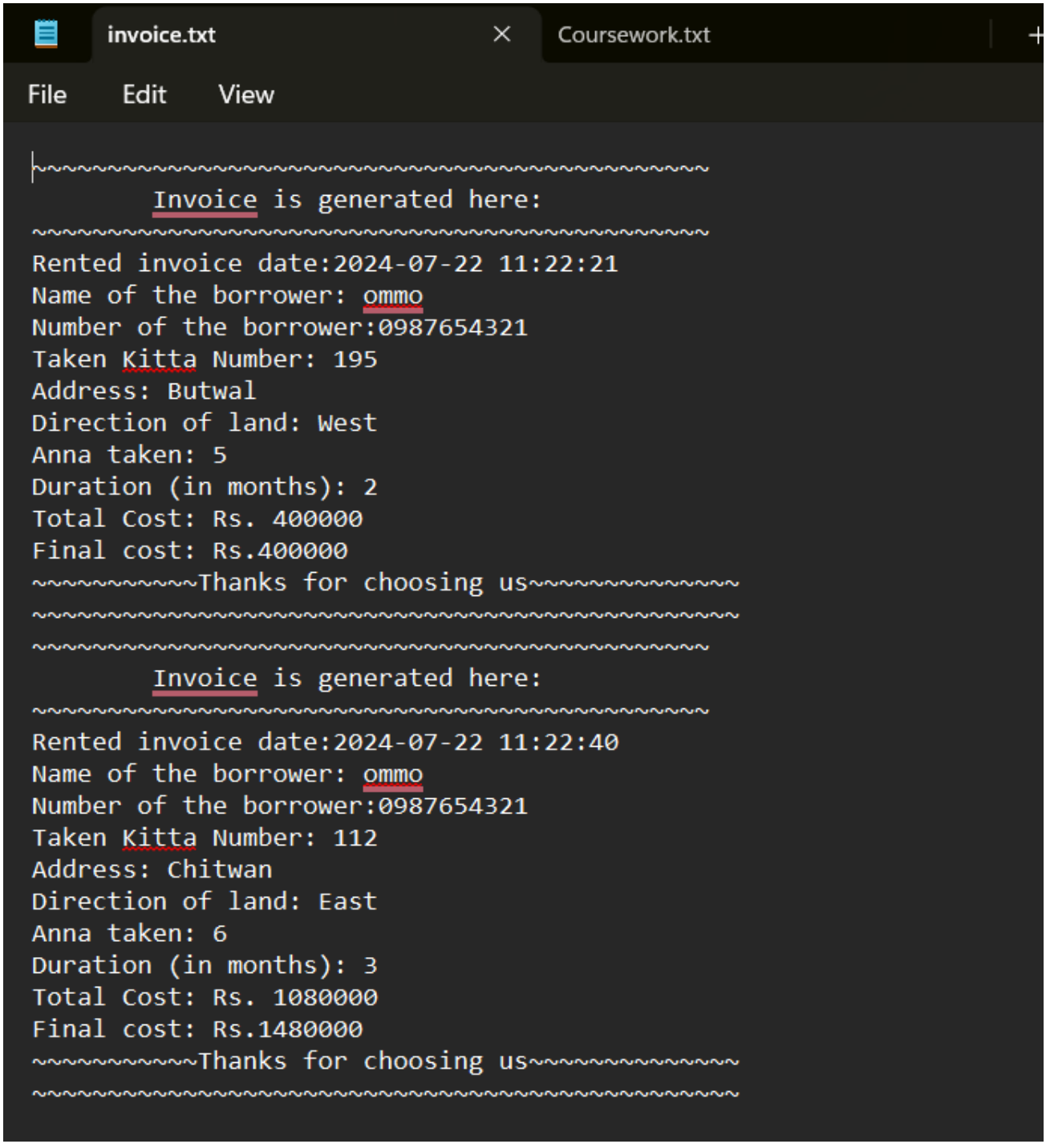


Figure 24: Test 3.3 Land returned invoice in txt

7.4 Test 4

Objective	To show file generation of renting multiple lands in the shell as well as a text file.
Action	The program was opened, and then name, numb, kitta_number, anna,duration is entered.
Expected Result	The invoice should be generated while returning multiple lands in both shell and txt file.
Actual Result	The land should be returned 2 times.
Conclusion	The test was successful.

Table 4: Test 4 for multiple return of land

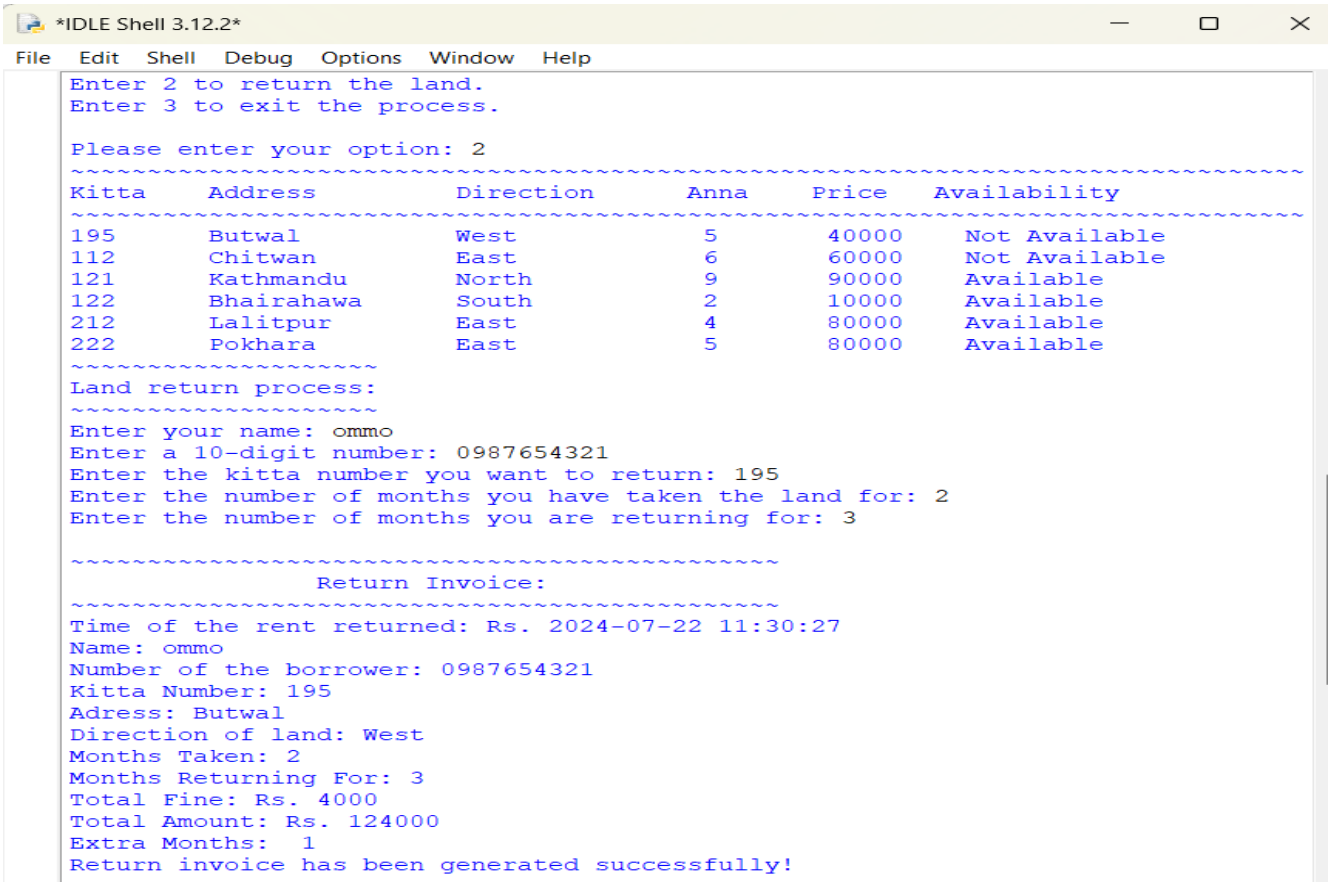


Figure 25: Test 4.1 Multiple land returned

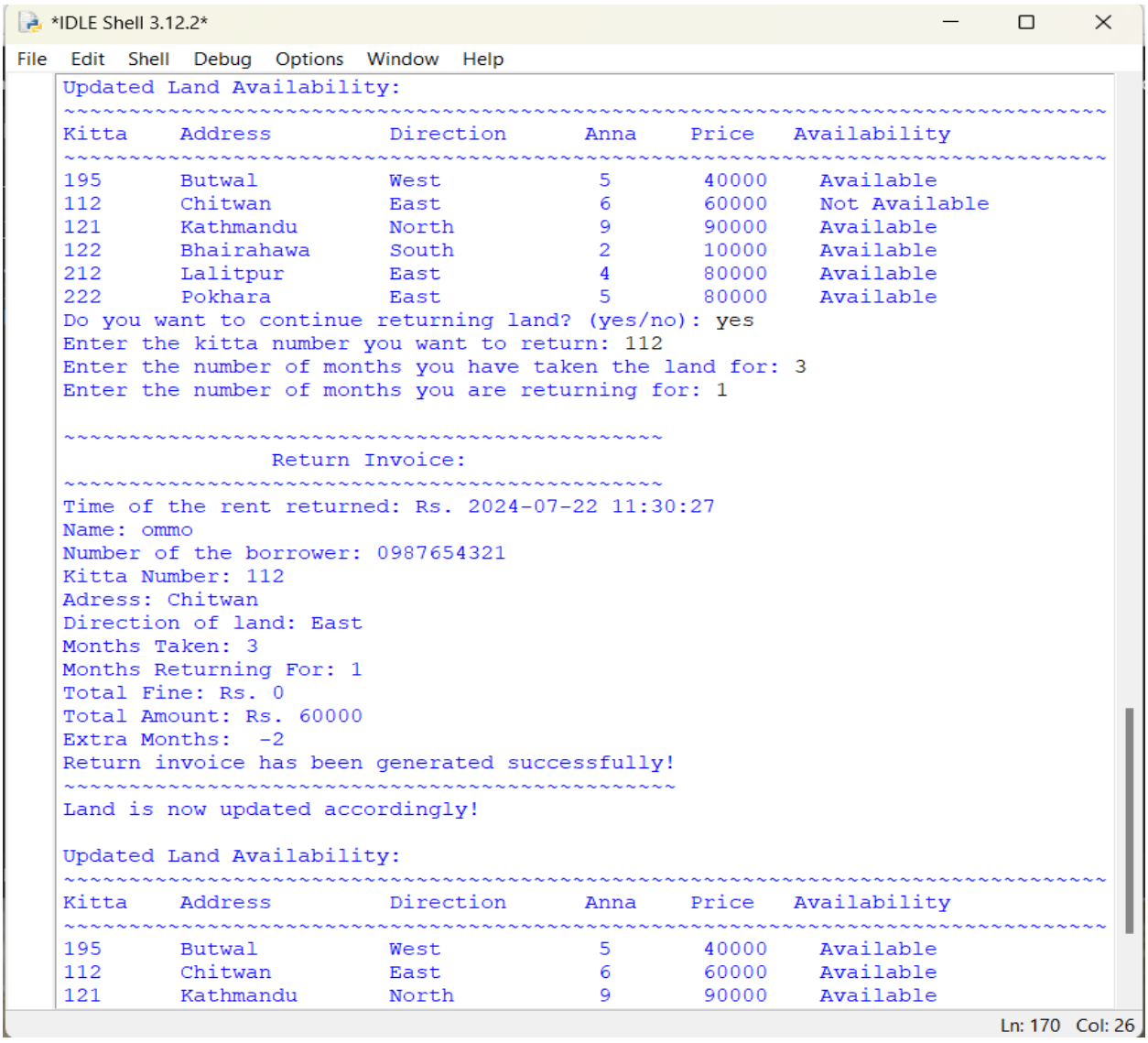
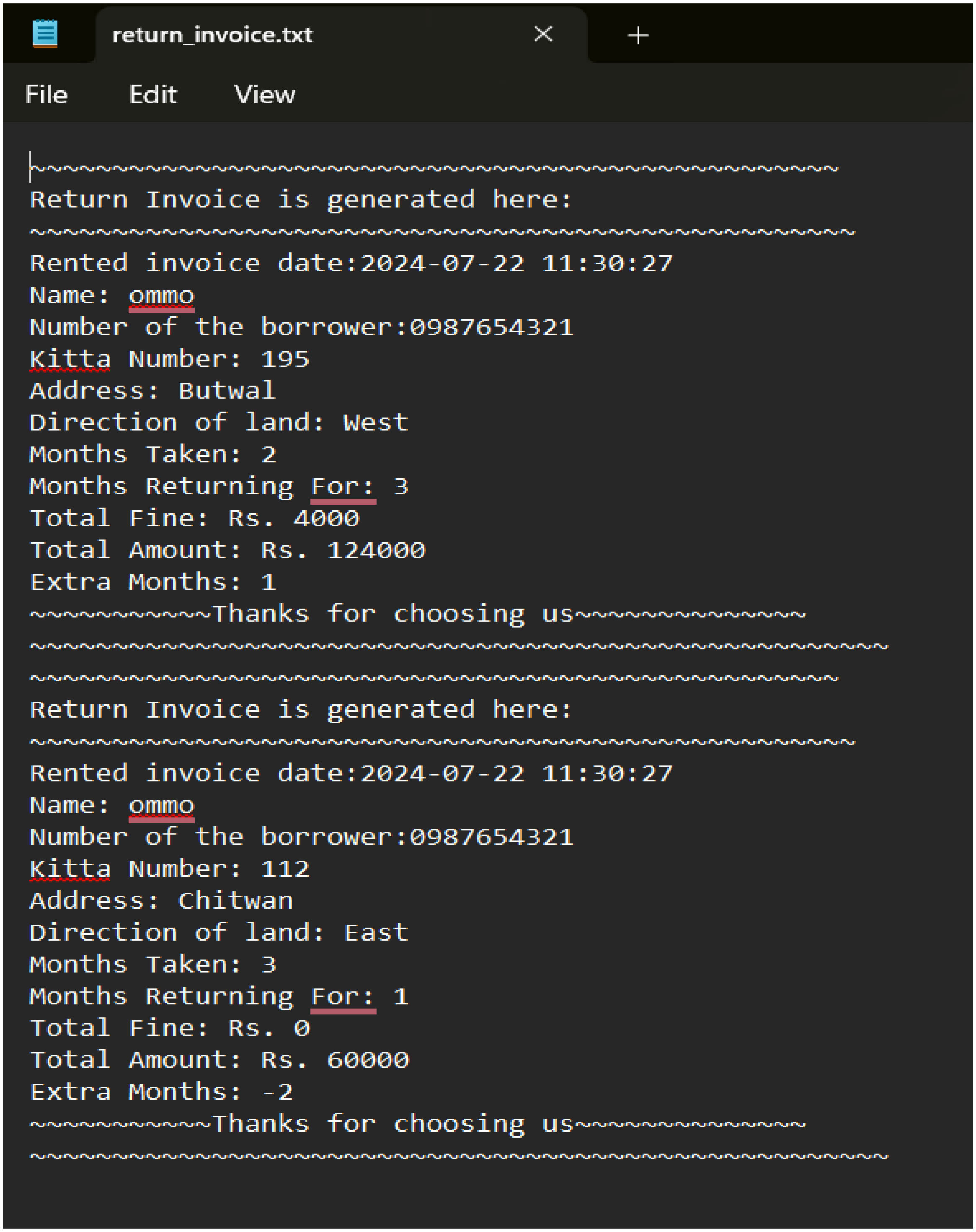


Figure 26 :Test 4.1 Multiple land returned




```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Return Invoice is generated here:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Rented invoice date:2024-07-22 11:30:27
Name: ommo
Number of the borrower:0987654321
Kitta Number: 195
Address: Butwal
Direction of land: West
Months Taken: 2
Months Returning For: 3
Total Fine: Rs. 4000
Total Amount: Rs. 124000
Extra Months: 1
~~~~~~~~~~~Thanks for choosing us~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Return Invoice is generated here:
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Rented invoice date:2024-07-22 11:30:27
Name: ommo
Number of the borrower:0987654321
Kitta Number: 112
Address: Chitwan
Direction of land: East
Months Taken: 3
Months Returning For: 1
Total Fine: Rs. 0
Total Amount: Rs. 60000
Extra Months: -2
~~~~~~~~~~~Thanks for choosing us~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Figure 27: Test 4.3 Multiple land returned invoice in txt

7.5 Test 5

Objective	To show the update in stock of land after renting and returning.
Action	The land was rented and then the availability status changed to unavailable and after returning it changed to available.
Expected Result	The availability status should change in both the console and the txt file.
Actual Result	The land data changed accordingly.
Conclusion	The test was successful.

Table 5: Test 5 for change in availability

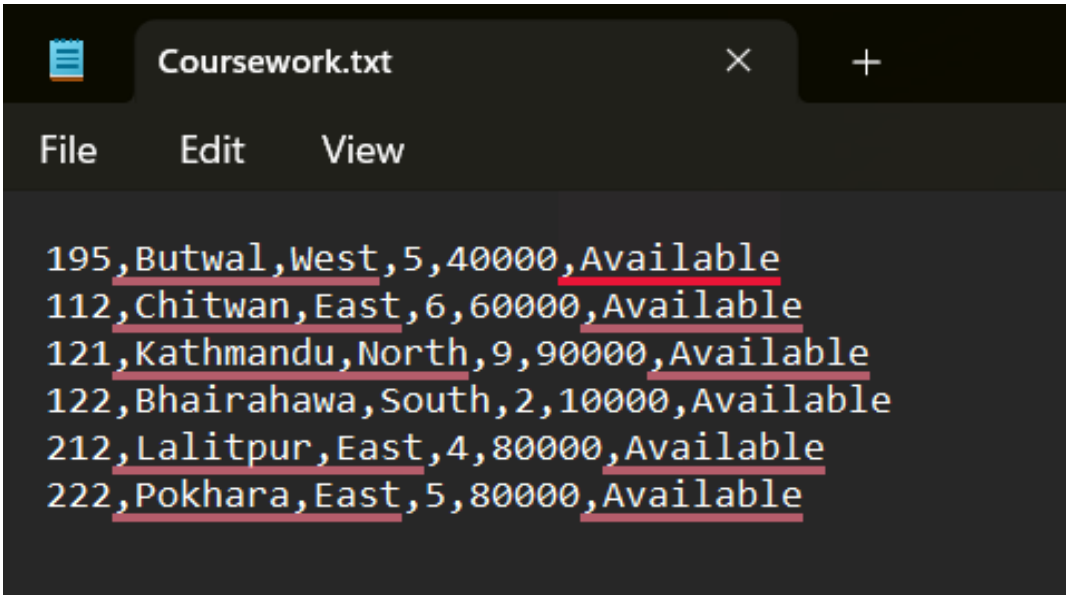


Figure 28: Test 5.1 Availability status in txt

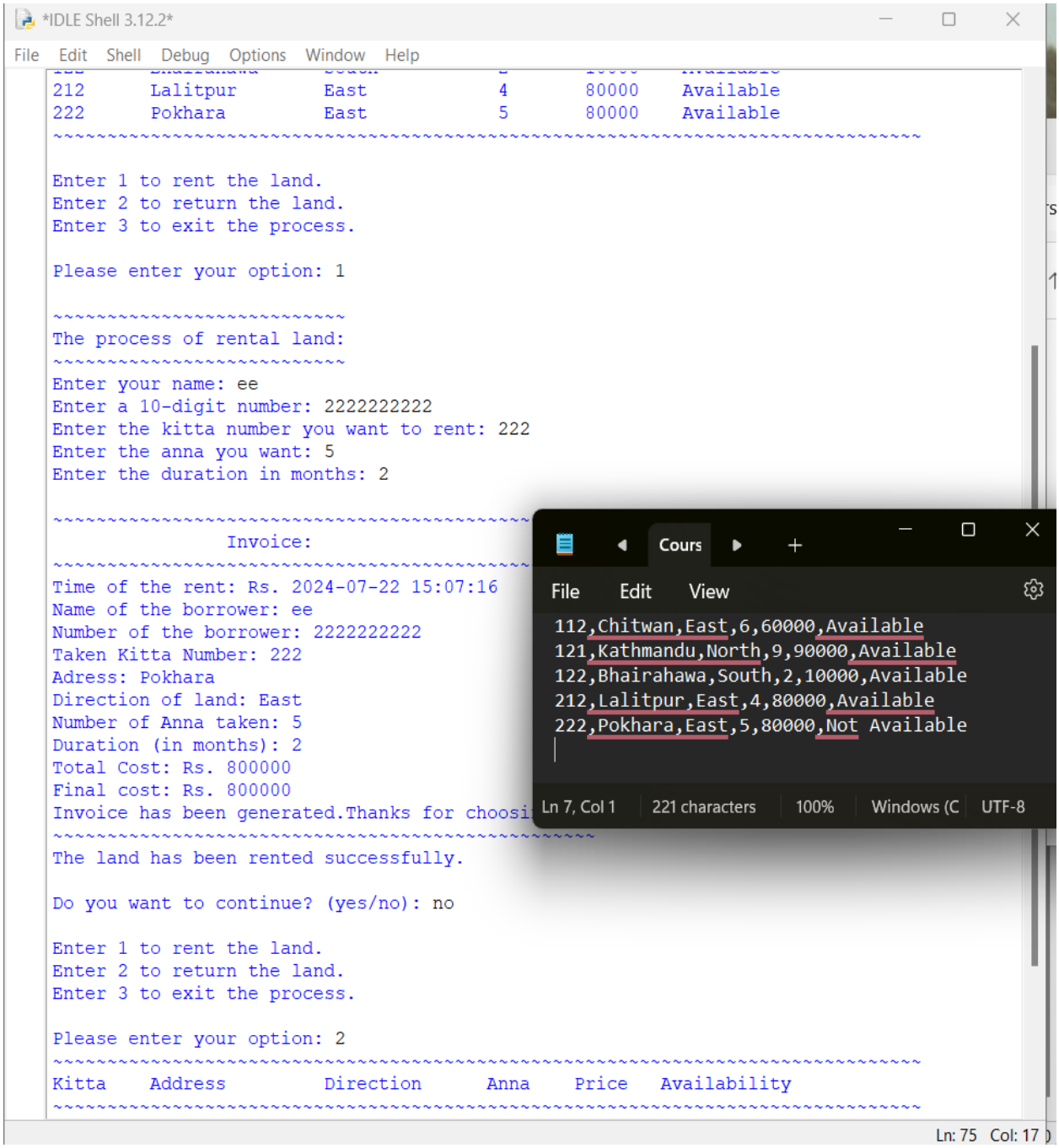


Figure 29: Test 5.2 Availability status change

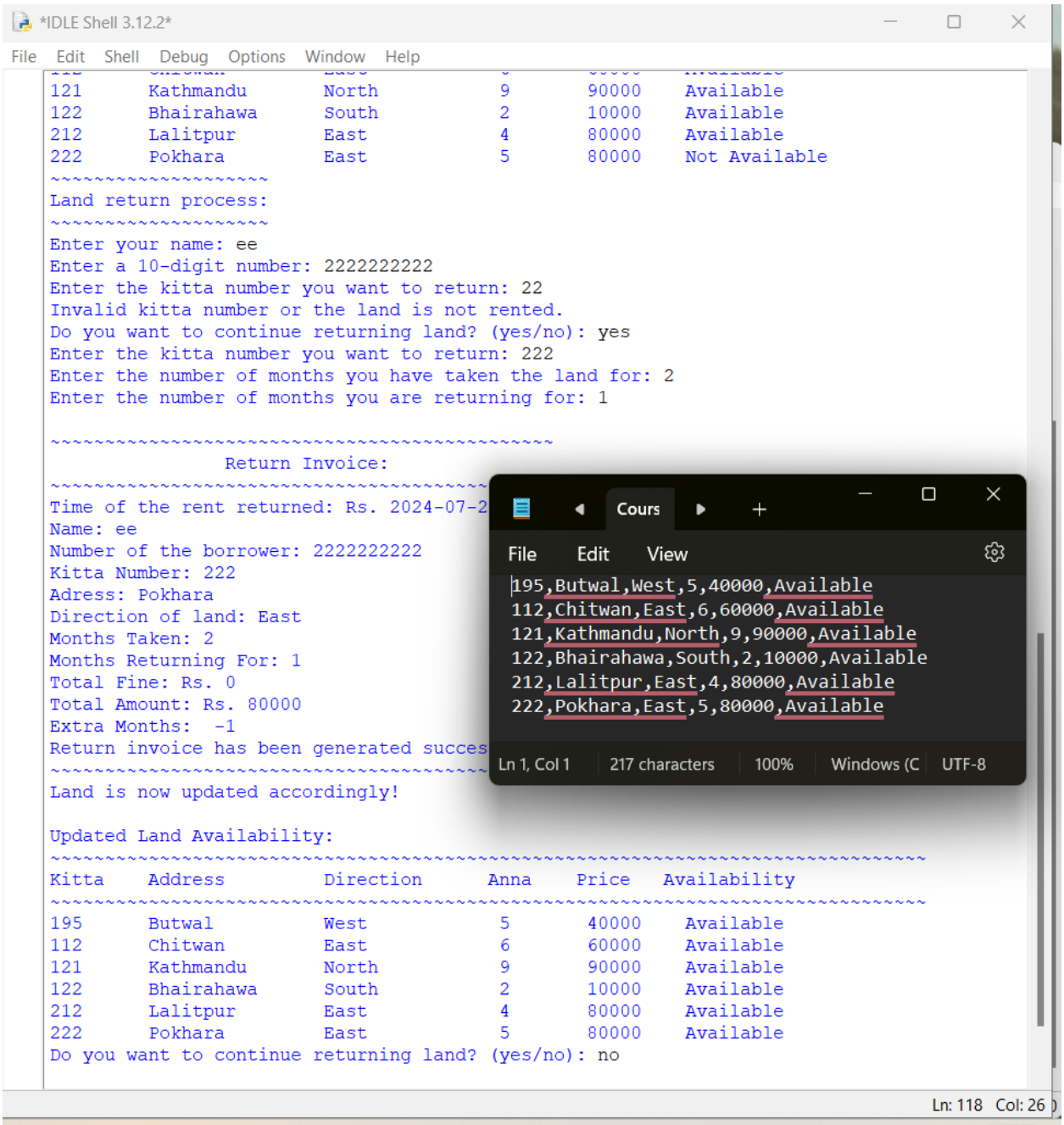


Figure 30: Test 5.3 Availability status change

8. Conclusion

In this coursework, we were asked to create a land rental system that rents land according to the lender's requirements. I have created a user-friendly system where the user can rent land according to their needs. For this assignment to be completed, we were asked to create a report where an algorithm, flowchart, pseudocode, data structure program, and testing are supposed to be written.

The ending of this coursework taught me how to make a proper use of Python. Python is a very powerful tool in the modern business development for running out business. The rental management system designed for TechnoPropertyNepal provides a robust solution for efficiently managing their land rental operations.

I have learned a lot of things like how to organize a program on a txt file by displaying available land, facilitating transactions, and generating invoices of multiple land in both invoice and txt file. After that when the customer returns the land the availability is changed from not available to available. This project is all about transparency and smooth operation.

9. Appendix

9.1 For read_file.py

```
def opening():
    f = open("Coursework.txt", "r")
    temp = [ ]
    for each in f:
        ff = each.replace("\n", "")
        temp.append(ff.split(","))
    return temp
```

9.2 For write_file.py

```
import datetime

def write_invoice(name,numb, kitta_number, address, direction, anna, duration,total_cost,total_cumulative_cost,rented_date):

    # Write invoice to the console
    print("\t")
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    print("\t\tInvoice:")
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    print("Time of the rent: Rs.", rented_date)
    print("Name of the borrower:", name)
    print("Number of the borrower:", numb)
    print("Taken Kitta Number:", kitta_number)
    print("Adress:", address)
    print("Direction of land:", direction)
    print("Number of Anna taken:", anna)
    print("Duration (in months):", duration)
    print("Total Cost: Rs.", total_cost)
    print("Final cost: Rs.", total_cumulative_cost)
    print("Invoice has been generated.Thanks for choosing us")
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

    invoice_date=datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    # Write invoice to a text file
    with open("invoice.txt", "a") as file:
        file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
        file.write("\tInvoice is generated here:\n")
        file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
```

```
        file.write("Rented invoice date:" + rented_date + "\n")
        file.write("Name of the borrower: " + name + "\n")
        file.write("Number of the borrower:"+ str(numb) + "\n")
        file.write("Taken Kitta Number: " + kitta_number + "\n")
        file.write("Address: " + address + "\n")
        file.write("Direction of land: " + direction + "\n")
        file.write("Anna taken: " + str(anna) + "\n")
        file.write("Duration (in months): " + str(duration) + "\n")
        file.write("Total Cost: Rs. " + str(total_cost) + "\n")
        file.write("Final cost: Rs."+str( total_cumulative_cost)+"\n")
        file.write("~~~~~~~~~~~~Thanks for choosing us~~~~~~~~~~~~~~~\n")
        file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")



def write_return_invoice(name, numb, kitta_number, address, direction, months_taken, months_returning, total_fine, total_amount,returned_date):
    # Write return invoice to the console
    print("\t")
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    print("\t\tReturn Invoice:")
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    print("Time of the rent returned: Rs.", returned_date)
    print("Name:", name)
    print("Number of the borrower:", numb)
    print("Kitta Number:", kitta_number)
    print("Adress:", address)
    print("Direction of land:", direction)
    print("Months Taken:", months_taken)
    print("Months Returning For:", months_returning)
    print("Total Fine: Rs.", total_fine)
```

```
    print("Total Amount: Rs.", total_amount)
    print("Extra Months: ", months_returning - months_taken)
    print("Return invoice has been generated successfully!")
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

    # Write return invoice to a text file
    with open("return_invoice.txt", "a") as file:
        file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
        file.write("Return Invoice is generated here:\n")
        file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
        file.write("Rented invoice date:" + returned_date + "\n")
        file.write("Name: " + name + "\n")
        file.write("Number of the borrower:"+ str(numb) + "\n")
        file.write("Kitta Number: " + kitta_number + "\n")
        file.write("Address: " + address + "\n")
        file.write("Direction of land: " + direction + "\n")
        file.write("Months Taken: " + str(months_taken) + "\n")
        file.write("Months Returning For: " + str(months_returning) + "\n")
        file.write("Total Fine: Rs. " + str(total_fine) + "\n")
        file.write("Total Amount: Rs. " + str(total_amount) + "\n")
        file.write("Extra Months: " + str(months_returning - months_taken) + "\n")
        file.write("~~~~~~~~~~~~Thanks for choosing us~~~~~~~~~~~~~~~\n")
        file.write("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
```

9.3 For Operation.py

```
from write_file import write_invoice, write_return_invoice
from read_file import opening
import datetime

def rent_land():
    print("\t")
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    print("The process of rental land:")
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    name = input("Enter your name: ")

    #Try except for number
    while True:
        numb = input("Enter a 10-digit number: ")
        try:
            if not numb.isdigit() or len(numb)!= 10:
                raise ValueError
            break
        except ValueError:
            print("Invalid input. Please enter a 10-digit number.")
            continue

    # Retrieve the 2D list of land data
    listt = []
    total_cumulative_cost=0
    land_data = opening()

    while True:
        valid = False
```

```
    kitta_number = input("Enter the kitta number you want to rent: ")
    for i in land_data:
        if kitta_number == i[0] and i[-1] == "Available":
            address = i[1]
            direction = i[2]
            available_anna = int(i[3])
            while True:
                anna = int(input("Enter the anna you want: "))
                if anna == available_anna:
                    duration = int(input("Enter the duration in months: "))
                    rented_date = datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")
                    info = [kitta_number, duration, rented_date]
                    listt.append(info)
                    price_per_anna = int(i[4])
                    total_cost = price_per_anna * anna * duration
                    total_cumulative_cost += total_cost
                    i[-1] = "Not Available"
                    write_invoice(name, numb, kitta_number, address, direction, anna,
duration, total_cost,total_cumulative_cost, rented_date)
                    print("The land has been rented successfully.\n")
                    valid = True
                    break
                else:
                    print("The customer must rent all the available anna.")

    if not valid:
        print("Kitta number is not found or not available.")

    count = input("Do you want to continue? (yes/no): ")
    if count.lower() != "yes":
```

```
            break


    # Write the updated land data back to the file
    file = open("Coursework.txt", "w")
    for i in land_data:
        file.write(",".join(i) + "\n")
    return listt

def return_land():
    print("~~~~~~~~~~~~~~~~~~~~")
    print("Land return process:")
    print("~~~~~~~~~~~~~~~~~~~~")
    name = input("Enter your name: ")

    #Try except for number
    while True:
        numb = input("Enter a 10-digit number: ")
        try:
            if not numb.isdigit() or len(numb)!= 10:
                raise ValueError
            break
        except ValueError:
            print("Invalid input. Please enter a 10-digit number.")
            continue

    rented_date=datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")


    # Retrieve the 2D list of land data
    land_data = opening()
```

```
    while True:
        kitta_number = input("Enter the kitta number you want to return: ")

        # Find the land in the land data
        valid_land = None
        for i in land_data:
            if kitta_number == i[0] and i[-1] == "Not Available":
                valid_land = i
                break

        if not valid_land:
            print("Invalid kitta number or the land is not rented.")
            count = input("Do you want to continue returning land? (yes/no): ")
            if count.lower() != "yes":
                return  # Exit the function if user chooses not to continue
            else:
                continue

        address=i[1]
        direction=i[2]



        # Calculate fine, total amount, and extra months
        months_taken = int(input("Enter the number of months you have taken the land for:
"))
        months_returning = int(input("Enter the number of months you are returning for: "))

        if months_returning >= months_taken:
            #fine if the land is returned after the rented months
            extra_months = months_returning - months_taken
```

```
            price_per_month = int(valid_land[4])
            total_price = months_returning * price_per_month
            fine_percentage = 0.10  # 10% fine per month
            total_fine = round(extra_months * price_per_month * fine_percentage) # 10% fine for each extra month
            total_amount = total_price + total_fine
        else:
            #fine if returned before the rented months
            total_price = months_returning * int(valid_land[4])  # Price per anna * months taken
            total_fine = 0  # No fine if returned before rented months
            total_amount = total_price

        # Calling the function to write return invoice
        write_return_invoice(name, numb, kitta_number, address, direction, months_taken, months_returning, total_fine, total_amount, rented_date)



        # to update land availability in Coursework.txt file
        for i in land_data:
            if i[0] == kitta_number:
                i[-1] = "Available"
                break

        file = open("Coursework.txt", "w")
        for i in land_data:
            file.write(",".join(i) + "\n")

        # Print a success message
        print("Land is now updated accordingly!")
```

```
        # Update the unavability status to available
        print("\nUpdated Land Availability:")

print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~")
        print("Kitta""\t"" Address""\t"" Direction""\tAnna""\tPrice""\t""Availability")

print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~")
        for i in land_data:
            if len(i) >= 6:
                print("%-8s %-15s %-15s %-7s %-8s %-8s" % (i[0], i[1], i[2], i[3], i[4], i[5]))


        count = input("Do you want to continue returning land? (yes/no): ")
        if count.lower() != "yes":
            break
```

9.4 For Main.py

```
from operation import rent_land, return_land
from read_file import opening

def design():

print(">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<
<<<<<<<<<<<<<<<<<<<<")
    print("\t\t~~~~~~~~~~Welcome to Techno Property, Nepal~~~~~~~~~\n")
    print("\t\t\t\tTillottama-4, Butwal")
    print("\t\t\t\tMobile:9800711771")
    print("\t\t\tEmail add:companytechno11@gmail.com")

print(">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<
<<<<<<<<<<<<<<<<<<<<\n")



def table():
    data = opening()

print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~")
    print("Kitta""\t"" Address""\t"" Direction""\t""Anna""\t""Price""\t""Availability")

print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~")

    for i in data:
        if len(i) >= 6: # Added a check to ensure the row has enough elements
            print("%-8s %-15s %-15s %-7s %-8s %-8s" % (i[0], i[1], i[2], i[3], i[4], i[5]))
```

```
design()

def main():

print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~")

    while True:
        print("\nEnter 1 to rent the land.")
        print("Enter 2 to return the land.")
        print("Enter 3 to exit the process.\n")
        option = int(input("Please enter your option: "))

        if option <= 0:
            print("Invalid option. Please enter a positive option.")
            continue

        if option == 1:
            table()
            rent_land()
        elif option == 2:
            table()
            return_land()
        elif option == 3:
            print("The process is exited.Thank you")
            break
        else:
            print("Invalid option. Please enter a valid option.")

main()
```