CS4051NI Fundamentals of Computing

60% Individual Coursework

2023/24 Spring

Student Name: Anupam Shrestha

London Met ID: 23047592

College ID: NP01NT4A230238

Assignment Due Date: 7 may2024

Assignment Submission Date: 7 may 2024

Word Count:

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

Table of content

Table of figure

Introduction

The project I am working on in the course of my Computing Fundamentals (CC4059NI) course is the development of a spatialized system for the management of land parcels. The system creates opportunities for people and businesses to let land [out] in their ownership for different functions like agriculture, construction, and events. To build this system, I choose to use Python which is a popular programming language due to its simple nature and functionality.

Python an intuitive language for the beginners which don’t ask about the type of the corresponding variable but, its type can be specified automatically to fit the current task. The code looks flexible and easily adjustable. It notches this as tackling of data also includes executing complex arithmetic, important in case of handling land rental transactions and processing payments.

, The system provides basic functionalities such as management of the register of available pieces of land, recording tenant's signing agreements, issuing statements, and taking care of cash returns. The system will do that as well; there will be notifications about the approaching due date of the return and refund for those who do not return them on the due date.

I need to create algorithms and flowcharts define the whole operation. For this to be truthful, I'll have to proceed with first mechanics then create flowcharts at the end. Besides, I tested the code by myself to be sure of its full functionality and non-presence of error.

The objective of the project is to develop a user-friendly and accessible system through which landlords and renters can easily share their individual land without much hassle on their part. Through the establishment of an organized and productive platform by which various parties interested in land rentals may transact, the purpose of the system is to meet all those who may find such structures an asset when leasing land for various tasks.

Tool used

Microsoft Word:

It is fair to admit that Microsoft Word turned out to be a lifesaver as far as setting the project requirements, features, and specifications are concerned. Documentation that was clear and unambiguous about the systems' purposes, user interfaces and the way to operate was made possible by it. Beyond that, the team's productivity was enhanced as Microsoft Word supplied an editorial platform to take care of documents related to the project.

Draw.io:

Draw.io with its diagramming capabilities played a role of a pivotal player during the process of creating the architecture and data flow of the land management system. As a result of Draw.io's simple to use interface as well as a variety of structure models, it was easy to make different diagrams explaining how all the components of the application are related to each other. These diagrams that include entity-relationship and workflow charts provided an essential conceptual framework for structuring and thinking through the system operations and logic.

IDLE (Integrated Development and Learning Environment):

IDLE, the integrated development environment for Python, was the key coding environmental that was made use of for creating the land management application. Based on the features such as syntax highlighting, code completion, and interactive debugging, IDLE accentuates the ease of writing, testing, and executing code quickly. It's a pretty easy to use a tool to create and run Python scripts. With it, developers were able to implement what the app does quickly and efficiently.

Algorithm

Step 1.Import the datetime module.

Step 2.Import the write module.

Step 3. In lists the display_lands function is defined with the land_data being a parameter.

Step 4. Print a landheader to display infomation about the property.

Step 5. Generate a dictionary that will contain each land dictionary in land_data.

Step 6. Extract relevant data from the country's dictionary, prepare it and make it readable.

Step 7. Dispatch this land details in a formatted manner.

Step 8. Defined the rent_land function with inputs of land_data and rented_lands in the function scope.

Step 9. Establish the current date and time.

Step 10.Start a while loop.

Step 11. Show the land specification by displaying_specific_land.

Step 12. Guide the user in a provision of customer details.

Step 13. Print the ister to enter the Kitta figures (or land) they want to rent our.

Step 14. Find out your total available area.

Step 15. Demand of the user should be verified that if he/she wants to rent all the given lands.

Step 16. Provide the user with an input field to enter how long they will rent the property for.

Step 17. Start a list without any information to display invoice data.

Step 18. Apply KittalaK number entered by the user as a Gather operator.

Step 19. Consult the map whether Kitta number belongs to the available land for rent.

Step 20. Grant Assumption: Store current land status in NotAvailable field, record customer details, and run calculation on return date and amount.

Step 21. Concatenate the invoice data to the all_invoice_data list using concatenation operator.

Step 22. Find out from the user whether he/she also wants to rent the nearby land.

Step 23. Write invoice to a file in calling the write.write_invoice function.

Step 24. The land information file can be changed by the write_land_data method.

Step 25.Return all invoice data.

Step 26. Assign the function to return_lands with arguments land_information and rented_lands.

Step 27. Make the program output today's date and time.

Step 28.Start a while loop.

Step 29. Demonstrate the available lands utilising the display_lands fuction.

Step 30. It's important to ask the user to enter customer information.

Step 31. The screen should alert the person to type in the Kitta numbers of the land they want to claim.

Step 32. Herald to the user to enter the duration of returning.

Step 33. Make a list of empty data structures which would hold all the data related to the invoice.

Step 34. Iterate repeatedly through the Kitta numbers given by the user.

Step 35. Review the plot on the given Kitta number to see if the specified customer is renting the field.

Step 36. If rented, if necessary enter the land status to 'Available', and rent returned date and amount.

Step 37. Append the invoice data to the all_invoice_data list, if there is any.

Step 38. Ask the user it they want to reclaim more land.

Step 39. Write invoice data to file with the presence of the write.write_invoice function.

Step 40. Run the write_land_data function for updating land file data.

Step 41.Return all invoice data.

Step 42. End

Flow Chart

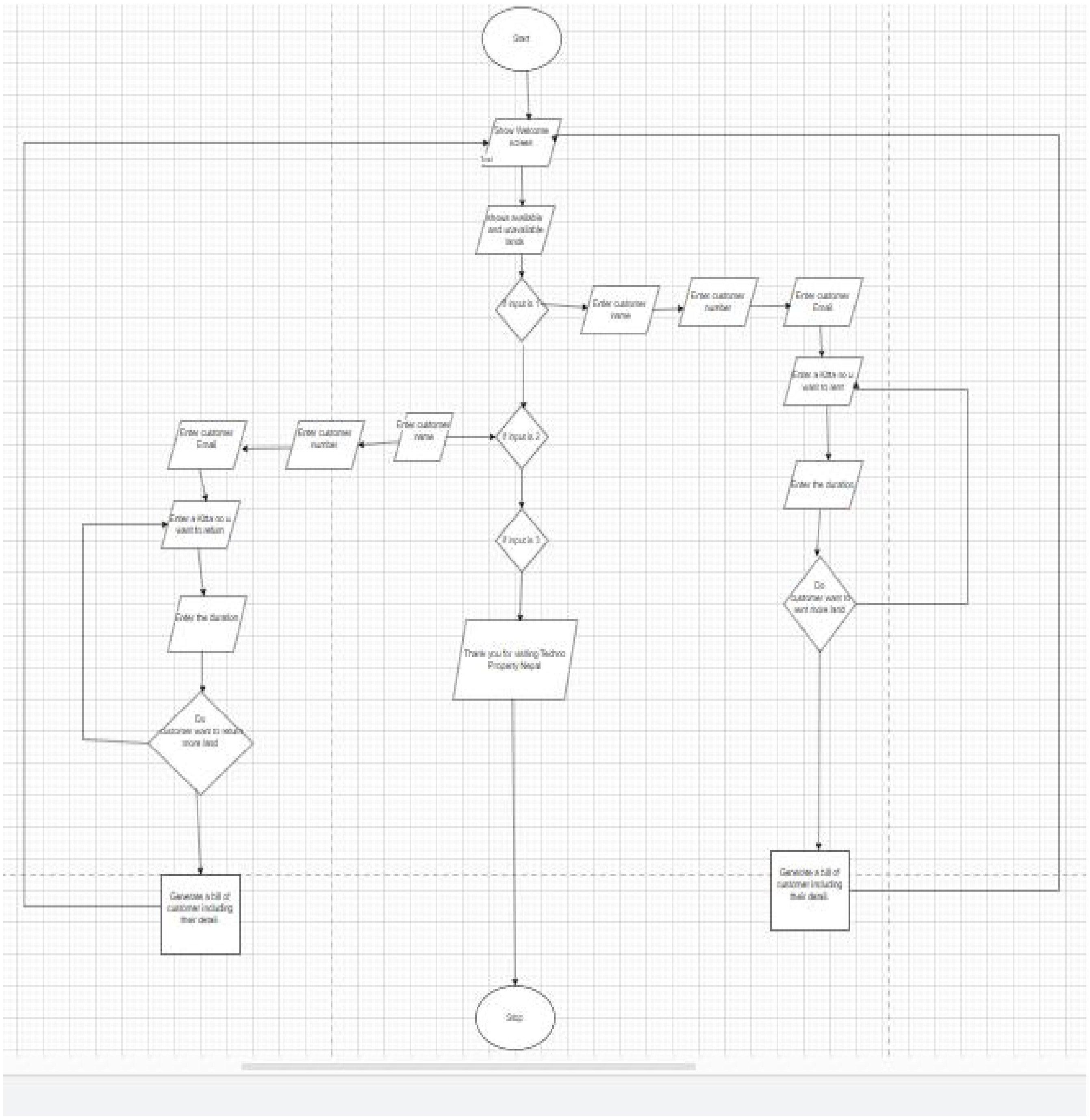

Start
Show Welcome screen
shows available and unavailable lands
if input is 1
Enter customer name
Enter customer number
Enter customer Email
Enter a Kitta no u want to rent
Enter the duration
Do customer want to rent more land
Generate a bill of customer including their detail.
if input is 2
Enter customer name
Enter customer number
Enter customer Email
Enter a Kitta no u want to return
Enter the duration
Do customer want to return more land
Generate a bill of customer including their detail.
if input is 3
Thank you for visiting Techno Property Nepal
Stop

Figure 1 : Flow chart

Pseudo code

main.py

```
# Importing necessary modules and functions
DO
   from datetime IMPORT *
   from operation IMPORT *
   from write IMPORT *
   from read IMPORT *
ENDDO

# Creating an empty list for land data
DO
   land_data = read_land_data()
ENDDO

# Displaying a welcome message
DO
   Welcome()
ENDDO

# Initializing a loop flag
DO
   loop = True
ENDDO

# Getting the current date and time
DO
   current_date_time = datetime.now()
ENDDO

WHILE loop == True DO
   # Displaying the menu options
   DO
      display_menu()
   ENDDO

   # Asking the user for their choice
```

```
    DO
        user_choice = get_user_choice()
    ENDDO

    IF user_choice == 1 THEN
        # Renting land
        DO
            rent_land(land_data)
        ENDDO

    ELSE IF user_choice == 2 THEN
        # Returning rented land
        DO
            return_land(land_data)
        ENDDO

    ELSE IF user_choice == 3 THEN
        # Exiting the system
        DO
            loop = False
        ENDDO
    ELSE
        DO
            print("Invalid choice. Please enter a valid option (1, 2, or 3).")
        ENDDO
    ENDFI
ENDWHILE
```

operation.py

```
FUNCTION rent_land(land_data)
    # Get customer details
    DO
        customer_name, customer_number, customer_email =
get_customer_details()
    ENDDO

    # Get land details to rent
    DO
        kitta_numbers = get_kitta_numbers_to_rent()
    ENDDO
```

```
   # Rent the land
   DO
      rent_land(land_data, kitta_numbers, customer_name,
customer_number, customer_email)
   ENDDO
ENDFUNCTION

FUNCTION return_land(land_data)
   # Get customer details
   DO
      customer_name, customer_number, customer_email =
get_customer_details()
   ENDDO

   # Get land details to return
   DO
      kitta_numbers = get_kitta_numbers_to_return()
   ENDDO

   # Return the land
   DO
      return_land(land_data, kitta_numbers, customer_name,
customer_number, customer_email)
   ENDDO
ENDFUNCTION

FUNCTION read_land_data()
   # Read land data from file
   DO
      land_data = read_land_data_from_file()
   ENDDO
   RETURN land_data
ENDFUNCTION

FUNCTION write_land_data(land_data)
   # Write land data to file
   DO
      write_land_data_to_file(land_data)
   ENDDO
```

```
ENDFUNCTION

FUNCTION display_menu()
   # Display menu options
   DO
      print("--------------------------------------------------")
      print("1. Rent Land")
      print("2. Return Rented Land")
      print("3. Exit")
      print("--------------------------------------------------")
   ENDDO
ENDFUNCTION

FUNCTION get_user_choice()
   # Get user choice
   DO
      user_choice = int(input("Enter your choice: "))
   ENDDO
   RETURN user_choice
ENDFUNCTION

FUNCTION get_customer_details()
   # Get customer details
   DO
      customer_name = input("Enter customer name: ")
      customer_number = int(input("Enter customer number: "))
      customer_email = input("Enter customer email: ")
   ENDDO
   RETURN customer_name, customer_number, customer_email
ENDFUNCTION

FUNCTION get_kitta_numbers_to_rent()
   # Get kitta numbers to rent
   DO
      kitta_numbers = input("Enter the Kitta numbers you want to rent
(separated by commas): ").split(',')
      kitta_numbers = [int(k) for k in kitta_numbers]
   ENDDO
   RETURN kitta_numbers
ENDFUNCTION
```

```
FUNCTION get_kitta_numbers_to_return()
   # Get kitta numbers to return
   DO
      kitta_numbers = input("Enter the Kitta numbers you want to return
(separated by commas): ").split(',')
      kitta_numbers = [int(k) for k in kitta_numbers]
   ENDDO
   RETURN kitta_numbers
ENDFUNCTION
```

write.py

```
FUNCTION write_land_data_to_file(land_data)
   # Write land data to file
   DO
      OPEN "land_data.txt" AS file
      FOR EACH land_info IN land_data DO
         land_info_str = ','.join(land_info)
         file.write(land_info_str + '\n')
      ENDFOR
      CLOSE file
   ENDDO
ENDFUNCTION
```

read.py

```
FUNCTION read_land_data_from_file()
   # Read land data from file
   DO
      DECLARE land_data AS LIST
      OPEN "land_data.txt" AS file
      land_data = []
      FOR EACH line IN file DO
         land_info = line.strip().split(',')
         land_data.append(land_info)
      ENDFOR
      CLOSE file
   ENDDO
   RETURN land_data
ENDFUNCTION
```

Data Structure

a) Dictionary: Within the scope of land data, a dictionary can be used to store specific information of each plot of land assigned to be represented as a key string which is unique numerically. Let us say the key is the plot number, and the value for that key would be a dictionary for example with field structure which would contain owner name, area, location, and type (residential, commercial, agricultural, etc.). Dictionaries equip you with swift access to land properties which may be found by entering their id's, and as a good example such information can be obtained with ease.

b) Tuple: Tuples are because they could be used to mark either a coordinate or boundary around the land plot. Let's say a tuple is (x, y) representing the geographical coordinates of a plot or a tuple (length, width) for the dimensions of a rectangle plot. Since tuples are immutable, they therefore may come in handy to be used as a data store for the fixed size data like coordinates and dimensions which should not be changed.

c) List: In the other hand, lists are a good instrument to deal with a set of landmarks. On the other hand, the list can include types of crops grown on the agricultural land, services offered on the house plot and zoning regulations of the commercial plot straight on. The lists allow for their sequential storage and manipulation of the data. So, they are used for the scenarios where the order of the elements may be important, like storing the list of features that relate to each plot area.

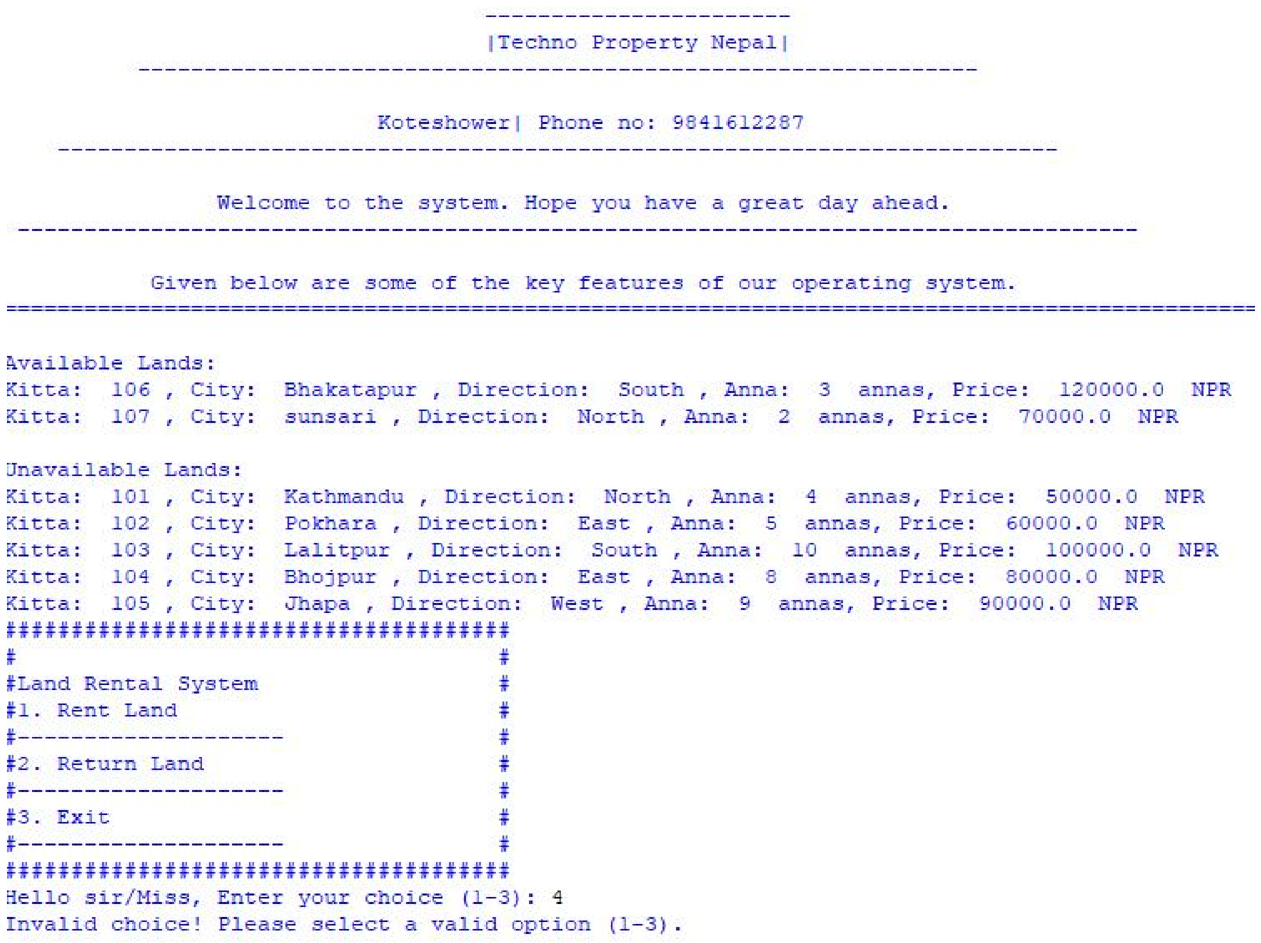
d) Set: Sets might be utilized in order to impart to land plots especially objects of a specific kind. For instance, an array of identifiers designating various elements like lakes, rivers or forests could be used for each one of these plots. The descriptor sets uniquely enable the inclusion of a certain environmental feature only once in the overall picture without duplicating, thus presenting a detailed and easily recognizable geographical image.

Through portraying different kinds of land data structures properly, land information can be routed more effectively thus, enhancing quick access, manipulation and analysis.

Testing

Testing 1: To demonstrate implementation of try, except

Objective	To demonstrate try except
Action:	- The main.py is executed and then non existing value is entered in place of choosing option.
Expected Result:	The error message would be displayed and it would request the user to re-enter the option.
Actual Result:	The error message was displayed.
Conclusion:	The test is successful.

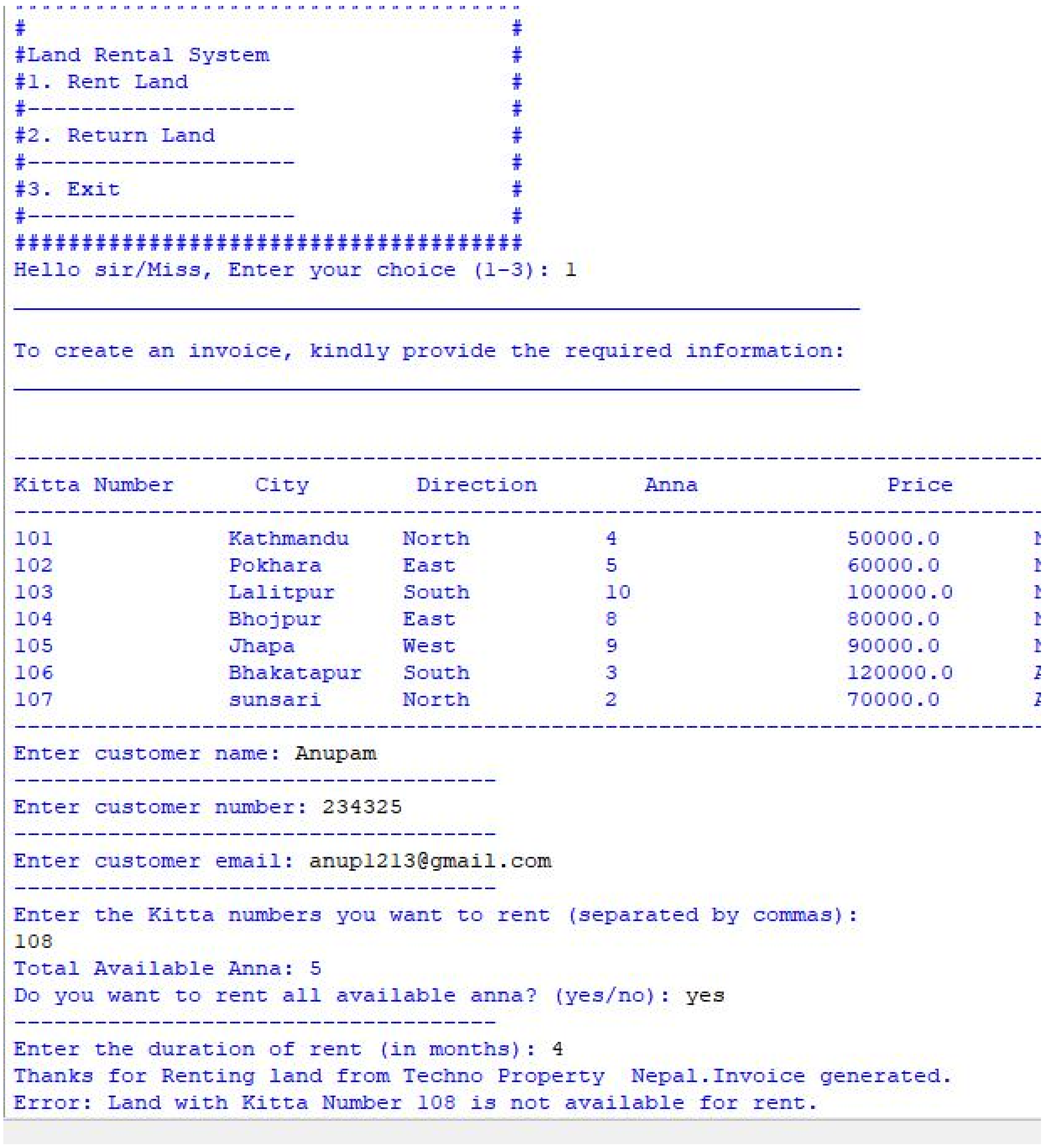


```
                                  -----------------------
                                  |Techno Property Nepal|
          ----------------------------------------------------------------

                          Koteshower| Phone no: 9841612287
    ----------------------------------------------------------------------------

                 Welcome to the system. Hope you have a great day ahead.
 -------------------------------------------------------------------------------------

           Given below are some of the key features of our operating system.
==============================================================================================

Available Lands:
Kitta:  106 , City:  Bhakatapur , Direction:  South , Anna:  3  annas, Price:  120000.0  NPR
Kitta:  107 , City:  sunsari , Direction:  North , Anna:  2  annas, Price:  70000.0  NPR

Unavailable Lands:
Kitta:  101 , City:  Kathmandu , Direction:  North , Anna:  4  annas, Price:  50000.0  NPR
Kitta:  102 , City:  Pokhara , Direction:  East , Anna:  5  annas, Price:  60000.0  NPR
Kitta:  103 , City:  Lalitpur , Direction:  South , Anna:  10  annas, Price:  100000.0  NPR
Kitta:  104 , City:  Bhojpur , Direction:  East , Anna:  8  annas, Price:  80000.0  NPR
Kitta:  105 , City:  Jhapa , Direction:  West , Anna:  9  annas, Price:  90000.0  NPR
######################################
#                                    #
#Land Rental System                  #
#1. Rent Land                        #
#--------------------                #
#2. Return Land                      #
#--------------------                #
#3. Exit                             #
#--------------------                #
######################################
Hello sir/Miss, Enter your choice (1-3): 4
Invalid choice! Please select a valid option (1-3).
```

Figure 2Showing the exception occurred when non existing value is entered.

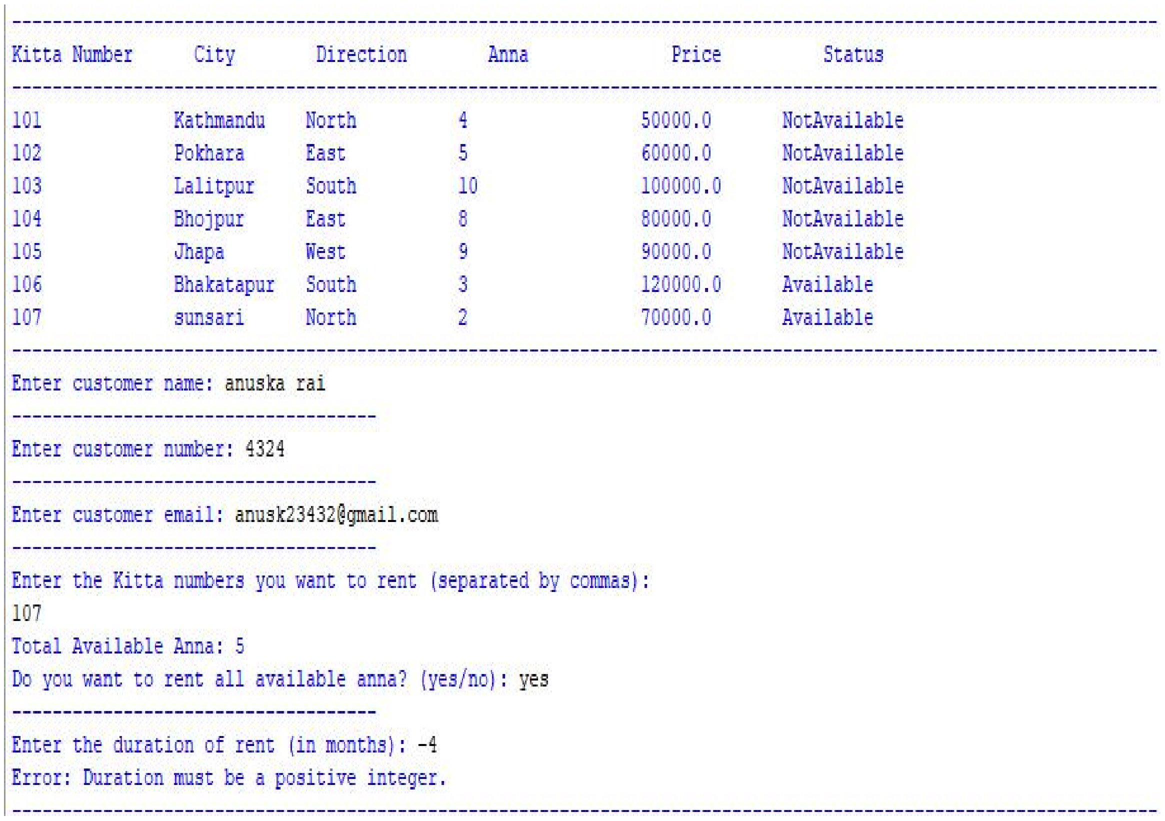
Testing 2:Renting and Returning land with a negative value and non-existing value.

Objective:	To Rent and return land with a negative value and non-existing value.
Action:	The main.py is executed and then option 1 is chosen. Kitta no :108 Duration:-4 Kitta no:109 Kitta no:-110 Duration:-5
Expected Result:	The error message is displayed as -5 is not a positive integer
Actual Result:	The error message was displayed.
Conclusion:	The test is successful.



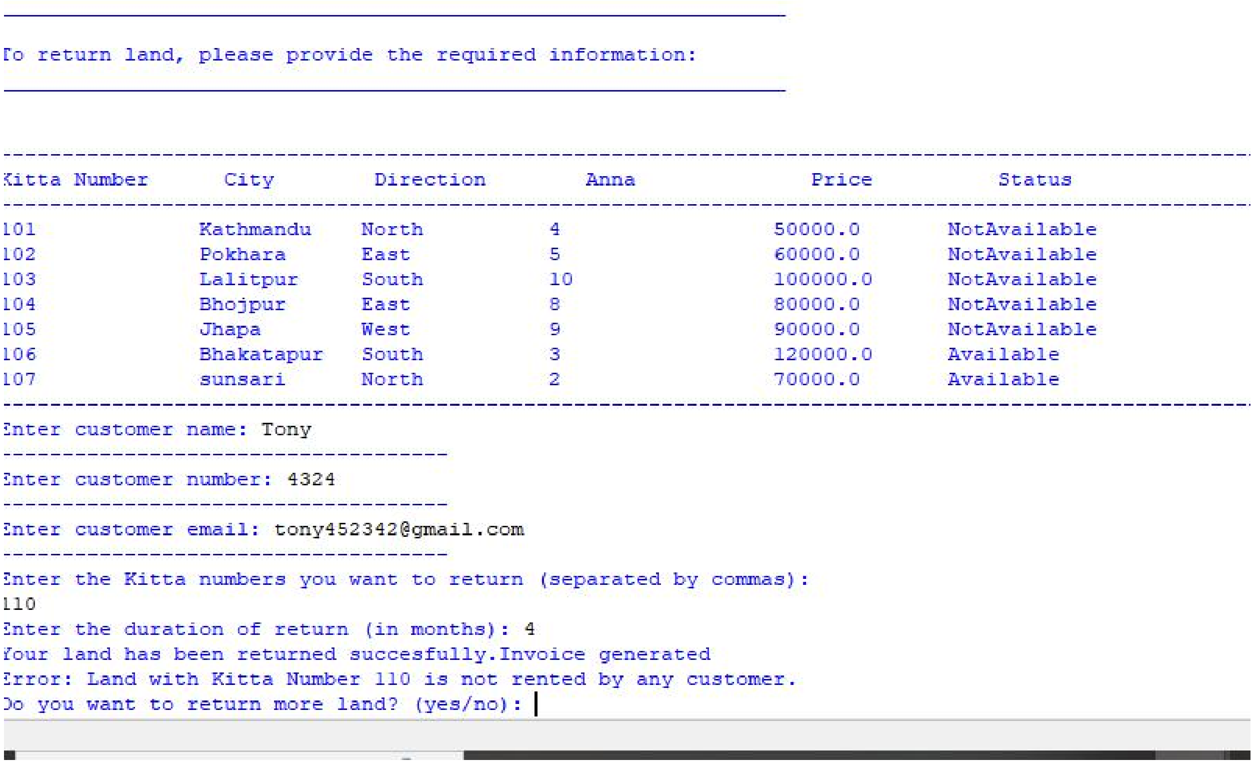
```
#                                     #
#Land Rental System                   #
#1. Rent Land                         #
#--------------------                 #
#2. Return Land                       #
#--------------------                 #
#3. Exit                              #
#--------------------                 #
#######################################
Hello sir/Miss, Enter your choice (1-3): 1
_______________________________________________________________

To create an invoice, kindly provide the required information:
_______________________________________________________________


-----------------------------------------------------------------------------
Kitta Number      City        Direction        Anna             Price
-----------------------------------------------------------------------------
101             Kathmandu    North         4                50000.0       N
102             Pokhara      East          5                60000.0       N
103             Lalitpur     South         10               100000.0      N
104             Bhojpur      East          8                80000.0       N
105             Jhapa        West          9                90000.0       N
106             Bhakatapur   South         3                120000.0      A
107             sunsari      North         2                70000.0       A
-----------------------------------------------------------------------------
Enter customer name: Anupam
------------------------------------
Enter customer number: 234325
------------------------------------
Enter customer email: anup1213@gmail.com
------------------------------------
Enter the Kitta numbers you want to rent (separated by commas):
108
Total Available Anna: 5
Do you want to rent all available anna? (yes/no): yes
------------------------------------
Enter the duration of rent (in months): 4
Thanks for Renting land from Techno Property  Nepal.Invoice generated.
Error: Land with Kitta Number 108 is not available for rent.
```

Figure 3The error message displayed when non-existing kitta no is entered.



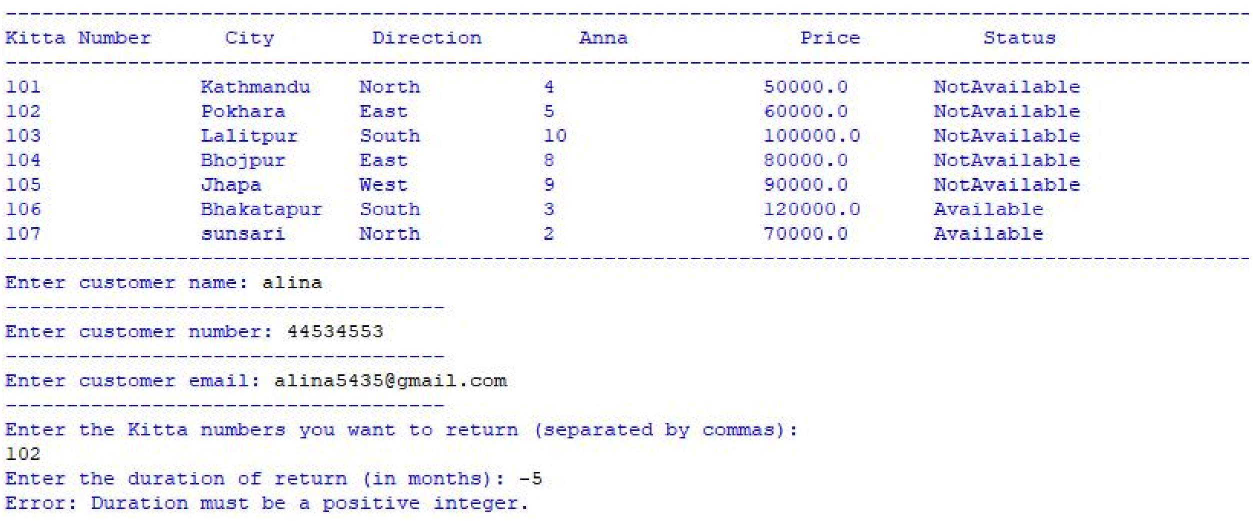
```
-----------------------------------------------------------------------------------------------------------
Kitta Number     City       Direction       Anna             Price         Status
-----------------------------------------------------------------------------------------------------------
101            Kathmandu    North         4                50000.0      NotAvailable
102            Pokhara      East          5                60000.0      NotAvailable
103            Lalitpur     South         10               100000.0     NotAvailable
104            Bhojpur      East          8                80000.0      NotAvailable
105            Jhapa        West          9                90000.0      NotAvailable
106            Bhakatapur   South         3                120000.0     Available
107            sunsari      North         2                70000.0      Available
-----------------------------------------------------------------------------------------------------------
Enter customer name: anuska rai
------------------------------------
Enter customer number: 4324
------------------------------------
Enter customer email: anusk23432@gmail.com
------------------------------------
Enter the Kitta numbers you want to rent (separated by commas):
107
Total Available Anna: 5
Do you want to rent all available anna? (yes/no): yes
------------------------------------
Enter the duration of rent (in months): -4
Error: Duration must be a positive integer.
-----------------------------------------------------------------------------------------------------------
```

Figure 4: The error message displayed when negative quantity is entered.



```
______________________________________________________________

To return land, please provide the required information:
______________________________________________________________

--------------------------------------------------------------------------------------------------
Kitta Number      City        Direction        Anna              Price          Status
--------------------------------------------------------------------------------------------------
101            Kathmandu    North          4                50000.0       NotAvailable
102            Pokhara      East           5                60000.0       NotAvailable
103            Lalitpur     South          10               100000.0      NotAvailable
104            Bhojpur      East           8                80000.0       NotAvailable
105            Jhapa        West           9                90000.0       NotAvailable
106            Bhakatapur   South          3                120000.0      Available
107            sunsari      North          2                70000.0       Available
--------------------------------------------------------------------------------------------------
Enter customer name: Tony
------------------------------------
Enter customer number: 4324
------------------------------------
Enter customer email: tony452342@gmail.com
------------------------------------
Enter the Kitta numbers you want to return (separated by commas):
110
Enter the duration of return (in months): 4
Your land has been returned succesfully.Invoice generated
Error: Land with Kitta Number 110 is not rented by any customer.
Do you want to return more land? (yes/no):
```

Figure 5:The error message displayed when user enters non-existing value after choosing option 2 (i.e.) To Return land

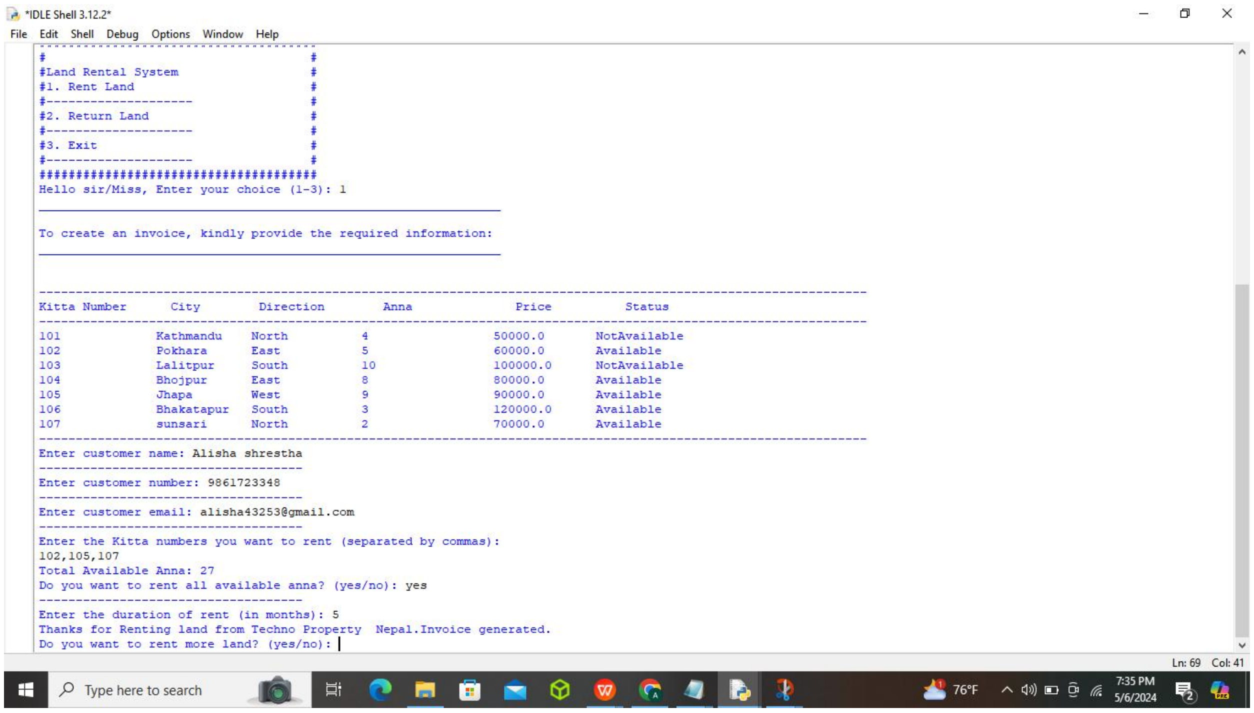


```
--------------------------------------------------------------------------------------------------
Kitta Number      City        Direction        Anna              Price          Status
--------------------------------------------------------------------------------------------------
101            Kathmandu    North          4                50000.0       NotAvailable
102            Pokhara      East           5                60000.0       NotAvailable
103            Lalitpur     South          10               100000.0      NotAvailable
104            Bhojpur      East           8                80000.0       NotAvailable
105            Jhapa        West           9                90000.0       NotAvailable
106            Bhakatapur   South          3                120000.0      Available
107            sunsari      North          2                70000.0       Available
--------------------------------------------------------------------------------------------------
Enter customer name: alina
------------------------------------
Enter customer number: 44534553
------------------------------------
Enter customer email: alina5435@gmail.com
------------------------------------
Enter the Kitta numbers you want to return (separated by commas):
102
Enter the duration of return (in months): -5
Error: Duration must be a positive integer.
```

Figure 6The error message displayed when the user enters negative value after choosing option 2 (i.e.) that is returning land.

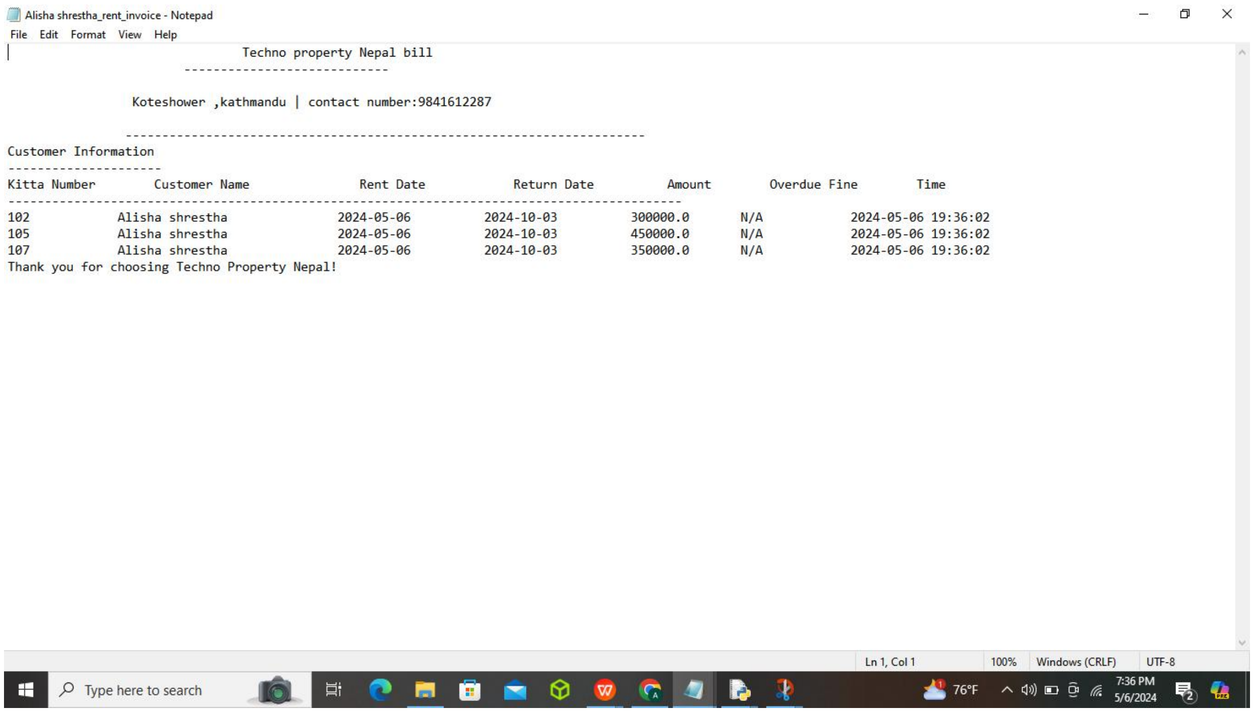
Testing 3:Renting lands

Objective:	To rent land and generate the invoice in a text file and shell.
Action:	The main.py is executed and then option 1 is chosen. Name:Alisha shrestha Kitta no:102,105,107
Expected Result:	The invoice would be generated in both the shell and text file.
Actual Result:	The invoice is generated in both the shell and text file.
Conclusion:	The test is successful.



```
#                                   #
#Land Rental System                 #
#1. Rent Land                       #
#--------------------               #
#2. Return Land                     #
#--------------------               #
#3. Exit                            #
#--------------------               #
#####################################
Hello sir/Miss, Enter your choice (1-3): 1
_______________________________________________________________

To create an invoice, kindly provide the required information:
_______________________________________________________________


-------------------------------------------------------------------------------------------------------------------
Kitta Number      City        Direction        Anna              Price          Status
-------------------------------------------------------------------------------------------------------------------
101             Kathmandu    North          4                 50000.0       NotAvailable
102             Pokhara      East           5                 60000.0       Available
103             Lalitpur     South          10                100000.0      NotAvailable
104             Bhojpur      East           8                 80000.0       Available
105             Jhapa        West           9                 90000.0       Available
106             Bhakatapur   South          3                 120000.0      Available
107             sunsari      North          2                 70000.0       Available
-------------------------------------------------------------------------------------------------------------------
Enter customer name: Alisha shrestha
------------------------------------
Enter customer number: 9861723348
------------------------------------
Enter customer email: alisha43253@gmail.com
------------------------------------
Enter the Kitta numbers you want to rent (separated by commas):
102,105,107
Total Available Anna: 27
Do you want to rent all available anna? (yes/no): yes
------------------------------------
Enter the duration of rent (in months): 5
Thanks for Renting land from Techno Property  Nepal.Invoice generated.
Do you want to rent more land? (yes/no): 
```

Figure 7:choosing option 1 for renting multiple land.

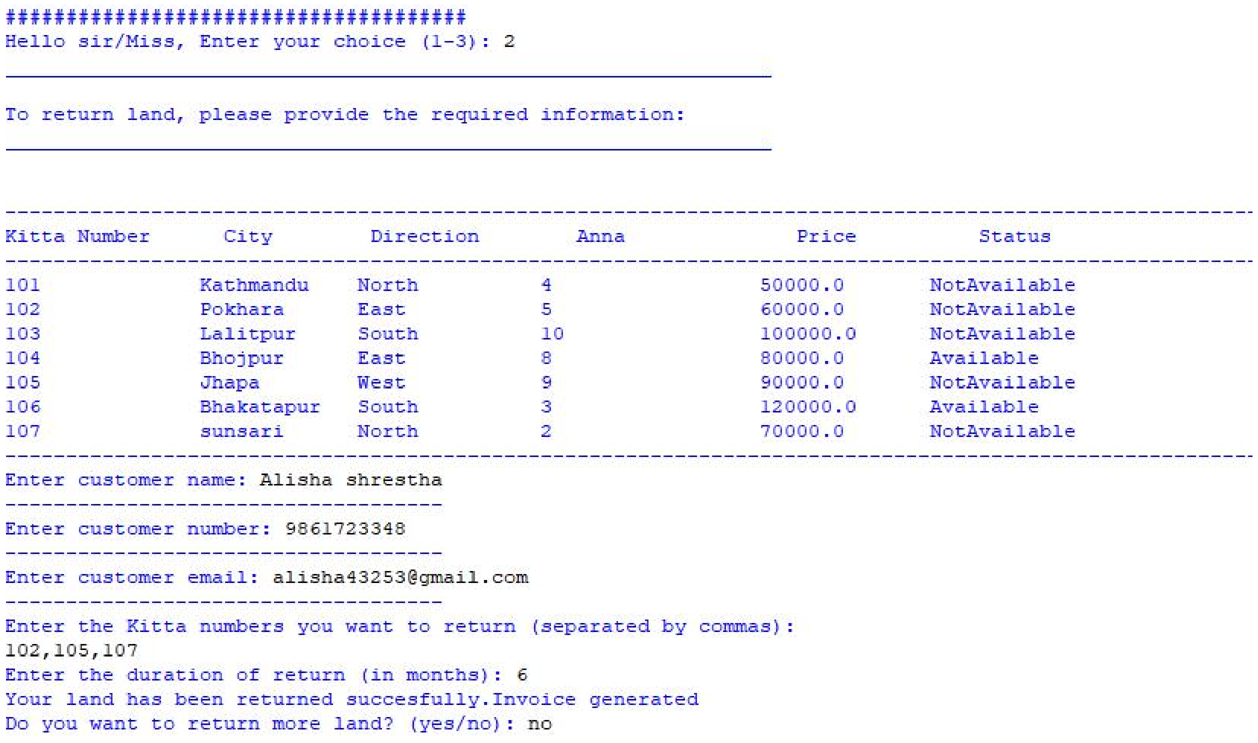


```
                                Techno property Nepal bill
                        ----------------------------


                 Koteshower ,kathmandu | contact number:9841612287


                ----------------------------------------------------------------------
Customer Information
---------------------
Kitta Number        Customer Name                Rent Date           Return Date          Amount        Overdue Fine        Time
-------------------------------------------------------------------------------------------
102           Alisha shrestha               2024-05-06          2024-10-03          300000.0       N/A             2024-05-06 19:36:02
105           Alisha shrestha               2024-05-06          2024-10-03          450000.0       N/A             2024-05-06 19:36:02
107           Alisha shrestha               2024-05-06          2024-10-03          350000.0       N/A             2024-05-06 19:36:02
Thank you for choosing Techno Property Nepal!
```

Figure 8 : Invoice of renting multiple land in .txt file.

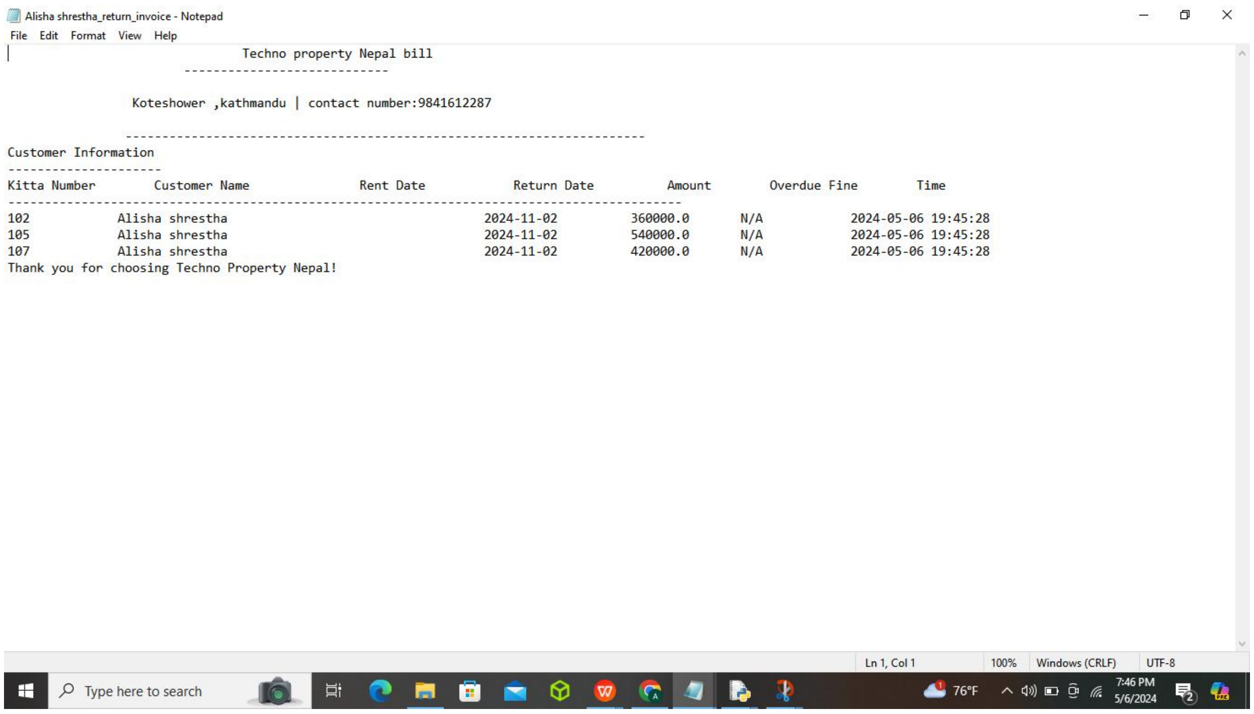
Testing4: Returning lands

Objective:	To Return land and generate invoices in both the text file and shell.
Action:	The main.py is executed and then option 2 is chosen. Customer name:Alisha shrestha Kitta no:102,105,107
Expected Result:	The invoice would be generated in both the shell and text file.
Actual Result:	The invoice was generated in both the shell and text file
Conclusion:	The test is successful.



```
#######################################
Hello sir/Miss, Enter your choice (1-3): 2
_________________________________________________________________

To return land, please provide the required information:
_________________________________________________________________


----------------------------------------------------------------------------------------------------------
Kitta Number      City        Direction        Anna              Price           Status
----------------------------------------------------------------------------------------------------------
101            Kathmandu    North          4                 50000.0       NotAvailable
102            Pokhara      East           5                 60000.0       NotAvailable
103            Lalitpur     South          10                100000.0      NotAvailable
104            Bhojpur      East           8                 80000.0       Available
105            Jhapa        West           9                 90000.0       NotAvailable
106            Bhakatapur   South          3                 120000.0      Available
107            sunsari      North          2                 70000.0       NotAvailable
----------------------------------------------------------------------------------------------------------
Enter customer name: Alisha shrestha
------------------------------------
Enter customer number: 9861723348
------------------------------------
Enter customer email: alisha43253@gmail.com
------------------------------------
Enter the Kitta numbers you want to return (separated by commas):
102,105,107
Enter the duration of return (in months): 6
Your land has been returned succesfully.Invoice generated
Do you want to return more land? (yes/no): no
```

Figure 9:choosing option 2 for returning multiple land.

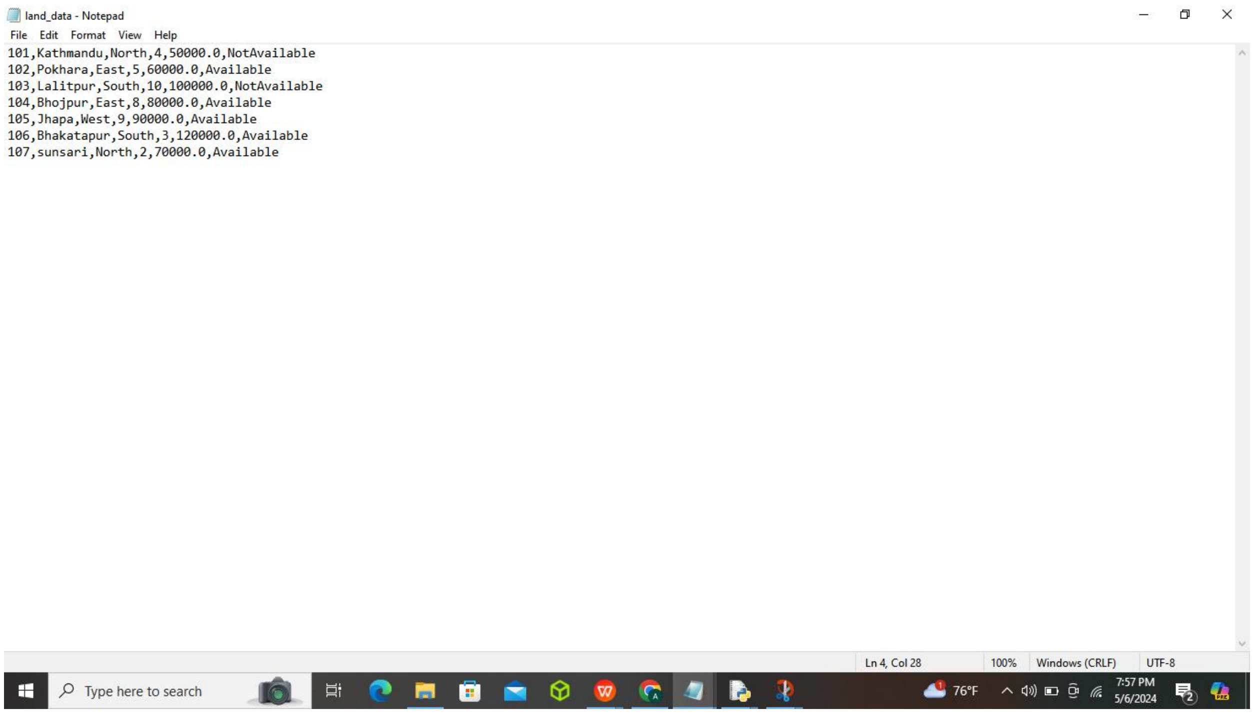


```
                              Techno property Nepal bill
                       ---------------------------

                Koteshower ,kathmandu | contact number:9841612287

                ---------------------------------------------------------------------
Customer Information
--------------------
Kitta Number       Customer Name              Rent Date            Return Date          Amount        Overdue Fine        Time
------------------------------------------------------------------------------------------
102           Alisha shrestha                                 2024-11-02        360000.0       N/A            2024-05-06 19:45:28
105           Alisha shrestha                                 2024-11-02        540000.0       N/A            2024-05-06 19:45:28
107           Alisha shrestha                                 2024-11-02        420000.0       N/A            2024-05-06 19:45:28
Thank you for choosing Techno Property Nepal!
```

Figure 10: invoice of returning multiple land in .txtfile

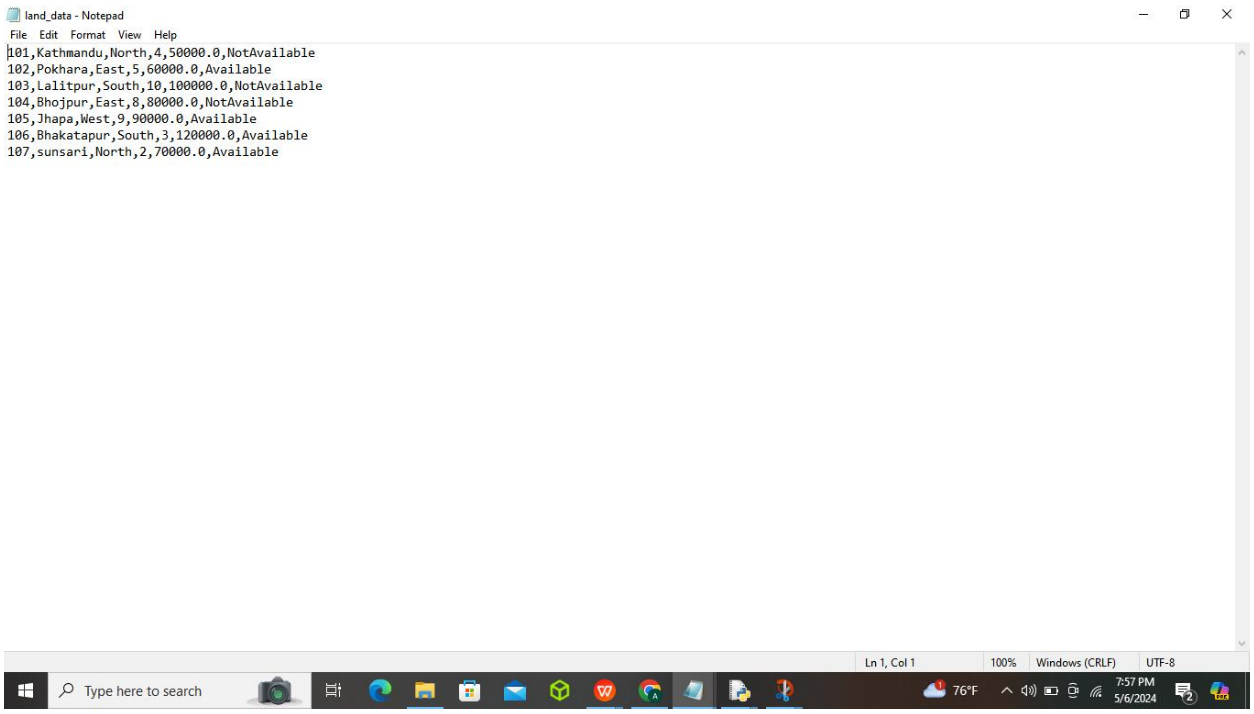
Testing 5: Update of land in .txt file before and after renting and returning land

Objective:	Updating .txt file Available to Not Available Or Not available to Available
Action:	Kitta no:104
Expected Result:	.txt file should be updated from Available to Not AvailableOr Not available to Available
Actual Result:	.txt file is updated from Available to Not Available Or Not available to Available
Conclusion:	The .txt file is update accordingly



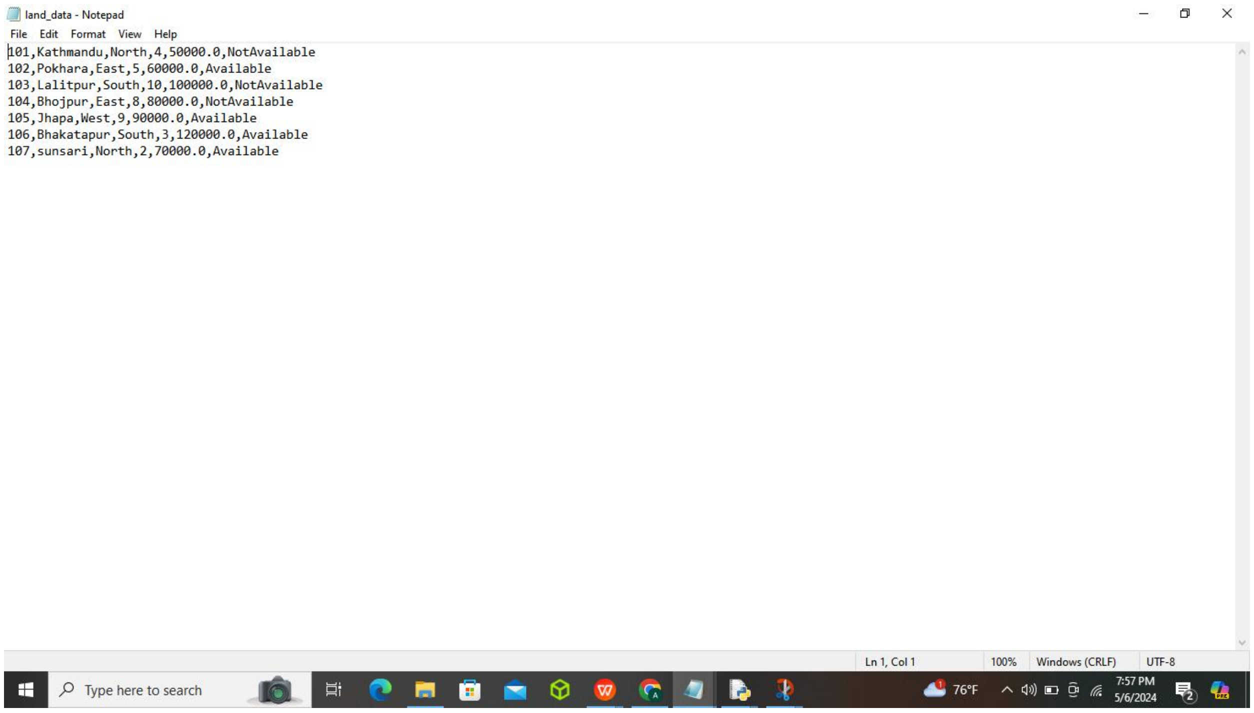
```
101,Kathmandu,North,4,50000.0,NotAvailable
102,Pokhara,East,5,60000.0,Available
103,Lalitpur,South,10,100000.0,NotAvailable
104,Bhojpur,East,8,80000.0,Available
105,Jhapa,West,9,90000.0,Available
106,Bhakatapur,South,3,120000.0,Available
107,sunsari,North,2,70000.0,Available
```

Figure 11 :Before renting land txt.file



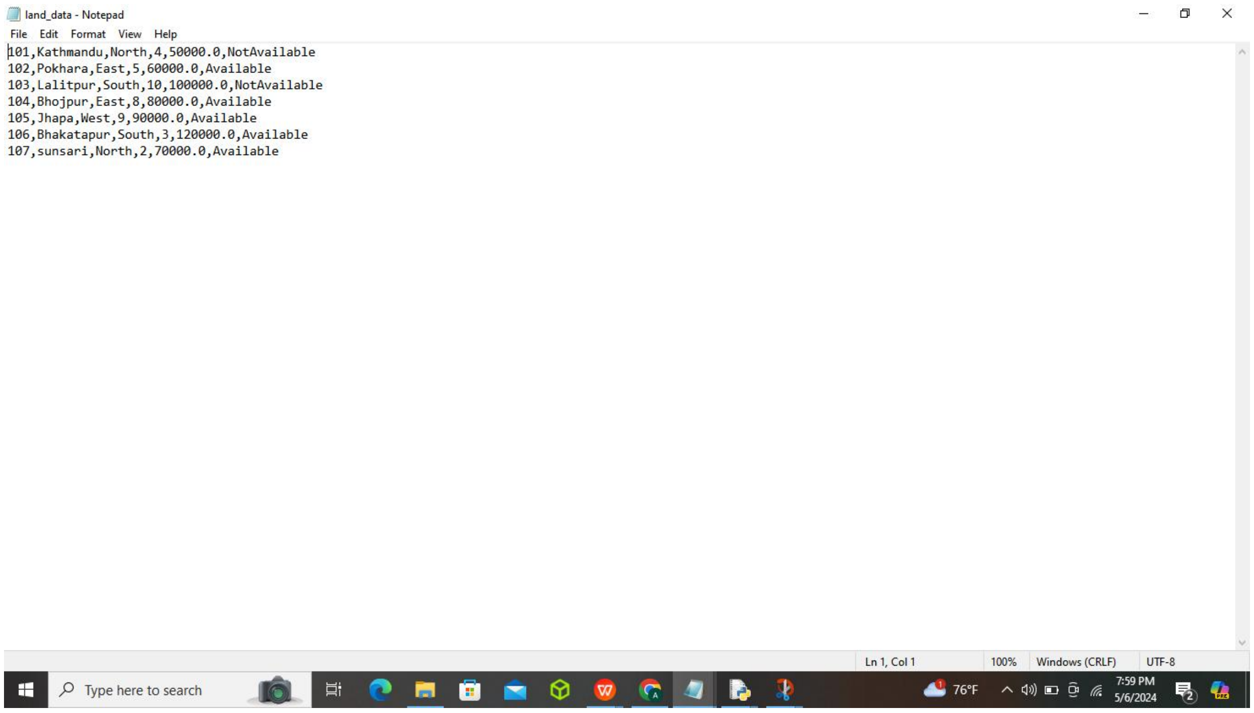
```
101,Kathmandu,North,4,50000.0,NotAvailable
102,Pokhara,East,5,60000.0,Available
103,Lalitpur,South,10,100000.0,NotAvailable
104,Bhojpur,East,8,80000.0,NotAvailable
105,Jhapa,West,9,90000.0,Available
106,Bhakatapur,South,3,120000.0,Available
107,sunsari,North,2,70000.0,Available
```

Figure 12:After renting land.txt.file.



```
101,Kathmandu,North,4,50000.0,NotAvailable
102,Pokhara,East,5,60000.0,Available
103,Lalitpur,South,10,100000.0,NotAvailable
104,Bhojpur,East,8,80000.0,NotAvailable
105,Jhapa,West,9,90000.0,Available
106,Bhakatapur,South,3,120000.0,Available
107,sunsari,North,2,70000.0,Available
```

Figure 13:Before returning land .txt.file



```
101,Kathmandu,North,4,50000.0,NotAvailable
102,Pokhara,East,5,60000.0,Available
103,Lalitpur,South,10,100000.0,NotAvailable
104,Bhojpur,East,8,80000.0,Available
105,Jhapa,West,9,90000.0,Available
106,Bhakatapur,South,3,120000.0,Available
107,sunsari,North,2,70000.0,Available
```

Figure 14:After returning land.txt.file

Conclusion

In conclusion, the creation of the land rental system in Python ensures the operations are running smoothly and efficiently to avoid a loss of profit when managing landholdings. This tool enables seamless communication via multichannel, ensures timely transactions, and performs stock-keeping functions which make operations in the company smooth leading to satisfied users.

The smart-problem interface in land rental system considerably makes the transactions easy and saves time. Owners of property can easily list their currently available inventory in the system, and all crucial elements like location, size, and value should be necessarily shown Queries by the potential renters, either a person or a business, can easily be viewed at the available land listings and more details can be scrutinized as well as rental requests can be directly communicated to the property owners through the listings.

As a result, the plots crops built in Python stand as a reliable and effective information hub for land leasing and management. This is made possible thanks to the interface that is friendly for the users and tracking of inventory in real time. Its safe, speedy transaction process that reflects the high level of efficiency, productivity, and positive feedback from the customers.

Appendix

Main.py

```
import datetime
import read
import operation
import write

def display_land_data(land_data):
    # Display available lands
    print("Available Lands:")
    for land in land_data:
        if land['status'] == 'Available':
            print("Kitta: ", land['kitta'], ", City: ", land['city'],
", Direction: ", land['direction'], ", Anna: ", land['area'],
" annas, Price: ", land['price'], " NPR")
    print("\nUnavailable Lands:")
    # Display unavailable lands
    for land in land_data:
        if land['status'] == 'NotAvailable':
            print("Kitta: ", land['kitta'], ", City: ", land['city'],
", Direction: ", land['direction'], ", Anna: ", land['area'],
" annas, Price: ", land['price'], " NPR")

def main():
    land_data_file = 'land_data.txt'
    land_data = read.read_land_data(land_data_file)
    rented_lands = {}  # Dictionary to store rented
lands with customer details
    all_invoice_data = []  # List to store invoice data
for all transactions

    while True:
        print("                                  ----------------------")
        print("                                  |Techno Property
Nepal|")
        print("         --------------------------------------------
----------------")
        print("")
        print("                          Koteshower| Phone no:
9841612287")
        print("    ----------------------------------------------
----------------------")
        print("")
        print("             Welcome to the system. Hope
you have a great day ahead.")
        print(" ------------------------------------------------
----------------------------")
        print("")
        print("          Given below are some of the key
features of our operating system.")

print("====================================
=======================================
========================================
========================================
=======")
    print("    ")

    # Display land data
    display_land_data(land_data)

    while True:

print("########################################
#")
        print("#                              #")
        print("#Land Rental System            #")
        print("#1. Rent Land                  #")
        print("#--------------------          #")
        print("#2. Return Land                #")
        print("#--------------------          #")
        print("#3. Exit                       #")
        print("#--------------------          #")

print("########################################
#")
        choice = input("Hello sir/Miss, Enter your
choice (1-3): ")

        if choice == '1':
            # Rent Land
```

```
print("________________________________________
____________________________")
            print("
")
            print("To create an invoice, kindly provide
the required information: ")

print("________________________________________
____________________________")
            print("\n")
            invoice_data =
operation.rent_land(land_data, rented_lands)
            all_invoice_data.extend(invoice_data)

         elif choice == '2':
            # Return Land

print("________________________________________
____________________________")
            print("
")
            print("To return land, please provide the
required information: ")

print("________________________________________
____________________________")
            print("\n")
            invoice_data =
operation.return_land(land_data, rented_lands)
            all_invoice_data.extend(invoice_data)

         elif choice == '3':
            # Exit
            print("\n")

print("********************************************")
            print("*                                  *")
            print("* Thank you for using Techno
Property Nepal *")
            print("*                                  *")

print("********************************************")
            break

         else:
            print("Invalid choice! Please select a valid
option (1-3).\n")

      exit_choice = input("Do you want to exit the
program? (yes/no): ")
      if exit_choice.lower() == 'yes':
         break

   # Generate a single invoice for all transactions
```

```
    customer_name = input("Enter customer name for
generating invoice: ")
    operation.generate_invoice(customer_name,
all_invoice_data)

if __name__ == "__main__":
    main()
```

Operation.py

```
import datetime
import write

def display_lands(land_data):
    # Display header
    print("-------------------------------------------------
-----------------------------------------------")
    print("Kitta Number      City        Direction
Anna              Price          Status         ")
    print("-------------------------------------------------
-----------------------------------------------")

    # Display land data
    for land in land_data:
        kitta_number = str(land['kitta']).ljust(16)
        city = land['city'].ljust(13)
        direction = land['direction'].ljust(15)
        area = str(land['area']).ljust(18)
        price = str(land['price']).ljust(14)
        status = land['status']
        print(kitta_number + city + direction + area +
price + status)

    # Display footer
    print("------------------------------------------------------------
---------------------------------------------------------")

def rent_land(land_data, rented_lands):
    # Get current date and time
    current_date = datetime.datetime.now()
    all_invoice_data = []

    # Loop for renting land
    while True:
        try:
            # Display available lands
            display_lands(land_data)

            # Input customer details
            customer_name = input("Enter customer
name: ")
            print("--------------------------------------")
            if any(char.isdigit() for char in
customer_name):
```

```
            print("Error: Customer name cannot
contain numbers.")
            continue
        customer_number = int(input("Enter
customer number: "))
        print("-----------------------------------")
        customer_email = input("Enter customer
email: ")
        print("-----------------------------------")

        # Input kitta numbers to rent
        print("Enter the Kitta numbers you want to
rent (separated by commas):")
        kitta_numbers = [int(x) for x in input().split(',')]
        total_available_anna = sum(land['area'] for
land in land_data if land['status'] == 'Available')

        # Confirm renting all available anna
        print("Total Available Anna: " +
str(total_available_anna))
        confirmation = input("Do you want to rent all
available anna? (yes/no): ").lower()
        print("-----------------------------------")

        if confirmation != 'yes':
            print("Error: You must rent all the available
anna.")
            continue
        # Input duration of rent
        duration_months = int(input("Enter the
duration of rent (in months): "))
        if duration_months <= 0:
            print("Error: Duration must be a positive
integer.")
            continue
        print("Thanks for Renting land from Techno
Property  Nepal.Invoice generated.")

        # Generate invoice data for each rented land
        invoice_data_list = []
        for kitta_number in kitta_numbers:
            land = next((land for land in land_data if
land['kitta'] == kitta_number and land['status'] ==
'Available'), None)
            if land:
                land['status'] = 'NotAvailable'
                land['customer_name'] =
customer_name
                return_date = current_date +
datetime.timedelta(days=duration_months * 30)
                invoice_data = {
                    'Kitta Number': kitta_number,
                    'Customer Name': customer_name,
                    'Customer Number':
customer_number,
                    'Customer Email': customer_email,
                    'Rent Date':
current_date.strftime('%Y-%m-%d'),
                    'Return Date':
return_date.strftime('%Y-%m-%d'),
                    'Amount': land['price'] *
duration_months
                }
                invoice_data_list.append(invoice_data)
                rented_lands[kitta_number] =
{'customer_name': customer_name,
'customer_number': customer_number,
'customer_email': customer_email}
            else:
                print("Error: Land with Kitta Number " +
str(kitta_number) + " is not available for rent.")

        all_invoice_data.extend(invoice_data_list)

        more = input("Do you want to rent more land?
(yes/no): ")
        if more.lower() != 'yes':
            break
    except ValueError:
        print("Error: Invalid input data.")

# Write invoice data to file
invoice_file = str(customer_name) +
"_rent_invoice.txt"
write.write_invoice(invoice_file, all_invoice_data)

# Update land data file
write_land_data(land_data)
return all_invoice_data

def return_land(land_data, rented_lands):
# Get current date and time
current_date = datetime.datetime.now()
all_invoice_data = []

# Loop for returning land
while True:
    try:
        # Display available lands
        display_lands(land_data)

        # Input customer details
        customer_name = input("Enter customer
name: ")
        print("-----------------------------------")
        if any(char.isdigit() for char in
customer_name):
            print("Error: Customer name cannot
contain numbers.")
            continue
        customer_number = int(input("Enter
customer number: "))
        print("-----------------------------------")
        customer_email = input("Enter customer
email: ")
        print("-----------------------------------")

        # Input kitta numbers to return
        print("Enter the Kitta numbers you want to
return (separated by commas):")
        kitta_numbers = [int(x) for x in input().split(',')]
        return_duration_months = int(input("Enter the
duration of return (in months): "))
        if return_duration_months <= 0:
            print("Error: Duration must be a positive
integer.")
            continue
        print("Your land has been returned
succesfully.Invoice generated")

        # Generate invoice data for each returned
land
        invoice_data_list = []
        for kitta_number in kitta_numbers:
            if kitta_number in rented_lands:
                rented_info =
rented_lands[kitta_number]
```

```
                if rented_info['customer_name'] ==
customer_name and rented_info['customer_number']
== customer_number and
rented_info['customer_email'] == customer_email:
                    land = next((land for land in
land_data if land['kitta'] == kitta_number and
land['status'] == 'NotAvailable'), None)
                    if land:
                        land['status'] = 'Available'
                        del land['customer_name']
                        return_date = current_date +
datetime.timedelta(days=return_duration_months *
30)
                        invoice_data = {
                            'Kitta Number': kitta_number,
                            'Rent Date': '',
                            'Return Date':
return_date.strftime('%Y-%m-%d'),
                            'Amount': land['price'] *
return_duration_months,
                            'Customer Name':
customer_name,
                            'Customer Number':
customer_number,
                            'Customer Email':
customer_email
                        }
```

```
invoice_data_list.append(invoice_data)
                  else:
                     print("Error: Land with Kitta
Number " + str(kitta_number) + " is not rented by any
customer.")
               else:
                  print("Error: Customer details do not
match the rented land's details.")
            else:
               print("Error: Land with Kitta Number " +
str(kitta_number) + " is not rented by any customer.")

         all_invoice_data.extend(invoice_data_list)

         more = input("Do you want to return more
land? (yes/no): ")
         if more.lower() != 'yes':
            break
      except ValueError:
         print("Error: Invalid input data.")

   # Write invoice data to file
   invoice_file = str(customer_name) +
"_return_invoice.txt"
   write.write_invoice(invoice_file, all_invoice_data)

   # Update land data file
```

```
    write_land_data(land_data)
    return all_invoice_data

def write_land_data(land_data):
    # Write land data to file
    with open('land_data.txt', 'w') as file:
        for land in land_data:
            status = 'NotAvailable' if land['status'] ==
'NotAvailable' else land['status']
            file.write(str(land['kitta']) + ',' + land['city'] + ','
+ land['direction'] + ',' + str(land['area']) + ',' +
str(land['price']) + ',' + status + '\n')

if __name__ == "__main__":
    # Test the functions if required
    pass
```

Write.py

```
import datetime

def write_invoice(file_name,
invoice_data_list):
    current_time =
datetime.datetime.now().strftime('%Y-
%m-%d %H:%M:%S')# Give current
time

    # Write invoice data to the file
    with open(file_name, 'w') as file:
        file.write("\t\t\t\tTechno property
Nepal bill\n")
        file.write("\t\t\t---------------------------
\n")
        file.write("\n")
        file.write("\t \t
Koteshower ,kathmandu | contact
number:9841612287 \n")
        file.write("\n")
        file.write("\t \t------------------------------
----------------------------------------\n")
        file.write("Customer Information\n")
        file.write("--------------------\n")
        file.write("Kitta Number" + " " * 8 +
"Customer Name" + " " * 15 + "Rent
Date" + " " * 12 + "Return Date" + " " *
```

```
10 + "Amount" + " " * 8 + "Overdue Fine"
+ " " * 8 + "Time" + "\n")
        file.write("--------------------------------
--------------------------------------------
----\n")

        for invoice_data in
invoice_data_list:
            kitta_number =
str(invoice_data['Kitta Number']).ljust(15)
            customer_name =
str(invoice_data['Customer
Name']).ljust(30)
            rent_date =
str(invoice_data['Rent Date']).ljust(20)
            return_date =
str(invoice_data['Return Date']).ljust(20)
            amount =
str(invoice_data['Amount']) if 'Amount' in
invoice_data else 'N/A'
            amount = amount.ljust(15)
            overdue_fine = 
str(invoice_data['Overdue Fine']) if 
'Overdue Fine' in invoice_data else 'N/A'
            overdue_fine = 
overdue_fine.ljust(15)
            time = current_time.ljust(20)

            file.write(kitta_number + 
customer_name + rent_date + 
return_date + amount + overdue_fine + 
time + "\n")

        file.write("Thank you for choosing 
Techno Property Nepal!")#write footer
```

read.py

```
def read_land_data(file_path):
    land_data = []  # Initialize an empty 
list to store land data
    try:
        with open(file_path, 'r') as file:  #
Open the file for reading
            for line in file:  # Iterate over
each line in the file
                # Split the line by comma and
assign values to variables
                kitta, city, direction, area, price,
status = line.replace('\n', '').split(',')
                # Append a dictionary
representing land data to the list
                land_data.append({
                    'kitta': int(kitta),  # Convert
kitta to integer
                    'city': city.replace(' ', ''),  #
Remove leading and trailing
whitespaces from city
                    'direction':
direction.replace(' ', ''),  # Remove
leading and trailing whitespaces from
direction
```

```
                    'area': int(area),  # Convert
area to integer
                    'price': float(price),  #
Convert price to float
                    'status': status.replace(' ', '')
                })
    except FileNotFoundError:
        print("Error: Land data file not
found.")  # Handle FileNotFoundError
    except ValueError:
        print("Error: Incorrect data format in
the land data file.")  # Handle ValueError
    return land_data  # Return the list of
land data
```