CS4051NI Fundamentals of Computing

60% Individual Coursework

2023/24 Spring

Student Name: Pratik Man Pradhan

London Met ID: 23048640

College ID: NP01AI4A230084

Assignment Due Date: Tuesday, May 7, 2024

Assignment Submission Date: Monday, May 6, 2024

Word Count: 242

Project File Links:

YouTube Link:	**NO**
Google Drive Link:	**NO**

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 a marks of zero will be awarded.

Table of Contents

List of Figures

List of Tables

INTRODUCTION

An effective land management system is an essential real estate component for efficiently organizing and monitoring land-related activities, such as rentals, returns, and inventory tracking. This report offers valuable insights into a project that aims to develop a land management system, designed to streamline the process of renting and returning lands. By enabling efficient transactions, precise record-keeping, and enhanced communication between landowners and renters, this system intends to improve overall efficiency and productivity.

This system offers a user-friendly, organized platform for renting and returning land and tracking the land availability keeping the information transparent with the user. It includes functionality such as automatically generating invoices while renting and returning, maintaining land availability records, and charging fines on overdue returns. By digitizing these processes, this land management system reduces errors, increases efficiency, and provides a user-friendly interface.

Goals And Objectives

The main focus while developing this system was to create a robust land management system that facilitates land transactions while ensuring accuracy and reliability in data processing. To achieve this, the project focuses on these several key objectives:

- **Automation and Efficiency**: The system automates the essential tasks related to renting and returning land. By doing this, it reduces the need for manual work, speeds up transaction processing, and lowers the chance of errors.
- **Accurate Land Status Tracking**: The main goal of this system is to give users real-time information about land availability. The system lets users check whether land is available, making operations clear and straightforward.
- **Transparent Financial Transactions**: The system generates invoices for every rental and return transaction, detailing costs, rental durations, and fines (if applicable). This transparency maintains trust among users and an organization.
- **User-Friendly Interaction**: The design of the land management system aims to be intuitive and accessible. Users can navigate through the system with ease, making land rentals and returns straightforward processes.
- **Reliable Data Management**: Accurate record-keeping is critical for any land management system. The project uses solid methods to read, write, and update land information, ensuring the data is accurate and reliable.

Tools Used during the development of this system:

1. **Python IDLE**: Python's Integrated Development and Learning Environment (IDLE) was used to write, test, and debug the project's code. IDLE provides a straightforward interface for coding in Python, with features like syntax highlighting and a built-in interactive shell for testing snippets of code quickly.

Figure 1:Python IDLE logo

2. **Draw.io**: This online diagramming tool was utilized exclusively for flowchart creation. Draw.io helps visually map out the logic and relationships between different components of the system, aiding in the planning and design phases which helped drastically in system development.

Figure 2: Draw.io logo

3. **Microsoft Word**: MS Word served as the primary platform for writing and formatting this project report. It was used for documentation, including creating tables, adding images, and organizing content in a readable format.

Figure 3:Microsoft Word logo

Discussion and Analysis:

Algorithm:

Step 1: Display the welcome message in the terminal.

Step 2: Read the data from the file “land.txt” with the help of function name readDataFromFile(fileName) in read.py file and store it in the variable named data.

Step 3: Create a data dictionary through the function createLandDict(data) of read.py file by passing the parameter data and store it in the variable named landDict.

Step 4: Display the table of land in terminal by calling the function printTableOfLand() of message.py file.

Step 5: Display the user menu in the terminal that instruct the user to enter 1 to rent land, 2 to return land and 3 to exit the program.

Step 6: Ask user to enter the value and store it in the variable called userChoice.

Step 7: If userChoice is 1 then call the function named handleLandRental(landDict, data) of operations.py file, otherwise go to Step 8.

Step 8: If userChoice is 2 then call the function named handleLandReturn(landDict, data) of operations.py file, otherwise go to Step 9.

Step 9: If userChoice is 3 then call the function named printThankYouMessage() and end the program, otherwise go to Step 10.

Step 10: Call the function named printInvalidMessage() of message, and return to Step 4.

Pseudocode:

Main.py:

```
 Declare landManagementSystem():

    printWelcomeMessage()
    data = readDataFromFile("land.txt")
    landDict = createLandDict(data)
    sorted=True

    while sorted == True:


            printTableOfLand()
            output("Enter '1' to rent land")
            output("Enter '2' to return land")
            output("Enter '3' to exit")


            Input userChoice

            if userChoice == 1 then
               operations.handleLandRental(landDict, data)
            else if userChoice == 2 then
               operations.handleLandReturn(landDict, data)
            else if userChoice == 3 then
               message.printThankYouMessage()
               sorted=False
            else:
               message.printInvalidMessage()
            end if

    end while

landManagementSystem():
```

Read.py

```
Declare readDataFromFile(fileName):
    data = []
        open(fileName, 'r') as file:
            for line in file:
                lineData = line.strip().split(',')
                lineData[0] = int(lineData[0])
                lineData[3] = int(lineData[3])
                lineData[4] = int(lineData[4])
                data.append(lineData)
            end for
    return data
```

```
Declare createLandDict(data):
    landDict = {}
        for i = 0 to len(data)-1
            key = i + 1
            value = data[i]
            landDict[key] = value
        end for
    return landDict
```

Write.py

```
Declare updateLandAvailability(landId, newStatus, fileName="land.txt")

        open(fileName, 'r') as file
            lines = file.readlines()

        updatedLines = []
        for line in lines
            lineData = line.strip().split(',')
            currentLandId = int(lineData[0])


            if currentLandId == landId then
                lineData[5] = newStatus
            end if

            updatedLine = ','.join(lineData) + '\n'
            updatedLines.append(updatedLine)

        end for

        open(fileName, 'w') as file
            file.writelines(updatedLines)

Declare writeInvoiceToFile(name, invoiceContent):
:
            uniqueValue = str(datetime.datetime.now().minute +
            datetime.datetime.now().second + datetime.datetime.now().microsecond)
            with open(f"Rent_{name}_{uniqueValue}.txt", 'w') as file:
                file.write(invoiceContent)



Declare writeInvoiceToFileReturn(name, invoiceContent)
        uniqueValue = str(datetime.datetime.now().minute +
datetime.datetime.now().second + datetime.datetime.now().microsecond)
        open(f"Return_{name}_{uniqueValue}.txt", 'w') as file
            file.write(invoiceContent)
```

Message.py

```
Declare printWelcomeMessage()

        Output("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++”)
        Output("            Hello and Welcome to the Land Management System")
        Output("                                Techno Property Nepal")
        Output(“+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++”)

Declare printTableOfLand()
   data = read.readDataFromFile("land.txt")
   landDict = read.createLandDict(data)

   Output(
      "-------------------------------------------------------------------------------------------------------------------------------")
   Output("Land ID", "Kitta No", "City/District Name", "Direction", "Anna", "Price", "Availability Status")
   Output(
      "-------------------------------------------------------------------------------------------------------------------------------")
   for key, value in landDict.items()
      print(key, "    ", value[0], "    ", value[1], "    ", value[2], "    ", value[3], "    ", value[4], "      ", value[5])
   end for
   Output(
      "-------------------------------------------------------------------------------------------------------------------------------")

Declare printThankYouMessage()

        Output("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++”)
        Output("              Thank you for using our Land Management System")
        Output("                          Techno Property Nepal")
        Output("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++”)


Declare printInvalidMessage()
   Output("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
   Output("Invalid input! Please enter a valid choice (1, 2, or 3).")
   Output("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
```

Operations.py

```
Declare handleLandRental(landDict, data)
   rentals = []
   grandTotal = 0
   renterName = ""
   flag=True
   while flag = True
         printTableOfLand()
         Input landId

         if landId <= 0 or landId > len(landDict) then
            Output("Invalid land ID! Please enter a valid ID.")
         Else if
            landStatus = data[landId - 1][5]
            if landStatus.lower() == "available" then
               if renterName == "" then
                  Input renterName

               rentalDate = datetime.datetime.now()
               Input duration


               updateLandAvailability(landDict[landId][0], "Not Available")
               landDict[landId][5] = "Not Available"

               landPrice = landDict[landId][4] * duration
               grandTotal = grandTotal + landPrice


               rentals.append((landId, renterName, duration, rentalDate))

               Input decision
               if decision.lower() == 'n':
                  invoice = generateInvoice(rentals, renterName, rentalDate, landDict,
grandTotal)
                  writeInvoiceToFile(renterName, invoice)
                  Output invoice
                  Flag= False
            Else If:
               print(f"Land ID {landId} is not available for rental.")
               Input decision
               if decision.lower() == 'n':
```

```
                    Flag=False
    End while

Declare handleLandReturn(landDict, data):
    rentalsToReturn = []
    renterName = ""
    totalAmount = 0
    flag=True


        while flag== True:
            printTableOfLand()
            Input landId
            if landId == -1 then
                flag=False
            end if

            if landId <= 0 or landId > len(landDict) then
                Output("Invalid land ID! Please enter a valid ID.")
                continue
            end if

            landDetails = data[landId - 1]
            landStatus = landDetails[5]

            if landStatus.lower() == "not available" then
                if renterName == "" then
                    Input renterName
                End if


                Input rentalDateStr
                rentalDate = parseDate(rentalDateStr)
                Input rentalDuration
                expectedDuration = 30 * rentalDuration

                returnDate = datetime.datetime.now()
                actualDuration = (returnDate - rentalDate).days
                monthlyRate = landDict[landId][4]
                fine = calculateFine(actualDuration, expectedDuration, monthlyRate)


                landRent = rentalDuration * monthlyRate
                totalAmount = totalAmount+landRent + fine

                write.updateLandAvailability(landDict[landId][0], "Available")
```

```
            landDict[landId][5] = "Available"

            rentalsToReturn.append({
                "landId": landId,
                "landDetails": landDetails,
                "rentalDate": rentalDate,
                "returnDate": returnDate,
                "fine": fine,
                "totalAmount": landRent,
            })

            Output(f"Land ID {landId} has been returned and is now available.")
        else
            Output(f"Land ID {landId} is already available or not currently rented.")
        End if
    End while
    if rentalsToReturn !=[] then
        totalFine = sum(rental["fine"] for rental in rentalsToReturn)
        grandTotal = totalAmount + totalFine


        returnInvoice = generateReturnInvoice(renterName, rentalsToReturn, totalFine,
grandTotal)


        writeInvoiceToFileReturn(f"{renterName}_return", returnInvoice)


        Output("Here is your return invoice:")
        Output(returnInvoice)

    else
        print("No lands were returned.")
    end if


Declare parseDate(dateStr)
    return datetime.datetime.strptime(dateStr, "%Y-%m-%d")


Declare calculateFine(actualDuration, expectedDuration, monthlyRate)
  if actualDuration > expectedDuration then
    fine = int(((monthlyRate / 30) * (actualDuration - expectedDuration)))
  else
    fine = 0
  return fine
```

```
Declare generateInvoice(rentals, customerName, rentalDate, landDict, grandTotal)

    invoice = f"""
-----------------------------------------------------------
Invoice for Land Rental
-----------------------------------------------------------

Name of Customer: {customerName}
Date and Time of Rent of Land: {rentalDate}
Total amount for all rented lands: Rs {grandTotal}
-----------------------------------------------------------
"""

    for rental in rentals
        landId, _, duration, _ = rental
        landDetails = landDict[landId]
        amount = landDetails[4]*duration
    end for

        invoice += f"""
Kitta No: {landDetails[0]}
Location: {landDetails[1]}
Direction: {landDetails[2]}
Anna: {landDetails[3]}
Monthly Rate of Land: {landDetails[4]}
Rental Duration: {duration} months
Amount = Rs{amount}
-----------------------------------------------------------
"""
    return invoice


Declare generateReturnInvoice(renterName, rentalsToReturn, totalFine, grandTotal):
    returnInvoice = f"""
-----------------------------------------------------------
Land Return Invoice
-----------------------------------------------------------

Name of Customer: {renterName}
OTotal Fine: Rs {totalFine}
Grand Total: Rs {grandTotal}
```

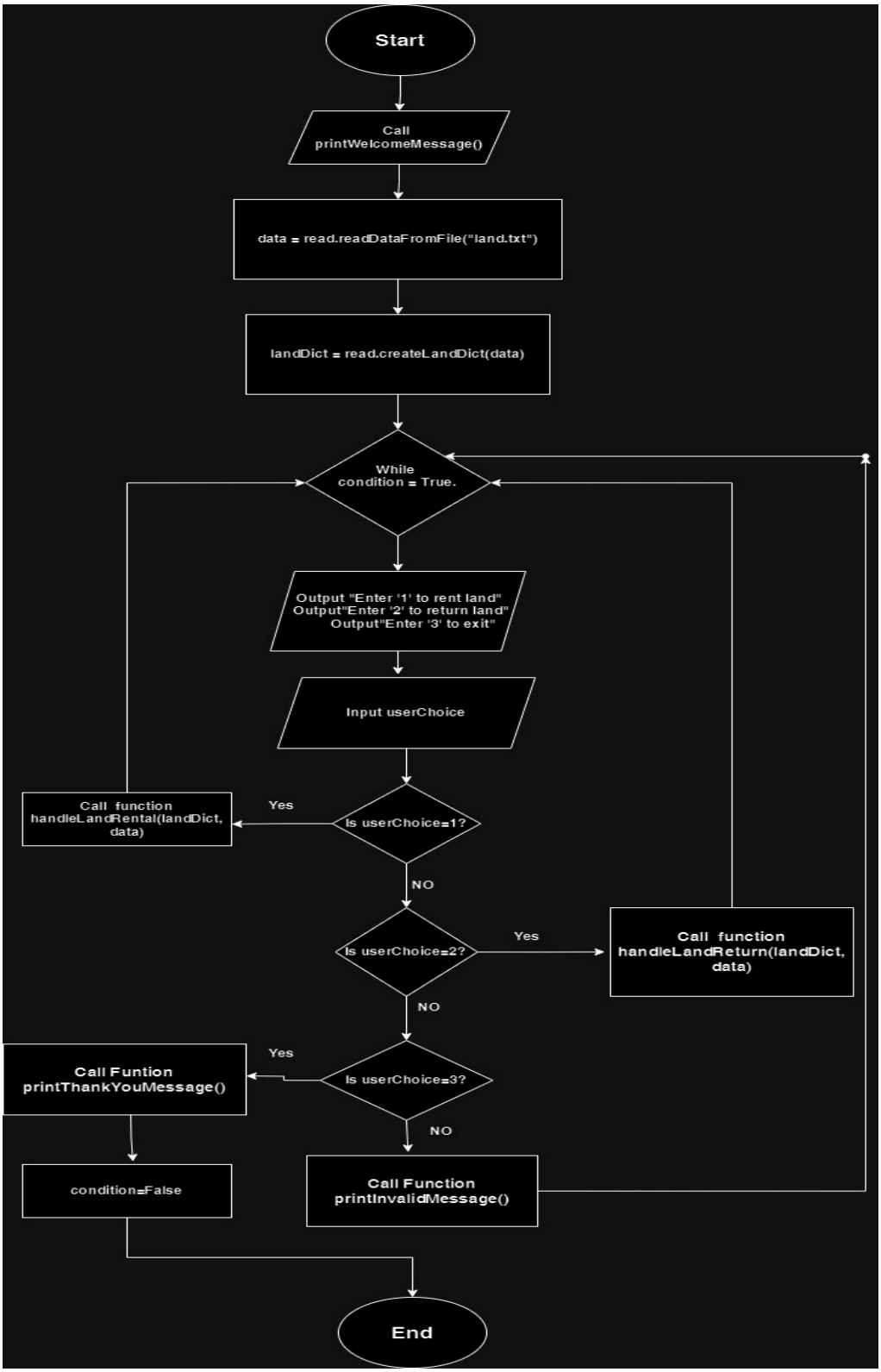
```
-------------------------------------------------------
"""

    for rental in rentalsToReturn
        landDetails = rental["landDetails"]
    end for

        returnInvoice += f"""
Kitta No: {landDetails[0]}
Location: {landDetails[1]}
Direction: {landDetails[2]}
Rental Start Date: {rental["rentalDate"].date()}
Return Date: {rental["returnDate"].date()}
Fine: Rs {rental["fine"]}
Total Amount: Rs {rental["totalAmount"]}
-------------------------------------------------------
"""
    return returnInvoice
```

Flowchart:

Figure 4: Flowchart of main module



Data Structures:

During the development of this project, several data structures came in very handy in the process of organizing, managing, and processing information efficiently. Some of them are:

- **List:** Lists are a fundamental data structure in Python, providing an ordered collection of elements. In this project, lists were used to store and manage multiple related items. This was used in various stages for storing the data read from the file “land.txt” and storing each line as an element and also while renting and returning land to store its details which is later used to generate invoices, calculate fine, and update land availability status.

read.py - C:\Users\Miracle\OneDrive\Desktop\23048640 Pratik Man Pradhan\read.py (3.11.1)

File Edit Format Run Options Window Help

```
import datetime

def readDataFromFile(fileName):
    # Reads data from a specified file and returns a list of lines
    data = []
    try:
        with open(fileName, 'r') as file:
            for line in file:
                lineData = line.strip().split(',')
                lineData[0] = int(lineData[0])  # Convert Kitta No to integer
                lineData[3] = int(lineData[3])  # Convert Anna to integer
                lineData[4] = int(lineData[4])  # Convert Price to integer
                data.append(lineData)
```

Figure 5: data is a list used to store all the data read from the file

- **Dictionaries**: Dictionaries in Python are key-value data structures, allowing quick access to values based on keys. In this project, dictionaries were used to map land IDs to their corresponding details. Land IDs were inserted as keys to its corresponding land details.

```
def createLandDict(data):
    # Creates a dictionary from the given data
    landDict = {}
    try:
        for i in range(len(data)):
            key = i + 1
            value = data[i]
            landDict[key] = value
```

Figure 6: Dictionary storing land details

- **Tuples**: Tuples are similar to lists but are immutable, meaning they cannot be changed after creation. Each rental record is stored as a tuple with the land ID, renter's name, rental duration, and rental date in a list named rental.

```
# Store rental details
rentals.append((landId, renterName, duration, rentalDate))
```

Figure 7: data being stored as tuple in this list named rentals

- **Strings**: Strings are sequences of characters and are used throughout the project to represent textual data. One of its various uses in this project is to store customer's name input from the user.

```
if renterName == "":
    renterName = input("Enter the name of the person renting the land:")  # Ask for the name once
```

Figure 8: The name given by the user is stored as a string.

- **Integers**: Integers represent whole numbers and are used for numerical operations. One of its many uses in this project is to store land ID stored as an integer for easy comparison later.

```
lineData[0] = int(lineData[0])  # Convert Kitta No to integer
lineData[3] = int(lineData[3])  # Convert Anna to integer
lineData[4] = int(lineData[4])  # Convert Price to integer
```

Figure 9: Land id converted into integer after being read from file as String.

Program:

Implementation of the Program:

The land management system is a Python-based application that helps manage the rental and return of land. It uses various data structures and modules to facilitate operations, such as reading and writing data, generating invoices, and calculating fines for overdue returns. This program consists of 5 .py files named main, read, write, operations, and message. Each Python file has different functions divided through it according to its work of task. At last, all these functions are called through the main function named landManagementSystem().

- When the landManagementSystem() is called, at first it prints the welcome message, table of land parcels, and user menu. Then, it asks the user to input the value according to the user's need, checking the information in the user menu.

```
*IDLE Shell 3.11.1*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Miracle\OneDrive\Desktop\23048640 Pratik Man Pradhan\main.py
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        Hello and Welcome to the Land Management System
                Techno Property Nepal
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-------------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-------------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-------------------------------------------------------------------------------------------------------------------------
Enter '1' to rent land
Enter '2' to return land
Enter '3' to exit
Please enter a value:
```

Figure 10: First action after calling landManagementSystem()

Rental Process :

To start the rental process, the user needs to input the value 1. After the value is entered, it prints the table of land parcels and asks the user to enter the land ID s/he wants to rent.

```
-----------------------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to rent: |
```

Figure 11: After entering the rental process

If the land ID user inputs aren't available at the moment, then it displays the suitable message and asks the user if they want to continue renting the land.

```
-----------------------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to rent: 2
Land ID 2 is not available for rental.
Do you want to rent another land? (n to stop, anything else to continue):
```

Figure 12: After trying to rent the land that isn't available.

If n is entered then, the program will go back to the user menu.

```
Land ID 2 is not available for rental.
Do you want to rent another land? (n to stop, anything else to continue): n
-----------------------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------------------
Enter '1' to rent land
Enter '2' to return land
Enter '3' to exit
Please enter a value:|
```

Figure 13: After entering n to rent another land

If any other value rather than n is entered then, the program will continue asking for the land ID to the user.

```
Do you want to rent another land? (n to stop, anything else to continue): y
-----------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to rent: |
```

Figure 14: After entering any other value than n (Rental process)

Now, when the user inputs the available land ID, it asks the user to input the customer's name and duration of the land rental. And also asks the user if they want to rent any other lands.

```
-----------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to rent: 1
Enter the name of the person renting the land:pratik
Enter the number of months to rent: 2
Do you want to rent another land? (n to stop, anything else to continue): |
```

Figure 15: After entering the available land ID. (Rental process)

When n is entered, the program will go back to the user menu as usual. When any other value is entered, you can enter another land ID and rent more than one land at the same time. And also the land ID user just booked will be updated to "Not Available" in the table as well as in txt file.

```
Do you want to rent another land? (n to stop, anything else to continue): y
-----------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Not Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to rent: |
```

Figure 16: After entering any other value than n to rent more than one land

Now, when you enter the land ID that isn't available it will display a suitable message and ask if they want to continue to rent the land. But if the user enters the available land ID, it will only ask for the duration to rent the land and skip asking for the customer's name.

```
Do you want to rent another land? (n to stop, anything else to continue): y
-----------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Not Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to rent: 3
Enter the number of months to rent:
```

Figure 17: After entering available land ID second time

Now, after entering the number of months to rent the land and entering n to "Do you want to rent another land?", It prints an invoice in the terminal as well as generates a Txt file containing the invoice in the folder.

```
Enter the number of months to rent: 2
Do you want to rent another land? (n to stop, anything else to continue): n

-------------------------------------------------------------
Invoice for Land Rental
-------------------------------------------------------------

Name of Customer: pratik
Date and Time of Rent of Land: 2024-05-06 12:09:20.730409
Total amount for all rented lands: Rs 300000
-------------------------------------------------------------

Kitta No: 101
Location: Kathmandu
Direction: North
Anna: 4
Monthly Rate of Land: 50000
Rental Duration: 2 months
Amount = Rs100000
-------------------------------------------------------------

Kitta No: 105
Location: Bhaktapur
Direction: South
Anna: 10
Monthly Rate of Land: 100000
Rental Duration: 2 months
Amount = Rs200000
-------------------------------------------------------------
```

Figure 18: Printing invoice in the terminal

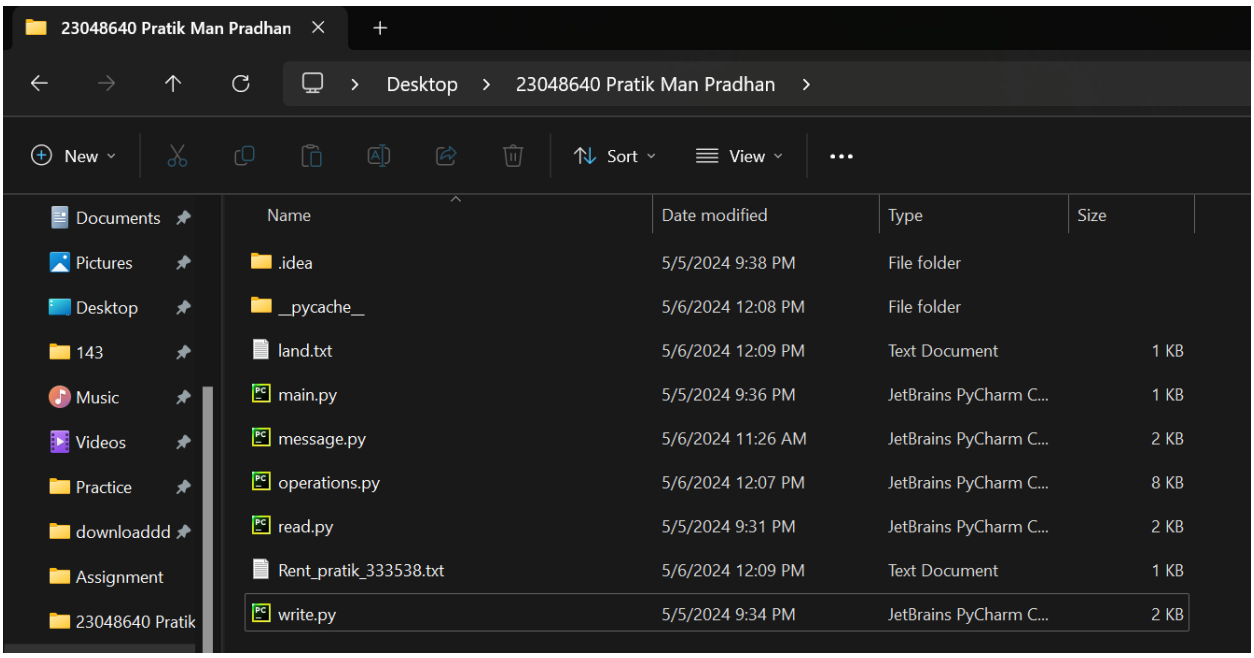


Figure 19: Rent Invoice Txt File generation

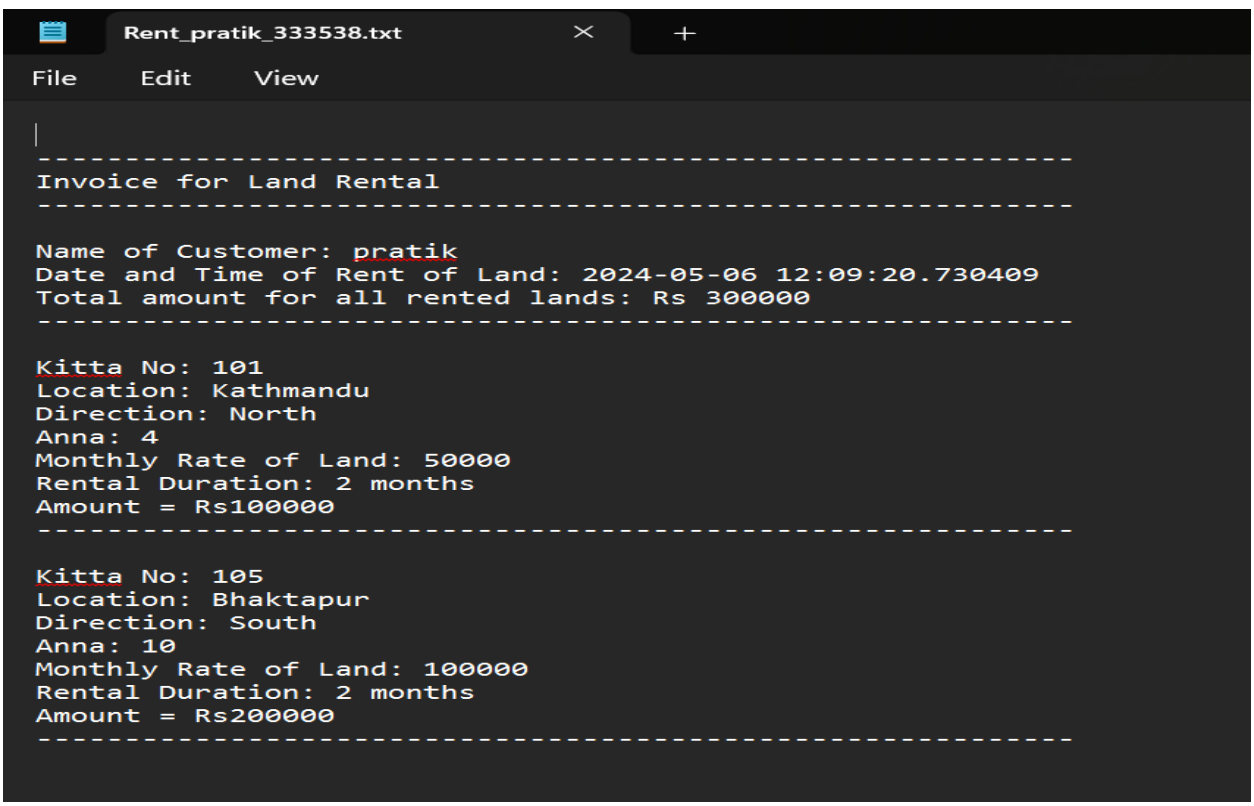


Figure 20: Invoice Format in the Txt File

Return Process:

Now, to start the return process, the user needs to input the value 2. After the value is entered, it prints the table of land parcels and asks the user to enter the land ID s/he wants to return.

```
-----------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Not Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to return (or -1 to stop):
```

Figure 21:After entering the return process

. If the land ID user inputs are already available at the moment and not rented, then it displays the suitable message and asks the user to input another land id or -1 to stop returning the land.

```
Enter the land ID you want to return (or -1 to stop): 4
Land ID 4 is already available or not currently rented.
-----------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Not Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to return (or -1 to stop):
```

Figure 22: After trying to return the land that isn't rented at all.

If the user enters -1 now, then the suitable message is displayed and the program goes back to the user menu.

```
Enter the land ID you want to return (or -1 to stop): -1
No lands were returned.
-----------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Not Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------
Enter '1' to rent land
Enter '2' to return land
Enter '3' to exit
Please enter a value:
```

Figure 23: After entering -1 without returning any land

But if the user now enters the land ID that is currently “Not Available”, then the program asks the user to input the name of the person returning the land, the date of rental start, and the duration the land was rented for. After entering all these value, the suitable message is displayed and the availability of land status is updated in the table and again asks the user for land ID if they want to return more or -1 to stop returning.

```
Enter the land ID you want to return (or -1 to stop): 1
Enter the name of the person returning the land: pratik
Enter the rental start date (YYYY-MM-DD): 2024-1-1
Enter the expected rental duration in months: 2
Land ID 1 has been returned and is now available.
-------------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-------------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-------------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to return (or -1 to stop):
```

Figure 24: After entering all the values asked in return process

If entered -1 now, the invoice will print in the terminal and a Txt file will be to store the return invoice.

But the user can return more than one land at a time. When another land ID is entered that is not available, the program will ask the user the date of rental start and duration this land was rented for only and not the name of the user again as it was provided already in the first land return. After entering all these values, availability status is again updated of this land ID and will again ask the user if they want to return more land.

```
Enter the land ID you want to return (or -1 to stop): 2
Enter the rental start date (YYYY-MM-DD): 2024-4-4
Enter the expected rental duration in months: 2
Land ID 2 has been returned and is now available.
-------------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-------------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-------------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to return (or -1 to stop):
```

Figure 25: After entering second land details to return

Now, when -1 is entered the invoice of both land returns will be printed in the terminal and fine will be applied if the return date was late than it was supposed to. If the return date was on time, then Rs 0 fine will be applied.

```
Enter the land ID you want to return (or -1 to stop): -1
Here is your return invoice:

------------------------------------------------------------
Land Return Invoice
------------------------------------------------------------

Name of Customer: pratik
OTotal Fine: Rs 110000
Grand Total: Rs 440000
------------------------------------------------------------

Kitta No: 101
Location: Kathmandu
Direction: North
Rental Start Date: 2024-01-01
Return Date: 2024-05-06
Fine: Rs 110000
Total Amount: Rs 100000
------------------------------------------------------------

Kitta No: 102
Location: Lalitpur
Direction: East
Rental Start Date: 2024-04-04
Return Date: 2024-05-06
Fine: Rs 0
Total Amount: Rs 120000
------------------------------------------------------------
```

Figure 26: Return invoice in terminal after exiting return process

After the return invoice is printed in the terminal, the program goes back to user menu.

```
Return Date: 2024-05-06
Fine: Rs 0
Total Amount: Rs 120000
------------------------------------------------------------

------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
------------------------------------------------------------------------------------------------------------
Enter '1' to rent land
Enter '2' to return land
Enter '3' to exit
Please enter a value:
```

Figure 27: After exiting return process

A Txt file is also generated in the folder after exiting the return process.

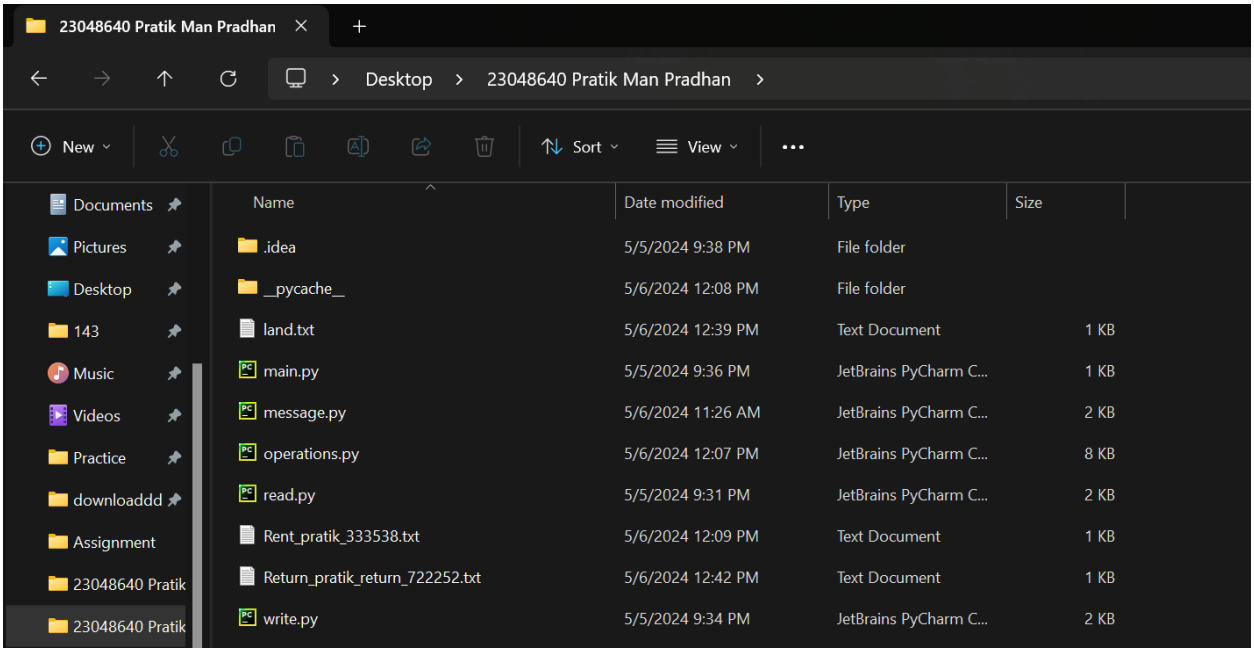


Figure 28: Return Invoice Txt File generation in the folder

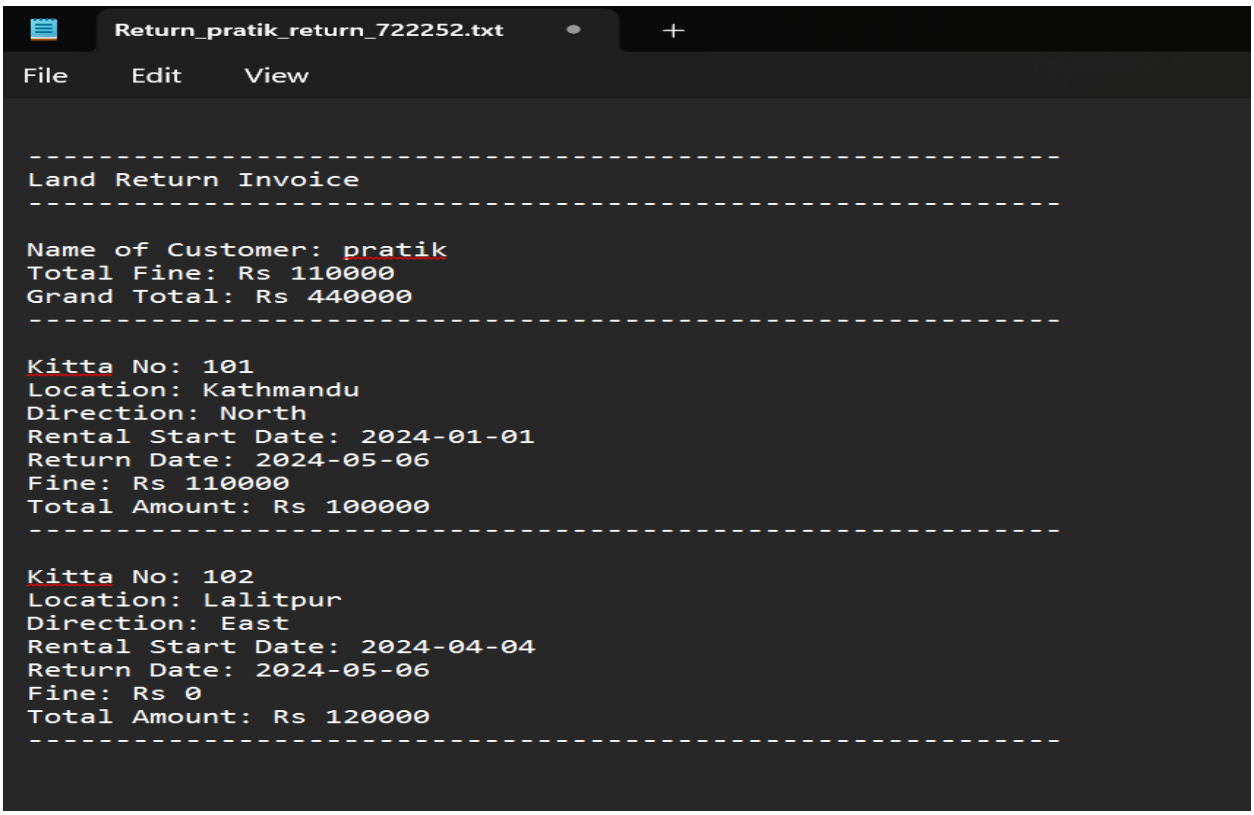


Figure 29: Return Invoice Txt File Format

Termination of the Program:

To exit the whole program, the user must enter 3 in the user menu. After that, the program will display a thank you message and will stop executing.

```
Enter '1' to rent land
Enter '2' to return land
Enter '3' to exit
Please enter a value:3
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Thank you for using our Land Management System
                Techno Property Nepal
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 30: Termination of Program

Testing:

Test 1: Show implementation of try, except

Objective:	**The user is only allowed to enter integers.**
Action:	Entering the string value as input which actually should be integer.
Expected Result:	Error should occur and the program would crash.
Actual Result:	"Invalid input! Please enter a valid value." was printed in the terminal and the program didn't crash.
Conclusion:	This example demonstrates the common use of try-except in Python to manage errors gracefully, allowing the program to continue running even when exceptions occur. By catching the error and providing feedback to the user, you create a more user-friendly experience and avoid program crashes due to invalid input.

Table 1:Implementation of Try Except (test-1)

```
*IDLE Shell 3.11.1*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec  6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Miracle\OneDrive\Desktop\23048640 Pratik Man Pradhan\main.py
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        Hello and Welcome to the Land Management System
                Techno Property Nepal
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
--------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
--------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
--------------------------------------------------------------------------------------------------------------
Enter '1' to rent land
Enter '2' to return land
Enter '3' to exit
Please enter a value:w
Invalid input! Please enter valid value.
--------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
--------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
--------------------------------------------------------------------------------------------------------------
Enter '1' to rent land
Enter '2' to return land
Enter '3' to exit
Please enter a value:
```

Figure 31:Try Except (putting value as a string where it should be integer)

Test 2: Selection of rent and return of lands

Objective:	**The user is only allowed to enter the land ID that exists in the file.**
Action:	Entering the land ID that doesn't exist.
Expected Result:	The error should display in the terminal and the program would crash.
Actual Result:	"Invalid land ID! Please enter a valid ID." was printed in the terminal and the program didn't crash.
Conclusion:	This shows the importance of checking whether the value entered exists in the file or not. If it doesn't exist then the program outputs the suitable message and ask for the value again and doesn't crash.

Table 2: Test-2 (Selection of rent and return of lands)

```
Enter the land ID you want to rent: -1
Invalid land ID! Please enter a valid ID.
-----------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------
Enter the land ID you want to rent:
```

Figure 32: Entering negative value to rent

```
Enter the land ID you want to rent: 10
Invalid land ID! Please enter a valid ID.
-------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-------------------------------------------------------------------------------------------------------------
Enter the land ID you want to rent: |
```

Figure 33: Entering non existant land ID in rent

```
Enter the land ID you want to return (or -1 to stop): -2
Invalid land ID! Please enter a valid ID.
-------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-------------------------------------------------------------------------------------------------------------
Enter the land ID you want to return (or -1 to stop):
```

Figure 34:Entering negative value in land ID that doesn't exist while returning

```
Enter the land ID you want to return (or -1 to stop): 10
Invalid land ID! Please enter a valid ID.
-------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Not Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-------------------------------------------------------------------------------------------------------------
Enter the land ID you want to return (or -1 to stop): |
```

Figure 35:Entering non-existant positive value in land ID while returning the land

Test 3: File generation of renting of lands

Objective:	**The file should be generated after renting the land**
Action:	Renting the land.
Expected Result:	A Txt file should be generated after renting the land.
Actual Result:	A Txt file was generated with the name of the customer and a unique value.
Conclusion:	After renting a land, the txt file is generated named with the customer's name and a unique value containing all the information about the rental including the rental amount that is stored and saved in the folder for future transparency.

Table 3: Test-3(File generation of renting of land)

```
Enter '1' to rent land
Enter '2' to return land
Enter '3' to exit
Please enter a value:1
-----------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to rent: 1
Enter the name of the person renting the land:pratik
Enter the number of months to rent: 2
Do you want to rent another land? (n to stop, anything else to continue): y
-----------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Not Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to rent: 2
Enter the number of months to rent: 2
Do you want to rent another land? (n to stop, anything else to continue): n
```

Figure 36: Complete rental process of multiple land

```
------------------------------------------------------------
Invoice for Land Rental
------------------------------------------------------------

Name of Customer: pratik
Date and Time of Rent of Land: 2024-05-06 15:53:26.213679
Total amount for all rented lands: Rs 220000
------------------------------------------------------------

Kitta No: 101
Location: Kathmandu
Direction: North
Anna: 4
Monthly Rate of Land: 50000
Rental Duration: 2 months
Amount = Rs100000
------------------------------------------------------------

Kitta No: 102
Location: Lalitpur
Direction: East
Anna: 5
Monthly Rate of Land: 60000
Rental Duration: 2 months
Amount = Rs120000
------------------------------------------------------------
```

Figure 38: Invoice printed in the terminal after land rental process has ended

Rent_pratik_561796.txt

File Edit View

```
------------------------------------------------------------
Invoice for Land Rental
------------------------------------------------------------

Name of Customer: pratik
Date and Time of Rent of Land: 2024-05-06 15:53:26.213679
Total amount for all rented lands: Rs 220000
------------------------------------------------------------

Kitta No: 101
Location: Kathmandu
Direction: North
Anna: 4
Monthly Rate of Land: 50000
Rental Duration: 2 months
Amount = Rs100000
------------------------------------------------------------

Kitta No: 102
Location: Lalitpur
Direction: East
Anna: 5
Monthly Rate of Land: 60000
Rental Duration: 2 months
Amount = Rs120000
------------------------------------------------------------
```

Figure 37: Invoice format in the Txt file

Test 4: File generation of returning of lands

Objective:	**The file should be generated after returning the land**
Action:	Returning the land.
Expected Result:	A Txt file should be generated after returning the land.
Actual Result:	A Txt file was generated with the name of the customer and a unique value after the returning process.
Conclusion:	After returning a land, the txt file is generated named with the customer's name and a unique value containing all the information about the return including the total amount (including fine if applied) that is stored and saved in the folder for future transparency.

Table 4:Test- 4(File generation on returning of land)

```
1        101             Kathmandu               North           4       50000         Not Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to return (or -1 to stop): 1
Enter the name of the person returning the land: michael
Enter the rental start date (YYYY-MM-DD): 2024-1-1
Enter the expected rental duration in months: 2
Land ID 1 has been returned and is now available.
-------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Not Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to return (or -1 to stop): 2
Enter the rental start date (YYYY-MM-DD): 2024-4-10
Enter the expected rental duration in months: 2
Land ID 2 has been returned and is now available.
-------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to return (or -1 to stop): -1
```

Figure 39: Return process of Multiple land.

```
Enter the land ID you want to return (or -1 to stop): -1
Here is your return invoice:

------------------------------------------------------------
Land Return Invoice
------------------------------------------------------------

Name of Customer: michael
OTotal Fine: Rs 110000
Grand Total: Rs 440000
------------------------------------------------------------

Kitta No: 101
Location: Kathmandu
Direction: North
Rental Start Date: 2024-01-01
Return Date: 2024-05-06
Fine: Rs 110000
Total Amount: Rs 100000
------------------------------------------------------------

Kitta No: 102
Location: Lalitpur
Direction: East
Rental Start Date: 2024-04-10
Return Date: 2024-05-06
Fine: Rs 0
Total Amount: Rs 120000
------------------------------------------------------------
```

Figure 41: Return Invoice printed in the terminal after all the rental process.

Return_michael_return_855821.txt

File Edit View

```
------------------------------------------------------------
Land Return Invoice
------------------------------------------------------------

Name of Customer: michael
OTotal Fine: Rs 110000
Grand Total: Rs 440000
------------------------------------------------------------

Kitta No: 101
Location: Kathmandu
Direction: North
Rental Start Date: 2024-01-01
Return Date: 2024-05-06
Fine: Rs 110000
Total Amount: Rs 100000
------------------------------------------------------------

Kitta No: 102
Location: Lalitpur
Direction: East
Rental Start Date: 2024-04-10
Return Date: 2024-05-06
Fine: Rs 0
Total Amount: Rs 120000
------------------------------------------------------------
```

Figure 40:Return Invoice format in Txt File

Test 5: Show the update in stock of lands

Objective:	**Changing the Availability of land after renting or returning of land.**
Action:	Renting and Returning the land.
Expected Result:	Availability of land should be updated after the rental and return process
Actual Result:	Availability of land was updated after the rental and return process
Conclusion:	Every time after the rental and return process is executed, the land availability status is updated in the Txt file and also in data dictionary in order to update the table in terminal and keep the user informed about the land status

Table 5: Test-5 (Show the update in stocks of land)

```
*IDLE Shell 3.11.1*
File  Edit  Shell  Debug  Options  Window  Help
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-----------------------------------------------------------------------------------------------------------
Enter the land ID you want to rent: 1
Enter the name of the person renting the land:pratik
Enter the number of months to rent: 2
Do you want to rent another land? (n to stop, anything else to continue): n

-------------------------------------------------------
Invoice for Land Rental
-------------------------------------------------------

Name of Customer: pratik
Date and Time of Rent of Land: 2024-05-06 17:08:39.163119
Total amount for all rented lands: Rs 100000
-------------------------------------------------------

Kitta No: 101
Location: Kathmandu
Direction: North
Anna: 4
Monthly Rate of Land: 50000
Rental Duration: 2 months
Amount = Rs100000
-------------------------------------------------------

-----------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-----------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Not Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
```

Figure 42:After renting the land no 1, availability status in terminal was updated to "Not Available"

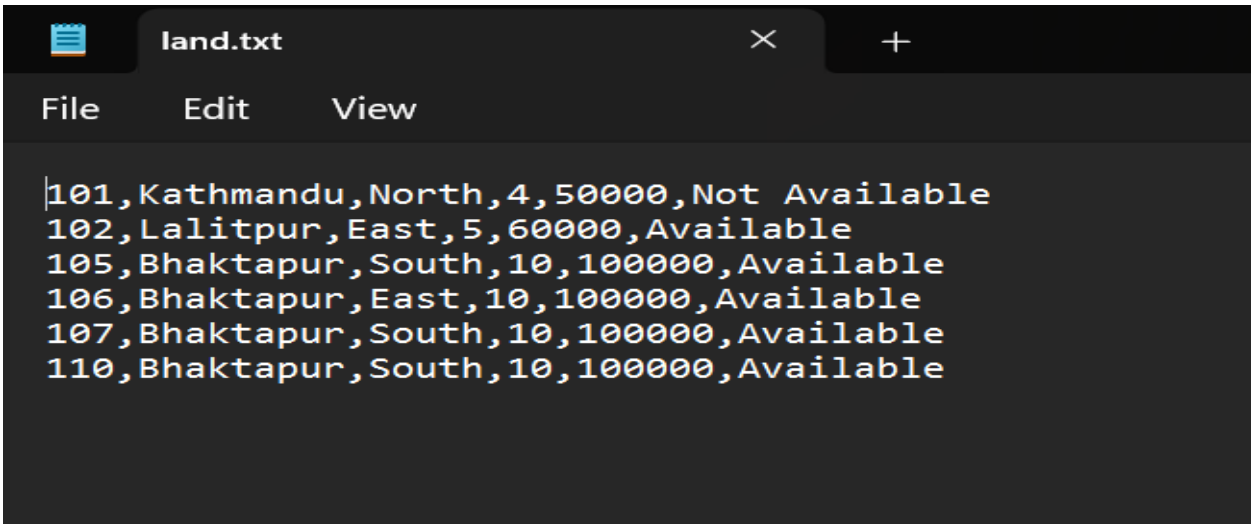


Figure 43: Land Availability changed to "Not Available" in Txt File too after land renting

```
Please enter a value:2
-------------------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-------------------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Not Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-------------------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to return (or -1 to stop): 1
Enter the name of the person returning the land: pratik
Enter the rental start date (YYYY-MM-DD): 2024-1-1
Enter the expected rental duration in months: 2
Land ID 1 has been returned and is now available.
-------------------------------------------------------------------------------------------------------------------------------
Land ID  Kitta No        City/District Name      Direction       Anna    Price   Availability Status
-------------------------------------------------------------------------------------------------------------------------------
1        101             Kathmandu               North           4       50000         Available
2        102             Lalitpur                East            5       60000         Available
3        105             Bhaktapur               South           10      100000         Available
4        106             Bhaktapur               East            10      100000         Available
5        107             Bhaktapur               South           10      100000         Available
6        110             Bhaktapur               South           10      100000         Available
-------------------------------------------------------------------------------------------------------------------------------
Enter the land ID you want to return (or -1 to stop): -1
```

Figure 44:After returning the land no 1, availability status in terminal was updated to "Available"

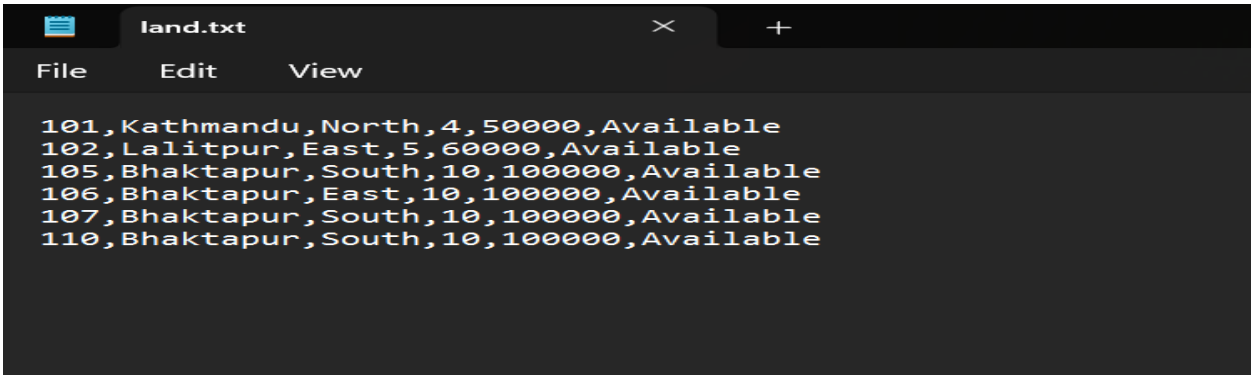


Figure 45:Land Availability changed to "Available" in Txt File too after land returning

CONCLUSION

The land management system project demonstrates the effective application of Python programming to address a real-world problem: managing land rentals and returns. Throughout the project, we utilized various data structures, including lists and dictionaries, to organize and manipulate data efficiently. The core functionalities, such as reading and writing data, handling land rentals and returns, generating invoices, and updating land availability, were implemented through a combination of different ways of operations and thoughtful design.

One of the crucial aspects of this project is to make it user-friendly. By displaying clear messages, a simple user menu option, and appropriate error handling using try-except blocks, the system provides an easy-to-operate user experience. The combination of error handling ensures that the program can gracefully recover from unexpected input, guiding the user back on track without crashing.

The project achieves its primary goals: automating land transactions, providing real-time information on land availability, and ensuring transparency through invoice generation. The use of automation and efficient data management reduces the chances of errors, leading to a more reliable system.

In conclusion, this demonstrates the effectiveness of Python in building practical applications and highlights the importance of error handling and user-friendly design. Future enhancements could include additional features, such as database integration, user authentication, and more advanced reporting, to further extend the system's capabilities. Overall, this project serves as a valuable starting point for more comprehensive land management solutions.

Appendix

Main.py Code:

```
import read
import write
import operations
import message
import datetime


def landManagementSystem():
    # Start point for the Land Management System

    message.printWelcomeMessage()
    data = read.readDataFromFile("land.txt")
    landDict = read.createLandDict(data)

    while True:
        message.printTableOfLand()

        # User menu
        print("Enter '1' to rent land")
        print("Enter '2' to return land")
        print("Enter '3' to exit")

        try:
            userChoice = int(input("Please enter a value:"))

            if userChoice == 1:
                operations.handleLandRental(landDict, data)
            elif userChoice == 2:
                operations.handleLandReturn(landDict, data)
            elif userChoice == 3:
                message.printThankYouMessage()
                break
            else:
                message.printInvalidMessage()


        except ValueError:
            print("Invalid input! Please enter valid value.")
        except Exception as e:
            print(f"An unexpected error occurred: {e}")
```

```
# Run the Land Management System
landManagementSystem()
```

---

read.py code:

```
import datetime

def readDataFromFile(fileName):
    # Reads data from a specified file and returns a list of lines
    data = []
    try:
        with open(fileName, 'r') as file:
            for line in file:
                lineData = line.strip().split(',')
                lineData[0] = int(lineData[0])  # Convert Kitta No to integer
                lineData[3] = int(lineData[3])  # Convert Anna to integer
                lineData[4] = int(lineData[4])  # Convert Price to integer
                data.append(lineData)
    except FileNotFoundError:
        print(f"Error: The file '{fileName}' was not found.")
    except Exception as e:
        print(f"An error occurred while reading the file: {e}")
    return data


def createLandDict(data):
    # Creates a dictionary from the given data
    landDict = {}
    try:
        for i in range(len(data)):
            key = i + 1
            value = data[i]
            landDict[key] = value
    except Exception as e:
        print(f"An error occurred while creating the land dictionary: {e}")
    return landDict
```

Write.py code:

```
import datetime

def updateLandAvailability(landId, newStatus, fileName="land.txt"):
    # Updates the availability status of a specific land ID
    try:
        with open(fileName, 'r') as file:
            lines = file.readlines()

        updatedLines = []
        for line in lines:
            lineData = line.strip().split(',')
            currentLandId = int(lineData[0])

            if currentLandId == landId:
                lineData[5] = newStatus

            updatedLine = ','.join(lineData) + '\n'
            updatedLines.append(updatedLine)

        with open(fileName, 'w') as file:
            file.writelines(updatedLines)

    except Exception as e:
        print(f"An error occurred while updating land availability: {e}")


def writeInvoiceToFile(name, invoiceContent):
    # Writes the rental invoice to a file with a unique identifier
    try:
        uniqueValue = str(datetime.datetime.now().minute +
datetime.datetime.now().second + datetime.datetime.now().microsecond)
        with open(f"Rent_{name}_{uniqueValue}.txt", 'w') as file:
            file.write(invoiceContent)
    except Exception as e:
        print(f"An error occurred while writing the invoice: {e}")


def writeInvoiceToFileReturn(name, invoiceContent):
    # Writes the return invoice to a file with a unique identifier
    try:
        uniqueValue = str(datetime.datetime.now().minute +
datetime.datetime.now().second + datetime.datetime.now().microsecond)
        with open(f"Return_{name}_{uniqueValue}.txt", 'w') as file:
            file.write(invoiceContent)
```

```
    except Exception as e:
        print(f"An error occurred while writing the return invoice: {e}")
```

---

operations.py code:

```
import datetime
import write
import message
import read


def handleLandRental(landDict, data):
    # Handles the rental of land based on user input
    rentals = []
    grandTotal = 0
    renterName = ""  # Initialize the renter's name variable

    while True:
        try:
            message.printTableOfLand()
            landId = int(input("Enter the land ID you want to rent: "))

            if landId <= 0 or landId > len(landDict):
                print("Invalid land ID! Please enter a valid ID.")
            else:
                landStatus = data[landId - 1][5]
                if landStatus.lower() == "available":
                    if renterName == "":
                        renterName = input("Enter the name of the person renting the land:")  #
Ask for the name once

                    rentalDate = datetime.datetime.now()
                    duration = int(input("Enter the number of months to rent: "))

                    # Update land status
                    write.updateLandAvailability(landDict[landId][0], "Not Available")
                    landDict[landId][5] = "Not Available"

                    landPrice = landDict[landId][4] * duration
                    grandTotal += landPrice

                    # Store rental details
                    rentals.append((landId, renterName, duration, rentalDate))
```

```
                decision = input("Do you want to rent another land? (n to stop, anything
else to continue): ")
                if decision.lower() == 'n':
                    invoice = generateInvoice(rentals, renterName, rentalDate, landDict,
grandTotal)
                    write.writeInvoiceToFile(renterName, invoice)
                    print(invoice)
                    break
            else:
                print(f"Land ID {landId} is not available for rental.")
                decision = input("Do you want to rent another land? (n to stop, anything
else to continue): ")
                if decision.lower() == 'n':
                    break

        except ValueError:
            print("Invalid input! Please enter valid data.")
        except Exception as e:
            print(f"An unexpected error occurred: {e}")


def handleLandReturn(landDict, data):
    # Handles the return of land based on user input
    rentalsToReturn = []
    renterName = ""
    totalAmount = 0  # Cumulative total for all returns

    try:
        while True:
            message.printTableOfLand()
            landId = int(input("Enter the land ID you want to return (or -1 to stop): "))

            if landId == -1:  # Exit condition for loop
                break

            if landId <= 0 or landId > len(landDict):
                print("Invalid land ID! Please enter a valid ID.")
                continue

            landDetails = data[landId - 1]
            landStatus = landDetails[5]

            if landStatus.lower() == "not available":
                if renterName == "":
                    renterName = input("Enter the name of the person returning the land: ")
```

```
            # Collect rental details
            rentalDateStr = input("Enter the rental start date (YYYY-MM-DD): ")
            rentalDate = parseDate(rentalDateStr)
            rentalDuration = int(input("Enter the expected rental duration in months: "))
            expectedDuration = 30 * rentalDuration

            # Calculate the fine and total amount
            returnDate = datetime.datetime.now()  # Current date and time
            actualDuration = (returnDate - rentalDate).days
            monthlyRate = landDict[landId][4]
            fine = calculateFine(actualDuration, expectedDuration, monthlyRate)

            # Update total amounts
            landRent = rentalDuration * monthlyRate
            totalAmount += landRent + fine

            # Update land availability
            write.updateLandAvailability(landDict[landId][0], "Available")
            landDict[landId][5] = "Available"

            # Add to list of rentals to return
            rentalsToReturn.append({
                "landId": landId,
                "landDetails": landDetails,
                "rentalDate": rentalDate,
                "returnDate": returnDate,
                "fine": fine,
                "totalAmount": landRent,
            })

            print(f"Land ID {landId} has been returned and is now available.")
        else:
            print(f"Land ID {landId} is already available or not currently rented.")

    # Generate the return invoice for all returned lands
    if rentalsToReturn:
        totalFine = sum(rental["fine"] for rental in rentalsToReturn)
        grandTotal = totalAmount + totalFine

        # Generate the return invoice
        returnInvoice = generateReturnInvoice(renterName, rentalsToReturn, totalFine,
grandTotal)

        # Write the invoice to a file and display it in the terminal
        write.writeInvoiceToFileReturn(f"{renterName}_return", returnInvoice)
```

```
            # Display the invoice
            print("Here is your return invoice:")
            print(returnInvoice)

        else:
            print("No lands were returned.")

    except ValueError as ve:
        print(f"Invalid input: {ve}")
    except Exception as e:
        print(f"An error occurred while returning the land: {e}")


def parseDate(dateStr):
    # Parses a date string into a datetime object
    try:
        return datetime.datetime.strptime(dateStr, "%Y-%m-%d")
    except ValueError:
        raise ValueError("Invalid date format. Please use YYYY-MM-DD.")


def calculateFine(actualDuration, expectedDuration, monthlyRate):
    # Calculates the fine based on the difference between actual and expected durations
    if actualDuration > expectedDuration:
        fine = int(((monthlyRate / 30) * (actualDuration - expectedDuration)))
    else:
        fine = 0
    return fine


def generateInvoice(rentals, customerName, rentalDate, landDict, grandTotal):
    # Generates a rental invoice based on the rental details

    invoice = f"""
------------------------------------------------------------
Invoice for Land Rental
------------------------------------------------------------

Name of Customer: {customerName}
Date and Time of Rent of Land: {rentalDate}
Total amount for all rented lands: Rs {grandTotal}
------------------------------------------------------------
"""

    for rental in rentals:
```

```
        landId, _, duration, _ = rental
        landDetails = landDict[landId]
        amount = landDetails[4]*duration

        invoice += f"""
Kitta No: {landDetails[0]}
Location: {landDetails[1]}
Direction: {landDetails[2]}
Anna: {landDetails[3]}
Monthly Rate of Land: {landDetails[4]}
Rental Duration: {duration} months
Amount = Rs{amount}
-----------------------------------------------------------
"""
    return invoice


def generateReturnInvoice(renterName, rentalsToReturn, totalFine, grandTotal):
    # Generates a return invoice based on the return details
    returnInvoice = f"""
-----------------------------------------------------------
Land Return Invoice
-----------------------------------------------------------

Name of Customer: {renterName}
OTotal Fine: Rs {totalFine}
Grand Total: Rs {grandTotal}
-----------------------------------------------------------
"""

    for rental in rentalsToReturn:
        landDetails = rental["landDetails"]
        returnInvoice += f"""
Kitta No: {landDetails[0]}
Location: {landDetails[1]}
Direction: {landDetails[2]}
Rental Start Date: {rental["rentalDate"].date()}
Return Date: {rental["returnDate"].date()}
Fine: Rs {rental["fine"]}
Total Amount: Rs {rental["totalAmount"]}
-----------------------------------------------------------
"""
    return returnInvoice
```

message.py code:

```
import read


def printWelcomeMessage():
    # Prints a welcome message for the Land Management System

print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
")
    print("\tHello and Welcome to the Land Management System")
    print("\t        Techno Property Nepal")

print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
")


def printTableOfLand():
    # Prints the current table of lands with their details
    data = read.readDataFromFile("land.txt")
    landDict = read.createLandDict(data)

    print(
        "-------------------------------------------------------------------------------------------------------------------------------")
    print("Land ID\t", "Kitta No\t", "City/District Name\t", "Direction\t", "Anna\t", "Price\t",
"Availability Status")
    print(
        "-------------------------------------------------------------------------------------------------------------------------------")
    for key, value in landDict.items():
        print(key, "\t", value[0], "\t\t", value[1], "\t\t", value[2], "\t\t", value[3], "\t", value[4], "
", value[5])
    print(
        "-------------------------------------------------------------------------------------------------------------------------------")


def printThankYouMessage():
    # Prints a thank you message for using the system

print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("Thank you for using our Land Management System")
    print("\t        Techno Property Nepal")

print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
```

```
def printInvalidMessage():
    # Prints an error message for invalid input
    print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("Invalid input! Please enter a valid choice (1, 2, or 3).")
    print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
```

Bibliography

Anon., n.d. [Online]
Available at: https://www.python.org/community/logos/
Anon., n.d. *logos-world.net.* [Online]
Available at: https://logos-world.net/microsoft-word-logo/
Anon., n.d. *wikipedia.org.* [Online]
Available at: https://en.m.wikipedia.org/wiki/File:Diagrams.net_Logo.svg