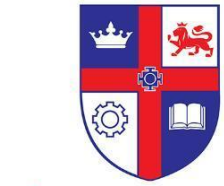
LONDON METROPOLITAN UNIVERSITY



islington college
(इस्लिङ्टन कलेज)

CS4051NI Fundamentals of Computing

60% Individual Coursework

2023/24 Spring

Student Name: Samarpan Khadka

London Met ID: 23047551

College ID: np01nt4a230214

Assignment Due Date: Tuesday, May 7, 2024

Assignment Submission Date: Tuesday, May 7, 2024

Word Count:4120

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

Contents

Introduction

Introduction of the project.

This project is about making an application for a private stock land rental company, which stores all the available lands in various locations in a text file. The application is made for managing and handling the file. The application reads the text file and gathers all the information or data about the availability of land. With every transaction such as renting of land or multiple lands, an invoice should be generated and stored in a text file. The quantity is automatically updated as the number of transactions made by customers. At the time of returning the rented lands, the quantity is updated again accordingly, and a bill is generated stating the total price of customer. A bill should be generated for each transaction. Also, if the land rent money is returned late than a certain quota of days, then a certain amount of fine is applied. Then the total rent money is included in the text file.

Python was the programming language used to write the complete code. The program, which serves as the college's basic coding curriculum, was developed using the Integrated Development Learning Environment (IDLE).

Goal:

The goal of the program or project is to develop a land rental company that will assist user or customers to rent lands if available with their demand. The program should handle transactions between customers and generate invoice after every transaction to maintain an accurate up-to-date records of availability of land and record of fines.

Objectives.

1.Develop a program that can read a txt file which contains information of different lands available on different places.

2.Develop a program that can update the txt file on every transaction made such as renting the land or returning the land.

3.Develop a program that can generate receipts after every transaction.

4.Testing the project to make sure that is works properly and meet the requirements of rental company.

5.Documentation of program and instruct on how to use it.

By achieving these objectives, our land rental company can manage and rent the available land according to customers' needs which makes easier for customers as our program is easy and improve customer service. The program will have a big impact on maintaining good relation between customer and company which will have a great impact on our company.

1.1

1.2 Tools Used:

To make this project successful.3 tools were used. They are:

1.IDLE
2.Draw.io:
3.MS-WORD

1.IDLE:

The Python integrated development environment (IDE) is called IDLE (Integrated Development and Learning Environments). Like Python Shell, IDLE allows you to run a single statement in addition to creating, editing, and running Python programs. With capabilities like autocompletion, smart indent, and syntax highlighting, IDLE offers a feature-rich text editor for writing Python scripts. Additionally, it supports a debugger with breakpoints and stepping.
(TutorialsTeahcer, n.d.)

2.Draw.io:

draw.io is free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams.Draw.io is a flowchart solution designed to help developers, network admins, IT analysts, and designers use drag-and-drop functionality to create and publish diagrams.

3.MS-Word

Microsoft Word is a word processing program that allows for the creation of both simple and complex documents. Given its wide variety of designs and styes, it allows users to create a document according to their desired layout. In the education sector, it serves as a documentation medium for coursework. (ualr, 2022)

Algorithm:

An algorithm is a process which highlights sequence or order of actions to be carried out in a particular order to give required output. an algorithm is a mathematical process for solving a problem using a finite number of steps. Algorithms are a key component of any computer program and are the driving force behind various systems and applications, such as navigation systems, search engines, and music streaming service. (Scribbr, n.d.)

Step 1: Start

Step 2: Display the name of the Rental company, its location, and a motto of company.

Step 3: Display three suitable options for renting, returning the land and exiting the interface or program.

Step 4: Get the user's input.

Step 5: If the user's input is 1, ask user to input his/her name, contact name and validate them.

Step 6: Display the details of lands along with location, kitta, price, availability and ask user to input the kitta number and duration to rent the land.

Step 7: Validate the name and land details along with the price.

Step 8: Update the text file based on the information the user has entered.

Step 9: Ask the user whether continue using the platform or exiting the program.

Step 10: If the users want to rent more land, run the loop again.

Step 11: If the user is done using the platform, generate a bill in terminal.

Step 12: Create an external invoice (rent bill) in the form of txt.file including the name and contact details of user and the total amount.

Step 13: Exit the loop and take the user to first 3 options.

Step 14: if user input is 2, ask username and contact not along with the land kitta number which they want to return. Duration of time rented, and time are also asked for users to calculate the bill along with fine.

Step 15: Validate the information provided by user.

Step 16: Ask the user whether continue using the platform or exiting the program.

Step 17: If the users want to return more land, run the loop again.

Step 18: If the user is done returning the land, generate a bill in terminal.

Step 19: create an external bill in the form of text file including the name and contact details of user.

Step 20: Include the fine price in the bill and display it.

Step 21: Exit the loop and take the user to first 3 options.

Step 24: If the user input is 3, terminate the program.

Step 25: End

3.Pseudocode:

Pseudocode is an effective tool that programmers can use to break large problems down into more digestible sections Similar to an outline for a writer, pseudocode serves as a guide for creating code. Reading pseudocode requires some fundamental programming expertise to comprehend what you're seeing. However, because pseudocode isn't bound to a particular syntax or used as a real programming language, even non-programmers or beginners can understand the high-level concepts of the application's blueprint or early draft. (Team, 2024)

3.1 Pseudocode for main.py

Define function lining ():

PRINT a line of
**********"

Define function main ():

Call lining () function.

PRINT "\n"

PRINT "\t\t\t\t\t\t\t\welcome to Techno Property Nepal \n"

PRINT "\t\t\t\t\t\t\t We are located at Kamalpokhari , Kathmandu | Contact us: 9852678924\n\n\n"

lining ()

PRINT " \t\t\t\Searching For A Property? \n\n\t\t\t\t\t Find it on Techno Property Nepal..."

lining ()

PRINT "\t\tClick here to find the suitable options”!

WHILE True:

lining ()

PRINT "Press 1 to rent land."

PRINT "Press 2 to return land."

PRINT "Press 3 to exit the interface."

```
        lining ()
        TRY:
            choice = GET_USER_INPUT ("Enter your choice: ")
            IF choice == 1 THEN
                operation.rent()
            ELSE IF choice == 2 THEN
                operation.return_rent()
            ELSE IF choice == 3 THEN
                PRINT "Thank You for using our services!!!!!"
                EXIT_PROGRAM ()
            ELSE
                PRINT "Enter the valid option from (1, 2, and 3)"
            lining ()
            continue_using = GET_USER_INPUT ("Do you want to continue using this
platform for rent and returning the land? (y/n):")
            IF continue_using.lower() == 'n' THEN
                PRINT "Thanks for using our service!!!"
                BREAK_LOOP ()
        EXCEPT:
            PRINT "Values can't be left empty”!
            CALL main ()


IF __name__ == '__main__':
    CALL main ()
```

3.2 Pseudocode for read.py

```
Define function read ():
    dictiona = EMPTY_DICTIONARY
    OPEN_FILE ("input.txt", "r") AS file:
        FOR EACH line IN file:
            line = line.strip().split(",")
```

```
        kitta = CONVERT_TO_INTEGER (line [0]. strip ())
        dictiona[kitta] = line [0:]
    CLOSE_FILE. (file)
    RETURN dictiona
```

3.3 Pseudocode for write.py

```
FUNCTION write(dic):
    OPEN_FILE ('input.txt', 'w') AS file:
        FOR EACH value IN dic.values():
            file.write(CONVERT_TO_STRING(value[0]) + ',' +
CONVERT_TO_STRING(value[1]) + ',' + CONVERT_TO_STRING(value[2]) + ',' +
CONVERT_TO_STRING(value[3]) + ',' + CONVERT_TO_STRING(value[4]) + ',' +
CONVERT_TO_STRING(value[-1]))
            file.write("\n")
    CLOSE_FILE (file)


FUNCTION rent_bill(name, phone, kitta, duration, cost):
    dictiona = CALL_FUNCTION read.read()
    total_cost = duration * cost
    OPEN_FILE ('input.txt', 'r') AS file:
        FOR EACH line IN file:
            IF CONVERT_TO_STRING (kitta) IN line THEN
                OPEN_FILE ('rent.txt', 'w') AS data:
                    FOR EACH value IN dictiona.values():
                        IF CONVERT_TO_STRING (kitta) == value [0] THEN
   data. Write ("----------Invoice (Rent Bill) ----------
\n")
                            data. Write ('Customer name: ' + name + '\n')
                            data. Write ('Contact Details: ' + phone + '\n')
                            data. Write("----------
----------\n")
                            data. Write ('Kitta Number: ' + CONVERT_TO_STRING (value [0]) +
'\n')
```

```
                    data. Write ('City: ' + value [1] + '\n')
                    data. Write ('Direction: ' + value [2] + '\n')
                    data. Write ('Rented Duration: ' + CONVERT_TO_STRING (duration)
+ '\n')
                    data. Write ('Land Price: ' + CONVERT_TO_STRING (value [4]) + '\n')
                    data. Write("------------------------------------------------------------------------------
-------------\n")
                    data. Write ('Total Cost: ' + CONVERT_TO_STRING (total_cost) + '\n')
                    data. Write ("------------------------------------------------------------------------------
--------------")
```

3.4 Pseudocode for operation.py

```
FUNCTION calculate fine (expected_return_date, return_date, cost, duration):
    price = duration * cost
    IF return_date > expected_return_date THEN
        delayed_days = (return_date - expected_return_date). days
        fine = 0.10 * (delayed_days * cost)
        total_cost = price + fine
        RETURN total_cost
    ELSE:
        RETURN price.


FUNCTION invoice (name, phone, rented_date, expected_return_date, return_date,
cost, duration):
    total_cost = CALL_FUNCTION calculate_fine(expected_return_date, return_date,
cost, duration)
    bill = "Returned by: " + name + "\n"
    bill += "Contact number: " + phone + "\n"
    bill += "Rent Date: " + rented_date.strftime('%Y-%m-%d') + "\n"
    bill += "Return Date: " + return_date.strftime('%Y-%m-%d') + "\n"
    bill += "Duration of Rent: " + CONVERT_TO_STRING (duration) + " months\n"
```

```
    bill += "Cost per Month: Rs. " + CONVERT_TO_STRING (cost) + "\n"
    bill += "Total Rental Cost: Rs. " + CONVERT_TO_STRING (total_cost) + "\n"
    RETURN bill.

FUNCTION display_invoice():
    PRINT "Invoice Detail"
    OPEN_FILE ('invoice.txt', 'r') AS file:
        FOR EACH line IN file:
            PRINT line.strip()

FUNCTION display_rent_bill():
    OPEN_FILE ('rent.txt', 'r') AS file:
        FOR EACH line IN file:
            PRINT line.strip()

FUNCTION rent ():
    PRINT "Rent"
    TRY:
        dictiona = CALL_FUNCTION read.read()
        name = GET_USER_INPUT ("Enter the name of the customer: ")
        phone = GET_USER_INPUT ("Enter the contact number: ")
        PRINT "Kitta\t\t City\t\t\t Direction \t Anna \t\t Cost (Rs) \t Status"
        OPEN_FILE ("input.txt", "r") AS file:
            FOR EACH line IN file:
                data = line.replace(",", "\t\t")
                PRINT data.
        kitta = CONVERT_TO_INTEGER (GET_USER_INPUT("Enter the kitta number to
rent: "))
        IF kitta IN dictiona THEN
            IF dictiona[kitta][5] == "Available" THEN
```

```
            dictiona[kitta][5] = "Not Available"
            times = CONVERT_TO_INTEGER (GET_USER_INPUT("Enter the duration of time in months: "))
            CALL_FUNCTION write.write(dictiona)
            PRINT "The land is available for rent\n."
            CALL_FUNCTION write.rent_bill(name, phone, kitta, times, CONVERT_TO_INTEGER(dictiona[kitta][4]))
            CALL_FUNCTION display_rent_bill()
        ELSE:
            PRINT "The land is not available."
    ELSE:
        PRINT "Please enter a valid Kitta number."
  EXCEPT Exception AS e:
    PRINT "An error occurred: " + CONVERT_TO_STRING(e)

FUNCTION return_rent():
  TRY:
    name = GET_USER_INPUT ("Enter the name of the customer: ")
    phone = GET_USER_INPUT ("Enter the contact number: ")
    kitta = CONVERT_TO_INTEGER (GET_USER_INPUT ("Enter the kitta number to return: "))
    dictiona = CALL_FUNCTION read.read()
    IF kitta IN dictiona THEN
        times = CONVERT_TO_INTEGER (GET_USER_INPUT ("Enter the duration of time in months: "))
        return_date = CONVERT_TO_INTEGER (GET_USER_INPUT ("Enter the time you rented for (in months): "))
        IF dictiona[kitta][5] == 'Not Available' THEN
            dictiona[kitta][5] = "Available"
            PRINT "Land returned successfully."
            PRINT "Kitta Number\tCity\tDirection\tAnna\tCost (Rs)\tAvailability"
```

```
            FOR EACH values IN dictiona.values():
                PRINT f"{values [0]}
\t{values[1]}\t{values[2]}\t{values[3]}\t{values[4]}\t{values[5]}"
            rented_date = CALL_FUNCTION datetime.now()
            expected_return_date = rented_date.replace(month=rented_date.month +
times)
            return_date = rented_date.replace(month=rented_date.month + return_date)
            bill = CALL_FUNCTION invoice (name, phone, rented_date,
expected_return_date, return_date, CONVERT_TO_INTEGER (dictiona[kitta][4]), times)
            OPEN_FILE ("invoice.txt", 'w') AS file:
                file.write("-----------------------------------------Total Bill-----------------------------
------------------------------\n")
                file.write(bill + '\n')
            PRINT bill.
            CALL_FUNCTION write.write(dictiona)
        ELSE:
            PRINT "The land is not available."
    ELSE:
        PRINT "Please enter a valid Kitta number!"
EXCEPT ValueError:
    PRINT "Enter the valid input."
```

4.Flowchart:

A flowchart is a diagram that shows a process using arrows and symbols. Known by another name, flow diagrams, or flowcharts, are a useful tool for planning, outlining, and recording each step in a process as well as showing connections between them. With a common set of symbols, such as shapes and arrows, flowcharts enable the communication of intricate processes through a common visual language. (mural, 2023)

5.Data structures:

To efficiently access and deal with data, data structures are a means of organizing and storing the information. They explain the connection between the data and the possible operations on it. Instead of becoming bogged down in the specifics of data description and access, data scientists and computer engineers can focus on the bigger picture of addressing problems by using the many different types of data structures that have been described. (geeksforgeeks, 2023)

There are two types of data in python:

a) Primitive data types:
Primitive data types are the built-in data types provided python itself. Some of the primitive data types are:

i) String:
Strings are basically a bunch of characters. String is used in Python by the keyword str.

```
for value in dic.values():
    file.write(str(str(value[0])+','+str(value[1])+','+str(value[2])+','+str(value[3])+' ,' +str(value[4])+',' +str(value[-1])))
    file.write("\n")
```

Figure 1 Implementation of str.

ii) Integer:

Integers are any numeric values in Python. It goes by the “int” keyword.

```
times = int(input("Enter the duration of time in months: "))
return_date = int(input("Enter the time you rented for (in months): "))
```

Figure 2 Implementation of integer.

iii) iv) Boolean: Boolean data types are the types that either have true or false as their only options. Boolean are used to run loops.

```
# If status of kitta  number is available then you can take the rent
if(dictiona[kitta][5]=="Available"):
    dictiona[kitta][5]="Not Available"
```

Figure 3implemantation of Boolean.

iv) Float: Float data types that convert values into point numbers. Float is used for numbers that are with decimal values.

```
fine = 0.10 * (delayed_days * cost)
total_cost = price + fine
```

Figure 4implemantation of float data types

b) Non-primitive data types:
Non-primitive data types are the types of data which are created by users and are not built in.

i) Dictionary:
A dictionary is a mutable container that stores values. Dictionaries in Python store values in form of key-value pairs. They are made using curly bracket.

```
dictiona={}
with open("input.txt","r") as file:
    for line in file:
        line=line.strip().split(",")
        kitta=int(line[0].strip())
        dictiona[kitta]=line[0:]
file.close()
return dictiona
```

Figure 5 Dictionary used in project.

ii) Lists:
Lists are simply a collection of items and one of the most important data structures in Python. Each item in a list is uniquely identified. Items can be added to a list using the append.

```
for value in dic.values():
    file.write(str(str(value[0])+','+str(value[1])+','+str(value[2])+','+str(value[3])+' ,' +str(value[4])+',' +str(value[-1])))
    file.write("\n")
```

Figure 6 implementation of lists.

iii) Set:
Sets are like lists in the sense that they are mutable, but they must contain unique elements. A set cannot contain two same or equal values. Sets can be used when you need specific and non-repetitive data in a stored form.

iv) Tuples:
Tuples are simply lists that are immutable which means that they cannot be changed once they are formed. They are designed by developers to have limitations for certain reasons and operations.

6. Program:

6.1 Implementation of Program

This is a private land rental system implemented in Python file. The file is executed in main.py. The program begins with a welcome message and enables users to search for land. Through a loop, it continuously offers options to rent, return land or exit the interface. Upon selection from user, it calls functions from operation module to execute or run the chosen operation. Users are then asked whether to continue using the platform. Exception handling ensures smooth execution, prompting users to re-enter values if any errors occur.

The read.py file consist of a function that is used the read the text file which has all available lands with its price and location. The function used there helps to read and display the text file in the terminal.Write.py file helps to overwrite the text file and form a new one which helps to store bills. In Write.py we can rent or return the land.Write.py consist of code that is used to print the external invoice bill with the fine price.Input.txt file is the main source or main .txt file which shows or displays all the properties that are both available and unavailable

This program has main validations and exception handling which helps program to run smoothly and tells the user if there are any mistakes. Overall, the program provides a user-friendly interface for renting land or properties.

6.2 Rent and Return of land.

Renting and Returning process is done with all the files used. The main.py is used to call the functions that are creature in other files. Similarly, read.py displays with availability of land, write.py helps to update the text file and create an external text. File. Finally, operation.py carries out all the logic and billing part of the program.

6.3 To Rent Land:

```
Welcome to Techno Property Nepal
We are located at Kamalpokhari , Kathmandu  |  Contact us: 9852678924

Searching For A Property?
Find it on Techno Property Nepal...
Click  here  to find the suitable options
Press 1 to rent land
Press 2  to return land
Press 3 to exit the interface
Enter your choice:
```

Figure 7 initial options

```
Enter the kitta number to rent:102
 Enter the duration of time in month(): 2
 The land is available for rent

----------Invoice (Rent Bill)----------
Customer name: sammar khadka
Contact Details: 9863032731
Kitta Number: 102
City:  Pokhara
Direction:  East
Rented Duration: 2
Land Price:  60000
Total Cost: 120000
```

Figure 8Rent bill.

```
Do you want to continue using this platform for rent and returning the land?(y/n):y
Press 1 to rent land
Press 2  to return land
Press 3 to exit the interface
Enter your choice: 1
 Rent
Enter the name of the customer: sammar khadka
```

Figure 9 option to rent more land.

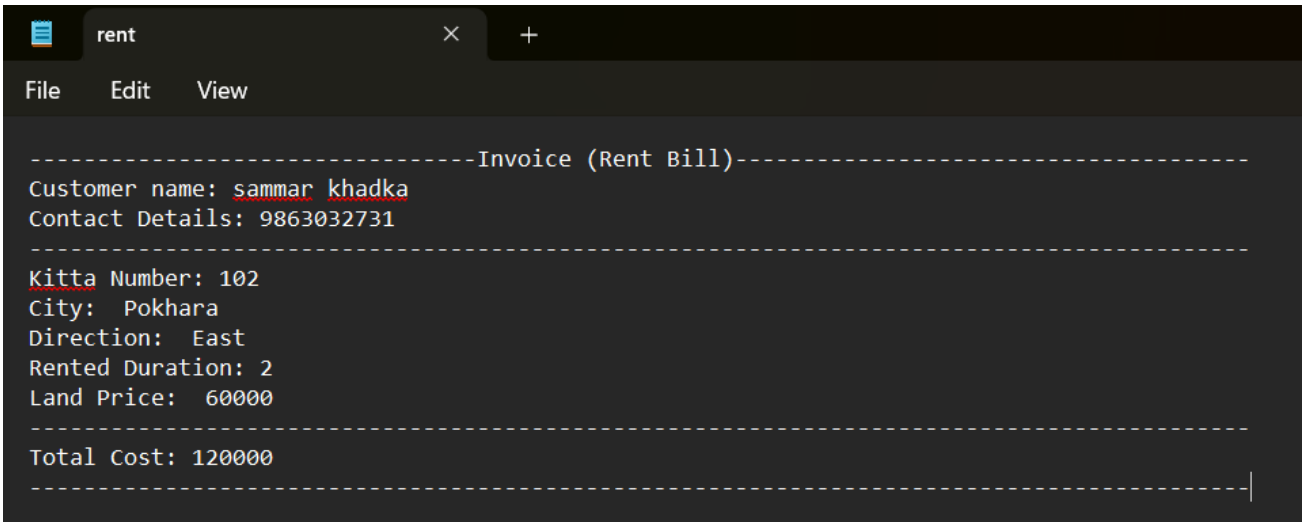


Figure 10 Rent bill print in. txt file.

6.4 For returning:

Figure 11 options displayed to return.

```
Duration of Rent: 3 months
Cost per Month: Rs. 60000
Total Rental Cost: Rs. 732000.0

**********
**********
Do you want to continue using this platform for rent and returning the land?(y/n):y
**********
**********
Press 1 to rent land
Press 2  to return land
Press 3 to exit the interface
**********
**********
Enter your choice: 2
Enter the name of the customer: Sammar
```

Figure 12option to return more land

```
Kitta Number        City        Direction   Anna    Cost (Rs)   Availability
101         Kathmandu           North       4       50000       Available
102         Pokhara     East    5           60000   Available
103         Lalitpur    South   10          100000  Not Available
104         Bhaktapur   West    12          70000   Not Available
105         Biratnagar  South   20          110000  Available
Returned by : sammar khadka
Contact number: 9863032731
Rent Date: 2024-05-06
Return Date: 2024-11-06
Duration of Rent: 3 months
Cost per Month: Rs. 60000
Total Rental Cost: Rs. 732000.0
```

Figure 13 return bill.

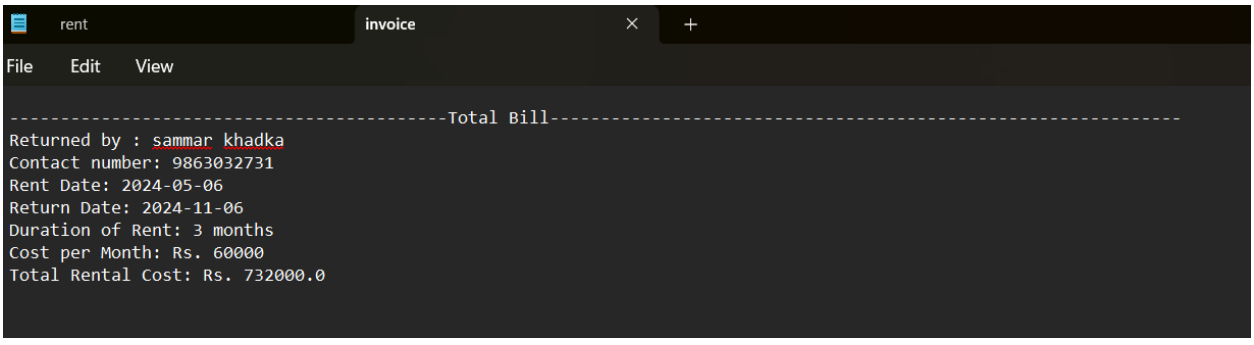


Figure 14return bill in .txt with fine included.

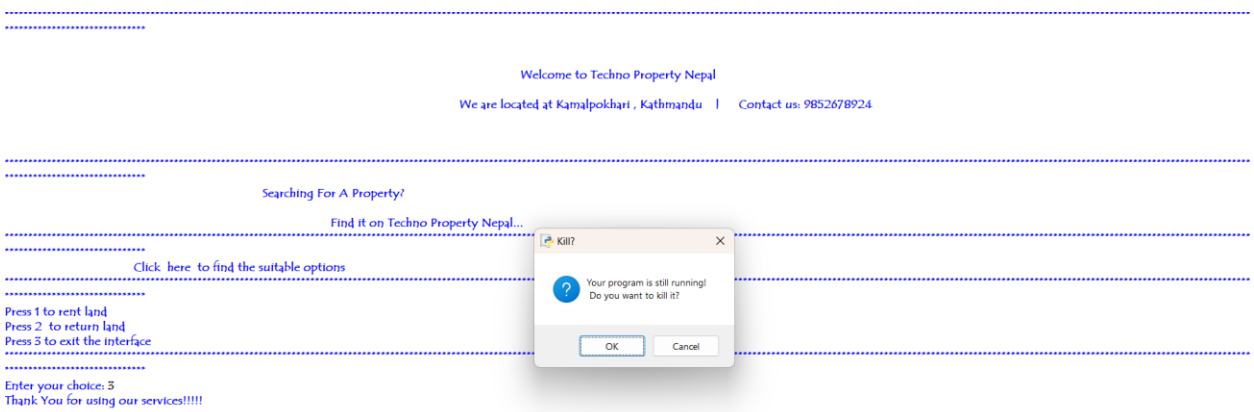


Figure 15exiting the program.

7.Testing

Testing is the process of verifying and validating the actual software with expected requirements to ensure that the product is bult in right way and is bug free. his process is not just restricted to finding faults and bugs in the software but also to improve software efficiency, accuracy, and usability. (Dwivedi, 2023)

7.1 Test 1 = Implementation of try, except

Test No.	1
Objective	To show implementation of try, except
Action	Try block in operation.py. Invalid or different kitta number is entered. Error message is inspected.
Expected Result	Error message will be displayed if the other kitta number which is not given or invalid in .txt file as input is entered.
Actual Result	Error message was displayed.
Conclusion	The test was successful.

Output:

```
try:
    choice=int(input("Enter your choice: "))
    if choice==1:
        operation.rent()
    elif choice==2:
        operation.return_rent()
    elif choice==3:
        print("Thank You for using our services!!!!!")
        exit()
    else:
        print("Enter the valid option from (1,2 and 3)")
    lining()
    continue_using=input("Do you want to continue using this platform for rent and returning the land?(y/n):")
    if(continue_using.lower()=='n'):
        print("Thanks for using our service!!!")
        break
except:
    print("Values cant be set empty")
    main()
```

Figure 16 Implementation of try except.

```
**********
**********
                                                    Searching For A Prc

                                                                  Find it
**********
**********
                                Click  here  to find the suitable options
**********
**********
Press 1 to rent land
Press 2  to return land
Press 3 to exit the interface
**********
**********
Enter your choice: 1
 Rent
Enter the name of the customer: Sammar Khadka
Enter the contact number: 9863052731
Kitta Number            City                    Direction
101                     Kathmandu                       North

102                     Pokhara                 East

103                     Lalitpur                South

104                     Bhaktapur               West

105                     Biratnagar              South

Enter the kitta number to rent:107
Please Enter the valid Kitta number
**********
**********
Do you want to continue using this platform for rent and returning
```

Figure 17 Try except used in program.

7.2. Test 2 – Selection rent and return of lands.

Test No.	2
Objective	To select negative and non-existent selection rent and return of land rent.
Action	Negative value is entered in rent selection of land rent. ➢ Error message is inspected ➢ Non-existent value is entered in rent selection of land rent. ➢ Error message is inspected. ➢ Negative value is entered in return selection of land rent. ➢ Error message is inspected ➢ Non-existent value is entered in return selection of land rent. ➢ Error message is inspected.
Expected Result	Error message will be displayed when any invalid or negative input is inserted in rent selection and return of lands.
Actual Result	The error message was displayed.
Conclusion	Test was successful.

Output:

```
Enter the name of the customer: Sammar Khadka
Enter the contact number: 9863032731
Kitta Number    City        Direction       Anna        Cost (Rs)
101             Kathmandu   North           4           50000       Availa

102             Pokhara     East            5           60000       Available

103             Lalitpur    South           10          100000      Not Available

104             Bhaktapur   West            12          70000       Not Available

105             Biratnagar  South           20          110000      Available

Enter the kitta number to rent:107
Please Enter the valid Kitta number
**********
Do you want to continue using this platform for rent and returning the land?(y/n):y
**********
Press 1 to rent land
Press 2  to return land
Press 3 to exit the interface
**********
Enter your choice: 1
 Rent
Enter the name of the customer: Sammar
Enter the contact number: 9863032731
Kitta Number    City        Direction       Anna        Cost (Rs)
101             Kathmandu   North           4           50000       Availa

102             Pokhara     East            5           60000       Available

103             Lalitpur    South           10          100000      Not Available

104             Bhaktapur   West            12          70000       Not Available

105             Biratnagar  South           20          110000      Available

Enter the kitta number to rent:-101
Please Enter the valid Kitta number
**********
Do you want to continue using this platform for rent and returning the land?(y/n):y
```

Figure 18 Negative and non-existent kitta entered in renting.

```
Do you want to continue using this platform for rent and returning the land?(y/n):y
**********
**********
Press 1 to rent land
Press 2  to return land
Press 3 to exit the interface
**********
**********
Enter your choice: 2
Enter the name of the customer: Sammar Khadka
Enter the contact number: 9863032731
Enter the kitta number to return: -101
Please enter a valid Kitta number!
**********
**********
Do you want to continue using this platform for rent and returning the land?(y/n):y
**********
**********
Press 1 to rent land
Press 2  to return land
Press 3 to exit the interface
**********
**********
Enter your choice: 2
Enter the name of the customer: Sammar Khadka
Enter the contact number: 9863032731
Enter the kitta number to return: 109
Please enter a valid Kitta number!
**********
**********
Do you want to continue using this platform for rent and returning the land?(y/n):
```

Figure 19 Negative and non-existent kitta entered in returning.

7.3 Test-3 File generation of renting of land(s) .

Test No.	3
Objective	To show complete renting process.
Action	Select option 1. ➢ Enter name, contact of buyer ➢ Enter the kitta numbers of land. ➢ Enter multiple kitta numbers. ➢ Inspect shell after renting. ➢ Inspect the .txt file after renting.
Expected Result	The selected lands will be rented and displayed in the shell and text file.
Actual Result	The selected lands were rented and displayed in the shell and text file
Conclusion	The test was successful.

Output:

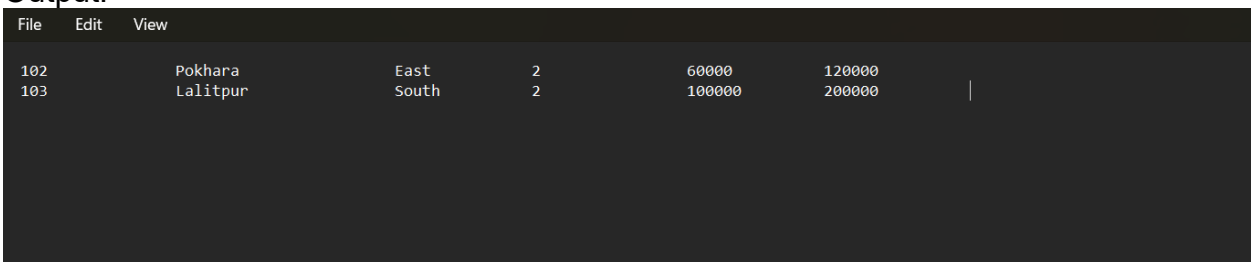


Figure 20 output in txtfile

```
102             Pokhara         East            5               60000           Not Available
103             Lalitpur        South           10              100000          Not Available
104             Bhaktapur       West            12              70000           Not Available
105             Biratnagar      South           20              110000          Not Available

Enter the kitta number to rent:102
 The land is not available
**********
**********
Do you want to continue using this platform for rent and returning the land?(y/n):y
**********
**********
Press 1 to rent land
Press 2  to return land
Press 3 to exit the interface
**********
**********
Enter your choice: 2
Enter the name of the customer: Sammar
Enter the contact number: 9863032731
Enter the kitta number to return: 102
Enter the duration of time in months: 4
Enter the time you rented for (in months): 5
Land returned successfully
Kitta Number            City            Direction       Anna            Cost (Rs)       Availability
101             Kathmandu               North           4               50000           Not Available
102             Pokhara         East            5               60000           Available
103             Lalitpur        South           10              100000          Not Available
104             Bhaktapur       West            12              70000           Not Available
105             Biratnagar      South           20              110000          Not Available
Returned by : Sammar
Contact number: 9863032731
Rent Date: 2024-05-07
Return Date: 2024-10-07
Duration of Rent: 4 months
Cost per Month: Rs. 60000
Total Rental Cost: Rs. 420000.0

**********
**********
Do you want to continue using this platform for rent and returning the land?(y/n):
```

Figure 21 selection of multiple land at once

7.4 Test-4 File generation of renting of land(s).

Test No	4
Objective	To show complete sales process.
Action	➢ Select option 1 ➢ Enter name, contact of buyer ➢ Enter the kitta numbers of land. ➢ Enter multiple kitta numbers. ➢ Inspect shell after returning. ➢ Inspect the text file after returning.
Expected Result	The kitta selected by the user will be returned and displayed in the shell and text file.

Actual Result	The kitta selected by the user were returned and displayed in the shell and text file.
Conclusion	The test was successful.

7.5 Test 5 - Show the update in stock of land(s)

Test no.	5
Objective	To Show the availability land being rented to not available after renting the land.
Action	Inspect return.txt file. Select rent option. Enter name, contact. Enter kitta number. Reinspect. return.txt file after renting. Inspect return.txt file. Select return option. Enter name, contact. Enter the kitta number. Reinspect return.txt file.
Expected Result	While renting, available land should be change into unavailable and while

	returning, unavailable land should be available.
Actual Result	While renting, available land was changed into unavailable and while returning, available land was changed in unavailable land.
Conclusion	The test was successful.

Output:

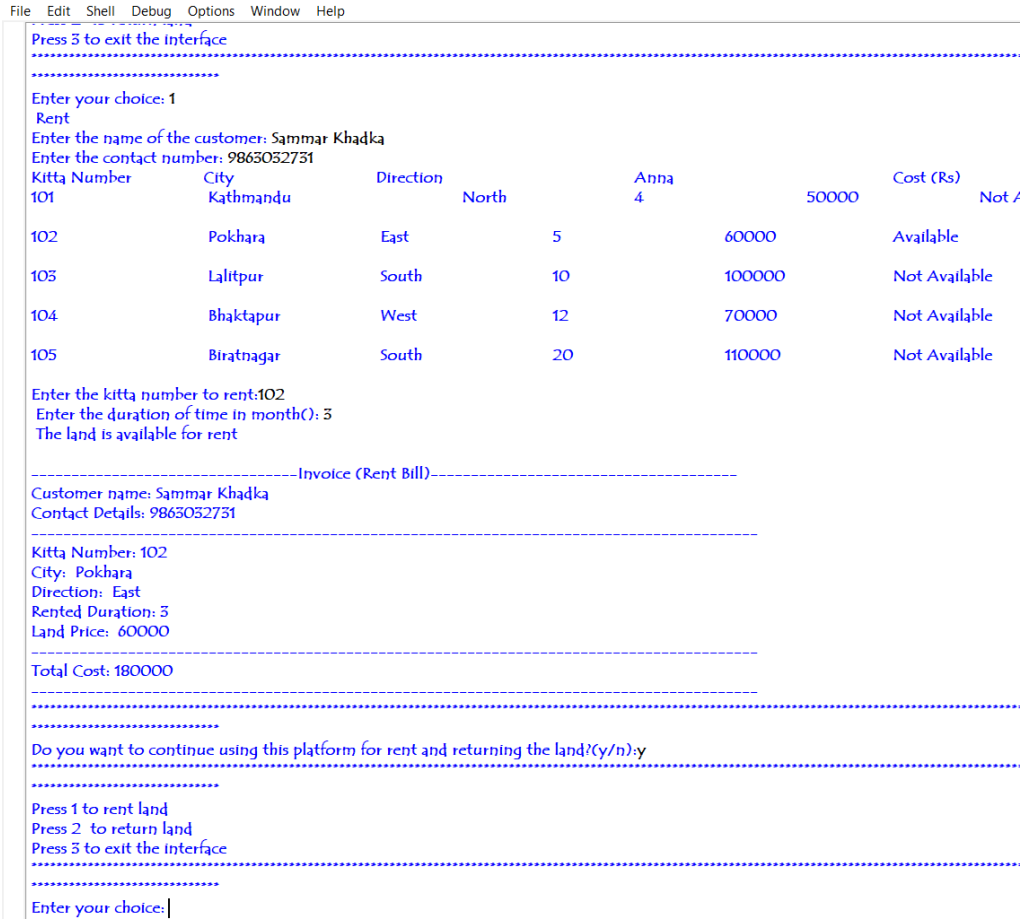


Figure 22 rent bill printed in terminal

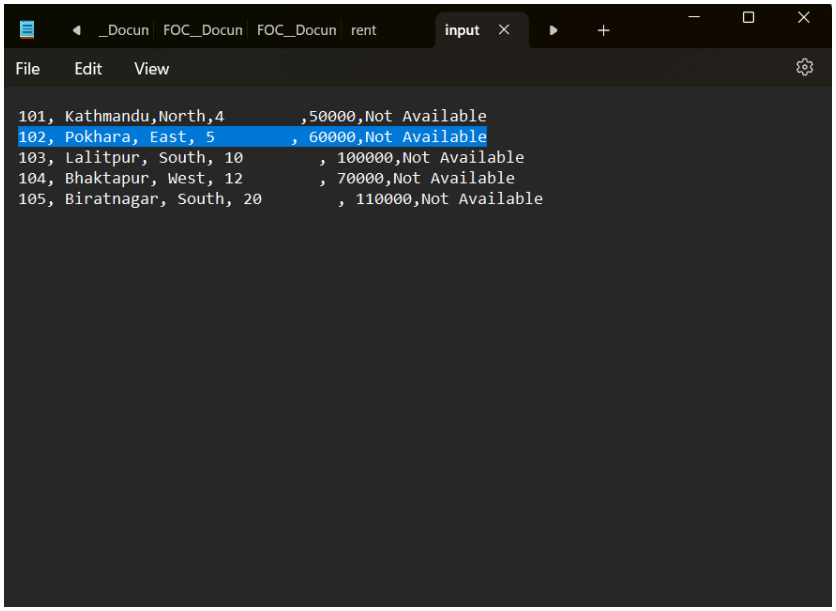


Figure 23 input.txt updated and changed from available to unavailable.

```
102          Pokhara       East          5          60000          Not Available

103          Lalitpur      South         10         100000         Not Available

104          Bhaktapur     West          12         70000          Not Available

105          Biratnagar    South         20         110000         Not Available

Enter the kitta number to rent:102
 The land is not available
**********
**********
Do you want to continue using this platform for rent and returning the land?(y/n):y
**********
**********
Press 1 to rent land
Press 2  to return land
Press 3 to exit the interface
**********
**********
Enter your choice: 2
Enter the name of the customer: Sammar
Enter the contact number: 9863032731
Enter the kitta number to return: 102
Enter the duration of time in months: 4
Enter the time you rented for (in months): 5
Land returned successfully
Kitta Number        City        Direction   Anna      Cost (Rs)   Availability
101          Kathmandu          North       4         50000       Not Available
102          Pokhara     East    5         60000     Available
103          Lalitpur    South   10        100000    Not Available
104          Bhaktapur   West    12        70000     Not Available
105          Biratnagar  South   20        110000    Not Available
Returned by : Sammar
Contact number: 9863032731
Rent Date: 2024-05-07
Return Date: 2024-10-07
Duration of Rent: 4 months
Cost per Month: Rs. 60000
Total Rental Cost: Rs. 420000.0
**********
**********
Do you want to continue using this platform for rent and returning the land?(y/n):
```

Figure 24 Return bill printed on terminal.

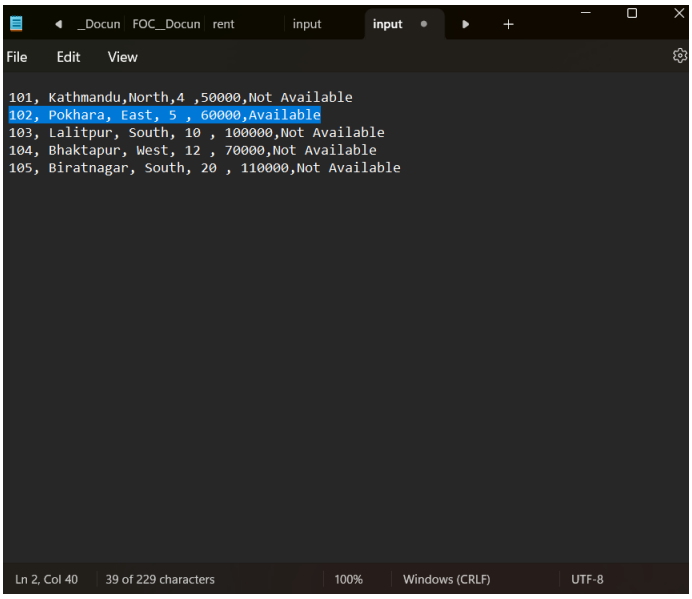


Figure 25 input.txt updated and changed from not available to available.

8.Conclusion

The coursework is aimed to create a program that could rent and return land of various location using Python programming language. The program involved separating the codes into four different sections (operations, main, read and write). Each sections have their own functions. After the completion of this coursework, it showed the understanding problem solving in Python and apply it in real world.

Data structures, flowcharts, pseudocode, and algorithms are used in the program report to emphasize the main elements of the code. It gave us the ideal chance to put into practice everything we had learned about Python over the previous few months. The program is simple and can be easily modified according to the needs of Company. However, development of program was not very easy, but it can provide a strong framework for future. From this project I clearly understood the concepts of data structures, exception handling, reading and overwriting txt files. This project helped us to showcase our problem-solving skill and gave chance to show our creativity. With the completion of this project, I understand the concept of Python and can implement it for a real-world scene. This coursework involves various research practices as there’re various ways to write or develop the program.

Various logics can be used to create the same type of code. The process of learning to create bill also required research. This was a worthwhile educational experience that has boosted my self-assurance in this line of work. To sum up, the coursework gave us a fantastic opportunity to learn about Python and how it is used in practical situations. In future I can work on this type of project as this project helped me a lot to know about python and also helped me to gain confidence.

9) Appendix

I) Code for main.py:

```
import operation

def lining():


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

def main():

    lining()

    print("\n")

    print("\t\t\t\t\t\t\t\tWelcome to Techno Property Nepal \n")

    print("\t\t\t\t\t\t\t We are located at Kamalpokhari , Kathmandu    |      Contact us:
9852678924\n\n\n")

    lining()

    print(" \t\t\t\tSearching For A Property? \n\n\t\t\t\t\t Find it on Techno Property Nepal...")

    lining()

    print("\t\tClick  here  to find the suitable options")

    while(True):

        lining()

        print("Press 1 to rent land")

        print("Press 2  to return land")

        print("Press 3 to exit the interface")

        lining()

        try:

            choice=int(input("Enter your choice: "))

            if choice==1:

                operation.rent()

            elif choice==2:

                operation.return_rent()
```

```
            elif choice==3:
                print("Thank You for using our services!!!!!")
                exit()
            else:
                print("Enter the valid option from (1,2 and 3)")
            lining()
            continue_using=input("Do you want to continue using this platform for rent and returning the land?(y/n):")
            if(continue_using.lower()=='n'):
                print("Thanks for using our service!!!")
                break
        except:
            print("Values cant be set empty")
            main()


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

ii)Code for write.py

```
# Function to write in a file
import read
from datetime import date
def  write(dic):
    with open('input.txt' ,'w' ) as file:
        for value in dic.values():
            file.write(str(str(value[0])+','+str(value[1])+','+str(value[2])+','+str(value[3])+' ,' +str(value[4])+',' +str(value[-1])))
            file.write("\n")
        file.close()
def rent_bill(kitta,duration):
```

```
    dictiona=read.read()
    total_cost=duration*int(dictiona[kitta][4])
    with open('input.txt','r') as file:
        for line in file:
            if str(kitta) in line:
                with open('rent.txt','a') as data:
                    for value in dictiona.values():
                        if str(kitta)==value[0]:
                            data.write(f"{value[0]} \t\t")
                            data.write(f"{value[1]} \t\t")
                            data.write(f"{value[2]} \t\t")
                            data.write(f"{duration} \t\t")
                            data.write(f"{value[4]} \t\t")
                            data.write(f"{total_cost} \t\t")
```

ii) Code for read.py.

```
def read():
    dictiona={}
    with open("input.txt","r") as file:
        for line in file:
            line=line.strip().split(",")
            kitta=int(line[0].strip())
            dictiona[kitta]=line[0:]
    file.close()
    return dictiona
```

iii) Code for operation.py.

```
from datetime import datetime,timedelta
import write
import read
```

```
from write import rent_bill
def calculate_fine(expected_return_date,return_date,cost,duration):
    price = duration * cost
    if return_date > expected_return_date:
      delayed_days = (return_date - expected_return_date).days
      fine = 0.10 * (delayed_days * cost)
      total_cost = price + fine
      return total_cost

    else:
       return price

def invoice(name,phone,rented_date,expected_return_date,return_date,cost,duration):
    total_cost=calculate_fine(expected_return_date,return_date,cost,duration)
    bill =f"Returned by : {name} \n"
    bill +=f"Contact number: {phone} \n"
    bill += f"Rent Date: {rented_date.strftime('%Y-%m-%d')}\n"
    bill += f"Return Date: {return_date.strftime('%Y-%m-%d')}\n"
    bill += f"Duration of Rent: {duration} months\n"
    bill += f"Cost per Month: Rs. {cost}\n"
    bill += f"Total Rental Cost: Rs. {total_cost}\n"
    return bill

def display_invoice():
    print("Invoice Detail")
    with open('invoice.txt','r') as file:
       for line in file:
          print(line.strip())
```

```
def display_rent_bill():
    with open('rent.txt','r') as file:
        for line in file:
            print(line.strip())
def rent():
    print(" Rent")
    try:
        dictiona=read.read()
        rented_kittas=[]
        duration=0
        name=input("Enter the name of the customer: ")
        phone=input("Enter the contact number: ")
        c='y'
        while(c=='y'):
            print("Kitta\t\t City\t\t Direction \t Anna \t\t Cost (Rs) \t  Status")
            file=open("input.txt","r")
            for line in file:
                data=line.replace(",","\t\t")
                print(data)
            kitta=int(input(("Enter the kitta number to rent:")))
            if kitta in dictiona:
                # If status of kitta  number is available then you can take the rent
                if(dictiona[kitta][5]=="Available"):
                    dictiona[kitta][5]="Not Available"
                    duration=int(input(" Enter the duration of time in month(): "))
                    write.write(dictiona)
                    print(" The land is available for rent\n")
```

```
                rented_kittas.append(kitta)
            else:
                print(" The land is not available")
        else:
            print("Please Enter the valid Kitta number")

        c=input("DO you want to continue?:")
        if(c.lower()=='n'):
            break
    for kittas in rented_kittas:
        rent_bill(kittas, duration)
        display_rent_bill()
except Exception as e:
    print(f"An error occurred: {str(e)}")



def return_rent():
    try:
        name = input("Enter the name of the customer: ")
        phone = input("Enter the contact number: ")
        kitta = int(input("Enter the kitta number to return: "))



        dictiona = read.read()
        if kitta in dictiona:

            times = int(input("Enter the duration of time in months: "))
            return_date = int(input("Enter the time you rented for (in months): "))
            if dictiona[kitta][5] == 'Not Available':
```

```
                dictiona[kitta][5] = "Available"
                print("Land returned successfully")
                print("Kitta \tCity\t\tDirection\tAnna\tCost (Rs)\tAvailability")
                for values in dictiona.values():
                    print(f"{values[0]}\t{values[1]}\t{values[2]}\t{values[3]}\t{values[4]}\t{values[5]}")
                # Taking today's date
                rented_date = datetime.now()
                expected_return_date = rented_date.replace(month=rented_date.month + times)
                return_date = rented_date.replace(month=rented_date.month + return_date)
                bill = invoice(name, phone, rented_date, expected_return_date, return_date,
int(dictiona[kitta][4]), times)
                with open("invoice.txt",'w') as file:
                    file.write("----------------------------------------Total Bill----------------------------------------------
----------\n")
                    file.write(bill+'\n')
                print(bill)
                write. write(dictiona)
            else:
                print("The land is not available")
        else:
            print("Please enter a valid Kitta number!")
    except ValueError:
        print("Enter the valid input")
```