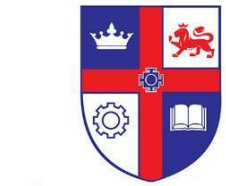
LONDON METROPOLITAN UNIVERSITY



islington college
(इस्लिङ्टन कलेज)

CS4051NI Fundamentals of Computing

60% Individual Coursework

2023/24 Spring

Student Name: Prashant Rijal

London Met ID: 23048683

College ID: np01ai4a230142

Assignment Due Date: Tuesday, May 7, 2024

Assignment Submission Date: Tuesday, May 7, 2024

Word Count: 242

Project File Links:

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

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

Table of Contents

List of Figures

List of Tables

Introduction

The project is a miniature portal for land rent and return management systems. The system is created using Python programming language. Throughout this report, we explore the key components and features of the Land Management System, including:

Data reading: This project utilizes the data stored in a file and presents that data to the user for him/her to access the data of lands.

Data manipulation: The data that has been taken from stored files we make changes to the data if the user completes some actions in this case rent or return.

User Interface and Accessibility: design user experience with an emphasis on intuitive navigation and interactive functionality rather than fancy looking but hard to use interface.

The goal of this project is to understand how Python is used in real-time scenarios and to familiarize myself with the concept of Procedural programming as well as Object Oriented Programming in Python.

Algorithm

Step 1: display a welcome message.

Step 2: Display the land info from the file.

Step3: display the option for user

Step 4: take input from user.as “1”,”2”or 3

Step 5: if the user input is “1” take another input from user which is key

Step 6: if the key user provides is true display land is available

Step 7: input name

Step 8: input age

Step 9: input address

Step 10: input contact information

Step 11: input the number of month to rent the land

Step 12: ask the user if he/she wants to rent another land

Step 13: if the user inputs yes repeat the process.

Step 14: if the user inputs no then make them go back to step 4

Step 15: if the user inputs 2 then display you have chosen to return the land

Step 16: ask the user to input the key of the land he wants to return.

Step 17: check if the key user provides has status not available.

Step 18: if the land is not available display the land is returnable

Step 18: ask for the invoice no

Step 19: if the invoice no is true then ask for the following.

Step 20: input name

Step 21: input age

Step 22: input address

Step 23: input contact information

Step 24: input the number of months you exceed in use

Step 25: Create an invoice.

Step 26: display land is returned

Step 27: return to step 4

Step 28: if the user enters 3 exit

Step 29: display exit message

Flowchart

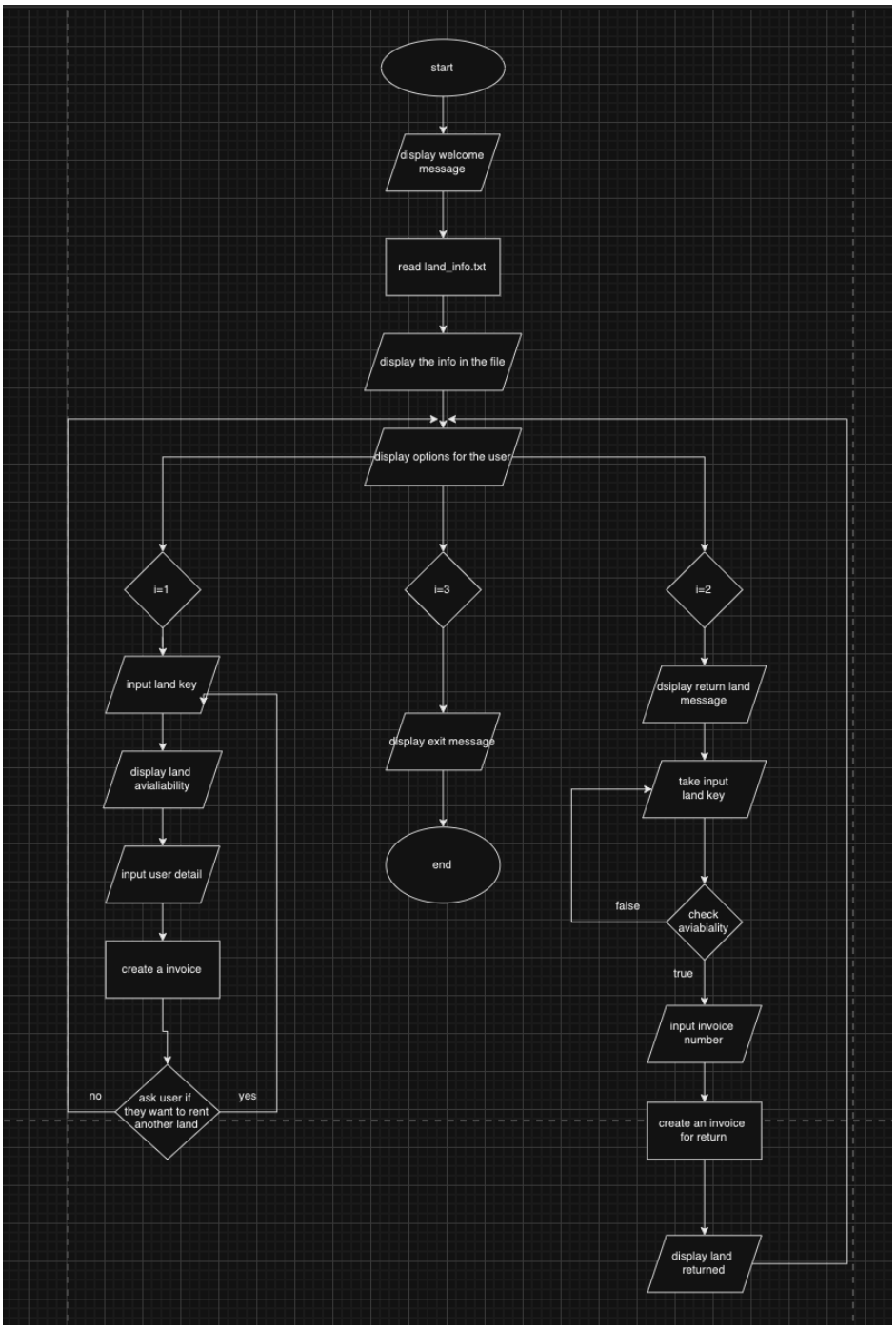


Figure 1 flowchart

Pseudocode

4.1 read.py

Do

Import datetime

Create a function rent_land with parameter data_dict

Do

While the condition is true

Do

Take input from the user and store it in a variable named key
If the key is in data_dict

Do
Value = data_dict[key]
Display the key and value
If the last element of the line is “Available”

DO
Display land is available.

Ask the user to input their information.

Create a variable rent_month and ask the user to set no of moth in it
In a variable named “cost” take the price of land from the file and convert it to an integer to store in this variable.

Calculate cost = cost * rent_month

Create variables minute, second, and microsecond to store the minute, second and microsecond from the imported function(datetime) in the form of string

create a variable named random and add the minute, second, and microsecond in it.

Create a file invoice_file_name and assign it to “random.txt”.

Open the invoice_file_name in write mode as invoice_file.

Write all the data that user has provided and the total cost in this file

Display invoice creation message with invoice name.

Open the dictionary with the key the user provided and remove the "Available" then add "Not Available" in the value of the key.

Open the land_info.txt file in write mode as file

For each value in data_dict
Do
write the values in file.

End do

End do

Else

Do
display land not available select another key.
Return to point where user enters key
End do

End do
Else

Do
display invalid key please enter again
return to the point where the user enters the key

End do

End do

End do

End do

4.2 write.py

Do

Import datetime

Create a function return_land with parameter data_dict
Display rent chosen message

Do

While the condition is true

Do

Take input from the user and store it in a variable named key
If the key is in data_dict

Do
Value = data_dict[key]
Display the key and value
If the last element of the line is “Not Available”

DO
Display land is not available so it can be returned.

Ask the user to input invoice number

Create a variable invoice_file_name and assign its value as invoice.txt

Create a variable invoice_exist and set it as False

Try

Open file invoice_file_name in read mode
Invoice_exist=True

Except

Print(‘file not found’)

If invoice_exist is True
Do
Try

Open invoice_file_name in read mode as file
For lines in file
Do
If “key=”user input key in file
Do
Set Invoice_exist to True
Return False
End Do
End Do
Except
display file error message
set invoice_exist to False

if invoice_exist is True
Do
Ask the user to input their information.
Create a variable user_month and ask the user to set no of extra months in it.
In a variable named “fine” do the following calculation
Take the price of land from the values and multiply it with user_month.
Then multiply the output by 10% or 10/100

Create variables minute, second, and microsecond to store the minute, second and microsecond from the imported function(datetime) in the form of string

create a variable named random and add the minute, second, and microsecond in it.

Create a file invoice_file_name and assign it to “random.txt”.

Open the invoice_file_name in write mode as invoice_file.

Write all the data that user has provided and the total cost in this file

Display invoice creation message with invoice name.

Open the dictionary with the key the user provided and remove the “Not Available” then add “Available” in the value of the key.

Open the land_info.txt file in write mode as file

For each value in data_dict

```
                    Do
                        write the values in file.

                    End Do

                End Do

            End Do

        End Do
            Else

            Do
                display invalid key message.
                Return to point where user enters key
            End do
    End Do

  End Do

End Do
```

4.3 operation.py

DO

Do

Create a function read_file_into_dictionary with parameter file_path
Do
Create a blank dictionary as data_dict
Open file file_path in read mode as file
Declare a counter line_number = 1
For line in file

Do

Assign data_dict with keys line_number as line and remove whitespace from line
increase the counter line_number by 1

End Do

Return the final value of data_dict

Do

Create a function print_data with parameter as data_dict

Do

Print key, kitta no, direction, address, price, and status
For key, value in data_dict
Do

Display key and value
End Do

End Do

End Do

End Do

END DO

4.4 main.py

DO

Import the python files operation, read and write
Print welcome message
Define file_path as 'land_info.txt'
Assign data_dictionary to operation.read_file_into_dictionary with parameter as file_path.
Call the function print_data from operation and assign its parameter as data_dictionary
create a variable cond and set its value to True
While cond is True
Do

Print the options users have
Take input from the user as 1, 2, or 3
If the user doesn't input 1, 2, or 3 or leaves it blank
Do

Print invalid input and ask the user to re-enter the value.

End Do

If the user input is 1
Do

Print you have chosen to rent land.
Assign variable condi as true
While condi is true
Do

Call function rent_land from read.py
Create a variable condition and ask the user to input yes if they want to rent another land
If the condition = "yes"
Do

Call the function rent_land from read.py
Set the condition to true
Break the loop
End Do

Else

Do

Assign condi to false
End Do

End Do

End Do

if user input is 2
Do

Call the function return_land from the write.py and assign its parameter as function read_file_into_dictionary with parameter as file_path from operation.py.

Set the condi to True.

End Do

if user input is 3
Do

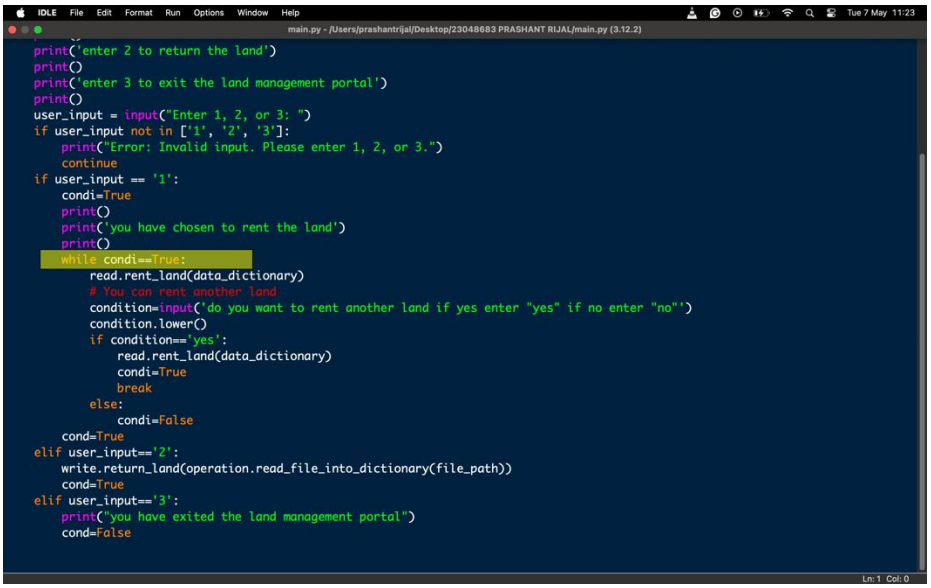
Display exit message

End Do

Data Structure

Some of the implemented data types in the code are:

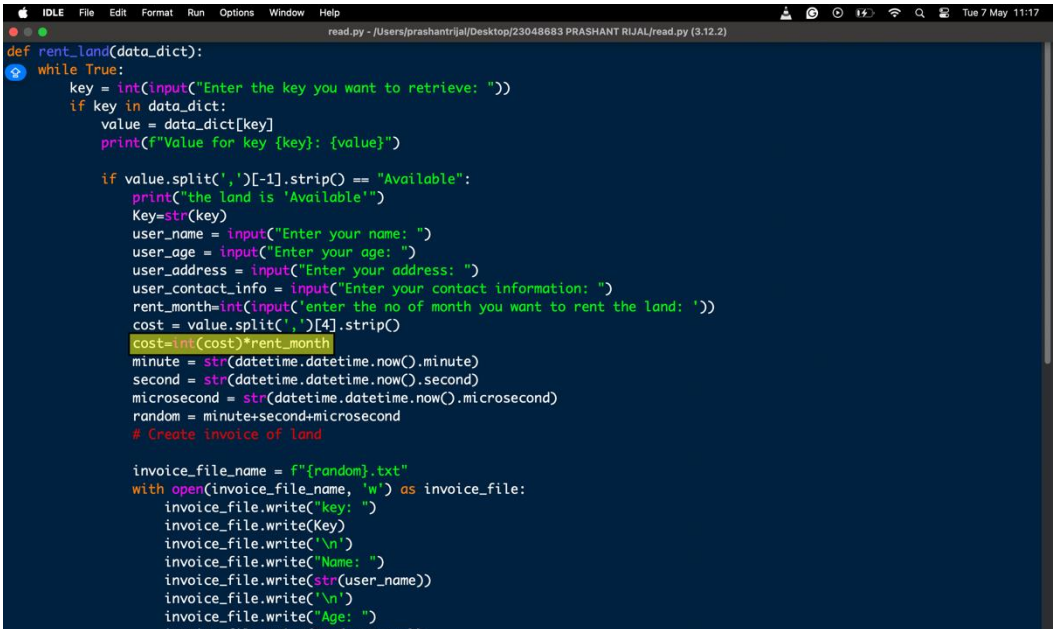
- Integer: this data type is used to do calculations in the code which cant be done if the values are in string form.



```
print('enter 2 to return the land')
print()
print('enter 3 to exit the land management portal')
print()
user_input = input("Enter 1, 2, or 3: ")
if user_input not in ['1', '2', '3']:
    print("Error: Invalid input. Please enter 1, 2, or 3.")
    continue
if user_input == '1':
    condi=True
    print()
    print('you have chosen to rent the land')
    print()
    while condi==True:
        read.rent_land(data_dictionary)
        # You can rent another land
        condition=input('do you want to rent another land if yes enter "yes" if no enter "no"')
        condition.lower()
        if condition=='yes':
            read.rent_land(data_dictionary)
            condi=True
            break
        else:
            condi=False
    cond=True
elif user_input=='2':
    write.return_land(operation.read_file_into_dictionary(file_path))
    cond=True
elif user_input=='3':
    print("you have exited the land management portal")
    cond=False
```

Figure 2 integer example

- Boolean: this data type is one of the most used types in the program because if can set a variable to True or False which helps in looping and applying certain conditions



```
def rent_land(data_dict):
    while True:
        key = int(input("Enter the key you want to retrieve: "))
        if key in data_dict:
            value = data_dict[key]
            print(f"Value for key {key}: {value}")

            if value.split(',')[-1].strip() == "Available":
                print("the land is 'Available'")
                Key=str(key)
                user_name = input("Enter your name: ")
                user_age = input("Enter your age: ")
                user_address = input("Enter your address: ")
                user_contact_info = input("Enter your contact information: ")
                rent_month=int(input('enter the no of month you want to rent the land: '))
                cost = value.split(',')[4].strip()
                cost=int(cost)*rent_month
                minute = str(datetime.datetime.now().minute)
                second = str(datetime.datetime.now().second)
                microsecond = str(datetime.datetime.now().microsecond)
                random = minute+second+microsecond
                # Create invoice of land

                invoice_file_name = f"{random}.txt"
                with open(invoice_file_name, 'w') as invoice_file:
                    invoice_file.write("key: ")
                    invoice_file.write(Key)
                    invoice_file.write('\n')
                    invoice_file.write("Name: ")
                    invoice_file.write(str(user_name))
                    invoice_file.write('\n')
                    invoice_file.write("Age: ")
```

Figure 3 boolean example

- String: It is the default data type when we enter something using input. It is mostly used in this program to write the data in text files as it can only store string data type.

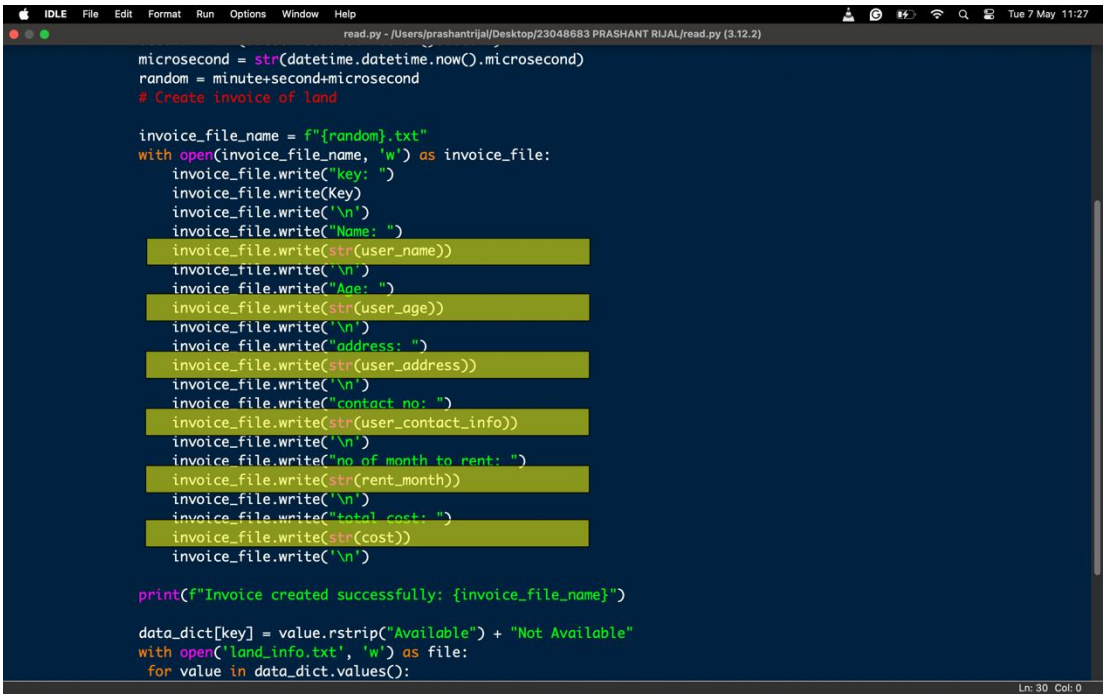


Figure 4 String example

Program

The system has a very light interface with no complicated visuals. So when we run the program the first welcome and list shows up

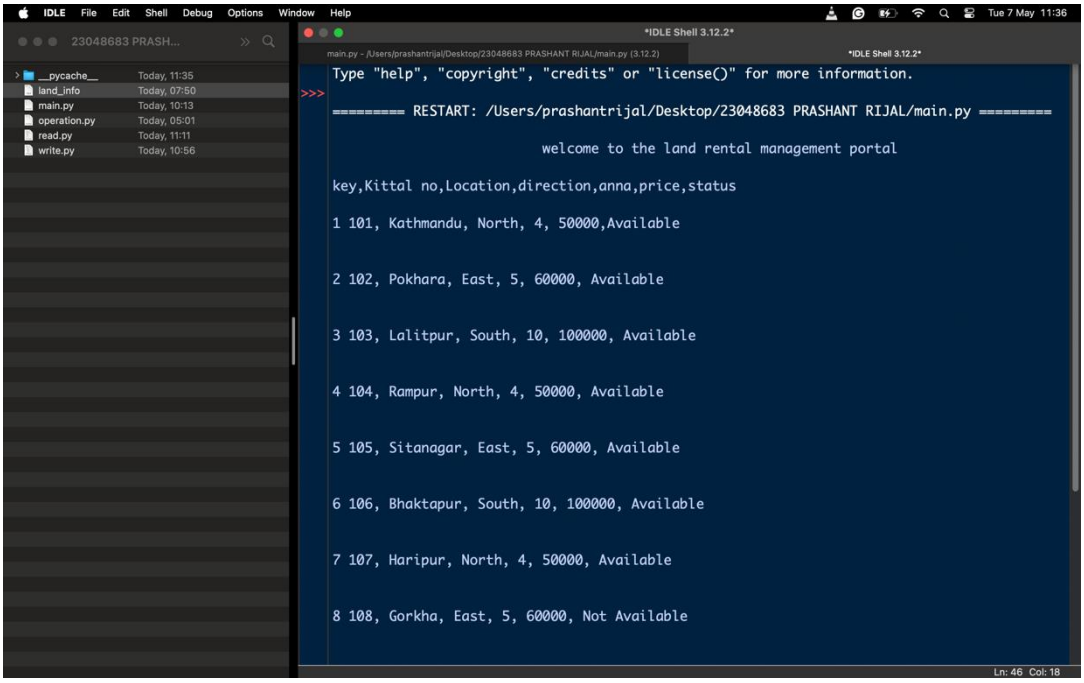


Figure 5 welcome page

Here the list will tell you if the land is available or not now if we scroll down a little the option and user input come up.

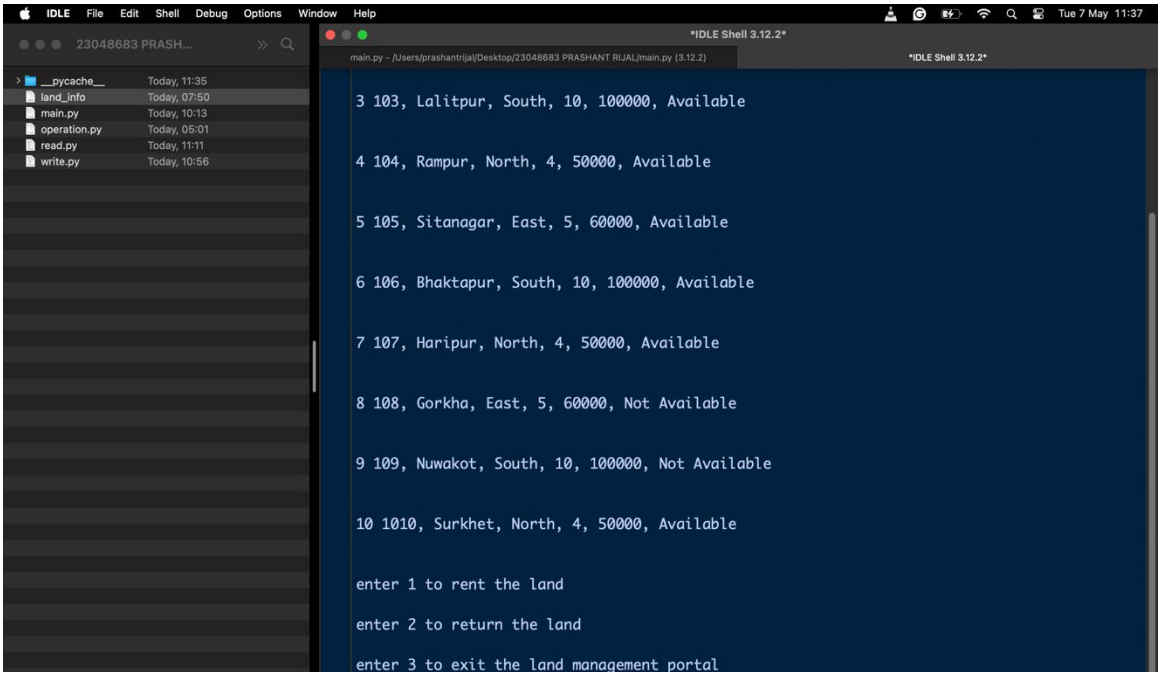


Figure 6 user input option

Now we can 1 to rent the land after pressing 1 a message shows up saying I have chosen to rent the land.

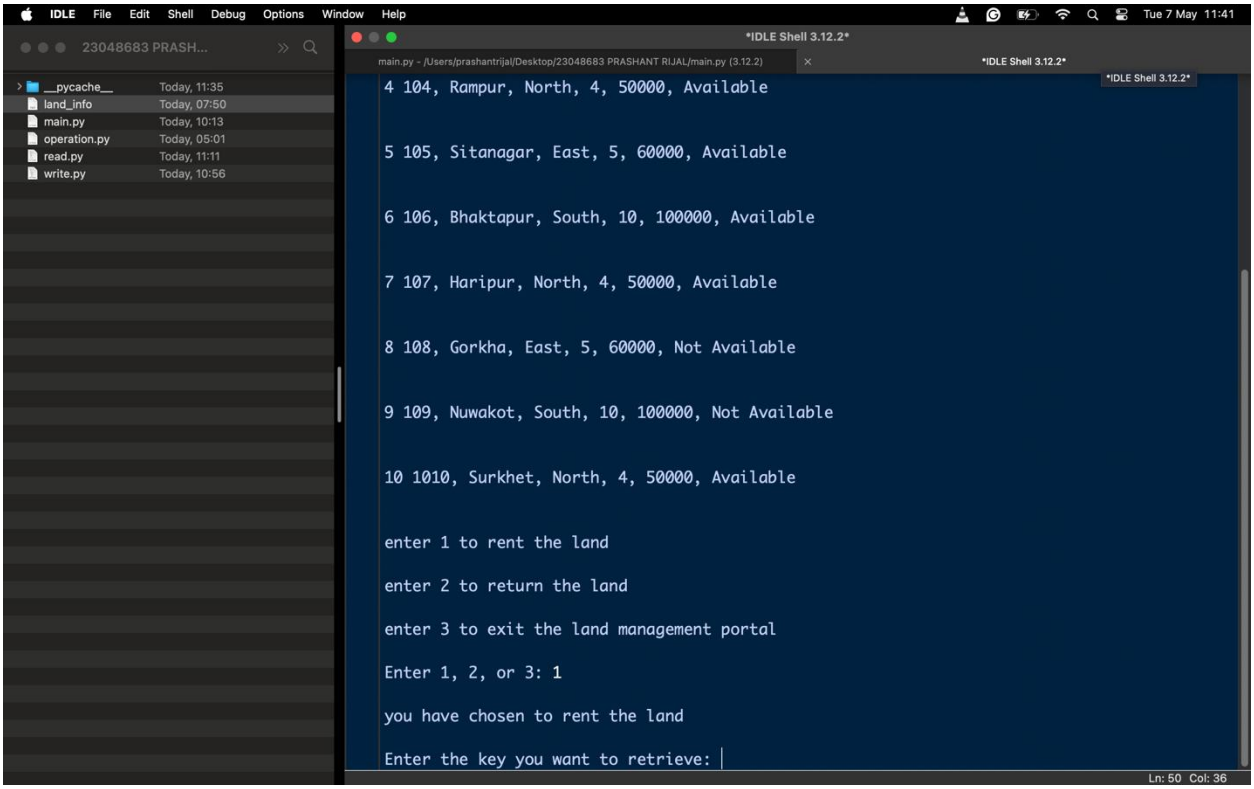


Figure 7 rent start

Now we enter the key of land we want to rent after entering it will show u your choice and also confirm if the land is available.

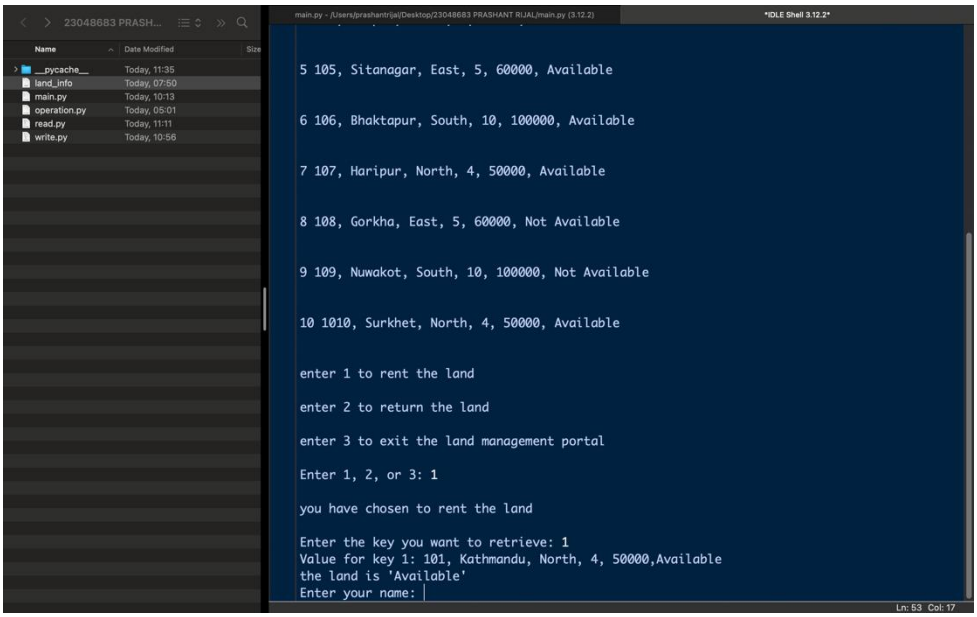


Figure 8 availability check

Now we fill in our information and then the no of month we want to rent land doing this creates an invoice and a message about invoice creation appears. Then it gives us option to rent another land.

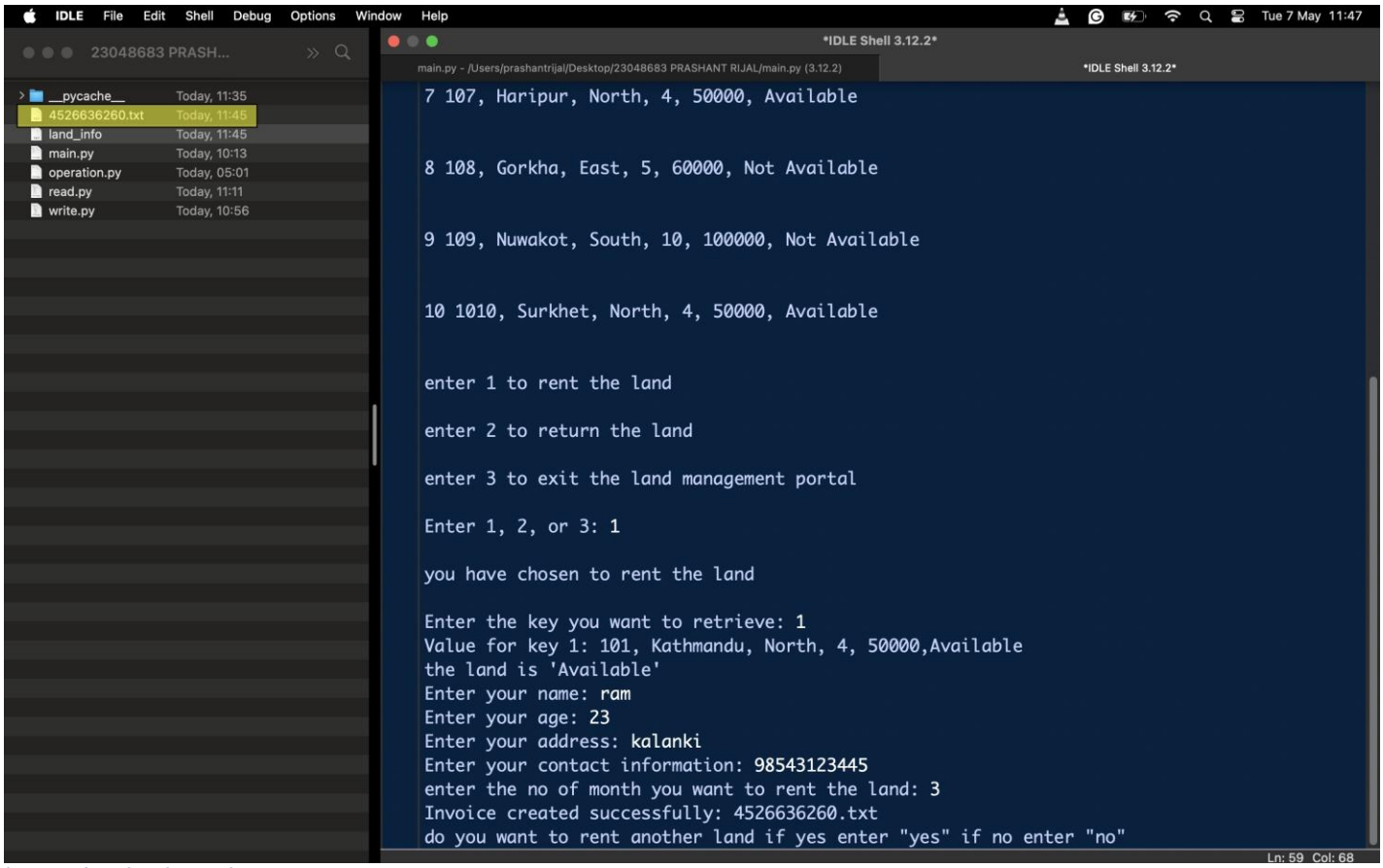


Figure 9 invoice formation

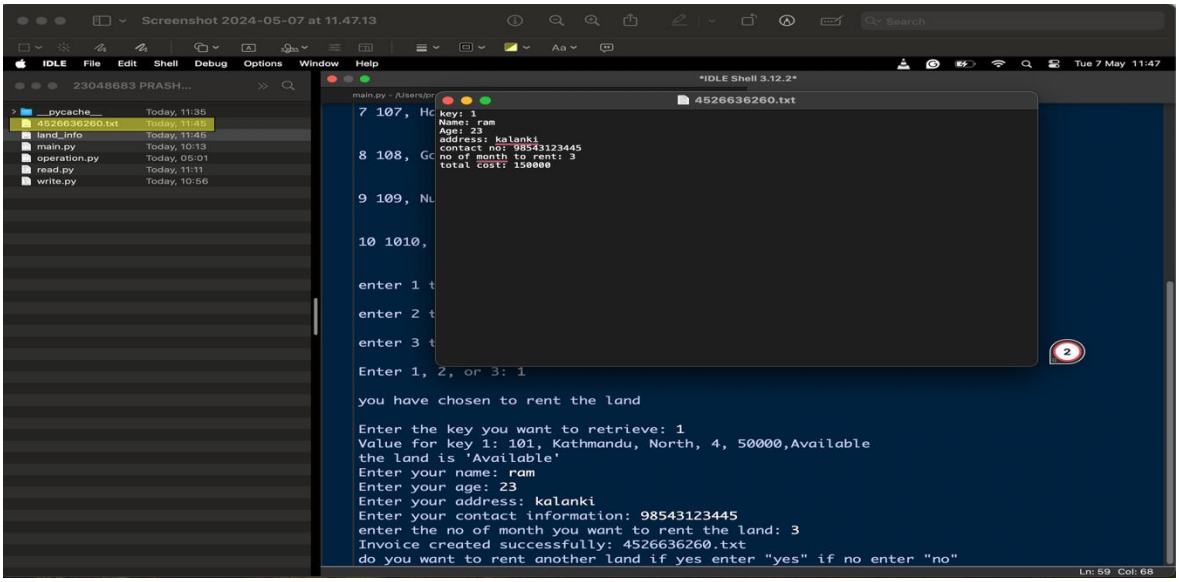


Figure 10 invoice

For now, we will not rent 2 lands so we input no after doing so it again takes you to the user option selection that was in the beginning.

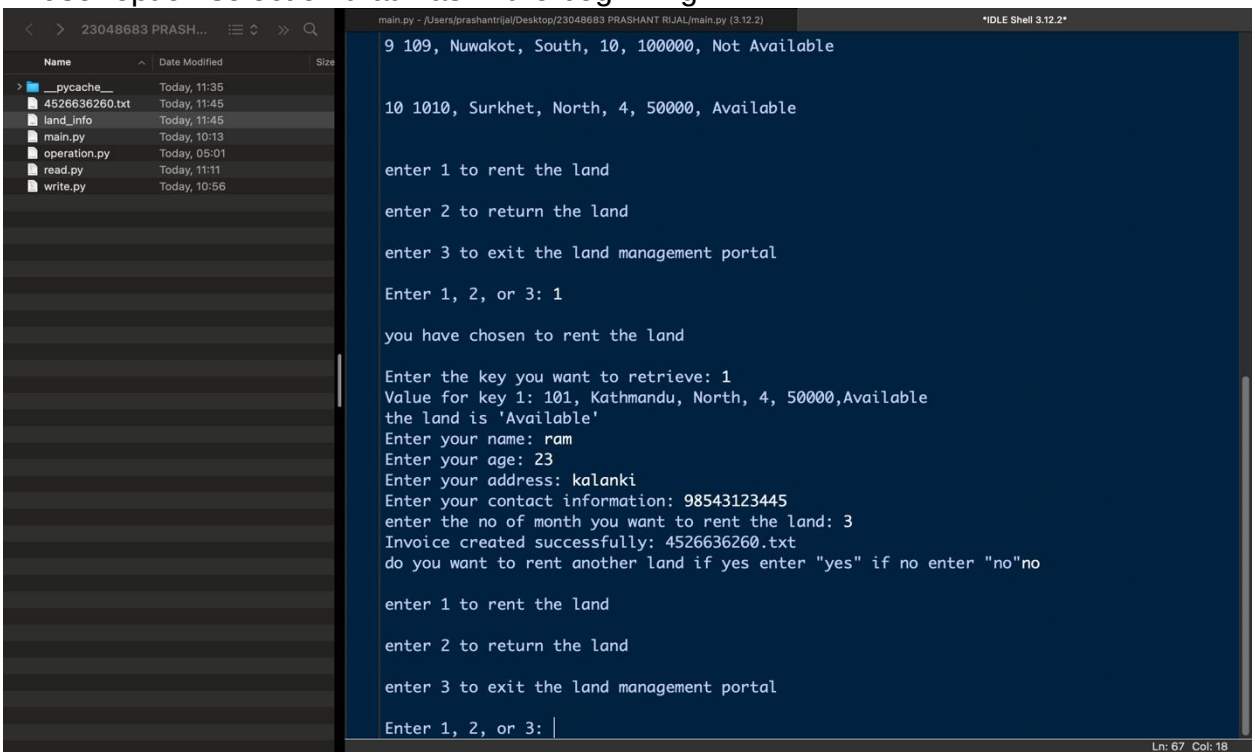


Figure 11 return to first input

Then lets return the land we just rented for that we input 2 as our value which in output shows that I have chosen to return land.

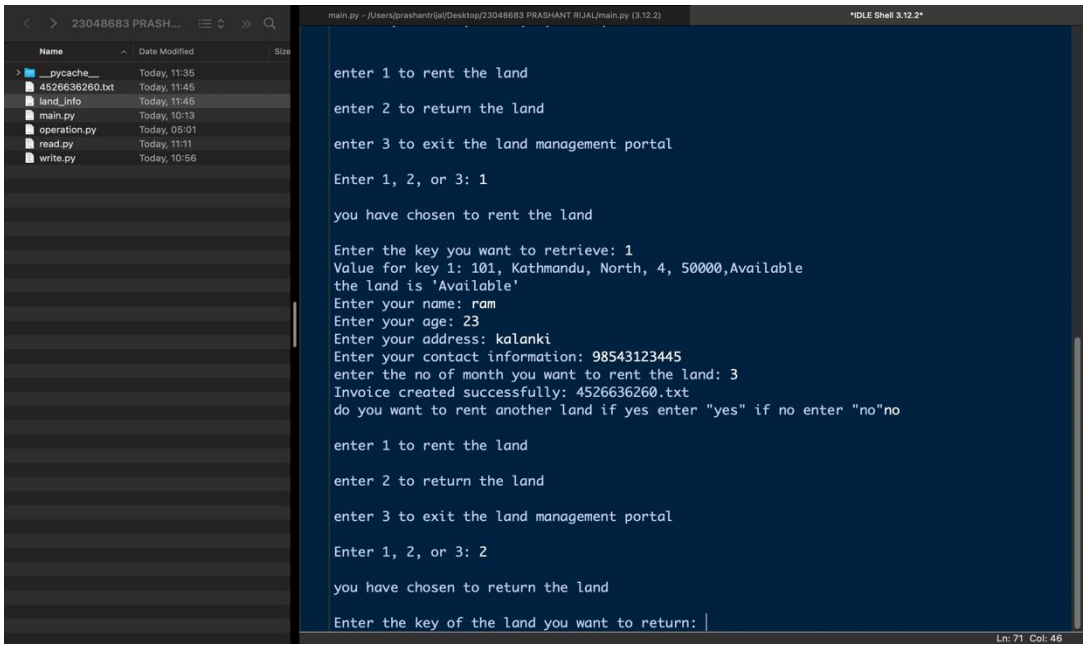


Figure 12 start of return

Now it is asking me to enter the key of land I want to borrow since I borrowed the land in key 1 I will return it. After doing so it will show the land I have selected and also check if it is available or not.

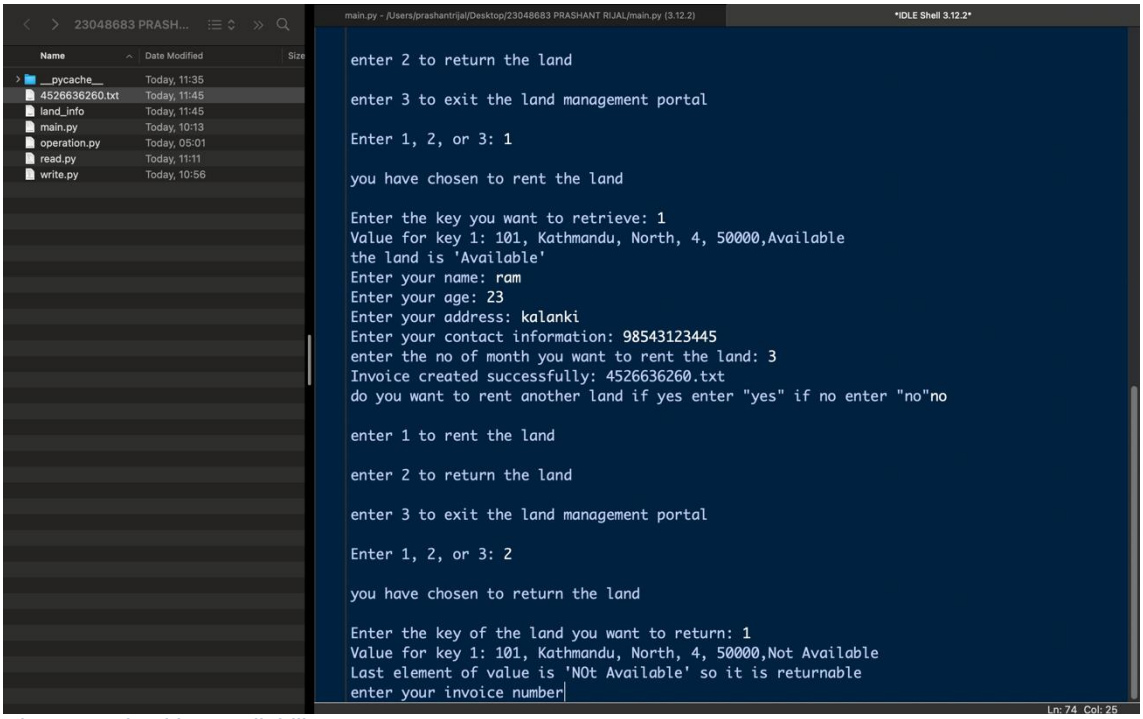


Figure 13 checking availability

Now it's asking me to enter the invoice no that we generated before after entering that number if it is correct it will ask me to fill in my info.

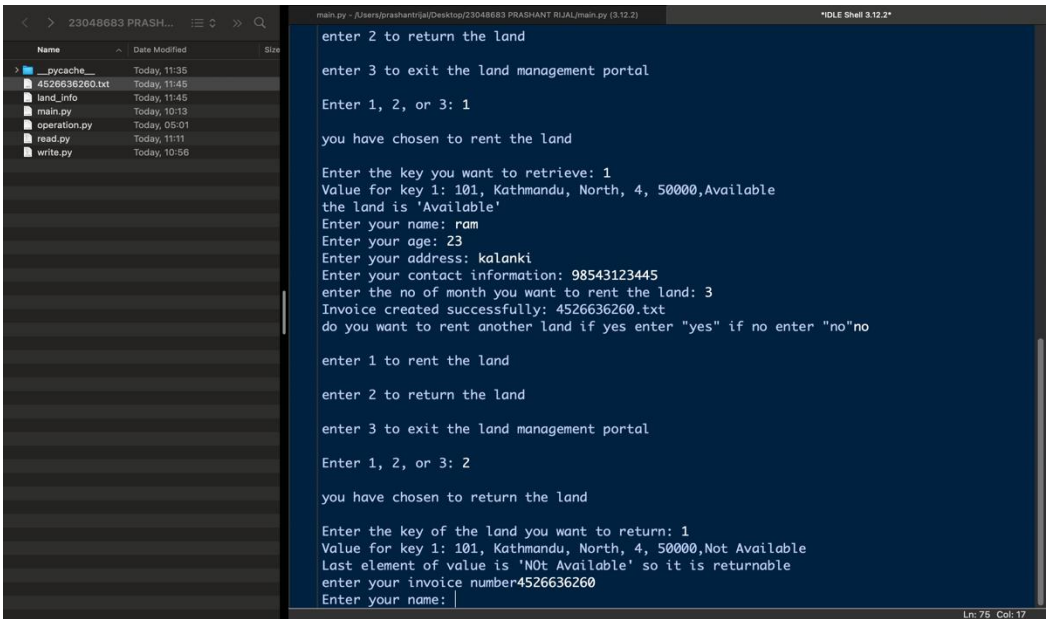


Figure 14 invoice filling

After filling all the information and since I have not overused by renting month I will fill in 0 and so the invoice is created.

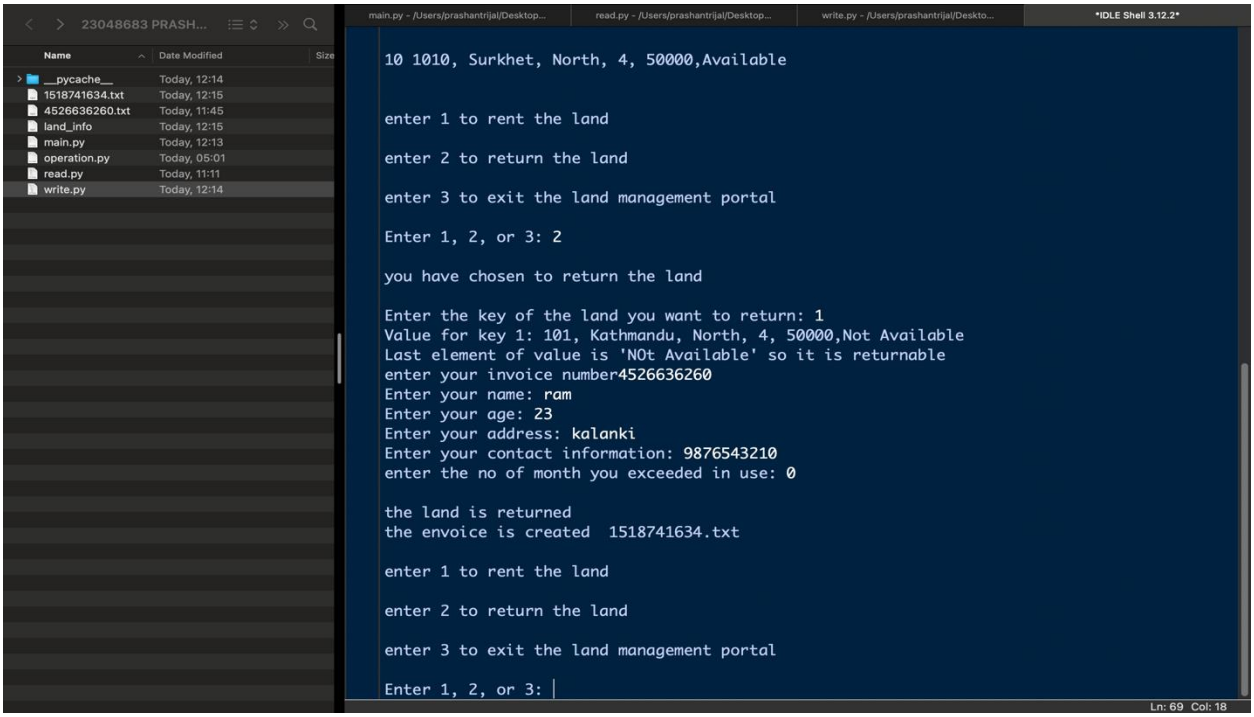


Figure 16 invoice creation

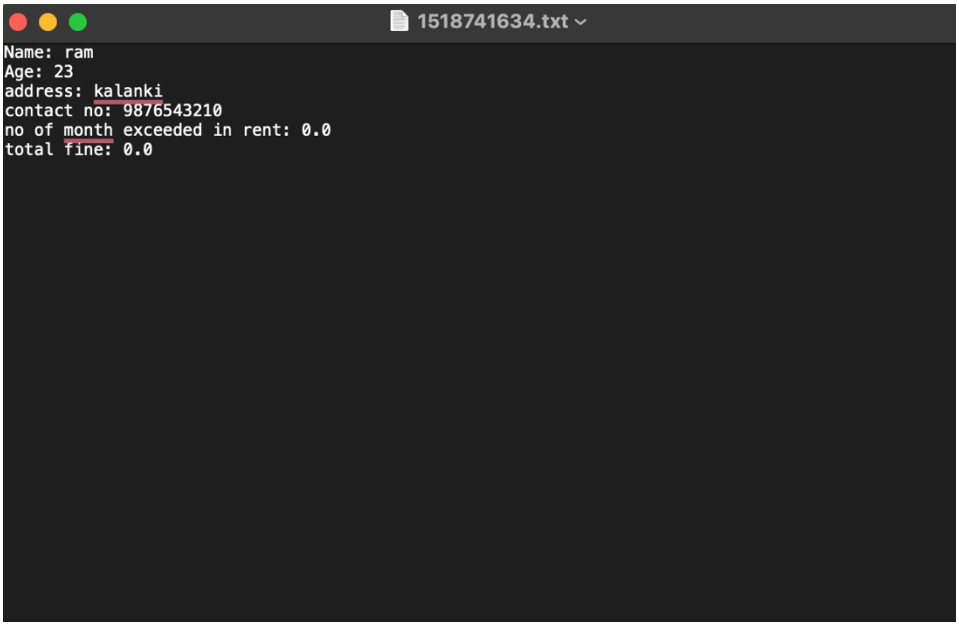


Figure 15 invoice

The program again goes to the first input and when we press 3 we exit the program.

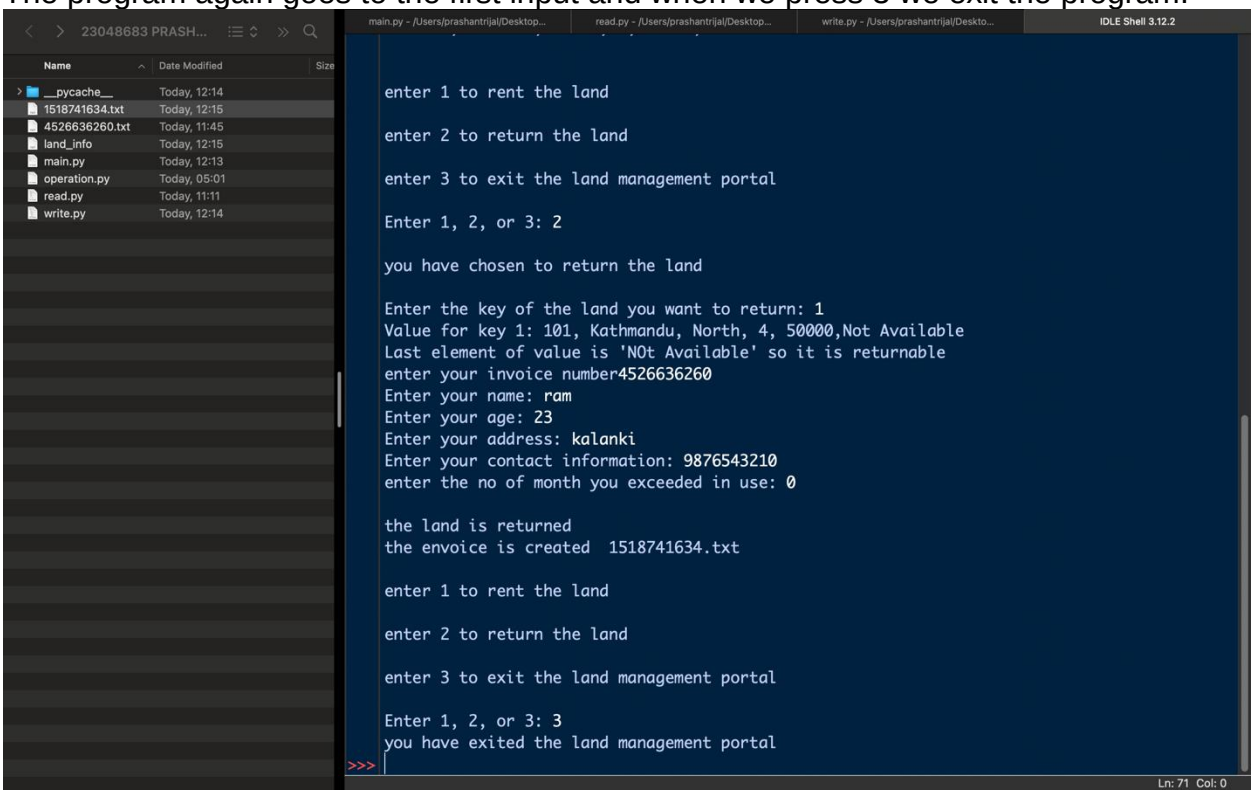


Figure 17Exiting portal

Testing

7.1 Test 1

Table 1 test 1

TEST NO.	1.
Objective	Try and except.
Action	The program was ran and invalid input was given where try and except was used
Expected Result	Error message
Actual Result	Error message was displayed
Conclusion	The test was successful.

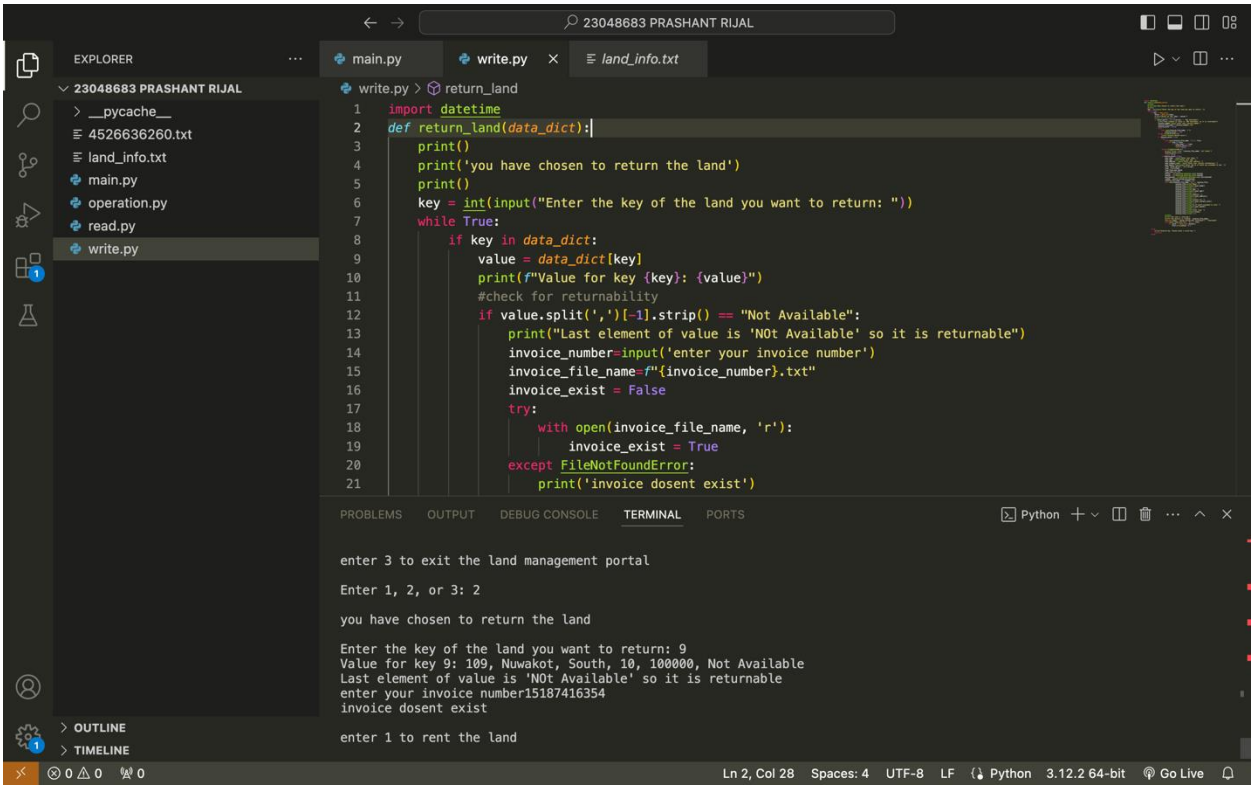


Figure 18 test 1

7.2 Test 2

TEST NO.	1.
Objective	Rent land entry retry
Action	The program was ran and non-existent input was given to check
Expected Result	Error message
Actual Result	Error message was displayed
Conclusion	The test was successful.

Table 2 test 2

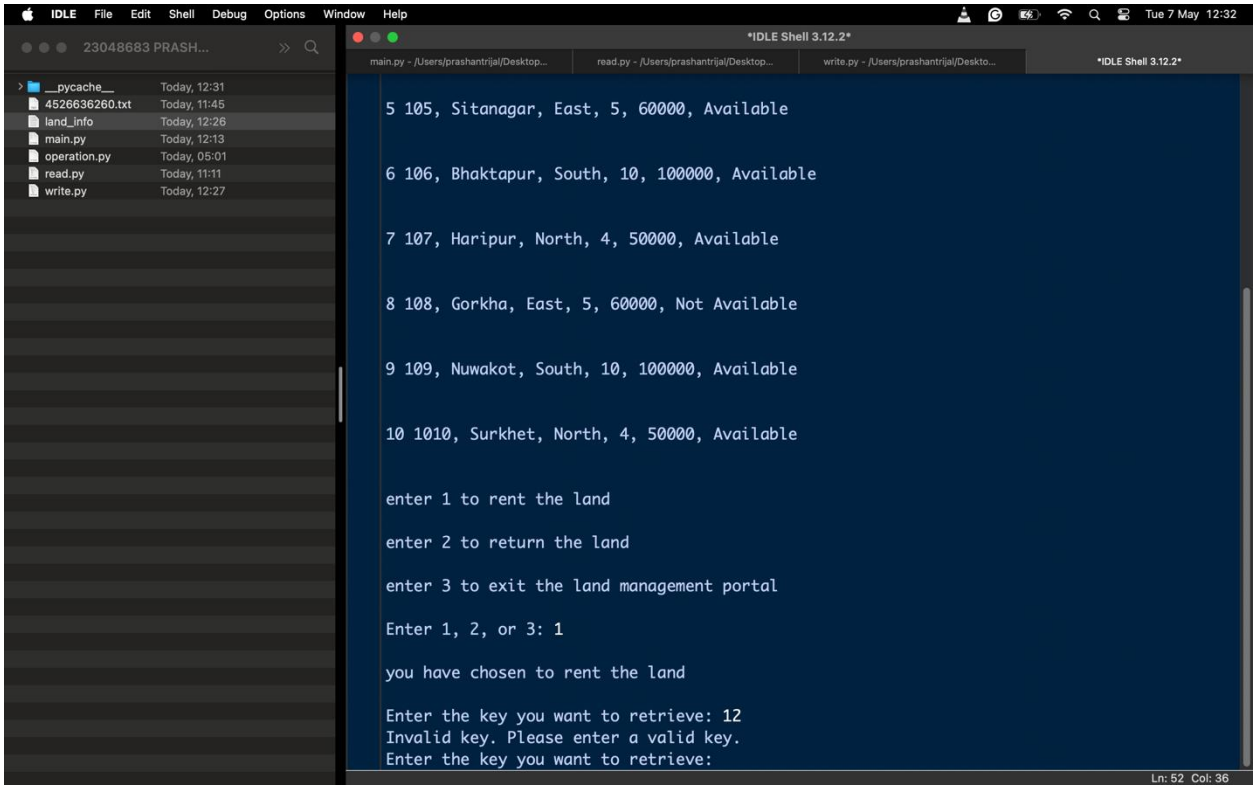


Figure 19 test 2

7.3 Test 3

TEST NO.	1.
Objective	Renting process
Action	The program was ran Option for rent was chosen Key 1 was inputted User info was filled
Expected Result	Invoice generation and change in land_info.txt file
Actual Result	Invoice was generated and land_info was changed.
Conclusion	The test was successful.

Table 3 test 3

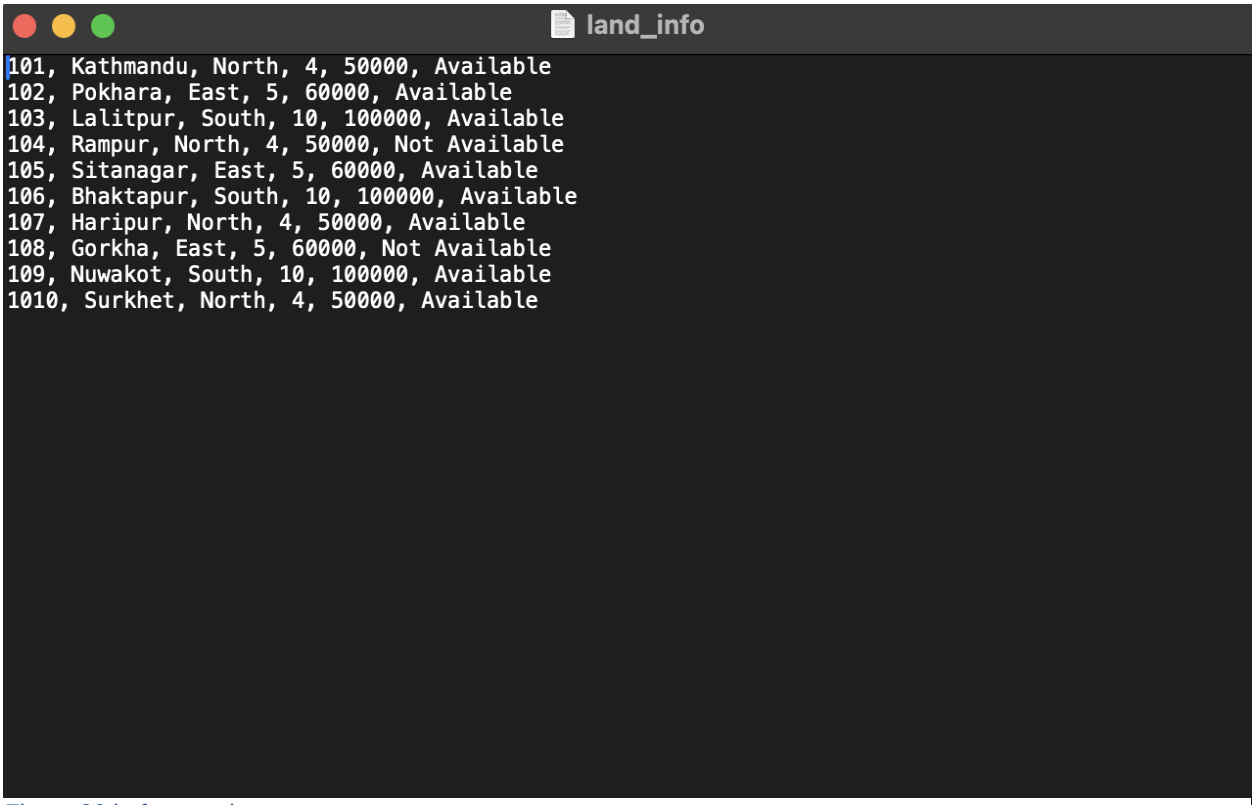


Figure 20 before renting

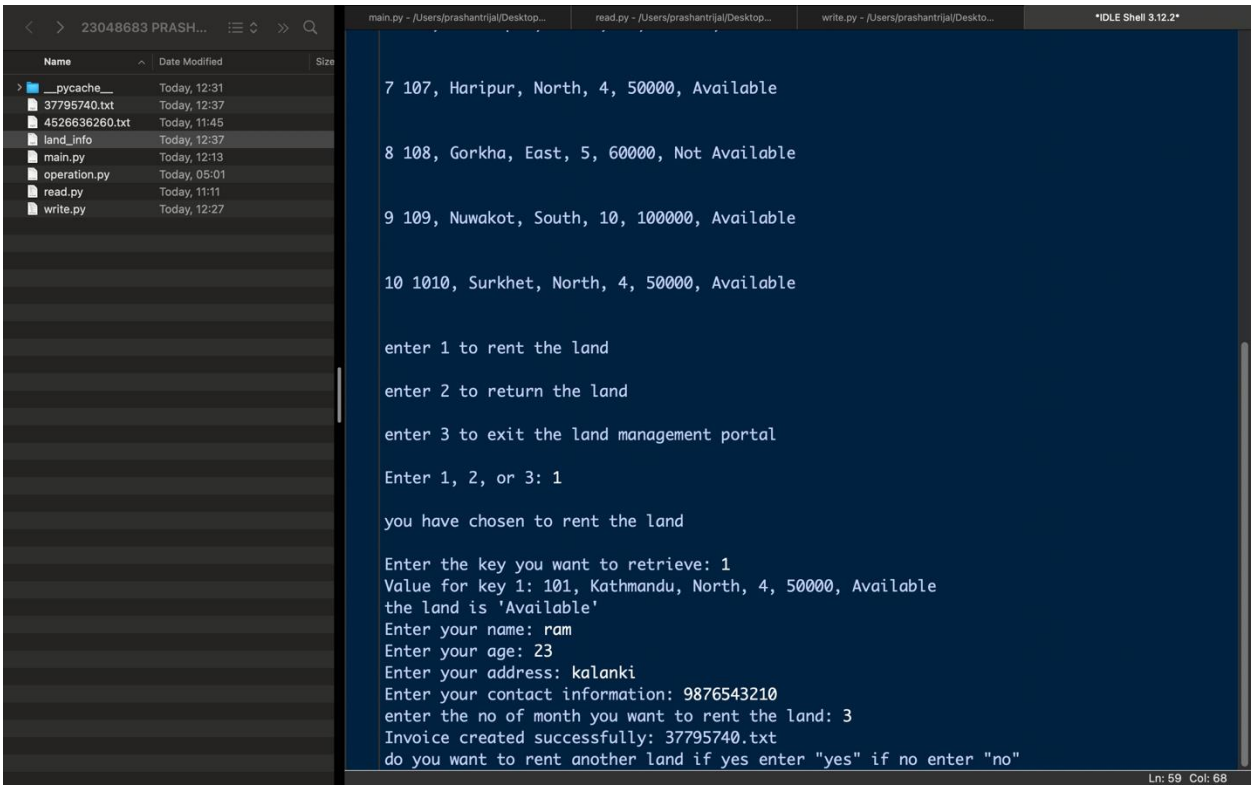


Figure 22 invoice creation

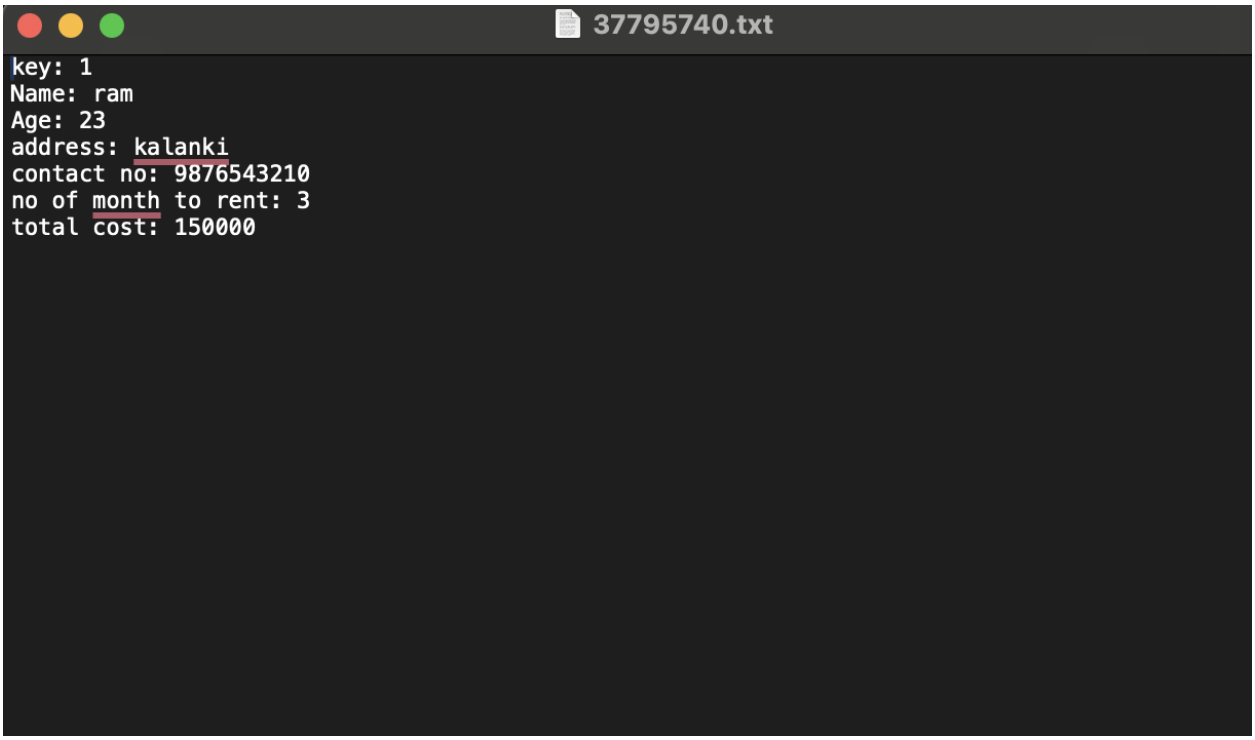


Figure 21 created invoice

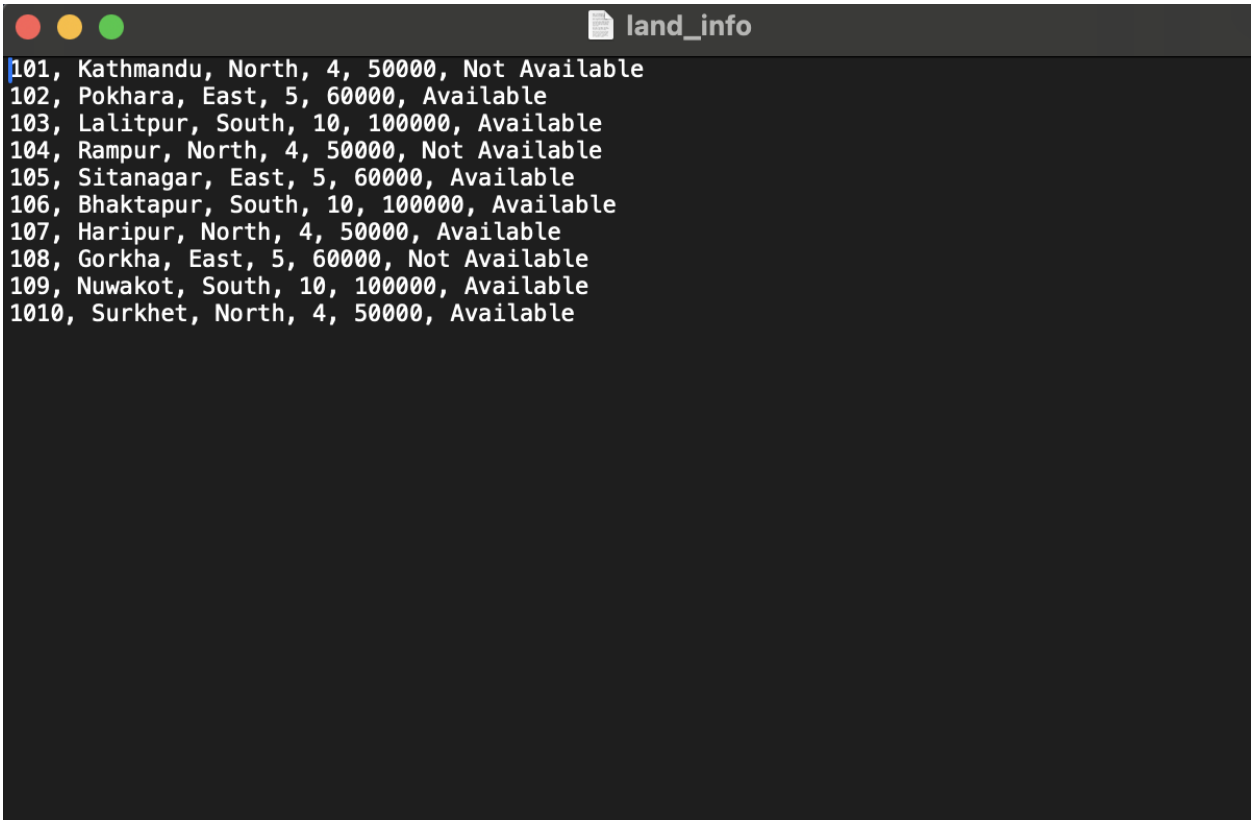


Figure 23 after renting

7.4 Test 4

TEST NO.	1.
Objective	Returing land
Action	The program was ran Option for return was chosen Key 1 was inputted Invoice no was filled User info was filled
Expected Result	Error message
Actual Result	Error message was displayed
Conclusion	The test was successful.

Table 4 test 4

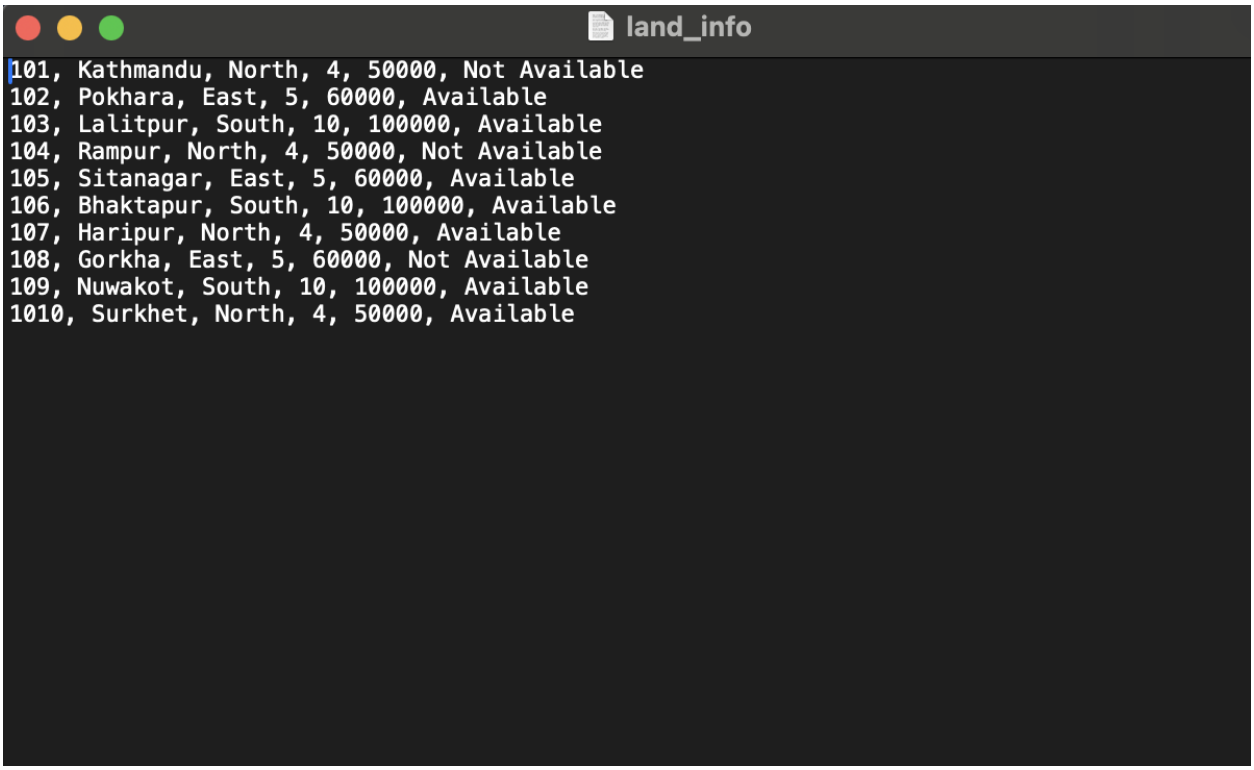


Figure 24 before return

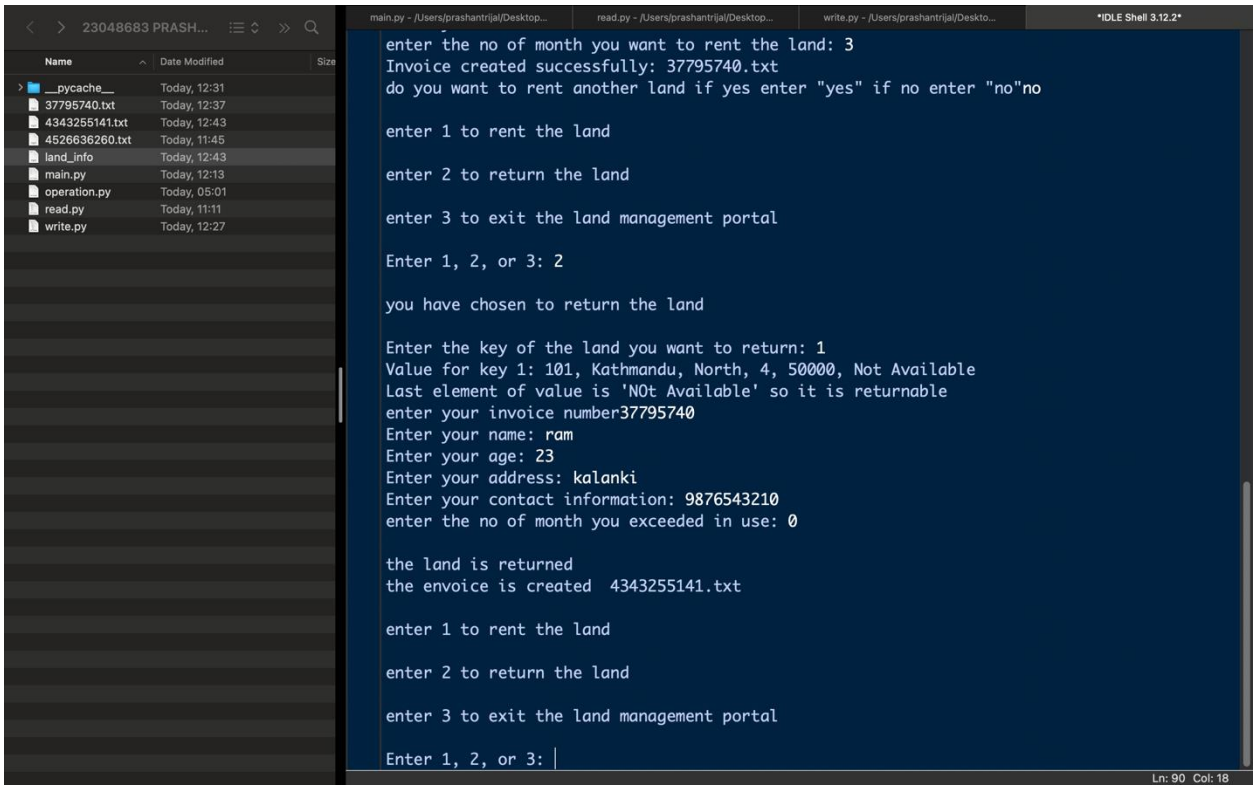


Figure 26 invoice creation

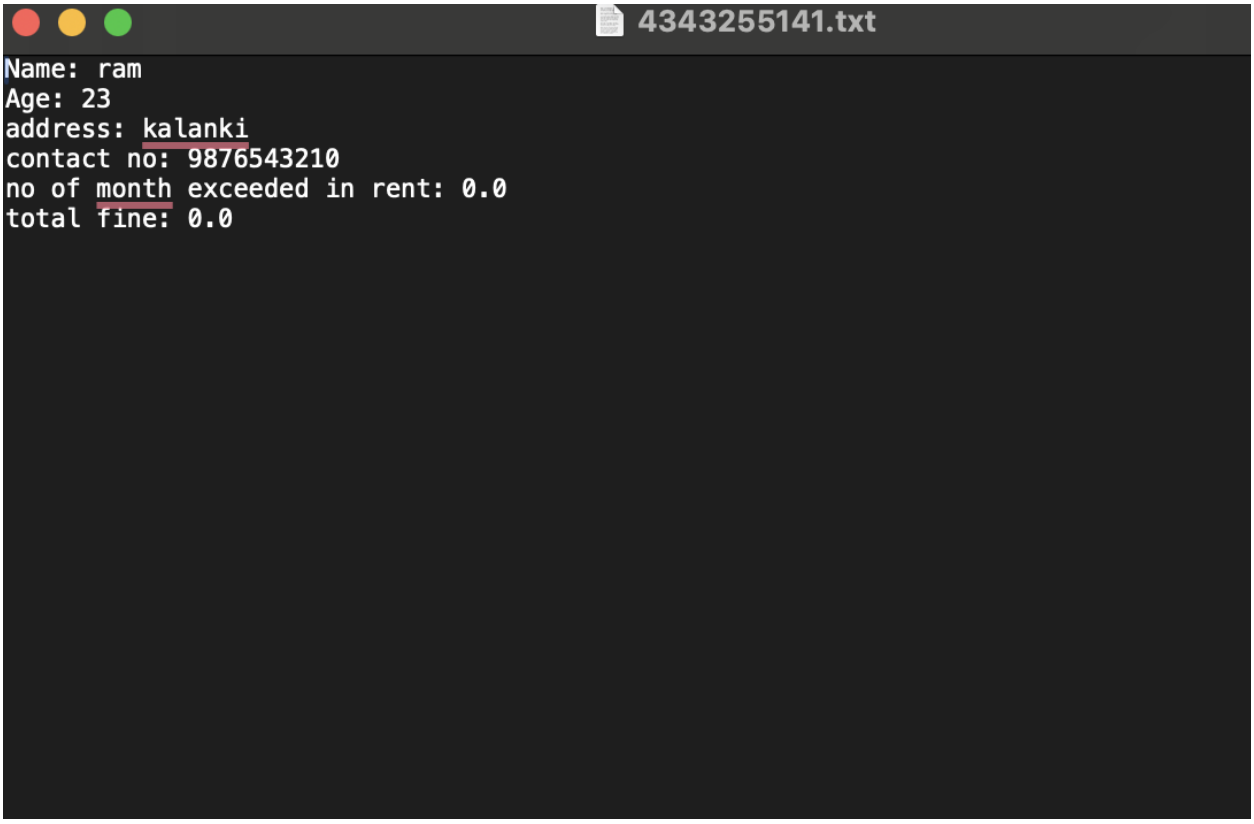


Figure 25 invoice

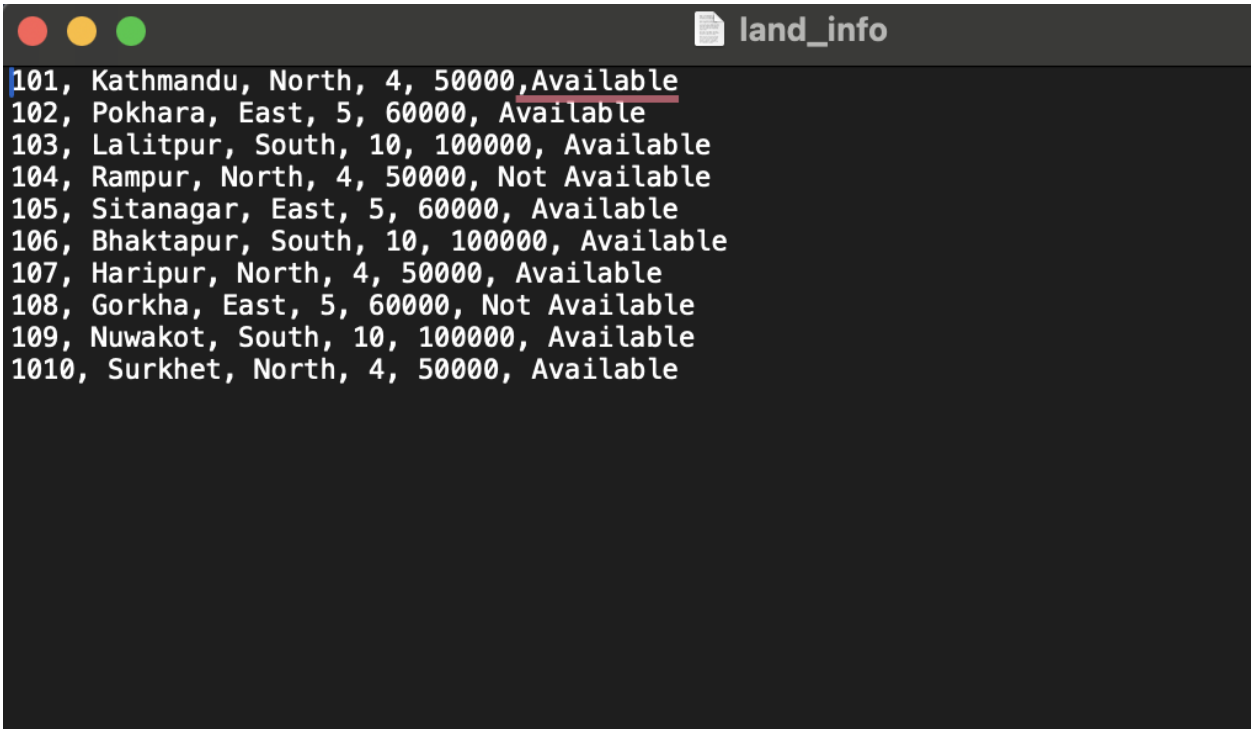


Figure 27 after renting

7.5 Test 5

TEST NO.	1.
Objective	Checking change in aviability
Action	The program was ran Option for return was chosen Key 1 was inputted Invoice no was filled User info was filled
Expected Result	Not avialiable changed to avialiable
Actual Result	It was changed
Conclusion	The test was successful.

Table 5 test 5

Figure 28 before return

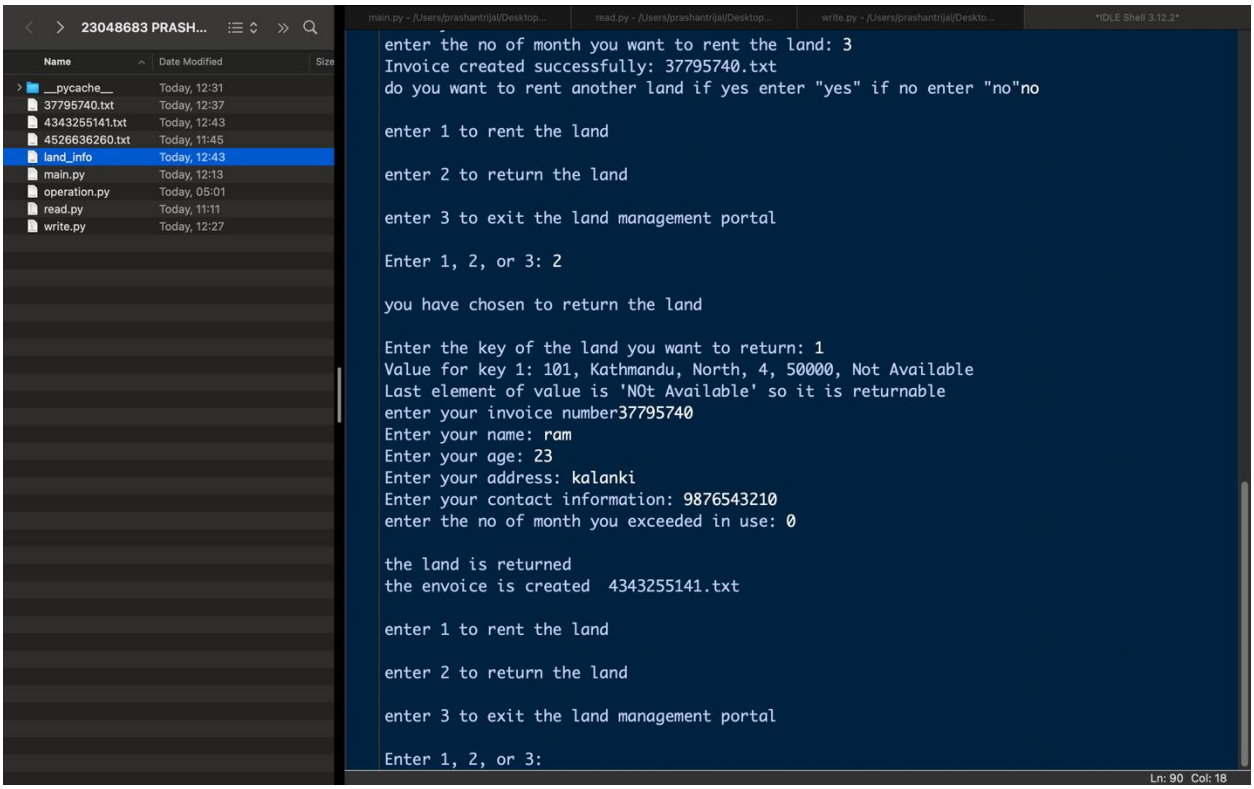


Figure 29 returning the land

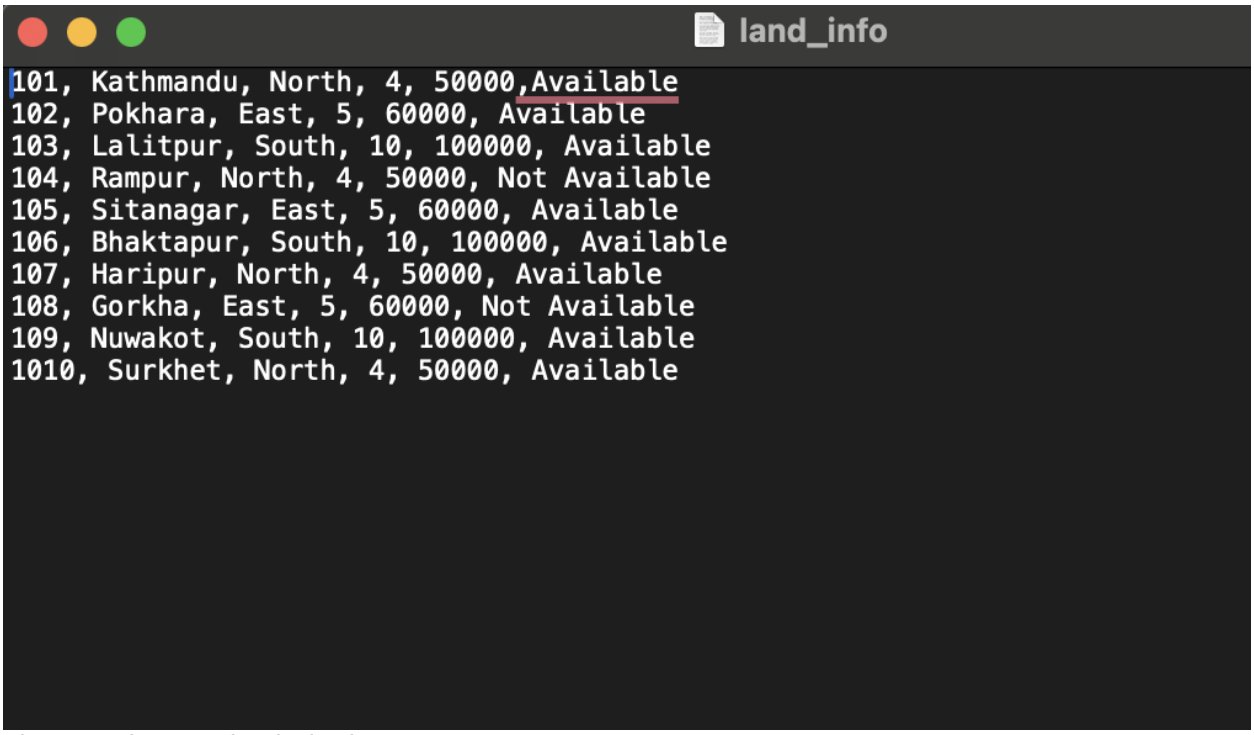


Figure 30 after returning the land

Conclusion

All in all the project was a great experience and a very lovely one al it taught us how the Python programming language can be used in actual cases rather than hypothetical cases

This experience has made me realize that programming is not just coding blindly but also making it such that the people who use it can be comfortable and confident while using the platform we create for them.

Appendix

Main.py

```
import operation
import read
import write

print()
print('                    welcome to the land rental management portal')
print()
file_path = 'land_info.txt'
data_dictionary = operation.read_file_into_dictionary(file_path)
operation.print_data(data_dictionary)
cond=True
while cond==True:
    print()
    print('enter 1 to rent the land')
    print()
    print('enter 2 to return the land')
    print()
    print('enter 3 to exit the land management portal')
    print()
    user_input = input("Enter 1, 2, or 3: ")
    if user_input not in ['1', '2', '3']:
        print("Error: Invalid input. Please enter 1, 2, or 3.")
        continue
    if user_input == '1':
        condi=True
        print()
        print('you have chosen to rent the land')
        print()
        while condi==True:
            read.rent_land(data_dictionary)
            # You can rent another land
            condition=input('do you want to rent another land if yes enter "yes" if no enter "no"')
            condition.lower()
            if condition=='yes':
                read.rent_land(data_dictionary)
                condi=True
                break
            else:
                condi=False
        cond=True
    elif user_input=='2':
        write.return_land(data_dictionary)
        cond=True
```

```
    elif user_input=='3':
        print("you have exited the land management portal")
        cond=False
```

Operation.py

```
def read_file_into_dictionary(file_path):   # function that converts the lines from txt in dictoinary
    data_dict = {}
    with open(file_path, 'r') as file:
        line_number = 1
        for line in file:
            data_dict[line_number] = line.strip()
            line_number += 1
    return data_dict

def print_data(data_dict):
    print("key,Kittal no,Location,direction,anna,price,status")
    for key, value in data_dict.items():
        print()
        print(key, value)#prints the data stored in the txt file with a key
        print()
```

read.py

```
import datetime
import operation
def rent_land(data_dict):
    while True:
        key = int(input("Enter the key you want to retrieve: "))
        if key in data_dict:
            value = data_dict[key]
            print(f"Value for key {key}: {value}")

            if value.split(',')[-1].strip() == "Available":
                print("the land is 'Available'")
                Key=str(key)
                user_name = input("Enter your name: ")
                user_age = input("Enter your age: ")
                user_address = input("Enter your address: ")
                user_contact_info = input("Enter your contact information: ")
                rent_month=int(input('enter the no of month you want to rent the land: '))
                cost = value.split(',')[4].strip()
                cost=int(cost)*rent_month
```

```
            minute = str(datetime.datetime.now().minute)
            second = str(datetime.datetime.now().second)
            microsecond = str(datetime.datetime.now().microsecond)
            random = minute+second+microsecond
            # Create invoice of land

            invoice_file_name = f"{random}.txt"
            with open(invoice_file_name, 'w') as invoice_file:
                invoice_file.write("key: ")
                invoice_file.write(Key)
                invoice_file.write('\n')
                invoice_file.write("Name: ")
                invoice_file.write(str(user_name))
                invoice_file.write('\n')
                invoice_file.write("Age: ")
                invoice_file.write(str(user_age))
                invoice_file.write('\n')
                invoice_file.write("address: ")
                invoice_file.write(str(user_address))
                invoice_file.write('\n')
                invoice_file.write("contact no: ")
                invoice_file.write(str(user_contact_info))
                invoice_file.write('\n')
                invoice_file.write("no of month to rent: ")
                invoice_file.write(str(rent_month))
                invoice_file.write('\n')
                invoice_file.write("total cost: ")
                invoice_file.write(str(cost))
                invoice_file.write('\n')

            print(f"Invoice created successfully: {invoice_file_name}")

            data_dict[key] = value.rstrip("Available") + "Not Available"
            with open('land_info.txt', 'w') as file:
             for value in data_dict.values():
                file.write(value+'\n')
        else:
            print("the land is not 'Available'. Please choose another key.")
            continue  # Ask for key again if land is not available
    else:
        print("Invalid key. Please enter a valid key.")
        continue  # Ask for key again if key is invalid
    break  # Break the loop if a valid key with available land is entered
```

write.py

```
import datetime
```

```
def return_land(data_dict):
    print()
    print('you have chosen to return the land')
    print()
    key = int(input("Enter the key of the land you want to return: "))
    while True:
        if key in data_dict:
            value = data_dict[key]
            print(f"Value for key {key}: {value}")
            #check for returnability
            if value.split(',')[-1].strip() == "Not Available":
                print("Last element of value is 'NOt Available' so it is returnable")
                invoice_number=input('enter your invoice number')
                invoice_file_name=f"{invoice_number}.txt"
                invoice_exist = False
                try:
                    with open(invoice_file_name, 'r'):
                        invoice_exist = True
                except FileNotFoundError:
                    print('invoice dosent exist')
                if invoice_exist==True:
                    try:
                        with open(invoice_file_name, 'r') as file:
                            for line in file:
                                if f"key={key}" in line:
                                    invoice_exist=True
                                    return False
                    except FileNotFoundError:
                        print(f"Error: File '{invoice_file_name}' not found.")
                        invoice_exist=False
                        return False
                    if invoice_exist==True:
                        user_name = input("Enter your name: ")
                        user_age = input("Enter your age: ")
                        user_address = input("Enter your address: ")
                        user_contact_info = input("Enter your contact information: ")
                        user_month=float(input('enter the no of month you exceeded in use: '))
                        fine= value.split(',')[4].strip()
                        fine= float(fine)
                        fine=fine*user_month
                        fine=fine*(10/100)
                        minute = str(datetime.datetime.now().minute)
                        second = str(datetime.datetime.now().second)
                        microsecond = str(datetime.datetime.now().microsecond)
                        random = minute+second+microsecond
                        invoice_file_name = f"{random}.txt"
                        with open(invoice_file_name, 'w') as invoice_file:
                                invoice_file.write("Name: ")
```

```
                    invoice_file.write(str(user_name))
                    invoice_file.write('\n')
                    invoice_file.write("Age: ")
                    invoice_file.write(str(user_age))
                    invoice_file.write('\n')
                    invoice_file.write("address: ")
                    invoice_file.write(str(user_address))
                    invoice_file.write('\n')
                    invoice_file.write("contact no: ")
                    invoice_file.write(str(user_contact_info))
                    invoice_file.write('\n')
                    invoice_file.write("no of month exceeded in rent: ")
                    invoice_file.write(str(user_month))
                    invoice_file.write('\n')
                    invoice_file.write("total fine: ")
                    invoice_file.write(str(fine))
                    invoice_file.write('\n')
            print()
            print('the land is returned')
            print('the envoice is created ',invoice_file_name)
            data_dict[key] = value.rstrip("Not Available") + "Available"
            with open('land_info.txt', 'w') as file:
                for value in data_dict.values():
                    file.write(value+'\n')

else:
    print("Invalid key. Please enter a valid key.")
    continue
break
```