

REPORT FOR GUESSING GAME

As a project work for Course

PYTHON PROGRAMMING (INT 213)

Name : K Sai Kalyan
Registration Number : 12004587
Name : N Varun Kumar
Registration Number : 12001673
Program : B.Tech CSE
Semester : Third
School : School of Computer
Science and Engineering
Name of the University : Lovely Professional
University
Date of submission : 15th NOVEMBER 2021

Lovely Professional University
Jalandhar, Punjab, India.



LOVELY
PROFESSIONAL
UNIVERSITY

Transforming Education Transforming India

GUESSING GAME

15th NOVEMBER 2021

ABSTRACT:-

This is a guessing game designed by using python, in this game you give two numbers of upper bound and lower bound respectively then the system will generate a random number and based on the difference of the lower and upper bounds you will be given a certain amount of chances and within the chances if you guessed the number correctly then it will print congratulations you have guessed the number correctly and if you failed to guess the number within the given amount of chances it will print the correct answer automatically.

ACKNOWLEDGEMENT:-

I would like to thank my mentor –Prof. Md. Imran Hussain for his advice and inputs on this project. Many thanks to my friends and seniors as well, who spent countless hours to listen and provide feedbacks.

Table of Contents

1. ABSTRACT	2
2. INTRODUCTION 2.1 CONTEXT 2.2 MOTIVATION 2.3 IDEA	4
3. TEAM MEMBERS WITH ROLES 3.1 TEAM LEADER 3.2 MEMBERS 3.3 CONTRIBUTIONS	5
4. LIBRARIES 4.1 DIFFERENT TYPES 4.2 WHY THEY ARE USED	6
5.Screenshots	7
6.Refrences	10

INTRODUCTION:-

1.1 Context

This project has been done as part of my course for the CSE at Lovely Professional University. Supervised by Md. Imran Hussian, I had a month time to fulfill the requirements in order to succeed the module.

1.2 Motivations

Being extremely interested in everything having a relation with the Python the group project was a great occasion to give us the time to learn our interest for this field. The fact that we can make predictions and give the ability for machines to generate a number is both efficient and limitless in term of application possibilities.

1.3 Idea:-

As a first experience, we wanted to make our project as much didactic as possible by approaching every different steps of the python programming process and trying to understand them deeply. Known as “guessing game” which is useful to illustrate and practice, we chose to take number guessing game as approach. The goal was to predict the random number generated by the computer and taking into account different “features” that will be developed in the following .

TEAM MEMBERS:-

TEAM LEADER:-

Varun Kumar:-

Contributions:-

1. Coding(joined)
2. Report

Sai Kalyan:-

Contributions:-

1. Coding(joined)
2. Report(joined)

LIBRARIES:-

Random:-

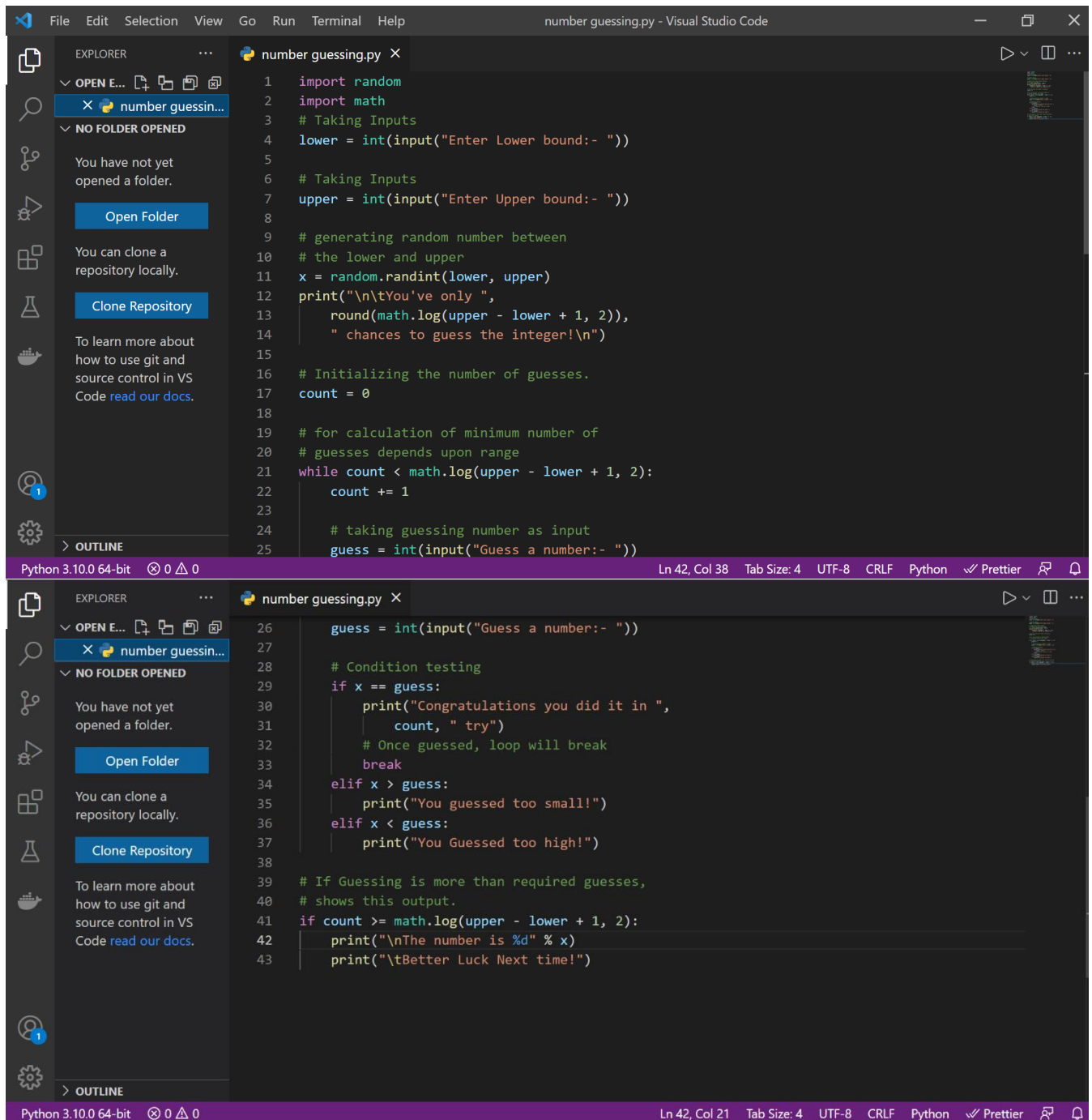
Python Random module is an in-built module of Python which is **used to generate random numbers**. These are pseudo-random numbers means these are not truly random. This module can be used to perform random actions such as generating random numbers, print random a value for a list or string, etc.

Math:-

The Python Math Library provides us access to some common math functions and constants in Python, which we can use throughout our code for more complex mathematical computations. The library is a built-in Python module, therefore you don't have to do any installation to use it.

SCREENSHOTS:-

1. Code:-



The image displays two screenshots of the Visual Studio Code editor interface, showing the code for a number guessing game in Python. The editor is titled "number guessing.py - Visual Studio Code".

Top Screenshot: Shows the initial code structure. The Explorer panel on the left indicates "NO FOLDER OPENED". The code in the editor is as follows:

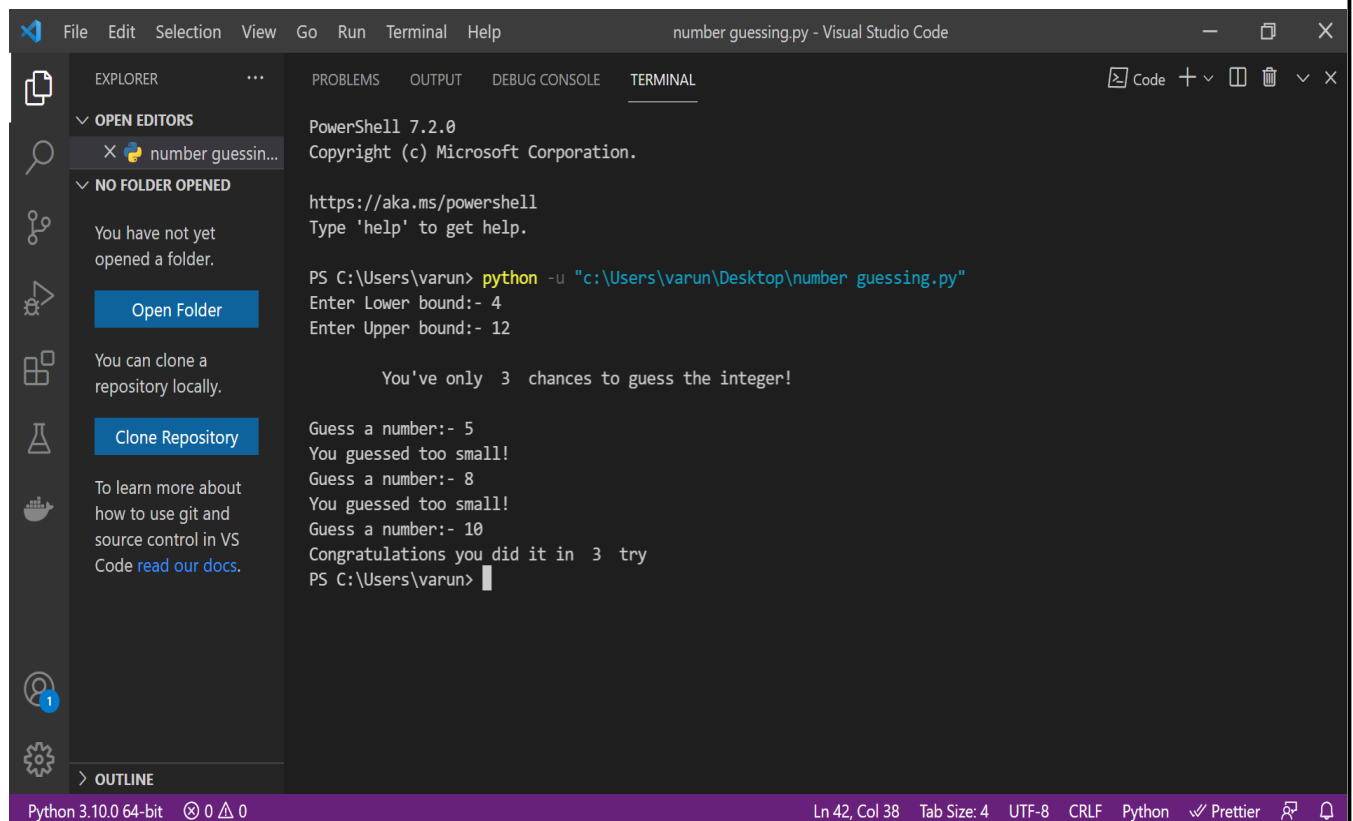
```
1 import random
2 import math
3 # Taking Inputs
4 lower = int(input("Enter Lower bound:- "))
5
6 # Taking Inputs
7 upper = int(input("Enter Upper bound:- "))
8
9 # generating random number between
10 # the lower and upper
11 x = random.randint(lower, upper)
12 print("\n\tYou've only ",
13       round(math.log(upper - lower + 1, 2)),
14       " chances to guess the integer!\n")
15
16 # Initializing the number of guesses.
17 count = 0
18
19 # for calculation of minimum number of
20 # guesses depends upon range
21 while count < math.log(upper - lower + 1, 2):
22     count += 1
23
24 # taking guessing number as input
25 guess = int(input("Guess a number:- "))
```

Bottom Screenshot: Shows the continuation of the code. The Explorer panel remains the same. The code in the editor is as follows:

```
26 guess = int(input("Guess a number:- "))
27
28 # Condition testing
29 if x == guess:
30     print("Congratulations you did it in ",
31           count, " try")
32     # Once guessed, loop will break
33     break
34 elif x > guess:
35     print("You guessed too small!")
36 elif x < guess:
37     print("You Guessed too high!")
38
39 # If Guessing is more than required guesses,
40 # shows this output.
41 if count >= math.log(upper - lower + 1, 2):
42     print("\nThe number is %d" % x)
43     print("\tBetter Luck Next time!")
```

2.Examples:-

1.



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal is running a PowerShell script for a number guessing game. The user has entered a lower bound of 4 and an upper bound of 12. The game allows 3 chances to guess the integer. The user has guessed 5, 8, and 10, all of which are too small. The game congratulates the user for guessing the number in 3 tries.

```
PowerShell 7.2.0
Copyright (c) Microsoft Corporation.

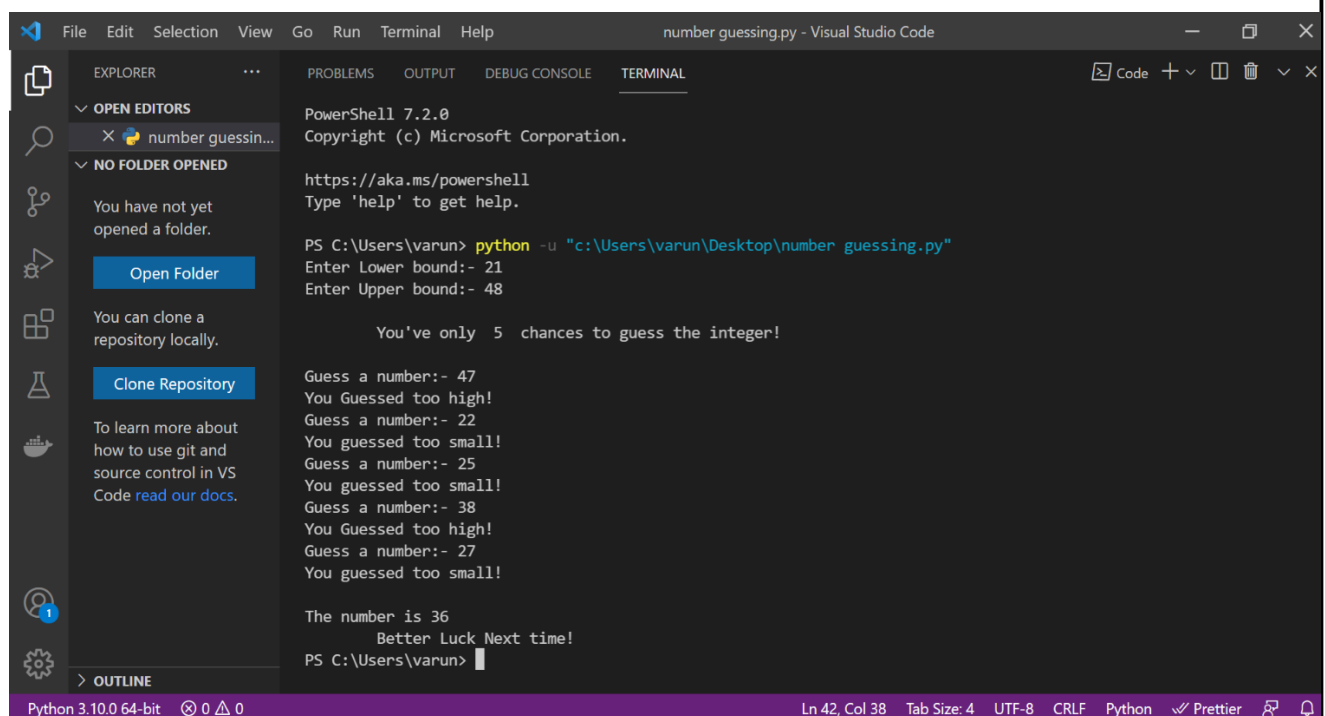
https://aka.ms/powershell
Type 'help' to get help.

PS C:\Users\varun> python -u "c:\Users\varun\Desktop\number_guessing.py"
Enter Lower bound:- 4
Enter Upper bound:- 12

        You've only 3 chances to guess the integer!

Guess a number:- 5
You guessed too small!
Guess a number:- 8
You guessed too small!
Guess a number:- 10
Congratulations you did it in 3 try
PS C:\Users\varun>
```

2.



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal is running a PowerShell script for a number guessing game. The user has entered a lower bound of 21 and an upper bound of 48. The game allows 5 chances to guess the integer. The user has guessed 47, 22, 25, 38, and 27, all of which are incorrect. The game reveals the number is 36 and wishes the user better luck next time.

```
PowerShell 7.2.0
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

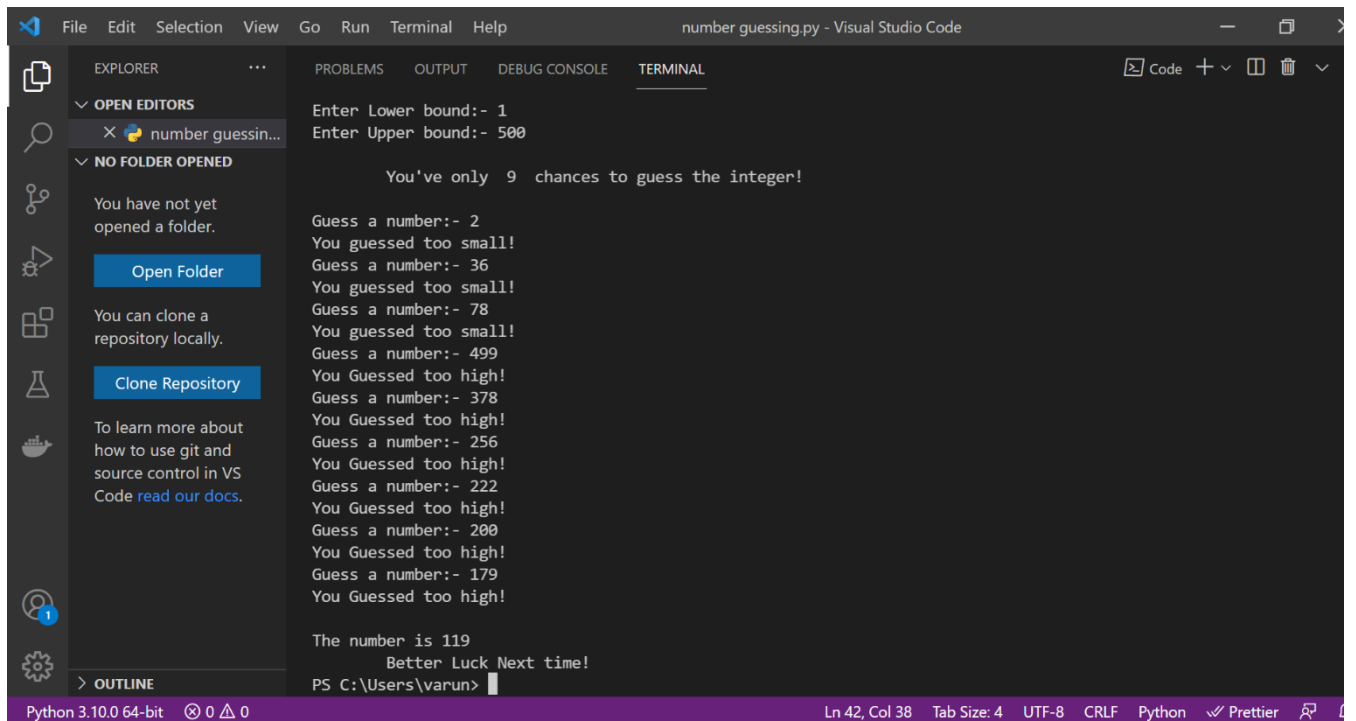
PS C:\Users\varun> python -u "c:\Users\varun\Desktop\number_guessing.py"
Enter Lower bound:- 21
Enter Upper bound:- 48

        You've only 5 chances to guess the integer!

Guess a number:- 47
You Gessed too high!
Guess a number:- 22
You guessed too small!
Guess a number:- 25
You guessed too small!
Guess a number:- 38
You Gessed too high!
Guess a number:- 27
You guessed too small!

The number is 36
Better Luck Next time!
PS C:\Users\varun>
```


3.



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the execution of a Python script named 'number_guessing.py'. The user has set a lower bound of 1 and an upper bound of 500. The program allows 9 chances to guess the integer. The user makes several guesses, some too small and some too high. The program eventually reveals the number is 119 and wishes the user better luck next time.

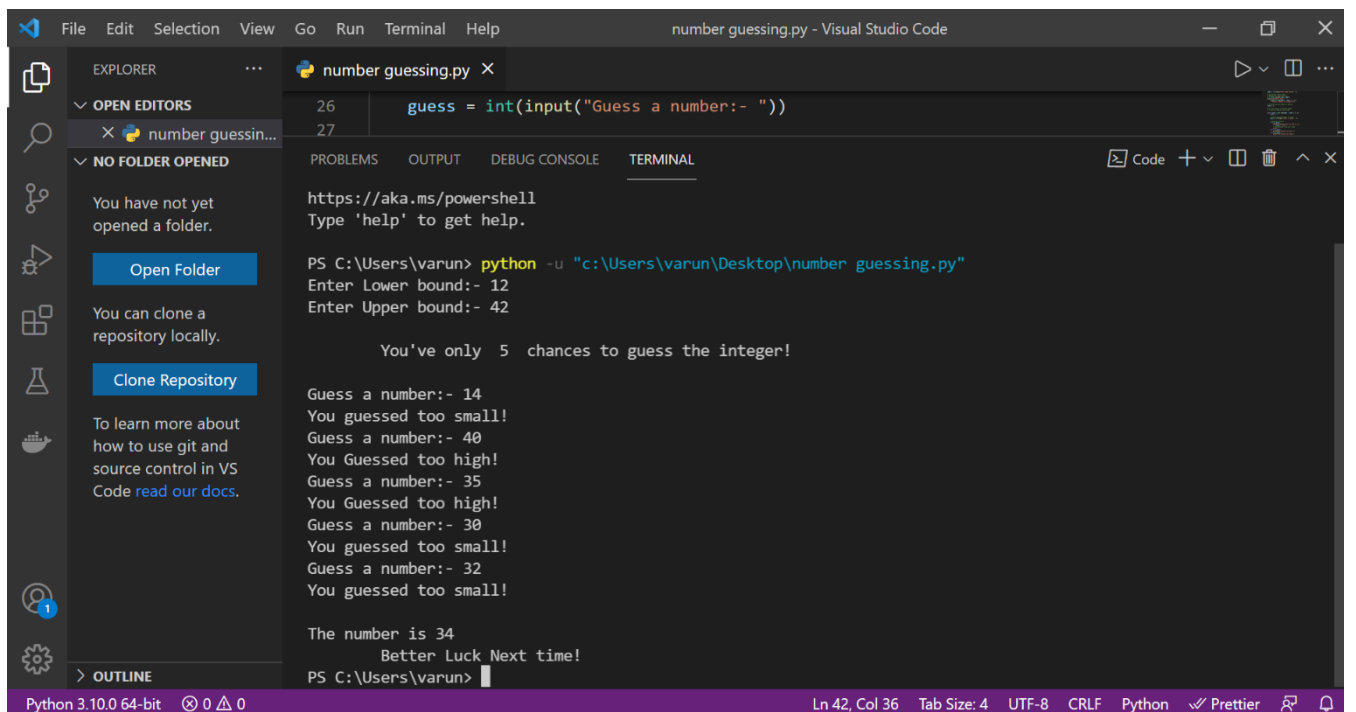
```
Enter Lower bound:- 1
Enter Upper bound:- 500

You've only 9 chances to guess the integer!

Guess a number:- 2
You guessed too small!
Guess a number:- 36
You guessed too small!
Guess a number:- 78
You guessed too small!
Guess a number:- 499
You Gessed too high!
Guess a number:- 378
You Gessed too high!
Guess a number:- 256
You Gessed too high!
Guess a number:- 222
You Gessed too high!
Guess a number:- 200
You Gessed too high!
Guess a number:- 179
You Gessed too high!

The number is 119
Better Luck Next time!
PS C:\Users\varun>
```

4.



The screenshot shows the Visual Studio Code interface with the source code of 'number_guessing.py' open in the editor. The code uses the 'random' module to generate a random number and implements a guessing game with 5 chances. The terminal window shows the execution of the script, where the user sets a lower bound of 12 and an upper bound of 42. The program allows 5 chances to guess the integer. The user makes several guesses, some too small and some too high. The program eventually reveals the number is 34 and wishes the user better luck next time.

```
26 guess = int(input("Guess a number:- "))
27
https://aka.ms/powershell
Type 'help' to get help.

PS C:\Users\varun> python -u "c:\Users\varun\Desktop\number_guessing.py"
Enter Lower bound:- 12
Enter Upper bound:- 42

You've only 5 chances to guess the integer!

Guess a number:- 14
You guessed too small!
Guess a number:- 40
You Gessed too high!
Guess a number:- 35
You Gessed too high!
Guess a number:- 30
You guessed too small!
Guess a number:- 32
You guessed too small!

The number is 34
Better Luck Next time!
PS C:\Users\varun>
```

Conclusions:-

It is our team's hope that this document will be of huge help with understanding of our little project as we have used a different approach which has proved beneficial for us and easy for us to understand the vast ocean that is Python programming.

REFERENCES:-

To conduct this project the following tools have been used :

- Visual Studio, Google Collab (For testing the code)
- <https://docs.python.org/3/library/math.html>
- <https://docs.python.org/3/library/random.html>