www.kiet.edu
Delhi-NCR, Ghaziabad

KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

PROJECT TITLE= AI-BASED NUMBER GUESSING GAME

NAME= ANGELINA

ROLL NUMBER= 202401100300041

BRANCH= CSE-AI

SECTION= A

INTRODUCTION

The AI-Based Number Guessing Game is an interactive program where the computer attempts to guess a number that the user is thinking of, using an optimized approach called the *Binary Search Algorithm*. Instead of randomly guessing numbers, the AI intelligently narrows down the range based on user feedback. This approach ensures that the game finds the correct number in the shortest possible time.

METHODOLOGY

1. Define the Game Rules

- Decide whether the AI will guess the user's number or vice versa.
- Set the range of numbers (e.g., 1 to 100).
- Define the number of attempts allowed (optional).
- Specify how feedback will be provided (e.g., "too high," "too low," or "correct").

2. Choose the AI Approach

- **If the AI is guessing the user's number:**
  - Use a **binary search algorithm** for efficient guessing.
  - The AI starts by guessing the midpoint of the range and adjusts based on user feedback ("too high" or "too low").
  - This approach ensures the AI guesses the number in the fewest attempts (logarithmic time complexity).
- **If the user is guessing the AI's number:**
  - The AI randomly selects a number within the defined range.
  - The user makes guesses, and the AI provides feedback ("too high," "too low," or "correct").
  - Optionally, the AI can track the user's guesses and provide hints or adaptive difficulty.

3. Implement the Game Logic

- **For AI guessing the user's number:**
  - Initialize the range (e.g., `low = 1`, `high = 100`).
  - Use a loop to repeatedly:
    1. Calculate the midpoint: `guess = (low + high) // 2`.
    2. Ask the user for feedback (e.g., "Is your number higher, lower, or equal to [guess]?").
    3. Adjust the range based on feedback:
       - If "too high," set `high = guess - 1`.
       - If "too low," set `low = guess + 1`.
    4. Repeat until the correct number is guessed.
- **For user guessing the AI's number:**
  - Randomly generate a target number within the range.
  - Use a loop to repeatedly:
    1. Prompt the user to enter a guess.
    2. Compare the guess to the target number.
    3. Provide feedback ("too high," "too low," or "correct").
    4. Repeat until the user guesses correctly or runs out of attempts.

4. Add Enhancements

- **Difficulty Levels:**
  - Adjust the range of numbers or the number of attempts based on difficulty (e.g., easy: 1-50, hard: 1-1000).
- **Adaptive AI:**
  - If the AI is guessing, it can learn from repeated games to optimize its strategy.
- **User Interface:**
  - Create a simple text-based or graphical interface for user interaction.
- **Error Handling:**
  - Validate user input to ensure it is within the range and is a valid number.
- **Score Tracking:**
  - Keep track of the number of attempts or time taken to guess the number.

5. Test and Debug

- Test the game with different scenarios to ensure it works as expected.

- Debug any issues with the logic or user interaction.

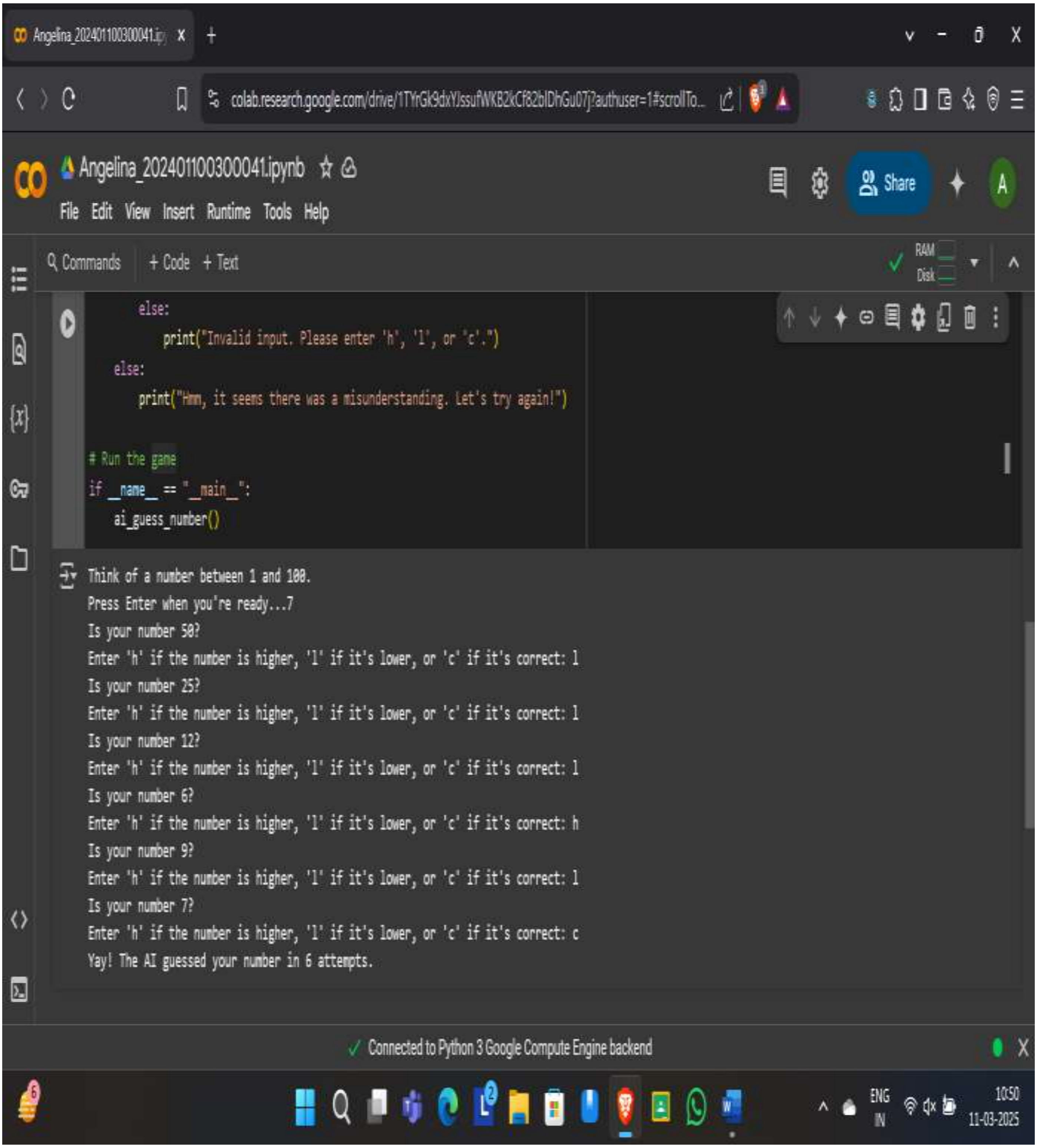
6. Deploy the Game

- Package the game as an executable, web app, or mobile app.
- Optionally, add features like leaderboards, multiplayer modes, or AI vs. AI competitions.

CODE

```
import random


def ai_guess_number():
    """
    This function implements an AI-based number guessing game.
    The AI will try to guess the number you are thinking of within a specified range.
    """


    # Step 1: Define the range of numbers
    low = 1
    high = 100
    print(f"Think of a number between {low} and {high}.")
    input("Press Enter when you're ready...")

    # Step 2: Initialize the number of attempts
    attempts = 0

    # Step 3: AI starts guessing using binary search
    while low <= high:
        attempts += 1

        # AI makes a guess in the middle of the current range
        guess = (low + high) // 2
        print(f"Is your number {guess}?")

        # Step 4: Get user feedback
        feedback = input("Enter 'h' if the number is higher, 'l' if it's lower, or 'c' if it's correct: ").strip().lower()

        # Step 5: Adjust the range based on user feedback
        if feedback == 'h':
            low = guess + 1
        elif feedback == 'l':
            high = guess - 1
        elif feedback == 'c':
            print(f"Yay! The AI guessed your number in {attempts} attempts.")
            break
        else:
            print("Invalid input. Please enter 'h', 'l', or 'c'.")
    else:
        print("Hmm, it seems there was a misunderstanding. Let's try again!")

# Run the game
if __name__ == "__main__":
    ai_guess_number()
```

OUTPUT


Angelina_202401100300041.ipy
colab.research.google.com/drive/1TYrGk9dxYJssufWKB2kCf82bIDhGu07j?authuser=1#scrollTo...
Angelina_202401100300041.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands
+ Code + Text
RAM
Disk
Connected to Python 3 Google Compute Engine backend
ENG
IN
10:50
11-03-2025

```
        else:
            print("Invalid input. Please enter 'h', 'l', or 'c'.")
    else:
        print("Hmm, it seems there was a misunderstanding. Let's try again!")

# Run the game
if __name__ == "__main__":
    ai_guess_number()
```

```
Think of a number between 1 and 100.
Press Enter when you're ready...7
Is your number 50?
Enter 'h' if the number is higher, 'l' if it's lower, or 'c' if it's correct: l
Is your number 25?
Enter 'h' if the number is higher, 'l' if it's lower, or 'c' if it's correct: l
Is your number 12?
Enter 'h' if the number is higher, 'l' if it's lower, or 'c' if it's correct: l
Is your number 6?
Enter 'h' if the number is higher, 'l' if it's lower, or 'c' if it's correct: h
Is your number 9?
Enter 'h' if the number is higher, 'l' if it's lower, or 'c' if it's correct: l
Is your number 7?
Enter 'h' if the number is higher, 'l' if it's lower, or 'c' if it's correct: c
Yay! The AI guessed your number in 6 attempts.
```