**AI-Based Number Guessing Game Report**

**Title Page**

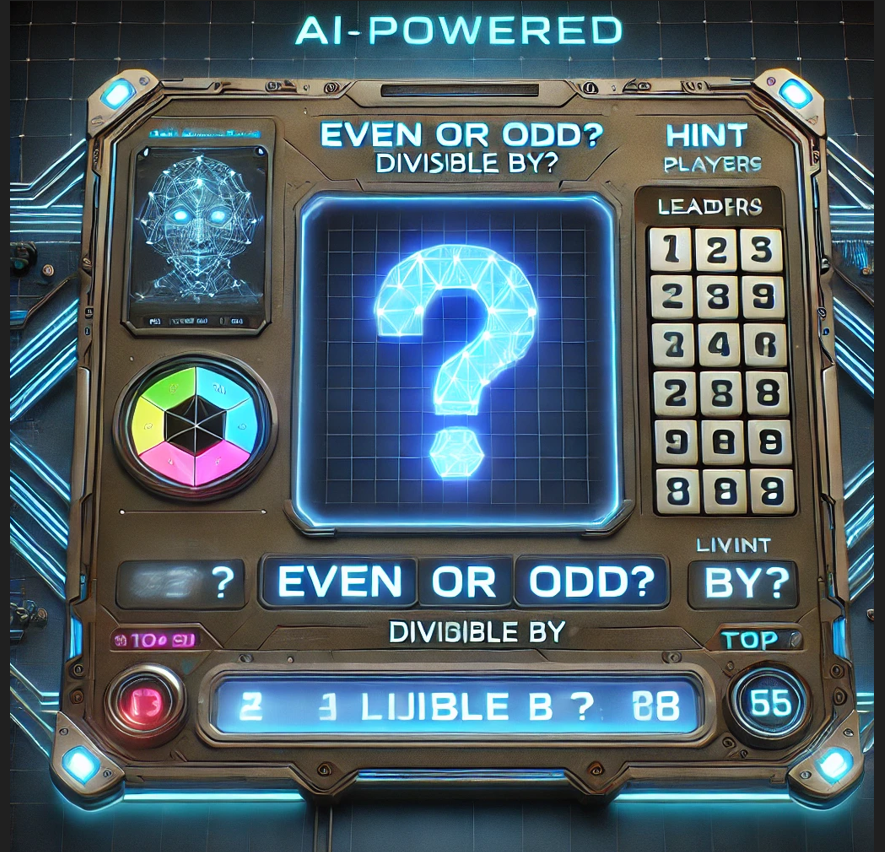
**Problem Statement: Develop an AI-Based Number Guessing Game that allows users to select a difficulty level, receive hints, and track their scores on a leaderboard.**

**Personal Details:**  
**Name:** [Sunny Kumar]  
**Roll No.:** [202401100400195]

**Introduction**

The AI-Based Number Guessing Game is designed to challenge users to guess a randomly generated number within a limited number of attempts. The game provides hints at specific intervals to assist players in making educated guesses. Additionally, the program features a leaderboard to track high scores.

This game aims to make number guessing more engaging by incorporating an AI-like hint system that dynamically provides clues to guide players toward the correct answer.



**Methodology**

The approach used to develop the AI-Based Number Guessing Game includes the following steps:

1. **Difficulty Selection**: Players can select from three difficulty levels that determine the range of the secret number.
2. **Number Generation**: The game generates a random number within the chosen range using Python's random module.
3. **Guessing Mechanism**: The player is given a maximum of 20 attempts to guess the number correctly.
4. **Hint System**:
   * After 5 attempts: Parity hint (Even/Odd)
   * After 10 attempts: Divisibility hint
   * After 15 attempts: Range hint (Upper or Lower half)
5. **Scoring & Leaderboard**: Scores are recorded based on the number of attempts taken, and a leaderboard ranks players based on their best scores.

**Code**

import random

# Dictionary to store player scores

leaderboard = {}

def get\_divisor\_hint(secret\_number):

    """Returns a small divisor of the secret number, if any, else None."""

    for i in range(2, secret\_number // 2 + 1):

        if secret\_number % i == 0:

            return i

    return None

def choose\_difficulty():

    """Allows the player to choose a difficulty level."""

    print("Choose a difficulty level:")

    print("1. Easy (1-50)")

    print("2. Medium (1-100)")

    print("3. Hard (1-200)")

    while True:

        choice = input("Enter 1, 2, or 3: ")

        if choice == '1':

            return 1, 50

        elif choice == '2':

            return 1, 100

        elif choice == '3':

            return 1, 200

        else:

            print("Invalid choice. Please select 1, 2, or 3.")

def play\_game(player\_name):

    """Main game logic where the player guesses the secret number."""

    low, high = choose\_difficulty()

    secret\_number = random.randint(low, high)

    max\_attempts = 20

    attempts = 0

    hints\_given = {"parity": False, "divisor": False, "range": False}

    print(f"\nI have selected a number between {low} and {high}.")

    print(f"You have {max\_attempts} attempts to guess it correctly!")

    while attempts < max\_attempts:

        try:

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

        except ValueError:

            print("Please enter a valid number.")

            continue

        attempts += 1

        if guess < secret\_number:

            print("Too low!")

        elif guess > secret\_number:

            print("Too high!")

        else:

            print(f"Congratulations {player\_name}! You guessed it in {attempts} attempts.")

            score = max\_attempts - attempts + 1

            print(f"Your score for this round is: {score}")

            leaderboard[player\_name] = max(leaderboard.get(player\_name, 0), score)

            break

        # Provide hints at specific attempts

        if attempts == 5 and not hints\_given["parity"]:

            print(f"Hint: The secret number is {'even' if secret\_number % 2 == 0 else 'odd' }.")

            hints\_given["parity"] = True

        elif attempts == 10 and not hints\_given["divisor"]:

            divisor = get\_divisor\_hint(secret\_number)

            print(f"Hint: The secret number is {'a prime number' if not divisor else f'divisible by {divisor}' }.")

            hints\_given["divisor"] = True

        elif attempts == 15 and not hints\_given["range"]:

            midpoint = (low + high) // 2

            print(f"Hint: The secret number is in the {'lower' if secret\_number <= midpoint else 'upper'} half of the range.")

            hints\_given["range"] = True

        print(f"Attempts remaining: {max\_attempts - attempts}")

    else:

        print(f"Sorry, you've used all {max\_attempts} attempts. The secret number was: {secret\_number}")

def display\_leaderboard():

    """Displays the leaderboard sorted by the highest score."""

    if not leaderboard:

        print("No scores yet. Play a game first!")

        return

    print("\n--- Leaderboard ---")

    sorted\_leaderboard = sorted(leaderboard.items(), key=lambda x: x[1], reverse=True)

    for rank, (player, score) in enumerate(sorted\_leaderboard, start=1):

        print(f"{rank}. {player}: {score}")

    print("-------------------\n")

def main():

    """Runs the game loop."""

    print("Welcome to the AI-Based Number Guessing Game!")

    player\_name = input("Enter your name: ")

    while True:

        play\_game(player\_name)

        display\_leaderboard()

        play\_again = input("Do you want to play again? (yes/no): ").strip().lower()

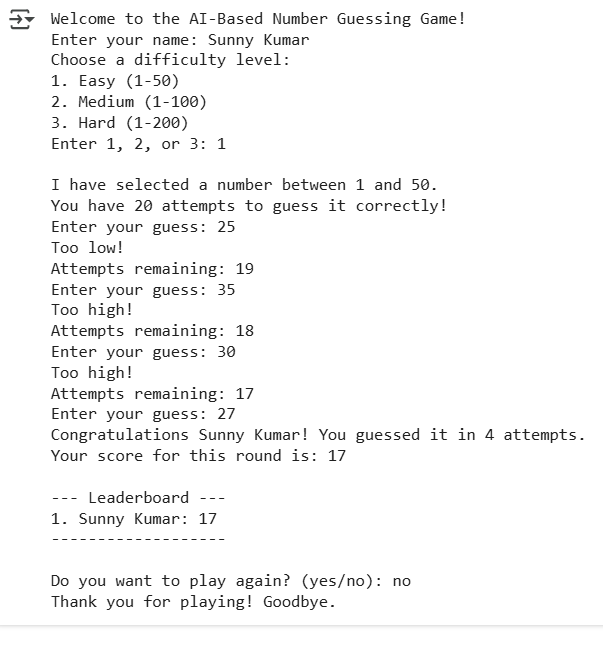
        if play\_again not in ("yes", "y"):

            print("Thank you for playing! Goodbye.")

            break

if \_\_name\_\_ == "\_\_main\_\_":

    main()



**References/Credits**

* Help from the Internet and Python resources.
* Python Random Library Documentation.