Use Case

AIM:

To design and implement a Python application that simulates the rolling of dice using NumPy, allowing the user to specify:

- Number of dice
- Number of sides per die
- Number of rolls

Algorithm:

1. **Start the program.**
2. **Import the NumPy library.**
3. **Ask the user** to input:
   - Number of dice
   - Number of sides on each die
   - Number of rolls
4. **Use NumPy's randint()** function to generate random values simulating die rolls:
   - Each die produces a random number between 1 and number of sides.
5. **Store the results** in a 2D NumPy array where each row represents one roll.
6. **Display each roll's values** and the total for that roll.
7. **Calculate and display statistics**:
   - Average of totals
   - Maximum and minimum total rolled
8. **End the program.**

Program:

```
    import numpy as np


    def roll_dice(num_dice=1, num_sides=6, num_rolls=1):
        """
        Simulates rolling dice using NumPy.
        Returns a 2D array of shape (num_rolls, num_dice).
        """
        return np.random.randint(1, num_sides + 1, size=(num_rolls, num_dice))


    def display_results(rolls):
        """
        Displays each roll and calculates statistics.
        """
        print("\n🎲 Dice Roll Results:")
        for i, roll in enumerate(rolls, start=1):
            print(f"Roll {i}: {roll} → Total: {np.sum(roll)}")
         totals = np.sum(rolls, axis=1)
        print("\n📊 Summary:")
        print(f"- Average Total: {np.mean(totals):.2f}")
        print(f"- Maximum Total: {np.max(totals)}")
        print(f"- Minimum Total: {np.min(totals)}")
if __name__ == "__main__":
        print("🎲 Welcome to the Dice Rolling Simulator 🎲")
        num_dice = int(input("Enter number of dice: "))
        num_sides = int(input("Enter number of sides per die: "))
        num_rolls = int(input("Enter number of rolls: "))
            rolls = roll_dice(num_dice, num_sides, num_rolls)
    display_results(rolls)
```

output:

```
Welcome to the Dice Rolling Simulator 🎲
Number of dice: 2
Number of sides per die: 6
Number of rolls: 5

🎲 Dice Roll Results:
Roll 1: [2 6] → Total: 8
Roll 2: [2 3] → Total: 5
Roll 3: [3 3] → Total: 6
Roll 4: [6 6] → Total: 12
Roll 5: [6 5] → Total: 11

📊 Summary:
- Average Total: 8.40
- Maximum Total: 12
- Minimum Total: 5
```

Result:

The dice rolling simulation was successfully implemented using NumPy. The user was able to input the number of dice, number of sides per die, and number of rolls. The program correctly simulated and displayed each roll and provided useful statistics (average, max, min total) of the rolls.