**The Doomed Dice challenge**

**Given:**

Consider two six-sided faces namely Die A and Die B with faces numbered from 1 to 6. Both the dice are rolled together.

**PART-A**

1. **How many total combinations are possible? Show the math along with the code?**

The total number of combinations when two dice are rolled together can be calculated using the formula:

TotalCombinations=(NumberOfFaces)^(NumberOfDice)

For six-sided dice, each die has 6 faces.

NumberOfFaces=6

NumberOfDice=2

If two dice are rolled together,then

TotalCombinations=6^2

=36

**Code:**

public ResponseEntity<Integer> getTotalCombinations() {

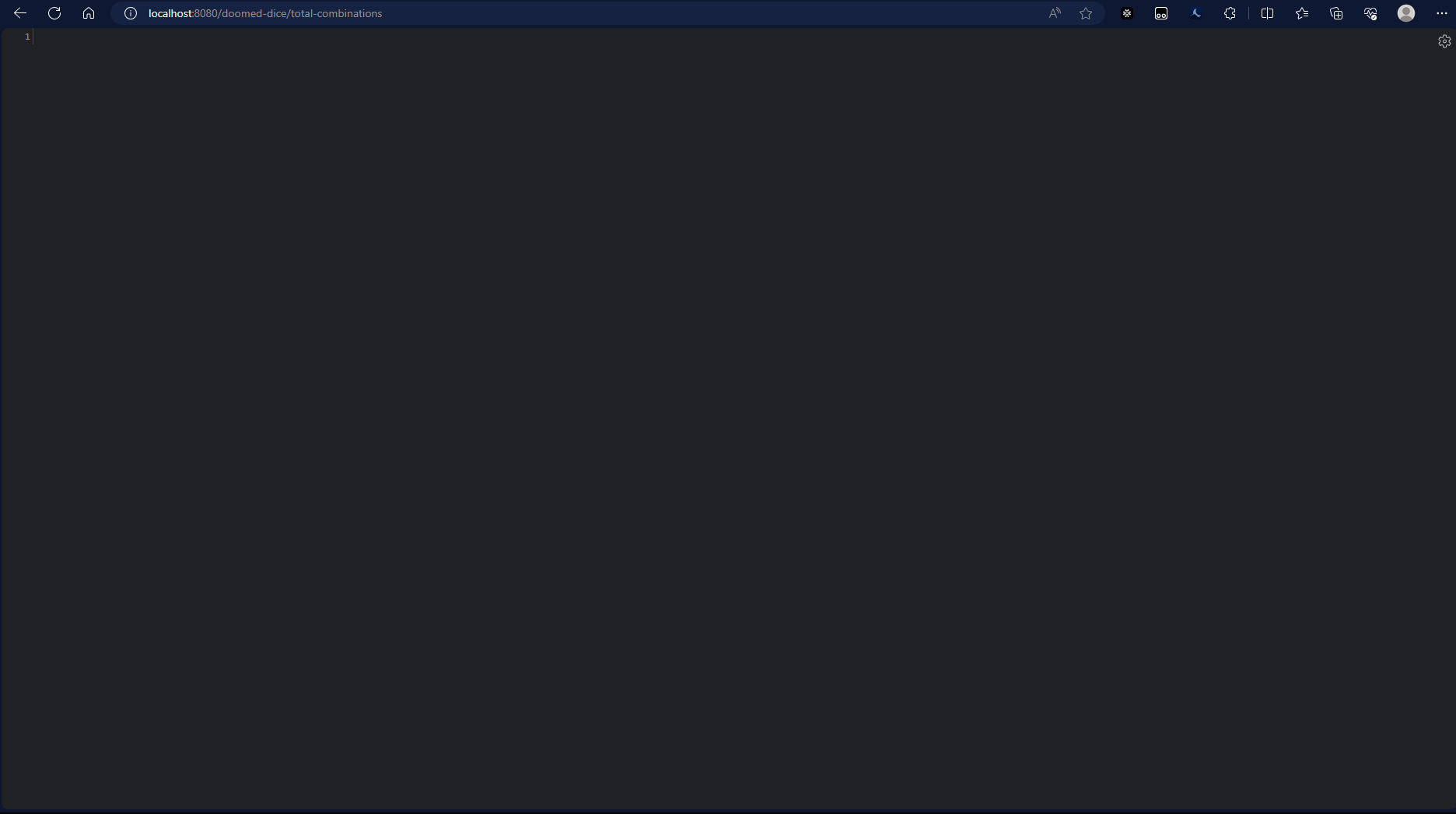
int numberOfDice = 2;

int sides = 6;

int total = (int) Math.pow(sides, numberOfDice);

return ResponseEntity.ok(total);

}



1. **Calculate and display the distribution of all possible combinations that can be obtained when rolling both Die A and Die B together.**

NumberOfFaces=6

TotalCombinations=36

Distribution:

We use two nested loops to iterate through each face of Die A and Die B. For each combination, we calculate the sum of the values on Die A and Die B. We create a Map to store the combination details, including the values on Die A, Die B, and their sum. The distribution list is then populated with each combination map.

**Code:**

public ResponseEntity<List<Map<String, Integer>>> getDistribution() {

int sides = 6;

List<Map<String, Integer>> distributionList = new ArrayList<>();

for (int i = 1; i <= sides; i++) {

for (int j = 1; j <= sides; j++) {

int sum = i + j;

Map<String, Integer> combinationMap = new HashMap<>();

combinationMap.put("Die A", i);

combinationMap.put("Die B", j);

combinationMap.put("Sum", sum);

distributionList.add(combinationMap);

System.out.println("Combination Map: " + combinationMap);

}

}

return ResponseEntity.ok(distributionList);

}  
 }  
] 





1. **Calculate the Probability of all Possible Sums occurring among the number of combinations from (2).Example: P(Sum = 2) = 1/X as there is only one combination possible to obtain Sum = 2. Die A = Die B = 1.**

Two nested loops iterate through each face of Die A and Die B. For each combination, the sum of the values on Die A and Die B is calculated.

The distribution array is updated to track the count of occurrences for each sum.

The probabilityMap is initialized to store the probability for each possible sum.

The loop iterates through the distribution array to calculate the probability for each sum.

The probability is expressed as a fraction of the total combinations (36).

**Code:**

public ResponseEntity<Map<Integer, String>> getProbability() {

int sides = 6;

int[] distribution = new int[11];

for (int i = 1; i <= sides; i++) {

for (int j = 1; j <= sides; j++) {

int sum = i + j;

distribution[sum-2]++;

}

}

Map<Integer, String> probabilityMap = new HashMap<>();

for (int i = 0; i <= 10; i++) {

probabilityMap.put(i + 2, distribution[i]+ "/" + 36);

}

return ResponseEntity.ok(probabilityMap);

}

