TOPIC 1 : INTRODUCTION

1.Given an array of strings words, return the first palindromic string in the array. If there is no such string, return an empty string "". A string is palindromic if it reads the same forward and backward.

Example 1:

Input: words = ["abc","car","ada","racecar","cool"]

Output: "ada"

Explanation: The first string that is palindromic is "ada".

Note that "racecar" is also palindromic, but it is not the first.

Example 2:

Input: words = ["notapalindrome","racecar"]

Output: "racecar"

Explanation: The first and only string that is palindromic is "racecar".

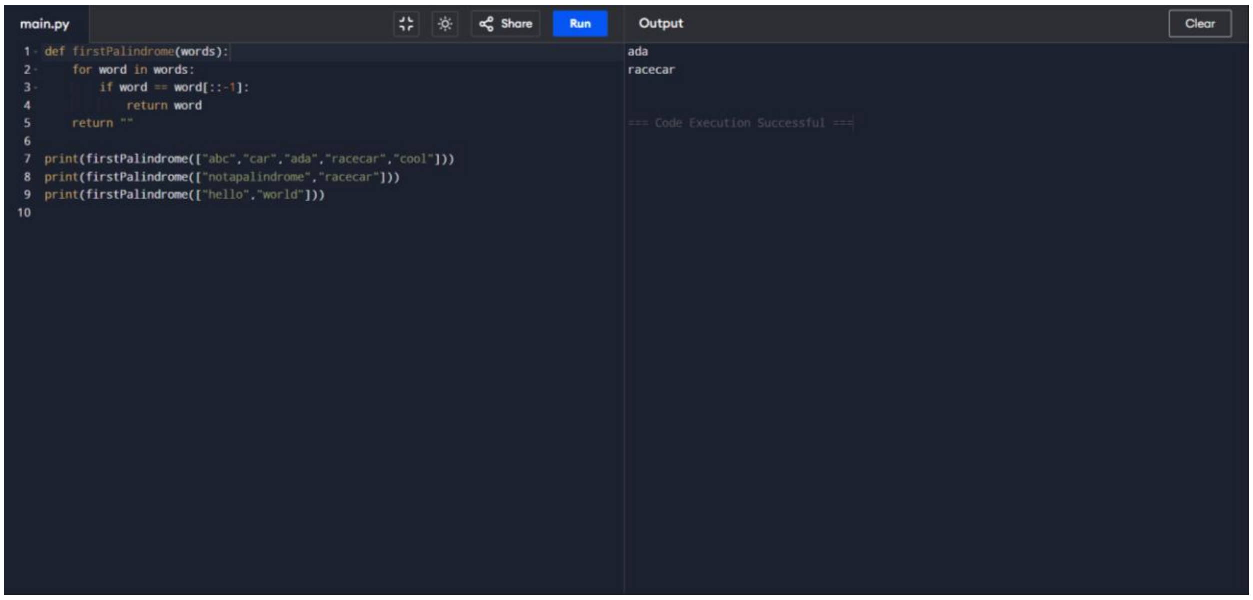
Aim

To write a program that finds the first palindromic string in the given array of words and returns it. If no palindrome exists, return an empty string "".

Algorithm

1. Start.
2. Read the array of strings words.
3. For each word in the array:
   a. Check if the word is equal to its reverse.
   b. If yes, return the word immediately.
4. If no word is palindromic, return "".
5. Stop.

Input and Output



```python
def firstPalindrome(words):
    for word in words:
        if word == word[::-1]:
            return word
    return ""

print(firstPalindrome(["abc","car","ada","racecar","cool"]))
print(firstPalindrome(["notapalindrome","racecar"]))
print(firstPalindrome(["hello","world"]))
```

```
ada
racecar

=== Code Execution Successful ===
```

Result

The program successfully finds and returns the first palindromic string in the given list of words.

Performance Analysis

Time Complexity:

- General: O(n·m)
- Example: n = 5 words, m = 7 (longest word "racecar")
- Value → O(35)

Space Complexity:

- Using reverse method → O(m) = 7
- Using two-pointer method → O(1)