**Tracking Unique Website Visitors**

**Problem Statement**: Tracking Unique Website Visitors Using HashSet in C#

**Objective**: Write a C# program to track unique website visitors for a single day using a `HashSet`, ensuring that repeat visits by the same user are not counted multiple times.

**Description**:

You are developing a website analytics tool that monitors visitor traffic. Each visitor is identified by a unique visitor ID (e.g., a user ID or session ID). The tool receives a stream of visitor IDs throughout the day, and some visitors may visit the website multiple times. Your task is to use a `HashSet` to record only unique visitor IDs, ignoring duplicates, and provide a report of new and repeat visitors. The program should display whether each visitor is new or a repeat visitor and provide the total count of unique visitors at the end.

**Requirements**:

1. Use `HashSet<string>` to store unique visitor IDs.

2. Process a list of visitor IDs, simulating incoming visits.

3. For each visitor ID, check if it’s new or a repeat visit using the `HashSet`.

4. Display a message for each visitor ID indicating whether it’s a new or repeat visitor.

5. Display the total number of unique visitors at the end.

6. Ensure the solution is efficient and handles duplicates automatically using `HashSet`.

**Input**:

- A list of visitor IDs (strings), e.g., { "user123", "user456", "user123", "user789", "user456" }.

**Output**:

- For each visitor ID, a message indicating whether it’s a new or repeat visitor.

- The total count of unique visitors.

**Example output:**

New visitor: user123

New visitor: user456

Repeat visitor: user123

New visitor: user789

Repeat visitor: user456

Total unique visitors: 3

**Constraints**:

- Visitor IDs are case-sensitive strings.

- The input list may contain any number of visitor IDs, including duplicates.

- The solution must use `HashSet` for efficient deduplication and lookup.

**Learning Goals:**

- Understand how to use `HashSet` in C# for tracking unique elements.

- Learn to use the `Add` method to detect new versus duplicate entries.

- Practice processing a stream of data and summarizing results.

- Explore the efficiency of `HashSet` for real-time deduplication tasks.