Individual Assignment #4

Implement the ADT Map Using Hashing with Chaining.

You are given two class templates: *MapEntry<KeyType, ValueType>* (a simple key–value pair) and *HashedEntry<KeyType, ValueType>* (extends MapEntry with a nextPtr for chaining).

1. Complete the implementations of *HashedEntry* and *MapEntry* in *HashedEntry.cpp* and *MapEntry.cpp*, respectively.
2. Implement a class template *HashedMap<KeyType, ValueType>* that:
   - Uses a *std::vector<std::shared_ptr<HashedEntry<KeyType, ValueType>>>* as the hash table.
   - Resolves collisions using chaining, where each bucket is a singly linked list.
   - Supports at least the following operations:
     - ✓ *bool add(const KeyType&, const ValueType&)*
     - ✓ *bool remove(const KeyType&)*
     - ✓ *bool getValue(const KeyType&, ValueType& out) const*
     - ✓ *bool contains(const KeyType&) const*
     - ✓ *bool isEmpty() const*
     - ✓ *int getNumberOfEntries() const*
     - ✓ *void clear()*
   - Use the hash function we discussed in class.

3. Test your implementation by:
   - Creating a *HashedMap<std::string, int>* and inserting several *key-value* pairs.
   - Demonstrating that collisions are handled (e.g., by designing inputs that hash to the same index).
   - Showing that *add, remove, getValue*, and *contains* all work correctly.

Submission Requirements:

- Capture screenshots of your test cases and submit them together with your files.
- Deadline: 11/26/2025.