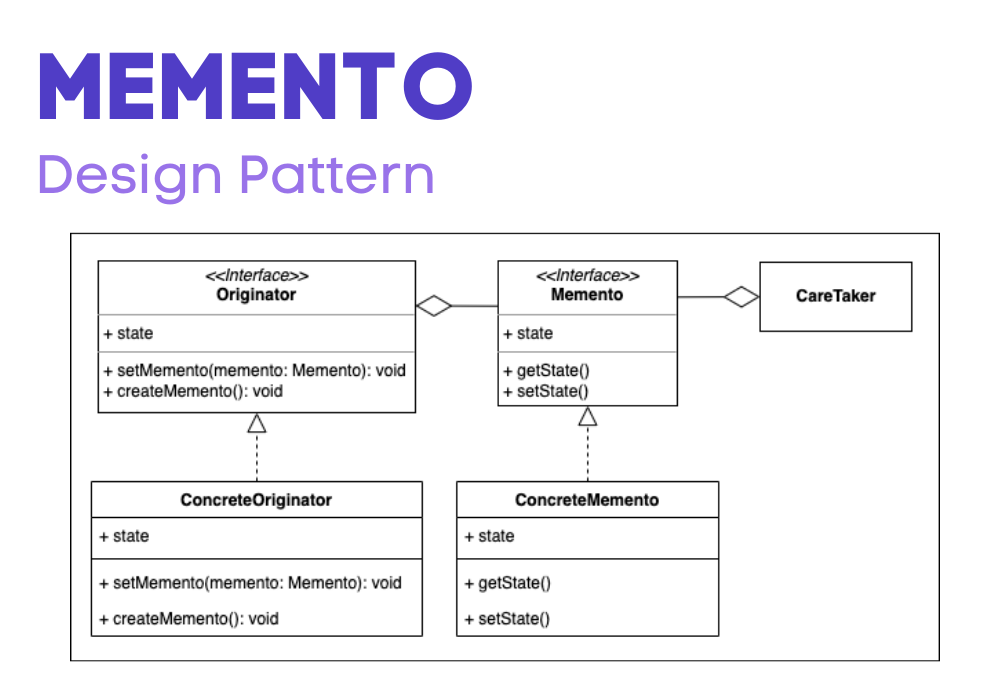
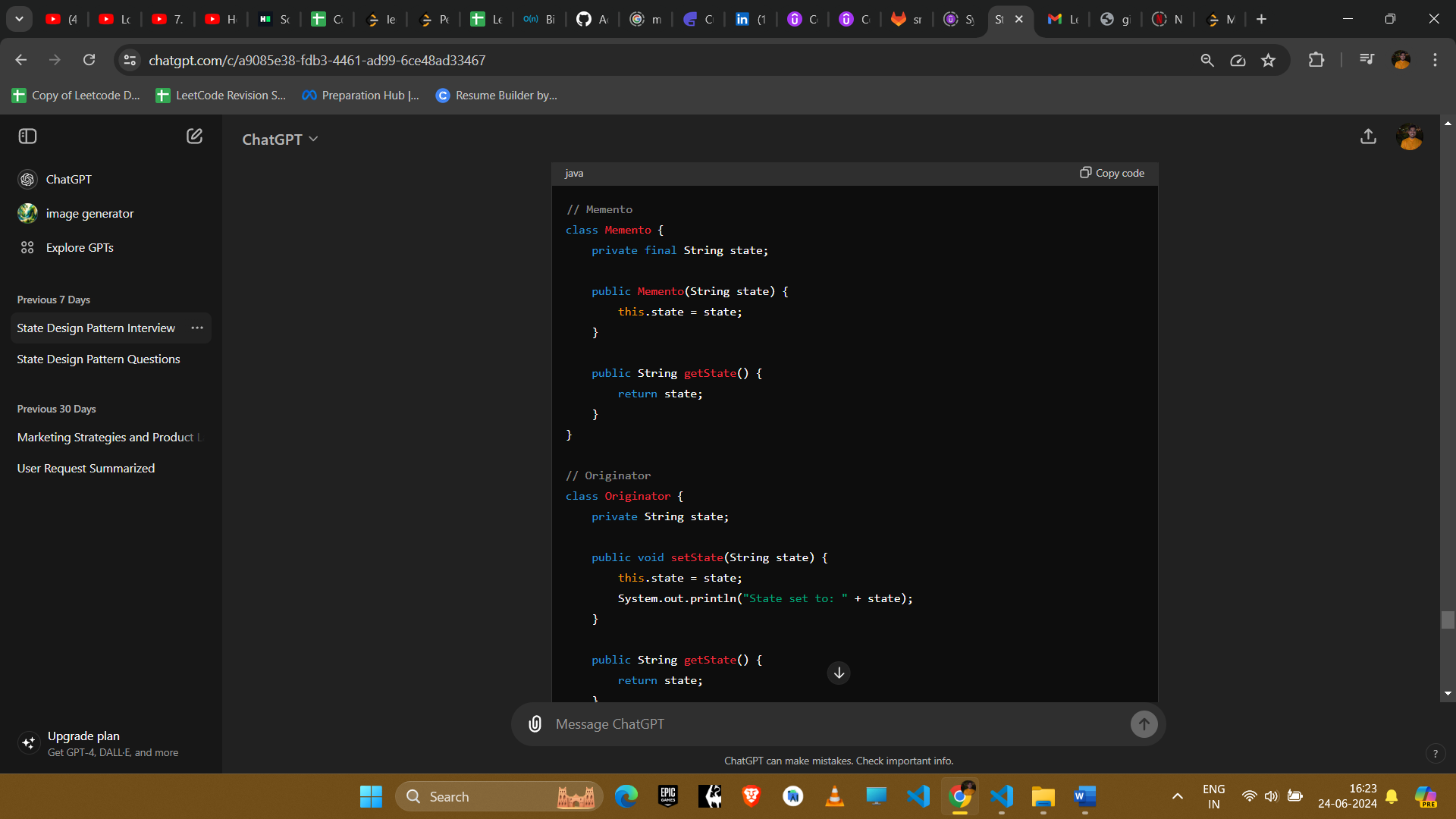
**Memento Design Pattern**

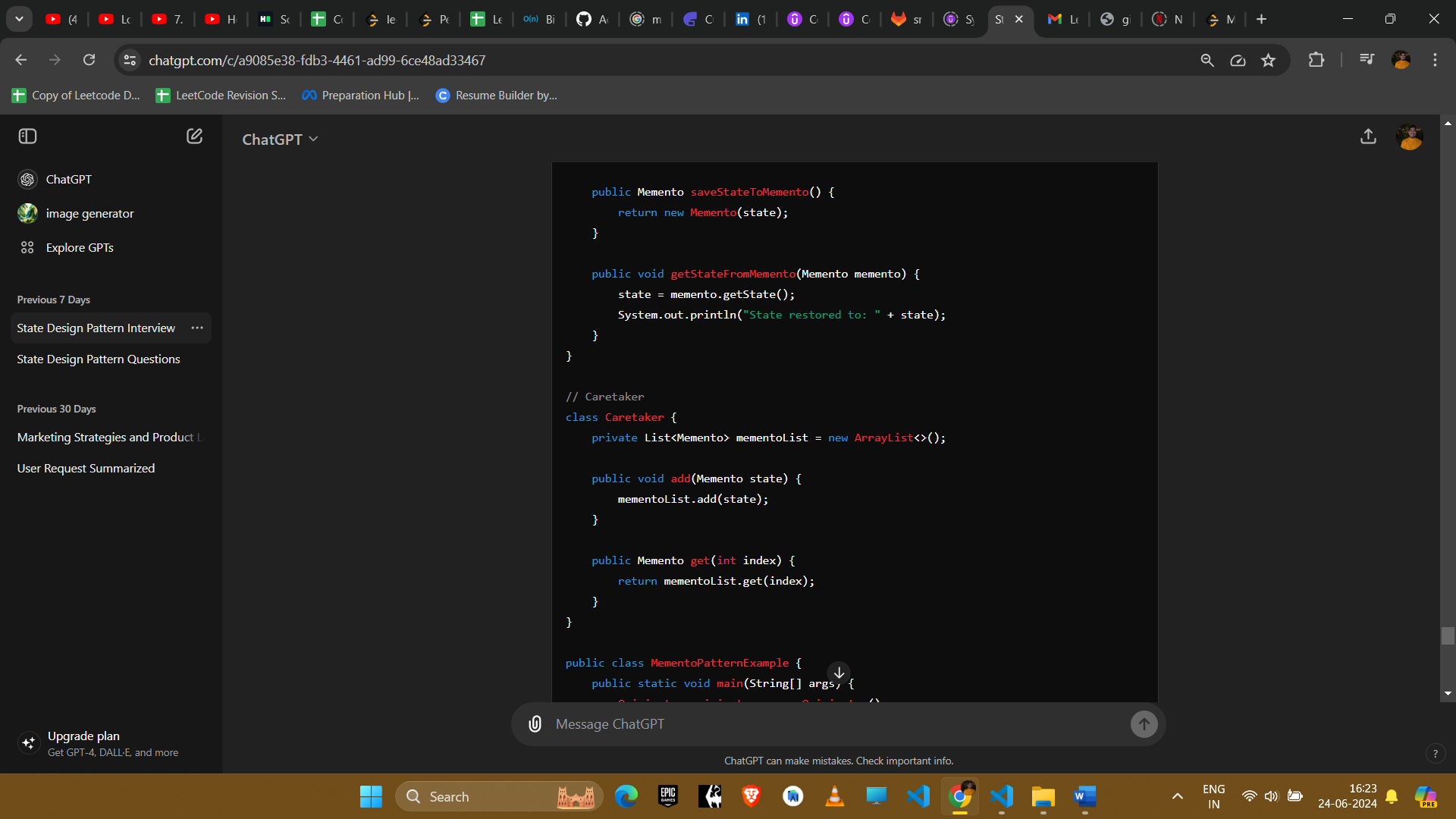
The Memento design pattern is a behavioural pattern used to capture and externalize an object's internal state so that the object can be restored to this state later, without violating encapsulation. This pattern is commonly used for implementing undo mechanisms.

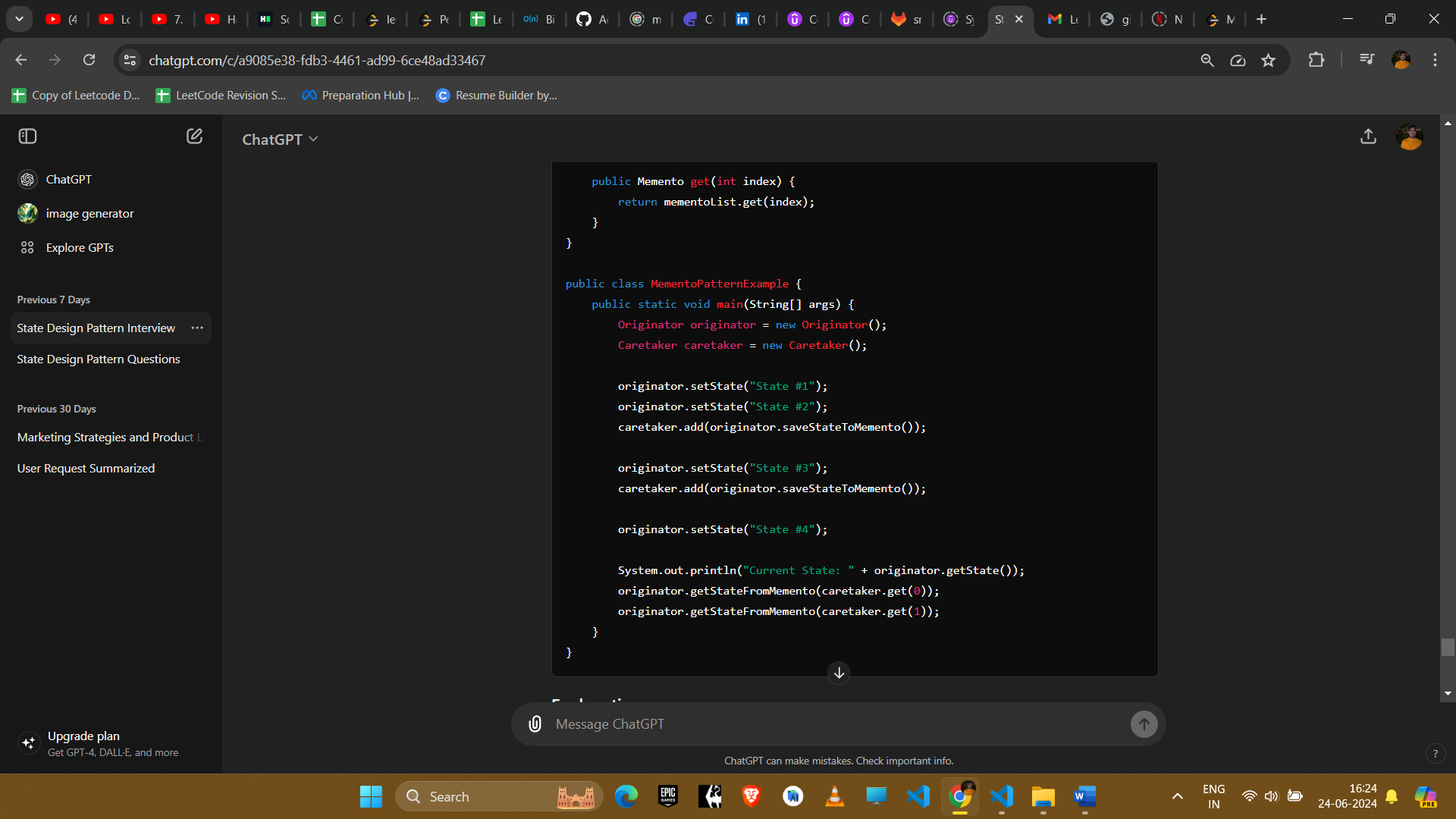
Consists of 3 categories :

1. **Originator**: Creates a memento containing a snapshot of its current state and uses the memento to restore its state.
2. **Memento**: Stores the internal state of the Originator object.
3. **Caretaker**: Manages the memento. It is responsible for storing and retrieving mementos.









### Explanation

1. **Memento**: Stores the internal state of the Originator object.
2. **Originator**: Creates a memento containing a snapshot of its current state and uses the memento to restore its state.
3. **Caretaker**: Manages the memento. It is responsible for storing and retrieving mementos.

### Example Uses in Amazon Interviews

#### 1. **Text Editor**

* **Scenario**: Implementing undo/redo functionality in a text editor.
* **Implementation**: Use the Memento pattern to save the state of the text after each operation (typing, deleting, formatting) and restore it on undo.

#### 2. **Game Development**

* **Scenario**: Saving and restoring game states.
* **Implementation**: Use mementos to capture the state of a game (e.g., player position, score, level) and restore it when needed (e.g., loading saved games).

#### 3. **State Management**

* **Scenario**: Keeping track of different states in an application.
* **Implementation**: Use mementos to store the state of an object at different points in time and restore previous states when necessary.

#### 4. **Object Serialization**

* **Scenario**: Saving and loading the state of an object.
* **Implementation**: Use mementos to serialize an object's state to a persistent storage and deserialize it later to restore the state.

#### 5. **Database Transactions**

* **Scenario**: Implementing savepoints in database transactions.
* **Implementation**: Use mementos to capture the state of the database at a specific point in a transaction and roll back to that state if needed.

### Conclusion

The Memento design pattern is useful for scenarios where you need to capture and restore the state of an object. It provides a way to implement undo mechanisms and manage object states without violating encapsulation. This pattern is particularly useful in applications requiring state rollback, history tracking, or state restoration.