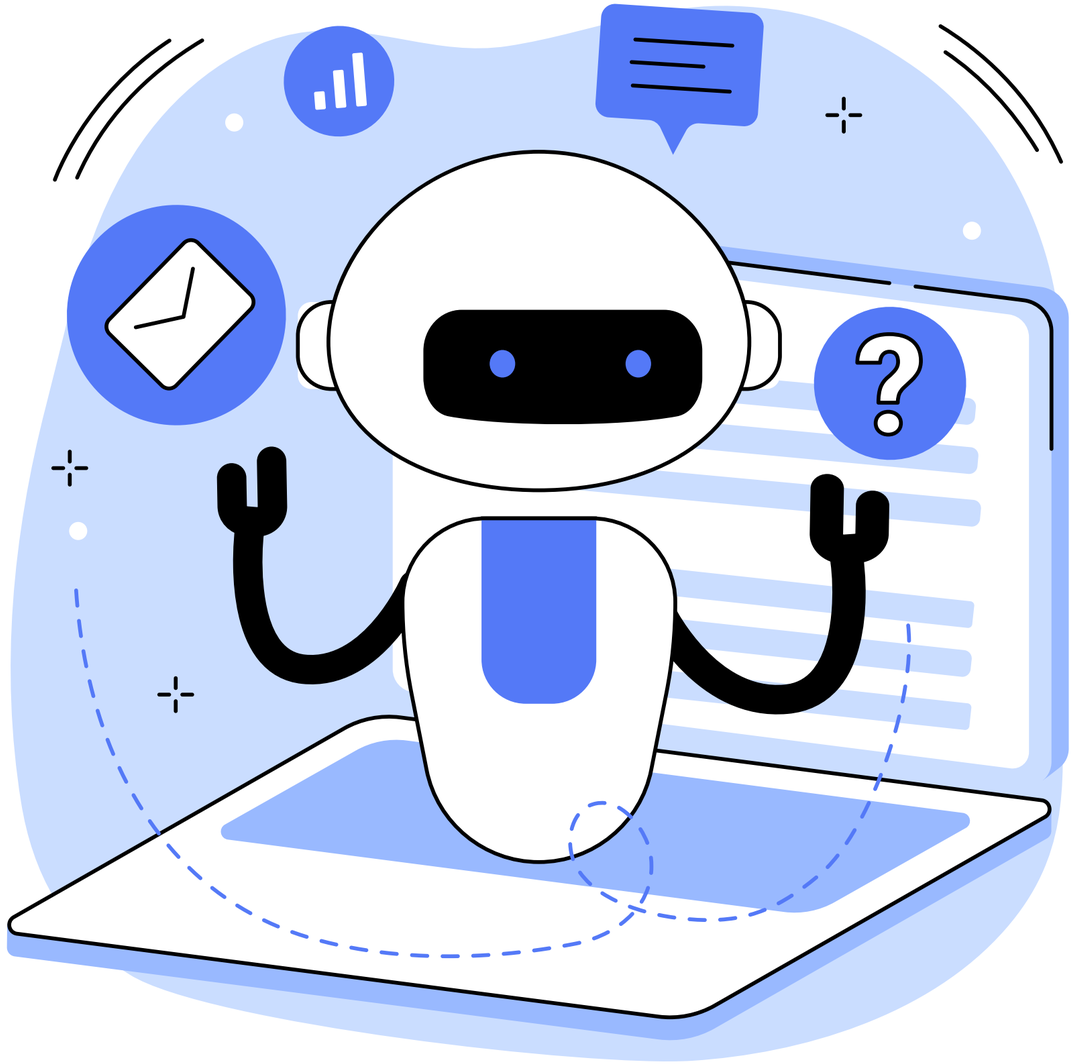
OBSERVER PATTERN

CONTENTS

WHAT IS OBSERVER PATTERN?

Observer is a behavioral design pattern that lets you define a subscription mechanism to notify multiple objects about any events that happen to the object they’re observing



ADVANTAGES

The Observer pattern promotes loose coupling between objects, as they are only loosely connected, allowing for more flexibility and maintainability of the code.

The Observer pattern provides a scalable way to handle multiple events and listeners.

The Observer pattern promotes reusable code.

The Observer pattern makes the code more maintainable.

DISADVANTAGES

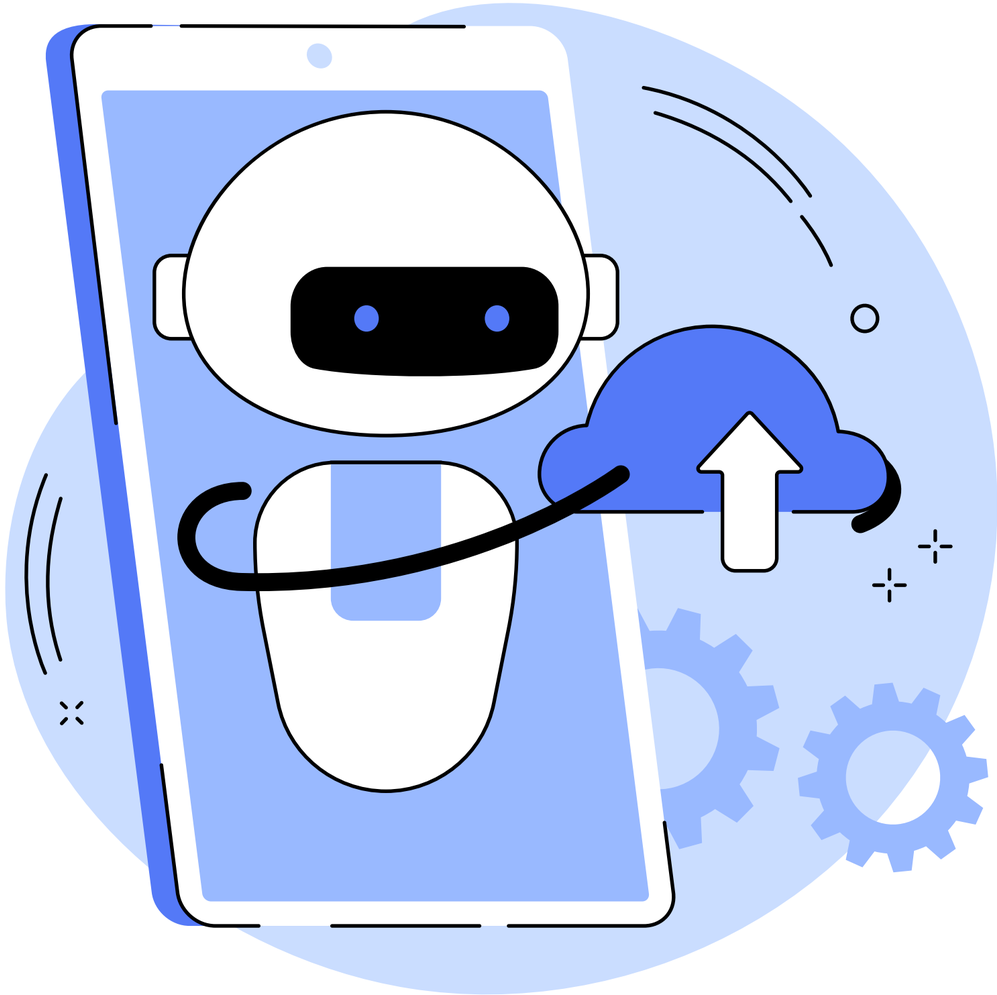
The Observer pattern can increase the complexity of the code.

The Observer pattern can incur a performance overhead due to the frequent notifications that are sent to the observers.

The Observer pattern can lead to memory leaks if the observers are not properly removed when they are no longer needed, leading to unnecessary memory consumption and reduced performance.

WHEN SHOULD WE USE THE OBSERVER PATTERN:

The Observer pattern is appropriate in situations where:



WHAT HAPPENS IF WE DON’T USE THE OBSERVER PATTERN WHEN IT’S NEEDED:

THANK YOU FOR LISTENING!