MAYANK KAUSHAL
EMP ID= 2579352
GITHUB LINK= https://github.com/mayank5654/OOPs-Based-System-for-Storing-School-Data-Using-Design-Patterns

Object-Oriented School Data Storage System with Design Patterns

Problem Statement:-The task at hand is to design and implement an Object-Oriented Programming (OOP) based system for storing comprehensive school data. This system is constructed in C# and emphasizes the integration of design patterns to enhance its modularity, flexibility, and maintainability. The overarching goal is to craft a robust solution using design patterns while ensuring the effective storage and retrieval of data related to students, teachers, and subjects.

Requirements:

Development Environment:

Utilize Visual Studio for the creation of a Windows Console Project.

OOP Principles:Implement classes following Object-Oriented Programming principles such as encapsulation, inheritance, and polymorphism.

Design Patterns Integration:Utilize design patterns to enhance the system's architecture.Explore patterns like Singleton, Factory, or Observer for specific functionalities.

Student Class:

Properties:Name,Class,Section

Teacher Class:

Properties:Name,Class,Section

Subject Class:

Properties:Name,Subject Code

Teacher (Reference to the teacher associated with the subject)

Console .NET Project:-Develop a Windows Console Project to serve as the user interface for interacting with the designed classes.

Data Storage and Retrieval:-Implement methods to store and retrieve data efficiently.Leverage principles like encapsulation to safeguard data integrity.

Implementation:Leverage Visual Studio to create the Console .NET Project and define the Student, Teacher, and Subject classes following OOP principles. Integrate design patterns where applicable to enhance the system's architecture. Implement methods for efficient data storage and retrieval, ensuring a seamless experience for users.

Design Pattern Integration:-Explore design patterns like Singleton for ensuring a single point of access to certain functionalities, Factory for creating instances of classes, or Observer for handling data updates efficiently.

Testing:-Functional Testing:-Execute the program to validate that the designed classes operate as intended.

Design Pattern Validation:-Ensure that the integration of design patterns enhances the system's flexibility and maintainability.

User Interaction Testing:-Verify that the Console .NET Project provides an intuitive interface for users to interact with the system.

Conclusion:-The OOP-based school data storage system, enriched with design patterns, delivers a robust solution for managing school-related information. The careful application of OOP principles and design patterns ensures scalability and maintainability, creating a foundation for efficient data storage and retrieval. The Console .NET Project offers a user-friendly interface, facilitating seamless interaction with the designed classes.