**Assignment – 1: 15/01/2025**

Develop a "Database Connection Manager" program that supports connecting to different types of databases, such as MySQL, PostgreSQL, MSSQL and MongoDB(for available databases) . The system should use the Factory Design Pattern to abstract the creation of database connection objects.

1.

Create an interface/abstract class DatabaseConnection with methods:

connect(): Simulates establishing a database connection.

disconnect(): Simulates closing the database connection.

Implement concrete classes for each database type:

MySQLConnec, and MongoDB(for available databases)tion: Connects to a MySQL database.

PostgreSQLConnection: Connects to a PostgreSQL database.

MongoDBConnection: Connects to a MongoDB database.

Create a DatabaseFactory class with a static method getDatabaseConnection(String dbType) that returns the appropriate DatabaseConnection implementation based on the dbType parameter.

Include a main method to simulate connecting to a database based on user input

2.

The connect() method in each class should print a message like:

For MySQL: Connected to MySQL Database.

For PostgreSQL: Connected to PostgreSQL Database.

For MongoDB: Connected to MongoDB Database.

The disconnect() method should print a message indicating the connection is closed.

3.

Optional :(this upto individual students - to implement or not to implements )

Simulate additional functionality for each database:

MySQL: Perform a query like SELECT \* FROM users.

PostgreSQL: Perform a query like SELECT \* FROM orders.

MongoDB: Perform a query like db.collection.find({}).

User inputs the database type (mysql, postgresql, mongodb) and performs actions like connect, query, and disconnect.

4.

Submission Guidelines:

Submit a .java /C# file with well-commented code.

Include an intent, structure diagram, class diagram, roll of participants, mapping participants with actual class, consequences and consequences.

Provide test cases to demonstrate connecting to each database type.

Submission date- 20 /01/2025

--

**Assignment – 2: 27/01/2025**

Implement a food ordering system for a multi-cuisine restaurant that serves Indian and Chinese cuisine. Each cuisine has different types of dishes: main course, appetizers, and desserts. Use the Abstract Factory Pattern to create families of related food items while maintaining consistency within each cuisine.

implement any two methods in a concrete class related to problem definition

Submission date- 03 /02/2025

**Assignment – 3: 03/02/2025**

Design and implement a Database Connection Manager for a Student Management System using the Singleton design pattern. The system should maintain a single instance of the database connection throughout the application's lifecycle to efficiently manage database resources and prevent multiple unnecessary connections.

**Assignment – 4: 10/02/2025**

Problem Statement:

\* Implement a Decorator Pattern in Java to enhance the functionality of InputStream and OutputStream.

\* The decorators should add additional features such as:

\* - Logging the number of bytes read and written.

\* - Counting the number of characters, words, and lines in the input stream.

\* - Maintaining flexibility and modularity by wrapping standard Java I/O classes.

\* The InputStreamDecorator should:

\* - Extend InputStream and wrap another InputStream instance.

\* - Track and print the number of characters, words, and lines read from the stream.

\* - Log the bytes being read.

\* The OutputStreamDecorator should:

\* - Extend OutputStream and wrap another OutputStream instance.

\* - Log the bytes being written.