**WEEK 3**

**High-Level Architecture Diagram: A diagram of a computer

AI-generated content may be incorrect.**

**Selected Design Patterns & Justification:**

**1. Observer Pattern**

**Description:**  
The Observer Pattern is used to automatically notify multiple components when a particular event happens.

**Usage in Project:**  
In the social media app, the Observer Pattern can be applied for real-time notifications — for example, when a user receives a new message, a new comment, or a new follower.

**Justification:**  
By using the Observer Pattern, we can ensure that all relevant parts of the system (such as notification systems or live updates) respond automatically to changes without tightly coupling the components. This improves scalability and maintainability.

**2. Factory Pattern**

**Description:**The Factory Pattern provides an interface for creating objects without having to specify the exact class of the object that will be created.

**Usage in Project:**  
In the backend, the Factory Pattern can be used to create different types of posts (e.g., text posts, image posts, video posts) dynamically based on the user's input.

**Justification:**  
The Factory Pattern allows for flexible and scalable object creation. If the app needs to support new types of posts or messages in the future, we can easily extend the system without modifying existing code, following the Open/Closed Principle.

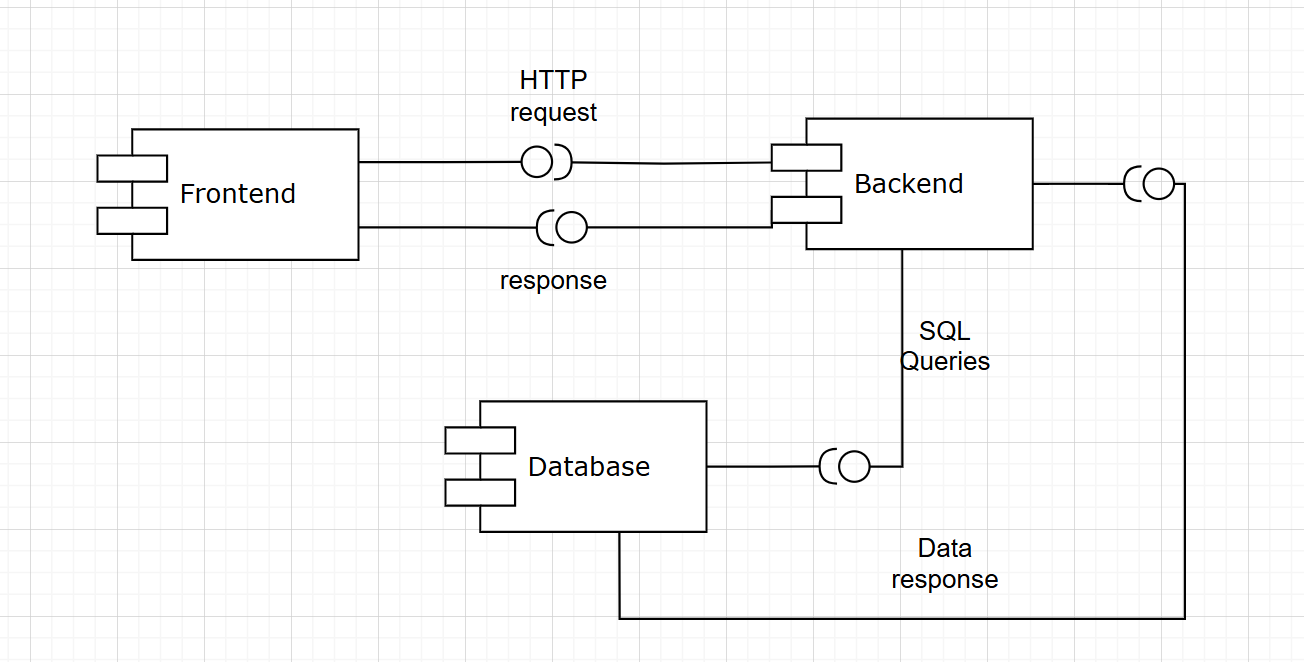
**3. Singleton Pattern**

**Description:**  
The Singleton Pattern ensures that a class has only one instance and provides a global point of access to it.

**Usage in Project:**  
The Singleton Pattern can be used for the database connection manager to ensure that there is only one connection being maintained between the Django server and the database.

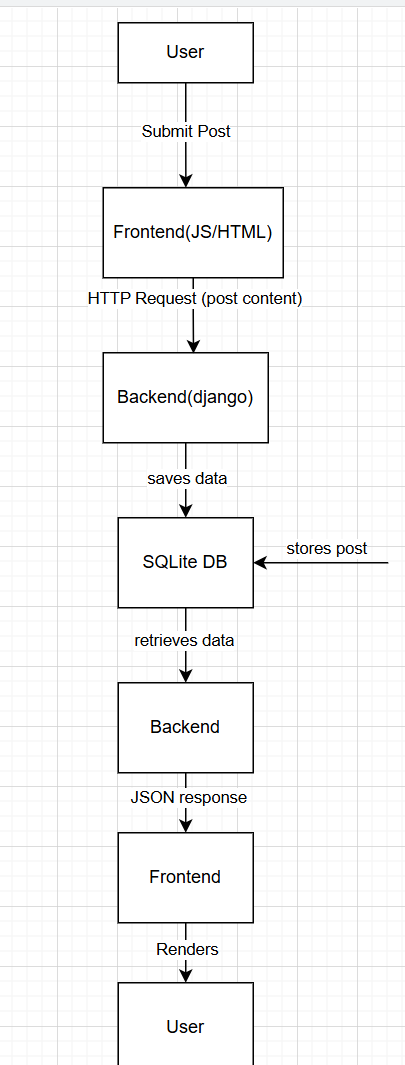
**Justification:**  
Using a Singleton for database management prevents resource wastage and conflicts caused by multiple simultaneous connections. It ensures efficient resource utilization and a centralized control point for database operations.

**Component diagram**

****

**Deployment diagram:**

**Dataflow Diagram:**

****