Explanation of Changes

1. **Encapsulation and Class Design**

   The code is refactored into a UserManager class, encapsulating all user management operations. This adheres to the **Single Responsibility Principle** (SRP), as each method within the class has one responsibility.

2. **SQL Injection Prevention**

   Prepared statements ($stmt->bind_param) are used to prevent SQL injection attacks, replacing the vulnerable string concatenation in SQL queries.

3. **Method Separation**

   The operations (addUser, deleteUser, updateUser, listUsers) are separated into different methods, making the code more modular and easier to maintain. This change adheres to the **Open/Closed Principle** (OCP), as the class is open for extension but closed for modification.

4. **Error Handling**

   Added basic error messages for operation failures, making the code more user-friendly and informative.

5. **Naming Conventions**

   Variables and methods are renamed to be more descriptive ($u to $user, op to operation). This makes the code more readable and easier to understand.

6. **SQL Queries**

   Moved away from concatenated SQL queries to prepared statements, which is a best practice for both security and readability.