**GHAZI KAISSI 9/10/2024**

**INTERNSHIP AT INTERNGRUB (back-end development)**

REPORT (FIRST TASK):

1. **Differences Between MySQL, MySQL I and PDO:**

**MySQL Extension:** It is the very first offered PHP extension regarding MySQL database. It simply provides a procedural programming interface which is untouchable by modern sophistication such as prepared statement. It is considered reproachable and its usage is discouraged in modern applications.

**MySQL i:** Prepared queries: This feature of MySQL I supports both procedural and object-oriented programming styles. It enables you to use prepared statements and execute multiple queries, but it is limited to MySQL databases only.

**PDO:** With PDO, the database access is not limited to one type of database, unlike just MySQL. It allows programming regardless of the current database system used, thus making it possible to switch to a different database with minimum modifications to the codes. The library also provides native support for prepared queries.

1. **Why Use Prepared Statements:**

Due to the nature of prepared statements in how they are structured, SQL logic is not written in close proximity to the data, as it is the case with normal statements. This makes it possible to execute any SQL injection easily since every user input is treated as a data hence no command can be executed. This becomes useful because repeated queries with prepared statements do not have to be prepared every time thus saving time.

1. **Benefits of PDO:**

**Cross-Database Support:** It is because it can work with other databases such as postgreSQL and sqlite in addition to MySQL unlike mysqlI’s support with only MySQL.

**Data-Access Abstraction:** On some occasions, it is the business logic that provides different functions to be performed, regardless of the different databases in place without making changes to the basic code structure.

**Prepared Statements**: Particularly useful in acquiring user details by providing a safe method of communicating with the database while also enhancing the efficiency of communication by eliminating the potential for SQL injection.

**Error Handling:** Such as ‘try and catch’ whereby PDO provides you with a uniform method of dealing with errant situations on any database so that the task of about restoring a database and controlling the outcome of such restorations becomes less complicated.

1. **PDO Components and Overview:**

**DSN (Data Source Name):** Refers to the type of database and the information used to connect to the database.

**Connection Management:** Though PDO is an abstract layer that is uniform across several database types and systems, it does not mean that it covers all aspects regarding the management of server connections.

**Prepared Statements and Binding:** Works in enhancing the security of the queries and the performance.

**Transactions:** Transaction management is present allowing serializable database actions to be implemented.

1. **Object-Oriented Programming (OOP) in PDO:**

As noted before, PDO is a completely object oriented, meaning of more systematic approach in the organization and reusing codes.

It is possible to write classes to interact with the database that will open connections, issue queries, and deal with errors all in an OOP style.

1. **Functions in PDO:**

Creating a connection, creating a database, making a query, retrieving data, and committing or rolling back transactions is made possible by PDO. These functions are the same in every supported database, and developmental aspect is simplified.

1. **Security and MySQL Injection (CMS):**

**Sql Injection Shield:** The technique of prepared statements used by PDO makes SQL injections impossible which means it is ideal for implementations like CMS (Content Management Systems).

**Error Management:** Application of PDO assists in better security of web applications particularly because of its inbuilt error handling and exception throwing features.

**CONCLUSION**

This summary highlights the advantages of using PDO, particularly for its flexibility, security, and modern approach to database management, particularly when developing secure CMS applications.

The data shows that most of the web applications that are developed use this method for better security of the application. Most of these systems that are developed publish a lot of content on the web, hence the need for web applications for content management system.

SUPERVISOR

MOHAMMAD AL ABED