CP 311: Internet Programming and Applications II

Markup Test

Question One

In Object Oriented Programming (OOP), a class is considered as the blueprint while an object is considered as an instance of a class. Create a class named Product with an instance of that class named product1. Add properties (name, description and price) to the class Product. For the object instantiated as product1, modify the property name to 'iPhone 12', description as "Apple product" and price as "Tsh 2,000,000". Then, display the name, description and price to see the result.

Question Two

You are developing a web application for a bookstore. The application requires storing information about books, including their title, author, and price. Design a PHP class called "Book" that represents a book object. The class should encapsulate the book's properties and provide methods to set and retrieve the book information. Additionally, implement a method in the "Book" class called "saveToDatabase()" that connects to a MySQL database and stores the book information in a table called "books". The "books" table should have the following columns: "id" (auto-incrementing primary key), "title" (varchar), "author" (varchar), and "price" (decimal). The database server can be accessed via the IP Address 10.10.10.8, and the source code is hosted at the server with IP address 10.10.10.10. Use "bookApp" as the name of the database, book123 as password and bookUser as the username.

> **NB:** to connect your php scripts with database use either MySQLi (object-oriented) or PDO

Question Three

Create a web application that has two parts. The front-end part features a JavaScript function that uses Ajax to send a request to the back-end part (located at https://library.example.com/api/books) to retrieve a list of available books. The request must use the GET method and include the following:

- **Bearer Token**: LibraryBearerToken123
- **JSON Body**: The request should include a key "requester" with the value "library_user" and a key "action" with the value "fetch_books".

The back-end part should have functions that:

1. Validate the Bearer token and the requester.
2. Check if the action is valid.
3. If all validations pass, respond with a JSON object containing a detailed list of available books. Each book should have properties: title, author, genre, year, and ISBN.