Promtovi koje smo iskoristili:

Zou are teacher of PhP cours4e on university level.

I onda smo dodali nas postojeciu kod. Chat GPT je predloyio sledece iymene koda:

It looks like you've provided a PHP code snippet for a class hierarchy related to shop products, along with a class for writing and fetching product data. This is a good start! Below are a few observations and suggestions for improvement:

**1. Consistency in Naming:**

* In the CdProduct and BookProduct classes, method names such as getPlayLength and getNumberPages follow camel case. For consistency, you might want to update the ShopProduct methods like getName and getSurname to follow the same convention.

**2. Type Hinting:**

* In the getPrice method of the ShopProduct class, the return type is declared as string, but it seems like it should be int|float to match the property type. Adjust the return type accordingly.

**3. Error Handling:**

* The fetchProductData function performs database operations. It's good practice to include error handling, especially when dealing with database connections. Consider using try-catch blocks to catch and handle exceptions.

**4. Database Connection:**

* In the fetchProductData function, the database connection is hardcoded to SQLite. You might want to make the database type configurable or provide a more flexible way to set up the database connection.

**5. SQL Injection:**

* The bindParam method is used in the fetchProductData function, which is good for preventing SQL injection. However, if possible, consider using named placeholders in the query to make the code more readable and maintainable.

**6. Returning Data in printgetSummary:**

* The printgetSummary method seems to be returning a discounted price, but the return type is declared as string. If the intention is to return the discounted price, you may want to change the return type to int|float.

**7. Documentation:**

* Consider adding comments or docblocks to explain the purpose of each class and method, especially if the code is intended for collaborative development.

**8. Autoloading and Namespace:**

* As the codebase grows, consider using autoloading for classes and organizing them into namespaces for better code organization.

**9. Testing:**

* Implement unit tests for your classes and methods to ensure that they work as expected.