OOPs Project

Graded Project – 70 marks

Problem Statement:

An Ecommerce vendor management system supports various features such as storing the product data for different vendors. Here each vendor can save various products and fetch that information whenever required. The focus of the project is to design and implement a few functionalities based on that.
Each vendor will have its user_name and password, based on that they can login and retrieve the saved product information.

This product/feature will leverage the core OOPs concepts covered such as class, object, methods, inheritance, abstraction etc.

There are two main tasks at hand in this project. These involve writing codes to implement the Models package and Implementation package.These packages will have implementation of functionalities for Products and Vendors both. In this project, you will write the functionality to add a vendor, login using the credentials of an existing vendor, add products in the catalog and query the products by name as well as the entire list, and logout.

Domain: Ecommerce Vendor Management

Data Set & Program Organization: The simple program is structured in various layers.

1. **Vendors.csv:** This is a sample csv which contains the vendor details
   a. Vendor Id
   b. Vendor Name
   c. User Name
   d. Password
2. **The Boilerplate:** The boilerplate is sections of code that are repeated in multiple places with little or no variation. Programmer write lot of boilerplate code to accomplish only minor functionality. Here a pre-designed boilerplate is being provided so that the learners can focus on the core programming

   a. **Abstractions:**
      i. Vendor: This file will have the implementation abstract class and methods that are going to be implemented in the Implementation package for vendor related operations. The abstract method in Vendor section will be there to do *login(), logout().*
      ii. Products: This Segment will have the implementation abstract class and methods that are going to be implemented in the Implementation package

        for product related tasks. The abstract method in Product segment will be there to do *add_product()*, *search_product(), all_products().*

   b. **Models:**
      i. VendorModel: Implementation of class and methods that implements the functionality of loading existing vendors. In this model we are also checking if the correct vendor is passed for VendorSessionModel.
      ii. ProductModel: Implementation of the class and methods that are performing Create and Read operation on the product database.
      iii. VendorSessionModel: Before performing several operations such as adding the product information, reading the product data, functionality such as login, logout of the vendors are implemented in this file.

   c. **Implementation:**
      i. ProductsImplementation: In this segment learners should implement the actual functionalities that are mentioned in driver segment for product implementation. The methods in the created class will be the extension of the methods created as abstract methods in the abstraction package.
      ii. VendorImplementation: In this segment, learners should implement the actual functionalities that are mentioned in driver segment for vendor implementation. The methods in the created class will be the extension of the methods created as abstract methods in the abstraction package.

   d. **Driver:** This segment imports the implementation package for product and vendor that has all the methods already implemented and available for use. Take a close look at the method names that are being used to perform the functionality.

Tasks & Evaluation Criteria:
You are provided with a project, which contains the boilerplate code which includes the Driver. You have to write the Models and Implementations of the project.

1. Add code for implementation package
   a. Implement the login method available under the VendorImplementation segment. Implementation of this method has two operations (**10 Marks)**
      i. it should check if the passed `username` vendor is available and it is the credentials of a legitimate vendor.
      ii. If it is true then it should perform the login method available under `vendorsessionmodel`.
         - Valid Vendor Output: `User xyz logged in successfully!`
         - Invalid Vendor Output: `Invalid username or password.`

   b. Implement the logout method available under the VendorImplementation segment. (**10 Marks)**
      Valid Vendor Output: `User xyz logged out successfully!`

   c. Implement `add_product` method that will call the relevant method available in ProductModel segment after checking if the username is valid or not for the given operation. **(10 Marks)**

   d. Implement `search_product_by_name` method that will call the relevant method available in ProductModel segment after checking if the username is valid or not for the given operation. If the given product is not available print the message that "the asked product is not available". **(10 Marks)**
   e. Implement `get_all_products` method that will call the relevant method available in ProductModel segment after checking if the username is valid or not for the given operation. If there are no products then it should print an error message, else print the product details. **(10 Marks)**
2. Add code for Models package
   a. Implement `search_product` method. If the product_name passed as parameter is available in the dictionary then return the product details else return `None` **(10 Marks)**
   b. Implement `all_products` method. Return all the data saved in the product dictionary. **(10 Marks)**

Instructions

1. Download the **CSV** and **Boilerplate** give
2. Open the Jupyter notebook with Python 3.6 or higher version of python installed.
3. In the boilerplate Models package and driver have some codes.. Once you have added the code based on the given task, you can run the Driver to verify if the added code is working successfully or not.
4. Run the **Driver commands,** and examine the result. This file will currently run various simple calls that will communicate with different classes and methods in those classes.
5. Once the program is ready to submit, upload the ".ipnyb" and "HTML" file on the portal.