Programming language for Business Analytics-Lab Project

Submission deadline: 24th December 2024

Project Overview:

Students will design and implement an **Online Ordering System** using Python to simulate the operations of a business accepting orders from customers. The system will allow customers to browse products, place orders, and receive invoices. It will support functions like managing product catalogues, handling customer orders, generating invoices, and maintaining data persistence. This project will combine programming concepts with real-world business scenarios, enhancing students' skills in system development and data management.

Key Features:

1. **Product Catalogue Management:**
    - Add, update, or delete products (e.g., Product ID, Name, Category, Price, Stock Quantity).
    - Display the catalogue for customers to browse available products.
2. **Customer Order Management:**
    - Allow customers to add products to a shopping cart.
    - Calculate the total price, including optional discounts or taxes.
    - Validate stock availability before confirming orders.
3. **Invoice Generation:**
    - Automatically generate an invoice for each order, including:
        - Order ID, customer details, date, itemized list of products, total price, taxes, and discounts.
    - Save invoices in a file (CSV, JSON, or PDF).
4. **Order History and Reporting:**
    - Store and retrieve past orders for customer and business reference.
    - Generate sales reports for specified time periods (e.g., daily, weekly, monthly).
5. **Data Persistence:**

- Maintain product catalogs, order history, and invoices using file handling (CSV, JSON, or SQLite).
- Load data on program start and save updates when the program exits.

6. **User Interface (bonus points):**
    - Provide a menu-driven console interface.
    - Offer separate modes for customers (placing orders) and admins (managing products and viewing reports).

Project Deliverables:

1. **Source Code:** Python scripts implementing the SIS with modular code design.
2. **Sample Data Files:** Include preloaded CSV or JSON files for testing the system.
3. **User Manual:** A guide explaining the system's features, navigation, and usage.
4. **Presentation + Viva:** A brief demonstration of the project and its functionality. Followed by viva voce.

Development Guidelines:

1. **Core Programming Concepts:**
    - Use **functions** for modular programming.
    - Leverage **lists and dictionaries** to manage data in memory.
    - Implement **file I/O operations** for persistent data storage.
    - Use **try-except blocks** to manage runtime errors.
2. **Code Quality:**
    - Write clean, readable code with meaningful variable names.
    - Include appropriate comments and documentation in the code.

Assessment Criteria:

Criteria	Marks	Description
System Design and Planning	5	Quality of system structure and feature organization.
Functionality	10	Implementation of core modules (student, attendance, and fees management).
Code Quality	15	Readability, modularity, and comments.
Error Handling	5	Robustness, input validation, and overall user experience.
Creativity and Novelty	5	Inclusion of advanced or unique features (e.g., user interface, use of functions or OOP).
Documentation, Presentation and Viva	10	Clear user manual and professional project presentation.