

# Advanced Programming with Java

## Final Sprint- Java Console-Based E-Commerce Platform

---

### Project Overview:

You have been tasked to develop a console-based E-Commerce platform using a Maven based Java project using PostgreSQL. This project is designed to simulate a real-world online marketplace where users can register as buyers or sellers, list products for sale, and browse available products. There should be a CLI Menu like the first sprint approach. The project should be implemented as a **TEAM** effort, with each group consisting of up to three students. Each Member should contribute to the project

---

### Project Objectives:

- Develop a robust Java application that simulates an E-Commerce platform.
  - Implement user authentication and role-based functionality. Buyer, Admin and Seller (They are all a user) should be able to do different functions in the menu.
  - Integrate with a PostgreSQL database to manage user and product data.
  - Demonstrate effective teamwork and project management skills (Project board, working in an agile way, use of PRs and branching in your repo)
- 

### Functional Requirements:

1. **User Registration and Authentication:**
  - Users can register with a username, password, email, and role (buyer, seller, or admin).
  - Registered users can log in to the system. Based on their role they will be able to see a specific menu.
  - Passwords should be securely stored (Use the BCrypt Maven dependency)
  - Examples of required classes( User, UserDao, UserService, Buyer(Inheritance from User), Seller(Inheritance from User), Admin(Inheritance from User),)
2. **Role-Based Functionality:**
  - **Buyers** can browse and view products, search for a specific product, see product info.
  - **Sellers** can add, update, and delete their own products, they should be able to view a list of all their products that they have listed
  - **Admins** can view a list of all users including their contact information and delete a user from the system. Admins can also see a list of products that are in the system. The list should include the Seller name and information.

### 3. **Product Management:**

- Sellers can add new products with details like name, price, quantity, and seller\_id
- Sellers can view and manage their listed products.
- Example classes could include (Product, ProductDAO, ProductService)

### 4. **Database Integration:**

- Use PostgreSQL to store and manage user and product information.
- Implement CRUD operations (Create, Read, Update, Delete) for users and products.
- (Hint for your database... There should be at least a table for Users and a table for products. There could also be more if you want to!)

### 5. **User Interface:**

- Develop a console-based user interface for interaction.
  - Provide clear prompts and options for users to navigate the application.
- 

## **Technical Requirements:**

1. **Programming Language:** Java
  2. **Database:** PostgreSQL or one that you choose
  3. **IDE:** Any preferred Java IDE (e.g., IntelliJ IDEA, Eclipse, VS Code)
  4. **Version Control:** Use Git/GitHub for version control
- 

### 5. **Maven Build System**

## **Submission Guidelines:**

- **Code Submission: 60 Marks**
  - Submit the complete source code of your application include your documentation with your repository submission (**ZIP FILES WILL NOT BE MARKED**). **If you need help with understanding GitHub, please reach out to your TA.**
- **Database Schema: 10 Marks**
  - Provide the SQL scripts used to create and populate your database.
- **Documentation: 20 Marks Prepare** one document that will act as a report for your program. It should include **all** the below sections.
  - **User Documentation** This include a document stating what the application is about, explanation of all the classes and their working, and how to start it/access it. Also include the class diagram with the associations between them.
  - **Development Documentation** This includes at least the Javadocs, a description of the source code directory structure, the build process (i.e., how to compile the project), compiler time dependencies, development standards, how to set up a database for development, and how to get the source code from the repository
  - **Deployment Documentation** This is basically the installation manual of the application, describing any steps needed to make it run

- **Submit an individual report of team contributions (PRs/Branches Worked On), and any challenges faced during this project from your perspective**
- **Presentation: 10 marks**
  - Prepare a short video (5-10 minutes) demonstrating your application and explaining your development process. **All team members should take part in the presentation this is part of your mark!**

### **Challenge (Is Not Required for Project Submission)**

- **Run The Code in another IDE besides the one you are using. i.e. IntelliJ**
- **Implement proper error handling in your program (Use the try catch approach showed in class)**

### **Evaluation Criteria:**

Your project will be evaluated based on the following criteria:

1. **Functionality:** How well does the application meet the outlined functional requirements?
2. **Code Quality :** Is the code well-organized, easy to navigate , does it follow proper naming conventions?
3. **Database Integration :** How effectively is the database used to manage and store data?
4. **Team Collaboration :** How well did the team work together to deliver the project? Is Branching present on your repo? Was a task board set up to work in an agile way?

---

### **Important Dates:**

- **Project Kickoff:** November 25<sup>th</sup>, 2024
- **Code Submission Deadline:** December 15<sup>th</sup> , 2024

---

### **Resources:**

- [Java Documentation](#)
- [GitHub Guides](#)
- [PostgreSQL Documentation](#)
- [Example Repo From TA Sessions](#)

---

### **Instructor Support:**

Your instructor and TA are available to provide guidance and support throughout the project. Do not hesitate to reach out for help with technical issues or team management. Getting stuck is good! Don't get stuck for too long... Reach out to us for help!

---

**Good Luck and Happy Coding!**

---