CS 202: Semester Project - Part 2

Fall 2025

Important Information

Deadlines:

- **Part 1**: **4th of January 2026 (Sunday) at 23:59**

The **deadline** is very **strict** and it will **NOT** be changed under any conditions.

Friendly Reminder: All TA's of this course are a member of Artificial Intelligence Center of Özyeğin University, which means they can understand the use of AI tools such as *ChatGPT* in your project just at a glance.

TAs' Email Address:

- Tuğçe Özgirgin tugce.ozgirgin@ozu.edu.tr
- Azra Oymaağaç: azra.oymaagac@ozu.edu.tr

Honor Code

As part of our commitment to academic integrity and ethical conduct, all students undertaking the *Database Management System* assignment/project are expected to adhere to the following honor code:

1. **Original Work:** All project submissions must be the original work of the individual student or group. Plagiarism or any form of unauthorized collaboration is strictly prohibited.
2. **Citations and References:** Properly cite and reference any external sources, including books, articles, websites, or any other materials used in the project. Failure to acknowledge sources is considered a violation of academic honesty.
3. **Independent Effort:** Each student or group is expected to complete the project independently without seeking unauthorized help from other students, online sources, or any other external parties unless explicitly permitted by the instructor.

4. **Honesty and Authenticity:** The project submission must accurately represent the student's or group's own understanding and effort, reflecting the knowledge and skills gained throughout the course.

5. **Respect for Academic Integrity:** Uphold the principles of academic integrity and abide by the rules and guidelines provided by the instructor and the institution.

6. **Adherence to Course Policies:** Follow all guidelines, deadlines, and instructions specified in the project description and course syllabus. Non-compliance may result in penalties.

7. **Report Violations:** Students are encouraged to report any suspected violations of the honor code to the instructor for appropriate investigation and action.

Consequences of Violating the Honor Code: Violations of the honor code will result in disciplinary action as per the course or institutional policies. Penalties may include but are not limited to failing grades on the project, failing the course, or academic probation.

By submitting their project, each student or group acknowledges their understanding and commitment to upholding this honor code.

1 Description

In Part 2 of this project, you will implement a user-friendly GUI (Graphical User Interface) application for the E-Commerce Order Management System that you designed and submitted in Part 1.

When writing your program, you should use a schema/database in your MySQL server for this project. You can't create the tables in Workbench; you must write the corresponding DDL and DML files by hand (You can use the same ones as in Part 1).

Preject Overview:

- Language Used: **Java + JDBC**
- GUI Toolkit: **Swing**
- Database: **MySQL**

All database operations (insertion, deletion, modification, retrieval, filtering, and aggregation) must be performed using SQL queries executed via JDBC. You are not allowed to retrieve large datasets and process or filter them in Java code instead of SQL. The final outcome of Part 2 should be a fully functional application that demonstrates the correct and efficient use of relational databases through a graphical interface, reflecting both the theoretical design principles and practical database management skills covered in the course.

2 Requirements for Part II

You are required to determine the DB dynamics and mandatory actions. In the following a list of some example actions are given; you can extend your project's, but these specifications are a **MUST**.

The **mandatory** requirements of the E-Commerce Order Management System:

- The system must support separate user types, each with distinct access rights. At a minimum, the following user roles must be implemented. **At least**:
  - Customer
  - Seller / Manager
  - Admin/Manager
- Users, can create a customer and seller accounts from the home page with a sign-up functionality.
- Sellers and Customers will login to the system with their own password in order to use the system.
- Admin accounts must already be present in the DB, or a DB Admin should only be able to add them.
- Customers must be able to view catalogs belonging to different sellers.
  - Each catalog must list the products offered by that seller.
  - Customers must be able to view detailed information for each product, including at least the product name, description, price, category, stock availability, past customer reviews and ratings.
  - Customers must be able to add products to an ongoing order directly from the catalog or item detail view. The system must prevent adding products whose stock quantity is zero.
- Each customer must have at most one ongoing order(Shopping Cart) at a time, which they can view. An ongoing order may contain products from only one seller.
- An order can only be submitted if the order contains at least one product
- Customers must be able to view their order history, including:
  - Past orders.
  - Order dates and total amounts.
  - Order status (e.g., pending, paid, shipped, delivered, canceled)
  - Or leave a review. (Review can be submitted only after the associated order has been marked as "shipped")

- The customer interface must provide statistical summaries, calculated using SQL queries, including:
  - Monthly Total Purchase Amount.
  - Most Purchased Product Category.
  - Average Monthly Purchase Amount

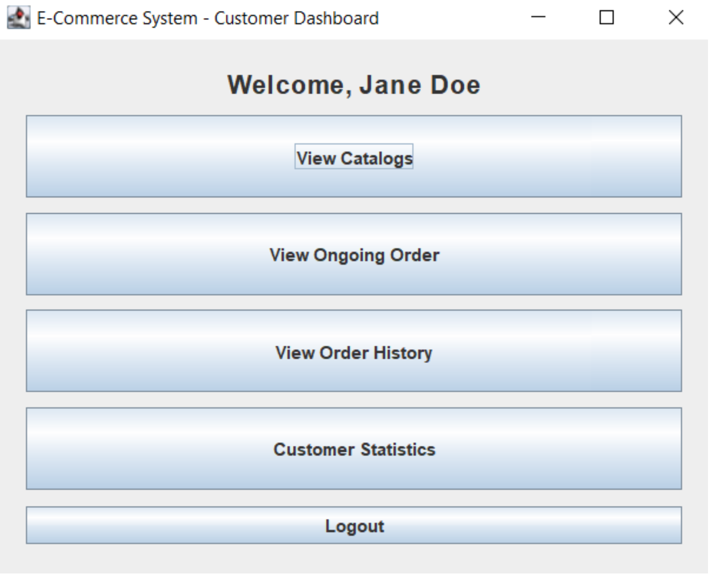

Figure 1: Representative Customer Landing Page (You can customize it as you wish)

- Sellers must be able to view their own catalog and the products listed within it. They must be able to:
  - Add new products to their catalog.
  - Sellers Update existing product information.
  - Remove products from their catalog
- Sellers must only be allowed to manage products within their own catalog.
- Sellers must be able to view orders placed for their catalog and update order-related statuses.
- Sellers must be able to view reviews left by customers for products in their catalog.
- The seller interface must provide aggregated statistics, computed using SQL queries, including at least:
  - Total revenue per month.
  - Best-Selling Product(s).
  - Most Rated Product(s).
  - Average Order Value

- Sellers must be able to update inventory levels explicitly to reflect restocking events. This operation must update stock quantities without modifying past order records.

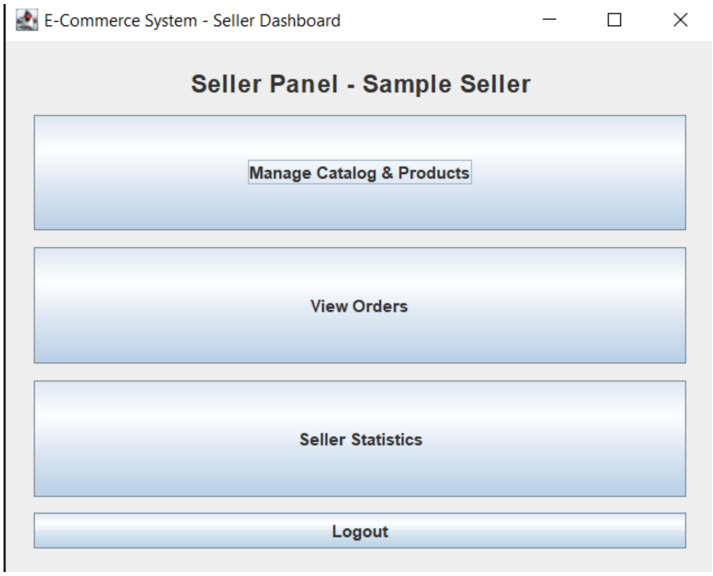
E-Commerce System - Seller Dashboard
Seller Panel - Sample Seller
Manage Catalog & Products
View Orders
Seller Statistics
Logout

Figure 2: Representative Seller Landing Page (You can customize it as you wish)

- Administrators must be able to view a list of all users in the system.

- Administrators must be able to add new users, modify existing user information, and delete users.

- The administrators can add/delete/modify categories.

- Administrators must be able to view all shipments in the system. Administrators must be able to approve or update shipment statuses only for orders that have been confirmed by the seller (payment status done).

- The administrator interface must provide aggregated system-level statistics, computed using SQL queries, including at least Total Sales, Top-Selling Categories, Top Sellers, and Most Popular Items.

You must conduct your research on the e-commerce system and incorporate new features into it (It is **NOT** optional it is mandatory to add additional features).

Also, please keep in mind that all of these functionalities must be implemented using SQL queries. For example, for calculating the statistic, you can't just retrieve all the data and filter this data in Java code, your solutions must be SQL-based!

3 Deliverables of Part 2

Second Part's Deadline: 4th of January 2026 (Sunday) at 23:59

In this part you are required to implement the followings:

- An **USER-FRIENDLY** GUI application, which **MUST** have a Home page, where users can Register to the system, or if they already registered, they can Login to the system.
- The application **MUST** have different interface for different users, their capability to manipulate the program should differ based on their access level.
- Your application program **MUST** handle all possible *Exceptions*.

The expected deliverables for this part of the project:

- You can only submit the working version of your code and nothing else.
- You are allowed to only submit a single .zip file. No other file type is not accepted (i.e. rar)
- Do not add directories that contain auto-generated compiler output folders like bin, tmp and out.
- Before sending the .zip file, make sure that it works when it is extracted from the .zip file!
- Your code should not be scoped inside a package, in other words not part of your Java code should include the phrase ”package ozu.xxxx.xxx(etc);”

Your submission must follow the structure below:

- `Group_[GROUP_ID]_CS202_HW_PART2.zip`
  - `Report.pdf`
  - `DDL.sql`
  - `DML.sql`
  - A folder containing your Java source file(s)

Note: You are expected to refine your reports from Part 1 based on the provided feedback.

4 Final Notes

Your project will be graded both in terms of functionality and usability, so please don't forget to look at your project from a 'User Experience' perspective.

You are free to add additional entities based on your design.

After the submission, demo sessions will be conducted. During the demo, your code will be downloaded directly from your submission, and you will be required to present your system and answer related questions. The demo session schedule (location and time) will be announced later. If you fail to attend your allocated demo session, **no points will be awarded**. Please ensure that you are present on time.