

CTIS 256
Introduction to Backend Development
PROJECT
Spring 2025

Deadline for Submission: 18 May 2025, Friday, 23:59

Demonstration (15 mins): The team will demonstrate their project on a laptop at my office on **19th of May, Monday**. Any member who doesn't attend to the meeting will not be graded. *The team should reserve an appropriate time slot from the course's Moodle page.*

Submission: All project source codes, assets and database exports should be uploaded to Moodle by one of the team members by deadline (18th of May).

Team work: Each team member will be responsible for a particular part of the project.

A Sustainability e-Commerce Project

Problem: The waste of expired products in the markets poses a serious problem in terms of the effective use of resources.

Solution: An information system that makes it easier to sell products that are nearing their expiration date at a lower price than their normal price can reduce this problem. Thus, the products of the markets will not be wasted and the consumer can buy the product cheaper. This creates a win-win situation for both the markets and the consumers.

Objective: You will develop a multi-user, web based application in which you are free to design your own user interfaces and the database design using php, and mysql.

Functional Requirements:

- There are two different types of users in the system, namely, consumers and markets.
- **Market User:**
 - A market can sign up for the system with its email, market name, password, city, district :
{
 email : "abc1@gmail.com", **name** : "ABC – Birlik ", **password** : "*****",
 city : "Ankara", **district** : "Çankaya"
}
 - **(5Pts)** A market can update/edit its own information described above. (Edit)
 - **(10Pts)** A market user can add a new product nearing its expiration date into its inventory. A product is described by its "title", "stock", "normal price", "discounted price", "expiration date", and an image. Example: "Toblerone 100gr", "25" pieces, "20" tl, "12" tl, "22-05-2022", and the image of the product, respectively.
 - **(5Pts)** A market user can delete and edit a product (including a new product image)
 - **(5Pts)** When the market user logs in, it marks all expired products in the list.

- **Consumer User:**

- A consumer will sign up for the system with her *email* address, *fullname*, *city*, and district.
- (5Pts) A consumer can update her own information. (Edit)
- (10Pts) A consumer will search products with keywords (search bar). It finds all products involving any keywords, exactly or partially. For instance, searching “mag” keyword finds all products containing “**magnum**”, “**magnolia**-cake”, “nut**mag**”, etc. The result shows only the products in the same city with the customer’s, and it gives precedence to the products in the same district. The result should not contain any expired products. It will paginate the results with a page size of 4 products.
- (5Pts) A consumer can add a product into the shopping cart. Shopping cart will be saved into the database. It remembers the shopping cart once the user logs in.
- (10Pts) In the shopping cart, she can update the shopping cart (delete, change the amount), and it shows the grand total. Use **AJAX** for this part.
- (5Pts) There will be “Purchase” button in the shopping cart page, and it will empty the cart and drops the products from the system. Use **AJAX** for this part.
- (15Pts) **Registration**: The system should register **Market** and **Consumer Users** with email addresses and it verifies given email addresses with an email message which contains a confirmation code such as a 6-digit random number.
- (5Pts) **Login**: The users will log into the system via email address and password. It gives error message if the authentication fails. Password should be stored in hashed form.

Non-Functional Requirements:

- (5Pts) All forms should be validated and offer sticky-form functionality for usability.
- (5Pts) Applications must be protected against sql injection, XSS, and CSRF attacks.
- (10Pts) The system must have a decent user interface.
 - You can make use of html templates, third party jQuery plugins and/or bootstrap-like css frameworks in frontend.
- The system should have sufficient amount of meaningful data in the demonstration, do not use garbage test data for products and markets.
- Use PHP Session to track the users.

- The requirements give the overall picture of the application. Therefore, you can use your own initiative for missing parts.
- Cheating is strictly prohibited. Otherwise, all team members get zero.
- In the demonstration, I’ll ask questions to all team members about the project. I’ll assess your grades based on your answers. You’ll get grades depending on your contribution to the project.
- **20 Points penalty** for late submission. The submission will be closed on 18th of May, 23:59.