

TEAM 02 PHP PROJECT: PROJECT DOCUMENTATION
DE ONLINE GROCERY

1. PROJECT DESCRIPTION (3-5 PAGES)

Leader: Manj, Dilraj 300331319

Member: Edison CHAN 300370372

1.1 SPECIFY YOUR GROUP MEMBERS AND THE ENTITIES THEY ARE RESPONSIBLE FOR.

Distribution of work (Dil – Dilraj, Ed – Edison)

1. Criteria Grading (QA):
 - a. Dil and Ed both reviewed and evaluated the web application based on specific criteria, and the following points were assessed:
2. Project Submission (Dil):
 - a. Dil took responsibility for submitting the project to Blackboard and ensuring that all the required assets were included.
 - b. The submitted project file was named appropriately according to the naming convention.
 - c. Dil earned 1 point for successfully fulfilling this task.
3. Project Documentation (Both of us):
 - a. Project Description (Ed)
 - b. Installation Manual (Dil)
 - c. Class Diagram (Ensuring accuracy and inclusion of all properties and cardinality) (Dil)
 - d. Meeting Minutes (Ed)
 - e. User Manual (Concise, graphical, properly labeled) (Dil)
4. Program Structure and Comments (Both of us):
 - a. Both Dil and Ed ensured that the program structure was well-organized and used appropriate naming conventions.
 - b. Dil documented all functions and classes with descriptive comments.
5. Database Design and SQL Script (Dil):
 - a. Dil was responsible for designing the database to meet the requirements and reflect real-life web applications.
 - b. The complexity of the database was ensured to be similar to or greater than any database design used in class practices.
 - c. The SQL script was included in the project submission.
6. CRUD Operations and SQL Queries (Dil):
 - a. Dil implemented CRUD operations using PDO, DAO, and prepared statements.
 - b. Errors were captured and logged, and users received positive confirmation of CRUD operations.
 - c. Statistics were coded using SQL queries and displayed.
7. Input Validation and Error Handling (Ed):

- a. Ed was responsible for validating all input according to expected criteria.
 - b. The web app prompted users for corrections when necessary, and stack traces were logged to a file in case of errors.
8. HTML Forms and Interface (Ed):
 - a. Ed ensured that HTML forms were coded properly, and the layout was efficient and user-friendly.
 - b. Interface controls were made intuitive for users.
9. Overall Project Quality (Both of us):
 - a. Dil and Ed collaborated to ensure that the overall quality and complexity of the submitted project exceeded any class demo practices and assignments.
10. Bonus requirements 2 points (~~Ed~~ → ~~Publish on GitHub~~ → **Used worst case backup, project is edited on GitHub but no hosting took place**)

1.2 YOU MUST DESCRIBE YOUR PROJECT AND EXPLAIN HOW ALL THE ENTITIES INTERCONNECT TO CREATE YOUR APPLICATION.

TABLE: USER

Fields: This table contains information about the users of the website. Fields might include [userid, username, email, password]

Purpose: The "user" table serves as the central repository for all user, allowing user acting as admin to add and remove product, and also changing their price

TABLE: PRODUCT

Fields: This table contains information about the products available on the website. Common fields might include product_id, name, description, price, category, availability, and other product-specific attributes.

Purpose: The "product" table stores data related to all the products offered on the website. This includes their details, availability, and pricing information.

THE TWO ENTITIES WERE COMPLETELY SEPARATED.

1. **Simplicity:** A smaller number of tables means a simpler database schema. It reduces the complexity of the database design and eases the process of development and maintenance.
2. **Data Integrity:** Having dedicated tables for users and products ensures data integrity and reduces the chances of redundant or inconsistent data.
3. **Scalability:** A simpler database schema with fewer tables can be more scalable. As the website grows and the amount of data increases, it becomes easier to manage and optimize database performance with only two tables.

1.3 INCLUDE A LIST OF FEATURES YOU IMPLEMENT AND WHY THESE ADD VALUE TO YOUR PROJECT.

FEATURE 1: SEARCH BAR

Value: The search bar enhances the user experience by enabling users to quickly find specific products they are interested in. Instead of manually browsing through pages of products, users can simply enter keywords or product names in the search bar to get relevant results.

FEATURE 2: ITEM VIEW (WITH PLACEHOLDER FOR PICTURES):

!10 Image in placement was created and was originally planned to replace placeholder

Value: The item view provides users with detailed information about a specific product when they click on it. This feature allows users to make informed decisions about whether to purchase the product or not. While placeholder images may not show the actual product pictures, they still serve an essential purpose by visually representing the products.

Feature 3: Interactive Feedback: By providing interactive feedback during the login process (e.g., displaying error messages for incorrect credentials or successful login messages), users are kept informed about the status of their actions. This improves user satisfaction and helps prevent confusion during the login process.

FEATURE 4: ADMIN MANAGEMENT TO ADD/REMOVE PRODUCTS

Value: The admin management system empowers the website administrators to have control over the products listed on the website. They can easily add new products, update existing ones, or remove outdated or out-of-stock items.

1.4 INCLUDE AN INSTALLATION MANUAL, HOW TO SET UP (DATABASE, FOLDER STRUCTURE, WEB ALIAS, ETC).

See UserManual.PDF

1.5 INCLUDE A LIST WITH SPECIFIC EXAMPLES OF HOW THE TECHNICAL CONCEPTS WERE IMPLEMENTED (SEE REQUIREMENTS)

4. Entities Stored in a Database using PDO:
 - a. Completed: The application successfully stores entities in a database and utilizes PDO (PHP Data Objects) to interact with the database.
 - b. See inc\Utility\GroceryProductsDAO.class.php & inc\Utility\UserDAO.class.php
5. CRUD Functionality for Each Entity:
 - a. Completed: The application fully supports CRUD operations (Create, Read, Update, Delete) for each entity, providing a comprehensive web application experience with complete views for adding, editing, and displaying records.
 - b. See DashBoard.php, EditProduct.php, AddProduct.php
6. Multiple HTML Forms and Pages:
 - a. Completed: The application includes multiple HTML forms and pages, enhancing user interaction and navigation throughout the website.
 - b. See DashBoard.php, index.php, ItemSearchResults.php
7. Search and Lookup Functionality:
 - a. Completed: Users can search and look up records from the database, improving data accessibility and user experience.
 - b. See ItemSearchResults.php
8. Statistics Display for At Least One Entity:
 - a. Completed: The application showcases statistics for at least one entity, offering valuable insights to users.
 - b. See inc\Entities\Page.class.php
9. Input Validation and Error Handling:

- a. Completed: The application thoroughly validates all user inputs and appropriately handles events, providing a seamless and error-free user experience with clear English prompts.
 - b. See inc\Entities\login_form.php
- 10. User-Friendly and Visually Pleasing Design:
 - a. Completed: The application features an easy-to-use and visually appealing design with professionally crafted layouts, text, page elements, and user input elements.
 - b. See DashBoard.php, inc\Entities\Page.class.php
 - c. There is an error when using the css script for login_form.php and thus it was replaced with a place holder style
- 11. User Login with Encrypted Credentials:
 - a. Completed: Users can securely log in to the application, and their login credentials are encrypted to safeguard their information. The application effectively manages user sessions for logging in and out.
 - b. See inc\Entities\login_form.php
- 12. Additional Technologies and Topics Integration:
 - a. Completed: The project stands out by successfully integrating new technologies and topics not covered in the class, enriching the overall functionality and user experience.
 - b. See DashBoard.php and ItemSearchResults.php
- 13. Rewarded Features Beyond Class Coverage:
 - a. Completed: The application includes additional features that go beyond what was covered in the class, demonstrating innovation and creativity, which will be rewarded accordingly.
 - b. See DashBoard.php, which implement class UserDao to DashBoard Template

1.6 INCLUDE A LIST OF THE TECHNOLOGIES IMPLEMENTED AS AN EXTENSION BEYOND WHAT WAS COVERED IN THE CLASSROOM

SEARCH BAR:

Benefit: The HTML store template includes a prominent search bar at the top of the page. This allows users to enter keywords, product names, or any other relevant information to search for specific products they want to purchase.

How it works: While we both user and grocery DAO is known, we make the template with search bar fit GroceryProduct.class and thus allow us to do search by cat and sort by cat

INTERACTIVE FEEDBACK:

Benefit: The HTML store template includes a login system that allows users to create accounts, log in, and access personalized features.

How it works: A new interactive login system is used with slightly better interaction than echo “@ need to be in place for email”. This is only made possible using css and js script from the template, despite issue of changing the store front and header causing the other css to be not functional

ADMIN MANAGEMENT TO ADD/REMOVE PRODUCTS:

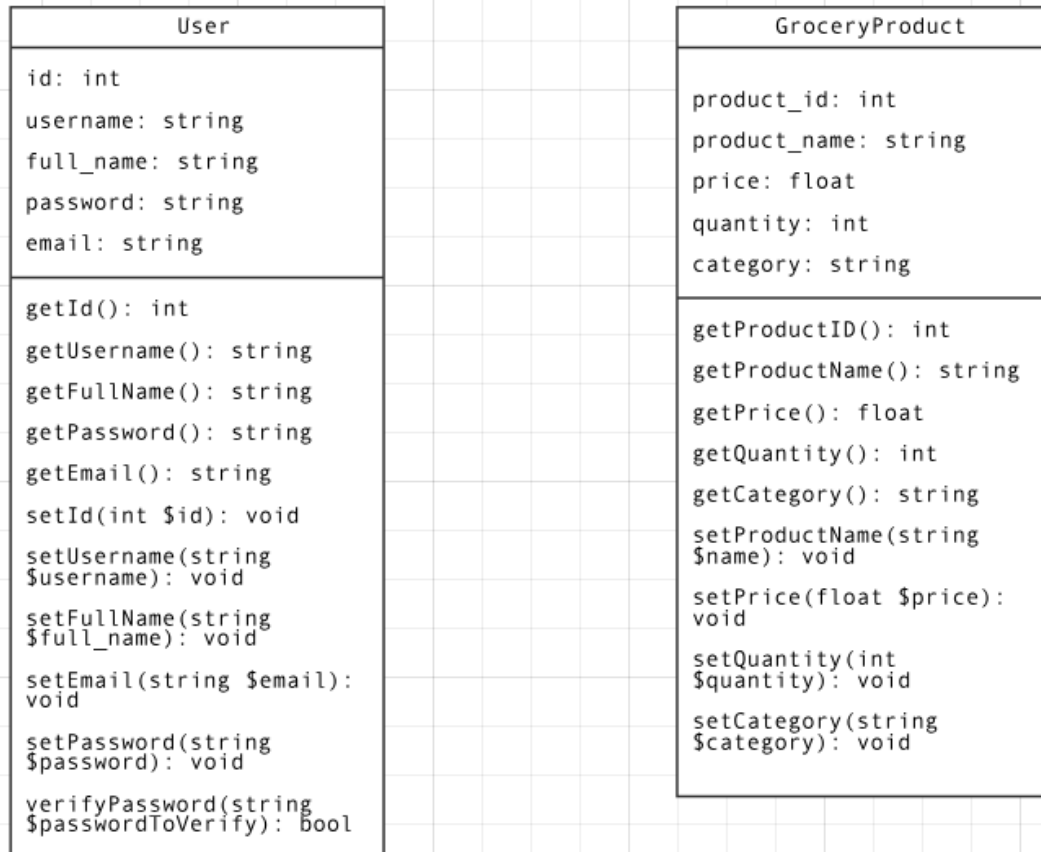
Benefit: The HTML store template includes an admin dashboard that allows website administrators to manage products.

How it works: Through the admin dashboard, the User.class and GroceryProduct.class is better displayed using an existing dashboard, but we had removed edit user function and allow CRUD on GroceryProduct.

APPENDIX 1: CLASS DIAGRAM

Specify the entities, the PK and FK and cardinalities. You can use tools like Software Ideas Modeller or any other

UML drawing tools like lucidchart.com to create your model



APPENDIX 2: MEETING MINUTES

- Attendance (who is present for your meeting)

Both Dill and Ed

- What was done since the last time the group met (per person)?

There is only 1 official meeting on 1330 on 27 JUL 2023, the rest of the communication take place over WhatsApp

- What is working well?

Efficiency of the discussion, the whole discussion took 50 minutes and all designs were confirmed then

- What is not working well?

The change in design and reduced refining time, luckily, we use worst case plan and simplify the system

- What will be done before the next time the team meets?

See 4. To do & milestone

Course: CSIS3280-001 BACKEND WEB DEVELOPMENT

Logistics:

Links:

Latest meeting note:

https://docs.google.com/document/d/1CPxe6M_CG1k4EhlHPL1iZhRxJfs_5yUqt5mI81Nvvc/edit?usp=sharing

Google Drive:

https://drive.google.com/drive/folders/1idI_rNAA2-JcdJI2lJoRg3FRLYv7vzqO?usp=sharing

GitHub:

https://github.com/WCEdison/CSIS3280_PHP_Project.git

Personal info:

Leader: Manj, Dilraj 300331319

Member: Edison CHAN 300370372

1. Reconfirm key design feature

- 1. Format: Store front*
- 2. Major tech decision*
 - a. Use Laravel*

- b. *Use Github to publish webpage*
 - c. *Find Food/Grocery API*
 - d. *Use local database*
- 3. *Design direction*
 - a. *Purpose:*
 - i. *Grocery Store*
 - b. *Features:*
 - i. *Login -> As manager*
 - ii. *Customer*
 - 1. *View*
 - 2. *Filter (Basically Sql select, use desc/asec, sort by)*
 - iii. *Management (use from last demo)*
 - 1. *Add*
 - 2. *Edit*
 - 3. *Delete*
 - c. *Database*
 - i. *2 x Unrelated Tables*
 - ii. *Admin login*
 - iii. *Product*
- 4. *Question to ask: How much weight is this project and when is the deadline*
 - a. *If high then we add more feature (Might expand if scope expands)*
 - b. *keep current feature + grocery api + login admin + database editing for admin*
- 2. *Logistics: How to communicate, GitHub project and publishing*
 - 1. *WhatsApp to do normal communication*
 - a. *Error*
 - b. *To dos*
 - 2. *Teams for bigger meeting → On demand only (one more handing it in)*
 - 3. *GitHub + GitHub desktop*
 - 4. *Use [Visio] for Class Diagram*
- 3. *Task to be done (Same as job distribution mentioned above)*
- 4. *To do & milestone*
 - 1. *Set github*
 - 2. *Find a Grocery/Food API*
 - 3. *Learning how to publish webpage GitHub*
 - 4. *Set up a database design*
 - 5. *Set up database*
 - 6. *Set up PDO and DAO*
 - 7. *Set up validation*
 - 8. *Worst Case: Switch to CSS + HTML (if laravel doesn't work) + Switch to React*

APPENDIX 3: REFERENCES

Template from Code Project, Licenses: Apache License Version 2.0, January 2004

Codeprojects. “Online Shopping System in PHP with Source Code.” *Source Code & Projects*, 13 June 2023, code-projects.org/online-shopping-system-in-php-with-source-code/.

Images from Wikimedia Commons

Abhijit Tembhekar from Mumbai, India, CC BY 2.0

<<https://creativecommons.org/licenses/by/2.0/>>, via Wikimedia Commons

DenesFeri, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

Dmitry Makeev, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

Filo gèn', CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

Forest & Kim Starr, CC BY 3.0 US <<https://creativecommons.org/licenses/by/3.0/us/deed.en>>, via Wikimedia Commons

Forest & Kim Starr, CC BY 3.0 US <<https://creativecommons.org/licenses/by/3.0/us/deed.en>>, via Wikimedia Commons

Gaurav Dhawaj Khadka, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

Ivar Leidus, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

Jacek Halicki, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

Macgills2, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

Mashkawat.ahsan, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

Ravi Dwivedi, CC BY-SA 4.0 <<https://creativecommons.org/licenses/by-sa/4.0/>>, via Wikimedia Commons

Softeis, CC BY-SA 3.0 <<http://creativecommons.org/licenses/by-sa/3.0/>>, via Wikimedia Commons

Véronique PAGNIER, CC BY-SA 3.0 <<https://creativecommons.org/licenses/by-sa/3.0/>>, via Wikimedia Commons