Note: Combined deadline with Phase 3

PHASE 2B: DATA PRESENTATION & MANAGEMENT (DEADLINE: 9 March 2025) (SUBTOTAL: 18')

In this phase, you will implement the core functions of the website with mainly Node and SQL.

1. SQL: Create a database with the following structures ______/ 1'
   - A table for *categories*
     - Required columns: *catid (primary key), name*
     - Data: at least 2 categories of your choice
   - A table for *products*
     - Required columns: *pid (primary key), catid, name, price, description*
     - Data: at least 2 products for each category
2. HTML, Node** & SQL: Create an *admin panel* *[+Backend functions]*
   - Design several HTML forms to manage* *products* in DB ______/ 2'
     - Dropdown menu to select *catid* according to its *name*
     - Input fields for inputting *name*, *price*
     - Textarea for inputting *description*
     - ^ File field for uploading a product image (format: jpg/gif/png, size: <=10MB)
   - Design several HTML forms to manage* *categories* in DB ______/ 2'
   - Submitting the Form to the backend server API result in a DB update. ______/ 3'
     - (part of Phase 4 Requirement) Try to apply input validation

* In terms of management, it includes the capabilities of insert, update, and delete products

^ For the file uploaded, store it with its name based on the unique ID(or other reasonable ways)

______/ 1'

3. HTML, Node**, SQL: Update the *main page* created in Phase 1
   - Populate the *category list* from DB ______/ 1'
     - It can be server-rendered or updated on client-side with Javascript
   - Based on the category picked by user, populate the corresponding *product list* from DB ______/ 3'
     - e.g., the *catid=[x]* is reflected as a query string in the URL (or other method)
4. HTML, Node** & SQL: Update the *product details page* created in Phase 1 ______/ 2'
   - Display the details of a product according to its DB record
5. Supporting automatic image resizing for product images ________/3'
   - When a large image is uploaded, the server will resize it (to a fixed, reasonable resolution) and show a thumbnail image. [e.g., two image files with different names for a product]
   - On the main page, display thumbnails. In the product description page, display the larger image.

**: Other backend languages accepted

Phase 3: AJAX Shopping List (Deadline: 9 March 2025) (Subtotal: 10')

In this phase, you will implement the shopping list which allows users to shop around your products. This phase is designed to let you practice Javascript programming.

1. JS: Dynamically update# the *shopping list* (to be covered in tutorial)
   - When the *addToCart* button of a product is clicked, add it to the shopping list ______/ 1'
     - Adding the same product twice will display only one row of record
   - Once a product is added,
     - Users are allowed to update its *quantity* and delete it with a number input, or two buttons for increment and decrement ______/ 1'
     - Store its *pid* and *quantity* in the browser's localStorage ______/ 2'
     - Get the *name* and *price* over AJAX (with *pid* as input) ______/ 3'
     - Calculate and display the total amount at the client-side ______/ 1'
   - Once the page is reloaded, the *shopping list* is restored ______/ 2'
     - Page reloads when users browse another category or visit the product detail page
     - Populate and retrieve the stored products from the localStorage
   - [Optional] Try to adopt an OOP design for the shopping cart (and cart item).

#The whole process of *shopping list* management must be done without a page load