Refactor Java Code, CSS and HTML

Why Refactor?

1. Remove Duplicate Code
2. Code is better designed
3. Improve application performance
4. Easy to maintain in future

Problems with existing Java Code

- Inefficiently use of EntityManagerFactory and EntityManager
- Duplication Code that stores a List<Category> collection as a Request attribute in the front end

Refactor Solution for Java Code

- Update JpaDAO class:
  + Allow only one instance of EntityManagerFactory across the application
  + Create and Close EntityManager in each method
- Delete BaseServlet Class
- Create new Java Filter Servlet (this filter retrieves the List of Category objects from the DB)

Problems with existing CSS + HTML:

- Too much style information is mixed in HTML
- Solution:
  Move style information into a CSS file to increase the separation between content (HTML) and format(CSS)

Task List:

- Debug the application
- Refactor Java Code
- Test all functionalities
- Refactor CSS + HTML + Test Code
- Code a Utility Class

* Experience:

- Debug the Application’s Front/Back-end
- Understand EntityManagerFactory is created each servlet class is invoked
  -> Bad Practice -> Should only one instance of EntityManagerFactory to improve performance.