

ASSIGNMENT 2

CS 345-346

DEAL FINDER APPLICATION

Group 4

Anjali Godara - 180101008

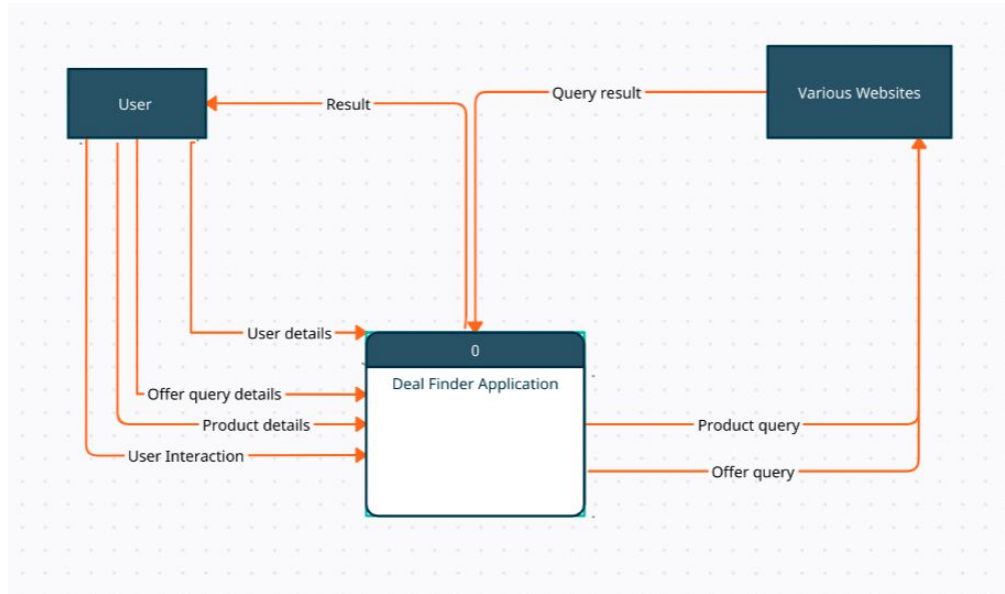
Kotkar Anket - 180101037

Niharika Bhamera - 180101048

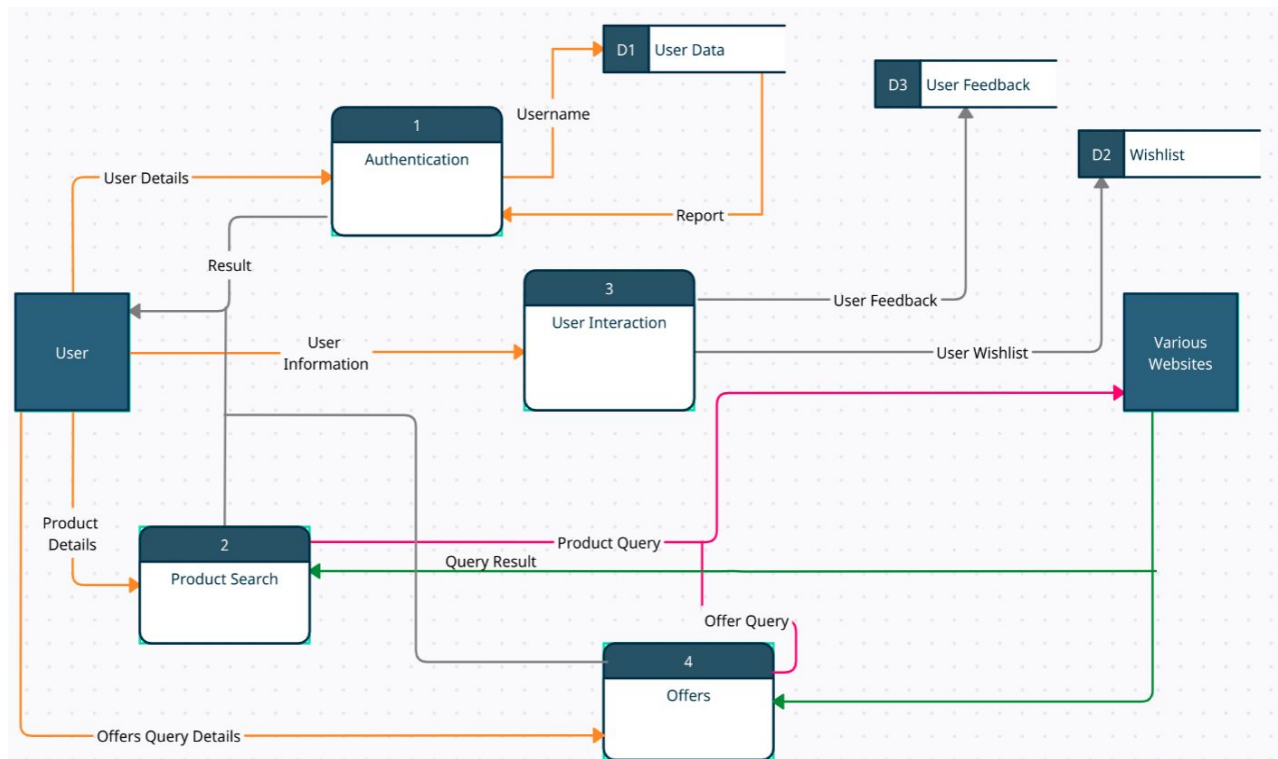
Ritwik Ganguly - 180101067

Data Flow Diagrams

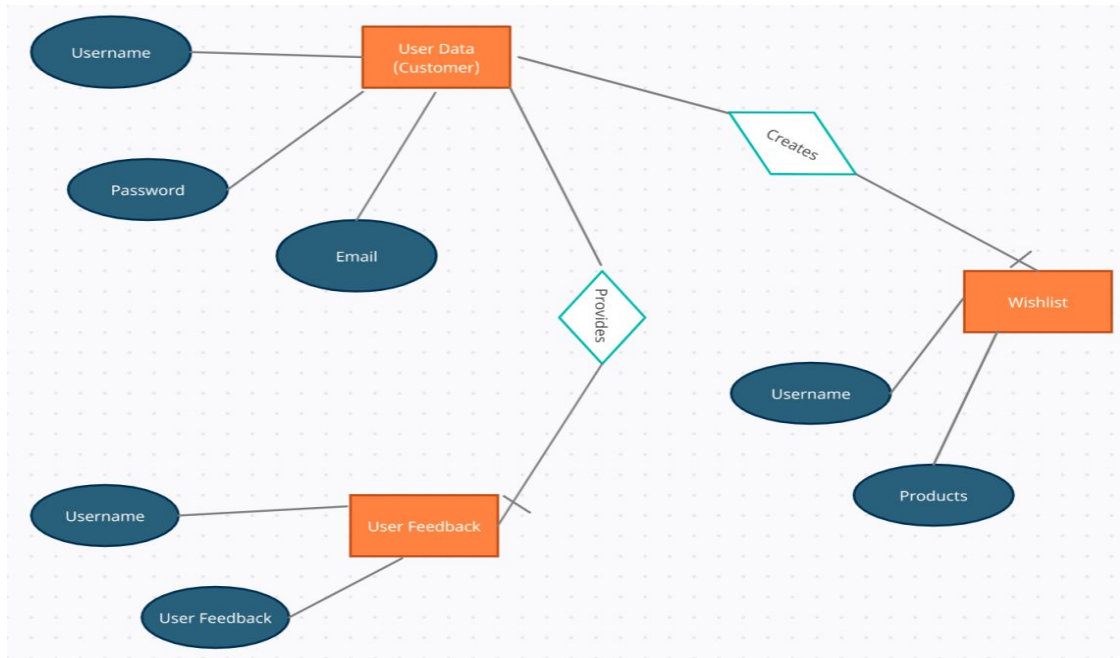
Level 0



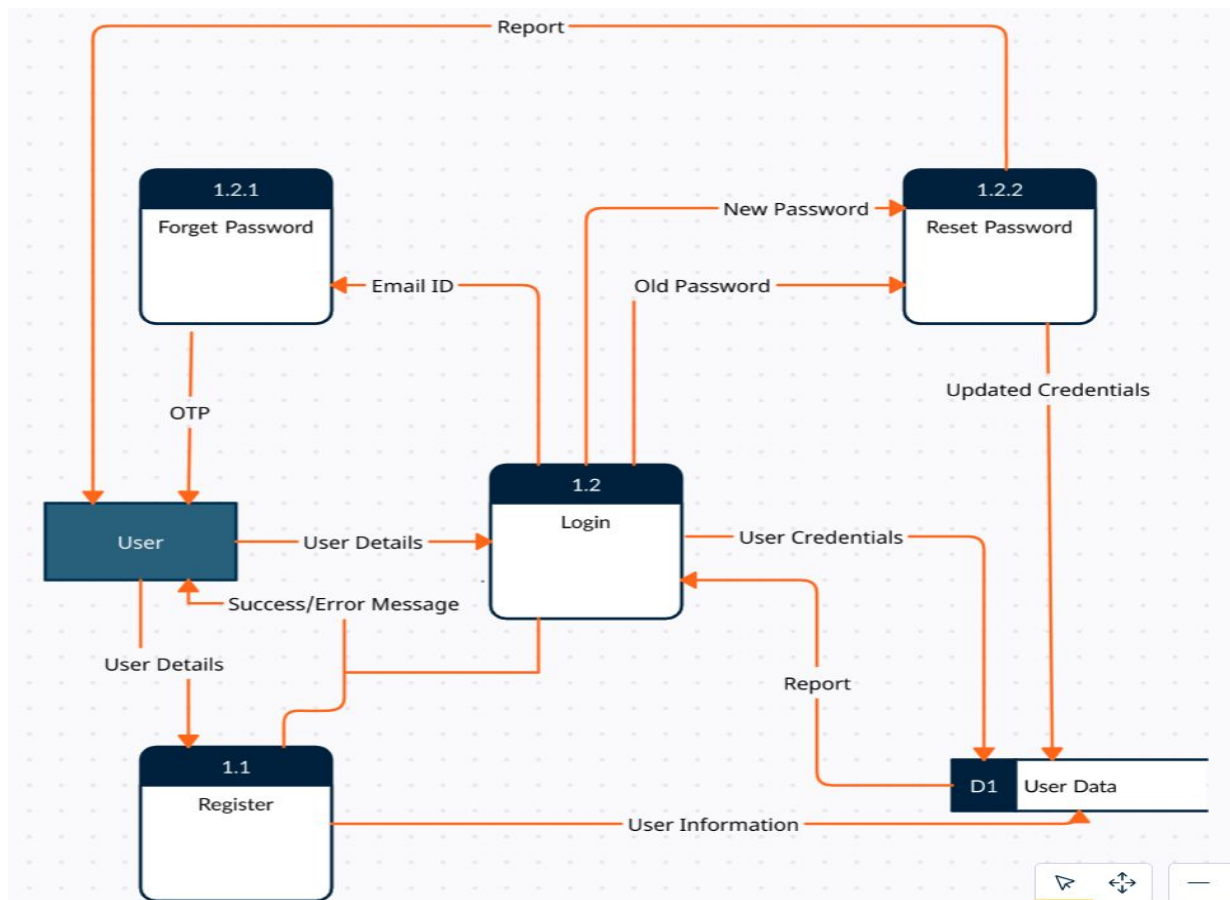
Level 1



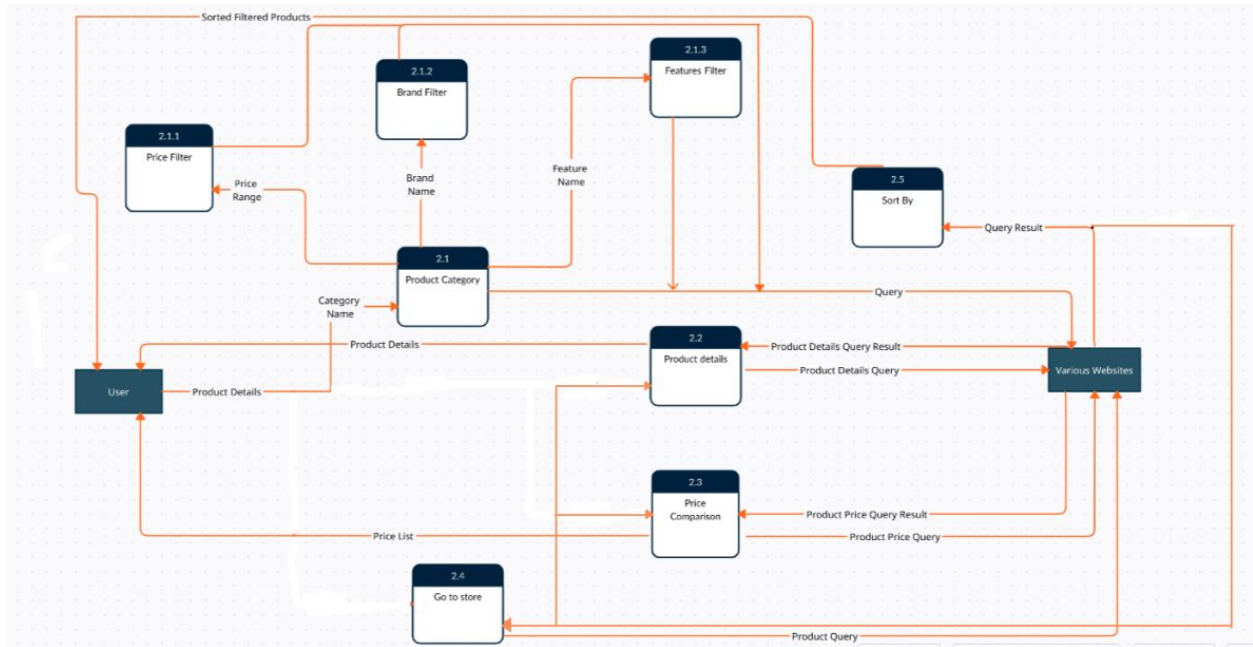
ERD



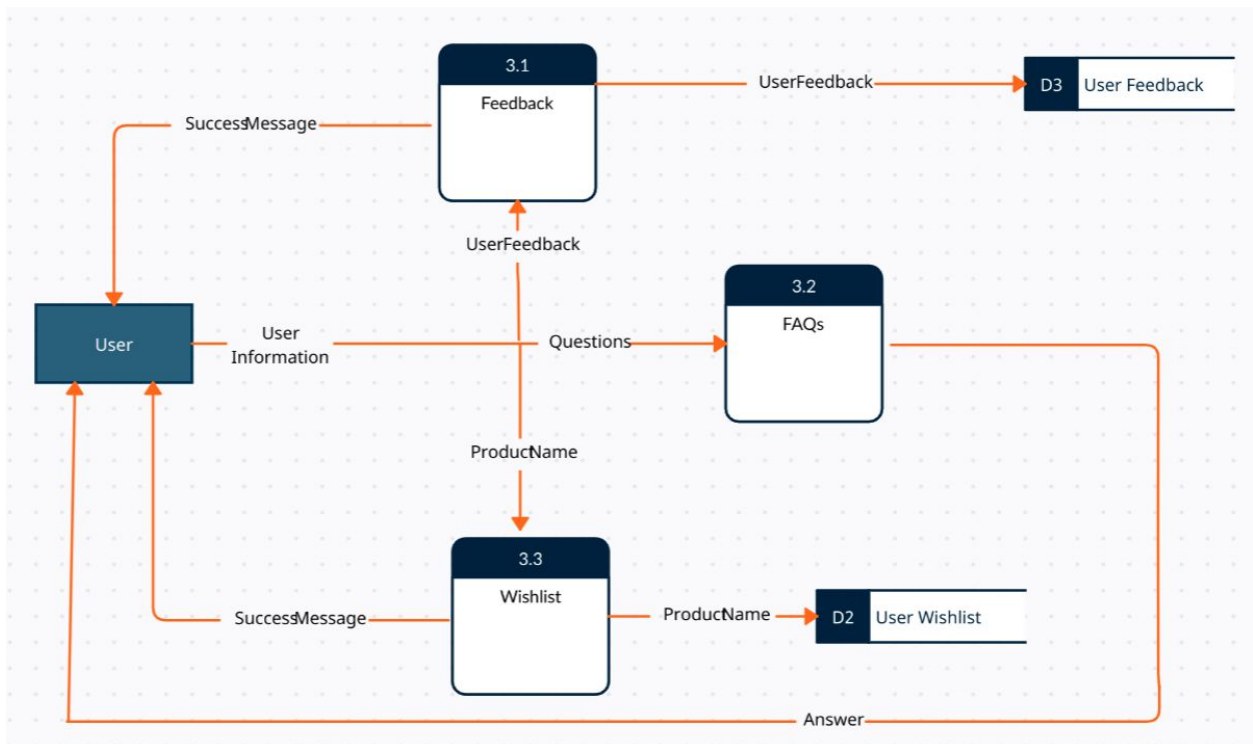
Process 1 Level 2



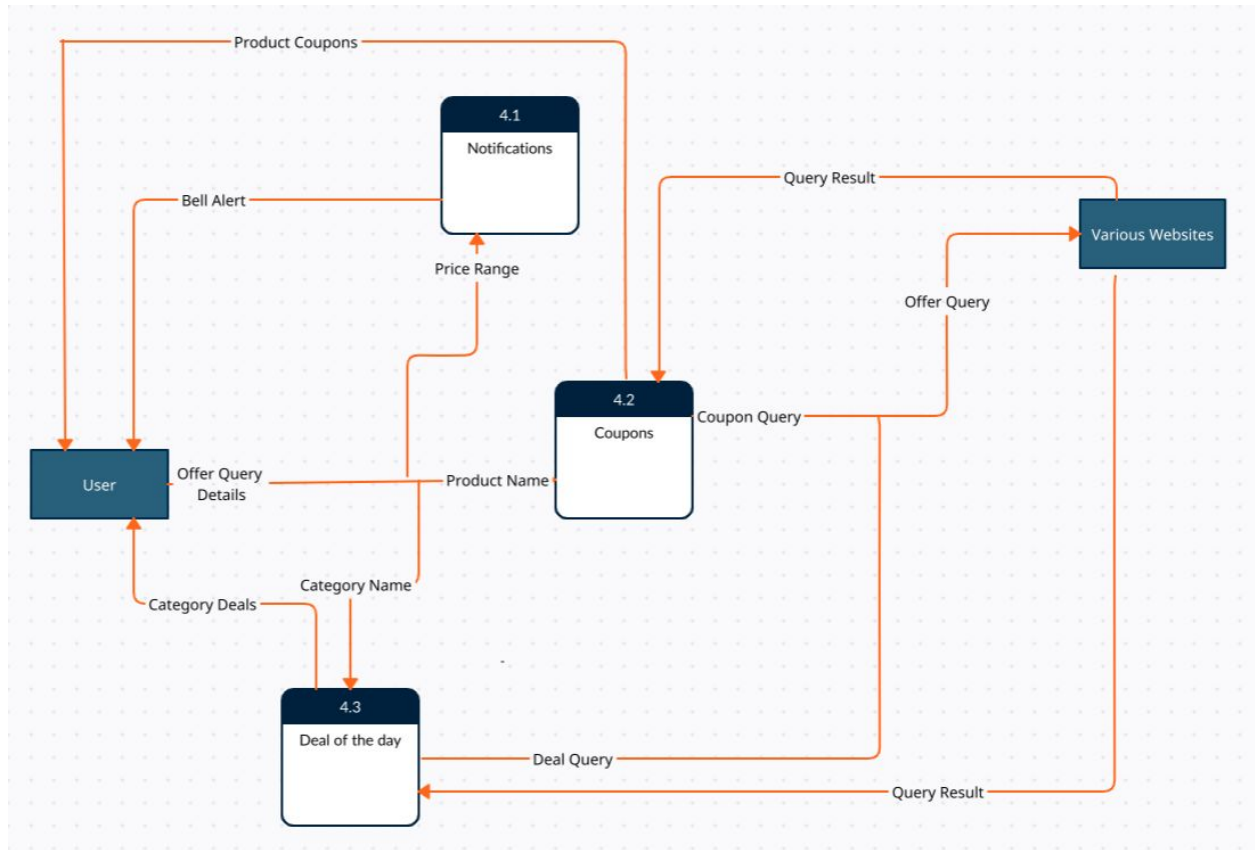
Process 2 Level 2



Process 3 Level 2



Process 4 Level 2



DESIGN JUSTIFICATION

Modularisation is the process of dividing a software system into multiple independent modules where each module works independently. Highly modularized code is preferred for good software design. Measuring the degree of interdependence between modules is called coupling, and low coupling is desirable. Measuring the degree to which a module's elements are functionally related is called cohesion, and high cohesion is desirable. In our 'Deal Finder Application' design, we have tried to implement modularisation in the following way.

The cohesion of different modules in our design is as described below:

1. Module 1 - Authentication

In this module, there are two sub-modules, the 'Login' and the 'Registration'. They both deal with user account authorisation and authentication. They both perform sequential tasks such as taking the user credentials as input for login purposes while taking an additional input of email-Id for registration purposes. These inputs are then sent to the database, 'User Data', for verification and store, login and registration, respectively. If an input is found and verified, the user is permitted to log in, and a flag 'is_logged_in' is updated to reflect the same that leads to updating the user interface. Otherwise, an error message is generated. Hence in this module, **Sequential Cohesion** can be seen.

2. Module 2 - Product Search

In this module, under the submodule 'Product Category', there are three further sub-levels, namely 'Price Filters', 'Brand Filters' and 'Feature Filters'. This submodule fetches the list of products from various online shopping platforms. The list can be further modified to display selected products that meet the filter functions' criteria. 'Product details', 'Price comparison', 'Go to store' and 'Sort By' submodules use this filtered list as their input. These submodules generate their result by referring to the list and giving out details about products, different prices for the same product, redirecting to the website from where a product is being displayed and sorting the output based on different factors. So overall, all submodules either modify or refer to the same list of queried category products. Hence in this module, **Communication Cohesion** can be seen.

3. Module 3 - User Interaction

This module contains three sub-levels, namely, 'Feedback', 'Wishlist' and 'FAQ'. These are related to the user-created product-preference list, their feedback about it and their questions/answers. All three sub-levels perform tasks of updating data concerning user customisation, improvement suggestions and questions/responses, and all performing user engagement tasks with the platform. Hence in this module, **Logical Cohesion** can be seen.

4. Module 4 - Offers

In this module, the function 'Notifications' alerts the user about price changes. A user can set a desired or expected price/price range on a product of their choice. Whenever the target price is achieved, the user is notified about it through bell alerts. The function 'Coupons' shows available coupons for specific product queries. Similarly, 'Deal of the day' finds special deals for a specified product category. All these functions perform similar tasks of notifying or searching for price variations/changes.

So, in this module, **Logical Cohesion** can be seen.

The coupling between different modules in our design is as described below:

1. Data Coupling

In data coupling, modules communicate using data. In our design, no modules communicate through data. So data coupling has been reduced to zero.

2. Control coupling

In control coupling, output or flags from one module set the flow of instructions of other modules. The authentication module sets the flag 'is_logged_in' to true. This flag is used by module 'User Interaction' as all the modules' functions are for user customisation or user engagement. Hence a logged-in user is a prerequisite. Thus there is control coupling between 'Authentication' and 'User Interaction'.

3. Content Coupling

In our design, no two modules share code between them. So there is no content coupling in the design. Hence content coupling has been reduced to zero.

USABILITY DOCUMENT

Our proposed design has been formulated keeping in mind the usability requirements in light of the **Eight Golden Rules of Schneidermann**.

1. Strive for Consistency:

- a. Internal Consistency: Our proposed design will maintain a uniform colour consistency so that it is easier for users to keep track of the functionalities of each button/link.
- b. External/Environmental Consistency: To facilitate user interaction, all functionalities of our system will cater to the established notion in the society. For example, the close buttons used are red in color and are placed at the top right corner, which is useful for users to keep track.

2. Design for universal usability:

- a. For first time novice users, our system will adopt a step by step approach where the users will be able to search for a product, check the prices and buy it right away without logging in.
- b. For expert users, additional features like keeping track of previously viewed items in wishlist and notifications about price changes will help in improving overall search experience.
- c. Thus, our system is designed for all kinds of users, keeping in mind universal usability requirements.

3. Offer Informative Feedback:

- a. For a user buying a product, we can keep a progress bar to take him through all the necessary steps like searching, adding to wishlist and then redirecting to the correct website once the desired product is found.
- b. We will also change the button structure and color for buttons like “Add to wishlist” once they are clicked to provide feedback to the user that the specific task is completed.

4. Design dialogues to yield closure:

- a. The entire process of deal comparison has been properly broken down into modules for ease of user understanding.
- b. In the beginning, the user will have to sign up/sign in for using the website.
- c. In the middle, they can search for their desired products by using appropriate filters and sorting features.
- d. In the end, once they have finalized the product of their choice, they can either add the item to their wishlist to keep track of price changes or go to the respective website for buying the product.
- e. This kind of modularity and grouping helps a lot in online shopping since there are a lot of inherent steps involved in the entire process.

5. Offer error prevention and simple error handling:
 - a. None of the conflicting functionality buttons/links will be kept close together so as to prevent inadvertent errors from the users. For example, the close button will be kept farther from the add to wishlist button or sign in button.
 - b. While logging in or registering or adding products to wishlist, if any error occurs, the corresponding intimation messages will be short and precise for ease of user understanding.
6. Permit easy reversal of actions:
 - a. For ease of reversal of error-inducing actions, we will use cookies on our website to keep track of the last correct state of the user. This will help in preserving wishlist even if the website is mistakenly closed.
 - b. We will also keep the undo option to revert back to the previous wishlist if someone mistakenly deleted an item from the wishlist.
7. Keep users in control:
 - a. The entire website is designed keeping in mind the user's interests so that the user feels in control throughout.
8. Reduce short term memory load:
 - a. Our website structure is such that other than the user credentials, nothing else needs to be remembered by the user.
 - b. To reduce the things that the user needs to remember, we have devised a notification system where any changes in price of a product will be notified to the user without him or her needing to remember every time he or she logs in.