

The project will contain 2 pages:

- Main page: It will contain a form with a combo box and a submit button. All the "asin" (product id) will be loaded in the combo box as dynamic choices fetched from the json. The submit button will forward us to:
- /product/<asin>: First add a back_to_home link. Then, the page will show all comments, sorted by helpful votes (descending) in separate divs. Every div will have all the data contained in the comment. At the bottom of the page, add a new form to add a new comment. It will have a field for every possible column of the comment that can be filled by the reviewer. Keep in mind that the date will be set automatically and the helpful/unhelpful votes will be 0. All fields have to be validated depending on the type of data. For example, overall score must be a number within range, comments can't be empty but also need a length limit, etc. After clicking on "add comment", the user will see all comments including the new one.

Constraints:

1. Dynamic options must be loaded with a for loop over the dictionary.
2. Product ids can't be duplicated in the combo box.
3. You need to prevent a user to input 2 comments for the same product in the same session. To do so try to keep track in the "session" variable of the list of products of the products commented. To prevent it from adding a new one, whenever you detect a user listing all comments from an already reviewed product, don't show the "add comment" form at all, showing a text at the bottom saying something like "product already commented".
4. Flash messages when interacting with the "add comment" form. If something goes wrong, show a red dismissable alert with the validation errors. If it all goes well, show a green dismissable alert with a message similar to "comment added successfully".
5. The template will get all comments already filtered and sorted.
6. Model/view/template structure must be enforced.