|  |  |
| --- | --- |
| 1. Visibility of system status | In this web application, I have used alert system to inform user about what is happening in the background after doing a certain action. If the task completed, then notify user with success alert. Or if the task failed, then notify user with the error and inform user to take necessary actions. Eg. Login to buy items. |
| 1. Match between system and the real world | Since this is a spare part selling website, the potential users of this system are aware of the terms and language used in vehicle and maintenance industry. So the words/phrases and commands used in the web system are much similar to that language styles and consists of the words and phrases in that language style. |
| 1. User control and freedom | This web application doesn’t include high user interactive actions. All the simple actions can be performed with minimal user interactions. Eg. Add items to shopping cart. If in a case that the user add an item he/she doesn’t require, then the user can remove that item from the cart at the checkout menu. |
| 1. Error prevention | In the pages with user input fields, different ways used to guide user to input required data of insert data in required pattern. And before sending the data to the backend, an initial corrections are performed in the front end app. Eg. Converting the user input price string in to number/int format before sending to the backend. |
| 1. Recognition rather than recall | Login data are stored as caches in local memory to allow user a smooth interaction in the system without login again and again when visiting the site. And maintain a user profile instead of sessions to allow user to buy and checkout items in different logins. |
| 1. Aesthetic and minimalist design | Used material design pattern with simple shapes and with only 4 main colors. All the background images are used with decreasing the brightness of images to allow calm eye conditions. |
| 1. Help users recognize, diagnose, and recover from errors | Notify user about error inputs before submitting the input forms. And alert users about wrong login/signup attempts. Etc. |

Database Description: The DB used in this system is MongoDB non-relational database hosted in Amazon cloud platform. MongoDB is a non-relational document database that provides support for JSON-like storage. The MongoDB database has a flexible data model that enables the developer to store unstructured data, and it provides full indexing support, and replication with rich and intuitive APIs. The database design of this system consists of two main database schemas. One for store user related data and the other one for store spare parts related data. The database connection between NodeJS-Express server and MongoDB was established by Mongoose. Mongoose is a JavaScript object-oriented programming library that creates a connection between MongoDB and the Express web application framework.