       Paper View       Plain Text View

**GITHUB LINK**: [https://github.com/sjsu-cmpe281/shoppingcart (Links to an external site.) (Links to an external site.)](https://github.com/sjsu-cmpe281/shoppingcart)

*account id*: rakeshdatta (Rakesh Datta)

**WAFFLE LINK:** [https://waffle.io/sjsu-cmpe281/shoppingcart (Links to an external site.) (Links to an external site.)](https://waffle.io/sjsu-cmpe281/shoppingcart)

TASK 1: Refactoring the http server:

Refactored the nodejs based http server to handle both successful and error scenarios. Depending on the url type, methodtype etc. the response could be success (200 OK) or failure(400 Bad Request/405 Method Not allowed).

TASK 2: Mongo module (show product):

Over the mongo client designed last week, a full-fledged mongo module is designed. This module will be exported to the HTTPServer. This module will expose the REST APIs for mongodb CRUD operations. When a http GET request comes, it comes along with a querystring in the messagebody. This message body is extracted and passed to the appropriate mongo query handler. The mongo query handler then uses the query string and fires a mongo query on the db ‘cmpe281’ and collection ‘catalog’. The result from the db is then sent back to the front-end for displaying the product catalog.

This whole thing is tested with ‘curl’ operation and verified that the product catalog is displayed properly.

TASK 3: Addition/Modification facility in product catalog:

Continuation of the above task, the POST/PUT/ request for the product catalog is also handled.

POST scenario will occur when admin wants to add a new product in the catalog.

PUT scenario will occur when user buys a product kept in the shopping cart. In this case the available\_count of the product, in the product catalog, should be modified for data integrity.