**GITHUB LINK**: <https://github.com/sjsu-cmpe281/shoppingcart>

*account id*: rakeshdatta (Rakesh Datta)

**WAFFLE LINK:** <https://waffle.io/sjsu-cmpe281/shoppingcart>

TASK 1: Updating mongo product ‘available\_count’ on product being sold:

When a product is sold, the number of items should be reduced. When a new user comes and checks the product catalog, he should see the updated count. Otherwise the operation on shopping cart database would be based on stale data and this will lead to a data integrity issue.

TASK 2: Providing mongo product ‘available\_count’ to riak, prior to adding it to shopping cart:

Item(s) could be added to shopping cart at any point, however it could be bought at any point later. In the meanwhile, this product may be already sold and the available\_count would have been reduced. So, in order to stop operations to be performed on stale data, riak would first verify if the requested count of items is more than the available count in mongo.If the condition is true, then only the products will be allowed to be sold. This logic is implemented in terms of a http request. Riak will use the rest url to send a request to mongo and mongo will then reply back with the available\_count of the product.

TASK 3: URL parse logic to extract query parameters in REST calls:

When the front-end sends a REST call to the back-end for product catalog CRUD or shopping cart CRUD, nodejs server receives it. The parameters for DB queries and updations are sent in the URL params. URL received is parsed and the query params are deduced, which are then used for the mongo and riak queries.

TASK 4: Setting up EC2 and ELB topology and mongo replication:

3 ec2 instances are created for Primary-Secondary2-Secondary3 architecture. A ELB is created which will receive the HTTP requests from heroku server and route it to the ec2 instances. The 3 ec2 instances are running in the same VPC. A jump box is created in the same VPC which is used to ssh into the ec2 instances, which are in 3 different private subnets.