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

*account id*: rakeshdatta

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

TASK 1:

CRETE NODJS SERVER: Team discussed and decided that mongo and nodejs combines well, hence Java is discarded and  back-end is instead designed in nodejs. This will conform well to the mean stack.

I have designed a nodejs HTTServer that exposes REST APIs to the front-end.. It starts a HTTP server at port 80, listen to HTTP requests.On receiving a request it invokes the correct handler.

This server will be further extended to listen to multiple URLs for handlig requests for both product catalog(mongo) and shopping cart(riak).

TASK 2:

CRESTE NODEJS MONGO CLIENT CODE: To the realize product catalog, a GET request is sent to the httpserver  with a certain url. On receiving this request, I have implemented and invoked a simple handler for the Product Catalog fetch.This creates a mongo Client and executes a find() query in the mongo back-end, creates a HTTP response and sends it back . Testing using curl.

TASK 3:

DEPLOY MONGO REPLICATION (INITIAL STEP)To understand the replication architecture and partition tolerance of mongodb, 3 instances of mongod are started in 3 different ports. Replication is tested.. Partition is created and verified the re-election mechanism. This setup will be further extended to 3 different EC2 instances.