**Output Document**

**Part 1 Caching:**

Print statements in frontend.py (currently commented out in code) validated behavior of the cache. The first queries (not pictured here) all resulted in cache misses due to empty cache. Afterwards, the majority of queries resulted in a cache hit. However, we see here (mid sequence) that there are still some cache misses which are caused by successful buy requests that invalidate that item in the cache until an updated query resets it. Additionally, measured latency (see eval doc) verify the lift in performance provided by the cache.

A picture containing text, plaque

Description automatically generated

Here we see that restock is working when inventory is set to 0. It continuously checks every 10 seconds

Text

Description automatically generated

**Part 2 Replication:**

Order service contains 3 replicas

A screenshot of a computer

Description automatically generated with medium confidence

Here is replica 2 (set to be outdated) and replica 3 (set to be up to date) before coming online

Text

Description automatically generated

Text

Description automatically generated

Here is the outcome after both come online. We see that it synced up to the same data as replica 3.

Text

Description automatically generated

Here is data of replica 2 and 3 respectively after client call. We see that only replica 3 (leader) handles the processing, but propagates the orders to the follower nodes so that at the end, both have consistent data. We also see here that leader election is taking place when frontend comes online

Text

Description automatically generated

Text

Description automatically generated

**Part 3 Fault Tolerance:**

Here is the output for a failure simulation where replica 3 fails after 5 orders. The screenshots are of replica 3, replica 2, and the client respectively. We see that replica2 picks off where replica3 left off, and from client perspective, nothing is detected and everything works as intended.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**Part 4 Testing and Evaluation with Deployment on AWS:**

Output of server running on AWS m5a.large instance (catalog top left, front-end top right, 3 order replicas on the bottom)

Text

Description automatically generated

5 clients running on local machine

Text

Description automatically generated

Test cases all successful

Text

Description automatically generated