Design Document

Table of Content

Backend has been created in Node.js and and the Products and Delivery Agents have been stored in MongoDB database on mLabs.

1.1 Agent.js - Needed for sending the location of the Delivery Agent. Initially all the agents location have been assumed to be stored in the database .

1.2 Viewer.js - Contains the embeded Google Map and the markers for the origin and destination, and shows the route between them

1.3 Agent.HTML - To send the location of the agent.

1.4 Viewer.HTML - For the customer to see the map.

2.1 Insert – Mongo – Contains the JS files for updating the products and the delivery agents on the mLabs server

2.2 Models – Delivery Agent – contains a unique id attribute, availability, and the email-ID of the customer whose product the agent is supposed to deliver

Product – Contains sample data of Products

2.3 Mongoose – contains the mongoose connection

2.4 Routes – for setting up the /products and the /trackingInfo routes.

2.5 server.js - contains the socket logic and listening to the events from the client side and emitting back events.

Logic – The products and the Agents have already been inserted into the database by runnig the JS files in the Insert Mongo directory, and initially all the agents have been marked as available.

Deployment steps:

1) npm install in the server folder to install all the node modules required

2) Server will be up and running on localhost:3000

3) localhost:3000/products - will send the list of products in json format to the front end

And /products will be home page

4) On hitting the buy button, user will be redirected to localhost:3000/trackingInfo with the agent information, and the agent assigned will have the customerEmail attribute in their document updated to the customers email id value, and the agent will have the availability set to False.

5) and On clicking the Track button, user will be redirected to the viewer.html page containing the google map with the agent location and the route.

Assumptions:

1) On going to the viewer.html page, I have sent a Hard Coded email id to the server, (In the case of front end, email id of the customer needs to be sent as a parameter, which will fetch the assigned Delivery Agent only)

Testing Steps:

-node server/server.js

-localhost:3000/products

-localhost:3000/trackingInfo

-localhost:3000/viewer.html