LAB FEEDBACK WEB APP README

Purpose:

The README is designed to help you understand the basic architecture of the app, how to run the app and how to correctly and efficiently utilize the app.

Architecture:

This application is based on MVC architecture and uses client and server side controllers to gather the clients needs and produce a result.

Uses NodeJS, Express, Angular, MongoDB. Check package.json file for complete list of dependencies.

This application has only been tested on Google Chrome and Microsoft Edge. Other browsers may not provide full support.

The order of operations is as follows:

-A logged in user manipulates something in the HTML (user visible webpage)

-The client side controller for the manipulated data assesses what the client needs and sends the request to the client side factory

-The client side factory packages the data correctly in order to be sent to the server

-Before the data leaves the client side, the system checks to ensure the requesting user is still logged in

-If so the data reaches the routing module and is sent to the correct server side controller

-Once at the server controller all the business logic is done. If any communication with the data base is needed it will initiate here

-If database is queried that data is returned to the server side controller

-Data moves back to the client in the reverse order as listed here.

Running the application:

The key file is 'server.js'. You must know where this file is located and navigate through the directories to run this program.

There are multiple ways to run this file. The simplest is typing 'node server.js'. This will start but not monitor the health of the program.

A better option is to use a service like 'nohup' or 'pm2' which will monitor the application and restart it if there are any issues.

Both 'nohup' and 'pm2' can run in the background and will document all output to a file.

Correct use of the application:

Uploading files:

-Currently a hard coded path. Put files in the 'client' directory of the application.

-Excel files must follow a specific format. Changing the format will cause it to not function properly. Please reference the 'dummy' files for correct format.

-Since Excel files are comma separated, it is important that data within a cell is separated with a semicolon (;)

-Any data read in from a file will APPEND to the existing data not REPLACE it. Please use caution to not generate duplicate data.

These are the correct names for each file:

Noida customer visit feedback: NoidaCust.xlsx

Redmond customer visit feedback: RedmondCust.xlsx

Noida lab manager feedback: NoidaLab.xlsx

Redmond lab manager feedback: RedmondLab.xlsx

Purging data:

-This can only be done by a user with administrative rights. This is permanant and will delete all data associated with that model.

-It is best practice to download the correct files and store them safely before purging any data.

Downloading data:

-This can be done by any level user.

-Charts are downloaded from the dashboard by first selecting a start date, then clicking 'Get Charts' then clicking 'Get Report'

-Customer data is downloaded from the 'What customers are saying' page, by selecting a start date, then clicking 'Get Visits'

-Lab manager data is downloaded from the 'View and update Excalibur IDs' page, by selecting a start date, then clicking 'Get Visits'

-Charts are shown for a single year at the most starting from the start date. Only data up to the current month is shown.

-Lab and customer feedback is displayed from the choosen start date up to the current time.

What can be improved on:

-The UI can be greatly improved

-In the JSON chart data for all 'monthly' charts all the values are currently hard coded for this fiscal year. This should be made dynamic, or at least re-hardcoded next year