

CSCI 4140 A1

Dhairy Raval

B00845519

Assignment uploaded to Github:
<https://github.com/dhairyraval/4140-A2.git>

Table of Contents

→ A2-519.pdf:

1. Software & Libraries used
2. ER Model
3. SQL Script & folder description
4. Results & Limitations

1. Software & Libraries used:

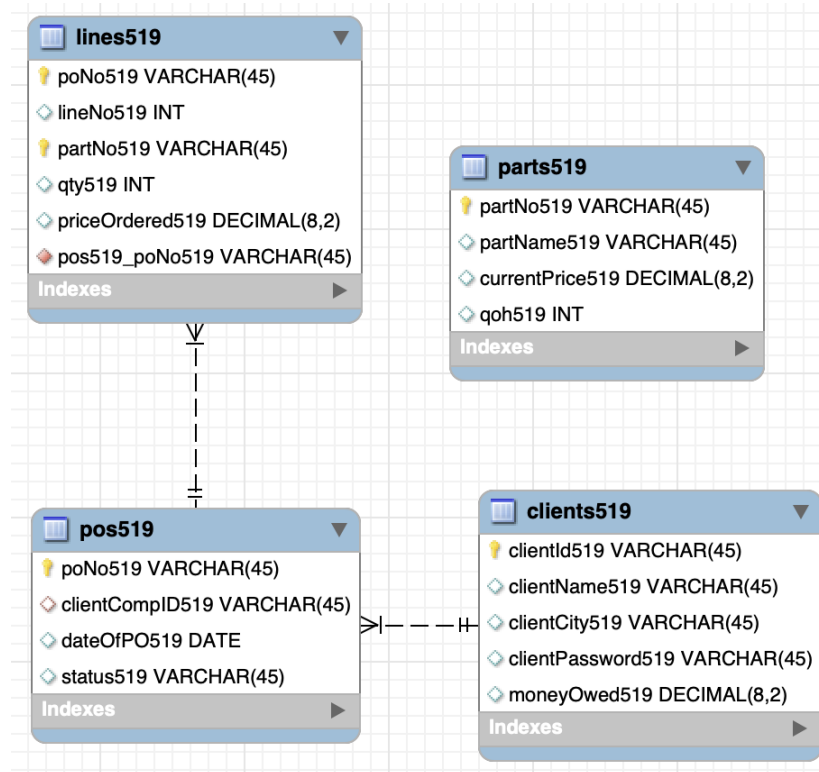
Software:

- MySQL Workbench - database (local instance)
- VS Code Editor - creating the UI files and backend server files
- React - frontend
- Node + Express - backend
- Github - version control and sharing the project w/ TA & Prof
 - URL - <https://github.com/dhairyraval/4140-A2.git>

Libraries:

- body-parser - used to parse the data from frontend
- cors - used w/ Axios to connect to DB
- express - hosting backend
- mysql - SQL library used to connect to DB
- nodemon - used to make development faster for backend

2. ER Model:



3. SQL scripts & folder description:

This section describes the folder structure of the 4140 A2 submission. The folder is uploaded as a zip file along with the index.pdf file to Brightspace and Github.

* NOTE: Github contains a single folder: 4140 A2 submission with all the following info.

4140 A2 Submission

- **4140 A2 Code.zip**
 - This is the code to run the React + Node/Express app to connect with the db
 - To start the front-end run - **npm start** inside the *client* folder
 - To start the back-end run - **npm run devStart** inside the *server* folder

- **table-creation.sql**

script file to create all the required tables

- **insert.sql**

script file to populate all the tables with sample data

- **index.pdf**

The index file (this file)

4. Results & Limitations:

When running the program, the client side (React) runs on localhost:3000 and the server side on localhost:3001. When you submit the form after entering the details, the page will update to look like this:

List Parts for sale ☒

Submit a PO:

PO No:

ClientCompID:

Date:

Status:
☐ Processing
☐ In-Progress
☒ Completed

Submit the Line (Optional)

PO No:

Line No:

Part No:

Quantity:

PriceOrdered - Price of the part:

List POs of Client:

List Lines of PO:

Submit

LIST OF PARTS: [{"partNo519":"1a1","partName519":"screw","currentPrice519":10}, {"partNo519":"2b2","partName519":"bolt","currentPrice519":15}, {"partNo519":"3c3","partName519":"tire","currentPrice519":100}, {"partNo519":"4d4","partName519":"caps","currentPrice519":5}, {"partNo519":"5e5","partName519":"axel","currentPrice519":500}]

POs OF GIVEN CLIENT: [{"poNo519":"po6","clientCompID519":"5a","dateOfPO519":"2022-09-29T03:00:00.000Z","status519":"completed"}, {"poNo519":"po7","clientCompID519":"5a","dateOfPO519":"2022-10-20T03:00:00.000Z","status519":"processing"}]

LINES OF GIVEN PO: [{"poNo519":"po10","lineNo519":15,"partNo519":"1a1","qty519":10,"priceOrdered519":10}]

Limitations:

The program will work as expected if the user enters valid data. However, if the user were to enter invalid data (for example, requesting the Lines of a PO which doesn't exist) the result would simply omit that part (it won't show any errors).