OneStop Computer Part Shop

1. Create your subgroup and project by logging into GitLab using the web link provided and using the “GitLab How-To” web link, and do the following:
2. Create a README file that includes notes describing where in the code to find the changes you made for each of parts C to J. Each note should include the prompt, file name, line number, and change.
3. Customize the HTML user interface for your customer’s application. The user interface should include the shop name, the product names, and the names of the parts

-Added OneStop shop name and changed title mainscreen.html

1. Add an “About” page to the application to describe your chosen customer’s company to web viewers and include navigation to and from the “About” page and the main screen

-Added Controller entry in MainScreenControllerr.java. Added About.html to view templates. Didn’t change the title in Letter C went into the mainscreen.html.

E.  Add a sample inventory appropriate for your chosen store to the application. You should have **five** parts and **five** products in your sample inventory and should not overwrite existing data in the database.

-Added Model information to BoostrapData.java. Set up attributes in the POJO under domain (part.java) for min and max.

F.  Add a “Buy Now” button to your product list. Your “Buy Now” button must meet each of the following parameters:

-Added Controller for buyNow in MainScreenController.java. Added 2 new views failure.html and success.html. Added the Buy Now button in mainscreen.html.

G.  Modify the parts to track maximum and minimum inventory by doing the following:

•  Add additional fields to the part entity for maximum and minimum inventory.  
•  Modify the sample inventory to include the maximum and minimum fields.  
•  Add to the InhousePartForm and OutsourcedPartForm forms additional text inputs for the inventory so the user can set the maximum and minimum values.  
•  Rename the file the persistent storage is saved to.  
•  Modify the code to enforce that the inventory is between or at the minimum and maximum value.

-Added ValidMinandMax.java and MinandMaxParts.java for a custom annotation. Added the custom annotation to the Part.java. Had some spacing errors in BoostrapData.java. Added fields for Max and Min in InhousePart.java and OutsourcedPart.java. Changed some lines in EnufParrtsValidator.java to add a rule that a products part inventory must be between max and min.

H.  Add validation for between or at the maximum and minimum fields. The validation must include the following:

-See G

•  Display error messages for low inventory when adding and updating parts if the inventory is less than the minimum number of parts.  
•  Display error messages for low inventory when adding and updating products lowers the part inventory below the minimum.  
•  Display error messages when adding and updating parts if the inventory is greater than the maximum.

1. Add at least **two** unit tests for the maximum and minimum fields to the PartTest class in the test package.

Added an entry in applications.properties to allow for H2 dialect (ran into issues with the DB). Then added my two unit tests in DemoApplicationTests.java.

J.  Remove the class files for any unused validators in order to clean your code

-Removed DeletePartValidator.java and ValidDeletePart.java. Got rid of the ValidDeletePart.java annotation from Part.java

K.  Demonstrate professional communication in the content and presentation of your submission.