**Smart Evaluator of MRO Supplies Vendibility**

Test Document

Team S1

Matthew Periut, Cohen Miller, Carter Ray, Dawniqueca Steele, Justin Hughes

Date: (4/13/2024)

Department: Software Engineering and Game Development

Course: SWE 4724

Professor: Yan Huang

**1. Testing Strategy**

**1.1 Overall strategy**

*This section should provide details about your unit-, integration-, system-, and regression-testing strategies. In particular, it should discuss which activities you will perform as part of your testing process, and who will perform such activities.*

**Unit Tests:** This phase of testing in the Smart Evaluator of MRO Supplies is directed towards ensuring that specific modules and functions of the software run efficiently. The various units/modules undergoing testing range from both the frontend and backend structures of the software. Below is a comprehensive list of the fundamental units that require testing. These units are also at the functional level.

**Backend Functions**

* **SessionController.js**
  + uploadSpreadsheet()
  + getSessionIDs()
  + getItem()
  + createSession()
  + addItem()
  + getTableFromSessionID()
* **VendibilityController.js**
  + itemVendibility()
* **DataCollection.js**
  + dataCollection()
* **Continuous\_gpt.js**
  + scrapeWebForKeywords()
  + continuousScrape()
* **DataAnalysis.js**
  + dataAnalysis()
  + getProlockCapacity()
  + GetProStockCapacity()
  + GetCoilVendibility()
  + GetCarouselVendibility()

**Frontend Functions**

* Header()
* Modal()
* TableDisplay()
* SessionIDs()

**Integration Testing:** This phase of testing will be directed towards the interactions among the modules or “units” listed above. These interactions are facilitated primarily through external interfaces, the API endpoints between the front and backend components. To test the communication between the various API endpoints, we will test execution using an external http request services such as Postman. During this phase we will also ensure that any front-end components that incorporate the backend endpoints are well tested. The endpoints being tested are listed below.

* POST /upload
* GET /itemVendibility
* GET /getSessionIDs
* GET /getItem
* POST /createSession
* POST /addItem
* GET /getTableFromSessionID

**System Testing:** The final phase of testing will test the system’s ability to deliver functionality in its entirety. This phase will test the deployed system’s functionality from a high level, particularly focusing on fulfilling the non-functional requirements such as performance, speed, usability. Finally, acceptance testing will be performed with the client.

**1.2 Test Selection**

*Here you should discuss how you are going to select your test cases, that is, which black-box and/or white-box techniques you will use. If you plan to use different techniques at different testing levels (e.g., unit and system), you should clarify that.*

**Unit Tests:** Our client company, CribMaster, supplied a comprehensive list of test data that includes information about items, their dimensions, and vendibility information. These items were carefully selected to represent real-world application of our software system. This test data will be used to create black box test cases and verify the results of the data collection and data vendibility functions. Functions not associated with data collection or analysis will be carefully selected to include all potential edge cases. Test cases for vendibility analysis functions such as *calculateLockerVendibility()* and *calculateCoilVendibility()* will be carefully selected to cover all potential control flows.

**Integration Tests:** To test the communication between the various API endpoints, we will focus more heavily on covering potential edge cases, rather than diversity of input data. The communications are straightforward, and there should be fewer integration tests than unit test cases.

**System Tests:** This phase will test the deployed system’s functionality from a high level. System tests will be evaluated by how well they test the following metrics, which were first outlined within the SEMROS requirements specification.

1. **Scalability:**
   1. *The system must be scalable to accommodate future growth in the number of clients, items, and vending solutions without compromising performance.*
2. **Security:**
   1. *The system must adhere to strict data security standards to protect sensitive client information and ensure compliance with relevant regulations.*
3. **Performance:**
   1. *The system must exhibit the capacity and reliability to handle large volumes of data and user interactions.*
   2. *The system must perform vendibility analysis on a high volume of items under a 12-hour period.*
4. **Usability:**
   1. *The system must be user-friendly and intuitive, with clear interfaces and navigation paths to support efficient usage by clients and sales personnel.*
5. **Support and Maintenance:**
   1. *The system must provide comprehensive training and documentation to ensure functionality and usability.*

**1.3 Adequacy Criterion**

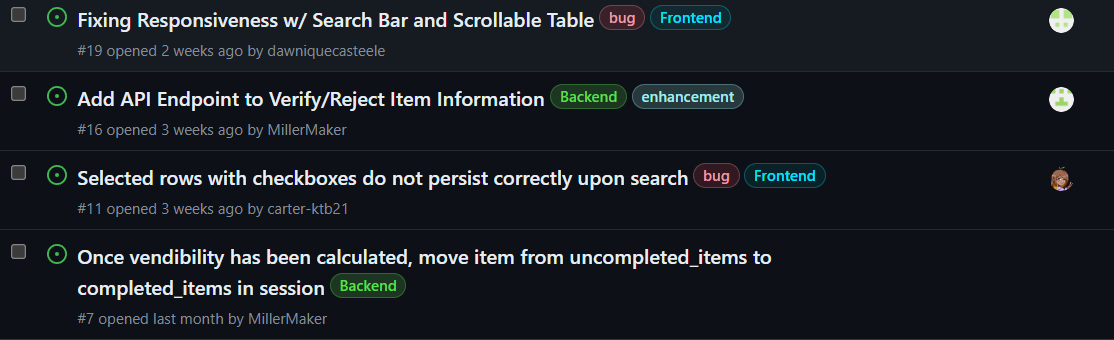
*Define how you are going to assess the quality of your test cases. Typically, this involves some form of functional or structural coverage. If you plan to use different techniques at different testing levels (e.g., unit and system), you should clarify that.*

The quality of our test cases will be determined through the use of functional coverage. This assessment will be used for all levels of the testing phase. Our test cases will cover key features/requirements that are vital to the software performing efficiently.

**1.4 Bug Tracking**

*Describe how bugs and enhancement requests will be tracked.*

Bugs and enhancement requests are being tracked using GitHub issues tracker. Open issues are labeled with their respective location and type, and issues are self-assigned by a member of the team.



**1.5 Technology and Tools**

*Describe any (automated) testing tool you intend to use for your project (e.g., JUnit, Selenium, Zap).*

**2. Test Cases**

*This section should be the core of this document. You should provide a table of test cases, one per row. For each test case, the table should provide its purpose, the steps necessary to perform the test, the expected result, the actual result (to be filled later), pass/fail information (to be filled later), and any additional information you think is relevant.*

**Unit Tests for SessionController.js**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***uploadSpreadsheet()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Input*** | *req.file* | *JSON response with session ID and uncompleted\_items array* |  |  |
| ***Invalid Input (No File Uploaded)*** | *req.file* | *Error response with status code 500 and message* |  |  |
| ***Error Loading Excel File*** | *req.file* | *Error response with status code 500 and message* |  |  |
| ***Error Connecting to Database*** | *req.file* | *Error response with status code 500 and message* |  |  |
| ***Successful Upload*** | *req.file* |  |  |  |
| ***Valid Input (with Session ID)*** | *req.query* | *JSON response with item data* |  |  |
| ***Invalid Session ID*** | *req.query* | *Error response with status code 500 and message* |  |  |
| ***Invalid Item ID*** | *req.query* | *Error response with status code 500 and message* |  |  |
| ***Successful Query*** | *req.query* | *JSON response with item data* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***getSessionIDs()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Input*** | *valid req* | *JSON response with session IDs, completedItems, and uncompletedItems arrays* |  |  |
| ***Error Connecting To Database*** | *invalid req* | *Error response with status code 500 and message* |  |  |
| ***Error Fetching Session IDs*** | *invalid req* | *Error response with status code 500 and message* |  |  |
| ***Successful Query*** | *valid req* | *JSON response with sessionIDs, completedItems, and uncompletedItems arrays* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***getItem()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Input*** | *valid req* | *JSON response with item data* |  |  |
| ***Invalid Session ID*** | *req.query with invalid Session ID* | *Error response with status code 500 and message* |  |  |
| ***Invalid Item ID*** | *req.query with invalid Item ID* | *Error response with status code 500 and message* |  |  |
| ***Error Connecting to Database*** | *invalid req* | *Error response with status code 500 and message* |  |  |
| ***Session Doesn’t Contain Item ID*** | *req.query with no Item ID* | *Error response with status code and message indicating ‘that item doesn’t exist in the requested session’* |  |  |
| ***Successful Query*** | *valid req.query* | *JSON response with item data* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***createSession()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Input*** | *valid req* | *JSON response with session ID* |  |  |
| ***Error connecting to Database*** | *invalid req* | *Error response with status code 500 and message* |  |  |
| ***Error Creating Session*** | *invalid req* | *Error response with status code 500 and message* |  |  |
| ***Successful Session Creation*** | *valid req* | *JSON response with session ID* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***addItem()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Input*** | *valid req* | *JSON response with success message* |  |  |
| ***Invalid Session ID*** | *req.body with invalid Session ID* | *Error response with status code 404 and message indicating ‘Session not found’.* |  |  |
| ***Error Connecting to Database*** | *invalid req* | *Error response with status code 500 ane message* |  |  |
| ***Error Adding Item to Session*** | *req.body with invalid item* | *Error response with status code 500 and message* |  |  |
| ***Successful Item Addition to Session*** | *valid req* | *JSON response with success message* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***getTableFromSesssionID()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Input*** | *valid req* | *JSON response with an array of items from the specified session ID* |  |  |
| ***Invalid Session ID*** | *invalid req.query.sessionID* | *Error response with status code 404 and message indicating ‘Session not found’* |  |  |
| ***Error Connecting to Database*** | *invalid req* | *Error response with status code 500 and message* |  |  |
| ***Session Found with No Items*** | *valid req, req.query.sessionID corresponds to empty session* | *JSON response with an empty array of items* |  |  |
| ***Session Found with Items*** | *valid req, req.query.sessionID corresponds to session with items* | *JSON response with an array of items from the specified session ID* |  |  |

**Unit Tests for VendibilityController.js**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***itemVendibility()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Input Valid Session and Item ID*** | sessionId:  ItemId: | *JSON response containing the vendibility analysis result for the specified item.* |  |  |
| ***Input Invalid Session ID*** | sessionId:  ItemId: | *Error response with status code 500 and a message indicating that the session doesn't exist.* |  |  |
| ***Invalid Item ID*** | sessionId: ItemId: | *Error response with status code 500 and a message indicating that the item doesn't exist in the session.* |  |  |
| ***Error During Data Collection*** | sessionId:  ItemId: | *Error message logged to console indicating the failure during data collection phase and an error response with status code 500.* |  |  |
| ***Error During Data Analysis*** | sessionId: ItemId: | *Error message logged to console indicating the failure during data analysis phase and an error response with status code 500.* |  |  |
| ***Error Updating Item Info in Database*** | sessionId:  ItemId: | *Error message logged to console indicating the failure during item update and an error response with status code 500* |  |  |
| ***Valid Input, No Item Found*** | sessionId:  ItemId: | *Error response with status code 500 and a message indicating that the item doesn't exist in the database* |  |  |

**Unit Tests for DataCollection.js**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dataCollection()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Item with All Information Provided*** | *Input: An item object with all attributes already filled: (manufacturer\_part\_num, height\_inch, width\_inch, length\_inch, weight\_lbs, fragile)* | *The same item object returned without any changes* |  |  |
| ***Valid Item with Missing Manufacturer Part Number*** | *Input: An item object with missing manufacturer\_part\_num* | *Error response indicating missing manufacturer part number* |  |  |
| ***Valid Item with Missing Dimensions*** | *Input: An item object with missing height\_inch, width\_inch, and length\_inch* | *Error response indicating missing dimensions* |  |  |
| ***Valid Item with Missing Weight*** | *Input: An item object with missing weight\_lbs* | *Error response indicating missing weight* |  |  |
| ***Valid Item with Missing Fragile Flag*** | *Input: An item object with missing fragile flag* | *Error response indicating missing fragile flag* |  |  |
| ***Valid Item with All Information Provided, No Scraping Required*** | *Input: An item object will all attributes already filled: (manufacturer\_part\_num, height\_inch, width\_inch, length\_inch, weight\_lbs, fragile)* | *The same item returned without any changes* |  |  |
| ***Error During Scraping for Manufacturer Part Number*** | *Input: An item object with an error occurring during scraping for manufacturer part number* | *Error message indicating scraping error* |  |  |
| ***Error During Scraping for Dimensions*** | *Input: An item object with an error occurring during scraping for dimensions* | *Error message indicating scraping error* |  |  |
| ***Error During Scraping for Weight*** | *Input: An item object with an error occurring during scraping for weight* | *Error message indicating scraping error* |  |  |
| ***Error During Scraping for Fragile Flag*** | *Input: An item object with an error occurring during scraping for fragile flag* | *Error message indicating scraping error* |  |  |

**Unit Tests for Continuous\_gpt.js**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***scrapeWebForKeywords()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Input (Successful Scraping)*** | *valid searchURL, valid keywords, limit, chars* | *Array of objects containing scraped data for each link* |  |  |
| ***Invalid Search URL*** | *invalid searchURL, valid keywords, limit, chars* | *Empty array* |  |  |
| ***Error Connecting to Webscraping API*** | *invalid webscraping request* | *Empty array* |  |  |
| ***Timeout Error*** | *searchURL that takes too long to respond* | *Error message handling timeout, returns an array of objects containing scraped data for links that loaded* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***continuousScrape()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Input (Successful Scraping)*** | *Valid item\_desc, manufacturer\_part\_num, variable, additional info* | *Data extracted from scraping based on user responses* |  |  |
| ***Invalid User Response Format*** | *User does not format response correctly* | *System message prompting user to reply with correct format* |  |  |
| ***Google Search Limit Reached*** | *User has made multiple Google search attempts and has reached the limit* | *System message informing user that Google search is disallowed and prompt for answer* |  |  |

**Unit Tests for DataAnalysis.js**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dataAnalysis()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Item and Complete Data*** | *Input: an item object with dimensions, weight, stackability, and fragility.* | *Expected Output: the calculated capacities and vendibility information for locker, coil, and caurousel vending methods.* |  |  |
| ***Invalid Item ( Empty Dimensions)*** | *Input: an item object with out any dimensional data.* | *Expected Output: return null for all capacities,* |  |  |
| ***Valid Item with null weight*** | *Input: an item object with complete data except for weight, which is null.* | *Expected Output: the calculated capacities and vendibility information for locker, coil, and caurousel vending methods*  *Item.vendibility\_notes: = ”Weight not found: assuming weight is negligible”* |  |  |
| ***Valid Item with null stackable*** | *Input: an item object with complete data except for stackable, which is null.* | *Expected Output: the calculated capacities and vendibility information for locker, coil, and caurousel vending methods*  *Item.vendibility\_notes: = ”Stackable not found: assuming item is not stackable”* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***getProlockCapacity()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Item with Dimensions within Locker Limits***  *Locker Depth: 28.25”*  *Locker Width: 11.00”*  *Locker Height (Single Compartment): 5.07”* | *Input: An item object with dimensions (width\_inch, length\_inch, height\_inch) within the limits of locker dimensions.* | *Expected Output: The calculated capacity from the Provided Results spreadsheet* |  |  |
| ***Valid Item with Dimensions Exceeding Locker Depth*** | *Input: An item object with dimensions (width\_inch, length\_inch, height\_inch) where the largest dimension exceeds the locker depth. (* | *Capacity calculation based on a different orientation of the item to fit within the locker constraints.* |  |  |
| ***Valid Item with Weight Exceeding Locker Weight Capacity***  *Weight Cap = 70 lbs* | *Input: An item object with weight (weight\_lbs) exceeding the locker weight capacity.* | *Expected Output: 0, indicating that the item cannot be stored in the locker due to weight constraints.* |  |  |
| ***Valid Item Requiring Vertical Storage*** | *An item object with store\_vertically set to true, and dimensions requiring vertical storage* | *Expected Output: Capacity calculation based on vertical storage requirements and locker constraints..* |  |  |
| ***Valid Item Requiring Vertical Storage but Exceeding Height Limit***  *Maximum height: 5.06 + (6.12 \* 11)* | *Input: An item object with store\_vertically set to true, but dimensions exceeding the maximum height limit for vertical storage* | *0, indicating that the item cannot be stored vertically in the locker due to height constraints* |  |  |
| ***Valid Item Requiring Vertical Storage, Dimensions within Limits*** | *Input: An item object with store\_vertically set to true, and dimensions within the limits for vertical storage* | *Expected Output: Capacity calculation based on vertical storage requirements and locker constraints* |  |  |
| ***Valid Item with Stackable Property, Single Compartment Locker*** | *Input: An item object with stackable set to true and a locker configuration with a single compartment per door.* | *Expected Output: Capacity calculation considering the item's stackability and locker constraints* |  |  |
| ***Valid Item with Stackable Property, Exceeding Locker Weight Capacity*** | *Input: An item object with stackable set to true and weight exceeding the locker weight capacity.* | *Expected Output: Capacity calculation considering stackability, but limited by locker weight capacity.* |  |  |
| ***Valid Item with Non-Stackable Property*** | *Input: An item object with stackable set to false* | *Expected Output: Capacity calculation based on the item's dimensions without considering stackability* |  |  |
| ***getProStockCapacity()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Item with Dimensions within Locker Limits*** | *Input: An item object with dimensions (width\_inch, length\_inch, height\_inch) within the limits of locker dimensions.* | *Expected Output: The calculated capacity from the Provided Results spreadsheet* |  |  |
| ***Valid Item with Dimensions Exceeding Locker Depth*** | *Input: An item object with dimensions (width\_inch, length\_inch, height\_inch) where the largest dimension exceeds the locker depth. (* | *Capacity calculation based on a different orientation of the item to fit within the locker constraints.* |  |  |
| ***Valid Item with Weight Exceeding Locker Weight Capacity*** | *Input: An item object with weight (weight\_lbs) exceeding the locker weight capacity.* | *Expected Output: 0, indicating that the item cannot be stored in the locker due to weight constraints.* |  |  |
| ***Valid Item Requiring Vertical Storage*** | *An item object with store\_vertically set to true, and dimensions requiring vertical storage* | *Expected Output: Capacity calculation based on vertical storage requirements and locker constraints..* |  |  |
| ***Valid Item Requiring Vertical Storage but Exceeding Height Limit*** | *Input: An item object with store\_vertically set to true, but dimensions exceeding the maximum height limit for vertical storage* | *0, indicating that the item cannot be stored vertically in the locker due to height constraints* |  |  |
| ***Valid Item Requiring Vertical Storage, Dimensions within Limits*** | *Input: An item object with store\_vertically set to true, and dimensions within the limits for vertical storage* | *Expected Output: Capacity calculation based on vertical storage requirements and locker constraints* |  |  |
| ***Valid Item with Stackable Property, Single Compartment Locker*** | *Input: An item object with stackable set to true and a locker configuration with a single compartment per door.* | *Expected Output: Capacity calculation considering the item's stackability and locker constraints* |  |  |
| ***Valid Item with Stackable Property, Exceeding Locker Weight Capacity*** | *Input: An item object with stackable set to true and weight exceeding the locker weight capacity.* | *Expected Output: Capacity calculation considering stackability, but limited by locker weight capacity.* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***getCoilVendibility()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Item within Shelf Limits*** | *Input: An item object with dimensions (width\_inch, length\_inch, height\_inch) within the limits of shelf dimensions* | *Expected Output: The coil vendibility object with appropriate values set based on the item dimensions and coil constraints* |  |  |
| ***Valid Item Exceeding Shelf Height Limit*** | *Input: An item object with height exceeding the maximum shelf height limit* | *Expected Output: The coil vendibility object with coil\_vendable set to false, indicating that the item cannot be vendable* |  |  |
| ***Valid Item with Dimensions Exceeding Dual Helix Diameter*** | *Input: An item object with dimensions (width\_inch, length\_inch, height\_inch) where the length exceeds the dual helix diameter limit.* | *Expected Output: The coil vendibility object with coil\_vendable set to false, indicating that the item cannot be vendable.* |  |  |
| ***Valid Item with Dimensions within Single Helix Diameter*** | *Input: An item object with dimensions (width\_inch, length\_inch, height\_inch) within the single helix diameter limit* | *Expected Output: The coil vendibility object with coil\_type set to "single" and appropriate pitch calculated* |  |  |
| ***Valid Item with Weight Exceeding Helix Max Weight*** | *Input: An item object with weight exceeding the maximum weight capacity for coil vendibility* | *Expected Output: The coil vendibility object with adjusted coil capacity based on weight constraints* |  |  |
| ***Valid Item Requiring Riser*** | *Input: An item object with dimensions (width\_inch, length\_inch, height\_inch) requiring a riser for proper vending* | *The coil vendibility object with riser\_required set to true* |  |  |
| ***Valid Item with Heavy Weight*** | *Input: An item object with weight exceeding a predefined threshold for heavy items* | *The coil vendibility object with preferred shelves set to the bottom shelf* |  |  |
| ***Valid Item with Vertical Storage Requirement*** | *Input: An item object with store\_vertically set to true.* | *Expected Output: The coil vendibility object with coil\_vendable set to false, indicating that the item cannot be vendable.* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***getCarouselVendibility()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Item within Carousel Limits*** | *Input: An item object with dimensions (width\_inch, length\_inch, height\_inch) within the limits of carousel dimensions and weight.* | *Expected Output: The carousel vendibility object with appropriate values set based on the item dimensions and carousel constraints.* |  |  |
| ***Valid Item Exceeding Maximum Weight*** | *An item object with weight exceeding the maximum weight capacity for the carousel.* | *The carousel vendibility object with carousel\_vendable set to false, indicating that the item cannot be vendable.* |  |  |
| ***Valid Item with Vertical Storage Requirement*** | *Input: An item object with store\_vertically set to true* | *Expected Output: The carousel vendibility object with carousel\_vendable set to false, indicating that the item cannot be vendable* |  |  |
| ***Valid Item with Invalid Dimensions:*** | *Input: An item object with weight exceeding the maximum weight capacity for the carousel* | *Expected Output: The carousel vendibility object with carousel\_vendable set to false, indicating that the item cannot be vendable.* |  |  |
|  |  |  |  |  |

**Unit Tests for Frontend**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Header()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Logo Displayed*** | *N/A* | *CribMaster Logo displayed in header* |  |  |
| ***CSS Styling Applied*** | *N/A* | *Header component has CSS styling applied from Header.css* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Modal()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Upload File*** | *Upload link clicked* | *File Explorer/Finder opened on device, allowing selecting and uploading an Excel file* |  |  |
| ***Clear File*** | *Excel File currently uploaded, Clear button clicked* | *File removed from Modal* |  |  |
| ***Set Solution Name*** | *Text Input* | *Solution name input field allows entering text* |  |  |
| ***Set Area Type*** | *Text input* | *Area type input field allows entering text* |  |  |
| ***Import Function*** | *Import button clicked* | *Triggers handleSubmit function correctly* |  |  |
| ***Cancel Function*** | *Cancel button clicked* | *Closes modal and resets file input* |  |  |
| ***File Upload to Backend*** | *Import button clicked, with all fields filled* | *Excel file is uploaded to backend server via Axios, and displayed on table* |  |  |
| ***Error for Missing Spreadsheet File*** | *Import button clicked, with no file uploaded* | *Error is shown when trying to import without selecting a file* |  |  |
| ***Error for Missing Solution Name*** | *Import button clicked, with no solution name added* | *Error is shown when trying to import without entering a solution name* |  |  |
| ***Error for Missing Area Type*** | *Import button clicked, with no area type added* | *Error is shown when trying to import without entering an area type* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***TableDisplay()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Display Session ID List*** | *N/A* | *Session ID List is displayed and clears local storage data* |  |  |
| ***Search Input Change*** | *Text Input* | *Search input field allows entering text and filters table data based on text* |  |  |
| ***Toggle Select All*** | *Select All Button Clicked* | *Button selects/deselects all rows in the table* |  |  |
| ***Toggle Row Selection*** | *Checkbox next to row clicked* | *Corresponding row is highlighted and selected* |  |  |
| ***Open Item Details*** | *Item Description is clicked* | *Modal of expanded Item Details for corresponding Item is opened* |  |  |
| ***Export All Button*** | *Export all button is clicked* | *All table data is exported to an Excel file and downloaded onto the user’s computer* |  |  |
| ***Export Selected Button*** | *Export selected button is clicked and there are items that are selected* | *The selected table data is exported to an Excel file and downloaded onto the user’s computer* |  |  |
| ***Error when Export Selected is clicked but no rows are selected*** | *Export selected button is clicked and no items are selected* | *Error is shown when trying to export when no data is selected* |  |  |
| ***Session ID List Button*** | *Session ID List button is clicked* | *Fetches and displays table data from the backend* |  |  |
| ***Handle Chosen Session ID*** | *Session ID button is clicked from the Session ID table* | *Fetches and displays table data from corresponding session* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SessionIDs()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Fetches Session IDs*** | *Valid Connection to Database* | *Fetches session IDs afrom backend and displays when requested* |  |  |
| ***Valid Session IDs Format (array)*** | *Session IDs list are in an array* | *Session IDs displayed* |  |  |
| ***Valid Completed Items Format (array)*** | *Completeed Items are in an array* | *Completed Item count displayed* |  |  |
| ***Valid Uncompleted Item Format (array)*** | *Uncompleted Items are in an array* | *Uncompleted Item count displayed* |  |  |

**3. Test Results**

*In this section, you’ll present/report the results of the actual testing stated in the proceeding sections.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dataCollection()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Item with All Information Provided,***  ***No Scraping Required***  *Input: An item object with all attributes already filled:* | item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,  manufacturer\_part\_num: "135354" ,  height\_inch: 0.5,  width\_inch: 0.5,  length\_inch: 0.5,  weight\_lbs: 0.5,  fragile: false,  default\_issue\_type: "ea",  default\_issue\_qty: "1" | *The same item object returned without any changes* | item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,  manufacturer\_part\_num: "135354" ,  height\_inch: 0.5,  width\_inch: 0.5,  length\_inch: 0.5,  weight\_lbs: 0.5,  fragile: false,  default\_issue\_type: "ea",  default\_issue\_qty: "1" | **PASS** |
| ***Valid Item with Missing Manufacturer Part Number***  *Input: An item object with missing manufacturer\_part\_num* | item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,  //manufacturer\_part\_num: "135354" ,  height\_inch: 0.5,  width\_inch: 0.5,  length\_inch: 0.5,  weight\_lbs: 0.5,  fragile: false,  default\_issue\_type: "ea",  default\_issue\_qty: "1" | *Response containing missing manufacturer part number and all other data* | item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',    height\_inch: 0.5,    width\_inch: 0.5,    length\_inch: 0.5,    weight\_lbs: 0.5,    fragile: false,    default\_issue\_type: 'ea',    default\_issue\_qty: '1',    manufacturer\_part\_num: '"135354"' | FAIL  Function returns manufacturer # correctly, but adds quotation marks. |
| ***Valid Item with Missing Dimensions***  *Input: An item object with missing height\_inch, width\_inch, and length\_inch* | item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',  //height\_inch: 3.7,  //width\_inch: 1,  //length\_inch: 1,  weight\_lbs: 0.5,  fragile: false,  default\_issue\_type: 'ea',  default\_issue\_qty: '1',  manufacturer\_part\_num: '135354' | *Response containing missing dimensions and all other data* | item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',  height\_inch: 3.7,  width\_inch: 1,  length\_inch: 1,  weight\_lbs: 0.5,  fragile: false,  default\_issue\_type: 'ea',  default\_issue\_qty: '1',  manufacturer\_part\_num: '135354' | **PASS** |
| ***Valid Item with Missing Weight***  *Input: An item object with missing weight\_lbs* | item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',  height\_inch: 3.7,  width\_inch: 1,  length\_inch: 1,  //weight\_lbs: 0.5,  fragile: false,  default\_issue\_type: 'ea',  default\_issue\_qty: '1',  manufacturer\_part\_num: '135354' | *Response containing missing manufacturer part number and all other data* | item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',  height\_inch: 3.7,  width\_inch: 1,  length\_inch: 1,  weight\_lbs: 0.5,  fragile: false,  default\_issue\_type: 'ea',  default\_issue\_qty: '1',  manufacturer\_part\_num: '135354' | **PASS** |
| ***Valid Item with Missing Fragile Flag***  *Input: An item object with missing fragile flag* | item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',  height\_inch: 3.7,  width\_inch: 1,  length\_inch: 1,  weight\_lbs: 0.5,  //fragile: false,  default\_issue\_type: 'ea',  default\_issue\_qty: '1',  manufacturer\_part\_num: '135354' | *Error response indicating missing fragile flag* | item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',  height\_inch: 3.7,  width\_inch: 1,  length\_inch: 1,  weight\_lbs: 0.5,  fragile: false,  default\_issue\_type: 'ea',  default\_issue\_qty: '1',  manufacturer\_part\_num: '135354' | **PASS** |
| ***Error During Scraping for Manufacturer Part Number***  *Input: An item object with an error occurring during scraping for manufacturer part number* | item\_description: 'Random Item, no Manufacturer ID',  height\_inch: 3.7,  width\_inch: 1,  length\_inch: 1,  weight\_lbs: 0.5,  fragile: false,  default\_issue\_type: 'ea',  default\_issue\_qty: '1',  //manufacturer\_part\_num: '135354' | *Return Null for manufacturer part number, Error Message indicating scraping error. Continue operation.* | item\_description: 'Random Item, no Manufacturer ID',  height\_inch: 3.7,  width\_inch: 1,  length\_inch: 1,  weight\_lbs: 0.5,  fragile: false,  default\_issue\_type: 'ea',  default\_issue\_qty: '1',  manufacturer\_part\_num: '"undefined"' | FAIL  Function returns a string saying ‘“undefined”’ The system does not error |
| ***Error During Scraping for Dimensions***  *Input: An item object with an error occurring during scraping for dimensions* | item\_description: 'Random Item, no Manufacturer ID',  //height\_inch: 3.7,  //width\_inch: 1,  //length\_inch: 1,  weight\_lbs: 0.5,  fragile: false,  default\_issue\_type: 'ea',  default\_issue\_qty: '1',  manufacturer\_part\_num: 'random Item, no manufacturer part number.' | *Return Null for manufacturer part number, Error Message indicating scraping error. Continue operation.* | *returning "Unable to determine, no information available"*  *Error during GPT prompt: TypeError: Cannot read properties of null* | FAIL  Function does not return item data,produces error message, Does not continue operation |
| ***Error During Scraping for Weight***  *Input: An item object with an error occurring during scraping for weight* |  | *Error message indicating scraping error* |  |  |
| ***Error During Scraping for Fragile Flag***  *Input: An item object with an error occurring during scraping for fragile flag* |  | *Error message indicating scraping error* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***getProlockCapacity()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Valid Item with Dimensions within Locker Limits*** | *item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',*  *manufacturer\_part\_num: '135354',*  *height\_inch: 0.5,*  *width\_inch: 0.5,*  *length\_inch: 0.5,*  *weight\_lbs: 0.5,*  *fragile: false,*  *default\_issue\_type: 'ea',*  *default\_issue\_qty: '1',*  *stackable: true,* | *Expected Output: The calculated capacity from the Provided Results spreadsheet* | *140* | *pass* |
| ***Valid Item with Dimensions Exceeding Locker Depth*** | *item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',*  *manufacturer\_part\_num: '135354',*  *height\_inch: 30,*  *width\_inch: 30,*  *length\_inch: 30,*  *weight\_lbs: 30,*  *fragile: false,*  *default\_issue\_type: 'ea',*  *default\_issue\_qty: '1',*  *stackable: true,* | *Capacity calculation based on a different orientation of the item to fit within the locker constraints.* | *Null*  *Locker vendable false* | *pass* |
| ***Valid Item with Weight Exceeding Locker Weight Capacity*** | *height\_inch: 10,*  *width\_inch: 10,*  *length\_inch: 10,*  *weight\_lbs: 75,* | *Expected Output: 0, indicating that the item cannot be stored in the locker due to weight constraints.* | *Null*  *Locker vendable false* | *pass* |
| ***Valid Item Requiring Vertical Storage*** | *item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',*  *manufacturer\_part\_num: '135354',*  *store\_vertically: true,*  *height\_inch: 0.5,*  *width\_inch: 0.5,*  *length\_inch: 0.5,*  *weight\_lbs: 0.5,*  *fragile: false,*  *default\_issue\_type: 'ea',*  *default\_issue\_qty: '1',*  *stackable: true,* | *Expected Output: Capacity calculation based on vertical storage requirements and locker constraints..* | *140* | *pass* |
| ***Valid Item Requiring Vertical Storage but Exceeding Height Limit*** | *height\_inch: 73, width\_inch: 10, length\_inch: 10, weight\_lbs: 10,* | *0, indicating that the item cannot be stored vertically in the locker due to height constraints* | *0* | *pass* |
| ***Valid Item Requiring Vertical Storage, Dimensions within Limits*** | *item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',*  *manufacturer\_part\_num: '135354',*  *store\_vertically: true,*  *height\_inch: 0.5,*  *width\_inch: 0.5,*  *length\_inch: 0.5,*  *weight\_lbs: 0.5,*  *fragile: false,*  *default\_issue\_type: 'ea',*  *default\_issue\_qty: '1',*  *stackable: true,* | *Expected Output: Capacity calculation based on vertical storage requirements and locker constraints* | *140* | *pass* |
| ***Valid Item with Stackable Property, Single Compartment Locker*** | *stackable: true,*  *height\_inch: 1,*  *width\_inch: 0.5,*  *length\_inch: 0.5,*  *weight\_lbs: 0.5,* | *Expected Output: Capacity calculation considering the item's stackability and locker constraints* |  |  |
| ***Valid Item with Stackable Property, Exceeding Locker Weight Capacity*** | stackable: true,  height\_inch: 1,  width\_inch: 0.5,  length\_inch: 0.5,  weight\_lbs: 75,  fragile: false, | *Expected Output: Capacity calculation considering stackability, but limited by locker weight capacity.* | *null* | *pass* |
| ***Valid Item with Non-Stackable Property*** | *manufacturer\_part\_num: "135354" ,*  *stackable: false,*  *height\_inch: 0.5,*  *width\_inch: 0.5,*  *length\_inch: 0.5,*  *weight\_lbs: 10,* | *Expected Output: Capacity calculation based on the item's dimensions without considering stackability* | *7* | *pass* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***dataAnalysis()*** | | | | | | |
| **Description** | **Input** | **Expected Output** | | | **Actual Output** | **Pass/Fail** |
| ***Valid Item and Complete Data*** | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *stackable: true,*  *height\_inch: 0.5,*  *width\_inch: 0.5,*  *length\_inch: 0.5,*  *weight\_lbs: 0.5,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *Expected Output: the calculated capacities and vendibility information for locker, coil, and caurousel vending methods.* | *coil\_vendability: {*  *coil\_vendable: true,*  *needs\_repack\_for\_coil: null,*  *coil\_pitch: 32,*  *coil\_capacity: 40,*  *coil\_type: 'single',*  *preferred\_shelves: [ 1, 2, 3, 4, 5, 6 ],*  *preferred\_row: null,*  *riser\_required: false,*  *flip\_bar\_required: null,*  *coil\_end\_clock\_position: null*  *},*  *carousel\_vendability: {*  *carousel\_vendable: true,*  *needs\_repack\_for\_carousel: null,*  *num\_slots\_per\_item: 1,*  *carousel\_capacity: 56*  *}* | | | *pass* |
| ***Invalid Item ( Empty Dimensions)*** | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *stackable: true,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *Expected Output: return null for all capacities,* | *Error during Analysis: TypeError: Cannot read properties of undefined (reading 'locker\_vendable')* | | |  |
| ***Valid Item with null weight*** | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *stackable: true,*  *height\_inch: 0.5,*  *width\_inch: 0.5,*  *length\_inch: 0.5,*  *weight\_lbs: null,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *Expected Output: the calculated capacities and vendibility information for locker, coil, and caurousel vending methods*  *Item.vendibility\_notes: = ”Weight not found: assuming weight is negligible”* | *locker\_vendability: {*  *locker\_vendable: true,*  *num\_compartments\_per\_locker\_door: 1,*  *capacity\_for\_express\_locker: null,*  *capacity\_for\_prostock\_locker: 18816,*  *capacity\_for\_prolock\_locker: 12320*  *},*  *coil\_vendability: {*  *coil\_vendable: true,*  *needs\_repack\_for\_coil: null,*  *coil\_pitch: 32,*  *coil\_capacity: 32,*  *coil\_type: 'single',*  *preferred\_shelves: [ 1, 2, 3, 4, 5, 6 ],*  *preferred\_row: null,*  *riser\_required: false,*  *flip\_bar\_required: null,*  *coil\_end\_clock\_position: null*  *},*  *carousel\_vendability: {*  *carousel\_vendable: true,*  *needs\_repack\_for\_carousel: null,*  *num\_slots\_per\_item: 1,*  *carousel\_capacity: 56*  *}* | | | *fail* |
| ***Valid Item with null stackable*** | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *stackable: null,*  *height\_inch: 0.5,*  *width\_inch: 0.5,*  *length\_inch: 0.5,*  *weight\_lbs: 0.5,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *Expected Output: the calculated capacities and vendibility information for locker, coil, and caurousel vending methods*  *Item.vendibility\_notes: = ”Stackable not found: assuming item is not stackable”* | | | *{*  *item\_description: '135354 24221 242 REMOVABLE TL 10 ML BO',*  *manufacturer\_part\_num: '135354',*  *stackable: true,*  *height\_inch: 0.5,*  *width\_inch: 0.5,*  *length\_inch: 0.5,*  *weight\_lbs: 0.5,*  *fragile: false,*  *default\_issue\_type: 'ea',*  *default\_issue\_qty: '1',*  *locker\_vendability: {*  *locker\_vendable: true,*  *num\_compartments\_per\_locker\_door: 1,*  *capacity\_for\_express\_locker: null,*  *capacity\_for\_prostock\_locker: 200,*  *capacity\_for\_prolock\_locker: 140*  *},*  *coil\_vendability: {*  *coil\_vendable: true,*  *needs\_repack\_for\_coil: null,*  *coil\_pitch: 32,*  *coil\_capacity: 40,*  *coil\_type: 'single',*  *preferred\_shelves: [ 1, 2, 3, 4, 5, 6 ],*  *preferred\_row: null,*  *riser\_required: false,*  *flip\_bar\_required: null,*  *coil\_end\_clock\_position: null*  *},*  *carousel\_vendability: {*  *carousel\_vendable: true,*  *needs\_repack\_for\_carousel: null,*  *num\_slots\_per\_item: 1,*  *carousel\_capacity: 56*  *}*  *}* | *fail* |
| ***getCoilVendibility()*** | | | | | | |
| **Description** | **Input** | **Expected Output** | | | **Actual Output** | **Pass/Fail** |
| ***Valid Item within Shelf Limits*** | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *stackable: null,*  *height\_inch: 6,*  *width\_inch: 1,*  *length\_inch: 3,*  *weight\_lbs: 0.5,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *Expected Output: The coil vendibility object with appropriate values set based on the item dimensions and coil constraints* | | *coil\_vendability: {*  *coil\_vendable: true,*  *needs\_repack\_for\_coil: null,*  *coil\_pitch: 15,*  *coil\_capacity: 15,*  *coil\_type: 'large',*  *preferred\_shelves: [ 1, 2, 3, 4, 5, 6 ],*  *preferred\_row: null,*  *riser\_required: false,*  *flip\_bar\_required: null,*  *coil\_end\_clock\_position: null*  *}* | | *pass* |
| ***Valid Item Exceeding Shelf Height Limit*** | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *stackable: null,*  *height\_inch: 9,*  *width\_inch: 1,*  *length\_inch: 3,*  *weight\_lbs: 0.5,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *Expected Output: The coil vendibility object with coil\_vendable set to false, indicating that the item cannot be vendable* | | *coil\_vendability: {*  *[start] coil\_vendable: false,*  *[start] needs\_repack\_for\_coil: null,*  *[start] coil\_pitch: null,*  *[start] coil\_capacity: null,*  *[start] coil\_type: null,*  *[start] preferred\_shelves: null,*  *[start] preferred\_row: null,*  *[start] riser\_required: null,*  *[start] flip\_bar\_required: null,*  *[start] coil\_end\_clock\_position: null*  *[start] },* | |  |
| ***Valid Item with Dimensions Exceeding Dual Helix Diameter*** | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *stackable: null,*  *height\_inch: 5,*  *width\_inch: 1,*  *length\_inch: 6,*  *weight\_lbs: 0.5,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *Expected Output: The coil vendibility object with coil\_vendable set to false, indicating that the item cannot be vendable.* | | *coil\_vendability: {*  *[start] coil\_vendable: false,*  *[start] needs\_repack\_for\_coil: null,*  *[start] coil\_pitch: null,*  *[start] coil\_capacity: null,*  *[start] coil\_type: null,*  *[start] preferred\_shelves: null,*  *[start] preferred\_row: null,*  *[start] riser\_required: null,*  *[start] flip\_bar\_required: null,*  *[start] coil\_end\_clock\_position: null*  *[start] },* | | *pass* |
| ***Valid Item with Dimensions within Single Helix Diameter*** | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *stackable: null,*  *height\_inch: 5,*  *width\_inch: 1,*  *length\_inch: 2.0,*  *weight\_lbs: 0.5,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *Expected Output: The coil vendibility object with coil\_type set to "single" and appropriate pitch calculated* | | *coil\_vendability: {*  *[start] coil\_vendable: true,*  *[start] needs\_repack\_for\_coil: null,*  *[start] coil\_pitch: 18,*  *[start] coil\_capacity: 18,*  *[start] coil\_type: 'single',*  *[start] preferred\_shelves: [ 1, 2, 3, 4, 5, 6 ],*  *[start] preferred\_row: null,*  *[start] riser\_required: false,*  *[start] flip\_bar\_required: null,*  *[start] coil\_end\_clock\_position: null*  *[start] },* | | *pass* |
| ***Valid Item with Weight Exceeding Helix Max Weight***  *Input: An item object with weight exceeding the maximum weight capacity for coil vendibility* | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *stackable: null,*  *height\_inch: 5,*  *width\_inch: 1,*  *length\_inch: 2.0,*  *weight\_lbs: 15.5,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *Expected Output: The coil vendibility object with adjusted coil capacity based on weight constraints* | | *coil\_vendability: {*  *[start] coil\_vendable: true,*  *[start] needs\_repack\_for\_coil: null,*  *[start] coil\_pitch: 18,*  *[start] coil\_capacity: 1,*  *[start] coil\_type: 'single',*  *[start] preferred\_shelves: [ 6 ],*  *[start] preferred\_row: null,*  *[start] riser\_required: false,*  *[start] flip\_bar\_required: null,*  *[start] coil\_end\_clock\_position: null*  *[start] },* | | *pass* |
| ***Valid Item Requiring Riser***  *Input: An item object with dimensions (width\_inch, length\_inch, height\_inch) requiring a riser for proper vending* | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *stackable: null,*  *height\_inch: 0.1,*  *width\_inch: 0.1,*  *length\_inch: 0.1,*  *weight\_lbs: 0.5,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *The coil vendibility object with riser\_required set to true* | | *coil\_vendability: {*  *[start] coil\_vendable: true,*  *[start] needs\_repack\_for\_coil: null,*  *[start] coil\_pitch: 32,*  *[start] coil\_capacity: 40,*  *[start] coil\_type: 'single',*  *[start] preferred\_shelves: [ 1, 2, 3, 4, 5, 6 ],*  *[start] preferred\_row: null,*  *[start] riser\_required: false,*  *[start] flip\_bar\_required: null,*  *[start] coil\_end\_clock\_position: null*  *[start] },* | | *fail* |
| ***Valid Item with Heavy Weight***  *Input: An item object with weight exceeding a predefined threshold for heavy items* | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *stackable: null,*  *height\_inch: 0.1,*  *width\_inch: 0.1,*  *length\_inch: 0.1,*  *weight\_lbs: 35,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *The coil vendibility object with preferred shelves set to the bottom shelf* | | *coil\_vendability: {*  *[start] coil\_vendable: true,*  *[start] needs\_repack\_for\_coil: null,*  *[start] coil\_pitch: 32,*  *[start] coil\_capacity: 0,*  *[start] coil\_type: 'single',*  *[start] preferred\_shelves: [ 6 ],*  *[start] preferred\_row: null,*  *[start] riser\_required: false,*  *[start] flip\_bar\_required: null,*  *[start] coil\_end\_clock\_position: null*  *[start] },* | | *pass* |
| ***Valid Item with Vertical Storage Requirement***  *Input: An item object with store\_vertically set to true.* | *item\_description: `135354 24221 242 REMOVABLE TL 10 ML BO`,*  *manufacturer\_part\_num: "135354" ,*  *store\_vertically: true,*  *height\_inch: 1,*  *width\_inch: 1,*  *length\_inch: 1,*  *weight\_lbs: 2,*  *fragile: false,*  *default\_issue\_type: "ea",*  *default\_issue\_qty: "1"* | *Expected Output: The coil vendibility object with coil\_vendable set to false, indicating that the item cannot be vendable.* | | *coil\_vendability: {*  *[start] coil\_vendable: true,*  *[start] needs\_repack\_for\_coil: null,*  *[start] coil\_pitch: null,*  *[start] coil\_capacity: null,*  *[start] coil\_type: null,*  *[start] preferred\_shelves: null,*  *[start] preferred\_row: null,*  *[start] riser\_required: null,*  *[start] flip\_bar\_required: null,*  *[start] coil\_end\_clock\_position: null*  *[start] },* | | *pass* |

**Unit Tests for Frontend**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Header()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Logo Displayed*** | *N/A* | *CribMaster Logo displayed in header* | *CribMaster Logo displayed* | *PASS* |
| ***CSS Styling Applied*** | *N/A* | *Header component has CSS styling applied from Header.css* | *CSS styling applied* | *PASS* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Modal()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Upload File*** | *Upload link clicked* | *File Explorer/Finder opened on device, allowing selecting and uploading an Excel file* | *File Explorer/Finder opened on device, allowing selecting and uploading an Excel file* | *PASS* |
| ***Clear File*** | *Excel File currently uploaded, Clear button clicked* | *File removed from Modal* | *File removed from Modal* | *PASS* |
| ***Set Solution Name*** | *Text Input* | *Solution name input field allows entering text* | *Solution name input field allows entering text* | *PASS* |
| ***Set Area Type*** | *Text input* | *Area type input field allows entering text* | *Area type input field allows entering text* | *PASS* |
| ***Import Function*** | *Import button clicked* | *Triggers handleSubmit function correctly* | *Triggers handleSubmit function correctly* | *PASS* |
| ***Cancel Function*** | *Cancel button clicked* | *Closes modal and resets file input* | *Closes modal and resets file input* | *PASS* |
| ***File Upload to Backend*** | *Import button clicked, with all fields filled* | *Excel file is uploaded to backend server via Axios, and displayed on table* | *Excel file is uploaded to backend server via Axios, and displayed on table* | *PASS* |
| ***Error for Missing Spreadsheet File*** | *Import button clicked, with no file uploaded* | *Error is shown when trying to import without selecting a file* | *No Alert, Stuck on Modal* | *FAIL* |
| ***Error for Missing Solution Name*** | *Import button clicked, with no solution name added* | *Error is shown when trying to import without entering a solution name* | *No Alert, Stuck on Modal* | *FAIL* |
| ***Error for Missing Area Type*** | *Import button clicked, with no area type added* | *Error is shown when trying to import without entering an area type* | *No Alert, Stuck on Modal* | *FAIL* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***TableDisplay()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Display Session ID List*** | *N/A* | *Session ID List is displayed and clears local storage data* | *Session ID List is displayed and clears local storage data* | *PASS* |
| ***Search Input Change*** | *Text Input* | *Search input field allows entering text and filters table data based on text* | *Search input field allows entering text and filters table data based on text* | *PASS* |
| ***Toggle Select All*** | *Select All Button Clicked* | *Button selects/deselects all rows in the table* | *Button selects/deselects all rows in the table* | *PASS* |
| ***Toggle Row Selection*** | *Checkbox next to row clicked* | *Corresponding row is highlighted and selected* | *Corresponding row is highlighted and selected* | *PASS* |
| ***Open Item Details*** | *Item Description is clicked* | *Modal of expanded Item Details for corresponding Item is opened* | *Modal of expanded Item Details for corresponding Item is opened* | *PASS* |
| ***Export All Button*** | *Export all button is clicked* | *All table data is exported to an Excel file and downloaded onto the user’s computer* | *All table data is exported to an Excel file and downloaded onto the user’s computer* | *PASS* |
| ***Export Selected Button*** | *Export selected button is clicked and there are items that are selected* | *The selected table data is exported to an Excel file and downloaded onto the user’s computer* | *The selected table data is exported to an Excel file and downloaded onto the user’s computer* | *PASS* |
| ***Error when Export Selected is clicked but no rows are selected*** | *Export selected button is clicked and no items are selected* | *Error is shown when trying to export when no data is selected* | *Does not export, but no alert shown* | *FAIL* |
| ***Session ID List Button*** | *Session ID List button is clicked* | *Fetches and displays table data from the backend* | *Fetches and displays table data from the backend* | *PASS* |
| ***Handle Chosen Session ID*** | *Session ID button is clicked from the Session ID table* | *Fetches and displays table data from corresponding session* | *Fetches and displays table data from corresponding session* | *PASS* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SessionIDs()*** | | | | |
| **Description** | **Input** | **Expected Output** | **Actual Output** | **Pass/Fail** |
| ***Fetches Session IDs*** | *Valid Connection to Database* | *Fetches session IDs afrom backend and displays when requested* | *Fetches session IDs afrom backend and displays when requested* | *PASS* |
| ***Valid Session IDs Format (array)*** | *Session IDs list are in an array* | *Session IDs displayed* | *Session IDs displayed* | *PASS* |
| ***Valid Completed Items Format (array)*** | *Completeed Items are in an array* | *Completed Item count displayed* | *Completed Item count displayed* | *PASS* |
| ***Valid Uncompleted Item Format (array)*** | *Uncompleted Items are in an array* | *Uncompleted Item count displayed* | *Uncompleted Item count displayed* | *PASS* |