

CP1295 Advanced JavaScript Assignment 02 v1



CP1295 Assignment 02

Contents

A. Brief Description	3
B. Initial Page.....	4
C. Process Cargo Button	5
D. Cargo Processing Form	6
E. Processing of the Data	7
F. Submission Requirements	8
Page 1.....	8
Page 2.....	8
Page 3.....	8
Page 4.....	8
Page 5.....	8
Page 6.....	8
Page 7.....	9
Page 8.....	9
Page 9.....	9
Page 10.....	9
G. Code Requirements	10
H. Grading Rubric	11

A. Brief Description

The specifications for Assignment 2 are based on updates to specifications of Assignment 01.

CNA Rail System is upgrading its software.

For safety reasons it wishes to ensure that the box cars are not overweight.

The track maintenance team require the know the total weight of the boxcar. (Empty Plus Cargo)

Upgrades to the software will include testing to ensure that the BoxCar with cargo will not exceed the max weight of (BoxCar Empty Weight + Cargo Weight).

The Data Management group wishes to have some basic error checking to include empty fields.

The Cargo Management Team wishes to keep track of the cargo one it has arrived at the cargo depot.

The cargo will be placed in the box car if possible. If the addition a any piece of cargo will cause the boxcar to exceed it maximum allowable weight, then the cargo will be placed into the warehouse and will be placed into the next boxcar that is available.

B. Initial Page

CNA RAIL SYSTEM

Box Car ID:

Empty Weight:

Max Weight:

Current Weight:

Transport ID:

Description:

Cargo Weight:

Cargo Status

Transport ID	Description	Weight	Status
None			

Manifest: BX500

Transport ID	Description	Weight
None		

Initial Page will contain only two forms.

(1) BoxCar form with four text fields and labels as shown here and 2 non functional buttons.

1a. BoxCar ID: Fixed Text BX500 readonly

1b Empty Weight: Fixed Text 15000 readonly

1c Max Weight: Fixed Text 105,000 readonly

1d Total Weight: To Be Set to Empty Weight value when DOM is loaded and field is readonly.

(2) Process Cargo Form with three text fields and labels as shown here. Process Cargo Form has two functional buttons.

2a All three fields will accept user input. No On-Form editing.

2b Process Cargo will only function on two conditions.

2b1 If all three fields contain data. Generate an alert message if a Blank Field is detected.

and

2b2 The value contained in 'Cargo Weight' is numeric and is between (inclusive) 1 to (Max Weight – Empty Weight). No cargo item can be accepted if it is too heavy for an empty boxcar. Generate an alert message if the weight item is "Non Numeric" or "Exceeds Weight Limit". You have a total of three possible alert messages.

3 Reset Button will blank only the three fields of Process Cargo Form.

The index.html should only contain code for what is shown in the initial form shown here.

MORE on Process Cargo Button activity to follow

C. Process Cargo Button

The button event handler will perform the necessary tests to see if there will be any cargo processing.

The processing will be as follows.

1. All accepted cargo will be added to Cargo Status. Accepted is defined as all fields are filled and the weight is within the boundaries specified during the cargo data test.

Cargo Status

Transport ID	Description	Weight	Status
TXL2031S01	50000 Shirts	25000	BX500
TXL2031S02	30000 Coats	30000	BX500
MED2033S01	Medical MX45000	16000	BX500
MED2033S02	Medical MX34111	16000	Warehouse
GSX2037S01	Lamp Oil K1 Drum	10000	Warehouse

A test will be made to ensure that the addition of the cargo to the box car manifest will not put the boxcar overweight.

2a. If the addition of the cargo will not put the box car manifest overweight then the cargo status is set to the ID of the BOXCAR and the cargo is also added to the boxcar manifest.

Manifest: BX500

Transport ID	Description	Weight
TXL2031S01	50000 Shirts	25000
TXL2031S01	30000 Coats	30000
MED2033S01	Medical MX45000	16000
Total Weight:		71000

The box manifest will contain all cargo that has been accepted.

Total Weight is calculated.

The Total Weight (of the Cargo) plus the empty weight of the BoxCar cannot exceed the Max Weight of the BoxCar. The Max Weight of the train is used to ensure that the path that the train will take will support the Total Weight of the

train.

D. Cargo Processing Form

Cargo processing component of the page has three active fields.

Transport ID:

Description:

Cargo Weight:

The fields are filled out then 'Process Cargo' is clicked to activate the processing of the data by calling a JavaScript 'process_cargo.js'.

Reset Form will set all three fields to blank.

Form Checking is not performed within the HTML section. The JavaScript will perform some validation as described in previous sections of this assignment.

Enter the 5 sets of data as shown in the spreadsheet extract below for processing.

After the entering of each set, click on 'Process Cargo'. This will call 'process_cargo.js'.

Transport ID	Description	Weight
TXL2031s01	50,000 Shirts	25000
TXL2031s02	30,000 Coats	30000
MED2033s01	Medical MX45000	16000
MED2033s02	Medical MX34111	16000
GSX2037s01	Lamp Oil K1 Drum	10000

E. Processing of the Data

CNA RAIL SYSTEM

Box Car ID:

Empty Weight:

Max Weight:

Current Weight: 102,000

Transport ID:

Description:

Cargo Weight:

Cargo Status

Transport ID	Description	Weight	Status
TXL2031S01	50000 Shirts	25000	BX500
TXL2031S02	30000 Coats	30000	BX500
MED2033S01	Medical MX45000	16000	BX500
MED2033S02	Medical MX34111	16000	Warehouse BX500
GSX2037S01	Lamp Oil K1 Drum	10000	Warehouse (OK)

Manifest: BX500

Transport ID	Description	Weight
TXL2031S01	50000 Shirts	25000
TXL2031S01	30000 Coats	30000
MED2033S01	Medical MX45000	16000
Total Weight:		71000
MED2033S02 Medical MX34111		16000

Total Cargo Weight 87000

(1) Cargo Status

(1a) Create output Labels for Cargo Status with table headers as shown.

(1b) Cargo that is NOT accepted for the boxcar will be placed in the warehouse to await the next train.

(1c) Cargo fields are added as they are accepted

(2) Create output Labels for Manifest.

(2a) Header as shown “Manifest: BX500” will determine the correct boxcar ID from the boxcar form.

(2b) Table headers are applied as shown in the sample output

(2c) All cargo data is displayed as it is added from the cargo form.

(2d) The ‘Total Weight’ in the Box Car form is updated as the cargo data is entered.

Technique to adding in the table data to the page is unrestricted for this assignment. You may use Document.write for this assignment.

F. Submission Requirements

Generate a word document called Assignment 02 – Your name and student number.

Page 1

- a. Add Course Number
- b. Your Name
- c. Your Student Number

Page 2

Screen Shot of initial FORM prior to any data entry

Page 3

Screen Shot of FIRST data entry after 'Process Cargo' button pressed.

Expect to observe 1 row of output with correct Total Cargo Weight calculations posted to output and to Box Car form for Total Weight.

Page 4

Screen shot of LAST data entry after 'Process Cargo' button pressed.

Expect to observe 5 rows of output with correct Total Cargo Weight calculations posted to output and to Box Car form for Total Weight.

Page 5

Screen shot showing a Blank Field situation with the alert message.

Page 6

Screen shot shown Non Numeric situation for cargo weight field with alert message.

Page 7

Screen shot showing attempt to enter Cargo with 99,000 kg including the alert message.

Page 8

Copy and paste the code for your CSS documents. (Not screen shots)

Page 9 (or next blank page thereafter)

Copy and paste the code for your index.html (Not screen shots)

Page 10 (or next blank page thereafter)

Copy and paste the code for 'process_cargo.js'

G. Code Requirements

The assignments are based course outline requirements.

Code used for the completion of this assignment must follow the following guidelines.

Inclusions

(1) Code must be based on code demonstrated in this course or its pre-requisite course(s)

- Course Text Book

- Course Notes

- Course Handouts

Exclusions

(2) Code must follow the following exclusion rule(s)

a. Note: getElementById, getElementsByTagName, innerHTML, outerHTML are not permitted in this course.

b. Where DOM element selection is required use only

- i. document.querySelector(sel)

- ii. document.querySelectorAll(sel) OR

- iii. jQuery selector.

H. Grading Rubric

To maximize your grade be sure to consider the Grading Rubric as part of the of list of requirements.

Items missed are marks that you will not receive.

CP1295

Assignment 02

Grading Rubric

A	Name, Student #, Assignment # on Page 1	2
B1	HTML file contained Box Car Form new Field Max Weight	2
B2	HTML file contained Process Cargo Form	2
	BOX CAR FORM (BCM)	
C1	Box Car ID / Text Field RO default set BX500	2
C2	Empty Weight set to 15000 RO Max Weight set to 15000 RO	4
C3	Current Weight NOT set. Value will be set of DOM event listener	4
C4	Process Box Car Non-Functional Button	1
C5	Reset Form Non-Functional Button	1
	Cargo Processing Form (CPF)	
D1	Transport ID / Text Field	2
D2	Description / Text Field	2
D3	Cargo Weight / Text Field	2
D4	Process Cargo Button (exists)	2
D5	Reset Form Button (Exists)	2
D6	Process Cargo Button - calls Java Script	5
D7	Reset Form Button - resets forms 3 fields to blank	5

Continued on next page

	Process Cargo JavaScript	
E1	DOM Test to set Current Weight to Empty Weight	6
E2	Process Cargo BTN Testing for Blank Fields with Alert	6
E3	Process Cargo BTN Testing for weight limit	6
E4	Correct addition of Cargo Data to Cargo Status Form	6
E5	Cargo Status set correctly to BX500 or Warehouse	6
E6	Test of Cargo for adding to manifest	8
E7	Display Tables initialized and created in Script	6
E8	Manifest Total Weight Displayed	6
	Submission	
FP2	Initial FORM SS - PRE data entry	1
FP3	First Data Entry SS	1
FP4	Last Data Entry SS	2
FP5	Blank Cargo Field Test SS	1
FP6	NON numeric cargo weight Test - showing alert SS	2
FP7	Cargo Weight of 99,000 Test - showing alert SS	2
FP8	CSS code	1
FP9	Index.html code	1
FP10	process_cargo.js code	1
		100

End of Assignment

