FOSSEE Xcos

Thank you for your interest in FOSSEE Xcos project. Before you start working on the project, we would like to assess your coding skills.

Please read the details mentioned below carefully :

Important note to be followed for all task:

1. In our screening task we have two phases namely Phase 1 and Phase 2.
2. From that Phase 1 is compulsory and in Phase 2, you have two options namely a & b, out of which you have to select any one compulsory.
3. Attempting both Phase 1 & Phase 2 is compulsory.
4. Everything should be created using only JavaScript, HTML and CSS.
5. Make sure code is properly commented and give relevant names to elements used in code.

Hint : Make use of **document.createElement** for creating elements of forms like textfield, button, and so on.

If any doubts please contact us at **contact@scilab.in** or **info@fossee.in** with Subject as **XCOS Query**.

Screening Task

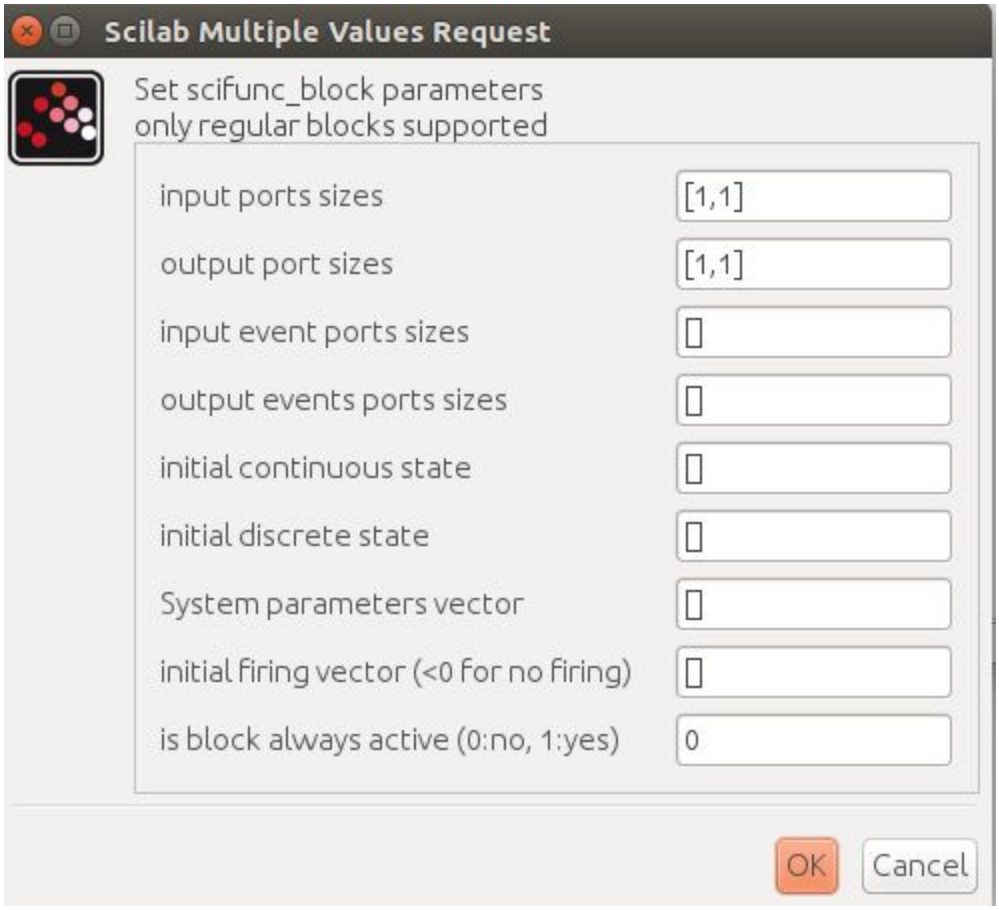
Phase 1:

"Multiple pop-ups forms with elements along with data storage of each form and displaying all data at end "

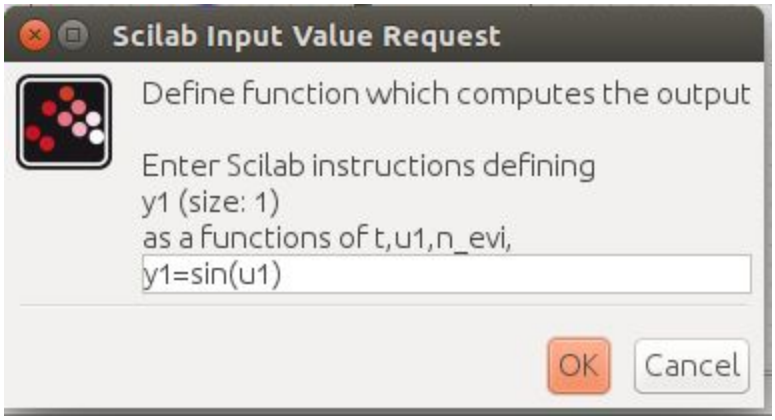
1. Create a HTML page with a button “Submit”
2. On clicking on “Submit” a pop up window with some fields (similar as below image except logo image) should open up user should be allowed to enter values and add validation to fields for example :

First field : input ports sizes

User can enter [1,1] for one input of size 1x1 and also [1,1;2,3] for two inputs of size 1x1 and 2x3. In this case validation should be to accept a matrix only. Rest validations can be assumed accordingly.



3. In the above form if user click on "Cancel" the form will close and no further process will take happen.
4. If user click on “OK” , another new form should pop-up (similar as below image), but along with new form pop-up, previous form should get closed. (But in background , you should make provision to save data of first form for further usage in program).

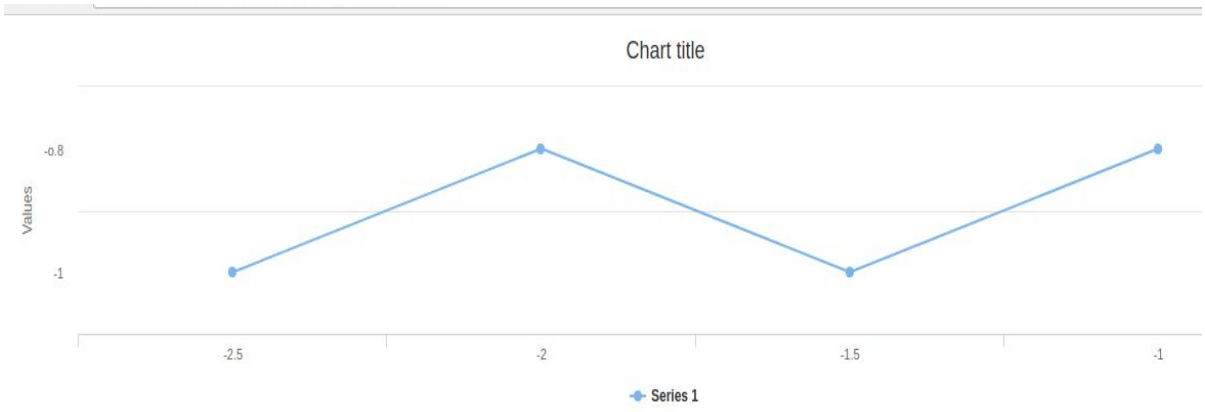


5. When user click on “OK” in second form , all data entered in first and second pop-up along with label should be print/display on HTML page (first page created with Submit button on it), after closing the second pop-up form.
6. If user click on “Cancel”, second pop-up will get close and only data enter by user in first pop-up form should get display on HTML page.

Phase 2:
(A)
"Draggable points in graph along with storing and displaying coordinates value of new points."

1. Create a html page with a button “Submit”

2. On clicking on “Submit”, a window should come up with a graph in it showing four points as shown in below image



3. Make use of **Highcharts library for chart**.

4. This chart should have **draggable** functionality ie. User can drag points and change its position, accordingly graph should change.

5. If user try to drag a any point on line connecting two existing point a new point should create from it and we will have 5 or 6 points as per dragged.

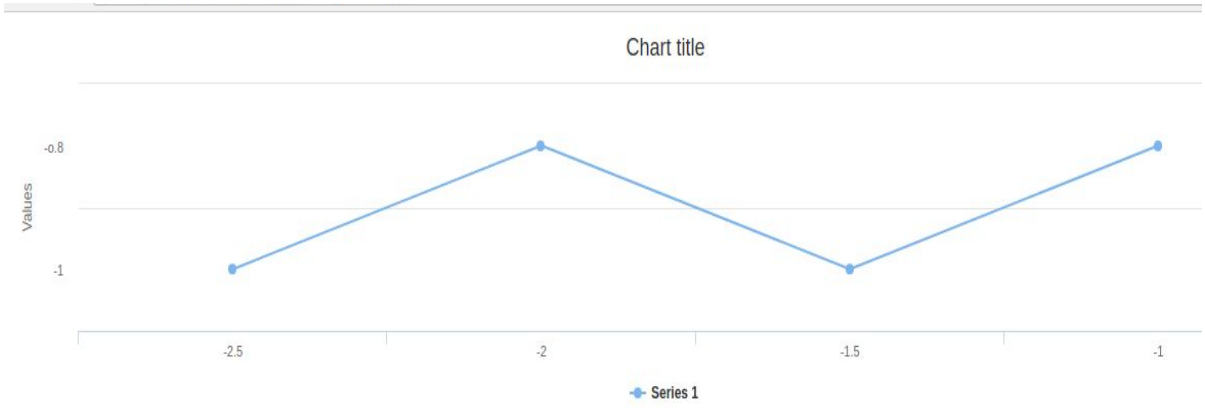
6. Make sure even axis should change according to new points.

7. Once user click on “close window” and this graph window get closed, all modified/newly created point’s **x y coordinate** values should be display on main page
   e.g. point 1 : [0.8,2.0] point 2 : [1.2,2.8] and so on.

OR

B) :
“Graph with menu and sub menu having different effect/action on graph ."

1. Create a html page with a button “Submit”
2. On clicking on “Submit”, a window should come up with a graph in it showing four points as shown in below image and a menu bar at top, above graph with menu **“EDIT”** and **“DATA”** and **sub menu** under each menu as shown in below table.



EDIT	DATA
Replot	Read
Ok	Save
Abort	Clear

3. Make use of **Highcharts library for chart**.
4. Graph points can be fixed values (**No-draggable points in this task**)
5. When user click on submenu **“Save”**, all points xy values should be written in file e.g. “filename.xy” (text file but extension should be ‘.xy’ only) and save and downloaded. File will contain values only in following format :

```
-2.00   -1.00
-1.00    1.00
 1.00   -1.00
 2.00    1.00
```

Here in first row, -2.00 is x value of point 1, -1.00 is y value of point 1 and so on for rest 3 points

6. When user Click on submenu **“Clear”** those points should get clear only graph window with no points can be seen.
7. When user click on submenu **“Read”**, it should open a file browser
8. Select a file having points value stored (“filename.xy”)and replot those points on graph again. Points should be strictly read from the chosen file and not hardcoded in function.
9. When user click on submenu **“Abort”** the graph window should get closed and values of graph points should be display on htm page.

Procedure to Submit Code

1. You will receive an email with the subject line "FOSSEE Summer Fellowship 2018 - Submission".
2. Click here to visit the submission portal and login with your username and password given in the email.
3. Select ‘Xcos’ under 'My Courses' section to submit the task(s).
4. Make separate files for both Phase 1 and Phase 2 and zip them together in one file with proper naming convention as stated in next point.
5. Files should be named as 'ID number-Course short name-Submission task name

Where to find ID number?

1. Go to your profile in the submission portal.
2. Click on 'Actions' icon.
3. Go to 'Edit profile'.
4. Go to 'Optional' section.
5. In this section, there is a field containing 'ID number' against which each student's unique ID number is present.).