1. The ECMAScript page provides an application for pricing and ordering coffee:

* the interface should have suitable form elements to allow the user to order multiple drinks by specifying:
* choice of drink from:
  + latte, cappuccino, flat white, espresso, americano
* of size drink:
  + small, medium (default) or large
* milk required if the drink is a latte, cappuccino or flat white, from:
  + whole, skimmed, semi-skimmed (default), coconut or soya
* any extras required:
  + sugar, cream, number of shots of syrup (can add more than 1)
* there should be an:
  + "add to order" button to add the current drink(s) to the overall order
  + "place order" button to display a confirmation message
* the user should be able to see the following information:
  + current drink(s)
  + cost of the current drink(s)
  + overall order
  + overall cost of the order
* the costs are calculated as follows:
  + small drinks LKR 200
  + medium drinks LRR 250
  + large drinks LKR 300
  + extra cream LKR 50
  + syrup LKR 25 per shot
* when the page first loads:
  + the current drink and overall order should be blank, and the cost of the current drink and the cost of the overall order should be zero
* when the cost of the current drink changes in any way (ie. size or extras):
  + the cost of the current drink must be updated and displayed
* when the "add to order" button is clicked:
  + the overall order and the overall cost are updated and displayed
  + the current drink is reset to blank and the cost of the current drink is reset to 0
* when the "place order" button is clicked:
  + a message is displayed thanking the user for their custom
  + the current drink and overall order are reset to blank and the cost of the current drink and the cost of the overall order are reset to zero

1. The ECMAScript page should provide advanced features to:

* save a drink as a favourite
  + 2 buttons need to be added to the interface
  + when the "add to favourites" button is clicked, the drink should be saved in local storage, overwriting any existing favourite
  + when the "order favourite" button is clicked, the drink is added to the current order
* earn loyalty points
  + the number of drinks purchased is held in local storage, initialised to 0
  + each time an order is placed, the number of drinks is increased by the number of drinks in the order
  + when a drink is added to an order, the number of drinks purchased is checked, and if it is 10, the drink is free, and the number of drinks purchased is reset to 0
* read a list of available drinks and a list of milk types from an external JSON file
  + the form elements for the available drinks and milk types should be created based on the data read from the file