Shopping Cart Project

- 1) Create a folder for project
- 2) express --hbs :- to download essential things

  npm install :- to install all dependencies.

  npm install nodemon :- to install nodemon server, that automatically restarts when we make changes to the code.

  we can start the project by

  npm start -

  nodemon bin/www

  or change mode to nodemon in the file "package.json" and type npm start.

Creating/editing "index.hbs" file

From bootstrap, copy the starter template.

- add a navbar.

  edit navbar according to your preference

- After that add cards for items in your shopping site.

- On the <header> ... </header> part add navbar there.

default ഉള്ള -
stylesheet ല് body യി
50px padding ട
കൊടുത്തിട്ടുണ്ടാകും.
If you don't need it
we can remove it.
(when we create
project using
express (--hbs)

* below that header add a new <section> - for listing items
  * ~~add a container~~
  * create a <div class="row"> row.
* ~~create~~ inside that 'row' div. ~~add~~ add another div.<class="col-md-3>
  to list 4 items/row. (3*4 = 12).
* Add a container div~~ide~~ (which helps to give some space on both left & right side), also give 'mt-5' to give ~~top~~ margin-top.

* Manually its impossible to add 100's of cards/lists in a site, so we need to use hbs. (using hbs for loop).

* For that, on `index.js` create an array
  After creating that array, pass it to the index.hbs page.
  After reaching the index.hbs, ~~using the~~ loop display it using loop.

* Setting layout like navbar and other to a common file (because its not practical to create navbar for each and every page, for that it is created in a common file and calls where its needed.

* for that create a layout folder in views folder and inside layout folder create layout.hbs.

* Also create a folder called partials in the views folder for ~~partial layouts~~

* And all the partial layouts are called in ~~the~~ "layout.hbs" and this "layout.hbs" is called in pages wherever needed.

* Then ~~to~~ set template engine in app.js. → hbs.engine

app.engine ($'hbs', hbs.({extname: 'hbs', defaultLayout:

[Don't need to byheart just gg google it.] ← 'layout', layouts Dir: ––dirname + '/views/layout/',
                                             partials Dir: ––dirname + '/views/partials/'}))

* Also install hbs module using npm

  npm install express-handlebars.

* And ~~use~~ use the library by assigning it into the app.js by require().

  var hbs = require('express-handlebars').

~~* We don't need all html~~

* We only need the essential section of the code in files. All other common codes/layouts are written in layout.hbs. and we call the body in that file using {{{ body }}}

* After that the header from layout is removed and individual (partials) header is created for admin & user and then calls it in the layout.hbs.

  '{{ > admin-header }}'

  will display the admin-header, but we need to use ~~the~~ on if-condition to check whether we need to use admin/user header.

* passing a ~~value~~ parameter `admin:true` in index.js to ~~track~~ get the value for checking if condition. (this done just to check the working, actually it is passed from backend)