# MY SHOPPING CART PROJECT

## TOOLS:

### IDE:

* VSCode

## GIT & GITHUB for PROJECT:

…or create a new repository on the command line

echo "# MyShoppingCart" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/kenvng/MyShoppingCart.git

git push -u origin main

…or push an existing repository from the command line

git remote add origin https://github.com/kenvng/MyShoppingCart.git

git branch -M main

git push -u origin main

…or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

### Main Command Line Using Throughout the Project:

* git status
* git add .
* git commit -m “message goes here”
* git push -u origin main

## DEPENDENCIES needed for PROJECT:

Download Node.js

Express-Generator

Nodemon

MongoDB / Mongoose

## STARTING:

### Installing Express Generator:

* TERMINAL (VSCode) – root directory of the project:

TERMINAL: npm install express-generator -g // express-generator is being installed globally -g

Text

Description automatically generated

TERMINAL OUTPUT:

Text

Description automatically generated

TERMINAL: express MyShoppingCart –hbs

Text

Description automatically generated with medium confidence

It created the project folder called MyShoppingCart and installed the following folders and files in project folder you just created (template)

Handlebars – generate the views folders with 3 files that end with the .hbs extension which is short for hanglebars:

* error.hbs – this file handle all the error message in the browser
* index.hbs -
* layout.hbs – hold the default html skeleton – will render the contents in the {{{body}}} part in the file

TERMINAL: cd MyShoppingList

TERMINAL: npm install

TERMINAL: **npm start** // to start the server which is a little script provided by express-generator

And your server is starting with some output as follow:

Text

Description automatically generated

If we refresh the browser @ http://localhosts:3000 , we will get the following message in terminal:



This means our localhosts server is working as intended.

## FILES

## HEADER

Allow user to go to user account signin/logout and shopping cart.

### Bootstrap

Head over to getbootstrap.com…

Grab this link:

<!-- CSS only -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor" crossorigin="anonymous">

and put it in the layout.hbs above the link element in the header.

AND grab this link:

<!-- JavaScript Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2" crossorigin="anonymous"></script>

and put it in the layout.hbs under the {{{body}}} hook

Create a subfolder in views called partials and in partials create a file called header.hbs and add the html below in

Look for component > navbar and copy the html code and put it in the

<https://getbootstrap.com/docs/5.2/components/navbar/#content>

### jQuery

we are also going to utilize some jQuery, so head over to:

<https://releases.jquery.com/> and grab the following link:

<script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>

Add it in layout.hbs below the {{{body}}} hook but above the bootstraps link

### Implementing/including the header.hbs into other files (index.hbs, layout.hbs)

Open another terminal to avoid interrupting the server terminal and type in:

TERMINAL: npm install –save express-handlebars

// this will installing another handlebars templating engine 3rd-party package which offers more features than the buit-in one

Open app.js and created a new variable/const as below:

const expressHbs = require('express-handlebars')

and go down to where the view engine set up and replace the following code:

app.set('views', path.join(\_\_dirname, 'views'));

with:

app.engine('.hbs', expressHbs({defaultLayout: 'layout', extname: '.hbs'}))

and then change some code from below line:

app.set('view engine', 'hbs');

to:

app.set('view engine', '.hbs');

### LAYOUT:

In the view folder… create a subfolder called layouts and in this layout folder move the layout.hbs in there

**layout.hbs**:

below the <body> element and above the {{{body}}} hook type in

{{> header }}

// specifying the partials I was to include – in this case the header

Save and refresh the page… now we can see the html is being populated by bootstrap and **header.hbs**

In this layout.hbs, where the bootstrap html for the header reside which including the navbar with:

* Brand – My Shopping Cart
* Shopping Cart
* User Management
  + User Account
  + Logout

In **views** folder: we create another folder called shop:

mkdir shop

Move index.hbs into shop folder

If you refresh the page it will break.

**routes>index.js**:

Change the path of the index page to:

shop/index

#### Thumbnails

Heading over to <https://getbootstrap.com/css/> , search for thumbnails and select the custom content of Thumbnail label I need

**index.hbs**: copy this code and paste it into this file and replace the current code in there

A screenshot of a computer

Description automatically generated with medium confidence

NOTICE: the display of the content thumbnail is not all that great because we need to add more code on it.

**layout.hbs**: **add the following code below** {{> header }} **and out** {{{body}}} inside this element tab.

**Graphical user interface, text

Description automatically generated**

**index.hbs**: we are going to edit our thumbnails code in this file and css file as well

A screenshot of a computer

Description automatically generated with medium confidence

We are now have the edit div for the thumbnails…. Not only that we created the other two and put them in a div element to create a roll.