HENHOUSE DOCUMENTATION

HTML

DEPENDENCIES

Outside libraries used by the page.

LIBRARY NAME	VERSION	USAGE
Vue.Js	2.6.x	Main Logic, state manager for the page
Jquery	3.6.x	Used as shorthands on some methods inside Vue
Bootstrap	4.0.0	Both js and css are used as grid reference

CSS FILES

Css files created to be used on the page, categorized by device to help further development.

FILE NAME	DESCRIPTION
main.js	JS file containing the vue instance.
main.css	Defines document root styles such as image source(s)
--mobile.css	Defines position and sizes for a few mobile media queries
--hdmobile.css	Defines position and sizes for 'tall' mobile media querie
--tablet.css	Defines position and sizes for tablet, based only on Ipad (Currently)
--laptop.css	Defines position and sizes for laptop and medium desktop screens
--xlarge.css	Defines position and sizes for large desktop screens (>1080p)

ID/CLASSES

Important class names used are listed here to help what identifies what on the page.

ID/CLASS NAME	DESCRIPTION
#app	To be identified by vue as element instance
#warning	To display portrait device warning
.warning-text	Displays the warning text
.bg-top	Scenerey part, houses and more, the rest are on body element
.house	Displays the house as background, objects are separated
.middle-container	This class defines the size of the rest of the objects attached on the house, use it as a frame of reference of where the objects needs to be and how big, since this is placed on the middle of the house (PARENT).
.billboard	Text message on the billboard itself
.door	The door itself
.info-btn	Info button on roof
.left-hatch	The closed left hatch state
.left-hatch-open	The open left hatch state
.right-hatch	The closed right hatch state
.right-hatch-open	The open right hatch state
.lpost	The post on the left "To Opinions" to trigger the left-side nav
.rpost	The post on the right "To Peckingorder" to trigger the right-side nav
.left-side	The left side nav containing Popular Opinions and user navigations
.right-side	The right side nav containing Popular Users
.zoom-left	The opinion zoom window
.zoom-right	The users zoom window
And more classes named appropriately to ease further development	

VUE

SUMMARY

1. There is only one instance created which is app declared on #app.
2. the data object stores the states (Screens, objects, values) used on the app.
3. the strings object which contains bits of strings displayed throughout the page, made for language support to be easily replaced with any language.
4. the categories object contains available categories, subcategories, and topics can be chosen by user, this object Is made to fulfil the demo needs about adding new Subs and Topics.
5. Opinions and Users contains popular opinions and users data by each attribute,
6. Functions of each method are explained above each method
7. The method used to add new Subs and Topics are for **demo purposes**, this means it can be replaced easily by other ways, such as hooking a database to the object itself, using async method to communicate in between and many more.

KNOWN ISSUE

1. **The display is not responsive and does not respond to window resizing very well**, this is caused due to the page tries to contantly keeps the page from 'stretching' and 'skewing' the objects and images inside by maintaining the scale, this could be helped by adding more media queries as needed, but for now the main scaling of object sizes are controlled by the class **.middle-container**, this container as explained briefly above, controls the position and the size of each object on the 'house' (coop).

2. **The position of left/right zoom window slides when chosen in certain position**, this is caused due to the triggering process, to walk trough it, the method first renders the zoom window, then it assigns a position and data into it, and then ***moves*** the zoom window to the desired calculated position, the render process isn't instant and causes the ***sliding*** to be visible.

3. **The added subs and topics are gone on refresh**, this happes because there is no database.