**El-King-stone Project**

**Description**

King stone website Technology has been used in it ( HTML5 – CSS3 – Bootstrap – JavaScript – wow.js library )

**Consists Of**

* 3 files HTML ( Index.html “Home Page” – Service.html – Who-we-are.html )
* 4 files CSS ( animate.css – bootstrap.min.css - bootstrap.min.css.map – main.css )
* 5 files JavaScript ( bootstrap.bundle.min.js - bootstrap.bundle.min.js.map – main.js – service.js – wow.js )
* File for all images at assets/img
* Font family “ Noto Sans Arabic ” Import it from google fonts
* Font awesome

**Main.js file**

In line number one wow.js file init

From line 5 to line 49 Catch Elements from Html files

Start Creating Variable from line 55 to line 71

Line 118 set the counter to start count when user arrive to the section

Window.onscroll to get scrolling line (76)

And make a condition if scroll y to the conter offsetTop – nav bar offset height

“nav bar offset height : to decrease the nav bar height from the counter offset height

To start counter “

And make condition if it started

“Create a Variable Boolean to make it false to still the function not start”

Get all of elements in the counter “catch this element in line 68 with name ‘counterNums’ ” and loop in it to run startCount(every element)

Window.onscroll line(94)

* Toggle class to nav bar to change background

startCount function line (107)

* get every dataset with name goal in every element in goal variable
* start setInterval to count every 10 / goal
* increase the text content of every element in count
* make a condition if the text content of element is = goal of dataset stop the interval
* 10 / goal “Calc the time for every element with his value”

Gallery Data Img and Name Array line (119)

* Array of objects for Gallery section data

Object include { Img URL , Name Of card and Category of Card }

Cards Data Array line (284)

* Array of objects for Service Section

Object include { Card Text “cardName”, Card Description “cardText”, Card Img URL, Card Category }

Filter Gallery Data line (359)

* Pushing all of data of Gallery in the filter array
* To loop in data to be showed (Gallery Section)

New array of Image line (365)

* Pushing all of Image of Gallery in the new array of image
* To loop in data to filter it with shuffle

Gallery Switcher list (Shuffle) line (371)

* We get
* Loop on shuffle buttons and make a filter for every category Image element
* With the button inner HTML
* Make a switch for every case
* case "تشطيب وجهات حجر ابيض ازازاى": // Select the case of Button innerHTML
* filterGalleryData = []; // Make the Array is empty
* filterGalleryData = galleryDataImgAndName.filter((el) => {
  + // Loop in cards to get filter the cards and showed
* return el.name == "تشطيب وجهات حجر ابيض ازازى"; // only data in this category
* });
* changeArrayFrom(filterGalleryData); // pushing the new Array to showed

Change Array from function line (426)

* Take a argument (new array) to take every time with shuffle only cards selected
* Reset the Current Page of Pagination to the first page every time using shuffle
* Update the galleryDisplayList function to the new array was already filtered
* Update the gallerySetupPagination function to the new array was already filtered

removeActive function line (451) “Service Section shuffle”

* Loop in all buttons of shuffle
* Remove all active class from it
* Add class active to this “element has clicked”

Filter of service data “pushing all data from default array of objects “ line (461)

service Switcher list (Shuffle) line (467)

* Loop on shuffle buttons and make a filter for every category service cards
* With the button inner HTML
* Make a switch for every case
* case "exterior-design": // Select the case of Button innerHTML
* filterOfServiceData = cardsData.filter((el) => {
* filterOfServiceData = []; // make the array empty to get only new elements
* return el.category == "exterior-design"; filter the old array
* });
* serviceShuflleHanlde(filterOfServiceData); // set data with new array

removeActive function line (509) (service section)

* Select all of buttons in shuffle and loop in it
* Remove all class active from all of buttons
* Add class active to the button click

manageScreenWith function line (519)

* Take a argument x = “Window.matchMedia”
* Make a condition if x.matchs = true ? make the elements “Images”in gallery 6 elements “Small Screen”

If it false ? make the elements “Images”in gallery 9 element “Large Screen”

galleryDisplayList function line(533)

* Take a four argument

(

items = “Array of data and image come from galleryDataImgAndName or filterGalleryData after shuffle buttons clicked ”,

wrapper = “Parent Element will append child in it ‘galleryImgParent element catch it line (21)’ ”,

elements\_per\_page = “number of elements will load for every page in pagination”,

page = “Gallery pagination current page”

)

* In first make the wrapper empty to take the new elements only
* Page-- = “to start from first page in pagination ‘0’”
* Loopstart = elements load in the page \* current page
* LoopEnd = number from loop start + elements load in the page
* paginationItems = “Array of my data . slice(loopstart, loopEnd)” to get array for only elements will load for every page
* loop in data to showed with 4 elements ( ParentOf elements = “article”

Overlay = “div”

Img of element = “img”

Category name = “span”

)

gallerySetupPagination function line(582)

* Take a three argument

(

items = “Array of data and image”,

wrapper = “Parent Element will append child in it ”,

elements\_per\_page = “number of elements will load for every page in pagination”,

)

* First make the wrapper empty to take only new elements
* Get the number pageCount = use Math.ceil(items.length / elements\_per\_page ) to get number of pages
* Loop to append all of buttons

PaginationButtons function (Gallery section) line (595)

* Take a two argument

(

Page = “numbr of pages in pagination

Items = “array data and image”

)

* Button text = page number
* Add to button his class’s
* Make a condition if current page = page add active class
* Add a click listener to all of buttons with argument (e)

Call toggle active class function and take two parameters ( button will clicked, all of buttons element)

Call galleryDisplayList and chage array in it to load new data

Call galleryDisplayList function and gallerySetupPagination function to get the default array of data line (625) to line (632)

imageView “Popup of image” (Gallery Section) add click listener (call function closePopUp) line (640)

exitBtn “exit button of popup” (Gallery Section) add click listener (call function closePopUp) line(644)

closePopUp function line (648)

* Make image parent and overlay display none

nextBtn “Next Button on PopUp ” (Gallery Section) add a click listener line (658)

* Make a condition if counter of image index in array = array of image length make the counter = 0 to return from first image of array
* Else make a counter + 1
* Call imgChangeHandle function

prevBtn “prev Button on PopUp ” (Gallery Section) add a click listener line (667)

* Make a condition if counter of image index in array = 0make the counter = array of image length to return from last image of array
* Else make a counter - 1
* Call imgChangeHandle function

imgChangeHandle function line (677)

* Change image of popup url with the new array of image “ it change with shuffle” [counter] to get a new image with new index
* Set background position = center
* Set background size = cover
* +1 to counter of count element
* Set to new image array length to imagelength.innerHTML

Stop the span “text of image” stop element clicked line (691)

document.querySelectorAll(".gallery .gallery-item span").forEach((el) => {

  el.addEventListener("click", (e) => {

    e.preventDefault();

  });

});

serviceDisplayList function line (699)

* Take a four argument

(

items = “Array of data and image”,

wrapper = “Parent Element will append child in it ”,

elements\_per\_page = “number of elements will load for every page in pagination”,

page = “Service pagination current page”

)

* In first make the wrapper empty to take the new elements only
* Page-- = “to start from first page in pagination ‘0’”
* Loopstart = elements load in the page \* current page
* LoopEnd = number from loop start + elements load in the page
* paginationItems = “Array of my data . slice(loopstart, loopEnd)” to get array for only elements will load for every page
* loop in data to showed with 4 elements ( ParentOf elements (card) = “section”,

cardImgParent = “div” image parent of card,

cardImg = “Img”,

card text (cardInfoParent) = “div”,

head of text (h3) = “h3”,

paragh of text (p) = “p”,

append child to parent of cards

setupServicePagination function line (759)

* Take a three argument

(

items = “Array of data”,

wrapper = “Parent Element will append child in it ”,

elements\_per\_page = “number of elements will load for every page in pagination”,

)

* First make the wrapper empty to take only new elements
* Get the number pageCount = use Math.ceil(items.length / elements\_per\_page ) to get number of pages
* Loop to append all of buttons

paginationServiceButtons function line (775) “service section”

* Take a two argument

(

Page = “numbr of pages in pagination

Items = “array data and image”

)

* Button text = page number
* Add to button his class’s
* Make a condition if current page = page add active class
* Add a click listener to all of buttons with argument (e)

Call toggle active class function and take two parameters ( button will clicked, all of buttons element)

Call serviceDisplayList and chage array in it to load new data

serviceShuffleHandle function line (805)

* Take a argument (new array) to take every time with shuffle only cards selected
* Reset the Current Page of Pagination to the first page every time using shuffle
* Update the serviceDisplayList function to the new array was already filtered
* Update the setupServicePagination function to the new array was already filtered

pagiationToggleButtonsActiveClass function line (824)

* Take a two argument (

targetElement = “button clicked”,

buttons = “array of buttons”

)

* Loop in buttons remove all active class from all
* Add active class to button clicked

callUsBtnToggle = ”buttons in the right side have all of contact” add click listener line (833)

* Loop in all buttons toggle class show to change his position
* Add to the message button class show to change I class

**Service.js**

From line 5 to line 22 Catch Elements from Html files

Start Creating Variable from line 26 to line 30

Cards Data Array line (34)

* Array of objects for Service Section

Object include { Card Text “cardName”, Card Description “cardText”, Card Img URL, Card Category }

Window.onscroll line(108)

* Toggle class to nav bar to change background

service Switcher list (Shuffle) line (126)

* Loop on shuffle buttons and make a filter for every category service cards
* With the button inner HTML
* Make a switch for every case
* case "exterior-design": // Select the case of Button innerHTML
* filterOfServiceData = cardsData.filter((el) => {
* filterOfServiceData = []; // make the array empty to get only new elements
* return el.category == "exterior-design"; filter the old array
* });
* serviceShuflleHanlde(filterOfServiceData); // set data with new array

removeActive function line (165)

* Select all of buttons in shuffle and loop in it
* Remove all class active from all of buttons
* Add class active to the button click

serviceDisplayList function line (175)

* Take a four argument

(

items = “Array of data and image”,

wrapper = “Parent Element will append child in it ”,

elements\_per\_page = “number of elements will load for every page in pagination”,

page = “Service pagination current page”

)

* In first make the wrapper empty to take the new elements only
* Page-- = “to start from first page in pagination ‘0’”
* Loopstart = elements load in the page \* current page
* LoopEnd = number from loop start + elements load in the page
* paginationItems = “Array of my data . slice(loopstart, loopEnd)” to get array for only elements will load for every page
* loop in data to showed with 4 elements ( ParentOf elements (card) = “section”,

cardImgParent = “div” image parent of card,

cardImg = “Img”,

card text (cardInfoParent) = “div”,

head of text (h3) = “h3”,

paragh of text (p) = “p”,

append child to parent of cards

setupServicePagination function line (235)

* Take a three argument

(

items = “Array of data”,

wrapper = “Parent Element will append child in it ”,

elements\_per\_page = “number of elements will load for every page in pagination”,

)

* First make the wrapper empty to take only new elements
* Get the number pageCount = use Math.ceil(items.length / elements\_per\_page ) to get number of pages
* Loop to append all of buttons

paginationServiceButtons function line (251)

* Take a two argument

(

Page = “numbr of pages in pagination

Items = “array data and image”

)

* Button text = page number
* Add to button his class’s
* Make a condition if current page = page add active class
* Add a click listener to all of buttons with argument (e)

Call toggle active class function and take two parameters ( button will clicked, all of buttons element)

Call serviceDisplayList and chage array in it to load new data

serviceShuffleHandle function line (281)

* Take a argument (new array) to take every time with shuffle only cards selected
* Reset the Current Page of Pagination to the first page every time using shuffle
* Update the serviceDisplayList function to the new array was already filtered
* Update the setupServicePagination function to the new array was already filtered

pagiationToggleButtonsActiveClass function line (300)

* Take a two argument (

targetElement = “button clicked”,

buttons = “array of buttons”

)

* Loop in buttons remove all active class from all
* Add active class to button clicked

callUsBtnToggle = ”buttons in the right side have all of contact” add click listener line (310)

* Loop in all buttons toggle class show to change his position
* Add to the message button class show to change I class