

```

1  /*-----
2  :*
3  :*
4  :*
5  :*
6  :*
7  :*
8  :* Archivo      : front.js
9  :* Autor        : Equipo de desarrollo Cv Software Studio
10 :* Fecha         : 18/Octubre/2021
11 :* Compilador    :
12 :* Descripcion   : Obtine los datos de Firebase y los muestra
13 :* Ultima modif  : 17/11/2021
14 :* Fecha         Modificacion      Motivo
15 :* 11/10/2021    Search Box         Creacion del DOM para inicar seccion
16 :* 14/10/2021    Tooltips init      Creacion de Dom para cerrar seccion
17 :* 15/10/2021    Sidebar Functionality Funcion para cosultar si hay seccion
18 :* 15/10/2021    Footer             Funcion para colocar html con datos
19 :*                                     Firebase.
20 :* 16/10/2021    Material Inputs     Extrea el los valores de Tem. de Fire
21 :* 16/10/2021    Profile page choices Extrea el los valores de Oxi. de Fire
22 :* 16/10/2021    getCardiaco         Extrea el los valores de Car. de Fire
23 :* 17/10/2021    onSnapshot          Actualiza los documentos
24 :*=====
25 :*
26 :*-----*/
27
28 'use strict';
29
30 document.addEventListener('DOMContentLoaded', function () {
31
32     // ----- //
33     // Search Box
34     // ----- //
35     const navBar = document.querySelector('.navbar');
36     const searchBtn = document.getElementById('search'),
37           searchBox = document.querySelector('.search-box'),
38           searchClose = document.querySelector('.dismiss');
39
40     if (navBar) {
41         searchBtn.addEventListener('click', (e) => {
42             e.preventDefault();
43             searchBox.classList.add('fadeIn');
44         });
45
46         searchClose.addEventListener('click', () =>
47 searchBox.classList.remove('fadeIn'));
48     }
49     // ----- //
50     // Tooltips init
51     // ----- //
52     var tooltipTriggerList = [].slice.call(document.querySelectorAll('[data-bs-
53 toggle="tooltip"]'))
54     var tooltipList = tooltipTriggerList.map(function (tooltipTriggerEl) {
55         return new bootstrap.Tooltip(tooltipTriggerEl)
56     });
57     // ----- //
58     // Sidebar Functionality
59     // ----- //

```

```

58  const sbToggleBtn = document.getElementById('toggle-btn'),
59      sideNavbar    = document.querySelector('.side-navbar'),
60      innerContent   = document.querySelector('.content-inner'),
61      smBrand        = document.querySelector('.navbar-header .brand-small'),
62      lgBrand        = document.querySelector('.navbar-header .brand-big');
63
64  if (sideNavbar) {
65      sbToggleBtn.addEventListener('click', function (e) {
66          e.preventDefault();
67          this.classList.toggle('active');
68
69          sideNavbar.classList.toggle('shrunked');
70          innerContent.classList.toggle('active');
71          document.dispatchEvent(new Event('sidebarChanged'));
72
73      });
74  }
75  // ----- //
76  // Footer
77  // ----- //
78  let footer = document.querySelector('#footer');
79  if (footer) {
80      document.addEventListener('sidebarChanged', function () {
81          adjustFooter();
82      });
83      window.addEventListener('resize', function () {
84          adjustFooter();
85      });
86  }
87
88  function adjustFooter() {
89      var footerBlockHeight = document.querySelector('#footer').outerHeight;
90      innerContent.style.paddingBottom = `${footerBlockHeight}px`;
91  }
92  // ----- //
93  // External links to new window
94  // ----- //
95  document.querySelectorAll('.external').forEach((el) => {
96      el.addEventListener('click', function (e) {
97          e.preventDefault();
98          window.open(el.getAttribute('href'));
99      });
100  });
101  // ----- //
102  // Material Inputs
103  // ----- //
104
105  let materialInputs = document.querySelectorAll('input.input-material');
106  let materialLabel = document.querySelectorAll('label.label-material');
107
108  // activate labels for prefilled values
109  let filledMaterialInputs = Array.from(materialInputs).filter(function (input) {
110      return input.value !== '';
111  });
112  filledMaterialInputs.forEach(input =>
input.parentElement.lastElementChild.setAttribute('class', 'label-material active'));
113
114  // move label on focus
115  materialInputs.forEach((input) => {
116      input.addEventListener('focus', function () {

```

```
117     input.parentElement.lastElementChild.setAttribute('class', 'label-  
material active');  
118   });  
119 });  
120  
121 // remove/keep label on blur  
122 materialInputs.forEach((input) => {  
123   input.addEventListener('blur', function () {  
124     if (input.value !== '') {  
125       input.parentElement.lastElementChild.setAttribute('class', 'label-  
material active');  
126     } else {  
127       input.parentElement.lastElementChild.setAttribute('class', 'label-  
material');  
128     }  
129   });  
130 });  
131  
132  
133 function bsValidationBehavior(errorInputs, form) {  
134   function watchError() {  
135     errorInputs.forEach((input) => {  
136       if (input.classList.contains('js-validate-error-field')) {  
137         input.classList.add('is-invalid');  
138         input.classList.remove('is-valid');  
139       } else {  
140         input.classList.remove('is-invalid');  
141         input.classList.add('is-valid');  
142       }  
143     });  
144   }  
145   watchError();  
146 }  
147 // ----- //  
148 // Login Form Validation  
149 // ----- //  
150 let loginForm = document.querySelector('.login-form');  
151 if (loginForm) {  
152   new window.JustValidate('.login-form', {  
153     rules: {  
154       loginUsername: {  
155         required: true,  
156         email: true  
157       },  
158       loginPassword: {  
159         required: true,  
160       },  
161     },  
162     messages: {  
163       loginUsername: 'Please enter a valid email',  
164       loginPassword: 'Please enter your password'  
165     },  
166     invalidFormCallback: function () {  
167       let errorInputs = document.querySelectorAll('.login-form  
input[required]');  
168       bsValidationBehavior(errorInputs, loginForm);  
169       loginForm.addEventListener('keyup', () =>  
bsValidationBehavior(errorInputs, loginForm))  
170     },  
171   });
```

```

172     }
173     // ----- //
174     // Register Form Validation
175     // ----- //
176     let registerForm = document.querySelector('.register-form');
177     if (registerForm) {
178         new window.BootstrapValidator('.register-form', {
179             rules: {
180                 registerUsername: {
181                     required: true
182                 },
183                 registerEmail: {
184                     required: true,
185                     email: true
186                 },
187                 registerPassword: {
188                     required: true,
189                 },
190                 registerAgree: {
191                     required: true,
192                 },
193             },
194             messages: {
195                 registerUsername: 'Please enter your username',
196                 registerEmail: 'Please enter a valid email address',
197                 registerPassword: 'Please enter your password',
198                 registerAgree: 'Your agreement is required'
199             },
200             invalidFormCallback: function () {
201                 let errorInputs = document.querySelectorAll('.register-form
input[required]');
202                 bsValidationBehavior(errorInputs, registerForm);
203                 registerForm.addEventListener('keyup', () =>
bsValidationBehavior(errorInputs, registerForm))
204                 registerForm.addEventListener('change', () =>
bsValidationBehavior(errorInputs, registerForm))
205             },
206         });
207     }
208     // ----- //
209     // Profile page choices
210     // ----- //
211     function injectClassess(x) {
212         let pickerCustomClass = x.dataset.customclass;
213         let pickerSevClasses = pickerCustomClass.split(' ');
214         x.parentElement.classList.add.apply(x.parentElement.classList,
pickerSevClasses);
215     }
216     const profileCountryChoices = document.querySelector('.profile-country-choices');
217     if (profileCountryChoices) {
218         const countryChoices = new Choices(profileCountryChoices, {
219             searchEnabled: false,
220             placeholder: false,
221             callbackOnInit: () => injectClassess(profileCountryChoices)
222         });
223     }
224
225 });
226

```