

# Folder system

27 printable files

(file list disabled)

system\backend\class\cms.php

```
1  <?php
2      class cms {
3          /* =====
4          Ansicht
5          ===== */
6
7          /* =====
8          Print
9          ===== */
10
11         public static function print($mData) {
12             echo '<pre>';
13             print_r($mData);
14             echo '</pre>';
15         }
16
17         /* =====
18         Dump
19         ===== */
20
21         public static function dump($mData) {
22             echo '<pre>';
23             var_dump($mData);
24             echo '</pre>';
25         }
26
27         /* =====
28         Icon
29         ===== */
30
31         public static function getIcon ($sIcon, $sStyle = 'solid', $sClass = '') {
32             ob_start();
33             echo '<i class="fa-' . $sStyle . ' fa-' . $sIcon . ' ' . $sClass . '"></i>';
34             return ob_get_clean();
35         }
36
37         /* =====
38         Wertumwandlung
39         ===== */
40
41         public static function convertValue ($mValue, $sType, $aOptions = []) {
42             switch ($sType) {
43                 case 'bool':
44                     if ($mValue) {
45                         return cms::getIcon('check');
46                     } else {
47                         return cms::getIcon('x');
48                     }
49                 break;
```

```

50         case 'price':
51             return number_format($mValue, 2, ',', '.') . ' ' . ($aOptions['unit'] ?? '');
52             break;
53     }
54 }
55
56 /* =====
57 Bild Rückgabe
58 ===== */
59
60 public static function img ($sName, $sPath = 'img', $iWidth = 25) {
61     $sFilePath = dirname(__DIR__, 2) . "/data/$sPath/$sName";
62     if (is_file($sFilePath)) {
63         return '';
65     } else {
66         return '';
67     }
68 }
69
70 /* =====
71 Sicherheit
72 ===== */
73
74 /* =====
75 Ist Eingeloggd
76 ===== */
77
78 public static function isLoggedIn ($bLoginPage = false) {
79     $bResult = false;
80     $sPreviousSite = "";
81
82     if (isset($_GET['site'])) {
83         $sPreviousSite = $_GET['site'];
84     }
85
86     if (isset($_SESSION['email'])) {
87         $bResult = true;
88     } else {
89         $bResult = false;
90     }
91
92     if ($bLoginPage && !$bResult) {
93         if ($sPreviousSite == "") {
94             header('Location: ?site=login&err=1');
95         } else {
96             header('Location: ?site=login&err=1&prev=' . $sPreviousSite);
97         }
98
99         die();
100     }
101
102     return $bResult;
103 }
104
105 /* =====
106 Daten

```

```

107 | ===== */
108 |
109 | /* =====
110 | In Array suchen
111 | ===== */
112 |
113 | public static function searchArray ($aData, $sKey, $sSearchKey) {
114 |     $aResult = [];
115 |
116 |     foreach ($aData AS $aValues) {
117 |         if ($aValues[$sKey] == $sSearchKey) {
118 |             $aResult[] = $aValues;
119 |         }
120 |     }
121 |
122 |     return $aResult;
123 | }
124 | }

```

#### system\backend\class\db.php

```

1  <?php
2  class db {
3      public $oConnect;
4      public $sQuery;
5      public $aParam;
6      public $oResult;
7
8      /* =====
9      Konstruktor
10     ===== */
11
12     public function __construct()
13     {
14         $sHost      = "45.91.102.172";
15         $iPort      = 3306;
16         $sUser      = "m_fassbender_onecue_de_web";
17         $sPassword  = "Ieb24Qg4DwhE5qPjejlh";
18         $sDatabase  = "m_fassbender_onecue_de_Tap2Table";
19         $this->oConnect = new mysqli($sHost, $sUser, $sPassword, $sDatabase, $iPort);
20
21         if ($this->oConnect->connect_error) {
22             throw new \Exception('Database Error: ' . $this->oConnect->connect_error);
23         }
24     }
25
26     /* =====
27     Daten
28     ===== */
29
30     /* =====
31     Abfrage
32     ===== */
33
34     public function query ($sQuery, $aParam = []) {
35         $this->sQuery = $sQuery;
36         $this->aParam = $aParam;
37     }

```

```

38
39
40 /* =====
41 Ausführen
42 ===== */
43
44 public function execute ($bSafeResult = false) {
45     if ($stm = $this->oConnect->prepare($this->sQuery)) {
46         if (!empty($this->aParam)) {
47             $aParam = [];
48
49             foreach ($this->aParam AS $iKey => &$mValue) {
50                 $aParam[] = &$mValue;
51             }
52             call_user_func_array([$stm, 'bind_param'], $aParam);
53         }
54
55         $stm->execute();
56         mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
57
58         if ($bSafeResult) {
59             $this->oResult = $stm->get_result();
60         }
61     }
62 }
63
64
65 /* =====
66 Daten erhalten
67 ===== */
68
69 public function get ($mMode = MYSQLI_NUM) {
70     if (!empty($this->oResult)) {
71         $aReturn = $this->oResult->fetch_all($mMode);
72         return $aReturn;
73     } else {
74         return [];
75     }
76 }
77 }

```

#### system\backend\class\pages.php

```

1 <?php
2 $sPage = $_GET['site'] ?? 'home';
3 $sPath = dirname(__DIR__, 2) . '/frontend/pages/' . $sPage . '.php';
4 if (is_file($sPath)) {
5     include($sPath);
6 } else {
7     echo '<div class="container d-flex justify-content-center align-items-center min-vh-
100">';
8         $oAlert = new snippetAlert('danger');
9         $oAlert->setIcon('circle-info');
10        $oAlert->setTitle('Fehler');
11        $oAlert->setContent('Diese seite wurde nicht gefunden');
12        $oAlert->setClasses('text-center');
13        echo $oAlert->view();
14    echo '</div>';
15 }

```

## system\backend\include.php

```
1 <?php
2 /* =====
3 Includes
4 ===== */
5
6 /* =====
7 class
8 ===== */
9
10 require_once('./system/backend/class/cms.php');
11 require_once('./system/backend/class/db.php');
12
13 /* =====
14 Snippets
15 ===== */
16
17 $sPath = './system/backend/snippets/';
18 $aFiles = scandir($sPath);
19 foreach ($aFiles AS $sFile) {
20     if (!is_file($sPath . $sFile)) continue;
21     str_starts_with($sFile, '_');
22     require_once($sPath . $sFile);
23 }
```

## system\backend\js\check-update.js

```
1 $(document).ready(function () {
2     let saveBar = $('#saveBar');
3     // saveBar.removeClass('d-none');
4     saveBar.fadeOut(0);
5
6
7     $('.in-menu').change(function() {
8         $.ajax({
9             url: '/ajax/inventory.php',
10            type: 'POST',
11            dataType: 'json',
12            data: {
13                id: $(this).data('item-id'),
14                value: $(this).is(':checked') ? 1 : 0,
15                action_type: 'menu'
16            },
17            success: function (res) {
18                const Toast = Swal.mixin({
19                    toast: true,
20                    position: "bottom-end",
21                    showConfirmButton: false,
22                    timer: 3000,
23                    timerProgressBar: true,
24                    theme: 'dark',
25                    didOpen: (toast) => {
26                        toast.onmouseenter = Swal.stopTimer;
27                        toast.onmouseleave = Swal.resumeTimer;
28                    }
29                });
30                if (res.success) {
31                    Toast.fire({
```

```

32         icon: "success",
33         title: "Der Status wurde Geändert"
34     });
35     } else {
36         Toast.fire({
37             icon: "warning",
38             title: "Es ist ein Fehler aufgetreten"
39         });
40     }
41 },
42 error: function (xhr) {
43     console.error('✖ Serverfehler', xhr.responseText);
44 }
45 });
46 })
47
48 $(' .sold-out').change(function() {
49     $.ajax({
50         url: '/ajax/inventory.php',
51         type: 'POST',
52         dataType: 'json',
53         data: {
54             id: $(this).data('item-id'),
55             value: $(this).is(':checked') ? 1 : 0,
56             action_type: 'sold_out'
57         },
58         success: function (res) {
59             const Toast = Swal.mixin({
60                 toast: true,
61                 position: "bottom-end",
62                 showConfirmButton: false,
63                 timer: 3000,
64                 timerProgressBar: true,
65                 theme: 'dark',
66                 didOpen: (toast) => {
67                     toast.onmouseenter = Swal.stopTimer;
68                     toast.onmouseleave = Swal.resumeTimer;
69                 }
70             });
71             if (res.success) {
72                 Toast.fire({
73                     icon: "success",
74                     title: "Der Status wurde Geändert"
75                 });
76             } else {
77                 Toast.fire({
78                     icon: "warning",
79                     title: "Es ist ein Fehler aufgetreten"
80                 });
81             }
82         },
83         error: function (xhr) {
84             console.error('✖ Serverfehler', xhr.responseText);
85         }
86     });
87 })
88
89 //-----

```

```

90 // Neuer Artikel
91 //-----
92
93 $('#newArticle').click(function() {
94     Swal.fire({
95         title: '<h2 class="text-uppercase">Neuen Artikel anlegen</h2>',
96         html: `
97             <form>
98             <hr>
99             <div class="input-group w-100 mb-3">
100                 <span class="input-group-text w-25">Name</span>
101                 <input id="name" type="text" class="form-control w-75" required>
102             </div>
103             <div class="input-group w-100 mb-3">
104                 <span class="input-group-text w-25">SKU</span>
105                 <input id="sku" type="text" class="form-control w-75" required>
106             </div>
107             <div class="input-group w-100 mb-3">
108                 <span class="input-group-text w-25">Anzahl</span>
109                 <input id="quantity" type="number" class="form-control w-75" min="0">
110             </div>
111             <div class="input-group w-100 mb-3">
112                 <span class="input-group-text w-25">Gruppen ID:</span>
113                 <input id="group-id" type="number" class="form-control w-75">
114             </div>
115             <div class="input-group w-100 mb-3">
116                 <span class="input-group-text w-25">Preis</span>
117                 <input id="price" type="number" class="form-control w-50" min="0" step="0.1"
required>
118                 <span class="input-group-text">€</span>
119             </div>
120             <div class="form-check form-switch mb-3 d-flex justify-content-between">
121                 <label class="form-check-label">Ausverkauft</label>
122                 <input id="sold_out" class="form-check-input" type="checkbox">
123             </div>
124             <div class="form-check form-switch mb-3 d-flex justify-content-between">
125                 <label class="form-check-label">Im Menü</label>
126                 <input id="menu" class="form-check-input" type="checkbox" checked>
127             </div>
128             <div class="input-group mb-3">
129                 <span class="input-group-text">Bild</span>
130                 <input id="image" type="file" class="form-control">
131             </div>
132             </form>
133         `,
134         confirmButtonText: 'Hinzufügen',
135         showCancelButton: true,
136         cancelButtonText: 'Abbrechen',
137         focusConfirm: false,
138         theme: 'dark',
139         preConfirm: () => {
140             const nameEl      = document.getElementById('name');
141             const skuEl       = document.getElementById('sku');
142             const priceEl     = document.getElementById('price');
143             const groupEl     = document.getElementById('group-id');
144             const quantityEl  = document.getElementById('quantity');
145             const soldOutEl   = document.getElementById('sold_out');
146             const menuEl      = document.getElementById('menu');

```

```

147     const imageEl      = document.getElementById('image');
148
149     const name  = nameEl.value.trim();
150     const sku   = skuEl.value.trim();
151     const price = priceEl.value;
152     const group = groupEl.value;
153
154     // 🟡 Required-Validierung
155     if (!name) {
156         Swal.showValidationMessage('Bitte einen Namen eingeben');
157         nameEl.focus();
158         return false;
159     }
160
161     if (!sku) {
162         Swal.showValidationMessage('Bitte eine SKU eingeben');
163         skuEl.focus();
164         return false;
165     }
166
167     if (!price || price <= 0) {
168         Swal.showValidationMessage('Bitte einen gültigen Preis eingeben');
169         priceEl.focus();
170         return false;
171     }
172
173     // 📦 FormData aufbauen
174     const formData = new FormData();
175     formData.append('action_type', 'add');
176     formData.append('name', name);
177     formData.append('sku', sku);
178     formData.append('quantity', quantityEl.value || 0);
179     formData.append('price', price);
180     formData.append('group-id', group);
181     formData.append('sold_out', soldOutEl.checked ? 1 : 0);
182     formData.append('menu', menuEl.checked ? 1 : 0);
183
184     // 🖼️ Bild nur anhängen, wenn vorhanden
185     if (imageEl.files && imageEl.files.length > 0) {
186         formData.append('image', imageEl.files[0]);
187     }
188
189     // 🚀 Absenden
190     return fetch('/ajax/inventory.php', {
191         method: 'POST',
192         body: formData
193     })
194     .then(response => {
195         if (!response.ok) {
196             throw new Error('Serverfehler');
197         }
198         return response.json();
199     })
200     .catch(err => {
201         console.error(err);
202         Swal.showValidationMessage('Fehler beim Speichern');
203     });
204 }

```

```

205
206     }).then((result) => {
207         if (result.value.success) {
208             location.reload();
209             Swal.fire('Erfolg', 'Artikel wurde angelegt', 'success');
210         }
211     });
212 })
213
214 /* =====
215 Neue Gruppe
216 ===== */
217
218 $(' .new-group').click(function() {
219     Swal.fire({
220         title: '<h2 class="text-uppercase">Neue Gruppe anlegen</h2>',
221         html: `
222             <form>
223                 <hr>
224                 <div class="input-group w-100">
225                     <span class="input-group-text w-25">Name</span>
226                     <input id="group-name" type="text" class="form-control w-75" required>
227                 </div>
228             </form>
229         `,
230         confirmButtonText: 'Hinzufügen',
231         showCancelButton: true,
232         cancelButtonText: 'Abbrechen',
233         focusConfirm: false,
234         theme: 'dark',
235         preConfirm: () => {
236             const nameEl = document.getElementById('group-name');
237             const name = nameEl.value.trim();
238
239             // 🚫 Required-Validierung
240             if (!name) {
241                 Swal.showValidationMessage('Bitte einen Namen eingeben');
242                 nameEl.focus();
243                 return false;
244             }
245
246             // 📦 FormData aufbauen
247             const formData = new FormData();
248             formData.append('action_type', 'add-group');
249             formData.append('name', name);
250
251             // 🚀 Absenden
252             return fetch('/ajax/inventory.php', {
253                 method: 'POST',
254                 body: formData
255             })
256                 .then(response => {
257                     if (!response.ok) {
258                         throw new Error('Serverfehler');
259                     }
260                     return response.json();
261                 })
262                 .catch(err => {

```

```

263         console.error(err);
264         Swal.showValidationMessage('Fehler beim Speichern');
265     });
266 }
267
268 }).then((result) => {
269     if (result.value.success) {
270         Swal.fire('Erfolg', 'Artikel wurde angelegt', 'success');
271     }
272 });
273 })
274
275 /* =====
276 Speisekarte Einstellung
277 ===== */
278
279 let changes    = [];
280 let defaults  = [];
281
282 $('<div>.menu-setting').change(function () {
283     saveBar.fadeIn();
284
285     var Value = null;
286     let key = $(this).data('setting');
287
288     if ($(this).attr('type') == 'checkbox') {
289         Value = $(this).is(':checked') ? 1 : 0;
290     } else {
291         Value = $(this).val();
292     }
293
294     changes[key] = Value;
295     console.log(changes);
296 })
297
298 // $('<div>.menu-setting').change(function() {
299 //     saveBar.fadeIn();
300 //     var Value = null;
301 //     if ($(this).attr('type') == 'checkbox') {
302 //         Value = $(this).is(':checked') ? 1 : 0;
303 //     } else {
304 //         Value = $(this).val();
305 //     }
306
307 //     $.ajax({
308 //         url: '/ajax/menu.php',
309 //         type: 'POST',
310 //         dataType: 'json',
311 //         data: {
312 //             value: Value,
313 //             action: $(this).data('setting'),
314 //             k_id: $(this).data('k_id')
315 //         },
316 //         success: function (res) {
317 //             const Toast = Swal.mixin({
318 //                 toast: true,
319 //                 position: "bottom-end",
320 //                 showConfirmButton: false,

```

```

321 //         timer: 3000,
322 //         timerProgressBar: true,
323 //         theme: 'dark',
324 //         didOpen: (toast) => {
325 //             toast.onmouseenter = Swal.stopTimer;
326 //             toast.onmouseleave = Swal.resumeTimer;
327 //         }
328 //     });
329 //     if (res.success) {
330 //         Toast.fire({
331 //             icon: "success",
332 //             title: "Die Änderung wurde gespeichert"
333 //         });
334 //     } else {
335 //         Toast.fire({
336 //             icon: "warning",
337 //             title: "Es ist ein Fehler aufgetreten"
338 //         });
339 //     }
340 // },
341 //     error: function (xhr) {
342 //         console.error('✖ Serverfehler', xhr.responseText);
343 //     }
344 // });
345 // })
346
347 /* =====
348 Speicher leiste
349 ===== */
350
351 /* =====
352 Speichern
353 ===== */
354
355 /* =====
356 Verwerfen
357 ===== */
358
359 $('#cancelBtn').click(function () {
360     changes = [];
361
362     saveBar.fadeOut();
363 })
364
365 $('.menu-setting-select').change(function() {
366     var Value = $(this).val();
367 })
368
369 //-----
370 // Artikel Löschen
371 //-----
372
373 $('.del-article').click(function () {
374     let article = $(this).closest('tr');
375
376     Swal.fire({
377         title: "Artikel Löschen?",
378         text: "Möchtest du diesen Artikel wirklich entfernen?",

```

```

379     icon: "warning",
380     showCancelButton: true,
381     confirmButtonColor: "#3085d6",
382     cancelButtonColor: "#d33",
383     confirmButtonText: "Yes, delete it!"
384   }).then((result) => {
385     if (result.isConfirmed) {
386       $.ajax({
387         url: '/ajax/inventory.php',
388         type: 'POST',
389         dataType: 'json',
390         data: {
391           del_id: $(this).data('del_id'),
392           action_type: 'remove'
393         },
394         success: function (res) {
395           const Toast = Swal.mixin({
396             toast: true,
397             position: "bottom-end",
398             showConfirmButton: false,
399             timer: 3000,
400             timerProgressBar: true,
401             theme: 'dark',
402             didOpen: (toast) => {
403               toast.onmouseenter = Swal.stopTimer;
404               toast.onmouseleave = Swal.resumeTimer;
405             }
406           });
407           if (res.success) {
408             Toast.fire({
409               icon: "success",
410               title: "Der Artikel wurde entfernt"
411             });
412             article.remove();
413           } else {
414             Toast.fire({
415               icon: "warning",
416               title: "Es ist ein Fehler aufgetreten"
417             });
418           }
419         },
420         error: function (xhr) {
421           console.error('✖ Serverfehler', xhr.responseText);
422         }
423       });
424     }
425   });
426 });
427
428 //-----
429 // Bild ausgrauen
430 //-----
431
432 $('.article-card').each(function () {
433   let ArticleImg = $(this).find('img');
434   let ArticeInput = $(this).find('input');
435
436   if (ArticeInput.attr('disabled')) {

```

```

437         ArticleImg.css('filter', 'grayscale(100%)');
438     }
439 });
440 });

```

#### system\backend\js\counter.js

```

1  const counters = document.querySelectorAll('.count');
2  const speed = 200;
3  counters.forEach((counter) => {
4      const updateCount = () => {
5          const target = parseInt(counter.getAttribute('data-target'));
6          const count = parseInt(counter.innerText);
7          const increment = Math.trunc(target / speed);
8          if (count < target) {
9              counter.innerText = count + increment;
10             setTimeout(updateCount, 1);
11         } else {
12             counter.innerText = target;
13         }
14     };
15     updateCount();
16 });
17
18 $(document).ready(function () {
19     $('.counter').each(function () {
20         const updateCount = () => {
21             let span = $(this).find('span');
22             let target = parseInt(span.data('target'));
23             let current = parseInt(span.text());
24             let inc = Math.trunc(target / 10);
25
26             if (current < target) {
27                 span.text(current + inc);
28                 setTimeout(updateCount, 100);
29             } else {
30                 span.text(target);
31             }
32         }
33         updateCount();
34     });
35 });

```

#### system\backend\js\navbar.js

```

1  //-----
2  // Position des Main Containers anpassen
3  //-----
4
5  let navbar = $('nav');
6  let main = $('main');
7  let distance = 10;
8  let navheight = navbar.outerHeight();
9  let windowheight = $(window).height();
10 let footerheight = $('footer').height();
11
12 main.css('margin-top', navheight + distance);
13 main.css('height', windowheight - navheight - (distance + 1))
14

```

```

15 //-----
16 // Dropdown menü Position
17 //-----
18
19 window.addEventListener('load', () => {
20     let windowHeight = $(window).width();
21     let navBarDropDown = $('nav div #dropdown-account .dropdown-menu');
22
23     if (windowWidth <= 992) {
24         navBarDropDown.addClass('dropdown-menu-start');
25         navBarDropDown.removeClass('dropdown-menu-end');
26     } else {
27         navBarDropDown.addClass('dropdown-menu-end');
28         navBarDropDown.removeClass('dropdown-menu-start');
29     }
30 })
31
32 window.addEventListener('resize', () => {
33     let windowHeight = $(window).width();
34     let navBarDropDown = $('nav div .dropdown .dropdown-menu');
35
36     if (windowWidth <= 992) {
37         navBarDropDown.addClass('dropdown-menu-start');
38         navBarDropDown.removeClass('dropdown-menu-end');
39     } else {
40         navBarDropDown.addClass('dropdown-menu-end');
41         navBarDropDown.removeClass('dropdown-menu-start');
42     }
43 })
44
45 //-----
46 // Maximale größ
47 //-----
48
49 $(window).ready(function () {
50     $('.mh').each(function () {
51         $(this).css('min-height', windowHeight - navheight - (distance) - footerheight);
52         $(this).css('height', 'auto');
53         $(this).css('overflow', 'auto');
54     });
55 })

```

#### system\backend\js\searchbar.js

```

1 $(document).ready(function () {
2     $('.table').each(function() {
3         var table = $(this).find('table tbody');
4
5         console.log($(this).find('div .searchbar'));
6         $(this).find('div .searchbar').on('keyup', function() {
7             let searchtext = $(this).val().toLowerCase()
8
9             table.find('tr').each(function() {
10                let row = $(this);
11                let name = row.find('td').text().toLowerCase()
12
13                row.toggle(name.includes(searchtext))
14            })

```

```
15     })
16   })
17 })
```

#### system\backend\js\theme.js

```
1  (() => {
2    'use strict'
3
4    const getStoredTheme = () => localStorage.getItem('theme')
5    const setStoredTheme = theme => localStorage.setItem('theme', theme)
6
7    const getPreferredTheme = () => {
8      const storedTheme = getStoredTheme()
9      if (storedTheme) {
10       return storedTheme
11     }
12
13     return window.matchMedia('(prefers-color-scheme: dark)').matches ? 'dark' : 'light'
14   }
15
16   const setTheme = theme => {
17     if (theme === 'auto') {
18       document.documentElement.setAttribute('data-bs-theme', (window.matchMedia('(prefers-color-
19 scheme: dark)').matches ? 'dark' : 'light'))
20     } else {
21       document.documentElement.setAttribute('data-bs-theme', theme)
22     }
23   }
24   setTheme(getPreferredTheme())
25
26   const showActiveTheme = (theme, focus = false) => {
27     const themeSwitcher = document.querySelector('#bd-theme')
28
29     if (!themeSwitcher) {
30       return
31     }
32
33     const themeSwitcherText = document.querySelector('#bd-theme-text')
34     const activeThemeIcon = document.querySelector('.theme-icon-active use')
35     const btnToActive = document.querySelector(`[data-bs-theme-value="${theme}"]`)
36     const svgOfActiveBtn = btnToActive.querySelector('svg use').getAttribute('href')
37
38     document.querySelectorAll('[data-bs-theme-value]').forEach(element => {
39       element.classList.remove('active')
40       element.setAttribute('aria-pressed', 'false')
41     })
42
43     btnToActive.classList.add('active')
44     btnToActive.setAttribute('aria-pressed', 'true')
45     activeThemeIcon.setAttribute('href', svgOfActiveBtn)
46     const themeSwitcherLabel = `${themeSwitcherText.textContent}
47     (${btnToActive.dataset.bsThemeValue})`
48     themeSwitcher.setAttribute('aria-label', themeSwitcherLabel)
49
50     if (focus) {
51       themeSwitcher.focus()
52     }
53   }
```

```

52 }
53
54 window.matchMedia('(prefers-color-scheme: dark)').addEventListener('change', () => {
55     const storedTheme = getStoredTheme()
56     if (storedTheme !== 'light' && storedTheme !== 'dark') {
57         setTheme(getPreferredTheme())
58     }
59 })
60
61 window.addEventListener('DOMContentLoaded', () => {
62     showActiveTheme(getPreferredTheme())
63
64     document.querySelectorAll('[data-bs-theme-value]')
65         .forEach(toggle => {
66             toggle.addEventListener('click', () => {
67                 const theme = toggle.getAttribute('data-bs-theme-value')
68                 setStoredTheme(theme)
69                 setTheme(theme)
70                 showActiveTheme(theme, true)
71             })
72         })
73 })
74 })()

```

#### system\backend\snippets\snippetAlert.php

```

1 <?php
2 class snippetAlert {
3     /* =====
4     Variablen
5     ===== */
6
7     public $sType;
8     public $sIcon;
9     public $sTitle;
10    public $sContent;
11    public $sClasses;
12
13    /* =====
14    Kosntruktor
15    ===== */
16
17    public function __construct($sType = 'info')
18    {
19        $this->sType = $sType;
20    }
21
22    /* =====
23    Datenverarbeitung
24    ===== */
25
26    /* =====
27    Icon
28    ===== */
29
30    public function setIcon ($sIcon, $sStyle = 'solid') {
31        $this->sIcon = cms::getIcon($sIcon, $sStyle);
32    }

```

```

33
34      /* =====
35      Titel
36      ===== */
37
38      public function setTitle ($sTitle) {
39          $this->sTitle = '<strong>' . $sTitle . '</strong>';
40      }
41
42      /* =====
43      Inhalt
44      ===== */
45
46      public function setContent ($sContent) {
47          $this->sContent = $sContent;
48      }
49
50      /* =====
51      Klassen
52      ===== */
53
54      public function setClasses ($sClasses) {
55          $this->sClasses = $sClasses;
56      }
57
58      /* =====
59      Typ
60      ===== */
61
62      public function setType ($sType) {
63          $this->sType = $sType;
64      }
65
66      /* =====
67      Ansicht
68      ===== */
69
70      public function view() {
71          if (!isset($this->sContent)) return;
72
73          $sHead = "";
74
75          if (isset($this->sIcon)) {
76              $sHead = $this->sIcon . ' ';
77          }
78
79          if (isset($this->sTitle)) {
80              $sHead .= $this->sTitle . '<hr>';
81          }
82
83          ob_start();
84          echo '<div class="alert alert-' . $this->sType . ' ' . $this->sClasses . '"
role="alert">';
85              if (isset($sHead)) {
86                  echo $sHead;
87              }
88
89          echo $this->sContent;

```

```

90         echo '</div>';
91         return ob_get_clean();
92     }
93 }

```

#### system\backend\snippets\snippetCounter.php

```

1  <?php
2  class snippetCounter {
3      /* =====
4      Variablen
5      ===== */
6
7      public $iTarget;
8      public $fTimeout = 10;
9
10     /* =====
11     Datenverarbeitung
12     ===== */
13
14     /* =====
15     Ziel Anzahl setzen
16     ===== */
17
18     public function setTarget ($iTarget) {
19         $this->iTarget = $iTarget;
20     }
21
22     /* =====
23     Timeout Definieren
24     ===== */
25
26     public function setTimeout ($fTimeout) {
27         $this->fTimeout = $fTimeout;
28     }
29
30     /* =====
31     Ansicht
32     ===== */
33
34     public function view () {
35         ob_start();
36         echo '<div class="counter">';
37         echo '<span data-setp="1" data-target="100">-100</span>';
38         echo '</div>';
39         return ob_get_clean();
40     }
41 }

```

#### system\backend\snippets\snippetInput.php

```

1  <?php
2  class snippetInput {
3      /* =====
4      Input Felder
5      ===== */
6
7      /* =====
8      Checkbox

```

```

9      ===== */
10
11     public function getCheckbox ($sLabel, $aAttributes = [], $bChecked = false) {
12         $aAttr = $this->getAttributes($aAttributes, 'form-check-input');
13
14         ob_start();
15         echo '<div class="mb-3 form-check form-switch switch-row d-flex mw-100 justify-
content-between align-items-center px-0">';
16         echo '<label class="input-group-text w-100">' . $sLabel . '</label>';
17         echo '<div>';
18         echo '<input name="menu" class="' . $aAttr['class'] . '" ' . $aAttr['attr'] .
' type="checkbox" ' . (($bChecked) ? 'checked' : '') . '>';
19         echo '</div>';
20         echo '</div>';
21         return ob_get_clean();
22     }
23
24     /* =====
25     Select
26     ===== */
27
28     public function getSelect ($sLabel, $aOptions = [], $aAttributes = [], $sSelected = '') {
29         $sClass      = "custom-select form-control w-75 text-end";
30         $sID          = str_replace(' ', '_', strtolower($sLabel));
31         $aAttr = $this->getAttributes($aAttributes, $sClass);
32
33         ob_start();
34         echo '<div class="input-group mb-3 w-100">';
35         echo '<label class="input-group-text w-25" for="' . $sID . '">' . $sLabel .
'</label>';
36         echo '<select class="' . $aAttr['class'] . '" id="' . $sID . '" ' . $aAttr['attr']
. '>';
37         foreach ($aOptions AS $sValue => $sLabel) {
38             echo '<option type="select" value="' . $sValue . '" ' . (($sSelected ==
$sValue) ? 'selected' : '') . '>' . $sLabel . '</option>';
39         }
40         echo '</select>';
41         echo '</div>';
42         return ob_get_clean();
43     }
44
45     /* =====
46     Number
47     ===== */
48
49     public function getNumber ($sLabel, $aAttributes = []) {
50         $aAttr = $this->getAttributes($aAttributes, 'form-control w-sm-50 text-end');
51
52         ob_start();
53         echo '<div class="mb-3 input-group form-switch switch-row d-flex w-100 justify-
content-between align-items-center px-0">';
54         echo '<label class="input-group-text">' . $sLabel . '</label>';
55         echo '<input name="menu" class="' . $aAttr['class'] . '" ' . $aAttr['attr'] . '
type="number">';
56         echo '</div>';
57         return ob_get_clean();
58     }
59

```

```

60      /* =====
61      Hilfsfunktionen
62      ===== */
63
64      /* =====
65      Attribute Verarbeiten
66      ===== */
67
68      public function getAttributes($aAttributes, $sClass = "") {
69          $aData = [];
70          $sAttributes = "";
71
72          foreach ($aAttributes AS $sKey => $mValue) {
73              switch ($sKey) {
74                  case 'class':
75                      $sClass .= ' ' . $mValue;
76                      break;
77                  case 'disabled':
78                      if ($mValue == true || $mValue == 'true') {
79                          $sAttributes .= ' disabled';
80                      }
81                      break;
82                  case 'checked':
83                      if ($mValue == true || $mValue == 'true') {
84                          $sAttributes .= ' checked';
85                      }
86                      break;
87                  default:
88                      $sAttributes .= ' ' . $sKey . '=' . $mValue . ' ';
89              }
90          }
91      }
92
93      $aData['attr'] = $sAttributes;
94      $aData['class'] = $sClass;
95
96      return $aData;
97  }
98  }

```

#### system\backend\snippets\snippetNotify.php

```

1  <?php
2  class snippetNotify {
3      /* =====
4      Variablen
5      ===== */
6
7      public $sType      = 'toast';
8      public $sContent  = '';
9
10     /* =====
11     Konstruktor
12     ===== */
13
14     /* =====
15     Ansicht
16     ===== */

```

```

17
18     /* =====
19     Toast
20     ===== */
21
22     public function viewToast () {
23         echo '<script>';
24         echo 'const Swal = require("sweetalert2")';
25         echo 'Swal.mixin({toast:true,position:"top-
end",showConfirmButton:false,timer:3000,timerProgressBar:true,didOpen:t=>
{t.onmouseenter=Swal.stopTimer;t.onmouseleave=Swal.resumeTimer}}).fire({icon:"success",title:"Signed
in successfully"});';
26         echo '</script>';
27     }
28 }

```

#### system\backend\snippets\snippetTab.php

```

1 <?php
2 class snippetTab {
3     /* =====
4     Variablen
5     ===== */
6
7     public $aData          = [];
8     public $bNoDataMessage = false;
9
10    /* =====
11    Datenverarbeitung
12    ===== */
13
14    /* =====
15    Tab Hinzufügen
16    ===== */
17
18    public function addTab ($sName, $sContent = null) {
19        if (array_key_exists($sName, $this->aData)) return;
20        $sKey = str_replace(' ', '_', strtolower($sName));
21        $this->aData[$sKey]['label'] = $sName;
22        if (!empty($sContent)) {
23            $this->aData[$sKey]['content'] = $sContent;
24        }
25    }
26
27    /* =====
28    Inhalt setzen
29    ===== */
30
31    public function setData ($sTab, $sContent) {
32        if (!array_key_exists($sTab, $this->aData)) return;
33    }
34
35    /* =====
36    Einstellungen
37    ===== */
38
39    /* =====
40    Keine Daten Nachricht
41    ===== */

```

```

42
43 public function noDataMessage ($bShowMessage = false) {
44     $this->bNoDataMessage = $bShowMessage;
45 }
46
47 /* =====
48 Ansicht
49 ===== */
50
51 public function view() {
52     ob_start();
53     echo '<div class="w-100">';
54     echo '<ul class="nav nav-tabs">';
55         $i = 0;
56         foreach ($this->aData AS $sKey => $aData) {
57             $sClass = 'nav-link';
58             if ($i == 0) {
59                 $sClass .= ' active';
60             }
61             $i ++;
62             echo '<li class="nav-item">';
63                 echo '<button class="' . $sClass . '" data-bs-toggle="tab" data-
bs-target="#' . $sKey . '">';
64                     echo $aData['label'];
65                     echo '</button>';
66                     echo '</li>';
67             }
68         echo '</ul>';
69
70         echo '<div class="tab-content mt-3">';
71             $i = 0;
72             foreach ($this->aData AS $sKey => $aData) {
73                 $sClass = 'tab-pane fade';
74                 if ($i == 0) {
75                     $sClass .= ' show active';
76                 }
77                 $i ++;
78
79                 echo '<div id="' . $sKey . '" class="' . $sClass . '">';
80                     echo $aData['content'] ?? $this->viewNoDataMessage();
81                 echo '</div>';
82             }
83             echo '</div>';
84         echo '</div>';
85         return ob_get_clean();
86     }
87
88 public function viewNoDataMessage() {
89     if ($this->bNoDataMessage) {
90         $oAlert = new snippetAlert();
91         $oAlert->setTitle('Information:');
92         $oAlert->setIcon('circle-info');
93         $oAlert->setContent('Hier wurden keine Daten gefunden...');
94         return $oAlert->view();
95     } else {
96         return '';
97     }
98 }

```

## system\backend\snippets\snippetTable.php

```

1  <?php
2  class snippetTable {
3      /* =====
4      Variabalen
5      ===== */
6
7      public $sMode      = "body";
8      public $aData      = [];
9      public $bTop       = false;
10     public $bSearchable = false;
11     public $sTopData    = '';
12     public $aRowData    = [
13         'head' => 0,
14         'body' => 0,
15     ];
16     public $sTableClass = "";
17     public $mNoDataMsg  = false;
18
19     /* =====
20     Konstruktor
21     ===== */
22
23     public function __construct ($sClasses = "") {
24         if ($sClasses != "") {
25             $this->sTableClass = $sClasses;
26         }
27     }
28
29     /* =====
30     Modi Umschalten
31     ===== */
32
33     /* =====
34     Head
35     ===== */
36
37     public function head () {
38         $this->sMode = 'head';
39     }
40
41     /* =====
42     Body
43     ===== */
44
45     public function body () {
46         $this->sMode = 'body';
47     }
48
49     /* =====
50     Einstellungen
51     ===== */
52
53     /* =====
54     Suchleiste

```

```

55 ===== */
56
57 public function setSearchable ($bSearchable = false) {
58     $this->bSearchable = $bSearchable;
59     if ($bSearchable) {
60         $this->bTop = true;
61     }
62 }
63
64 /* =====
65 Nachricht falls keine Daten vorhanden sind
66 ===== */
67
68 public function setNoDataMessage ($mMessage = false) {
69     $this->mNoDataMsg = $mMessage;
70 }
71
72 /* =====
73 Datenverarbeitung
74 ===== */
75
76 /* =====
77 Neue Reihe
78 ===== */
79
80 public function row () {
81     switch ($this->sMode) {
82         case 'head':
83             $this->aRowData['head'] = $this->aRowData['head'] + 1;
84             $this->aData['head'][$this->aRowData['head']] = [];
85             break;
86         case 'body':
87             $this->aRowData['body'] = $this->aRowData['body'] + 1;
88             $this->aData['body'][$this->aRowData['body']] = [];
89             break;
90     }
91 }
92
93 /* =====
94 Neue Überschrift
95 ===== */
96
97 public function th ($mValue, $aOptions = []) {
98     $sAttribute = "";
99     if (!empty($aOptions)) {
100         foreach ($aOptions AS $sParameter => $sValue) {
101             $sAttribute .= ' ' . $sParameter . '=' . $sValue . '';
102         }
103     }
104
105     $this->aData[$this->sMode][$this->aRowData[$this->sMode]][] = '<th ' . $sAttribute .
106 '>' . $mValue . '</th>';
107 }
108
109 /* =====
110 Neue Daten
111 ===== */

```

```

112     public function td ($mValue, $aOptions = []) {
113         $sAttribute = "";
114         if (!empty($aOptions)) {
115             foreach ($aOptions AS $sParameter => $sValue) {
116                 $sAttribute .= ' ' . $sParameter . '=' . $sValue . ' ';
117             }
118         }
119
120         $this->aData[$this->sMode][$this->aRowData[$this->sMode]][] = '<td ' . $sAttribute .
121 '>' . $mValue . '</td>';
122     }
123
124     /* =====
125     ZUSätzliche Top Daten
126     ===== */
127
128     public function setTop ($sElement) {
129         $this->sTopData = $sElement;
130         $this->bTop = true;
131     }
132
133     /* =====
134     Ansicht
135     ===== */
136
137     public function view () {
138         ob_start();
139         echo '<div class="table mt-2">';
140         if ($this->bTop) {
141             echo '<div class="w-100 d-flex align-items-center justify-content-between mb-
142 2">';
143
144             if ($this->bSearchable) {
145                 echo '<div class="w-50 d-flex align-items-center">' .
146 cms::getIcon('search', 'solid', 'me-3') . '<input type="text" class="searchbar form-control ml-3
147 w-75" placeholder="Suche..."></div>';
148             }
149             echo '<div class="mw-50">' . $this->sTopData . '</div>';
150             echo '</div>';
151         }
152         echo '<table class="table ' . $this->sTableClass . '">';
153         if (!empty($this->aData['head']) && !empty($this->aData['body'])) {
154             echo '<thead>';
155             foreach ($this->aData['head'] AS $iRow => $aRowData) {
156                 echo '<tr scope="col" class="align-middle">';
157                 foreach ($aRowData AS $iDataKey => $sData) {
158                     echo $sData;
159                 }
160                 echo '</tr>';
161             }
162             echo '</thead>';
163         }
164         echo '<tbody>';
165         if (!empty($this->aData['body'])) {
166             foreach ($this->aData['body'] AS $iRow => $aRowData) {
167                 echo '<tr class="align-middle">';
168                 foreach ($aRowData AS $iDataKey => $sData) {
169                     echo $sData;

```

```

166         }
167         echo '</tr>';
168     }
169     } else {
170         $oAlert = new snippetAlert();
171         $oAlert->setIcon('circle-info');
172         $oAlert->setTitle('Keine Daten');
173         if ($this->mNoDataMsg != false) {
174             $oAlert->setContent($this->mNoDataMsg);
175             echo $oAlert->view();
176         }
177     }
178     echo '</tbody>';
179     echo '</table>';
180     echo '</div>';
181
182     return ob_get_clean();
183 }
184 }

```

#### system\frontend\elements\footer.php

```

1 <footer>
2     <div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2)">
3         &copy; <? = date('Y') ?>
4         <a class="text-white" href="https://marcel.privat.1cue.dev/"
5             >Tap2Table.com</a
6         >
7     </div>
8     <!-- Copyright -->
9 </footer>

```

#### system\frontend\elements\navbar.php

```

1 <?php
2     /* =====
3     Logik
4     ===== */
5
6     $oDb = new db();
7     $oDb->query("SELECT * FROM pages");
8     $oDb->execute(true);
9     $aResult = $oDb->get(MYSQLI_ASSOC);
10 ?>
11
12 <nav class="navbar navbar-expand-lg bg-body-tertiary fixed-top shadow-sm">
13     <div class="container-fluid">
14
15         <!-- Toggle Button -->
16         <button
17             class="navbar-toggler"
18             type="button"
19             data-bs-toggle="collapse"
20             data-bs-target="#navbarSupportedContent"
21             aria-controls="navbarSupportedContent"
22             aria-expanded="false"
23             aria-label="Toggle navigation"
24         >
25         <span class="navbar-toggler-icon"></span>

```

```

26 </button>
27
28 <!-- Brand -->
29 <a class="navbar-brand" href="?site=home">
30   
35 </a>
36
37 <!-- Navbar Content -->
38 <div class="collapse navbar-collapse" id="navbarSupportedContent">
39
40   <!-- Left links -->
41   <ul class="navbar-nav me-auto mb-2 mb-lg-0">
42     <?php
43       /* =====
44       Items einfügen
45       ===== */
46
47       foreach ($arResult AS $aItem) {
48
49         /* =====
50         Loginüberprüfung
51         ===== */
52
53         if ($aItem['login'] && !cms::isLoggedIn(false)) {
54           continue;
55         }
56
57         /* =====
58         Kontent
59         ===== */
60
61         $sContent = "";
62
63         // Icon
64
65         if (isset($aItem['icon'])) {
66           $sContent = cms::getIcon($aItem['icon']) . ' ';
67         }
68
69         // Text
70
71         if (isset($aItem['name'])) {
72           $sContent .= $aItem['name'];
73         }
74
75         /* =====
76         Darstellung
77         ===== */
78
79         switch ($aItem['type']) {
80           case 'item':
81             if ($aItem['login'] && cms::isLoggedIn(false)) {
82               echo '<li class="nav-item">';

```

```

83         echo '<a href="?site=' . $aItem['page'] . '" class="nav-link">' .
$sContent . '</a>';
84     echo '</li>';
85     } else if (!$aItem['login']) {
86         echo '<li class="nav-item active">';
87         echo '<a href="?site=' . $aItem['page'] . '" class="nav-link">' .
$sContent . '</a>';
88         echo '</li>';
89     }
90     break;
91     case 'dropdown':
92         // Daten Laden
93
94         $oDb->query("SELECT * FROM pages WHERE parent = ?", ['i', $aItem['id']]);
95         $oDb->execute(true);
96         $aDropdownItems = $oDb->get(MYSQLI_ASSOC);
97
98         if (empty($aDropdownItems)) {
99             break;
100        }
101
102        // Ansicht
103        echo '<li class="dropdown">';
104        echo '<a class="dropdown-toggle nav-link d-flex align-items-center"
href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false" >';
105            echo $sContent;
106            echo '</a>';
107
108            echo '<ul class="dropdown-menu dropdown-menu-start">';
109            foreach ($aDropdownItems AS $aDropdownItem) {
110                if ($aDropdownItem['login'] && !cms::isLoggedIn(false))
continue;
111                $sItemContent = "";
112                if (isset($aDropdownItem['icon'])) {
113                    $sContent = cms::getIcon($aDropdownItem['icon']) . ' ' .
$aDropdownItem['name'];
114                } else {
115                    $sContent = $aDropdownItem['name'];
116                }
117                echo '<li><a class="dropdown-item" href="?site=' .
$aDropdownItem['page'] . '">' . $sContent . '</a></li>';
118            }
119            echo '</ul>';
120            echo '</li>';
121            break;
122            case 'dropdown-item':
123                break;
124        }
125    }
126    ?>
127    </ul>
128
129    <!-- Right elements -->
130    <div class="navbar-nav ms-auto">
131        <!-- Avatar -->
132        <?php
133            if (!cms::isLoggedIn()) {

```

```

134         echo '<a href="?site=login" class="btn btn-primary">Anmelden</a href="?
site=login">';
135     }
136     else {
137         echo '<div class="dropdown nav-item" id="dropdown-account">';
138         echo '<a ';
139             echo 'class="dropdown-toggle d-flex align-items-center nav-link" ';
140             echo 'href="#" ' ';
141             echo 'role="button" ' ';
142             echo 'data-bs-toggle="dropdown" ' ';
143             echo 'aria-expanded="true" ' ';
144         echo '>';
145         echo '';
151         echo '</a>';
152
153         echo '<ul class="dropdown-menu">';
154             echo '<li><a class="dropdown-item" href="#">My profile</a></li>';
155             echo '<li><a class="dropdown-item" href="#">Settings</a></li>';
156             echo '<li><a class="dropdown-item" href="?site=logout">Logout</a></li>';
157         echo '</ul>';
158     echo '</div>';
159 }
160 ?>
161
162
163 </div>
164 </div>
165 </div>
166 </nav>
167

```

#### system\frontend\pages\dashboard.php

```

1 <?php
2

```

#### system\frontend\pages\dev.php

```

1 <?php
2     cms::isLoggedIn(true);
3     echo '<div class="container mt-5">';
4
5         /* =====
6         Alert
7         ===== */
8
9         echo '<h2>Alert</h2>';
10        echo '<hr>';
11
12        $oAlert = new snippetAlert();
13        $oAlert->setIcon('circle-info');
14        $oAlert->setTitle('Titel');
15        $oAlert->setContent('Inhalt');
16        echo $oAlert->view();

```

```

17
18      /* =====
19      Tabelle
20      ===== */
21
22      echo '<h2>Tabelle</h2>';
23      echo '<hr>';
24
25      $oTable = new snippetTable();
26      $oTable->head();
27      $oTable->row();
28      $oTable->th('test');
29      $oTable->td('test2');
30      $oTable->body();
31      $oTable->row();
32      $oTable->th('test');
33      $oTable->td('test2');
34      echo $oTable->view();
35
36      /* =====
37      Tabelle
38      ===== */
39
40      echo '<h2>Tabs</h2>';
41      echo '<hr>';
42
43      $oTabs = new snippetTab();
44      $oTabs->addTab('Test 1', '<h1>Test 1</h1>');
45      $oTabs->addTab('Test 2', '<h1>Test 2</h1>');
46      echo $oTabs->view();
47
48      /* =====
49      Counter
50      ===== */
51
52      echo '<h2>Tabs</h2>';
53      echo '<hr>';
54
55      $oCounter = new snippetCounter();
56      $oCounter->setTarget(100);
57      echo $oCounter->view();
58
59      echo '</div>';
60
61      ?>
62
63      <div class="container">
64          <ul class="nav nav-tabs">
65              <li class="nav-item">
66                  <button class="nav-link active" data-bs-toggle="tab" data-bs-target="#london">
67                      London
68                  </button>
69              </li>
70              <li class="nav-item">
71                  <button class="nav-link" data-bs-toggle="tab" data-bs-target="#paris">
72                      Paris
73                  </button>
74              </li>

```

```

75     <li class="nav-item">
76         <button class="nav-link" data-bs-toggle="tab" data-bs-target="#tokyo">
77             Tokyo
78         </button>
79     </li>
80 </ul>
81
82     <div class="tab-content mt-3">
83 <div id="london" class="tab-pane fade show active">
84     <h3>London</h3>
85     <p>London is the capital city of England.</p>
86 </div>
87
88 <div id="paris" class="tab-pane fade">
89     <h3>Paris</h3>
90     <p>Paris is the capital of France.</p>
91 </div>
92
93 <div id="tokyo" class="tab-pane fade fade-in">
94     <h3>Tokyo</h3>
95     <p>Tokyo is the capital of Japan.</p>
96 </div>
97 </div>
98
99
100 </div>

```

system\frontend\pages\home.php

```

1 <?php
2

```

system\frontend\pages\inventory.php

```

1 <?php
2
3     /* =====
4     Login Logik
5     ===== */
6
7     cms::isLoggedIn(true);
8
9     /* =====
10    Tabelle Vorbereiten
11    ===== */
12
13    $oTable = new snippetTable('table-hover text-center');
14    $oTable->head();
15    $oTable->th('Bild');
16    $oTable->th('Name');
17    $oTable->th('SKU');
18    $oTable->th('Preis');
19    $oTable->th('Anzahl');
20    $oTable->th('Ausverkauft');
21    $oTable->th('Im Menu');
22    $oTable->th('Aktionen');
23
24    /* =====
25    Alert Vorbereiten

```

```

26     ===== */
27
28     $oAlert = new snippetAlert();
29     $oAlert->setIcon('circle-info');
30     $oAlert->setTitle('Inventar Information');
31     $oAlert->setClasses('p-2 w-100 wd-3 text-center');
32
33     /* =====
34     Datenverarbeitung
35     ===== */
36
37     /* =====
38     Artikel Laden
39     ===== */
40
41     $oDb = new db();
42     $oDb->query("SELECT * FROM inventory WHERE k_id = ?", ['s', $_SESSION['k_id']]);
43     $oDb->execute(true);
44     $aResult = $oDb->get(MYSQLI_ASSOC);
45
46     $oTable->body();
47     foreach ($aResult AS $aData) {
48         $oTable->row();
49         $oTable->td/cms::img(($aData['img'] ?? 'logo'), 'uploads');
50         $oTable->td($aData['name']);
51         $oTable->td($aData['sku']);
52         $oTable->td/cms::convertValue($aData['price'], 'price', ['unit' => '€']);
53         $oTable->td($aData['quantity'] ?? '---');
54         $oTable->td/cms::convertValue($aData['sold_out'], 'bool');
55         $oTable->td/cms::convertValue($aData['menu'], 'bool');
56         $oTable->td('<a class="btn btn-warning">' . cms::getIcon('pen') . '</a> <a href=?
site=inventory&del_id=' . $aData['id'] . '" class="btn btn-danger">' . cms::getIcon('trash') .
'</a>');
57     }
58
59     /* =====
60     Neuen Artikel Hinzufügen
61     ===== */
62
63     if (isset($_POST['name'])) {
64         // Bild Upload
65
66         if (isset($_FILES['image']) && $_FILES['image']['error'] === UPLOAD_ERR_OK) {
67             $sTargetDir     = dirname(__DIR__, 2) . "/data/uploads/";
68             $sExtentions    = pathinfo($_FILES['image']['name'], PATHINFO_EXTENSION);
69             $sFileName      = uniqid('inventory_', true) . '.' . $sExtentions;
70             $sTartgetPath   = $sTargetDir . $sFileName;
71             $fTargetFile    = $sTargetDir . basename($_FILES["image"]["name"]);
72             $sImagFileType  = strtolower(pathinfo($fTargetFile,PATHINFO_EXTENSION));
73
74             // Bild Überprüfen
75             if(isset($_FILES['image']) && $_FILES['image']['error'] === UPLOAD_ERR_OK) {
76                 $iCheck = getimagesize($_FILES["image"]["tmp_name"]);
77                 if($iCheck !== false) {
78                     if (!move_uploaded_file($_FILES['image']['tmp_name'], $sTartgetPath)) {
79                         die('Upload fehlgeschlagen');
80                     }
81                 } else {

```

```

82         }
83     }
84 }
85
86 // Artikel Daten
87
88 $bSoledOut = (isset($_POST['sold_out'])) ? 1 : 0;
89 $bMenu     = (isset($_POST['menu'])) ? 1 : 0;
90
91 $oDb->query("INSERT INTO `inventory` (`id`, `k_id`, `sku`, `name`, `price`, `quantity`,
`sold_out`, `menu`, `img`) VALUES (NULL, ?, ?, ?, ?, ?, ?, ?);", [
92     'ssdiiis',
93     (string) $_SESSION['k_id'],
94     (string) $_POST['sku'],
95     (string) $_POST['name'],
96     (float)  $_POST['price'],
97     (int)    $_POST['quantity'] ?? null,
98     (int)    $bSoledOut,
99     (int)    $bMenu,
100    (string)  $sFileName,
101    ]);
102 $oDb->execute(false);
103 header('Location: ?site=inventory&notif=1');
104 }
105
106 /* =====
107 Artikel Löschen
108 ===== */
109
110 if (isset($_GET['del_id'])) {
111     $oDb->query("DELETE FROM inventory WHERE `inventory`.`id` = ?", ['i', $_GET['del_id']]);
112     $oDb->execute(false);
113     header('Location: ?site=inventory&notif=2');
114 }
115
116 /* =====
117 Bild Upload
118 ===== */
119
120 /* =====
121 Benachrichtigung
122 ===== */
123
124 // $oAlert->setType('success');
125 // $oAlert->setContent()
126
127 if (isset($_GET['notif'])) {
128     switch ($_GET['notif']) {
129         case 1:
130             $oAlert->setType('success');
131             $oAlert->setContent('Der Artikel wurde Hinzugefügt');
132             break;
133         case 2:
134             $oAlert->setType('warning');
135             $oAlert->setContent('Der Artikel wurde Entfernt');
136             break;
137     }
138 }

```

```

139 ?>
140
141 <div class="container">
142   <div class="row">
143     <div class="col-12 col-md-6 col-6 p-2">
144       <div class="card shadow-sm p-2 mh">
145         <?= $oTable->view() ?>
146       </div>
147     </div>
148     <div class="col-12 col-md-6 p-2">
149       <?= $oAlert->view(); ?>
150       <div class="p-2 card shadow-sm text-center d-flex align-items-center">
151         <h2 class="text-uppercase">Neuen Artikel Anlegen</h2>
152         <hr>
153         <form method="post" enctype="multipart/form-data">
154           <div class="input-group w-100 mb-3">
155             <span class="input-group-text w-25">Name</span>
156             <input name="name" type="text" class="form-control w-75"
placeholder="Name" required>
157           </div>
158           <div class="input-group w-100 mb-3">
159             <span class="input-group-text w-25">SKU</span>
160             <input name="sku" type="text" class="form-control w-75"
placeholder="Artikelnummer" required>
161           </div>
162           <div class="input-group w-100 mb-3">
163             <span class="input-group-text w-25">Anzahl</span>
164             <input name="quantity" type="number" class="form-control w-75" min="0"
value="null" placeholder="---">
165           </div>
166           <div class="input-group w-100 mb-3">
167             <span class="input-group-text w-25">Preis</span>
168             <input name="price" type="number" class="form-control w-50" min="0"
step="0.1" placeholder="---" required>
169             <span class="input-group-text">€</span>
170           </div>
171           <div class="mb-3 form-check form-switch switch-row d-flex mw-100 justify-
content-between align-items-center px-0">
172             <label class="input-group-text w-100">Ausverkauft:</label>
173             <div class="">
174               <input name="sold_out" class="form-check-input" type="checkbox"
value="1">
175             </div>
176           </div>
177           <div class="mb-3 form-check form-switch switch-row d-flex mw-100 justify-
content-between align-items-center px-0">
178             <label class="input-group-text w-100">Im Menü:</label>
179             <div>
180               <input name="menu" class="form-check-input" type="checkbox" checked>
181             </div>
182           </div>
183           <div class="input-group mb-3">
184             <span class="input-group-text" id="inputGroupFileAddon">Bild</span>
185             <input type="file" class="form-control" id="fileToUpload" name="image"
aria-describedby="inputGroupFileAddon">
186           </div>
187
188           <button type="submit" class="w-100 btn btn-success">Hinzufügen</button>
189 </form>

```

```
190     </div>
191   </div>
192 </div>
193 </div>
```

#### system\frontend\pages\login.php

```
1 <?php
2
3  /* =====
4  Fehlermeldungen
5  ===== */
6
7  $iErrorCode = $_GET['err'] ?? null;
8
9  $oAlert = new snippetAlert('danger');
10 $oAlert->setIcon('circle-info');
11 $oAlert->setTitle('Fehler');
12
13 if (isset($iErrorCode)) {
14     switch ($iErrorCode) {
15         case 1:
16             $oAlert->setContent('Sie sind nicht angemeldet.<br> Bitte melden sie sich an.');
```

```
17             break;
18         case 2:
19             $oAlert->setContent('Bitte Geben sie E-Mail und Passwort an.');
```

```
20             break;
21         case 3:
22             $oAlert->setContent('Dieses Konto ist uns nicht bekannt.');
```

```
23     }
24 }
25
26 /* =====
27 Anmelde Logik
28 ===== */
29
30 if (cms::isLoggedIn(false)) {
31     header('Location: ?site=dashboard');
```

```
32 }
33
34 if (!empty($_POST)) {
35     if ($_POST['email'] == "" || $_POST['password'] == "") {
36         header('Location: ?site=login&err=2');
```

```
37         exit;
38     }
39
40     $sPassword = sha1($_POST['password']);
41
42     $oDb = new db();
43     $oDb->query("SELECT id, k_id, email FROM companys WHERE email = ? and password = ?", [
44         'ss',
45         $_POST['email'],
46         $sPassword,
47     ]);
48     $oDb->execute(true);
49     $aResult = $oDb->get(MYSQLI_ASSOC);
50     if (empty($aResult)) {
51         header('Location: ?site=login&err=3');
```

```

52         exit;
53     }
54
55     $_SESSION['id']      = $arResult[0]['id'];
56     $_SESSION['k_id']   = $arResult[0]['k_id'];
57     $_SESSION['email']  = $arResult[0]['email'];
58
59     if (isset($_GET['prev'])) {
60         header('Location: ?site=' . $_GET['prev']);
61     } else {
62         header('Location: ?site=dashboard');
63     }
64
65 }
66 ?>
67
68 <div class="w-100 h-100 d-flex justify-content-center align-items-center">
69     <div class="card p-3 text-center border-1 shadow-sm">
70         <form method="post">
71             <h2 class="fw-bold mb-2 text-uppercase">Login</h2>
72             <p class="mb-4">Please enter your E-Mail and Password</p>
73             <?php
74                 echo $oAlert->view();
75             ?>
76             <hr class="mb-4">
77
78
79             <div class="input-group mb-3">
80                 <span class="input-group-text" id="basic-addon1">E-Mail</span>
81                 <input name="email" type="text" class="form-control" placeholder="E-Mail" aria-
label="E-Mail" aria-describedby="basic-addon1" value="<?=$_POST['email'] ?? "" ?>">
82             </div>
83             <div class="input-group mb-4">
84                 <span class="input-group-text" id="basic-addon1">Password</span>
85                 <input name="password" type="password" class="form-control" placeholder="Password"
aria-label="Password" aria-describedby="basic-addon1" value="<?=$_POST['password'] ?? "" ?>">
86             </div>
87
88
89             <button class="btn btn-primary w-100" type="submit">Login</button>
90         </form>
91     </div>
92 </div>

```

#### system\frontend\pages\logout.php

```

1 <?php
2     session_destroy();
3     session_start();
4     header('Location: ?site=home');

```

#### system\frontend\pages\menu.php

```

1 <?php
2     $oDb      = new db();
3     $oInput   = new snippetInput();
4     $oAlert   = new snippetAlert();
5
6     /* =====

```

```

7   Daten Laden
8   ===== */
9
10  /* =====
11  Artikel
12  ===== */
13
14  $oDb->query('SELECT * FROM inventory WHERE k_id = ?', ['s', $_SESSION['k_id']]);
15  $oDb->execute(true);
16  $aItems = $oDb->get(MYSQLI_ASSOC);
17
18  /* =====
19  Gruppen
20  ===== */
21
22  $oDb->query('SELECT * FROM category WHERE k_id = ?', ['s', $_SESSION['k_id']]);
23  $oDb->execute(true);
24  $aGroups = $oDb->get(MYSQLI_ASSOC);
25
26  $aGroups[] = [
27      'id'     => null,
28      'k_id'  => $_SESSION['k_id'],
29      'name'  => 'Ohne Gruppe'
30  ];
31
32  /* =====
33  Einstellungen
34  ===== */
35
36  $oDb->query('SELECT * FROM menu_settings WHERE k_id = ?', ['s', $_SESSION['k_id']]);
37  $oDb->execute(true);
38  $aSettings = $oDb->get(MYSQLI_ASSOC)[0];
39
40  /* =====
41  Information - Außverkauft
42  ===== */
43
44  $oAlert->setContent('Dieser Artikel ist ausverkauft');
45  $oAlert->setType('warning');
46
47  /* =====
48  Tabs
49  ===== */
50
51  $oTabs = new snippetTab();
52
53  /* =====
54  Tabs Laden
55  ===== */
56
57  foreach ($aGroups AS $aGroup) {
58      $aGroupItems = cms::searchArray($aItems, 'group_id', $aGroup['id']);
59      if (!empty($aGroupItems)) {
60
61          ob_start();
62          echo '<div class="container">';
63          echo '<div class="row d-flex align-items-stretch">';
64          foreach ($aGroupItems AS $aItem) {

```

```

65         if (!$aItem['menu']) continue;
66
67         echo '<div class="p-2 col-md-4 col-6" h-100>';
68         echo '<div class="article-card card shadow-sm p-2 justify-
content-between d-flex align-items-center h-100">';
69         if ($aSettings['show_img']) {
70             echo cms::img($aItem['img'], 'uploads',
$aSettings['img_size']);
71         }
72         echo '<div class="w-100 text-center">';
73         echo '<p><strong>' . $aItem['name'] . '</strong><br><span
class="text-success">' . cms::convertValue($aItem['price'], 'price', ['unit' => '€']) . '</span>
<p>';
74         echo $oInput->getNumber('Menge:', ['value' => '0', 'data-
article' => $aItem['id'], 'min' => '0', 'max' => $aSettings['article_max'], 'disabled' =>
$aItem['sold_out']]);
75         if ($aItem['sold_out']) {
76             echo $oAlert->view();
77         }
78         echo '</div>';
79         echo '</div>';
80         echo '</div>';
81     }
82     echo '</div>';
83     echo '</div>';
84     $oTabs->addTab($aGroup['name'], ob_get_clean());
85
86 }
87 }
88
89 /* =====
90 Ansicht
91 ===== */
92
93 if ($aSettings['use_groups']) {
94     echo '<div class="container">';
95     echo $oTabs->view();
96     echo '</div>';
97 } else {
98     echo '<div class="container">';
99     echo '<div class="row d-flex align-items-stretch">';
100     foreach ($aItems AS $aItem) {
101         if (!$aItem['menu']) continue;
102
103         echo '<div class="p-2 col-md-4 col-6" h-100>';
104         echo '<div class="card p-2 justify-content-between d-flex align-items-
center h-100">';
105         if ($aSettings['show_img']) {
106             echo cms::img($aItem['img'], 'uploads', $aSettings['img_size']);
107         }
108         echo '<div class="w-100 text-center">';
109         echo '<p><strong>' . $aItem['name'] . '</strong><br><span
class="text-success">' . cms::convertValue($aItem['price'], 'price', ['unit' => '€']) . '</span>
<p>';
110         echo $oInput->getNumber('Menge:', ['value' => '0', 'data-article'
=> $aItem['id'], 'min' => '0', 'max' => $aSettings['article_max'], 'disabled' =>
$aItem['sold_out']]);
111         if ($aItem['sold_out']) {
112             echo $oAlert->view();

```

```

113         }
114         echo '</div>';
115     echo '</div>';
116     echo '</div>';
117 }
118 echo '</div>';
119 echo '</div>';
120 }
121

```

#### system\frontend\pages\menu\_edit.php

```

1 <?php
2     $oTable = new snippetTable('table-hover text-center');
3     $oTable->setSearchable(true);
4     $oTable->setNoDataMessage('Es wurden keine Artikel gefunden.<br>Erstelle deinen ersten
Artikel.');
```

5 \$oTable->setTop('<button id="newArticle" class="btn btn-success">Neuer Artikel</button>');

6

7 \$oDb = new db();

8 \$oDb->query("SELECT \* FROM inventory WHERE k\_id = ?", ['s', \$\_SESSION['k\_id']]);

9 \$oDb->execute(true);

10 \$aResult = \$oDb->get(MYSQLI\_ASSOC);

11

12 \$oDb->query("SELECT \* FROM `category` WHERE k\_id = ?", ['s', \$\_SESSION['k\_id']]);

13 \$oDb->execute(true);

14 \$aGroups = \$oDb->get(MYSQLI\_ASSOC);

15

16 \$oTable->head();

17 \$oTable->th('Bild');

18 \$oTable->th('Name');

19 \$oTable->th('SKU');

20 \$oTable->th('Preis');

21 \$oTable->th('Anzahl');

22 \$oTable->th('Ausverkauft');

23 \$oTable->th('Im Menü');

24 \$oTable->th('Gruppe');

25 \$oTable->th('Aktion');

26

27 \$oTable->body();

28 foreach (\$aResult AS \$aData) {

29 \$oTable->row();

30 \$oTable->td(cms::img((\$aData['img'] ?? 'logo'), 'uploads', 50));

31 \$oTable->td(\$aData['name']);

32 \$oTable->td(\$aData['sku']);

33 \$oTable->td(cms::convertValue(\$aData['price'], 'price', ['unit' => '€']));

34 \$oTable->td(\$aData['quantity'] ?? '---');

35 \$oTable->td('<div class="form-check form-switch w-100 d-flex justify-content-center">
<input class="form-check-input sold-out" data-item-id="' . \$aData['id'] . '" type="checkbox"
role="switch" id="switchCheckDefault" ' . ((int)\$aData['sold\_out'] === 1 ? 'checked' : '') . '>
</div>');

36 \$oTable->td('<div class="form-check form-switch w-100 d-flex justify-content-center">
<input class="form-check-input in-menu" data-item-id="' . \$aData['id'] . '" type="checkbox"
role="switch" id="switchCheckDefault" ' . ((int)\$aData['menu'] === 1 ? 'checked' : '') . '>
</div>');

37 \$oTable->td(cms::searchArray(\$aGroups, 'id', \$aData['group\_id'])[0]['name'] ??
\$aData['group\_id']);

38 \$oTable->td('<a class="btn btn-warning">' . cms::getIcon('pen') . '</a> <a data-del\_id=' .
\$aData['id'] . '" class="btn btn-danger del-article">' . cms::getIcon('trash') . '</a>');

```

39     }
40
41     echo '<div class="container">';
42         echo $oTable->view();
43     echo '</div>';
44
45     $oNotify = new snippetNotify();
46     $oNotify->viewToast();

```

#### system\frontend\pages\menu\_settings.php

```

1  <?php
2      /* =====
3      Variablen
4      ===== */
5
6      $oInput = new snippetInput();
7      $aData = [];
8
9      /* =====
10     Tabs Initialisieren
11     ===== */
12
13     $oTab = new snippetTab();
14     $oTab->noDataMessage(true);
15
16     /* =====
17     Datenverarbeitung
18     ===== */
19
20     /* =====
21     Einstellungen
22     ===== */
23
24     $oDb = new db();
25     $oDb->query('SELECT * FROM menu_settings WHERE k_id = ?', ['s', $_SESSION['k_id']]);
26     $oDb->execute(true);
27     $aData = $oDb->get(MYSQLI_ASSOC);
28
29     if (empty($aData)) {
30         $oDb->query('INSERT INTO `menu_settings` (`k_id`) VALUES (?)', ['s',
31 $_SESSION['k_id']]);
32         $oDb->execute();
33         $oDb->query('SELECT * FROM menu_settings WHERE k_id = ?', ['s', $_SESSION['k_id']]);
34         $oDb->execute(true);
35         $aData = $oDb->get(MYSQLI_ASSOC);
36     }
37     $aData = $aData[0];
38
39     /* =====
40     Gruppen
41     ===== */
42
43     $oDb->query('SELECT * FROM category WHERE k_id = ?', ['s', $_SESSION['k_id']]);
44     $oDb->execute(true);
45     $aGroups = $oDb->get(MYSQLI_ASSOC);
46
47     /* =====
48     Seiten

```

```

48  ===== */
49
50  /* =====
51  Einstellungen
52  ===== */
53
54  ob_start();
55      echo '<h2>Bild</h2><hr>';
56      echo $oInput->getCheckbox('Bild anzeigen:', ['data-k_id' => $_SESSION['k_id'], 'class' =>
'menu-setting', 'data-setting' => 'show_img'], $aData['show_img']);
57      echo $oInput->getNumber('Bild grÖÙe:', ['data-k_id' => $_SESSION['k_id'], 'class' =>
'menu-setting', 'data-setting' => 'img_size', 'min' => '25', 'max' => '70', 'value' =>
$aData['img_size'] ?? '25']);
58      echo '<h2>Artikel</h2><hr>';
59      echo $oInput->getNumber('Maximal Anzahl:', ['data-k_id' => $_SESSION['k_id'], 'class' =>
'menu-setting', 'data-setting' => 'article_max', 'min' => '0', 'max' => '100', 'value' =>
$aData['article_max'] ?? '10']);
60      echo $oInput->getCheckbox('Darstellung in Gruppen:', ['data-k_id' => $_SESSION['k_id'],
'class' => 'menu-setting', 'data-setting' => 'use_groups'], $aData['use_groups']);
61      $oTab->addTab('Ansicht', ob_get_clean());
62
63  /* =====
64  Gruppen
65  ===== */
66
67      $oTable = new snippetTable();
68      $oTable->head();
69      $oTable->setNoDataMessage('Es wurden keine Gruppen gefunden');
70
71
72      $oTable->th('Gruppe');
73      $oTable->th('Name');
74      $oTable->th('Aktion');
75
76      $oTable->body();
77
78      foreach ($aGroups AS $aGroup) {
79          $oTable->row();
80          $oTable->th($aGroup['id']);
81          $oTable->td($aGroup['name']);
82      }
83
84      ob_start();
85      echo '<div class="w-100 d-flex justify-content-end"><button class="btn btn-success new-
group">' . cms::getIcon('plus') . ' Neue Gruppe erstellen</button></div>';
86      echo $oTable->view();
87      $oTab->addTab('Gruppen', ob_get_clean());
88
89  /* =====
90  Ansicht
91  ===== */
92
93      echo '<div class="container">';
94          echo $oTab->view();
95      echo '</div>';
96
97      echo '<div class="savebar w-100 position-fixed bottom-0 start-0 mb-5 shadow fade"><div
class="container border border-warning bg-body-tertiary p-2 rounded d-flex justify-content-
between align-items-center">

```

```

98     <span>Es wurden Änderungen vorgenommen. Sollen diese gespeichert werden?</span>
99     <button class="btn btn-success">Speichern</button>
100 </div></div>';
101
102
103 ?>
104
105 <div id="saveBar"
106     class="alert alert-warning position-fixed bottom-0 start-0 w-100 rounded-0 mb-0 shadow">
107 <div class="container d-flex justify-content-between align-items-center">
108     <span>
109         <strong>Ungespeicherte Änderungen</strong>
110     </span>
111
112     <div>
113         <button class="btn btn-secondary me-2" id="cancelBtn">
114             Verwerfen
115         </button>
116         <button class="btn btn-primary" id="saveBtn">
117             Speichern
118         </button>
119     </div>
120 </div>
121 </div>

```

#### system\frontend\styles\style.css

```

1 body {
2     padding: 0;
3     margin: 0;;
4 }
5
6 * {
7     padding: 0;
8     margin: 0;
9 }

```