

```
1  -- phpMyAdmin SQL Dump
2  -- version 4.1.14
3  -- http://www.phpmyadmin.net
4  --
5  -- Client : 127.0.0.1
6  -- Généré le : Mar 12 Mai 2015 à 15:50
7  -- Version du serveur : 5.6.17
8  -- Version de PHP : 5.5.12
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET time_zone = "+00:00";
12
13
14 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
15 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
16 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
17 /*!40101 SET NAMES utf8 */;
18
19 --
20 -- Base de données : `catal_info_bdd`
21 --
22 CREATE DATABASE IF NOT EXISTS `catal_info_bdd` DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci;
23 USE `catal_info_bdd`;
24
25 -----
26
27 --
28 -- Structure de la table `brands`
29 --
30
31 CREATE TABLE IF NOT EXISTS `brands` (
32   `id` int(11) NOT NULL AUTO_INCREMENT,
33   `name` varchar(64) NOT NULL,
34   PRIMARY KEY (`id`)
35 ) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=22 ;
36
37 --
38 -- Vider la table avant d'insérer `brands`
39 --
40
41 TRUNCATE TABLE `brands`;
42 --
43 -- Contenu de la table `brands`
44 --
45
46 INSERT INTO `brands` (`id`, `name`) VALUES
47 (1, 'Nvidia'),
48 (2, 'Asus'),
49 (3, 'MSI'),
50 (4, 'Gigabyte'),
51 (5, 'Corsair'),
52 (6, 'Cooler Master'),
53 (7, 'Kingston'),
54 (8, 'HP'),
55 (9, 'Logitech'),
56 (10, 'AMD'),
```

```

57  (11, 'Intel'),
58  (12, 'EVGA'),
59  (13, 'DELL'),
60  (14, 'Cyborg'),
61  (15, 'Mad Catz'),
62  (16, 'Razer'),
63  (17, 'Samsung'),
64  (18, 'Sony'),
65  (19, 'Phanteks'),
66  (20, 'BitFenix'),
67  (21, 'Apple');
68
69  -- -----
70
71  --
72  -- Structure de la table `keywords`
73  --
74
75  CREATE TABLE IF NOT EXISTS `keywords` (
76    `id` int(11) NOT NULL AUTO_INCREMENT,
77    `name` varchar(64) NOT NULL,
78    PRIMARY KEY (`id`)
79  ) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=19 ;
80
81  --
82  -- Vider la table avant d'insérer `keywords`
83  --
84
85  TRUNCATE TABLE `keywords`;
86  --
87  -- Contenu de la table `keywords`
88  --
89
90  INSERT INTO `keywords` (`id`, `name`) VALUES
91  (1, 'carte graphique'),
92  (2, 'carte mere'),
93  (3, 'ram'),
94  (4, 'boitier'),
95  (5, 'processeur'),
96  (6, 'clavier'),
97  (7, 'souris'),
98  (8, 'ecran'),
99  (9, 'carte son'),
100 (10, 'carte reseau'),
101 (11, 'disque dur'),
102 (12, 'lecteur cd'),
103 (13, 'ventilateur'),
104 (14, 'graveur cd'),
105 (15, 'cle usb'),
106 (16, 'chipset'),
107 (17, 'alimentation'),
108 (18, 'gaming');
109
110  -- -----
111  --
112  --
113  -- Structure de la table `medias`

```

```
114  --
115
116  CREATE TABLE IF NOT EXISTS `medias` (
117    `id` int(11) NOT NULL AUTO_INCREMENT,
118    `src` varchar(255) NOT NULL,
119    `isImage` tinyint(1) NOT NULL,
120    PRIMARY KEY (`id`)
121  ) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=55 ;
122
123  --
124  -- Vider la table avant d'insérer `medias`
125  --
126
127  TRUNCATE TABLE `medias`;
128  --
129  -- Contenu de la table `medias`
130  --
131
132  INSERT INTO `medias` (`id`, `src`, `isImage`) VALUES
133  (1, 'up-content/img/gtx770_1.jpg', 1),
134  (2, 'up-content/img/blackwidow_1.jpg', 1),
135  (3, 'up-content/img/blackwidow_2.jpg', 1),
136  (4, 'up-content/img/blackwidow_3.jpg', 1),
137  (5, 'up-content/other/blackwidow_manual.pdf', 0),
138  (6, 'up-content/img/razernaga_1.png', 1),
139  (7, 'up-content/other/blackwidow_quick_start.pdf', 0),
140  (8, 'up-content/img/Carbide300R_1.jpg', 1),
141  (9, 'up-content/img/Carbide300R_2.jpg', 1),
142  (10, 'up-content/img/FORCELS_1.jpg', 1),
143  (11, 'up-content/img/GT610-SL-1GD3-L_1.jpg', 1),
144  (12, 'up-content/img/GTX960_1.jpg', 1),
145  (13, 'up-content/img/IntelXeon_1.jpg', 1),
146  (14, 'up-content/img/Inteli7_1.jpg', 1),
147  (15, 'up-content/img/RM650_1.jpg', 1),
148  (16, 'up-content/img/RM650_2.jpg', 1),
149  (17, 'up-content/img/RM650_3.jpg', 1),
150  (18, 'up-content/img/VENGENCE_1.jpg', 1),
151  (19, 'up-content/img/XMS3_1.jpg', 1),
152  (20, 'up-content/other/XMS3_2.pdf', 0),
153  (21, 'up-content/other/02G-P4-2771554c9b0ad059f.pdf', 0),
154  (22, 'up-content/img/evga_acx_g1554c9b0ad3868.jpg', 1),
155  (23, 'up-content/other/evga_com1554c9b0ad4420.pdf', 0),
156  (24, 'up-content/img/evga-gefor1554c9b0ad5b90.jpg', 1),
157  (25, 'up-content/img/img_84381554c9b0ad6b30.jpg', 1),
158  (26, 'up-content/img/img_85681554c9b0ad76e8.jpg', 1),
159  (27, 'up-content/other/xeon-e3-121554cc2bc0765d.pdf', 0);
160
161  -- -----
162
163  --
164  -- Structure de la table `products`
165  --
166
167  CREATE TABLE IF NOT EXISTS `products` (
168    `id` int(11) NOT NULL AUTO_INCREMENT,
169    `title` varchar(64) NOT NULL,
170    `short_desc` varchar(255) NOT NULL,
```

```

171 `long_desc` text NOT NULL,
172 `is_frontpage` tinyint(1) NOT NULL,
173 `availability_date` date NOT NULL,
174 `expiration_date` date NOT NULL,
175 `view_count` int(11) NOT NULL,
176 `id_brand` int(11) NOT NULL,
177 PRIMARY KEY (`id`)
178 ) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=18 ;
179
180 --
181 -- Vider la table avant d'insérer `products`
182 --
183
184 TRUNCATE TABLE `products`;
185 --
186 -- Contenu de la table `products`
187 --
188
189 INSERT INTO `products` (`id`, `title`, `short_desc`, `long_desc`, `is_frontpage`, `
availability_date`, `expiration_date`, `view_count`, `id_brand`) VALUES
190 (1, 'GeForce GTX 770', 'NVidia 2048 MB Graphics card', 'The GeForce 700 Series is a
family of graphics processing units developed by Nvidia, used in desktop and laptop
PCs. It is mainly based on a refresh of the Kepler microarchitecture (GK-codenamed
chips) used in the previous GeForce 600 Series, but also includes cards based on the
previous Fermi (GF) and later Maxwell (GM) architectures. A number of GeForce 700
series chips were released for mobile devices in April 2013. GeForce 700 series
cards were first released in May 2013, starting with the release of the GeForce GTX
Titan on February 19, 2013, and the GeForce GTX 780 on May 23, 2013.', 0,
'2015-04-29', '2015-05-31', 114, 1),
191 (2, 'Blackwidow Ultimate', 'Cherry MX Blue mechanical keyboard', 'The Razer
BlackWidow uses a PC layout but with 5 macro keys on the left-hand side. The key
layout uses 1.5u-sized modifiers on the bottom row, with 1u Windows, Fn and Menu
keys, as well as a 6u Space bar. There is also a version with US-ANSI/Macintosh
layout but otherwise the same physical keys.\r\n\r\nA unique feature of the
BlackWidow is that the function keys are shifted slightly to the right, so that they
are parallel to the WASD cluster.', 1, '2015-04-29', '2015-05-31', 28, 16),
192 (3, 'Naga Epic', '17 Buttons, Optimized 7800DPI Gaming Mouse', 'In any MMO, the
more spells you have instant access to, the greater your advantage becomes.
\r\nThe Razer Naga Epic features a total of 19 buttons, including the iconic Razer
Naga 12-button thumb grid to maximize the actions at your disposal. Now letting you
map and access even more abilities, macros, as well as controls, the Razer Naga Epic
makes it easy to hit the right button at the right time.\r\n\r\nThe Razer Naga Epic
gaming-grade wireless technology gives you completely unrestricted \r\nmovement
without compromising on mouse control or accuracy. With a response rate of 1ms
whether wired or wireless, your Razer Naga Epic delivers peak performance all the
time. Furthermore, by plugging in the USB when the battery is running low, your
Razer Naga Epic charges while you use it, so you get zero downtime all the time.', 1,
'2015-04-29', '2015-05-31', 31, 16),
193 (4, 'RM650 80 PLUS', 'Alimentation Gold Certified Fully Modular PSU', 'Reliable
operation, fully modular cabling, and optimized for silence.\r\nThe Corsair RM650 is
fully modular and optimized for silence and high efficiency. Built with low-noise
capacitors and transformers, and Zero RPM Fan Mode ensures that the fan does not
spin unless the power supply is under heavy load. And with a fan that is
custom-designed for low noise operation.\r\n\r\n80 PLUS Gold rated efficiency saves
you money on your power bill, and the low-profile black cables are fully modular, so
you can enjoy fast, neat builds. And, like all Corsair power supplies, the RM650 is
built with high-quality components and is guaranteed to deliver clean, stable,

```

continuous power. Want even more? Connect it to your Corsair Link system (available separately) and you can even monitor fan speed and +12V current directly from your desktop.', 0, '2015-04-29', '2015-05-31', 18, 5),

194 (5, 'Carbide 300R Black', '(3 x 5.25) Compact PC Gaming Case - No power', 'Everything You Need, Nothing You Don't.\r\n\r\nA compact expression of Corsair's gaming philosophy.\r\nGreat systems start with a great case, and Carbide Series 300R provides a remarkable number of features in a compact chassis. Easy access, lots of room for expansion, and superior cooling make the 300R a great choice for building powerful PCs that don't take up a lot of room.\r\n\r\nSerious cooling potential.\r\nThe 300R is equipped with two fans for intake and exhaust, and the integrated dust filters on the front and bottom intakes are easy removable. There's room for you to install five more fans, including mounting points for two side panel 120mm fans for drawing cool air directly over your GPUs. If you're building a multi-GPU gaming rig, proper graphics ventilation is essential.', 0, '2015-04-29', '2015-05-31', 56, 5),

195 (6, 'Force Series LS', 'SSD Drive 60 GB 2.5 Cm2 of diameter', 'With a sequential read speed of up to 560 MB/s, the Force Series LS from Corsair is a force to deal with. This SATA 6 Gbps solid state drive is available in three capacities namely, 60 GB, 120 GB and 240 GB. Each of these variants can reach a maximum sequential write speed of 535 MB/s. The Force Series LS follows a 7 mm form factor and absorbs 4.6 W when active, making it ideal for notebook computers.\r\nThe Phison PS3108-S8 controller inside the Force Series LS is an 8-channel controller capable of supporting up to 1 TB of flash storage capacity. Among its features are AES Encryption, S.M.A.R.T., NCQ/TRIM and DEVSLP mode. Together with this controller are MLC Toggle NAND modules and DRAM caches. The 240 GB model has a DRAM cache of 512 MB while the 120 GB and 60 GB has a DRAM cache of 256 MB.\r\nLike a typical Corsair SSD, the Force Series LS has a three-year warranty and a Mean Time Between Failure (MTBF) of 1,000,000 hours.', 1, '2015-04-29', '2015-05-31', 16, 5),

196 (7, 'Dual Channel Vengeance', 'DDR3 8GB [2x4GB] DDR1600 (PC3-12800)', 'Speed Rating : PC3-12800 \r\nTested Speed : 1600Mhz \r\nSize : 8GB Kit (2 x 4GB) \r\nTested Latency : 9-9-9-24 \r\nTested Voltage : 1.5 \r\nPerformance Profile : XMP \r\nSPD Speed : 1333Mhz \r\nSPD Latency : 9-9-9-24 \r\nPackage : 240pin DIMM \r\nFan Included : No \r\n\r\nVengeance DDR3 memory modules use high performance RAM, run at 1.5V, and have the aggressive look you want, all at an attractively low price. \r\n\r\nGreat Looking, Great Overclocking Memory at a Great Price\r\nCorsair Vengeance DDR3 memory modules are designed with overclockers in mind. Vengeance DIMMs are built using RAM specially selected for their high-performance potential. Aluminum heat spreaders help dissipate heat, and provide the aggressive look that you want in your gaming rig. As a bonus, the attractive low price of Vengeance memory will also leave lots of room in your system build budget.\r\n\r\nOptimized for Compatibility with the Latest CPUs and Motherboards\r\nVengeance memory is designed specifically for the latest CPUs. The 1.5V VDIMM spec ensures you get the performance you expect, even without increasing memory voltages. Vengeance DDR3 memory is available in single modules, and two or three module kits, making it easy for you to match the DIMM population requirements of your motherboard.\r\n\r\nNext-Generation Density for the Ultimate Power User\r\nMost Vengeance DIMMs are built with two gigabit RAM ICs. These extra-dense memory chips allow you to have 8GB of memory using only two DIMMs, or to populate your triple channel system with up to an insanely large 24GB of system memory for extreme multitasking performance.', 1, '2015-04-29', '2015-05-31', 59, 5),

197 (8, 'XMS3', 'DDR3 32GB [4x8GB] DDR1600 (PC3-12800)', 'Taille 32GB Kit (4 x 8GB)\r\nProfil des performances XMP\r\nVentilateur compris Non\r\nR  partiteur de chaleur XMS\r\nConfiguration de la m  moire Dual / Quad Channel\r\nType de m  moire DDR3\r\nProduit - Broches de la m  moire 240\r\nProduit - Format de la m  moire DIMM\r\nTension test  e 1.5\r\nTension SPD 1.5\r\nCote de vitesse PC3-12800 (1600MHz)\r\nVitesse SPD 1333MHZ\r\nVitesse test  e 1600Mhz\r\nLatence test  e 11-11-11-30\r\nLatence SPD 9-9-9-24', 0, '2015-04-29', '2015-05-31', 32, 5),

```

198      (9, 'GeForce GTX 960', 'GeForce GTX960 2048MB DVI HDMI 3xDisplay port', 'The GeForce
      900 Series is a family of graphics processing units developed by Nvidia, used in
      desktop and laptop PCs. It serves as the high-end introduction for the Maxwell
      architecture (GM-codenamed chips), named after the Scottish theoretical physicist
      James Clerk Maxwell.\r\n\r\nThe Maxwell microarchitecture, the successor to Kepler
      microarchitecture, will for the first time feature an integrated ARM CPU of its own.
      This will make Maxwell GPUs more independent from the main CPU according to
      Nvidia''s CEO Jen-Hsun Huang. Nvidia expects three major things from the Maxwell
      architecture: improved graphics capabilities, simplified programming as well as
      better energy-efficiency compared to the GeForce 700 Series and GeForce 600 Series
      \r\n\r\nMaxwell was announced in September 2010. The first GeForce consumer-class
      products based on the Maxwell architecture were released in early 2014.[ Nvidia is
      expected to release the Maxwell-powered Tesla accelerator cards as well as Quadro
      professional graphics cards based on this architecture in late 2014. Eventually,
      Maxwell architecture will be used for mobile application processors that belong to
      the Erista family of Tegra SoCs.', 1, '2015-04-29', '2015-05-31', 89, 12)),
199      (10, 'GT610-SL-lGD3-L', 'GeForce GT610 1024MB DVI/HDMI ', 'Where the goal of the
      previous architecture, Fermi, was to increase raw performance (particularly for
      compute and tessellation), Nvidia''s goal with the Kepler architecture was to
      increase performance per watt, while still striving for overall performance
      increases. The primary way Nvidia achieved this goal was through the use of a
      unified clock. By abandoning the shader clock found in their previous GPU designs,
      efficiency is increased, even though it requires more cores to achieve similar
      levels of performance. This is not only because the cores are more power efficient
      (two Kepler cores using about 90% of the power of one Fermi core, according to
      Nvidia''s numbers), but also because the reduction in clock speed delivers a 50%
      reduction in power consumption in that area.', 0, '2015-04-29', '2015-05-31', 10, 2)),
200      (11, 'Quad Core i7-4770S 3.1GHz', '3.1GHz up to 3.90 GHz & HD Graphics 4600', 'Intel
      Core is a brand name that Intel uses for various mid-range to high-end consumer and
      business microprocessors. These processors replaced the then-currently mid to high
      end Pentium processors, making them entry level, and bumping the Celeron series of
      processors to low end. Similarly, identical or more capable versions of Core
      processors are also sold as Xeon processors for the server and workstation
      market.\r\n\r\nAs of 2015 the current lineup of Core processors included the latest
      Intel Core i7, Intel Core i5, and Intel Core i3', 0, '2015-04-29', '2015-05-31', 39,
      11)),
201      (12, 'Xeon Quad Core E3-1276V3', '3.6GHz LGA1150 - 8MB - 22 nm - 84W', 'The Xeon
      /?zi??n/ is a brand of x86 microprocessors designed and manufactured by Intel
      Corporation, targeted at the non-consumer workstation, server, and embedded system
      markets. Primary advantages of the Xeon CPUs, when compared to the majority of
      Intel''s desktop-grade consumer CPUs, are their multi-socket capabilities, higher
      core counts, and support for ECC memory.', 0, '2015-04-29', '2015-05-31', 20, 11)),
202      (13, 'GTX770 Superclocked ACX', 'Overclocked graphics card by EVGA', 'Fast just got
      faster with the next-generation EVGA GeForce GTX 770, a high-performance graphics
      card designed from the ground up to deliver high-speed, super-smooth
      gaming.\r\n\r\nEVGA is also introducing a brand new cooling design; the redefining
      EVGA ACX Cooler. With a 40% increase in heatsink volume, the EVGA ACX is more
      efficient at dissipating heat, allowing for 15% lower GPU temperatures. A
      reinforcement baseplate maintains a straight PCB, and helps lower mosfet
      temperatures by 7% and memory by 15%.', 1, '2015-04-29', '2015-05-31', 24, 1));
203
204      -- -----
205
206      --
207      -- Structure de la table `products_has_keywords`
208      --
209

```

```
210 CREATE TABLE IF NOT EXISTS `products_has_keywords` (  
211     `id_products` int(11) NOT NULL,  
212     `id_keywords` int(11) NOT NULL,  
213     PRIMARY KEY (`id_products`,`id_keywords`)  
214 ) ENGINE=MyISAM DEFAULT CHARSET=utf8;  
215  
216 --  
217 -- Vider la table avant d'insérer `products_has_keywords`  
218 --  
219  
220 TRUNCATE TABLE `products_has_keywords`;  
221 --  
222 -- Contenu de la table `products_has_keywords`  
223 --  
224  
225 INSERT INTO `products_has_keywords` (`id_products`, `id_keywords`) VALUES  
226 (1, 1),  
227 (2, 6),  
228 (3, 7),  
229 (4, 17),  
230 (4, 18),  
231 (5, 4),  
232 (5, 18),  
233 (6, 11),  
234 (7, 3),  
235 (7, 18),  
236 (8, 3),  
237 (9, 1),  
238 (9, 18),  
239 (10, 1),  
240 (11, 5),  
241 (11, 16),  
242 (12, 5),  
243 (12, 16),  
244 (13, 1),  
245 (13, 18);  
246  
247 -- -----  
248  
249 --  
250 -- Structure de la table `products_has_medias`  
251 --  
252  
253 CREATE TABLE IF NOT EXISTS `products_has_medias` (  
254     `id_products` int(11) NOT NULL,  
255     `id_medias` int(11) NOT NULL,  
256     PRIMARY KEY (`id_products`,`id_medias`)  
257 ) ENGINE=MyISAM DEFAULT CHARSET=utf8;  
258  
259 --  
260 -- Vider la table avant d'insérer `products_has_medias`  
261 --  
262  
263 TRUNCATE TABLE `products_has_medias`;  
264 --  
265 -- Contenu de la table `products_has_medias`  
266 --
```

```
267
268 INSERT INTO `products_has_medias` (`id_products`, `id_medias`) VALUES
269 (1, 1),
270 (2, 2),
271 (2, 3),
272 (2, 4),
273 (2, 5),
274 (2, 7),
275 (3, 6),
276 (4, 15),
277 (4, 16),
278 (4, 17),
279 (5, 8),
280 (5, 9),
281 (6, 10),
282 (7, 18),
283 (8, 19),
284 (8, 20),
285 (9, 12),
286 (10, 11),
287 (11, 14),
288 (12, 13),
289 (12, 27),
290 (13, 21),
291 (13, 22),
292 (13, 23),
293 (13, 24),
294 (13, 25),
295 (13, 26);
296
297 -----
298
299 --
300 -- Structure de la table `users`
301 --
302
303 CREATE TABLE IF NOT EXISTS `users` (
304   `id` int(11) NOT NULL AUTO_INCREMENT,
305   `username` varchar(64) NOT NULL,
306   `password` varchar(64) NOT NULL,
307   `is_admin` tinyint(1) NOT NULL,
308   PRIMARY KEY (`id`)
309 ) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=3 ;
310
311 --
312 -- Vider la table avant d'insérer `users`
313 --
314
315 TRUNCATE TABLE `users`;
316 --
317 -- Contenu de la table `users`
318 --
319
320 INSERT INTO `users` (`id`, `username`, `password`, `is_admin`) VALUES
321 (1, 'Admin', '43cd17b0b3dc3bfa9d45d6f3e6688cbb73e2b65d', 1),
322 (2, 'User', '38758f53d77d5217d477433658b49a301f435feb', 0);
323
```



```
324  -- -----
325
326  --
327  -- Structure de la table `users_has_products`
328  --
329
330  CREATE TABLE IF NOT EXISTS `users_has_products` (
331    `id_users` int(11) NOT NULL,
332    `id_products` int(11) NOT NULL,
333    PRIMARY KEY (`id_users`,`id_products`)
334  ) ENGINE=MyISAM DEFAULT CHARSET=utf8;
335
336  --
337  -- Vider la table avant d'insérer `users_has_products`
338  --
339
340  TRUNCATE TABLE `users_has_products`;
341  /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
342  /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
343  /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
344
```

```
1  .mini-submenu{
2      display:block;
3      background-color: white;
4      border: 1px solid gray;
5      border-radius: 4px;
6      padding: 9px;
7      width: 42px;
8      z-index: 1;
9      position: absolute;
10     margin-bottom: 20px;
11 }
12
13 .mini-submenu:hover{
14     cursor: pointer;
15 }
16
17 .mini-submenu .icon-bar {
18     border-radius: 1px;
19     display: block;
20     height: 2px;
21     width: 22px;
22     margin-top: 3px;
23 }
24
25 .mini-submenu .icon-bar {
26     background-color: black;
27 }
28
29 select[multiple], select[size] {
30     height: 200px;
31 }
32
33 #slide-submenu{
34     display: inline-block;
35     padding: 0 8px;
36     border-radius: 4px;
37     cursor: pointer;
38 }
39
40 .pull-right{
41     background-color: #ADADAD;
42     color: white;
43 }
44
45 .pull-right:hover{
46     background-color: #CCCCCC;
47     color: white;
48 }
49
50 .list-group{
51     position: absolute;
52     min-width: 270px;
53     z-index: 1;
54 }
55
56 .carousel-indicators li{
57     border-radius: 35%;
```

```
58     border-color: black;
59 }
60
61 .carousel-inner{
62     max-height: 250px;
63 }
64
65 .panel{
66     -webkit-transition: all 0.2s ease-in-out;
67     -moz-transition: all 0.2s ease-in-out;
68     -o-transition: all 0.2s ease-in-out;
69     transition: all 0.2s ease-in-out;
70 }
71
72 .panel-footer{
73     width: 100%;
74 }
75
76 .thumbnail{
77     min-height: 170px;
78     max-height: 170px;
79     border: 1px #FFFFFF solid;
80 }
81
82 .carousel-inner .thumbnail{
83     min-height: 250px;
84     max-height: 250px;
85     border: 1px #FFFFFF solid;
86 }
87
88 #carousel-example-generic{
89     border: 1px #CCCCCC solid;
90 }
91
92 .img-thumbnail{
93     max-height: 200px;
94     border: none;
95     margin: auto;
96     padding-top: 50px;
97 }
98
99 .red{
100     color: #FF6961;
101 }
102
103 .green{
104     color: #23BA7E;
105 }
106
107 .search-bar{
108     width: 225%;
109 }
110
111 .form-container{
112     max-width: 700px;
113     margin: auto;
114 }
```

```
115     .media-object{
116         width: 64px;
117         height: 64px;
118     }
119     /* Quand on passe en mode mobile, on change les propriétés suivantes */
120     @media (max-width: 767px) {
121         .list-group{
122             position: relative;
123         }
124
125         .mini-submenu{
126             position: relative;
127         }
128         .search-bar{
129             width: 100%;
130         }
131
132         h1{
133             font-size: 25px;
134         }
135     };
```

```

1  <?php
2
3  /*
4  +-----+-----+-----+-----+-----+-----+-----+-----+-----+
5  |   BASICS BDD
6  +-----+-----+-----+-----+-----+-----+-----+-----+-----+
7  | Auteur :      | SEEMULLER
  Julien
8  | Classe :      |
  I.IN-P4B
9  | Date :        |
  08.05.2015
10 | Version :     |
  1.0
11 |
12 | Description :  | Script contenant les fonctions basiques relative à la base de
  donnée |
13 |               | -> Connexion base de donnée
  (PDO)
14 |               | -> Retourne le nombre d'enregistrement dans la table
  donnée
15 |               | -> Retourne un enregistrement par rapport à
  l'id
16 |               | -> Retourne tout les enregistrements d'une table
  donnée
17 |               | -> Supprime un enregistrement de la table donnée par rapport à
  l'id |
18 +-----+-----+-----+-----+-----+-----+-----+-----+-----+
19 */
20
21 require_once 'constantes.php';
22
23 /** connection
24  * Fonction qui retourne une connexion PDO à une base de donnée mysql ou une erreur.
25  * @return PDO
26  */
27 function connection() {
28     // On essaie de se connecter à la base à l'aide d'informations contenues
29     // dans des constantes, si cela échoue, on renvoie FALSE et si on s'est bien
30     // connecté à la base de donnée, on renvoie TRUE
31     try {
32         $bdd = new PDO('mysql:host=' . DB_HOST . ';dbname=' . DB_NAME, DB_LOGIN,
33             DB_PASS, array(PDO::ATTR_PERSISTENT => true));
34
35         return $bdd;
36     } catch (PDOException $e) {
37         print "Erreur !: " . $e->getMessage() . "<br/>";
38         die();
39     }
40 }
41 /** countFields

```

```
42  * Fonction qui retourne le nombre total d'enregistrement d'une table
43  * passée en paramètre
44  * @param $table string
45  * @return Integer
46  */
47  function countFields($table) {
48      $dbc = connection();
49      $dbc->quote($table);
50      //On compte tous les champs de la table donnée en paramètre
51      $req = "SELECT COUNT(*) FROM $table";
52
53      //On prépare la requete
54      $requPrep = $dbc->prepare($req);
55      $requPrep->execute();
56
57      //On récupère le nombre de champs dans un array
58      $number = $requPrep->fetch();
59      $requPrep->closeCursor();
60      return $number[0];
61  }
62
63  /** countFieldsCondition
64   * Fonction qui retourne le nombre total d'enregistrement d'une table selon une
65   * condition passée en parametre
66   * @param string $table
67   * @param string $condition
68   * @return type
69   */
70  function countFieldsCondition($table, $condition) {
71      $dbc = connection();
72      $dbc->quote($table);
73      $dbc->quote($condition);
74      //On compte tous les champs de la table donnée en paramètre
75      //et on ajoute une condition
76      $req = "SELECT COUNT(*) FROM $table $condition";
77
78      //On prépare la requete
79      $requPrep = $dbc->prepare($req);
80      $requPrep->execute();
81
82      //On récupère le nombre de champs dans un array
83      $number = $requPrep->fetch();
84      $requPrep->closeCursor();
85      return $number[0];
86  }
87
88  /** getFieldById
89   * Cette fonction récupère un enregistrement de la table donnée en paramètre grâce à
90   * l'id également donnée en paramètre
91   * @param Integer $id
92   * @param String $table
93   * @return PDO::FETCH_OBJ
94   */
95  function getFieldById($id, $table, $type = PDO::FETCH_OBJ) {
96      $dbc = connection();
97      $dbc->quote($table);
98      //On récupère le champ avec l'id correspondant à celui envoyé en paramètre
```

```
99     $req = "SELECT * FROM $table WHERE id=:id";
100     //On prépare la requete
101     $requPrep = $dbc->prepare($req); // on prépare notre requête
102     $requPrep->bindParam(':id', $id, PDO::PARAM_INT);
103     $requPrep->execute();
104     //On renvoie le résultat dans format spécifique, (Par défaut : objet)
105     return $requPrep->fetch($type);
106 }
107
108 /** getFieldByIdCondition
109  * Cette fonction récupère un enregistrement de la table donnée en paramètre grâce à
110  * l'id également donnée en paramètre et une condition
111  * @param Integer $id
112  * @param String $table
113  * @return PDO::FETCH_OBJ
114  */
115 function getFieldByIdCondition($id, $table, $condition) {
116     $dbc = connection();
117     $dbc->quote($table);
118     //On récupère le champ avec l'id correspondant à celui envoyé en paramètre
119     //et on ajoute une condition
120     $req = "SELECT * FROM $table WHERE id=:id $condition";
121     //On prépare la requete
122     $requPrep = $dbc->prepare($req);
123     $requPrep->bindParam(':id', $id, PDO::PARAM_INT);
124     $requPrep->execute();
125     //On renvoie un objet contenant le résultat
126     return $requPrep->fetch(PDO::FETCH_OBJ);
127 }
128
129 /** getAllFields
130  * Cette fonction retourne tous les enregistrements de la table passée en paramètre
131  * @param String $table
132  * @return PDO::FETCH_OBJ
133  */
134 function getAllFields($table) {
135     $dbc = connection();
136     $dbc->quote($table);
137     //On récupère tous les champs de la table donnée en paramètre
138     $req = "SELECT * FROM $table";
139     //On prépare la requete
140     $requPrep = $dbc->prepare($req); // on prépare notre requête
141     $requPrep->execute();
142     $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
143     $requPrep->closeCursor();
144     //On renvoie un tableau d'objets contenant les données
145     return $data;
146 }
147
148 /** getAllFieldsCondition
149  * Cette fonction retourne tous les enregistrements de la table passée en paramètre
150  * avec une condition
151  * @param string $table
152  * @param string $condition
153  * @return PDO::FETCH_OBJ
154  */
155 function getAllFieldsCondition($table, $condition) {
```

```
155     $dbc = connection();
156     $dbc->quote($table);
157     $dbc->quote($condition);
158     //On récupère tous les champs de la table donnée en paramètre
159     //et on ajoute une condition
160     $req = "SELECT * FROM $table $condition";
161
162     //On prépare la requête
163     $requPrep = $dbc->prepare($req); // on prépare notre requête
164     $requPrep->execute();
165     $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
166     $requPrep->closeCursor();
167     //On renvoie un tableau d'objets contenant les données
168     return $data;
169 }
170
171 /** deleteFieldById
172  * Cette fonction supprime un enregistrement de la table donnée en paramètre grâce à
173  * l'id également donnée en paramètre
174  * @param Integer $id
175  * @param String $table
176  */
177 function deleteFieldById($id, $table) {
178     $dbc = connection();
179     $dbc->quote($table);
180     //On supprime le champ correspondant à l'id passé en paramètre
181     //dans la table passé en paramètre
182     $req = "DELETE FROM $table WHERE id=:id";
183
184     $requPrep = $dbc->prepare($req); // on prépare notre requête
185     $requPrep->bindParam(':id', $id, PDO::PARAM_INT);
186     $requPrep->execute();
187     $requPrep->closeCursor();
188 }
189
190 /** deleteFieldByIdCondition
191  * Cette fonction supprime un enregistrement de la table donnée en paramètre grâce à
192  * l'id également donnée en paramètre et une condition supplémentaire
193  * @param type $id
194  * @param type $table
195  * @param type $condition
196  */
197 function deleteFieldByIdCondition($id, $table, $condition) {
198     $dbc = connection();
199     $dbc->quote($table);
200     //On supprime le champ correspondant à l'id passé en paramètre
201     //dans la table passé en paramètre et on ajoute une condition
202     $req = "DELETE FROM $table $condition";
203
204     $requPrep = $dbc->prepare($req); // on prépare notre requête
205     $requPrep->bindParam(':id', $id, PDO::PARAM_INT);
206     $requPrep->execute();
207     $requPrep->closeCursor();
208 }
```



```
1  <?php
2
3  /*
4  +-----+-----+
5  |  CONSTANTES  |
6  +-----+-----+
7  |  Auteur :    | SEEMULLER Julien |
8  |  Classe :    | I.IN-P4B |
9  |  Date :      | 08.05.2015 |
10 |  Version :    | 1.0 |
11 |  Description : | Script regroupant les constantes |
12 |               | aux scripts php du site |
13 +-----+-----+
14  */
15
16 DEFINE('DB_LOGIN', 'root');
17 DEFINE('DB_PASS', '');
18 DEFINE('DB_HOST', 'localhost');
19 DEFINE('DB_NAME', 'catal_info_bdd');
20 DEFINE('PATH_ROOT', 'SEEMULLERJ_TPI_2015');
21
22 DEFINE('NUMBER_OF_TOP_PRODUCTS', 8);
23 DEFINE('NB_PAGINATION', 8);
24 DEFINE('IMG_FOLDER', 'up-content/img/');
25 DEFINE('OTHER_FOLDER', 'up-content/other/');
26 DEFINE('MAX_FILE_SIZE', 5242880); // (5 MO to Byte)
27
28 DEFINE('ALLOWED_IMAGE_TYPES', serialize( array("image/png", "image/jpeg", "image/gif"
))) ;
```

```

1  <?php
2  /*
3  +-----+-----+
4  |  CRUD BRANDS  |
5  +-----+-----+
6  | Auteur :      | SEEMULLER Julien |
7  | Classe :      | I.IN-P4B        |
8  | Date :        | 08.05.2015      |
9  | Version :     | 1.0              |
10 | Description :  | Script contenant les fonctions |
11 |               | en relation l'identification et le crud |
12 |               | des marques (CREATE, READ, UPDATE, DELETE) |
13 +-----+-----+
14  */
15
16 require_once 'basics_bdd.php';
17 $tableBrands = 'brands';
18
19
20 /** getBrands
21  * Recupère la liste de toute les marques
22  * @return type
23  */
24 function getBrands() {
25     global $tableBrands;
26     $dbc = connection();
27     $req = "SELECT * FROM $tableBrands";
28     $requPrep = $dbc->prepare($req); // on prépare notre requête
29     $requPrep->execute();
30     $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
31     $requPrep->closeCursor();
32
33     return $data;
34 }
35
36 /** getBrandsSorted
37  * Recupère la liste de toute les marques (triée)
38  * @return type
39  */
40 function getBrandsSorted() {
41     global $tableBrands;
42     $dbc = connection();
43     $req = "SELECT * FROM $tableBrands ORDER BY name";
44     $requPrep = $dbc->prepare($req); // on prépare notre requête
45     $requPrep->execute();
46     $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
47     $requPrep->closeCursor();
48
49     return $data;
50 }
51
52 /** getBrandById
53  * Recupère la marque en fonction de l'id donné en paramètre
54  * @param type $id
55  * @return type
56  */
57 function getBrandById($id, $type = PDO::FETCH_OBJ) {

```

```
58     global $tableBrands;  
59     return getFieldById($id, $tableBrands, $type);  
60 }  
61
```

```
1  <?php
2
3  /*
4  +-----+-----+
5  | CRUD KEYWORDS |
6  +-----+-----+
7  | Auteur :      | SEEMULLER Julien |
8  | Classe :      | I.IN-P4B |
9  | Date :        | 08.05.2015 |
10 | Version :      | 1.0 |
11 | Description :  | Script contenant les fonctions |
12 |                | en relation l'identification et le |
13 |                | crud keywords(CREATE, READ, UPDATE, DELETE) |
14 +-----+-----+
15 */
16
17 require_once 'basics_bdd.php';
18 $tableKeywords = 'keywords';
19
20 /** getAllKeywords
21  * Renvoie le contenu de la table "keywords"
22  * @global string $tableKeywords
23  * @return type
24  */
25 function getAllKeywords(){
26     global $tableKeywords;
27
28     return getAllFields($tableKeywords);
29 }
30
31 /** getAllKeywordsSorted
32  * Renvoie le contenu de la table "keywords" trié par ordre alphabétique
33  * @global string $tableKeywords
34  * @return type
35  */
36 function getAllKeywordsSorted(){
37     global $tableKeywords;
38     $condition = 'ORDER BY name';
39
40     return getAllFieldsCondition($tableKeywords, $condition);
41 }
```

```

1  <?php
2  /*
3  +-----+-----+
4  |  CRUD MEDIAS  |
5  +-----+-----+
6  |  Auteur :      | SEEMULLER Julien |
7  |  Classe :      | I.IN-P4B        |
8  |  Date :        | 08.05.2015      |
9  |  Version :     | 1.0              |
10 |  Description :  | Script contenant les fonctions |
11 |                | en relation l'identification et le |
12 |                | crud medias(CREATE, READ, UPDATE, DELETE) |
13 +-----+-----+
14  */
15
16 require_once 'basics_bdd.php';
17 $tableMedias = 'medias';
18
19
20 /** getMedias
21  * Recupère la liste de tous les medias
22  * @return type
23  */
24 function getMedias() {
25     global $tableMedias;
26     $dbc = connection();
27     $req = "SELECT * FROM $tableMedias";
28     $requPrep = $dbc->prepare($req); // on prépare notre requête
29     $requPrep->execute();
30     $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
31     $requPrep->closeCursor();
32
33     return $data;
34 }
35
36 /** getMediaById
37  * Recupère le media en fonction de l'id donné en paramètre
38  * @param type $id
39  * @return type
40  */
41 function getMediaById($id) {
42     global $tableMedias;
43     return getFieldById($id, $tableMedias);
44 }
45
46 /** addMedia
47  * On ajoute un média dans la table medias à l'aide d'une source
48  * et du champ isImage passé en paramètre
49  * @global string $tableMedias
50  * @param type $src
51  * @param type $isImage
52  * @return type
53  */
54 function addMedia($src, $isImage){
55     global $tableMedias;
56
57     $dbc = connection();

```

```
58     $dbc->quote($tableMedias);
59     $req = "INSERT INTO $tableMedias (src, isImage) VALUES (:source, :isImage)";
60
61     $requPrep = $dbc->prepare($req); // on prépare notre requête
62     $requPrep->bindParam(':source', $src, PDO::PARAM_STR);
63     $requPrep->bindParam(':isImage', $isImage, PDO::PARAM_BOOL);
64     $requPrep->execute();
65     $requPrep->closeCursor();
66     return $dbc->lastInsertId();
67
68 }
```

```
1  <?php
2
3  /*
4  +-----+-----+
5  |  CRUD PRODUITS  |
6  +-----+-----+
7  |  Auteur :       | SEEMULLER Julien |
8  |  Classe :       | I.IN-P4B         |
9  |  Date :         | 08.05.2015       |
10 |  Version :      | 1.0               |
11 |  Description :   | Script contenant les fonctions |
12 |                 | en relation l'identification et le |
13 |                 | crud produits(CREATE, READ, UPDATE, DELETE) |
14 +-----+-----+
15 */
16
17 require_once 'basics_bdd.php';
18
19 $tableProducts = 'products';
20 $recommendedC = "WHERE is_frontpage=true";
21 $notRecommendedC = "WHERE is_frontpage=false";
22
23 /** countProducts
24  * Retourne le nombre de produits
25  * @global string $tableProducts
26  * @return integer
27  */
28 function countProducts() {
29     global $tableProducts;
30
31     return countFields($tableProducts);
32 }
33
34 /** countRecommendedProducts
35  * Retourne le nombre produits qui sont sur la première page
36  * @global string $tableProducts
37  * @global string $recommendedC
38  * @return integer
39  */
40 function countRecommendedProducts() {
41     global $tableProducts;
42     global $recommendedC;
43
44     return countFieldsCondition($tableProducts, $recommendedC);
45 }
46
47 /** getProductById
48  * Retourne un produit selon un id passé en parametre
49  * @global string $tableProducts
50  * @param type $id
51  * @return PDO::FETCH_OBJ
52  */
53 function getProductById($id) {
54     global $tableProducts;
55     return getFieldById($id, $tableProducts);
56 }
57
```

```
58  /** getProductByIdCondition
59  * Retourne un produit selon un id et une condition passé en parametre
60  * @global string $tableProducts
61  * @param type $id
62  * @return PDO::FETCH_OBJ
63  */
64  function getProductByIdCondition($id, $condition) {
65      global $tableProducts;
66      return getFieldByIdCondition($id, $tableProducts, $condition);
67  }
68
69  /** addProduct()
70  * Ajoute un produit à la base de données
71  * @global string $tableProducts
72  * @param type $title
73  * @param type $shortDesc
74  * @param type $longDesc
75  * @param type $isFrontpage
76  * @param type $startDate
77  * @param type $endDate
78  * @param type $idBrand
79  * @return type
80  */
81  function addProduct($title, $shortDesc, $longDesc, $isFrontpage, $startDate, $endDate
82  , $idBrand) {
83      global $tableProducts;
84      $viewCount = 0;
85
86      $dbc = connection();
87      $req = "INSERT INTO $tableProducts (title, short_desc, long_desc, is_frontpage,
88          availability_date, expiration_date, view_count, id_brand) "
89          . "VALUES (:title, :shortDesc, :longDesc, :isFrontpage, :startDate,
90              :endDate, :viewCount, :idBrand)";
91
92      $requPrep = $dbc->prepare($req); // on prépare notre requête
93      $requPrep->bindParam(':title', $title, PDO::PARAM_STR);
94      $requPrep->bindParam(':shortDesc', $shortDesc, PDO::PARAM_STR);
95      $requPrep->bindParam(':longDesc', $longDesc, PDO::PARAM_STR);
96      $requPrep->bindParam(':isFrontpage', $isFrontpage, PDO::PARAM_BOOL);
97      $requPrep->bindParam(':startDate', $startDate, PDO::PARAM_STR);
98      $requPrep->bindParam(':endDate', $endDate, PDO::PARAM_STR);
99      $requPrep->bindParam(':viewCount', $viewCount, PDO::PARAM_INT);
100     $requPrep->bindParam(':idBrand', $idBrand, PDO::PARAM_INT);
101     $requPrep->execute();
102     $requPrep->closeCursor();
103     return $dbc->lastInsertId();
104 }
105
106 /** updateProduct
107 * On met a jour un produit grâce a son id.
108 * Il est nécessaire de remplir tous les paramètres de la fonction.
109 * @global string $tableProducts
110 * @param type $id
111 * @param type $title
112 * @param type $shortDesc
113 * @param type $longDesc
114 * @param type $isFrontpage
```



```
112     * @param type $startDate
113     * @param type $endDate
114     * @param type $viewCount
115     * @param type $idBrand
116     * @return type
117     */
118     function updateProduct($id, $title, $shortDesc, $longDesc, $isFrontpage, $startDate,
119     $endDate, $viewCount, $idBrand) {
120         global $tableProducts;
121         $dbc = connection();
122         $dbc->quote($tableProducts);
123         $req = "UPDATE $tableProducts SET title=:title, short_desc=:shortDesc,
124         long_desc=:longDesc, is_frontpage=:isFrontpage, availability_date=:startDate,
125         expiration_date=:endDate, view_count=:viewCount, id_brand=:idBrand WHERE id=$id";
126
127         $requPrep = $dbc->prepare($req); // on prépare notre requête
128         $requPrep->bindParam(':title', $title, PDO::PARAM_STR);
129         $requPrep->bindParam(':shortDesc', $shortDesc, PDO::PARAM_STR);
130         $requPrep->bindParam(':longDesc', $longDesc, PDO::PARAM_STR);
131         $requPrep->bindParam(':isFrontpage', $isFrontpage, PDO::PARAM_BOOL);
132         $requPrep->bindParam(':startDate', $startDate, PDO::PARAM_STR);
133         $requPrep->bindParam(':endDate', $endDate, PDO::PARAM_STR);
134         $requPrep->bindParam(':viewCount', $viewCount, PDO::PARAM_INT);
135         $requPrep->bindParam(':idBrand', $idBrand, PDO::PARAM_INT);
136         $requPrep->execute();
137         $requPrep->closeCursor();
138         return $id;
139     }
140
141     /** deleteProductById
142     * Supprime un produit dans la base de donnée selon un id passé en parametre
143     * @global string $tableProducts
144     * @param type $id
145     */
146     function deleteProductById($id) {
147         global $tableProducts;
148         //On supprime les medias lié au produit
149         deleteProductMediasById($id);
150         //On supprime la relation avec les medias
151         deleteProductMediaRelation($id);
152         //On supprime la relation avec les keywords
153         deleteProductKeywordRelation($id);
154         //On supprime le produit
155         deleteFieldById($id, $tableProducts);
156     }
157
158     /** deleteProductMediaRelation
159     * On supprime la relation avec les medias grace à l'id produit
160     * @param type $id
161     */
162     function deleteProductMediaRelation($id) {
163         $table = 'products_has_medias';
164         $condition = "WHERE id_products = $id";
165         deleteFieldByIdCondition($id, $table, $condition);
166     }
167
168     /** deleteProductMediaRelation
```

```
166      * On supprime la relation avec les medias grace à l'id produit
167      * @param type $id
168      */
169      function deleteProductMediaRelationById($idProduct, $idMedia) {
170          $table = 'products_has_medias';
171          $condition = "WHERE id_products = $idProduct AND id_medias = $idMedia";
172          deleteFieldByIdCondition($id, $table, $condition);
173      }
174
175      /** deleteProductKeywordRelation
176      * On supprime la relation avec les keywords grace à l'id produit
177      * @param type $id
178      */
179      function deleteProductKeywordRelation($id) {
180          $table = 'products_has_keywords';
181          $condition = "WHERE id_products = $id";
182          deleteFieldByIdCondition($id, $table, $condition);
183      }
184
185      /** deleteProductMediaById
186      * On supprime un medias du produit grace a son id
187      * @param type $id
188      */
189      function deleteProductMediaById($idProduct, $idMedia) {
190          $dbc = connection();
191          $table = 'medias';
192          $dbc->quote($table);
193
194          $req = "DELETE FROM $table "
195              . "WHERE id IN ("
196              . "SELECT pm.id_medias "
197              . "FROM products_has_medias "
198              . "AS pm "
199              . "WHERE pm.id_medias = $idMedia AND pm.id_products = $idProduct)";
200
201          $requPrep = $dbc->prepare($req); // on prépare notre requête
202          $requPrep->execute();
203          $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
204          $requPrep->closeCursor();
205
206          deleteProductMediaRelationById($idProduct, $idMedia);
207      }
208
209      /** deleteProductMediasById
210      * On supprime les medias du produit grace a son id
211      * @param type $id
212      */
213      function deleteProductMediasById($id) {
214          //On supprime physiquement les medias avant de les retirer de la base
215          $medias = getProductMediasById($id);
216          foreach ($medias as $media) {
217              unlink($media->mediaSource);
218          }
219
220          $dbc = connection();
221          $table = 'medias';
222          $dbc->quote($table);
```

```
223
224     $req = "DELETE FROM $table "
225         . "WHERE id IN ("
226         . "SELECT pm.id_medias "
227         . "FROM products_has_medias "
228         . "AS pm "
229         . "WHERE pm.id_products = $id)";
230
231     $requPrep = $dbc->prepare($req); // on prépare notre requête
232     $requPrep->execute();
233     $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
234     $requPrep->closeCursor();
235 }
236
237 /** getRecommendedProducts
238  * Renvoie la liste des produits recommandés par les administrateurs
239  * @global string $tableProducts
240  * @global string $recommendedC
241  * @return type
242  */
243 function getRecommendedProducts() {
244     global $tableProducts;
245
246     $dbc = connection();
247     $dbc->quote($tableProducts);
248
249     $req = 'SELECT DISTINCT p.id AS idProduct, p.title as productTitle,
250 p.short_desc, MIN(m.src) AS mediaSource, '
251         . ' m.isImage, p.view_count, p.is_frontpage '
252         . 'FROM products AS p '
253         . 'INNER JOIN products_has_medias AS pm ON p.id = pm.id_products '
254         . 'INNER JOIN medias AS m ON pm.id_medias = m.id '
255         . 'WHERE m.isImage = 1 '
256         . 'AND is_frontpage=true '
257         . 'AND NOW() > availability_date '
258         . 'AND NOW() < expiration_date '
259         . 'GROUP BY p.title '
260         . 'ORDER BY p.title ASC ';
261
262     $requPrep = $dbc->prepare($req); // on prépare notre requête
263     $requPrep->execute();
264     $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
265     $requPrep->closeCursor();
266
267     return $data;
268 }
269
270 /** getMostViewedProducts
271  * Renvoie la liste des produits les plus vus
272  * @global string $tableProducts
273  * @param int $nbProductsShown
274  * @return type
275  */
276 function getMostViewedProducts($nbProductsShown) {
277     global $tableProducts;
278
279     $dbc = connection();
```

```
279     $dbc->quote($tableProducts);
280
281     //On récupère les produits les plus vus si ils n'ont pas expiré
282     //On récupère aussi les liens vers leurs images et on les organise par leur
    nombre de vues
283     $req = 'SELECT DISTINCT p.id AS idProduct, p.title as productTitle,
    p.short_desc, MIN(m.src) AS mediaSource, '
284         . ' m.isImage, p.view_count, p.is_frontpage '
285         . 'FROM products AS p '
286         . 'INNER JOIN products_has_medias AS pm ON p.id = pm.id_products '
287         . 'INNER JOIN medias AS m ON pm.id_medias = m.id '
288         . 'WHERE m.isImage = 1 '
289         . 'AND NOW() > availability_date '
290         . 'AND NOW() < expiration_date '
291         . 'GROUP BY p.title '
292         . 'ORDER BY view_count DESC '
293         . 'LIMIT ' . $nbProductsShown;
294
295     $requPrep = $dbc->prepare($req); // on prépare notre requête
296     $requPrep->execute();
297     $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
298     $requPrep->closeCursor();
299
300     return $data;
301 }
302
303 /** getProductDetails
304  * Renvoie la liste des produits et leur marques associées
305  * @global string $tableProducts
306  * @return type
307  */
308 function getProductDetails() {
309     global $tableProducts;
310
311     $dbc = connection();
312     $dbc->quote($tableProducts);
313     //On récupère les produits les plus vus si ils n'ont pas expiré
314     //On récupère aussi les liens vers leurs images et on les organise par leur nom
315     $req = "SELECT p.id AS idProduit, p.title, p.short_desc, p.long_desc,
    p.is_frontpage,"
316         . " p.availability_date, p.expiration_date, p.view_count, p.id_brand,
    b.id AS idBrand,"
317         . " b.name AS brandName, (NOW() < availability_date OR NOW() >
    expiration_date) AS isExpired"
318         . " FROM $tableProducts AS p"
319         . " INNER JOIN brands as b ON p.id_brand = b.id";
320
321     $requPrep = $dbc->prepare($req); // on prépare notre requête
322     $requPrep->execute();
323     $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
324     $requPrep->closeCursor();
325
326     return $data;
327 }
328
329 /** getProductDetailsById
330  * Renvoie un produit par son id et les informations concernant sa marque
```

```
331     * @global string $tableProducts
332     * @param type $id
333     * @return type
334     */
335     function getProductDetailsById($id) {
336         global $tableProducts;
337
338         $dbc = connection();
339         $dbc->quote($tableProducts);
340         $req = "SELECT p.id AS idProduit, p.title, p.short_desc, p.long_desc,
341             p.is_frontpage,"
342             . " p.availability_date, p.expiration_date, p.view_count, p.id_brand,
343             b.id AS idBrand,"
344             . " b.name AS brandName, (NOW() < availability_date OR NOW() >
345             expiration_date) AS isExpired"
346             . " FROM $tableProducts AS p"
347             . " INNER JOIN brands as b ON p.id_brand = b.id"
348             . " WHERE p.id = $id";
349
350         $requPrep = $dbc->prepare($req); // on prépare notre requête
351         $requPrep->execute();
352         $data = $requPrep->fetch(PDO::FETCH_OBJ);
353         $requPrep->closeCursor();
354
355         return $data;
356     }
357
358     /** getProductKeywordsById
359     * On récupère les mots-clefs du produit grâce à l'id du produit
360     * @global string $tableProducts
361     * @param type $id
362     * @param type $type
363     * @return type
364     */
365     function getProductKeywordsById($id, $type = PDO::FETCH_OBJ) {
366         global $tableProducts;
367         $dbc = connection();
368         $dbc->quote($tableProducts);
369         $req = 'SELECT p.title, k.id AS idKeyword, k.name AS keywordName '
370             . 'FROM products AS p '
371             . 'INNER JOIN products_has_keywords AS pk ON p.id = pk.id_products '
372             . 'INNER JOIN keywords AS k ON pk.id_keywords = k.id '
373             . 'WHERE p.id = ' . $id;
374
375         $requPrep = $dbc->prepare($req); // on prépare notre requête
376         $requPrep->execute();
377         $data = $requPrep->fetchAll($type);
378         $requPrep->closeCursor();
379
380         return $data;
381     }
382
383     /** getProductMedias
384     * Renvoie la liste de tous les produits avec leurs médias associés
385     * @global string $tableProducts
386     * @return type
```

```
385     */
386     function getProductMedias() {
387         global $tableProducts;
388
389         $dbc = connection();
390         $dbc->quote($tableProducts);
391         $req = "SELECT p.id AS idProduit, p.title as productTitle, m.id as idMedia,
392             m.src AS mediaSource, m.isImage "
393             . "FROM $tableProducts AS p "
394             . "INNER JOIN products_has_medias AS pm ON p.id = pm.id_products "
395             . "INNER JOIN medias AS m ON pm.id_medias = m.id";
396
397         $requPrep = $dbc->prepare($req); // on prépare notre requête
398         $requPrep->execute();
399         $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
400         $requPrep->closeCursor();
401
402         return $data;
403     }
404
405     /** getProductMediasById
406     * Renvoie un produit et ses médias associés en dépendant de son ID
407     * @global string $tableProducts
408     * @param type $id
409     * @return type
410     */
411     function getProductMediasById($id) {
412         global $tableProducts;
413
414         $dbc = connection();
415         $dbc->quote($tableProducts);
416         $req = "SELECT p.id AS idProduit, p.title as productTitle, m.id as idMedia,
417             m.src AS mediaSource, m.isImage "
418             . "FROM $tableProducts AS p "
419             . "INNER JOIN products_has_medias AS pm ON p.id = pm.id_products "
420             . "INNER JOIN medias AS m ON pm.id_medias = m.id "
421             . "WHERE p.id = $id";
422
423         $requPrep = $dbc->prepare($req); // on prépare notre requête
424         $requPrep->execute();
425         $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
426         $requPrep->closeCursor();
427
428         return $data;
429     }
430
431     /** addProductMediaRelation
432     * Ajoute une relation entre un produit et un media
433     * @param type $idProduct
434     * @param type $idMedia
435     * @return type
436     */
437     function addProductMediaRelation($idProduct, $idMedia) {
438         $table = "products_has_medias";
439
440         $dbc = connection();
441         $dbc->quote($table);
```

```
440     $req = "INSERT INTO $table (id_products, id_medias) VALUES (:idProduct,
441         :idMedia)";
442
443     $requPrep = $dbc->prepare($req); // on prépare notre requête
444     $requPrep->bindParam(':idProduct', $idProduct, PDO::PARAM_STR);
445     $requPrep->bindParam(':idMedia', $idMedia, PDO::PARAM_BOOL);
446     $requPrep->execute();
447     $requPrep->closeCursor();
448     return $dbc->lastInsertId();
449 }
450
451 /** addProductKeywordRelation
452  * Ajoute une relation entre un produit et un mot-clef
453  * @param type $idProduct
454  * @param type $idKeyword
455  * @return type
456  */
457 function addProductKeywordRelation($idProduct, $idKeyword) {
458     $table = "products_has_keywords";
459
460     $dbc = connection();
461     $dbc->quote($table);
462     $req = "INSERT INTO $table (id_products, id_keywords) VALUES (:idProduct,
463         :idKeyword)";
464
465     $requPrep = $dbc->prepare($req); // on prépare notre requête
466     $requPrep->bindParam(':idProduct', $idProduct, PDO::PARAM_STR);
467     $requPrep->bindParam(':idKeyword', $idKeyword, PDO::PARAM_BOOL);
468     $requPrep->execute();
469     $requPrep->closeCursor();
470     return $dbc->lastInsertId();
471 }
472
473 /** addViewById
474  * On ajoute une vue au produit possédant un id identique à celui passé en paramètre
475  * @global string $tableProducts
476  * @param type $id
477  * @return type
478  */
479 function addViewById($id) {
480     global $tableProducts;
481
482     $dbc = connection();
483     $dbc->quote($tableProducts);
484     $dbc->quote($id);
485     $req = "UPDATE $tableProducts SET view_count = view_count+1 WHERE id = $id";
486     $requPrep = $dbc->prepare($req); // on prépare notre requête
487     $requPrep->execute();
488     $requPrep->closeCursor();
489     return $dbc->lastInsertId();
490 }
491
492 /** searchForProduct
493  * Renvoie la liste des produits issu d'une recherche multicriteres
494  * passée en paramètre
495  * @global string $tableProducts
496  * @param type $query
```

```

495     * @return type
496     */
497     function searchForProduct($query) {
498         global $tableProducts;
499
500         $dbc = connection();
501         $dbc->quote($tableProducts);
502         $req = 'SELECT DISTINCT p.id AS idProduct, p.title as productTitle,
503             p.short_desc, MIN(m.src) AS mediaSource, '
504             . ' m.isImage, p.view_count, p.is_frontpage, b.name AS brandName '
505             . 'FROM ' . $tableProducts . ' AS p '
506             . 'INNER JOIN products_has_medias AS pm ON p.id = pm.id_products '
507             . 'INNER JOIN medias AS m ON pm.id_medias = m.id '
508             . 'INNER JOIN brands AS b ON p.id_brand = b.id '
509             . 'WHERE Concat(p.title, \'\'', p.short_desc, \'\'', p.long_desc, \'\'',
510             b.name) LIKE "%" . $query . "%" '
511             . 'AND m.isImage = 1 GROUP BY p.title ORDER BY p.title DESC';
512
513         $requPrep = $dbc->prepare($req); // on prépare notre requête
514         $requPrep->execute();
515         $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
516         $requPrep->closeCursor();
517
518         return $data;
519     }
520
521     /** searchForProductWithKeywords
522     * Renvoie la liste des produits possédant le mot-clé passé en paramètre
523     * @global string $tableProducts
524     * @param type $query
525     * @return type
526     */
527     function searchForProductWithKeywords($query) {
528         global $tableProducts;
529         $dbc = connection();
530         $dbc->quote($tableProducts);
531         $req = 'SELECT DISTINCT p.id AS idProduct, p.title as productTitle, '
532             . 'p.short_desc, p.view_count, p.is_frontpage, b.name AS brandName, '
533             . 'k.name AS keywordName '
534             . 'FROM ' . $tableProducts . ' AS p '
535             . 'INNER JOIN brands AS b ON p.id_brand = b.id '
536             . 'INNER JOIN products_has_keywords AS pk ON p.id = pk.id_products '
537             . 'INNER JOIN keywords AS k ON pk.id_keywords = k.id '
538             . 'WHERE k.name LIKE "%" . $query . "%" '
539             . 'GROUP BY p.title '
540             . 'ORDER BY p.title DESC';
541
542         $requPrep = $dbc->prepare($req); // on prépare notre requête
543         $requPrep->execute();
544         $data = $requPrep->fetchAll(PDO::FETCH_OBJ);
545         $requPrep->closeCursor();
546
547         return $data;
548     }
549
550     /*****
551     * REQUETES POUR RECHERCHE PRODUIT */

```



```
550  /*****/
551
552  /* SEARCH FOR PRODUCT
553  SELECT p.title AS productName, p.short_desc, p.long_desc, b.name AS brandName
554  FROM products AS p
555  INNER JOIN brands AS b ON p.id_brand = b.id
556  WHERE Concat(p.title, '', p.short_desc, '', p.long_desc, '', b.name)
557  LIKE "%Razer%"
558  */
559
560  /* SEARCH FOR PRODUCT VIA KEYWORD
561  SELECT p.title AS productName, p.short_desc, p.long_desc, k.name
562  FROM products AS p
563  INNER JOIN products_has_keywords AS pk ON p.id = pk.id_products
564  INNER JOIN keywords AS k ON pk.id_keywords = k.id
565  WHERE k.name LIKE "%Carte%"
566  */
```

```

1  <?php
2  /*
3  +-----+-----+
4  |  CRUD UTILISATEUR  |
5  +-----+-----+
6  |  Auteur :          | SEEMULLER Julien |
7  |  Classe :          | I.IN-P4B       |
8  |  Date :           | 08.05.2015      |
9  |  Version :         | 1.0             |
10 |  Description :      | Script contenant les fonctions |
11 |                    | en relation l'identification et le |
12 |                    | crud user (CREATE, READ, UPDATE, DELETE) |
13 +-----+-----+
14 */
15
16 require_once 'basics_bdd.php';
17
18 /** getUserByPseudo
19  * Recupère les informations de l'utilisateur grâce à son pseudo
20  * @param type $pseudo
21  * @return type
22  */
23 function getUserByPseudo($pseudo) {
24     $dbc = connection();
25     $req = "SELECT * FROM users WHERE username=:pseudo";
26     // preparation de la requete
27     $requPrep = $dbc->prepare($req); // on prépare notre requête
28     $requPrep->bindParam(':pseudo', $pseudo, PDO::PARAM_STR);
29     $requPrep->execute();
30     $data= $requPrep->fetch(PDO::FETCH_OBJ);
31     $requPrep->closeCursor();
32     return $data;
33 }
34
35 /** userConnect
36  * Connecte l'utilisateur et initialise la session
37  * @param type $username
38  * @param type $password
39  * @return boolean
40  */
41 function userConnect($username, $password) {
42     $connect = false;
43     $_SESSION['username'] = $username;
44     $user = getUserbyPseudo($username);
45     if ($user != NULL && $user->password === hashPerso($password)) {
46         $_SESSION['id'] = $user->id;
47         $_SESSION['username'] = $user->username;
48         $_SESSION['admin'] = $user->is_admin;
49         $connect = TRUE;
50     }
51     return $connect;
52 }

```

```
1  <?php
2
3  /*
4  +-----+-----+
5  | SPECIFIC FUNCTIONS |
6  +-----+-----+
7  | Auteur :          | SEEMULLER Julien |
8  | Classe :          | I.IN-P4B |
9  | Date :            | 08.05.2015 |
10 | Version :          | 1.0 |
11 | Description :      | Script regroupant les fonctions spécifiques |
12 |                    | au site web et la base de données |
13 +-----+-----+
14 */
15
16 session_start();
17 require_once 'crud_User.php';
18
19 /** isAdmin
20  * Verifie si l'utilisateur est un administrateur
21  * @return type
22  */
23 function isAdmin() {
24     if (isset($_SESSION['admin'])) {
25         if ($_SESSION['admin] == 1) {
26             $result = true;
27         }
28         else {
29             $result = false;
30         }
31     }
32     else {
33         $result = false;
34     }
35     return ($result);
36 }
37
38 /** isConnected
39  * Verifie si l'utilisateur est connecté
40  * @return type
41  */
42 function isConnected() {
43     return (isset($_SESSION['id']));
44 }
45
46 /* * HashPerso
47  * Hash le mot de passe
48  * @param string $value
49  * @return string
50  */
51
52 function hashPerso($password) {
53     return sha1(md5($password));
54 }
55
56 /** checkImageType
57  * Vérifie le type de l'image
```

```
58  * @param type $fileType
59  * @return type
60  */
61  function checkImageType($fileType) {
62      return in_array($fileType, unserialize(ALLOWED_IMAGE_TYPES));
63  }
64
65  function isImage($string) {
66      $result = false;
67
68      if ((strpos($string, '.jpg') !== false) ||
69          (strpos($string, '.jpeg') !== false) ||
70          (strpos($string, '.png') !== false) ||
71          (strpos($string, '.gif') !== false)) {
72          $result = true;
73      }
74
75      return "$result";
76  }
77
78  /** debug
79   * Debug de la variable donnée en paramètre
80   * @param type $sObj
81   */
82  function debug($sObj = NULL) {
83      echo '<pre>';
84      if (is_null($sObj)) {
85          echo '|Object is NULL|' . "\n";
86      } elseif (is_array($sObj) || is_object($sObj)) {
87          var_dump($sObj);
88      } else {
89          echo '| ' . $sObj . ' |' . "\n";
90      }
91      echo '</pre>';
92  }
93
```

```

1  <?php
2  /*
3  +-----+-----+
4  |  STRUCTURE PHP  |
5  +-----+-----+
6  |  Auteur :      | SEEMULLER Julien |
7  |  Classe :      | I.IN-P4B        |
8  |  Date :        | 08.05.2015      |
9  |  Version :     | 1.0              |
10 |  Description :  | Script permettant de définir quel élément du site |
11 |                | web a utiliser dépendamment de l'utilisateur |
12 +-----+-----+
13 */
14 require_once 'specific_funtions.php';
15 require_once 'crud_Produits.php';
16 require_once 'crud_Keywords.php';
17 require_once 'crud_Medias.php';
18
19 /** getHeader
20  * Affiche différents headers en fonction de la personne qui est connectée
21  */
22 function getHeader() {
23     include 'structure/header.php';
24 }
25
26 /** structRecommendedProducts
27  * Renvoie un string contenant le code html pour afficher les produits reccomendés
28  * @return string
29  */
30 function structRecommendedProducts() {
31     $products = getRecommendedProducts();
32     $str = '';
33
34     //On crée un panel pour chaque produits de la page
35     foreach ($products as $product) {
36         $str.= '<div class="col-xs-12 col-sm-6 col-md-4 col-lg-3">
37             <div class="panel panel-success">
38                 <div class="panel-heading">
39                     <a href="detail.php?id=' . $product->idProduct . '"><h3
40                         class="panel-title" style="text-align: center">' .
41                         $product->productTitle . '</h3></a>
42                     </div>
43                     <div class="panel-body">
44                         <div class="product-container thumbnail">
45                             <a href="detail.php?id=' . $product->idProduct .
46                                 '"></a>
49                             </div>
50                             <hr/>
51                             <div class="panel-footer" style="float: left;">
52                                 <p>
53                                     ' . $product->short_desc . '
54                                 </p>
55                             </div>
56                         </div>
57                     </div>
58                 </div>
59             </div>
60         </div>

```

```

52         </div>
53     </div>' ;
54 }
55
56     return $str;
57 }
58
59 /** structMostViewedProducts
60  * Renvoie un string contenant le code html pour afficher les produits les plus vus
61  * @return string
62  */
63 function structMostViewedProducts() {
64     $products = getMostViewedProducts(NUMBER_OF_TOP_PRODUCTS);
65     $str = '';
66
67     //On crée un panel pour chaque produits de la page
68     foreach ($products as $product) {
69         $str.= '<div class="col-xs-12 col-sm-6 col-md-4 col-lg-3">
70             <div class="panel panel-success">
71                 <div class="panel-heading">
72                     <a href="detail.php?id=' . $product->idProduct . '"><h3
73                         class="panel-title" style="text-align: center">' .
74                         $product->productTitle . '</h3></a>
75
76                     </div>
77                     <div class="panel-body">
78                         <div class="product-container thumbnail">
79                             <a href="detail.php?id=' . $product->idProduct .
80                                 '"></a>
83
84                             </div>
85                             <hr/>
86                             <div class="panel-footer" style="float: left;">
87                                 <p>
88                                     ' . $product->short_desc . '
89                                 </p>
90                             </div>
91                         </div>
92                     </div>' ;
93
94                 return $str;
95             }
96
97             /** structKeywordsList
98             * Renvoie un string contenant le code html pour afficher la liste des catégories
99             * @return string
100             */
101             function structKeywordsList() {
102                 $keywords = getAllKeywordsSorted();
103                 $str = '';
104
105                 //On affiche un lien pour chaque keywords de la base
106                 foreach ($keywords as $keyword) {
107                     $str.= '<a href="search.php?queryKW=' . $keyword->name . '"

```

```

        class="list-group-item">
103         ' . strtoupper($keyword->name) . '
104         </a>';
105     }
106
107     return $str;
108 }
109
110 /** structSearchedProducts
111  * Structure la page de recherches multicritères
112  * @param type $query
113  * @return string
114  */
115 function structSearchedProducts($query) {
116     $products = searchForProduct($query);
117     $str = '';
118     $query = str_replace(' ', '%', $query);
119     foreach ($products as $product) {
120         $str .= '<li class="media">
121             <div class="media-left">
122                 
124             </div>
125             <div class="media-body">
126                 <a href="detail.php?id=' . $product->idProduct . '"><h4
127                 class="media-heading"><b>' . $product->brandName . '</b> - '
128                 . $product->productTitle . '</h4></a>
129                 <i>' . $product->short_desc . '</i>
130                 <p></p>
131             </div>
132             </li>
133             <hr/>';
134     }
135
136     return $str;
137 }
138
139 /**
140  * Structure la page de recherche par mot-clefs
141  * @param type $query
142  * @return string
143  */
144 function structSearchedProductsKeywords($query) {
145     $products = searchForProductWithKeywords($query);
146     $str = '';
147
148     foreach ($products as $product) {
149         //On récupère la première image du produit
150         $medias = getProductMediasById($product->idProduct);
151         $imgProduct = '';
152         foreach ($medias as $media) {
153             if ($media->isImage) {
154                 $imgProduct[] = $media->mediaSource;
155             }
156         }
157
158         //On crée une liste html contenant les medias et les informations du produit

```

```

156         $str .= '<li class="media">
157             <div class="media-left">
158                 
160             </div>
161             <div class="media-body">
162                 <a href="detail.php?id=' . $product->idProduct . '"><h4
163                 class="media-heading"><b>' . $product->brandName . '</b> - '
164                 . $product->productTitle . '</h4></a>
165                 <i>' . $product->short_desc . '</i>
166                 <p></p>
167             </div>
168         </li>
169         <hr/>';
170     }
171
172     return $str;
173 }
174
175 /** structDetailProduct
176  * Structure le détail d'un produit grace à son id
177  * @param type $id
178  * @return string
179  */
180 function structDetailProduct($id) {
181     $product = getProductDetailsById($id);
182     $medias = getProductMediasById($id);
183
184     $carouselIndicators = '';
185     $carouselControls = '';
186     $imagesProduct = '';
187     $otherMedia = '';
188     $i = 0;
189     $j = 0;
190
191     foreach ($medias as $media) {
192         if ($media->isImage) {
193             $i++;
194             $arrayImage[$i] = $media->mediaSource;
195         } else {
196             $j++;
197             $arrayOthers[$j] = $media->mediaSource;
198         }
199     }
200
201     //Pour chaque image de produits, on ajoute un indicateur pour le carousel et la
202     //source de l'image
203     //pour la première fois, on rend le premier indicateur et la première image
204     "actif"
205     if (isset($arrayImage)) {
206         $i = 0;
207         foreach ($arrayImage as $image) {
208             if ($i == 0) {
209                 $carouselIndicators .= '<li data-target="#carousel-example-generic"
210                 data-slide-to="' . $i . '" class="active"></li>';
211                 $imagesProduct .= '<div class="item thumbnail active"></div>';

```



```

206         } else {
207             $carouselIndicators .= '<li data-target="#carousel-example-generic"
                data-slide-to="' . $i . '" class=""></li>';
208             $imagesProduct .= '<div class="item thumbnail"></div>';
209         }
210         $i++;
211     }
212
213     //On affiche les controls si il y a plus d'une image
214     if (count($arrayImage) > 1) {
215         $carouselControls = '<a class="left carousel-control"
                href="#carousel-example-generic" role="button" data-slide="prev">
216             <span class="glyphicon glyphicon-chevron-left"
                aria-hidden="true"></span>
217             <span class="sr-only">Previous</span>
218             </a>
219             <a class="right carousel-control"
                href="#carousel-example-generic" role="button"
                data-slide="next">
220                 <span class="glyphicon glyphicon-chevron-right"
                aria-hidden="true"></span>
221                 <span class="sr-only">Next</span>
222             </a>';
223     } else {
224         //Si il n'y a qu'un produit, on retire les indicateurs
225         $carouselIndicators = '';
226     }
227 }
228 //Si le produit n'as pas d'image, on affiche un placeholder
229 else {
230     $imagesProduct .= '<div class="item active"></div>';
231 }
232
233 //Pour chaque image de produits, on ajoute un media dans la liste des
téléchargements
234 if (isset($arrayOthers)) {
235     foreach ($arrayOthers as $other) {
236         //On crée le nom du fichier en soustrayant la longueur de l'arborescance
                au chemin complet
237         $filename = substr($other, strlen(OTHER_FOLDER));
238         $otherMedia .= '<li class="media">
                <div class="media-left">
239             <a href="#">
240                 
242             </a>
243             </div>
                <div class="media-body" style="margin: 10px;">
                <h4 class="media-heading" id="media-heading"><a
                href="download.php?file=' . $other . '">
                $filename . '</a><a class="anchorjs-link"
                href="#media-heading"><span
                class="anchorjs-icon"></span></a></h4>

```

```

246         </div>
247     </li>' ;
248 }
249 } else {
250     $arrayOthers = NULL;
251 }
252
253 if (is_null($arrayOthers)) {
254     $otherMedia = '<h4 class="red"><span class="glyphicon glyphicon-remove-sign"></span><i> Pas de fichiers actuellement disponible
pour ce produit</i></h4>';
255 }
256
257 //On regarde si le produit est expiré et on affiche à l'utilisateur l'information
258 if (!$product->isExpired) {
259     $expiredText = '<span class="green glyphicon glyphicon-ok-sign"></span><span
class="green"> Le produit est actuellement disponible (' . $product->
availability_date . ' - ' . $product->expiration_date . ')</span>';
260 } else {
261     $expiredText = '<span class="red glyphicon glyphicon-remove-sign"></span><span class="red"> Le produit est actuellement
indisponible (' . $product->availability_date . ' - ' . $product->
expiration_date . ')</span>';
262 }
263
264 $str = '<div class="row well">
265     <h1><span class="glyphicon glyphicon-search"></span> ' .
strtoupper($product->brandName) . ' - ' . $product->title . '</h1>
266     <hr>
267     <div class="col-xs-12 col-sm-12 col-md-12 col-lg-5">
268         <div id="carousel-example-generic" class="carousel slide"
data-ride="carousel">
269             <ol class="carousel-indicators">
270                 ' . $carouselIndicators . '
271             </ol>
272             <div class="carousel-inner" role="listbox">
273                 ' . $imagesProduct . '
274             </div>
275             ' . $carouselControls . '
276         </div>
277     </div>
278     <div class="col-xs-12 col-sm-12 col-md-12 col-lg-7">
279         <h4><i>' . $product->short_desc . '</i></h4>
280         <p>' . $product->long_desc . '</p>
281         ' . $expiredText . '
282     </div>
283 </div>
284 <div class="row well">
285     <h1><span class="glyphicon glyphicon-download"></span> Fichiers
téléchargeables</h1>
286     <hr>
287     <div class="media">
288         <ul class="media-list">
289             ' . $otherMedia . '
290         </ul>
291     </div>
292 </div>

```

```
293         <div class="row">' ;
294     return $str;
295 }
296
297 function structViewCount($id) {
298     $product = getProductById($id);
299     $str = '<span>Nombre de vues : ' . $product->view_count . ' <span
        class="glyphicon glyphicon-eye-open"></span></span>' ;
300
301     return $str;
302 }
303
```

```

1  <!--
2  +-----+-----+
3  |  HEADER  |
4  +-----+-----+
5  | Auteur : | SEEMULLER Julien |
6  | Classe : | I.IN-P4B |
7  | Date :   | 08.05.2015 |
8  | Version :| 1.0 |
9  |
10 | Description : | Fichier contenant du code HTML pour
11 |               | l'en-tête du site en fonction de
12 |               | l'utilisateur connecté
13 +-----+-----+
14 -->
15 <nav class="navbar navbar-default navbar-static-top" role="navigation">
16     <div class="container">
17         <div class="navbar-header">
18             <button type="button" class="navbar-toggle collapsed" data-toggle=
19             "collapse" data-target="#navbar" aria-controls="navbar">
20                 <span class="sr-only">Toggle navigation</span>
21                 <span class="icon-bar"></span>
22                 <span class="icon-bar"></span>
23                 <span class="icon-bar"></span>
24             </button>
25             <a class="navbar-brand" href="index.php"><span class="glyphicon
26             glyphicon-home"></span> Catal'info</a>
27         </div>
28
29         <div id="navbar" class="navbar-collapse collapse" aria-expanded="false" style
30         ="height: 1px;">
31             <form class="navbar-form navbar-left" method="get" action="search.php">
32                 <div class="input-group col-xs-12 search-bar">
33                     <input name="query" type="text" class="form-control" placeholder
34                     ="Rechercher...">
35                     <span class="input-group-btn">
36                         <button class="btn btn-success" type="submit"><span class=
37                         "glyphicon glyphicon-search"></span></button>
38                     </span>
39                 </div>
40             </form>
41
42             <?php
43             if (isConnected()) {
44                 //Si l'utilisateur est connecté et qu'il est un administrateur (ADMIN)
45                 if (isAdmin()) {
46                     ?>
47                     <form class="navbar-form navbar-right" role="form">
48                         <div class="btn-group">
49                             <button type="button" class="btn btn-default
50                             dropdown-toggle" data-toggle="dropdown" aria-expanded=
51                             "false">
52                                 <?php echo $_SESSION["username"]; ?> <span class=
53                                 "caret"></span>
54                             </button>
55                             <ul class="dropdown-menu" role="menu">
56                                 <li><a href="addProduct.php"><span class="glyphicon
57                                 glyphicon-plus-sign"></span> Ajouter un produit

```

```
49         </a></li>
50     </ul>
51 </div>
52 </form>
53 <?php
54     }
55     //Si l'utilisateur est connecté mais pas un admin (USER)
56     else {
57         ?>
58         <form class="navbar-form navbar-right" role="form">
59             <div class="btn-group">
60                 <button type="button" class="btn btn-default
61                 dropdown-toggle" data-toggle="dropdown" aria-expanded=
62                 "false">
63                     <?php echo $_SESSION["username"]; ?> <span class=
64                     "caret"></span>
65                 </button>
66                 <ul class="dropdown-menu" role="menu">
67                     <li><a href="logout.php"><span class="glyphicon
68                     glyphicon-log-out"></span> Se déconnecter</a></li>
69                 </ul>
70             </div>
71         </form>
72         <?php
73     }
74 }
75 //Si l'utilisateur n'est pas connecté et n'est pas un administrateur
76 (VISITEUR)
77 else {
78     ?>
79     <form class = "navbar-form navbar-right" role = "form">
80         <button name = "login" type = "button" class = "btn btn-success"
81         data-toggle = "modal" data-target = "#myModal">Connexion <span
82         class = "glyphicon glyphicon-log-in"></span></button>
83     </form>
84     <?php
85 }
86 ?>
87 </div>
88 </div>
89 </nav>
```

```
1  $(function() {  
2      $(document).ready(function() {  
3  
4          //Quand on clique sur le bouton pour quitter, on ferme le menu et on affiche  
          le bouton  
5          $('#slide-submenu').on('click', function() {  
6              $(this).closest('.list-group').fadeOut('slide', function() {  
7                  $('.mini-submenu').fadeIn();  
8              });  
9  
10         });  
11  
12         //Quand on clique sur le bouton, on affiche le menu  
13         $('.mini-submenu').on('click', function() {  
14             $(this).next('.list-group').toggle('slide');  
15             $('.mini-submenu').hide();  
16         });  
17     });  
18 });
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="utf-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <meta name="description" content="">
8      <meta name="author" content="">
9      <title>Catal'info</title>
10     <link href="/css/bootstrap.min.css" rel="stylesheet">
11     <link href="/css/style.css" rel="stylesheet">
12     <link href="/css/font-awesome.css" rel="stylesheet">
13     <style type="text/css"></style>
14   </head>
15   <body>
16     <!-- NAVBAR -->
17     <nav class="navbar navbar-default navbar-static-top" role="navigation">
18       <div class="container">
19         <div class="navbar-header">
20           <button type="button" class="navbar-toggle collapsed" data-toggle=
21             ="collapse" data-target="#navbar" aria-controls="navbar">
22             <span class="sr-only">Toggle navigation</span>
23             <span class="icon-bar"></span>
24             <span class="icon-bar"></span>
25             <span class="icon-bar"></span>
26           </button>
27           <a class="navbar-brand" href="index.php">Catal'info</a>
28         </div>
29         <div id="navbar" class="navbar-collapse collapse" aria-expanded=
30           "false" style="height: 1px;">
31           <form class="navbar-form navbar-left">
32             <div class="input-group col-xs-12">
33               <input type="text" class="form-control" placeholder=
34                 "Rechercher...">
35               <span class="input-group-btn">
36                 <button class="btn btn-success" type="button"><span
37                   class="glyphicon glyphicon-search"></span></button>
38               </span>
39             </div>
40           </form>
41           <form class="navbar-form navbar-right" role="form">
42             <div class="btn-group">
43               <button type="button" class="btn btn-default
44                 dropdown-toggle" data-toggle="dropdown">
45                 Admin <span class="caret"></span>
46               </button>
47               <ul class="dropdown-menu" role="menu">
48                 <li><a href="#"><span class="glyphicon
49                   glyphicon-plus-sign"></span> Ajouter un produit
50                 </a></li>
51                 <li><a href="logout.php"><span class="glyphicon
52                   glyphicon-log-out"></span> Se déconnecter</a></li>
53               </ul>
54             </div>
55           </form>
```

```
50         </div>
51     </div>
52     </nav>         <div class="left">
53         <div class="col-sm-2 col-md-2">
54             <div class="mini-submenu">
55                 <span class="icon-bar"></span>
56                 <span class="icon-bar"></span>
57                 <span class="icon-bar"></span>
58             </div>
59             <div class="list-group" style="display: none;">
60                 <h4 class="list-group-item">
61                     Recherche par catégorie
62                     <span class="pull-right" id="slide-submenu">
63                         <span class="glyphicon glyphicon-chevron-left"></span>
64                     </span>
65                 </h4>
66                 <a href="#" class="list-group-item">
67                     carte graphique
68                 </a><a href="#" class="list-group-item">
69                     carte mere
70                 </a><a href="#" class="list-group-item">
71                     ram
72                 </a><a href="#" class="list-group-item">
73                     boitier
74                 </a><a href="#" class="list-group-item">
75                     processeur
76                 </a><a href="#" class="list-group-item">
77                     clavier
78                 </a><a href="#" class="list-group-item">
79                     souris
80                 </a><a href="#" class="list-group-item">
81                     ecran
82                 </a><a href="#" class="list-group-item">
83                     carte son
84                 </a><a href="#" class="list-group-item">
85                     carte reseau
86                 </a><a href="#" class="list-group-item">
87                     disque dur
88                 </a><a href="#" class="list-group-item">
89                     lecteur cd
90                 </a><a href="#" class="list-group-item">
91                     ventilateur
92                 </a><a href="#" class="list-group-item">
93                     graveur cd
94                 </a><a href="#" class="list-group-item">
95                     cle usb
96                 </a>
97             </div>
98         </div>
99     </div>
100
101     <!-- CONTAINER -->
102     <div class="container">
103         <div class="container-fluid">
104             <!-- CONTAINER PANELS PRODUITS LES PLUS VUS-->
105             <div class="row well">
106                 <h1><span class="glyphicon glyphicon-list-alt"></span> Razer
```


Blackwidow</h1>

<hr>

<div class="col-sm-5">

<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">

<ol class="carousel-indicators">

<li data-target="#carousel-example-generic" data-slide-to="0" class="active">

<li data-target="#carousel-example-generic" data-slide-to="1" class="">

<li data-target="#carousel-example-generic" data-slide-to="2" class="">

<li data-target="#carousel-example-generic" data-slide-to="3" class="">

<div class="carousel-inner" role="listbox">

<div class="item active">

</div>

<div class="item">

</div>

<div class="item">

</div>

<div class="item">

</div>

</div>

Previous

Next

</div>

</div>

<hr/>

<div class="col-sm-5">

<h4><i>Cherry MX Blue mechanical keyboard</i></h4>

<p>The Razer BlackWidow uses a PC layout but with 5 macro keys on the left-hand side. The key layout uses 1.5u-sized modifiers on the bottom row, with 1u Windows, Fn and Menu keys, as well as a 6u Space bar. There is also a version with US-ANSI/Macintosh layout but otherwise the same

physical keys.

A unique feature of the BlackWidow is that the function keys are shifted slightly to the right, so that they are parallel to the WASD cluster.</p>

<p><i>Produit disponible du xx:xx:xx au xx:xx:xx</i></p>

</div>

</div>

<div class="row well">

<h1> Fichiers
téléchargeables</h1>

<hr>

<div class="media">

<ul class="media-list">

<li class="media">

<div class="media-left">

<img class="media-object" data-src=
"holder.js/64x64" alt="64x64" src=
"
iMS4wIiBlbmNvZGluc20iVVRGLTgiIHN0YW5kYWxvbmU9I
nllcyI/PjxzdmceG1sbnM9Imh0dHA6Ly93d3cudzMub3J
nLzIwMDAvMzZnIiB3aWR0aD0iNjQiIGhlaWdodD0iNjQiI
HZpZXh0d3c3g9IjAgMCA2NCA2NCIgcHJlc2Vydmlvc3B1Y3R
SYXRpbz0ibm9uZSI+PGRlZnMvPjxyZWN0IHdpZHRoPSI2N
CIgaGVpZ2h0PSI2NCIgcZmlsdD0iI0VFRUVFRSIvPjxnPjx
0ZXh0IHg9IjE0IiB5PSIzMmIgc3R5bGU9ImZpbGw6I0FBQ
UFBQTtmb250LXdlaWdodDpib2xkO2ZvbnQtZmFtaWx5OkF
yaWFsLCBIZWx2ZXRpY2EsIE9wZW4gU2FucywgY2Fucy1zZ
XJpZiwiZGw9ub3NwYWNlO2ZvbnQtZmFtaWx5OkFyaWFsLCBIZWx2ZXRpY2EsIE9wZW4gU2FucywgY2Fucy1zZ
uYW50LWJhc2VsaW5lOmNlbnRyYWwiPjY0eDY0PC90ZXh0P
jwvZz48L3N2Zz4=" data-holder-rendered="true"
style="width: 64px; height: 64px;">

</div>

<div class="media-body">

<h4 class="media-heading" id="media-heading">
Media heading<a class="anchorjs-link" href=
"#media-heading"><span class="anchorjs-icon"
></h4>

<p>Cras sit amet nibh libero, in gravida nulla.
Nulla vel metus scelerisque ante sollicitudin
commodo. Cras purus odio, vestibulum in
vulputate at, tempus viverra turpis.</p>

</div>

<li class="media">

<div class="media-left">

<img class="media-object" data-src=
"holder.js/64x64" alt="64x64" src=
"
iMS4wIiBlbmNvZGluc20iVVRGLTgiIHN0YW5kYWxvbmU9I
nllcyI/PjxzdmceG1sbnM9Imh0dHA6Ly93d3cudzMub3J
nLzIwMDAvMzZnIiB3aWR0aD0iNjQiIGhlaWdodD0iNjQiI
HZpZXh0d3c3g9IjAgMCA2NCA2NCIgcHJlc2Vydmlvc3B1Y3R
SYXRpbz0ibm9uZSI+PGRlZnMvPjxyZWN0IHdpZHRoPSI2N
CIgaGVpZ2h0PSI2NCIgcZmlsdD0iI0VFRUVFRSIvPjxnPjx
0ZXh0IHg9IjE0IiB5PSIzMmIgc3R5bGU9ImZpbGw6I0FBQ
UFBQTtmb250LXdlaWdodDpib2xkO2ZvbnQtZmFtaWx5OkF
yaWFsLCBIZWx2ZXRpY2EsIE9wZW4gU2FucywgY2Fucy1zZ
XJpZiwiZGw9ub3NwYWNlO2ZvbnQtZmFtaWx5OkFyaWFsLCBIZWx2ZXRpY2EsIE9wZW4gU2FucywgY2Fucy1zZ
uYW50LWJhc2VsaW5lOmNlbnRyYWwiPjY0eDY0PC90ZXh0P
jwvZz48L3N2Zz4=" data-holder-rendered="true"
style="width: 64px; height: 64px;">

```

169         CIgaGVpZ2h0PSI2NCIgZmlsbD0iI0VFRUVFRSIVPjxnPjx
170         0ZXh0IHg9IjE0IiB5PSIzMmIgc3R5bGU9ImZpbGw6I0FBQ
171         UFBQTtmb250LXdlaWdodDpib2xkO2ZvbnQtZmFtaWx5OkF
172         yaWFsLCBIZWx2ZXRpY2EsIE9wZW4gU2Fucywg2FucylzZ
173         XJpZiwiZm9ub3NwYWw0O2ZvbnQtZmFtaWx5OkFyaWFsLCBIZWx2ZXRpY2EsIE9wZW4gU2Fucywg2FucylzZ
174         uYW50LWJhc2VsaW5lOmNlbnRyYWwipjY0eDY0PC90ZXh0P
175         jwvZz48L3N2Zz4=" data-holder-rendered="true"
176         style="width: 64px; height: 64px;">
177     </a>
178 </div>
179 <div class="media-body">
180     <h4 class="media-heading" id="media-heading">
181     Media heading<a class="anchorjs-link" href=
182     "#media-heading"><span class="anchorjs-icon"
183     ></span></a></h4>
184     <p>Cras sit amet nibh libero, in gravida nulla.
185     Nulla vel metus scelerisque ante sollicitudin
186     commodo. Cras purus odio, vestibulum in
187     vulputate at, tempus viverra turpis.</p>
188 </div>
189 </li>
190 <li class="media">
191     <div class="media-left">
192         <a href="#">
193             <img class="media-object" data-src=
194             "holder.js/64x64" alt="64x64" src=
195             "
196             0ZXh0IHg9IjE0IiB5PSIzMmIgc3R5bGU9ImZpbGw6I0FBQ
197             UFBQTtmb250LXdlaWdodDpib2xkO2ZvbnQtZmFtaWx5OkF
198             yaWFsLCBIZWx2ZXRpY2EsIE9wZW4gU2Fucywg2FucylzZ
199             XJpZiwiZm9ub3NwYWw0O2ZvbnQtZmFtaWx5OkFyaWFsLCBIZWx2ZXRpY2EsIE9wZW4gU2Fucywg2FucylzZ
200             uYW50LWJhc2VsaW5lOmNlbnRyYWwipjY0eDY0PC90ZXh0P
201             jwvZz48L3N2Zz4=" data-holder-rendered="true"
202             style="width: 64px; height: 64px;">
203         </a>
204     </div>
205     <div class="media-body">
206         <h4 class="media-heading" id="media-heading">
207         Media heading<a class="anchorjs-link" href=
208         "#media-heading"><span class="anchorjs-icon"
209         ></span></a></h4>
210         <p>Cras sit amet nibh libero, in gravida nulla.
211         Nulla vel metus scelerisque ante sollicitudin
212         commodo. Cras purus odio, vestibulum in
213         vulputate at, tempus viverra turpis.</p>
214     </div>
215 </li>
216 </ul>
217 </div>
218 </div>
219 <div class="row">
```

```
191         <h3>Nombre de vues : 25 <span class="glyphicon  
192             glyphicon-eye-open"></span></h3>  
193     </div>  
194 </div>  
195  
196 <!-- Modal connexion -->  
197 <div class="modal fade" id="myModal" tabindex="-1" role="dialog"  
198     aria-labelledby="myModalLabel" aria-hidden="true">  
199     <div class="modal-dialog">  
200         <div class="modal-content">  
201             <form class="form-signin" method="post" action="">  
202                 <div class="modal-header">  
203                     <button type="button" class="close" data-dismiss="modal"  
204                         aria-label="Close"><span aria-hidden="true">x  
205                         </span></button>  
206                     <h3 class="modal-title" id="myModalLabel">Connexion</h3>  
207                 </div>  
208                 <div class="modal-body">  
209                     <label class="">Pseudo:</label><input class=  
210                         "form-control" name="username" type="text" value=""  
211                         placeholder="ex : 'Johndoe'"><br>  
212                     <label>Password :</label><input class="form-control" name  
213                         ="password" type="password"><br>  
214                 </div>  
215                 <div class="modal-footer">  
216                     <button type="button" class="btn btn-default"  
217                         data-dismiss="modal">Fermer</button>  
218                     <input name="login" class="btn btn-success" type="submit"  
219                         value="Se connecter">  
220                 </div>  
221             </form>  
222         </div>  
223     </div>  
224 </div>  
  
225 <!-- Bootstrap core JavaScript  
226 ===== -->  
227 <script src="./js/jquery.min.js"></script>  
228 <script src="./js/bootstrap.min.js"></script>  
229 <script src="js/dropdown.js"></script>  
230 </body>  
231 </html>
```

```

1  <!DOCTYPE html>
2  <html lang="en"><head>
3      <meta charset="utf-8">
4      <meta http-equiv="X-UA-Compatible" content="IE=edge">
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <meta name="description" content="">
7      <meta name="author" content="">
8      <title>Catal'info</title>
9      <link href="./css/bootstrap.min.css" rel="stylesheet">
10     <link href="./css/style.css" rel="stylesheet">
11     <link href="./css/font-awesome.css" rel="stylesheet">
12 </head>
13 <body>
14     <!-- NAVBAR -->
15     <nav class="navbar navbar-default navbar-static-top" role="navigation">
16         <div class="container">
17             <div class="navbar-header">
18                 <button type="button" class="navbar-toggle collapsed" data-toggle
19                     = "collapse" data-target="#navbar" aria-controls="navbar">
20                     <span class="sr-only">Toggle navigation</span>
21                     <span class="icon-bar"></span>
22                     <span class="icon-bar"></span>
23                     <span class="icon-bar"></span>
24                 </button>
25                 <a class="navbar-brand" href="index.html">Catal'info</a>
26             </div>
27             <div id="navbar" class="navbar-collapse collapse" aria-expanded=
28                 "false" style="height: 1px;">
29                 <form class="navbar-form navbar-left">
30                     <div class="input-group col-xs-12">
31                         <input type="text" class="form-control" placeholder=
32                             "Rechercher...">
33                         <span class="input-group-btn">
34                             <button class="btn btn-success" type="button"><span
35                                 class="glyphicon glyphicon-search"></span></button>
36                         </span>
37                     </div>
38                 </form>
39                 <form class="navbar-form navbar-right" role="form">
40                     <button name="login" type="submit" class="btn btn-success">
41                         Connexion <span class="glyphicon glyphicon-log-in"
42                         ></span></button>
43                 </form>
44             </div>
45         </nav>
46         <div class="left">
47             <div class="col-sm-2 col-md-2">
48                 <div class="mini-submenu">
49                     <span class="icon-bar"></span>
50                     <span class="icon-bar"></span>
51                     <span class="icon-bar"></span>
52                 </div>
53                 <div class="list-group" style="display: none;">
54                     <h4 class="list-group-item">
55                         Recherche par catégorie

```

```

52         <span class="pull-right" id="slide-submenu">
53             <span class="glyphicon glyphicon-chevron-left"></span>
54         </span>
55     </h4>
56     <a href="#" class="list-group-item">
57         Lorem ipsum
58     </a>
59     <a href="#" class="list-group-item">
60         Lorem ipsum
61     </a>
62     <a href="#" class="list-group-item">
63         Lorem ipsum
64     </a>
65     <a href="#" class="list-group-item">
66         Lorem ipsum
67     </a>
68     <a href="#" class="list-group-item">
69         Lorem ipsum
70     </a>
71 </div>
72 </div>
73 </div>
74
75 <!-- CONTAINER -->
76 <div class="container">
77     <div class="container-fluid">
78         <!-- CONTAINER PANELS PRODUITS LES PLUS VUS-->
79         <div class="row well">
80             <h1><span class="glyphicon glyphicon-fire"></span> Les plus vus
81             </h1>
82             <hr/>
83             <div class="col-sm-3">
84                 <div class="panel panel-success">
85                     <div class="panel-heading">
86                         <h3 class="panel-title" style="text-align: center;">
87                             Nom du produit</h3>
88                     </div>
89                     <div class="panel-body">
90                         
94                         <hr/>
95                         <div class="text-info" style="float: left;">
96                             <p>
97                                 Short description...
98                             </p>
99                         </div>
100                     </div>
101                 </div>
102             </div>
103             <div class="col-sm-3">
104                 <div class="panel panel-success">
105                     <div class="panel-heading">
106                         <h3 class="panel-title" style="text-align: center;">
107                             Nom du produit</h3>
108                     </div>

```

```
103         <div class="panel-body">
104             
105             <hr/>
106             <div class="text-info" style="float: left;">
107                 <p>
108                     Short description...
109                 </p>
110             </div>
111         </div>
112     </div>
113 </div>
114 <div class="col-sm-3">
115     <div class="panel panel-success">
116         <div class="panel-heading">
117             <h3 class="panel-title" style="text-align: center">
                Nom du produit</h3>
118         </div>
119         <div class="panel-body">
120             
121             <hr/>
122             <div class="text-info" style="float: left;">
123                 <p>
124                     Short description...
125                 </p>
126             </div>
127         </div>
128     </div>
129 </div>
130 <div class="col-sm-3">
131     <div class="panel panel-success">
132         <div class="panel-heading">
133             <h3 class="panel-title" style="text-align: center">
                Nom du produit</h3>
134         </div>
135         <div class="panel-body">
136             
137             <hr/>
138             <div class="text-info" style="float: left;">
139                 <p>
140                     Short description...
141                 </p>
142             </div>
143         </div>
144     </div>
145 </div>
146 <div class="col-sm-3">
147     <div class="panel panel-success">
148         <div class="panel-heading">
```

```
149         <h3 class="panel-title" style="text-align: center">
        Nom du produit</h3>
150     </div>
151     <div class="panel-body">
152         
153         <hr/>
154         <div class="text-info" style="float: left;">
155             <p>
156                 Short description...
157             </p>
158         </div>
159     </div>
160 </div>
161 </div>
162 <div class="col-sm-3">
163     <div class="panel panel-success">
164         <div class="panel-heading">
165             <h3 class="panel-title" style="text-align: center">
        Nom du produit</h3>
166         </div>
167         <div class="panel-body">
168             
169             <hr/>
170             <div class="text-info" style="float: left;">
171                 <p>
172                     Short description...
173                 </p>
174             </div>
175         </div>
176     </div>
177 </div>
178 <div class="col-sm-3">
179     <div class="panel panel-success">
180         <div class="panel-heading">
181             <h3 class="panel-title" style="text-align: center">
        Nom du produit</h3>
182         </div>
183         <div class="panel-body">
184             
185             <hr/>
186             <div class="text-info" style="float: left;">
187                 <p>
188                     Short description...
189                 </p>
190             </div>
191         </div>
192     </div>
193 </div>
```



```
194         </div>
195         <!-- CONTAINER PANELS PRODUITS LES RECOMMANDES -->
196         <div class="row well">
197             <h1><span class="glyphicon glyphicon-tags"></span> Recommandé</h1>
198             <hr/>
199             <div class="col-sm-3">
200                 <div class="panel panel-success">
201                     <div class="panel-heading">
202                         <h3 class="panel-title" style="text-align: center;">
203                             Nom du produit</h3>
204                     </div>
205                     <div class="panel-body">
206                         
210                         <hr/>
211                         <div class="text-info" style="float: left;">
212                             <p>
213                                 Short description...
214                             </p>
215                         </div>
216                     </div>
217                 </div>
218             </div>
219             <div class="panel panel-success">
220                 <div class="panel-heading">
221                     <h3 class="panel-title" style="text-align: center;">
222                         Nom du produit</h3>
223                     </div>
224                     <div class="panel-body">
225                         
229                         <hr/>
230                         <div class="text-info" style="float: left;">
231                             <p>
232                                 Short description...
233                             </p>
234                         </div>
235                     </div>
236                 </div>
237             </div>
238         </div>
239         <div class="text-center">
240             <button type="button" class="btn btn-lg btn-success center-block">
241                 Afficher Plus <span class="glyphicon glyphicon-chevron-down"
242                 ></span></button>
243         </div>
244     </div>
245     <!-- Bootstrap core JavaScript
246     ===== -->
247     <script src="./js/jquery.min.js"></script>
248     <script src="./js/bootstrap.min.js"></script>
249     <script src="js/dropdown.js"></script>
250 </body>
251 </html>
```

```
1  <?php
2  require_once '../includes/specific_funtions.php';
3
4  //Affichage du hash
5  debug(hashPerso('SuperUser'));
6
7  if (filter_input(INPUT_POST, 'login')) {
8      $pseudo = filter_input(INPUT_POST, 'username');
9      $pass = filter_input(INPUT_POST, 'password');
10
11
12      if ($pseudo && $pass && userConnect($pseudo, $pass)){
13          echo "connect";
14      }
15  }
16  debug($_SESSION);
17
18  ?>
19
20  <form method="post" action="login_Test.php">
21      <label>Pseudo:</label>
22      <input name="username" type="text"/>
23      <input name="password" type="password"/>
24      <input name="login" type="submit"/>
25
26  </form>
```

```
1  <?php
2  require_once 'includes/specific_funtions.php';
3  require_once 'includes/struct.php';
4  require_once 'includes/crud_Produits.php';
5  require_once 'includes/crud_Brands.php';
6  require_once 'includes/crud_Medias.php';
7
8  if (!isAdmin()) {
9      header('Location: index.php');
10 }
11 //On récupère les marques et les mot-clefs
12 $brands = getBrandsSorted();
13 $keywords = getAllKeywordsSorted();
14
15 //On récupère l'id du produit et on regarde si on est en mode edition
16 $editMode = filter_input(INPUT_GET, 'edit');
17 $id = filter_input(INPUT_GET, 'id');
18
19 $addedProduct = false;
20 $addedMedia = false;
21
22 //Si on modifie un produit
23 if ($editMode == 1) {
24     //On recupère le produit
25     $product = getProductDetailsById($id);
26     $productKeywords = getProductKeywordsById($id, PDO::FETCH_ASSOC);
27     $productBrand = $product->id_brand;
28     $productMedias = getProductMediasById($id);
29     //On initialise les variables pour remplir les champs
30     $txtMode = '<span class="glyphicon glyphicon-cog"></span> Modifier un produit';
31     $txtTitle = $product->title;
32     $txtShortDesc = $product->short_desc;
33     $txtLongDesc = $product->long_desc;
34     $txtKeywords = '';
35     $txtStartDate = $product->availability_date;
36     $txtEndDate = $product->expiration_date;
37     $viewCount = $product->view_count;
38     //On change le formatage pour qu'il soit compatible avec la checkbox
39     if ($product->is_frontpage) {
40         $txtIsRecommended = 'checked';
41     }
42     else {
43         $txtIsRecommended = '';
44     }
45 } else {
46     //Si on est en train d'ajouter un produit
47     //On initialise les variables
48     $txtMode = '<span class="glyphicon glyphicon-plus-sign"></span> Ajouter un
    produit';
49     $txtTitle = '';
50     $txtShortDesc = '';
51     $txtLongDesc = '';
52     $txtKeywords = '';
53     $txtStartDate = '';
54     $txtEndDate = '';
55     $txtIsRecommended = '';
56 }
```

```
57
58 // On récupère les informations du formulaire
59 $title = filter_input(INPUT_POST, 'title');
60 $selectedBrand = filter_input(INPUT_POST, 'brands');
61 $shortDesc = filter_input(INPUT_POST, 'short_desc');
62 $longDesc = filter_input(INPUT_POST, 'long_desc');
63 $selectedKeywords = filter_input(INPUT_POST, 'keywords', FILTER_DEFAULT,
FILTER_FORCE_ARRAY);
64 $startDate = filter_input(INPUT_POST, 'availability_date');
65 $endDate = filter_input(INPUT_POST, 'expiration_date');
66 $isRecommended = filter_input(INPUT_POST, 'isRecommended');
67
68 //On récupère les informations des medias envoyés
69 $uploadedFiles = $_FILES;
70
71 $error = false;
72
73 //Si on à reçu des données en post
74 if (!empty($_POST)) {
75 //Verification ddu titre du produit
76     if (empty($title) || strlen($title) > "64") {
77         $errorMessages[] = "Vous n'avez pas entré de nom de produit, ou le nom était
trop long";
78         $error = true;
79     }
80
81 //Verification de la marque produit
82     if (empty($selectedBrand)) {
83         $errorMessages[] = "Vous n'avez pas sélectionné la marque du produit";
84         $error = true;
85     }
86 //Verification de la description courte du produit
87     if (empty($shortDesc) || strlen($shortDesc) > "255") {
88         $errorMessages[] = "Vous n'avez pas entré de description courte pour le
produit, ou la description était trop longue";
89         $error = true;
90     }
91 //Verification de la description longue du produit
92     if (empty($longDesc)) {
93         $errorMessages[] = "Vous n'avez pas entré de description longue pour le
produit";
94         $error = true;
95     }
96 //Verification de la date de debut
97     if (empty($startDate)) {
98         $errorMessages[] = "La date de sortie du produit n'a pas été sélectionnée";
99         $error = true;
100     }
101
102 //Verification de la date de fin
103     if (empty($endDate)) {
104         $errorMessages[] = "La date d'expiration du produit n'a pas été
sélectionnée";
105         $error = true;
106     }
107
108 //Verification de la checkbox
```

```

109     if ($isRecommended == "checked") {
110         $isRecommended = true;
111     } else {
112         $isRecommended = false;
113     }
114
115     //On ajoute les données du produit dans la base si il n'y a pas d'erreurs et si on à
    envoyé des données
116     if ($error == false) {
117         if ($editMode == 1) {
118             $idProduct = updateProduct($id, $title, $shortDesc, $longDesc,
                $isRecommended, $startDate, $endDate, $viewCount, $selectedBrand);
119             $errorList = '<div class="alert alert-success alert-error">
120                 <a href="#" class="close" data-dismiss="alert">&times;</a>
121                 Le produit #' . $idProduct . ' à été mis à jour
                    correctement
122                 </div>';
123         } else {
124             $idProduct = addProduct($title, $shortDesc, $longDesc, $isRecommended,
                $startDate, $endDate, $selectedBrand);
125         }
126         $addedProduct = true;
127
128         //On ajoute la liason entre le produit et les keywords si il n'y a pas
        d'erreurs
129         //Si on à envoyé un ou plus keyword
130         if (count($selectedKeywords) > 0) {
131             foreach ($selectedKeywords as $selectedKeyword) {
132                 addProductKeywordRelation($idProduct, $selectedKeyword);
133             }
134         }
135
136         //Si on à envoyé un ou plus media
137         if (count($uploadedFiles) > 0) {
138             //Pour chaque médias ajoutés
139             foreach ($uploadedFiles["medias"]["error"] as $key => $value) {
140                 //Si l'upload n'a pas eu d'erreurs
141                 if ($value == UPLOAD_ERR_OK) {
142                     //Si le fichier existe
143                     if (file_exists($uploadedFiles["medias"]["tmp_name"][$key])) {
144                         //On vérifie que la taille de fichier n'exède pas la taille
                            maximale
145                         if (filesize($uploadedFiles["medias"]["tmp_name"][$key]) <
                            MAX_FILE_SIZE) {
146                             //On récupère la vieille extension du fichier
147                             $oldExtension = pathinfo($uploadedFiles["medias"]["name"]
                                [$key], PATHINFO_EXTENSION);
148                             //On récupère l'ancien nom du fichier
149                             $oldFilename = pathinfo($uploadedFiles["medias"]["name"]
                                [$key], PATHINFO_FILENAME);
150                             //On crée un nom de fichier unique
151                             $hash = uniqid($_SESSION['id']) . '.' . $oldExtension;
152                             //On coupe le vieux nom du fichier pour le limiter à 10
                                caractères
153                             if (strlen($oldFilename) > 10) {
154                                 $oldFilename = substr($oldFilename, 0, 10);
155                             }

```

```

156 //On vérifie si le fichier fait parti des types d'images
    acceptée, sinon on le classe dans "autres"
157 if (checkImageType($uploadedFiles["medias"][$key]
    ))) {
158 //Si le fichier est bien une image, on l'ajoute dans
    le dossier "images"
159 $filename = IMG_FOLDER . $oldFilename . $hash;
160 //On indique que le fichier n'est pas une image
161 $isImage = true;
162 } else {
163 //Si le fichier n'est pas une image, on l'ajoute
    dans le dossier "autres"
164 $filename = OTHER_FOLDER . $oldFilename . $hash;
165 //On indique que le fichier n'est pas une image
166 $isImage = false;
167 }
168 //On déplace le fichier et on le renomme avant de
    l'ajouter au dossier de destination
169 move_uploaded_file($uploadedFiles["medias"][$key], $filename);
170 //On ajoute les donnée des media et sa relation avec le
    produit dans la base
171 $idMedia = addMedia($filename, $isImage);
172 addProductMediaRelation($idProduct, $idMedia);
173 $addedMedia = true;
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 if (!empty($errorMessages)) {
182     foreach ($errorMessages as $errorMessage) {
183         $errorList .= '<div class="alert alert-danger alert-error">
184             <a href="#" class="close" data-dismiss="alert">&times;</a>
185             ' . $errorMessage . '
186         </div>';
187     }
188 }
189
190 //Si on à ajouté un produit ou un media, on redirige la personne vers l'accueil
191 if (($addedMedia) || ($addedProduct)) {
192     if ($editMode == 1) {
193         header('Location: index.php?editSuccess=true');
194     } else {
195         header('Location: index.php?addSuccess=true');
196     }
197 }
198 ?>
199 <!DOCTYPE html>
200 <html lang="en"><head>
201     <meta charset="utf-8">
202     <meta http-equiv="X-UA-Compatible" content="IE=edge">
203     <meta name="viewport" content="width=device-width, initial-scale=1">
204     <meta name="description" content="">
205     <meta name="author" content="">

```

```

206         <title>Catal'info</title>
207         <link href="./css/bootstrap.min.css" rel="stylesheet">
208         <link href="./css/style.css" rel="stylesheet">
209         <link href="./css/font-awesome.css" rel="stylesheet">
210     </head>
211     <body>
212         <!-- NAVBAR -->
213         <?php
214         getHeader();
215         ?>
216
217         <!-- CONTAINER -->
218         <div class="container">
219             <div class="well form-container">
220                 <!-- CONTAINER LOGIN -->
221                 <div class="form-horizontal">
222                     <form class="form-add-product" enctype="multipart/form-data"
223                     method="post" action="">
224                         <h2><?php echo $txtMode; ?></h2><hr/>
225                         <label class="">Nom du produit :</label><input class=
226                         "form-control" name="title" type="text" value="<?php echo
227                         $txtTitle; ?>" required/><br/>
228
229                         <label class="">Marque :</label>
230                         <select class="form-control" name="brands" required>
231                             <?php
232                             //On compare les IDs des marques du produit avec ceux de
233                             la liste, on selectionne ceux du produit
234                             foreach ($brands as $brand) {
235                                 if ($brand->id == $productBrand) {
236                                     echo '<option selected value="' . $brand->id .
237                                     '">' . $brand->name . '</option>';
238                                 } else {
239                                     echo '<option value="' . $brand->id . '">' .
240                                     $brand->name . '</option>';
241                                 }
242                             }
243                             ?>
244                         </select>
245                         <br/>
246
247                         <label class="">Description courte :</label><input class=
248                         "form-control" name="short_desc" type="text" value="<?php
249                         echo $txtShortDesc; ?>" required/><br/>
250                         <label class="">Description longue :</label><textarea class=
251                         "form-control" name="long_desc" rows="3" required><?php echo
252                         $txtLongDesc; ?></textarea><br/>
253
254                         <label class="">Mots-clefs :</label>
255                         <select class="form-control keyword-container" name=
256                         "keywords[]" multiple required>
257                             <?php
258                             //On compare les IDs des keywords du produit avec ceux
259                             de la liste, on selectionne ceux du produit
260                             $idProductKeywords = array_column($productKeywords,
261                             "idKeyword");
262                             foreach ($keywords as $keyword) {

```

```

250         if (in_array($keyword->id, $idProductKeywords)) {
251             echo '<option value="' . $keyword->id . '"
                selected>' . strtoupper($keyword->name) .
                '</option>';
252         } else {
253             echo '<option value="' . $keyword->id . '">' .
                strtoupper($keyword->name) . '</option>';
254         }
255     }
256     ?>
257 </select>
258 <br/>
259
260 <label class="">Date de sortie :</label><input class=
"form-control" name="availability_date" type="date" value="
<?php echo $txtStartDate; ?>" required/><br/>
261 <label class="">Date d'expiration :</label><input class=
"form-control" name="expiration_date" type="date" value="
<?php echo $txtEndDate; ?>" required/><br/>
262 <div class="well">
263     <?php
264     if ($editMode == 1) {
265         echo '<label class="">Ajouter des medias :</label>';
266     } else {
267         echo '<label class="">Medias du produit :</label>';
268     }
269     ?>
270 <hr/>
271 <div class="container-fluid">
272     <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12">
273         <?php
274         //Si on modifie le produit, ajouter un media
                n'est pas obligatoire
275         if ($editMode == 1) {
276             echo '<input class="" name="medias[]"
                type="file"
                accept=".png,.gif,.jpg,.jpeg,.pdf,.docx"
                multiple/>';
277         } else {
278             echo '<input class="" name="medias[]"
                type="file"
                accept=".png,.gif,.jpg,.jpeg,.pdf,.docx"
                multiple required/>';
279         }
280         ?>
281     </div>
282     <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12"
                hidden="true">
283         <input name="upload" class="btn btn-primary" type
                ="submit" value="Upload">
284     </div>
285 </div>
286 <br/>
287 <div class="progress" hidden="true">
288     <div class="progress-bar progress-bar-striped active"
                role="progressbar" aria-valuenow="45" aria-valuemin=
                "0" aria-valuemax="100" style="width: 100%"></div>

```



```

289         </div>
290         <hr/>
291         <div class="media">
292             <ul class="media-list">
293                 <?php
294                     if ($editMode == 1) {
295                         echo "<label>Gestion des medias :</label>";
296                         //Pour chaque medias, on vérifie si le il
297                         //est une image, et on l'ajoute accordement
298                         foreach ($productMedias as $productMedia) {
299                             $idMedia = $productMedia->idMedia;
300                             if ($productMedia->isImage) {
301                                 $mediaName = substr($productMedia->
302                                     mediaSource, strlen(IMG_FOLDER));
303                                 echo '<li class="media">
304                                     <div class="media-left">
305                                         <a href="#">
306                                             
314                                         </a>
315                                     </div>
316                                     <div class="media-body"
317                                         style="margin: 10px;">
318                                         <h4 class="media-heading">' .
319                                             $mediaName . '</h4>
320                                         <a href="delete.php?idMedia='
321                                             . $idMedia . '&idProduct=' .
322                                             $id . '" class="btn
323                                             btn-danger">Supprimer le
324                                             media</a>
325                                     </div>
326                                 </li>';
327                             } else {
328                                 $mediaName = substr($productMedia->
329                                     mediaSource, strlen(OTHER_FOLDER));
330                                 echo '<li class="media">
331                                     <div class="media-left">
332                                         <a href="#">
333                                             
340                                         </a>
341                                     </div>
342                                     <div class="media-body"
343                                         style="margin: 10px;">
344                                         <h4 class="media-heading">' .
345                                             $mediaName . '</h4>

```

```

322         <a href="delete.php?idMedia='
           . $idMedia . '&idProduct=' .
           $id . '" class="btn
           btn-danger">Supprimer le
           media</a>

323     </div>
324 </li>';
325     }
326 }
327 }
328 ?>
329
330 </ul>
331 </div>
332 </div>
333 <label>
334     <input type="checkbox" value="checked" name=
           "isRecommended" <?php echo $txtIsRecommended ?>>
           En première page
335 </label>
336 <br/>
337 <br/>
338 <?php
339 if ($editMode == 1) {
340     echo '<input name="submit" class="btn btn-success"
           type="submit" value="Modifier le produit"/>';
341 } else {
342     echo '<input name="submit" class="btn btn-success"
           type="submit" value="Ajouter le produit"/>';
343 }
344 ?>
345 </form>
346 </div>
347 </div>
348 </div>
349 </div>
350 <!-- Bootstrap core JavaScript
351 ===== -->
352 <script src="./js/jquery.min.js"></script>
353 <script src="./js/bootstrap.min.js"></script>
354 <script src="js/dropdown.js"></script>
355 </body>
356 </html>

```

```
1  <?php
2  require_once 'includes/specific_funtions.php';
3  require_once 'includes/crud_Produits.php';
4  require_once 'includes/crud_Medias.php';
5
6  //On détruit le media envoyé en paramètre
7  $idMedia = filter_input(INPUT_GET, 'idMedia');
8  $idProduct = filter_input(INPUT_GET, 'idProduct');
9
10 //On récupère le "path" du media et on le supprime du server
11 $media = getMediaById($idMedia);
12 unlink($media->src);
13
14 deleteProductMediaById($idProduct, $idMedia);
15 header('Location: addProduct.php?edit=1&id='. $idProduct);
```

```

1  <?php
2  require_once 'includes/specific_funtions.php';
3  require_once './includes/struct.php';
4
5  //On initialise les variables
6  $adminControls = '';
7  $errorLogin = '';
8  $deleteSuccess = '';
9  $id = filter_input(INPUT_GET, 'id');
10
11
12  //Si le produit n'existe pas, on renvoie l'utilisateur à l'accueil
13  $product = getProductById($id);
14  if ($product == NULL) {
15      header('Location: index.php');
16  }
17
18  //Connexion utilisateur
19  if (filter_input(INPUT_POST, 'login')) {
20      $pseudo = filter_input(INPUT_POST, 'username');
21      $pass = filter_input(INPUT_POST, 'password');
22
23      if (!userConnect($pseudo, $pass)) {
24          $errorLogin = '<div class="alert alert-danger alert-error">
25                          <a href="#" class="close" data-dismiss="alert">&times;</a>
26                          Le nom d\'utilisateur ou le mot de passe est incorrect.
27                          </div>';
28      }
29  }
30
31  //On ajoute une vue à la page
32  addViewById($id);
33
34  //Si l'utilisateur est un administrateur on lui affiche les controles du produit
35  if (isAdmin()) {
36      $adminControls = '<a href="addProduct.php?edit=1&id=' . $id . '" class="btn
37                      btn-warning">
38                          <span class="glyphicon glyphicon-cog"
39                          aria-hidden="true"></span> Modifier le produit
40                      </a>
41                      <button class="btn btn-danger" data-toggle="modal"
42                      data-target="#deleteModal">
43                          <span class="glyphicon glyphicon-trash"
44                          aria-hidden="true"></span> Supprimer le produit
45                      </button>
46                      <hr/>';
47  }
48
49  if ((isAdmin()) && (filter_input(INPUT_POST, 'delete')) {
50      deleteProductById($id);
51      header('Location: index.php?deleteSuccess=true');
52  }
53  ?>
54  <!DOCTYPE html>
55  <html lang="en"><head>
56      <meta charset="UTF-8">
57      <meta http-equiv="Content-type" content="text/html; charset=UTF-8">

```

```
54     <meta name="viewport" content="width=device-width, initial-scale=1">
55     <meta name="description" content="">
56     <meta name="author" content="">
57     <title>Catal'info</title>
58     <link href="/css/bootstrap.min.css" rel="stylesheet">
59     <link href="/css/style.css" rel="stylesheet">
60     <link href="/css/font-awesome.css" rel="stylesheet">
61 </head>
62 <body>
63     <!-- NAVBAR -->
64     <?php
65     echo $errorLogin;
66     echo $deleteSuccess;
67     getHeader();
68     ?>
69     <div class="left">
70         <div class="col-sm-2 col-md-2">
71             <div class="mini-submenu">
72                 <span class="icon-bar"></span>
73                 <span class="icon-bar"></span>
74                 <span class="icon-bar"></span>
75             </div>
76             <div class="list-group" style="display: none;">
77                 <h4 class="list-group-item">
78                     Recherche par catégorie
79                     <span class="pull-right" id="slide-submenu">
80                         <span class="glyphicon glyphicon-chevron-left"></span>
81                     </span>
82                 </h4>
83                 <?php
84                 echo structKeywordsList();
85                 ?>
86             </div>
87         </div>
88     </div>
89
90     <!-- CONTAINER -->
91     <div class="container">
92         <?php
93         echo $adminControls
94         ?>
95         <div class="container-fluid">
96             <!-- CONTAINER PANELS PRODUITS LES PLUS VUS-->
97             <?php
98             echo structDetailProduct($id);
99             ?>
100             <h3>
101                 <?php
102                 echo structViewCount($id);
103                 ?>
104             </h3>
105         </div>
106     </div>
107 </div>
108
109 <!-- Modal connexion -->
110 <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-hidden=
```

```

"true">
111     <div class="modal-dialog">
112         <div class="modal-content">
113             <form class="form-signin" method="post" action="">
114                 <div class="modal-header">
115                     <button type="button" class="close" data-dismiss="modal"
                        aria-label="Close"><span aria-hidden="true">&times;
116                     </span></button>
117                     <h3 class="modal-title" id="myModalLabel">Connexion</h3>
118                 </div>
119                 <div class="modal-body">
120                     <label class="">Pseudo:</label><input class="form-control"
                        name="username" type="text" value="" placeholder="ex :
121                     'Johndoe'"/><br/>
122                     <label>Password :</label><input class="form-control" name=
123                     "password" type="password"/><br/>
124                 </div>
125                 <div class="modal-footer">
126                     <button type="button" class="btn btn-default" data-dismiss=
127                     "modal">Fermer</button>
128                     <input name="login" class="btn btn-success" type="submit"
129                     value="Se connecter"/>
130                 </div>
131             </form>
132         </div>
133     </div>
134 </div>
135 <!-- Modal suppression -->
136 <div class="modal fade" id="deleteModal" tabindex="-1" role="dialog" aria-hidden=
137 "true">
138     <div class="modal-dialog">
139         <div class="modal-content">
140             <form method="post" action="">
141                 <div class="modal-header">
142                     <button type="button" class="close" data-dismiss="modal"
143                     aria-label="Close"><span aria-hidden="true">&times;
144                     </span></button>
145                     <h3 class="modal-title" id="myModalLabel">Suppression</h3>
146                 </div>
147                 <div class="modal-body">
148                     <p>Êtes vous sur de vouloir supprimer le produit suivant : "
149                     <?php echo $product->title ?>" ?</p>
150                 </div>
151                 <div class="modal-footer">
152                     <button type="button" class="btn btn-default" data-dismiss=
153                     "modal">Fermer</button>
154                     <input name="delete" class="btn btn-danger" type="submit"
155                     value="Supprimer"/>
156                 </div>
157             </form>
158         </div>
159     </div>
160 </div>
161 <!-- Bootstrap core JavaScript
162 ===== -->

```

```
154     <script src="./js/jquery.min.js"></script>
155     <script src="./js/bootstrap.min.js"></script>
156     <script src="js/dropdown.js"></script>
157 </body>
158 </html>
```

```
1  <?php
2
3  if (isset($_GET['file'])) {
4      $file = $_GET['file'];
5      if (file_exists($file) && is_readable($file)) {
6          header('Content-Type: application/octet-stream');
7          header("Content-Disposition: attachment; filename=\"$file\"");
8          readfile($file);
9      }
10 } else {
11     header("HTTP/1.0 404 Not Found");
12     echo "<h1>Error 404: File Not Found: <br /><em>$file</em></h1>";
13 }
```



```
1  <?php
2  require_once 'includes/specific_funtions.php';
3  require_once './includes/struct.php';
4  $errorLogin = '';
5  $deleteSuccess = '';
6  $addSuccess = '';
7  $editSuccess = '';
8
9  //Connexion utilisateur
10 if (filter_input(INPUT_POST, 'login')) {
11
12     $pseudo = filter_input(INPUT_POST, 'username');
13     $pass = filter_input(INPUT_POST, 'password');
14
15     if (!userConnect($pseudo, $pass)) {
16         $errorLogin = '<div class="alert alert-danger alert-error">
17             <a href="#" class="close" data-dismiss="alert">&times;</a>
18             Le nom d\'utilisateur ou le mot de passe est incorrect.
19         </div>';
20     }
21 }
22
23 //Si on vient de supprimer un produit, on affiche un alert
24 if (filter_input(INPUT_GET, 'deleteSuccess') == true) {
25     $deleteSuccess = '<div class="alert alert-success">
26         <a href="#" class="close" data-dismiss="alert">&times;</a>
27         Le produit à été supprimé avec succès.
28     </div>';
29 }
30
31 if (isAdmin()) {
32     //Si on vient d'ajouter un produit, on affiche un alert
33     if (filter_input(INPUT_GET, 'addSuccess') == true) {
34         $addSuccess = '<div class="alert alert-success">
35             <a href="#" class="close" data-dismiss="alert">&times;</a>
36             Le produit à été ajouté avec succès.
37         </div>';
38     }
39
40     //Si on vient de modifier un produit, on affiche un alert
41     if (filter_input(INPUT_GET, 'editSuccess') == true) {
42         $editSuccess = '<div class="alert alert-success">
43             <a href="#" class="close" data-dismiss="alert">&times;</a>
44             Le produit à été modifié avec succès.
45         </div>';
46     }
47 }
48 ?>
49 <!DOCTYPE html>
50 <html lang="en"><head>
51     <meta charset="UTF-8">
52     <meta http-equiv="Content-type" content="text/html; charset=UTF-8">
53     <meta name="viewport" content="width=device-width, initial-scale=1">
54     <meta name="description" content="">
55     <meta name="author" content="">
56     <title>Catal'info</title>
57     <link href="./css/bootstrap.min.css" rel="stylesheet">
```

```

58         <link href="./css/style.css" rel="stylesheet">
59         <link href="./css/font-awesome.css" rel="stylesheet">
60     </head>
61     <body>
62         <!-- NAVBAR -->
63         <?php
64         echo $addSuccess;
65         echo $editSuccess;
66         echo $deleteSuccess;
67         echo $errorLogin;
68         getHeader();
69         ?>
70         <div class="left">
71             <div class="col-sm-2 col-md-2">
72                 <div class="mini-submenu">
73                     <span class="icon-bar"></span>
74                     <span class="icon-bar"></span>
75                     <span class="icon-bar"></span>
76                 </div>
77                 <div class="list-group" style="display: none;">
78                     <h4 class="list-group-item">
79                         Recherche par catégorie
80                         <span class="pull-right" id="slide-submenu">
81                             <span class="glyphicon glyphicon-chevron-left"></span>
82                         </span>
83                     </h4>
84                     <?php
85                     echo structKeywordsList();
86                     ?>
87                 </div>
88             </div>
89             <!-- CONTAINER -->
90             <div class="container">
91                 <!-- CONTAINER PANELS PRODUITS LES PLUS VUS-->
92                 <div class="well">
93                     <h1><span class="glyphicon glyphicon-fire"></span> Produits les
94                     plus vus</h1>
95                     <hr/>
96                     <div class="row">
97                         <!--INSERTION DES PRODUITS LES PLUS VU-->
98                         <?php
99                         echo structMostViewedProducts();
100                         ?>
101                     </div>
102                 </div>
103                 <!-- CONTAINER PANELS PRODUITS LES RECOMMANDES -->
104                 <div class="well">
105                     <h1><span class="glyphicon glyphicon-tags"></span> Produits
106                     recommandés</h1>
107                     <hr/>
108                     <div class="row">
109                         <!--INSERTION DES PRODUITS RECOMMANDES-->
110                         <?php
111                         echo structRecommendedProducts();
112                         ?>
113                     </div>
114                 </div>

```

```
113         </div>
114     </div>
115
116     <!-- Modal connexion -->
117     <div class="modal fade" id="myModal" tabindex="-1" role="dialog"
118     aria-labelledby="myModalLabel" aria-hidden="true">
119         <div class="modal-dialog">
120             <div class="modal-content">
121                 <form class="form-signin" method="post" action="">
122                     <div class="modal-header">
123                         <button type="button" class="close" data-dismiss="modal"
124                         aria-label="Close"><span aria-hidden="true">&times;
125                         </span></button>
126                         <h3 class="modal-title" id="myModalLabel">Connexion</h3>
127                     </div>
128                     <div class="modal-body">
129                         <label class="">Pseudo:</label><input class=
130                         "form-control" name="username" type="text" value=""
131                         required/><br/>
132                         <label>Password :</label><input class="form-control" name
133                         ="password" type="password" required/><br/>
134                     </div>
135                     <div class="modal-footer">
136                         <button type="button" class="btn btn-default"
137                         data-dismiss="modal">Fermer</button>
138                         <input name="login" class="btn btn-success" type="submit"
139                         value="Se connecter"/>
140                     </div>
141                 </form>
142             </div>
143         </div>
144     </div>
145
146     <!-- Bootstrap core JavaScript
147     ===== -->
148     <script src="./js/jquery.min.js"></script>
149     <script src="./js/bootstrap.min.js"></script>
150     <script src="js/dropdown.js"></script>
151 </body>
152 </html>
```

```
1  <?php
2  //On détruit la session de l'utilisateur et on le redirige à l'accueil
3  session_start();
4  session_unset();
5  session_destroy();
6
7  header('location: index.php');
8
```

```

1  <?php
2  require_once 'includes/specific_funtions.php';
3  require_once './includes/struct.php';
4  $errorLogin = '';
5  $keywordQuery = false;
6
7  //Connexion utilisateur
8  if (filter_input(INPUT_POST, 'login')) {
9
10     $pseudo = filter_input(INPUT_POST, 'username');
11     $pass = filter_input(INPUT_POST, 'password');
12
13     if (!userConnect($pseudo, $pass)) {
14         $errorLogin = '<div class="alert alert-danger alert-error">
15             <a href="#" class="close" data-dismiss="alert">&times;</a>
16             Le nom d\'utilisateur ou le mot de passe est incorrect.
17         </div>';
18     }
19 }
20
21 if (filter_input(INPUT_GET, 'queryKW')) {
22     $query = filter_input(INPUT_GET, 'queryKW');
23     $keywordQuery = true;
24 } else {
25     $query = filter_input(INPUT_GET, 'query');
26     $keywordQuery = false;
27 }
28 ?>
29 <!DOCTYPE html>
30 <html lang="en"><head>
31     <meta charset="UTF-8">
32     <meta http-equiv="Content-type" content="text/html; charset=UTF-8">
33     <meta name="viewport" content="width=device-width, initial-scale=1">
34     <meta name="description" content="">
35     <meta name="author" content="">
36     <title>Catal'info</title>
37     <link href="./css/bootstrap.min.css" rel="stylesheet">
38     <link href="./css/style.css" rel="stylesheet">
39     <link href="./css/font-awesome.css" rel="stylesheet">
40 </head>
41 <body>
42     <!-- NAVBAR -->
43     <?php
44     echo $errorLogin;
45     getHeader();
46     ?>
47     <div class="left">
48         <div class="col-sm-2 col-md-2">
49             <div class="mini-submenu">
50                 <span class="icon-bar"></span>
51                 <span class="icon-bar"></span>
52                 <span class="icon-bar"></span>
53             </div>
54             <div class="list-group" style="display: none;">
55                 <h4 class="list-group-item">
56                     Recherche par catégorie
57                     <span class="pull-right" id="slide-submenu">

```

```

58         <span class="glyphicon glyphicon-chevron-left"></span>
59     </span>
60 </h4>
61 <?php
62     echo structKeywordsList();
63 ?>
64 </div>
65 </div>
66 <!-- CONTAINER -->
67 <div class="container">
68     <!-- CONTAINER PANELS PRODUITS LES PLUS VUS-->
69     <div class="well">
70         <ul class="media-list">
71             <?php
72                 if ($keywordQuery) {
73                     echo structSearchedProductsKeywords($query);
74                 } else {
75                     echo structSearchedProducts($query);
76                 }
77             ?>
78         </ul>
79     </div>
80 </div>
81 </div>
82
83 <!-- Modal connexion -->
84 <div class="modal fade" id="myModal" tabindex="-1" role="dialog"
85     aria-labelledby="myModalLabel" aria-hidden="true">
86     <div class="modal-dialog">
87         <div class="modal-content">
88             <form class="form-signin" method="post" action="">
89                 <div class="modal-header">
90                     <button type="button" class="close" data-dismiss="modal"
91                         aria-label="Close"><span aria-hidden="true">&times;
92                     </span></button>
93                     <h3 class="modal-title" id="myModalLabel">Connexion</h3>
94                 </div>
95                 <div class="modal-body">
96                     <label class="">Pseudo:</label><input class="
97                         form-control" name="username" type="text" value=""
98                         required/><br/>
99                     <label>Password :</label><input class="form-control" name
100                         ="password" type="password" required/><br/>
101                 </div>
102                 <div class="modal-footer">
103                     <button type="button" class="btn btn-default"
104                         data-dismiss="modal">Fermer</button>
105                     <input name="login" class="btn btn-success" type="submit"
106                         value="Se connecter"/>
107                 </div>
108             </form>
109         </div>
110     </div>
111 </div>
112
113 <!-- Bootstrap core JavaScript
114 ===== -->
115 <script src="./js/jquery.min.js"></script>

```

```
107         <script src="./js/bootstrap.min.js"></script>
108         <script src="js/dropdown.js"></script>
109     </body>
110 </html>
```