



### Etude de cas

#### A TELECHARGER ET A LIRE EN AMONT

**Format d'évaluation :** QCM

**Documents autorisés :** tous

**Matériel autorisé :** calculatrice, tableur, etc.



Cette étude de cas contient tous les éléments nécessaires à votre épreuve.

**Cette étude de cas est l'énoncé de l'épreuve** sur lequel vous devez vous appuyer pour répondre aux questions qui seront posées dans le QCM.

**Aucune question ne figure dans ce document.**



Durant votre épreuve, **votre téléphone portable doit être éteint.**

**Tout échange avec un autre apprenant est interdit durant l'épreuve.**

Vous êtes autorisé à naviguer uniquement sur les sites métier. Sachez que votre connexion est tracée durant l'examen, nous sommes donc alertés en cas de fraude.

## À vous de jouer !

## Dossier 1 - Bloc front-end

Bk Agency est une agence de communication intégrée, issue de la fusion de Genesis et de Benchmark Agency, deux spécialistes de leurs domaines : le marketing opérationnel et le digital. Notre promesse : être provocateur d'opportunités pour nos clients. Aujourd'hui, le consommateur est soumis à la multiplication des points de contact avec les marques : sms, réseaux sociaux, courriers, affichage, presse, applications... Le risque de saturation est proche, mais en ayant une approche créative et user centric, nous ouvrons un champ des possibilités autant pertinent qu'efficace.

Vos missions consistent notamment à :

- Recueil des besoins fonctionnels remontés par les clients et l'équipe technique
- Développement front-end responsive (HTML5 et JavaScript)
- Mise en place et suivi de l'implémentation des optimisations SEO
- Support technique et fonctionnel de nos clients

Types de projets :

- Interventions et prise de brief en amont de projet
- Sites institutionnels mais aussi évènementiels
- Administration des contenus de sites internet
- Déploiement de webservice orienté marketing

## Dossier 2 : Bloc back-end

Nure est le pionnier et leader français du co-living, offrant aux jeunes nomades urbains des chambres meublées, équipées, tout inclus, connectées, réservables en 1 minute depuis un smartphone.

Votre mission est de construire en interne une plateforme digitale permettant la connexion de nos applications et logiciels (réalisation états des lieux, signature électronique, paiements, CRM, base de donnée...) pour piloter par exemple nos flux financiers, d'information, de communication, et permettre une collaboration fluide entre nos équipes, nos clients, nos propriétaires, nos artisans et prestataires.

L'objectif est de digitaliser et automatiser tout ce qui peut l'être pour décharger nos équipes des tâches à faible valeur ajoutée, limiter les erreurs, accroître la sécurité, rendre autonomes nos utilisateurs, réduire à zéro les délais de traitement, accentuer la scalabilité de notre business (déjà en très forte croissance), poursuivre la centralisation de nos métiers ...

Votre rôle :

- Concevoir et bâtir un back-end au service des besoins de nos clients et équipes
- Gérer, améliorer et maintenir nos base de données et notre base de code
- Garantir la qualité, la pertinence, la robustesse et la scalabilité de notre architecture technique

## Annexe 1.1 : "index.html"

```
1  <!DOCTYPE html>
2  <html Lang="fr">
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=1.0">
7      <link rel="stylesheet" href="style.css">
8
9      <title>Découvrez Bk Agency, notre agence de communication intégrée</title>
10 </head>
11 <body>
12     <h1>Bk Agency, la fusion du marketing opérationnel et du digital</h1>
13
14     <div class="post">
15         <h2 class="post-title">Lorem ipsum dolor sit amet</h2>
16         
17         <a href="/lorem">En savoir plus</a>
18     </div>
19 </body>
20 </html>
```

## Annexe 1.2 : “style.css”

```

1  .grid {
2      display: grid;
3      grid-template-columns: repeat(3, 1fr);
4      grid-template-rows: repeat(2, 1fr);
5  }
6
7  .post {
8      display: flex;
9      justify-content: center;
10 }
11
12 .post-title {
13     font-size: 2rem;
14     color: #ff0000;
15 }
16
17 .post-title:hover {
18     transition: margin-right 2s;
19 }
20
21
22 @media screen and (min-width: 798px) {
23     .post-title {
24         font-size: 3.25rem;
25     }
26 }

```

## Annexe 1.3 : “app.js”

```

1  let nbPartners = 7;
2  if(nbPartners === 7) {
3      console.log("Nombre de partenaires: "+ nbPartners);
4  }
5
6
7  const testimonials = [
8      "Une très belle collaboration, dans le respect et la confiance.",
9      "“Un accompagnement très pro pour la refonte de notre site web”,
10     "Michel nous a accompagnés et dynamisés au delà de notre demande",
11     "“Je ne sais pas ce qui se passe mais c’est génial, j’ai plein de demandes de consultation !”
12 ]
13
14 for(let i=0; i<testimonials.length;i++) {
15
16     showTestimonial(testimonials[i]);
17 }
18
19 let testimonialsElement = document.getElementById('testimonials');
20
21 XXX showTestimonial(testimonial) {
22     let newTestimonialElement = document.createElement('blockquote');
23     newTestimonialElement.textContent = testimonial;
24     testimonialsElement.append(newTestimonialElement)
25 }

```

## Annexe 1.4 : “OceanWP Thème enfant - style.css”

```

1  /*
2  Theme Name: OceanWP Child
3  Theme URI: https://oceanwp.org/
4  Description: OceanWP WordPress theme example child theme.
5  Author: Studi
6  Author URI: https://app.studi.fr
7  Template: oceanwp
8  Version: 1.0
9  */

```

## Annexe 2.1 : “languages.php”

```

1  <?php
2
3  $framework = 'Django';
4  $framework = 'Symfony';
5  echo $framework;
6  $framework = 'Spring';
7
8
9  $languages = ['JavaScript', 'Python', 'PHP', 'Rust'];
10
11 for ($i = 0; $i < count($languages); $i++) {
12     if($languages[$i] === 'PHP') {
13         echo 'My choice: ';
14     }
15     foo($languages[$i]);
16 }
17
18 XXX foo($language)
19 {
20     echo $language."<br />";
21     return $retval;
22 }

```

## Annexe 2.2 : “class.php”

```

1  <?php
2
3  class Drink {
4      public function getDescription(): string
5      {
6          return 'Une boisson';
7      }
8  }
9
10 class Soda extends Drink {
11     public function getDescription(): string
12     {
13         return parent::getDescription().' pétillante';
14     }
15 }
16
17 class Water extends Drink {}

```

## Annexe 2.3 : “table utilisateur” et “table commande”

**Table utilisateur**

id	prenom	nom	email	ville
1	Aimée	Marechal	aime.marechal@example.com	Paris
2	Esmée	Lefort	esmee.lefort@example.com	Lyon
3	Marine	Prevost	m.prevost@example.com	Lille
4	Luc	Rolland	lucrolland@example.com	Marseille

**Table commande**

utilisateur_id	date_achat	num_facture	prix_total
1	2013-01-23	A00103	203.14
1	2013-02-14	A00104	124.00
2	2013-02-17	A00105	149.45
2	2013-02-21	A00106	235.35
5	2013-03-02	A00107	47.58

## Annexe 2.4 : “controller.php”

```

1  /**
2   * @Route("/user/new", name="user")
3   */
4  public function new(Request $request): Response
5  {
6      $user = new User();
7
8      $form = $this->createForm(UserType::class, $user);
9
10     $form->handleRequest($request);
11
12     if ($form->isSubmitted() && $form->isValid()) {
13         // ...
14     }
15
16     return $this->render('user/new.html.twig', [
17         'form' => $form->createView(),
18     ]);
19 }

```

## Annexe 2.5 : “template.php”

```

1  <!DOCTYPE HTML>
2  <html Lang="fr">
3      <head>
4          <meta charset="UTF-8">
5          <title>Document</title>
6      </head>
7      <body>
8          {{ form_start(form) }}
9          {{ form_row(form.email, { 'label' : 'Email' }) }}
10         {{ form_row(form.birthdate, { 'label' : 'Date de naissance' }) }}
11         <input type="submit" />
12         {{ form_end(form) }}
13     </body>
14 </html>
15

```



## Annexe 2.6 : "entity.php"

```

1  /**
2   * @ORM\Entity(repositoryClass="App\Repository\UserRepository")
3   */
4  class User
5  {
6      /**
7       * @ORM\Id()
8       * @ORM\GeneratedValue()
9       * @ORM\Column(type="integer")
10     */
11     private int $id;
12
13     /**
14      * @ORM\Column(type="string", length=255)
15     */
16     private string $email;
17
18     /**
19      * @ORM\Column(type="date", nullable=true)
20     */
21     private \DateTimeInterface $birthdate;
22

```

## Annexe 2.7 : “extension.php”

```

1  <?php
2  /*
3  Plugin Name: Display Admin User Welcome
4  Plugin URI: http://localhost/wordpress/plugins/
5  Description: ?
6  Author: John Doe
7  Version: 1.0.0
8  Author URI: http://localhost/wordpress/
9  */
10
11 // Définition de l'action et de ses fonctionnalités
12 function display_user_welcome() {
13     do_action( 'display_user_welcome' );
14 }
15
16 // Déclaration de l'action personnalisée.
17 add_action( 'display_user_welcome', 'callback_display_user_welcome' );
18
19 function callback_display_user_welcome(){
20     $current_user = wp_get_current_user();
21     echo '<p>Bienvenue ' . $current_user->display_name . '! Votre adresse email est : ' .
22     $current_user->user_email . '</p>';
23 }
24
25 // Déclaration de l'action WordPress
26 add_action( 'admin_notices', 'callback_admin_notices' );
27
28 // Fonction callback
29 function callback_admin_notices() {
30     display_user_welcome();
31 }

```