

# DS – Développement WEB – BUT2 DACS

12/02/2025

Prénom :

NOM :

Règles :

- ☐ Question choix unique
- ☐ Question choix multiple
  - Uniquement les bonnes réponses
- Question libre
- Temps : 1h30
- Points totaux : 40
- Objectif : Répondre aux questions

---

## Partie 1 – Bases de PHP

Question 1 – Quelles affirmations s'appliquent à PHP

/2

- ☐ Scripting
  - ☐ Compilé
  - ☐ Interprété
  - ☐ Multi-plateforme
  - ☐ Faiblement typé
  - ☐ Exécuté côté client
- 

Question 2 – Quel code PHP ne retourne aucune erreur

/1

☐

```
<?php  
  
<div>  
  <p>Mon paragraphe</p>  
  <div><?php echo "test"; ?></div>  
</div>  
  
?>
```

☐

```
<div>  
  <p>Mon paragraphe</p>  
  <div><?php echo $ma_var; ?></div>  
</div>
```

☐

```
<div>  
  <p>Mon paragraphe</p>  
  <div><?php echo $_SERVER["SERVER_ADDR"]; ?></div>  
</div>
```

Question 3 – Décrire le contenu des superglobales/variables prédéfinies suivantes

/4

\$GLOBALS	
\$_SERVER	
\$_GET	
\$_POST	
\$_FILES	
\$_COOKIE	
\$_SESSION	
\$_REQUEST	
\$_ENV	
\$argc	
\$argv	

Question 4 – Qu'est-ce que permet de tester le regex suivant ? `/^[\\w\\-\\.]+@[a-z]+\\. [a-z]{2,4}$` /1

---

Question 5 – Quelles remarques pouvez-vous faire sur le code PHP suivant ? (1 point pour le détail)

```
1 <?php
2 $pdo = new PDO("mysql:host=localhost;dbname=testdb", "user", "password");
3
4 $username = $_GET['username'];
5 $query = "SELECT * FROM users WHERE username = '$username'";
6
7 $result = $pdo->query($query);
8
9 foreach ($result as $row) {
10     echo "User: " . $row['username'] . "<br>";
11 }
12 ?>
```

/3

---

**Question 6 – Qu'est-ce qu'il va se passer si j'effectue la requête suivante sur le code ci-dessous ? (2 points pour le détail)**

```
POST /register HTTP/2.0
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 13

username=IUT&mail=mon@super.mail&nationality=italien&password=mon_super_mdp
```

```
1 <?php
2
3 if (!empty($_POST)) {
4     $errors = [];
5
6     $username = $_POST['username'];
7     if (!isset($username) || !is_string($username) || strlen($username) < 4) {
8         $errors[] = "Erreur sur le nom d'utilisateur";
9     }
10
11     $mail = $_POST['mail'];
12     if (!isset($mail) || !is_string($mail) || !str_contains($mail, "@")) {
13         $errors[] = "Erreur sur l'email";
14     }
15
16     $nationality = $_POST['nationality'];
17     if (!isset($nationality) || !is_string($nationality) || strlen($nationality) <= 0
18 || !in_array($nationality, ["Français", "Anglais", "Italien"])) {
19         $errors[] = "Erreur sur la nationalité";
20     }
21
22     $password = $_POST['password'];
23     if (!isset($password) || !is_string($password) || strlen($password) < 4) {
24         $errors[] = "Erreur sur le mot de passe";
25     }
26
27     if (sizeof($errors) == 0) {
28         header('Location: profile.php');
29         return;
30     }
31 }
32 ?>
```

/3

## Partie 2 – Concepts du développement WEB

Question 7 – Remplissez le schéma suivant représentant une stack web (avec les communications) /2



Question 8 – Décrivez le rôle du client dans une stack web /1

---

Question 9 – Décrivez le rôle du front-end dans une stack web /1

---

Question 10 – Décrivez le rôle du back-end dans une stack web /1

---

Question 11 – Décrivez le rôle de la base-de-données dans une stack web /1

---

Question 12 – Quel terme correspond à l'ensemble du front-end, back-end et de la base de données /1

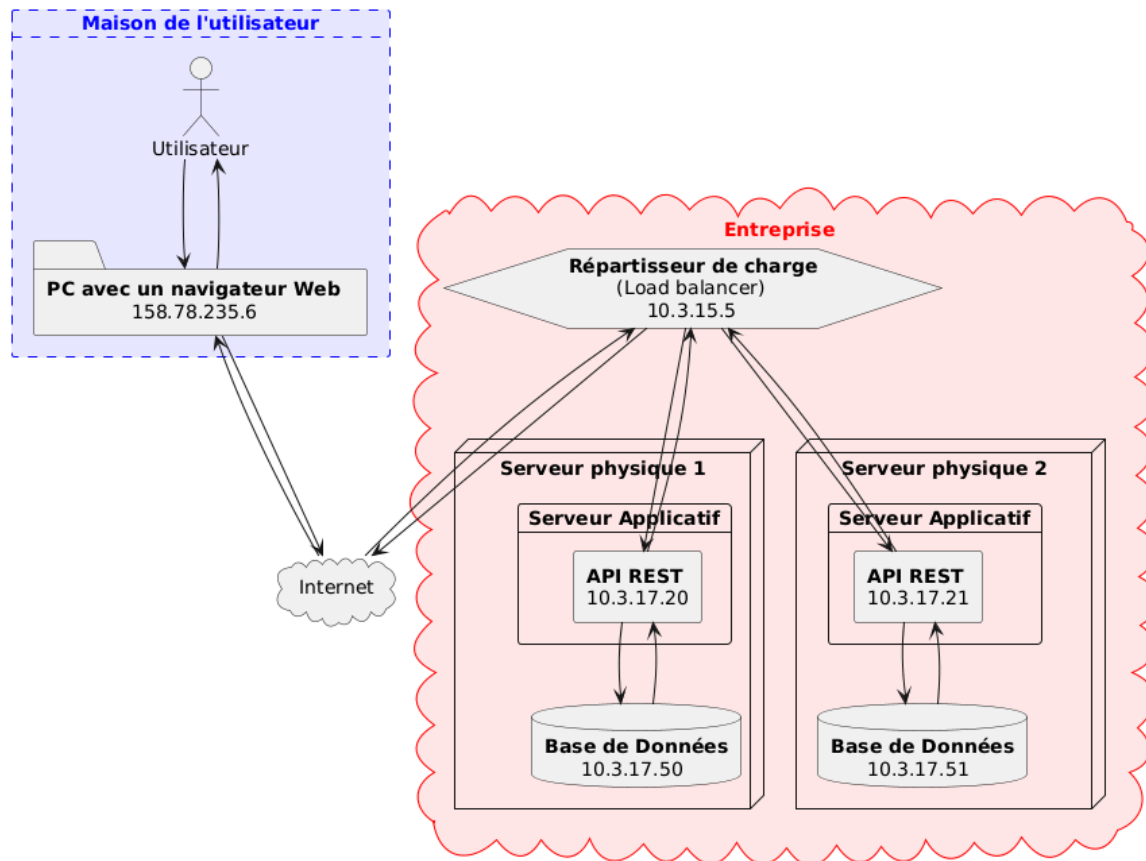
- ☐ Stack WEB
- ☐ Full-stack

**Question 13 – Numérotez l'ordre de communication entre les différents éléments du schéma pour la requête HTTP suivante**

La requête ci-dessous est celle interceptée au premier passage dans le load balancer.

```
GET /products/69 HTTP/1.1
Host: www.entreprise.com
User-Agent: Mozilla/5.0
Accept: application/json
X-Target-Server: backend-server-1
X-Forwarded-For: 158.78.235.6
```

/3



**Question 14 – Quelle est la différence entre le rendu côté client (client-side rendering, CSR) et le rendu côté serveur (server-side rendering, SSR) ? Quels sont les avantages et inconvénients de chaque approche ?**

/4

### Partie 3 – HTTP & API

#### Question 15 – Relier les codes de retour HTTP à leur catégorie

/2

- |                       |       |
|-----------------------|-------|
| • Information         | • 1XX |
| • Redirection         | • 2XX |
| • Erreur côté client  | • 3XX |
| • Succès              | • 4XX |
| • Erreur côté serveur | • 5XX |
- 

#### Question 16 – Qu'est-ce qu'est HTTPS par rapport à HTTP ?

/1

---

#### Question 17 – Qu'est-ce qu'une API ?

/2

---

#### Question 18 – Quelles méthodes standards existent dans le protocole HTTP ?

/1

- ☐ GET
- ☐ LOGIN
- ☐ UPDATE
- ☐ PUT
- ☐ POST
- ☐ REMOVE

**Question 19 – Ici, un simple échange entre un client (requête) et un serveur (réponse). Est-ce que la réponse du serveur semble appropriée ?**

Requête

```
GET /articles/42 HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept: application/xml
```

Réponse

```
HTTP/1.1 401 OK
Content-Type: application/json
Content-Length: 126

{
  "id": 42,
  "title": "Nouvel Article",
  "content": "Voici le contenu de
l'article."
}
```

/2

#### **Partie 4 – Framework et autres technologies**

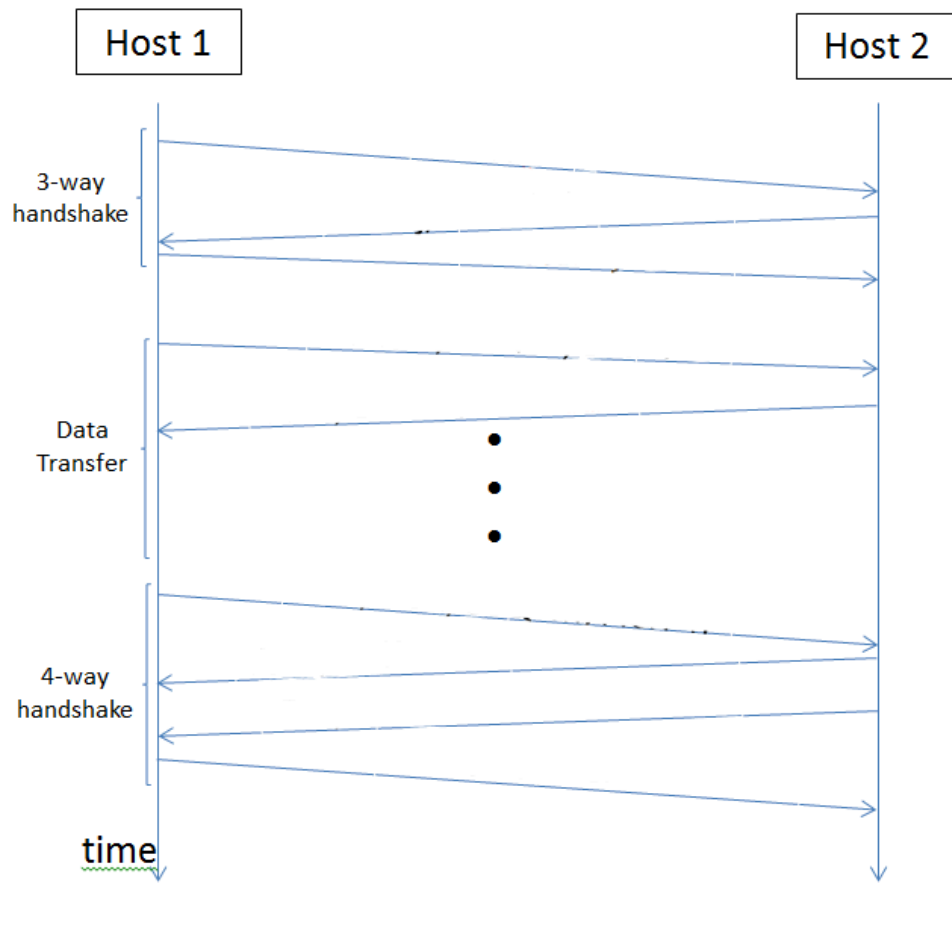
**Question 20 – Dans le futur, quels éléments pourraient vous permettre de comparer et choisir des technologies ? (framework, langage, bibliothèque, ...)**

/4

**Question 21 – Remplissez le schéma suivant représentant un échange TCP**

Les instructions possibles sont : SYN, ACK, SEQ, DATA, FIN

/2

**Question 22 – Comment fonctionne le système de migrations dans Laravel, et quels sont ses avantages par rapport à la gestion manuelle des bases de données ?**

/1

**Question 23 – Comment fonctionnent les composants Blade dans Laravel, et dans quels cas est-il préférable de les utiliser par rapport aux vues classiques ?**

/1