



# Rapport de Travaux Pratiques

**Première Année Transformation Digitale & AI**

**TP 2 / TP 3 :  
Développement WEB**

**Réalisé par :**

-

**Nouisser Zakaria**

**Encadré par :** Pr. Mohamed CHERRADI

---

**Année Universitaire : 2023/2024**

# Table des matières

---

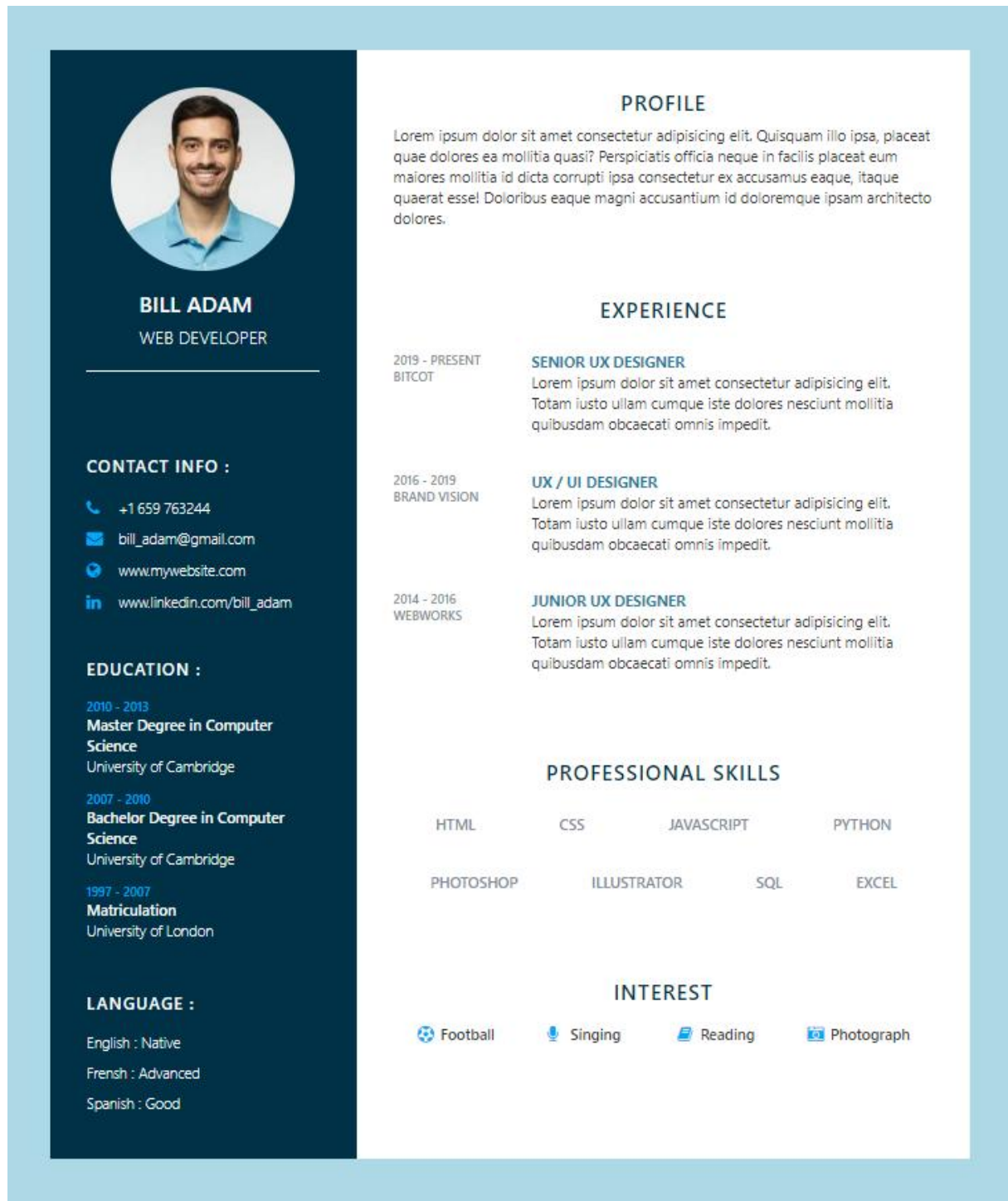
I- Titre 1 .....	Error! Bookmark not defined.
1. Sous-titre 1 .....	Error! Bookmark not defined.
1.1. Sous-titre 1 1 .....	Error! Bookmark not defined.
1.2. Sous-titre 1 2 .....	Error! Bookmark not defined.
2. Sous-titre 2 .....	8
II- Titre 2 .....	Error! Bookmark not defined.
1. Sous-titre 1 .....	Error! Bookmark not defined.
2. Sous-titre 2 .....	Error! Bookmark not defined.
3. Sous-titre 3 .....	Error! Bookmark not defined.

## I- TP - 2:

### 1. EX1:

Vous trouverez l'exercice dans le lien github ci-dessous:

[https://github.com/nwiuser/cv\\_tp2\\_web/tree/main](https://github.com/nwiuser/cv_tp2_web/tree/main)



## 2. EX2:

Tous les éléments `<a>` emboîtés dans un élément `<p>`.

```
/*question 1*/  
  
p a {  
    background-color: beige;  
}
```

Tous les éléments `<img>` appartenant à la classe *encadre* et placés dans un élément `<div>` appartenant à la classe *enonce*.

```
/*question 2*/  
  
.enonce div.encadre img {  
    color: black;  
}
```

Tous les éléments `<img>` dont l'attribut *width* vaut 300.

```
/*question 3*/  
  
img[width="300"] {  
    font-size: large;  
}
```

Tous les éléments `<time>` dont l'attribut *datetime* a été défini et qui sont situés dans un élément `<article>` appartenant à la classe *remarque*.

```
/*question 4*/  
  
article.remarque time[datetime] {
```

```
font-family: 'Courier New', Courier, monospace;
}
```

Tous les éléments `<p>` qui sont fils d'un élément `<div>` de la classe *exemple*.

```
/*question 5*/
div.exemple > p {
    display: block;
}
```

Tous les éléments `<img>` de classe *encadre* placés dans un élément `<p>` dont l'id est *important*.

```
/*question 6*/
p#important img.encadre {
    font-weight: 200;
}
```

Le premier élément `<img>` dont le père est un élément `<div>`.

```
/*question 7*/
div > img:first-child {
    max-width: 200px;
}
```

Tous les éléments `<img>` fils de rang impair d'un élément quelconque du document html.

```
/*question 8*/
```

```

*:nth-child(odd) > img {
    background-color: blue;
}

```

Tous les éléments <a> dont la cible est un fichier d'extension .jpg situé dans le premier élément <p> qui est fils d'un élément <article> de la classe *exercice*.

```

/*question 9*/
article.exercice > p:first-of-type a[href$=".jpg"] {
    display: grid;
    grid-template-columns: auto;
}

```

Les éléments <div> de la classe *remarque* qui sont les premiers frères (suivant) d'un élément <div> de classe *reponse*.

```

/*question 10*/
div.reponse + div.remarque {
    float: left;
}

```

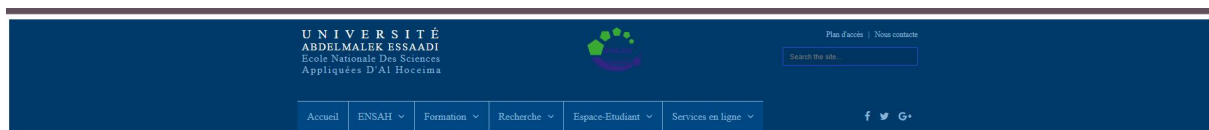
Le second élément de type <p> fils d'un élément <article> placé dans un élément <section>.

```

/*question 11*/
section article > p:nth-of-type(2) {
    float: left;
}

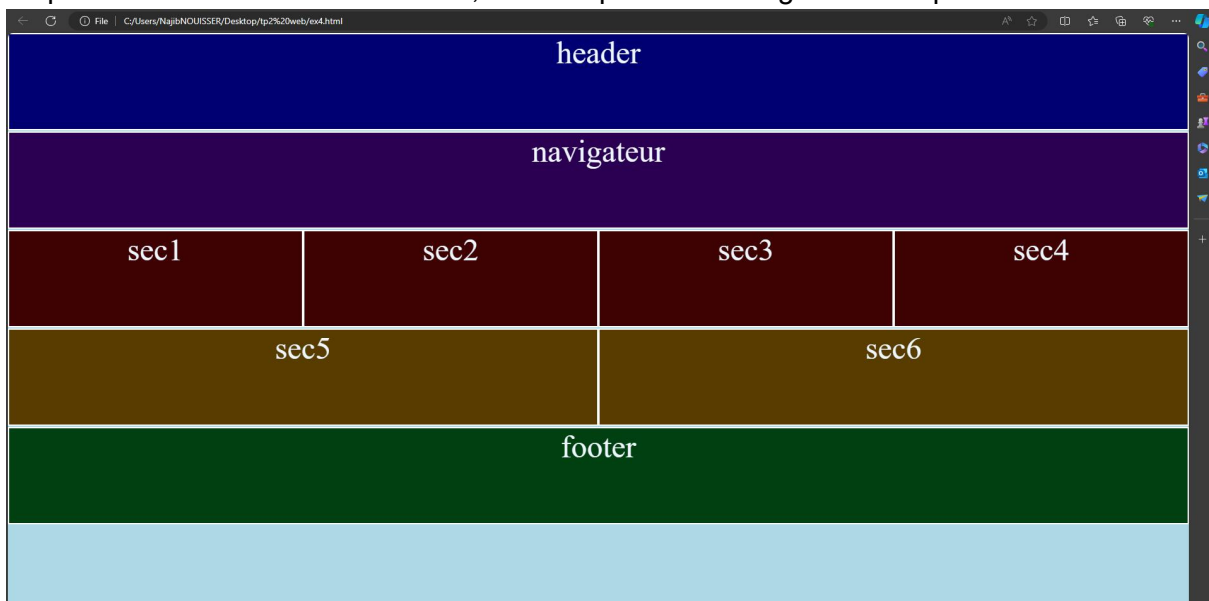
```

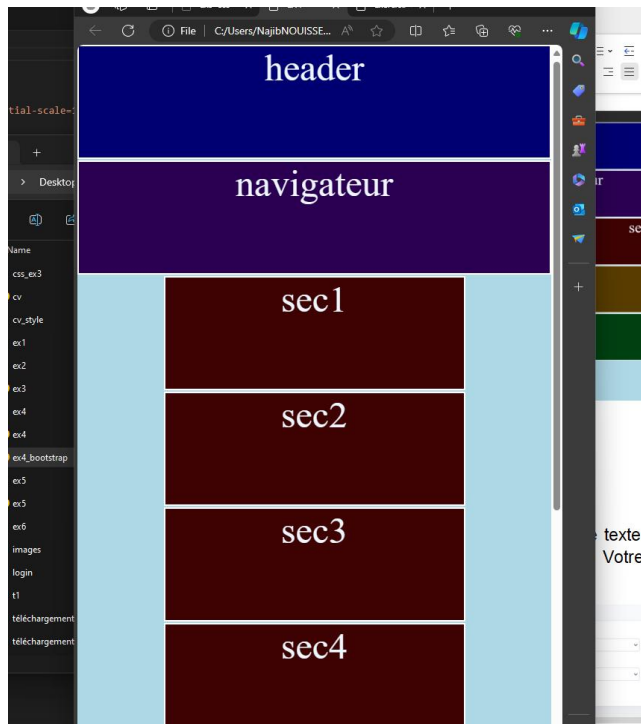
### 3. EX3:



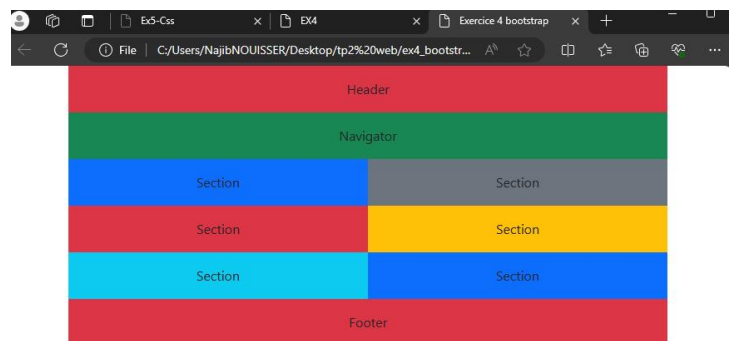
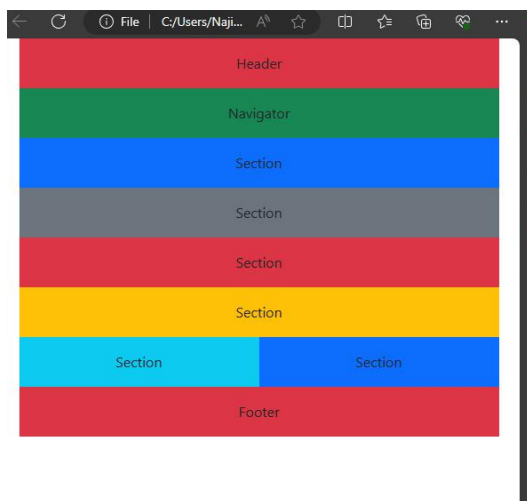
### 4. EX4:

La première est écrite avec Flexbox, avec responsive desing en Mediaqueries:



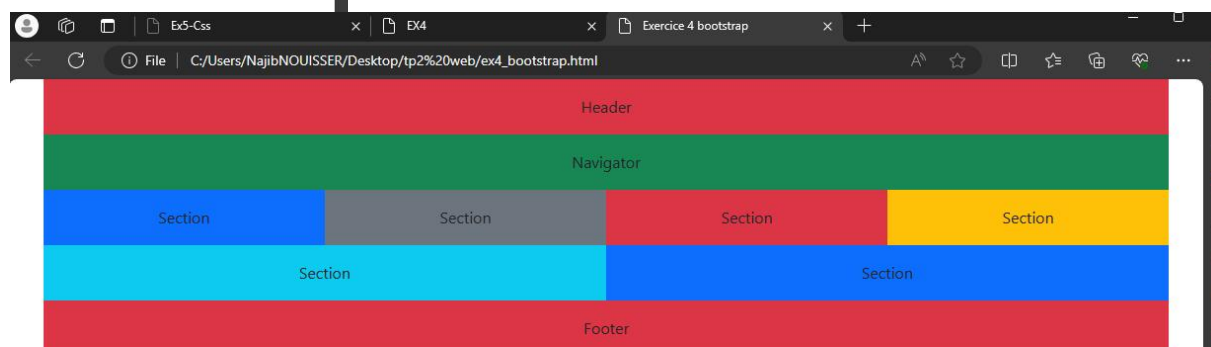


La deuxième est écrite en utilisant Bootstrap:

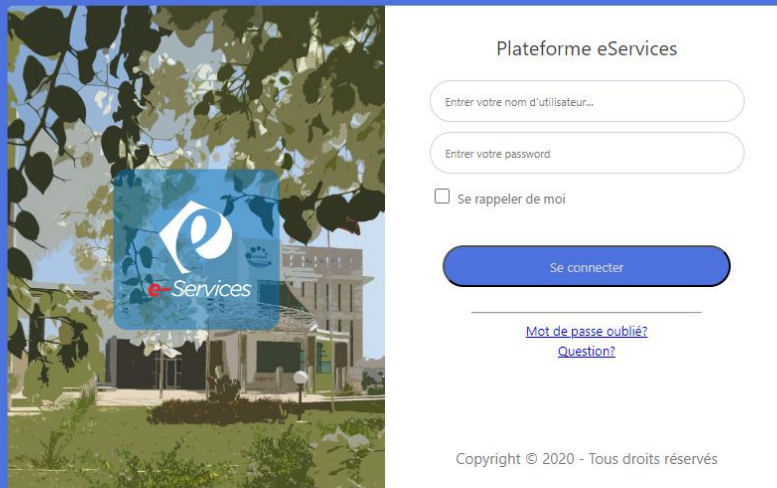


## 5. Ex5

La première est écrite en HTML et CSS:







La deuxième est écrite en Bootstrap:

## II- TP - 3:

### 1. EX1:

Le programme permet de trouver les nombres d'Armstrong qui sont inférieurs à 1000.

```
//--exercice 1

<?php
for($n = 0; $n < 1000; $n++){
    $q = (int)($n / 10);
    $r = $n % 10;
    $i = 0;
    $s = 0;
    while($i < strlen($n)){
        $s += $r ** strlen($n);
        $r = $q % 10;
        $q = (int)($q / 10);
        $i ++;
    }
    if($s == $n){
        echo "les nombres trouvées sont :" . $n . "<br>";
    }
}
?>
```

Résultat:

```
//--exercice 1 les nombres trouvées sont :0  
les nombres trouvées sont :1  
les nombres trouvées sont :2  
les nombres trouvées sont :3  
les nombres trouvées sont :4  
les nombres trouvées sont :5  
les nombres trouvées sont :6  
les nombres trouvées sont :7  
les nombres trouvées sont :8  
les nombres trouvées sont :9  
les nombres trouvées sont :153  
les nombres trouvées sont :370  
les nombres trouvées sont :371  
les nombres trouvées sont :407
```

## 2. EX2:

Le programme effectue une suite de tirages de nombres aléatoires jusqu'à obtenir une suite composée de deux nombres pair suivis d'un nombre impair.

```
//--exercice 2

<?php
$T = false;
$tab = array();
while($T == false){
    for($i = 0;$i<3; $i++){
        $tab[$i] = rand(0,100);
    }
    if($tab[0] % 2 == 0 && $tab[1] % 2 == 0 && $tab[2] % 2 != 0){
        $T = true;
    }
}
print_r($tab);

?>
```

Résultat:

```
//--exercice 2 Array ( [0] => 60 [1] => 20 [2] => 35 )
```

## 3. EX3:

Le programme permet de générer aléatoirement un nombre de trois chiffres N. Ensuite il effectue des tirages aléatoires jusqu'à réobtenir le même nombre N puis il affiche le nombre d'essais réalisés.

En utilisant d'abord 'while':

```
//avec boucle while

<?php

$t = rand(100,999);

$i = 0;

$p=0;

while($p != $t){

    $p = rand(100, 999);

    if($p != $t){

        $i ++;

    }

    else{

        break;

    }

}

echo "le nombre d'essais est : " . $i . "<br>";

?>
```

Résultat:

```
//avec boucle while le nombre d'essais est : 5
```

En utilisant 'for':

```
//avec boucle for

<?php

$t = rand(100,999);

$i = 0;
```

```
$p = 0;

for($j = 0; $p != $t; $j++){

    $p = rand(100, 999);

    if($p != $t){

        $i ++;

    }

    else{

        break;

    }

}

echo "le nombre d'essais est : " . $i . "<br>";

?>
```

Résultat:

---

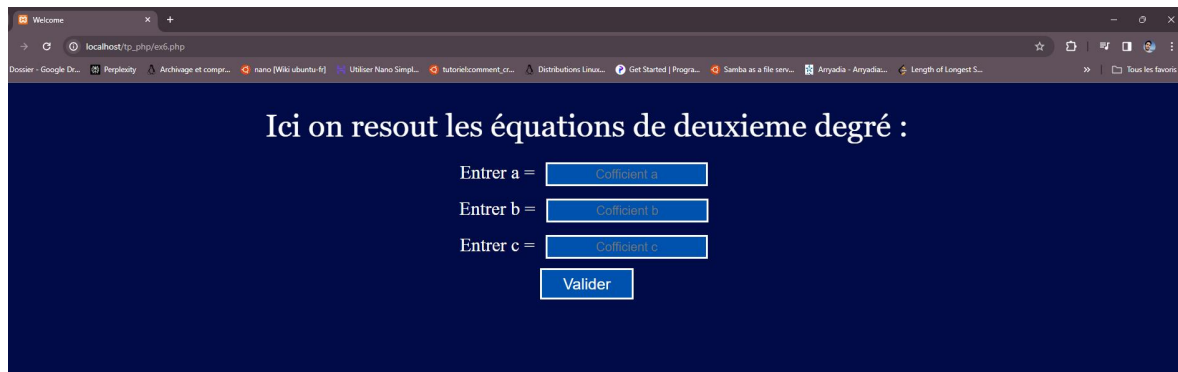
```
//avec boucle for le nombre d'essais est : 731
```

#### 4. EX4:

L'application web qui permet de résoudre les équations du deuxième degré en PHP.

L'utilisateur aura la possibilité de saisir les données nécessaires via un formulaire et l'application affiche la solution après la soumission de celui-ci.

D'abord on saisit des nombre dans les champs de saisie:



Le code app-polynome.php:

```
<!DOCTYPE html>

<html>

  <title>Welcome</title>

  <link rel="stylesheet" href="style.css">

  <body>

    <form action="result.php" method="get">

      <header id="marg">Ici on resout les équations de deuxieme degré :
</header>

      <label for="">Entrer a = </label><input class="bon" id="in"
type="text" name="a" placeholder="Coefficient a"><br>

      <label for="">Entrer b = </label><input class="bon" id="in"
type="text" name="b" placeholder="Coefficient b"><br>

      <label for="">Entrer c = </label><input class="bon" id="in"
type="text" name="c" placeholder="Coefficient c"><br>

      <input id="button" type="submit" name="send" value="Valider">

    </form>

  </body>

</html>
```

Les valeurs  $a = 1$ ,  $b = 5$ ,  $c = 3$  sont entrées pour être calculées dans le fichier result.php, et les afficher dans une autre fenêtre:



Voici le fichier result.php:

```
<?php

$msg;

$a = $_GET["a"];
$b = $_GET['b'];
$c = $_GET['c'];

$delta = ($b)**2 - 4*($a)*($c);

if($delta == 0 && $a != 0){

    $z = -($b) / (2*($a));

    $msg = "la solution est : " . number_format($z, 2) ;

}

elseif($delta > 0 && $a != 0){

    $x = (-($b) -sqrt($delta)) / (2*($a));

    $y = (-($b) +sqrt($delta)) / (2*($a));

    $msg = "les deux solutions sont : ( " . number_format($x,2) . " , " .
number_format($y, 2) . " )";

}

else{

    $msg = "les solutions ne sont pas dans IR";

}

?>

<!DOCTYPE html>

<html>
```



```
<title>Resultats</title>

<link rel="stylesheet" href="style.css">

<body>

    <h1 id="h1">Voici la solution : </h1>

    <h2 id="h2">

        <?php echo $msg ?>

    </h2>

</body>

</html>
```