

1. Présentation du système

Je vous propose de réaliser en **jquery** la page d'accès à vos comptes en ligne (CA).

CA Toute une banque pour vous NORD DE FRANCE

Saisissez votre numéro de compte : 28541541545

Cliquez dans la grille pour composer votre CODE PERSONNEL :

0	—	—
—	—	—
8	6	—
5	4	2
9	—	—
—	—	—
1	7	—
3	—	—

Votre code secret (6 chiffres) :

Corriger Annuler Soumettre

EN SAVOIR PLUS DÉCOUVREZ LA CARTE BANCAIRE ÉDITION LIMITÉE AUX COULEURS DES BLEUS

FRANCE FFF CA

2. Fonctionnement pour le n° de compte

🚦 Au chargement de la page, le **clavier numérique n'apparaît pas**.

CA Toute une banque pour vous NORD DE FRANCE

Saisissez votre numéro de compte : Numéro de compte...

Cliquez dans la grille pour composer votre CODE PERSONNEL :

Votre code secret (6 chiffres) :

Corriger Annuler

EN SAVOIR PLUS DÉCOUVREZ LA CARTE BANCAIRE ÉDITION LIMITÉE AUX COULEURS DES BLEUS

* Voir mentions et conditions sur site

FRANCE FFF CA

Page accueil - Crédit Agricole

Il n'apparaît que lorsque nous saisissons les 13 chiffres du n° de compte (**uniquement des chiffres**) :

CA NORD DE FRANCE
Toute une banque pour vous

Saisissez votre numéro de compte : 58768718717

Cliquez dans la grille pour composer votre CODE PERSONNEL :

—	—	4
—	—	—
3	—	5
—	8	—
0	—	—
—	1	—
—	6	9
2	7	—

Votre code secret (6 chiffres) :

Corriger Annuler

EN SAVOIR PLUS DÉCOUVREZ LA CARTE BANCAIRE ÉDITION LIMITÉE AUX COULEURS DES BLEUS

3. Architecture de la solution

e PC > Disque local (C:) > wamp64 > www > AgenceBancaireBdd-V4-2023				
Nom	Modifié le	Type	Taille	
images	03/09/2023 15:46	Dossier de fichiers		
PHP	03/09/2023 16:31	Dossier de fichiers		
accueil	03/09/2023 17:01	Avast HTML Docu...	3 Ko	
index	03/09/2023 17:07	Avast HTML Docu...	8 Ko	

4. index.html

L'essentiel du **code JQuery** suivant se situe au sein du fichier **index.html**

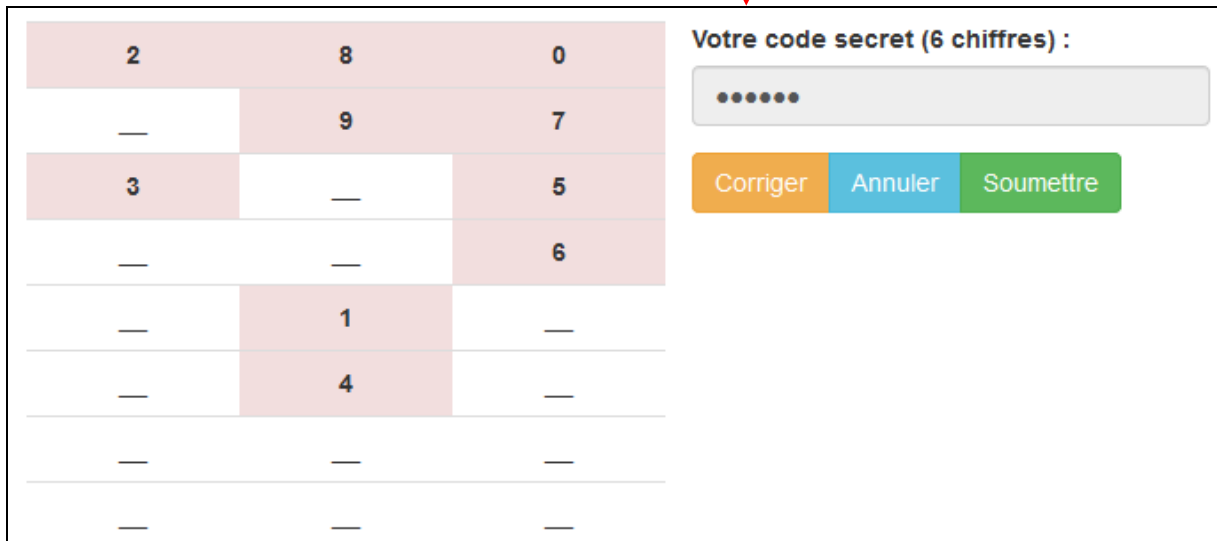
4.1. les chiffres

Tout caractère **non numérique est rejeté** (non pris en compte)

```
$("#Compte").on("input", function()
{
    leNumeroCompte = $(this).val();
    if (!$.isNumeric(leNumeroCompte) || (leNumeroCompte.length>13) )
    {
        leNumeroCompte = leNumeroCompte.slice(0, -1);
        $("#Compte").val(leNumeroCompte); // on supprime le dernier
caractère
    }
    .....
})
```

4.2. Fonctionnement pour le code secret

✚ Le code secret est un **code à 6 chiffres**.



Votre code secret (6 chiffres) :

.....

Corriger Annuler Soumettre

✚ Le code est généré par des **clics souris sur le clavier numérique** associé :



Votre code secret (6 chiffres) :

....

Corriger Annuler

✚ Lorsque le nombre de clics atteint **6**, le bouton **soumettre** apparaît :



Votre code secret (6 chiffres) :

.....

Corriger Annuler Soumettre

Un clic sur le bouton **Corriger** supprime le dernier chiffre du code et fait **disparaître le bouton soumettre** :

2	8	0
—	9	7
3	—	5
—	—	6
—	1	—
—	4	—
—	—	—
—	—	—

Votre code secret (6 chiffres) :

.....

Corriger Annuler

4.3. Quelques fonctions JQUERY utiles

4.3.1. showClavier()

Cette fonction fait apparaître le clavier numérique.

```
function showClavier()
{
    $("#clavier").html(table);
}
```

4.3.2. hideClavier()

Cette fonction fait disparaître le clavier numérique

```
function hideClavier()
{
    $("#clavier").html("");
}
```

4.3.3. Construction()

Cette fonction permet de générer aléatoirement un nouveau clavier numérique.

```
function Construction()
{
    GenerationAleatoire();
    GenererClavier();
}
```

4.3.4. GenerationAleatoire()

Cette fonction permet de générer **un tableau (8x3) avec les 10 chiffres disposés aléatoirement** au sein des cases.

```
<script>
var clavier = new Array(8);
clavier[0] = new Array(3);
clavier[1] = new Array(3);
clavier[2] = new Array(3);
clavier[3] = new Array(3);
clavier[4] = new Array(3);
clavier[5] = new Array(3);
clavier[6] = new Array(3);
clavier[7] = new Array(3);
```

```

var leCodeSecret    ="";
var leNumeroCompte  ="";
var table           ;

function GenerationAleatoire() {
    // initialisation du tableau 2d
    for(var i=0 ;      i<3 ;      i++)
        for(var j=0 ;      j<3 ;      j++)
            clavier [i][j] = -1 ;

    // générer aléatoirement la position des 10 chiffres
    for(var chiffre=0 ;      chiffre<10 ;      chiffre++) {
        do {
            ausuivant = false ;
            var cellule = Math.floor(Math.random() * 24) ;//0.....23
            var ligne   = Math.floor(cellule/3) ;      //0.....7
            var colonne = cellule % 3 ;      //0..2
            if(clavier [ligne][colonne]==-1) // si cellule pas déjà prise
            {
                clavier [ligne][colonne] = chiffre ;
                ausuivant = true ;
            }
        } while(ausuivant==false) ;
    }
}

```

4.3.5. GenererClavier()

Cette fonction permet de **créer une balise HTML** (<table></table>) à partir du tableau précédent.

```

function GenererClavier()
{
    table = "<table class='table .table-striped'>";
    // générer le tableau HTML avec le contenu du tableau précédent
    for(var i=0 ;      i<3 ;      i++) // car 3 lignes &
    {
        table += "<tr>";
        for(var j=0 ;      j<3 ;      j++)
        {
            val = clavier [i][j] ;
            if(val== -1)
                table += "<td >__</td>" ;
            else
                table += "<td class='danger'>"+val+"</td>" ;
        }
        table += "</tr>";
    }
    table += "</table>";
}

```

4.4. Le code HTML du body de "index.html"

```

<body>
<div class="container">

    <div class="row">
        <div class="jumbotron">
            </img>
        </div>
    </div>    <!-- fin de row -->

    <div class="row">
        <div class="input-group">
            <span class="input-group-addon">Saisissez votre numéro de
compte :</span>
            <input id="Compte" type="text" class="form-control"
placeholder="Numéro de compte...">
        </div>
    </div>    <!-- fin de row -->
<br>

```

```
<div class="row">
  <div class="col-xs-4 col-xs-offset-1">
    <div class="row">
      <h2> Cliquez dans la grille <br>pour composer
votre<br> CODE PERSONNEL : </h2>
    </div>
  </div>

  <div class="col-xs-4">
    <div id="clavier"></div>
  </div>

  <div class="col-xs-3">
    <div class="row">
      <div class="form-group">
        <label for="pwd">Votre code secret (6
chiffres) :</label>
        <input type="password" id="Code" readonly
class="form-control" >
      </div>
    </div>

    <div class="row">
      <div class="btn-group">
        <button type="button" id="corriger"
class="btn btn-warning">Corriger</button>
        <button type="button" id="annuler"
class="btn btn-info">Annuler</button>
        <button type="button" id="soumettre"
class="btn btn-success">Soumettre</button>
      </div>
    </div>
  </div> <!-- fin de row -->
</div>

<div class="row">
  <div class="jumbotron">
    </img>
  </div>
</div>

</div> <!-- fin de container -->
</body>
```

4.5. Le style

Très peu de choses ici

```
<style>

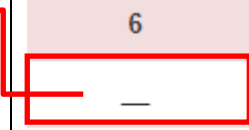
  td{
    text-align:center;
    font-weight:bold !important;
  }
</style>
```

4.6. Travail à réaliser

Il ne vous reste qu'à mettre en place **la logique de cette page**, c'est-à-dire gérer les événements suivant :

Le clic sur une **case du clavier numérique**

```
$(document).on('click', 'td', function()
{
    var leChiffre = $(this).text();
    .....
})
```



6	3	—
—	—	7
—	—	4
—	—	—
—	8	2
—	9	0
—	5	1
—	—	—

Le clic sur le bouton **corriger** :

```
$("#corriger").click(function()
{
    ...
})
```

6	3	—	Votre code secret (6 chiffres) : <input type="text"/> <input type="button" value="Corriger"/> <input type="button" value="Annuler"/>
—	—	7	
—	—	4	
—	—	—	
—	8	2	

Le clic sur le bouton **annuler** :

```
$("#annuler").click(function()
{
    ...
})
```

La **vérification du numéro de compte, 13 chiffres** (à compléter....)

```
$("#Compte").on("input", function()
{
    leNumeroCompte = $(this).val();
    ...
})
```

Saisissez votre numéro de compte :

✚ Le clic sur le bouton soumettre (à compléter....)

```
$("#soumettre").click(function()
{
    numeroCompte=$("#Compte").val();
    codeSecret = $("#Code").val();
    $.ajax({
        type : "POST",
        url : "PHP/login.php",
        data : { "numeroCompte" :
numeroCompte, "codeSecret" :
codeSecret},
        dataType: "json",
        success : handleResponse,
        .....
    })
})
```

```
function handleResponse(jsonDatas)
{
    if(jsonDatas.message == "OK")
    {
        firstName = jsonDatas.firstName ;
        lastName = jsonDatas.lastName ;
        window.location = './accueil.html?firstName='+ firstName+'&lastName='+ lastName;
    }
    else
    {
        // voir au sein de la suite du document .....
    }
}
```

- ✚ Cette action sera une **requête AJAX**.
- ✚ Cette requête consiste à demander l'exécution d'un script **"login.php"**
- ✚ Ce script PHP nous retournera une réponse de type OK (sous format JSON) si les **identifiants** sont corrects.

4.7. Le squelette de "index.html"

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Bootstrap Example</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <!-- jQuery library -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
    <!-- Latest compiled JavaScript -->
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

    <style> td{
        text-align:center;
        font-weight:bold !important;
    }
    </style>

<script>
    var clavier = new Array(8) ;
    clavier [0] = new Array(3) ;
    clavier [1] = new Array(3) ;
```



```
clavier [2]    = new Array(3) ;
clavier [3]    = new Array(3) ;
clavier [4]    = new Array(3) ;
clavier [5]    = new Array(3) ;
clavier [6]    = new Array(3) ;
clavier [7]    = new Array(3) ;
var leCodeSecret = '' ;
var leNumeroCompte = '' ;
var table      ;

$(function()
{
    Construction();
    // on cache le bouton de soumission
    $('#soumettre').hide();
    ...
    ...
    ...
});
</script>
</head>

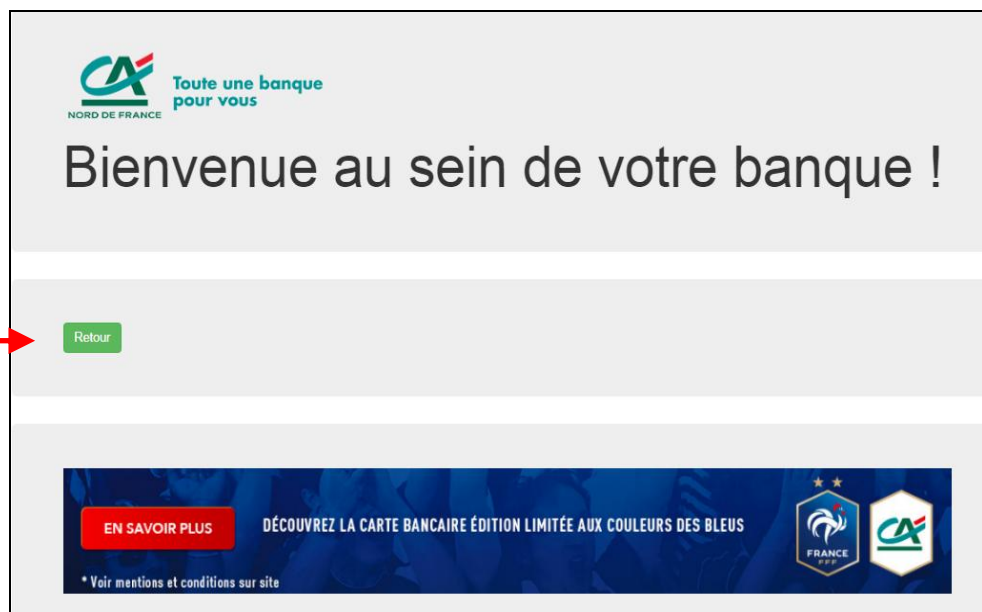
<body>
    <div class="container">
        <div class="row">
            <div class="jumbotron">
                </img>
                ...
                ...
                ...
                ...
            </div>
        </div>
    </body>
</html>
```

5. accueil.html

On souhaite maintenant poursuivre la démarche jusqu'à autoriser le client à **accéder à la page d'accueil** de son compte.

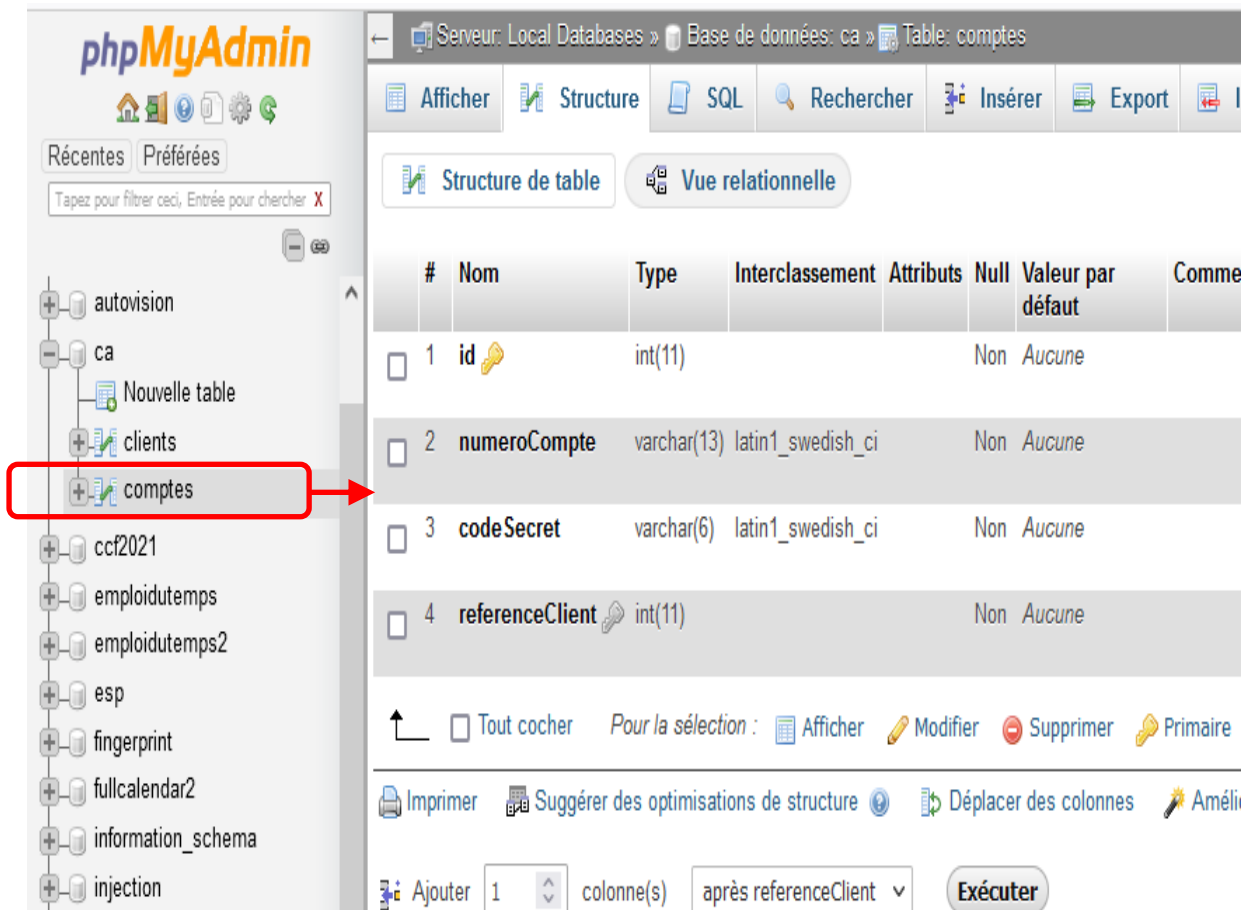
Voici un exemple de page d'accueil :

On y retrouve simplement un bouton afin **de revenir à la page de login**.



5.1. Base de données

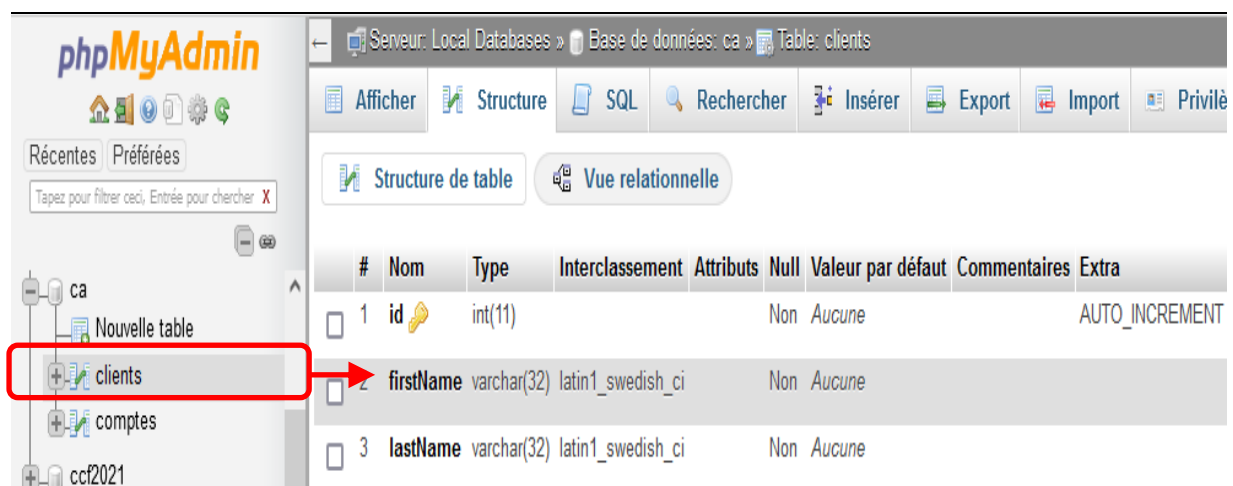
Pour cela, vous allez devoir **créer une base de données** afin d'y ranger les associations entre les numéros de compte des clients et leur mot de passe.



The screenshot shows the phpMyAdmin interface. On the left, the database 'ca' is selected, and the 'comptes' table is highlighted with a red box and an arrow. The main panel displays the 'Structure de table' for 'comptes'. The table has the following structure:

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Comme
1	id	int(11)			Non	Aucune	
2	numeroCompte	varchar(13)	latin1_swedish_ci		Non	Aucune	
3	codeSecret	varchar(6)	latin1_swedish_ci		Non	Aucune	
4	referenceClient	int(11)			Non	Aucune	

At the bottom, there is a button 'Ajouter' with a dropdown menu set to '1' and 'colonne(s) après referenceClient', and an 'Exécuter' button.



The screenshot shows the phpMyAdmin interface. On the left, the database 'ca' is selected, and the 'clients' table is highlighted with a red box and an arrow. The main panel displays the 'Structure de table' for 'clients'. The table has the following structure:



#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
1	id	int(11)			Non	Aucune		AUTO_INCREMENT
2	firstName	varchar(32)	latin1_swedish_ci		Non	Aucune		
3	lastName	varchar(32)	latin1_swedish_ci		Non	Aucune		

5.2. Service CRUD

Vous allez devoir écrire un **script php** de vérification de l'identité **du client** (numéro de compte et mot de passe).

Pour cela, nous allons développer :

- une **"class php"** que se nomme ici : crudPDO
- un **script php** qui se nomme : login.php

Ce PC > Disque local (C:) > wamp64 > www > AgenceBancaireBdd-V4-2023 > PHP				
Nom	Modifié le	Type	Taille	
 crudPDO	03/09/2023 15:48	Fichier PHP	4 Ko	
 login	03/09/2023 16:28	Fichier PHP	3 Ko	

5.2.1. crudPDO.php (intégralement fourni)

```
<?php
class CrudPDO {
    private $pdo;

    public function __construct($host, $dbname, $username, $password) {
        try {
            // Create a PDO instance for database connection
            $this->pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $username,
$password);

            // Set PDO to throw exceptions on errors
            $this->pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        } catch (PDOException $e) {
            throw new Exception("Database connection failed: " . $e->getMessage());
        }
    }

    public function create($table, $data) {
        try {
            // Build the SQL INSERT query
            $fields = implode(', ', array_keys($data));
            $values = ':' . implode(':', array_keys($data));
            $sql = "INSERT INTO $table ($fields) VALUES ($values)";

            // Prepare the statement
            $stmt = $this->pdo->prepare($sql);

            // Bind parameters and execute the statement
            $stmt->execute($data);

            return true;
        } catch (PDOException $e) {
            return false;
        }
    }

    public function read($table) {
        try {
            // Build the SQL SELECT query
            $sql = "SELECT * FROM $table";

            // Execute the query and fetch all records
            $stmt = $this->pdo->query($sql);
            $data = $stmt->fetchAll(PDO::FETCH_ASSOC);

            return $data;
        } catch (PDOException $e) {
            return [];
        }
    }
}
```

```
public function readCustom($sql, $params = []) {
    try {
        // Prepare the custom SQL query
        $stmt = $this->pdo->prepare($sql);

        // Bind parameters and execute the statement
        $stmt->execute($params);

        // Fetch all records
        $data = $stmt->fetchAll(PDO::FETCH_ASSOC);

        return $data;
    } catch (PDOException $e) {
        return [];
    }
}

public function update($table, $id, $data) {
    try {
        // Build the SQL UPDATE query
        $set = implode(' ', array_map(function ($key) {
            return "$key = :$key";
        }, array_keys($data)));
        $sql = "UPDATE $table SET $set WHERE id = :id";

        // Add the id to the data array
        $data['id'] = $id;

        // Prepare the statement
        $stmt = $this->pdo->prepare($sql);

        // Bind parameters and execute the statement
        $stmt->execute($data);
        return true;
    } catch (PDOException $e) {
        return false;
    }
}

public function delete($table, $id) {
    try {
        // Build the SQL DELETE query
        $sql = "DELETE FROM $table WHERE id = :id";

        // Prepare the statement
        $stmt = $this->pdo->prepare($sql);

        // Bind parameters and execute the statement
        $stmt->execute(['id' => $id]);
        return true;
    } catch (PDOException $e) {
        return false;
    }
}
?>
```

5.2.2. login.php

```
<?php
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Headers: access");
header("Access-Control-Allow-Methods: GET");
header("Access-Control-Allow-Credentials: true");
header("Content-Type: application/json; charset=UTF-8");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With");

require 'crudPDO.php';

$host      = '127.0.0.1';
$dbname    = 'CA';
$username  = 'root';
$password  = '';

try {
    // Attempt to create an instance of the CrudPDO class
    $crud = new CrudPDO($host, $dbname, $username, $password);

    // on teste si nos paramètres sont définies
    if(      isset($_POST['numeroCompte']) && isset($_POST['codeSecret']) )
        .....
        .....
        .....
```

5.2.3. Un exemple d'usage de la "class" crudPDO

Afin de vous aider dans le codage de login.php, voici **un exemple d'usage de la class "crudPDO"**

```
<?php
require_once 'CrudPDO.php';

// Database connection details
$host = 'localhost';
$dbname = 'your_database_name';
$username = 'your_username';
$password = 'your_password';
$crud = new CrudPDO($host, $dbname, $username, $password);

// Create a new record
$newUser = [
    'username' => 'john_doe',
    'email' => 'john.doe@example.com',
];
if ($crud->create('users', $newUser)) {
    echo "New user created successfully!";
} else {
    echo "Error creating user.";
}

// Read records
$users = $crud->read('users');
foreach ($users as $user) {
    echo "ID: " . $user['id'] . "<br>";
    echo "Username: " . $user['username'] . "<br>";
    echo "Email: " . $user['email'] . "<br>";
    echo "<hr>";
}
```

```
// Update a record
$userIdToUpdate = 1; // Replace with the ID of the user you want to update
$updatedUserData = [
    'username' => 'updated_username',
    'email' => 'updated_email@example.com',
];
if ($crud->update('users', $userIdToUpdate, $updatedUserData)) {
    echo "User updated successfully!";
} else {
    echo "Error updating user.";
}

// Delete a record
$userIdToDelete = 2; // Replace with the ID of the user you want to delete
if ($crud->delete('users', $userIdToDelete)) {
    echo "User deleted successfully!";
} else {
    echo "Error deleting user.";
}
?>
```

5.3. affichage des messages d'erreur (index.html)

En cas d'erreur (problème de connexion à la base de données **par exemple**), prévoir d'afficher l'information comme suit :

The screenshot shows the login interface of a Crédit Agricole branch (NORD DE FRANCE). It includes a logo, a text input for 'Saisissez votre numéro de compte', and a 'Numéro de compte...' label. Below this is a prompt to click on a grid to compose a 'CODE PERSONNEL'. To the right, there is a 'Votre code secret (6 chiffres):' input field with 'Contiger' and 'Annuler' buttons. An error message is displayed on the right side: 'Database connection failed: SQLSTATE[HY000] [1049] Base 'ca2' inconnue'. A red arrow points from the 'CODE PERSONNEL' input field to the error message. At the bottom, there is a banner for 'EN SAVOIR PLUS' and 'DÉCOUVREZ LA CARTE BANCAIRE ÉDITION LIMITÉE AUX COULEURS DES BLEUS' with logos for France and Crédit Agricole.

Ici, j'ai volontairement changé le nom de la base de données (**CA --> CA2**) afin de provoquer une erreur. Cet affichage se fera au sein de la fonction qui traite le retour de l'**appel ajax** :

```
function handleResponse(jsonData)
{
    if(jsonData.message == "OK") {
        .....
    }
    else
    {
        $("#response").fadeIn() ;
        $("#response").html("<h2>" + jsonData.message + "</h2>");
        $("#response").fadeOut(30000) ;
        $("#Compte").val("");
        $("#Code").val("");
        leCodeSecret = "";
        leNumeroCompte = "";
        hideClavier() ;
        $("#soumettre").hide();
    }
}
```

5.4. bouton "retour"

Vous allez devoir développer une page d'accueil



Code possible pour le bouton "Retour" :

```
$(function() {
    $('#retour').click(function() {
        url = "/index.html" ;
        window.location.replace(url) ;
    });
});
```

Début de code de cette page :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Page d'accueil</title>
    <meta charset="windows-1252">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

    <!-- jQuery library -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
    <!-- Latest compiled JavaScript -->
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

    <script type="text/javascript">
        $(function()
        {
            urlParams = new URLSearchParams(window.location.search);
            firstName = urlParams.get('firstName') ;
            lastName = urlParams.get('lastName') ;
```

```
.....
.....
.....

$("#retour").click(function() {
    url = "/agenceBancaire.html";
    window.location.replace(url);
});

});

</script>
.....
<body>
    <div class="container-fluid">
        <div class="row">
            <div class="jumbotron">
                </img>
                <h2>Bienvenue au sein de votre banque !</h2>
            </div>
        </div>
    </div>
    .....
```

5.5. Amélioration possible

Au sein de la table clients, ajoutez-y un champ "sexe" afin d'être en mesure de souhaiter la bienvenue de cette façon:

Bonjour Monsieur Dumoulin
ou
Bonjour Madame Lemoine

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut
<input type="checkbox"/> 1	id	int(11)			Non	Aucune
<input type="checkbox"/> 2	firstName	varchar(32)	latin1_swedish_ci		Non	Aucune
<input type="checkbox"/> 3	lastName	varchar(32)	latin1_swedish_ci		Non	Aucune
<input type="checkbox"/> 4	sexe	tinyint(1)			Non	Aucune

 **Toute une banque pour vous**

NORD DE FRANCE

Bienvenue au sein de votre banque !

Bonjour **Madame** Mireille Dupuis

[Retour](#)