1-COMPONENTS

A-Login form

Create a login form (Sign-In), including:

- 2 input fields named “Username” and “Password”
- 2 submit buttons named “Connect” and “Sign-Up” that allows players to submit the form or to access to the registration form.

B-Registration form

Create a registration form (Sign-Up), including:

- 3 input fields named “Username”, “Password”, and “Confirm Password”
- 2 submit buttons named “Create” and “Sign-In” that allows players to submit the form or to access to the login form.

C-Password modification form

Create a password modification form, including:

- 3 input fields named “Existing Username”, “New Password”, and “Confirm New Password”
- 2 submit buttons named “Modify” and “Sign-In” that allows players to submit the form or to access to the login form.

D-Guesser game form

Create a game form (Kid Guesser Game), including:

- 5 select fields that allow each to select a number between 0 to 12 in order to guess in advance 5 different numbers the system will randomly generate when the form will be submitted.

- 2 submit buttons named “Submit” and “Sign-Out” that allows players to submit the form or to disconnect and access to the login form.

After the player submits this form, the following 3 messages will be displayed.

1-We generate the numbers a, b, c, d, e

(where a, b, c, d, e are the numbers randomly generated)

2-You guessed the number f, g, h, i, j

(where f, g, h, i, j are the numbers the player guessed and submitted)

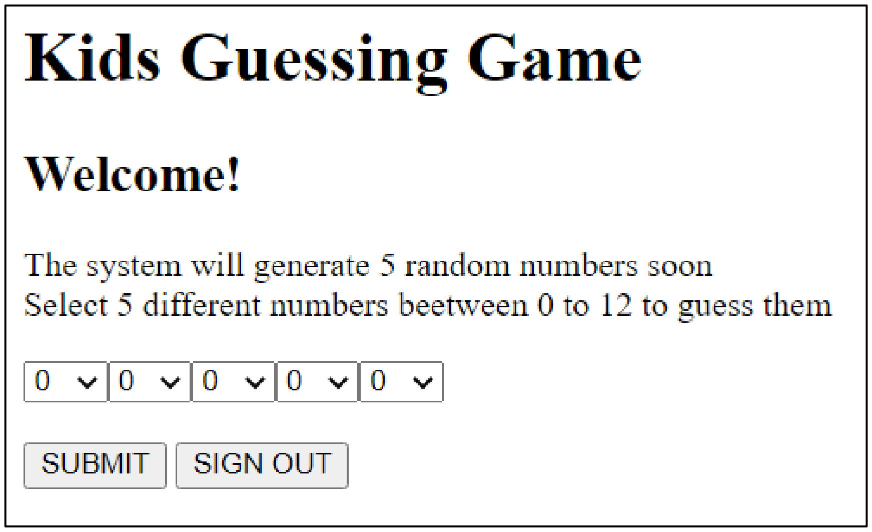
3- One (the most appropriate one) of the 3 sentences below:

1-**Result : You guessed none of the numbers we generated!**
You're an APPRENTICE guesser! Try again!

2-**Result : You guessed x of the numbers we generated!**
You're a GOOD guesser!
Where x is lower than between 1 and 5

3-**Result : You guessed all the numbers we generated!**
You're an EXCELLENT guesser!

See the screenshot below as models and used the indications provided to create the most appropriate interface and text.



Kids Guessing Game

Welcome!

The system will generate 5 random numbers soon
Select 5 different numbers beetween 0 to 12 to guess them

2-TECHNICAL SPECIFICATIONS

Login form

1. When the username and password written by the player within the login form are similar to a pair of username and password already registered in the database, display the guesser game form.

2. When the username written by the player within the login form is not similar to a username already registered in the database, display the message "*You entered a wrong username!*".

3. When the username written by the player within the login form is similar to a username already registered in the database but the password is not similar to the corresponding password already registered in the database, display the message "*You entered a wrong password!*".

4. When the player enters a wrong password display an additional hyperlink or button that allows the player to modify the password.

Registration form

1. When the username written by the player within the registration form is similar to a username already registered in the database, display the message: "This username already exists".

2. When the 2 passwords written by the player within the registration form are not the same, display the message: "You entered 2 different passwords.".

Password modification form

1. Based on the previous indications, decide how this form operates in a way it makes sense.

Database

1. Use the extension mysqli to connect the login and registration form to MySQL to store, check and modify (insert, select, and update) the usernames and passwords.
2. Database name: ACCOUNTS
3. Tables: USERS
4. Columns:

Name	Data type	Constraint	Other properties
ID	INT(5)	PRIMARY KEY	AUTO_INCREMENT
USERNAME	VARCHAR(50)	NOT NULL	
PASSWORD	VARCHAR(50)	NOT NULL	

```sql
CREATE TABLE users(
    userid INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(50) NOT NULL
);
```

Guesser game form

1. Use the super global variable POST to send the form data.
2. Use only 1 HTML select tag (<select><option></option></select>) and a PHP loop (for($i=0; $i<5; ++$i)) to create the 5 select input.
3. Use the built-in function rand() to generate the random numbers.

General Instructions

1. Build your code using and appropriate OOP structure, including class, property, method, and object.
2. Display the error messages within the forms to allow the player to be able to just modify the data entered and submit the form again when applicable.
3. Do not create isolated pages, without buttons or hyperlinks (as indicated) that allows the player to go to another page.
4. Use the built-in functions stripslashes(), strip_tags(), and htmlentities() to sanitize each data received from the forms or the database.

5. You are free to add style (e.g. CSS and Bootstrap) to customize the visual aspect of your tbe pages and components (e.g. forms, text, images…).
6. Create a different file when requires (e.g. one file for each form and form handling).
7. Use lower case letters to write the code, when applicable (e.g. capital for constants).
8. Add indents to your code to make it easy to be reviewed.
9. Add significant comments to your code to make it easy to understand.
10. Test all the functionalities of your program to make sure they work correctly.