

Programa IT Academy – Processo Seletivo – Edição #20

Nome Completo: Arthur Zanella Lovato

E-mail: Arthur.zanella.lovato.1998@gmail.com

Exercício Técnico

RESUMO DA SOLUÇÃO

Em virtude do projeto ter como objetivo simular o comportamento de um software de apostas e sorteios, optou-se pela construção de uma interface gráfica com login e sistema de usuários, onde cada usuário apostador poderá fazer quantas apostas desejar, assim como um usuário administrador será responsável pela inicialização do sorteio.

Para o desenvolvimento utilizou-se as seguintes tecnologias:

- HTML e CSS para a interface gráfica;
- JavaScript para o Front-end, comunicando-se com o HTML e chamando o PHP;
- PHP para o Back-end para lidar com o Banco de Dados;
- MySQL como Banco de Dados Relacional.

TESTES

FUNCIONALIDADE 1 – Iniciar



A primeira interface gráfica que o usuário irá se deparar é com a tela de login.

```
index.php X
index.php > ...
1  <?php
2  // Quando a página é aberta pela primeira vez chama-se o código responsável
3  $mysqli = require __DIR__ . "/PHP/database.php";
4  $resultUserr = $mysqli->query("SELECT 1 FROM userr LIMIT 1");
5  $resultEdition = $mysqli->query("SELECT 1 FROM edition LIMIT 1");
6  if ($resultUserr->num_rows == 0 && $resultEdition->num_rows == 0) {
7      require __DIR__ . "/PHP/beggin.php";
8  }
9  ?>
```

A primeira página a ser carregada será o **index.php**, o trecho apresentado acima tem como objetivo verificar da maneira mais veloz possível se já existe algum registro na tabela *userr* e *edition*, tal velocidade é alcançada em virtude da query “SELECT 1 ... LIMIT 1”, caso não existam dados nelas, significa que o código está sendo carregado pela primeira vez, chamando assim o arquivo **beggin.php**.

```
$mysqli->begin_transaction();

$sql = "INSERT INTO edition (year)
VALUES (?)";

$stmt = $mysqli->stmt_init();
if ( ! $stmt->prepare($sql) ) {
    error_log("Erro no SQL: " . $mysqli->error . " " . $mysqli->errno);
    echo "Erro de envio 7.1";
    exit();
}

$initialYear = 2024;
$stmt->bind_param("i", $initialYear);

if( ! $stmt->execute()){
    $mysqli->rollback();
    error_log("Erro no SQL: " . $mysqli->error . " " . $mysqli->errno);
    echo "Erro de envio 7.2";
    exit();
}

$stmt->close();
```

Nele, o código realiza a inserção da primeira edição de sorteios, começando pelo ano de 2024, junto com algumas precauções no tratamento de erros e nas manipulações de danos no Banco de Dados.

```
// Credenciais padrão para o usuário administrador
$cpf = "00000000000";
$name = "Administrador";
$password_hash = password_hash("senhaforteadmin", PASSWORD_DEFAULT);
$lvl_access = 1;
$stmt->bind_param("sssi", $cpf, $name, $password_hash, $lvl_access);
```

Ainda no **beggin.php**, é possível encontrar as credenciais da conta de administrador, para acessá-la basta inserir no CPF **00000000000** e a senha **senhaforteadmin**, que estará criptografada no Banco de Dados

✓ MySQL não retornou nenhum registro. (A consulta demorou 0,0003 segundos.)

```
select * from userr;
```

☐ Perfil [[Editar em linha](#)] [[Editar](#)] [[Explicar SQL](#)] [[Criar código PHP](#)]

user_id	cpf	name	password_hash	lvl_access	wins
---------	-----	------	---------------	------------	------

✓ MySQL não retornou nenhum

```
select * from edition;
```

☐ Perfil [[Editar em linha](#)] [[Edita](#)

edition_id	year
------------	------

Banco de dados antes de abrir o `index.php` pela primeira vez.

✓ A mostrar registros de 0 - 0 (1 total, A consulta demorou 0,0003 segundos.)

```
select * from userr;
```

☐ Perfil [[Editar em linha](#)] [[Editar](#)] [[Explicar SQL](#)] [[Criar código PHP](#)] [[Atualizar](#)]

☐ Mostrar tudo | Número de registros: 25 | Filtrar registros:

Opções extra

	user_id	cpf	name	password_hash	lvl_access	wins
<input type="checkbox"/> Editar Copiar Apagar	1	00000000000	Administrador	\$2y\$10\$6btQe95tN7NRQIU8JvzDeSWx0VDBM3Wlfgi4MN/xRg...	1	0

✓ A mostrar registros de 0 - 0 (1 total, A consulta demorou 0,0002

```
select * from edition;
```

☐ Perfil [[Editar em linha](#)] [[Editar](#)] [[Explicar SQL](#)] [[Criar código](#)

☐ Mostrar tudo | Número de registros: 25 | Filtrar

Opções extra

	edition_id	year
<input type="checkbox"/> Editar Copiar Apagar	1	2024

Banco de dados após abrir o `index.php` pela primeira vez.

FUNCIONALIDADE 2 – Registrar nova aposta

Registrar-se

Nome:

CPF:

Senha:

Confirmar Senha:

Confirmar

Para usuários apostadores será necessário registrar-se no sistema para poder apostar. O cadastro requisitará ao apostador seu CPF, nome, senha e confirmação de senha.

localhost diz

Erros encontrados:

Campo Usuário é requerido.

Campo CPF é requerido.

Campo Senha é requerido.

OK

Registrar-se

Nome:

CPF:

Senha:

Confirmar Senha:

Confirmar

localhost diz

Erros encontrados:

Senhas devem ser correspondentes.

OK

Registrar-se

Nome:

Arthur

CPF:

12345678900

Senha:

Confirmar Senha:

Confirmar

Caso o usuário apostador deixe algum dos campos em branco ou se a confirmação de senha não for igual a senha, uma mensagem de erro será exibida na tela e impedirá o apostador se cadastrar.

```
if (name.trim() === '') {
    errors.push("Campo Usuário é requerido.");
}

if (cpf.trim() === '') {
    errors.push("Campo CPF é requerido.");
}

if (password.trim() === '') {
    errors.push("Campo Senha é requerido.");
}

if (password !== confirmPassword) {
    errors.push("Senhas devem ser correspondentes.");
}

if (errors.length > 0) {
    alert("Erros encontrados:\n\n" + errors.join("\n"));
    return false;
}

this.submit();
});
```

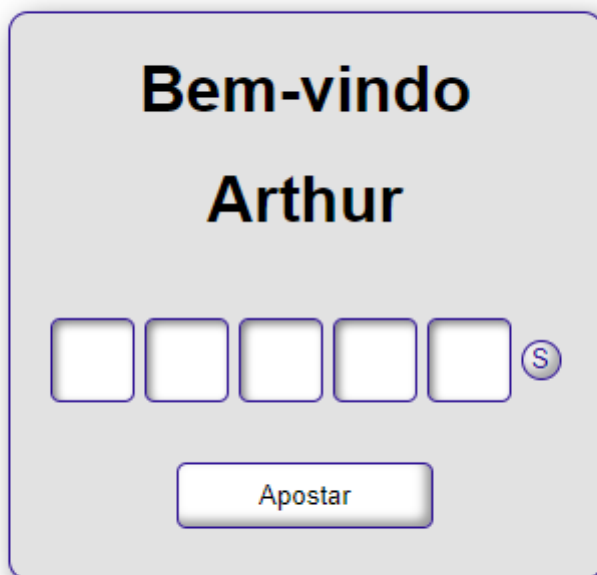
```
// Validações extras de segurança além das existentes no JS
if(empty($_POST["name"])){
    die("Campo Nome é requerido");
}

if(empty($_POST["cpf"])){
    die("Campo CPF é requerido");
}

if(empty($_POST["password"])){
    die("Campo Senha é requerido");
}

if($_POST["confirm-password"] !== $_POST["password"]){
    die("Senhas devem ser correspondentes");
}
```


Os registros passam por duas etapas de validação, uma no JavaScript que intercepta o envio do formulário caso haja algum erro e outra no PHP, que mata a execução do código caso encontre alguma inconsistência. Estes trechos podem ser encontrados respectivamente nos arquivos `signup.js` e `signup.php`.



Bem-vindo

Arthur

Após um usuário apostador se registrar e realizar o login ele será redirecionado para a página de aposta, aqui ele poderá digitar os números que deseja apostar ou clicar no botão “S” na direita para gerar cinco números aleatórios. Por fim, o usuário pode clicar em “Apostar” e confirmar seu palpite.

localhost diz

É necessário preencher os cinco campos

OK

Bem-vindo
Arthur

12

40

S

Apostar

localhost diz

Não podem haver números repetidos

OK

Bem-vindo
Arthur

01

1

48

49

50

S

Apostar

Caso o usuário não insira os cinco números, um alerta será emitido e o apostador será impedido de confirmar seu palpite. Isso também vale para casos em que sejam inseridos números com valores idênticos, por exemplo “01” e “1”.

Bem-vindo
Arthur

999|

Bem-vindo
Arthur

50|

Por fim, ao introduzir um valor fora do intervalo, o código imediatamente o converte para o valor permitido mais próximo.

```
if ( ! formDataCheck.get('n1') ||  
    ! formDataCheck.get('n2') ||  
    ! formDataCheck.get('n3') ||  
    ! formDataCheck.get('n4') ||  
    ! formDataCheck.get('n5')) {  
  
    alert("É necessário preencher os cinco campos");  
    return;  
}
```

```
const nums = [  
    formDataCheck.get('n1'),  
    formDataCheck.get('n2'),  
    formDataCheck.get('n3'),  
    formDataCheck.get('n4'),  
    formDataCheck.get('n5')  
].map(num => String(Number(num)));  
  
// Armazena valores distintos, se o tamanho dele for  
const uniqueNums = new Set(nums);  
if (uniqueNums.size !== nums.length) {  
    alert("Não podem haver números repetidos");  
    return;  
}
```

```
function validateNumber() {  
  
    var inputsNumber = document.querySelectorAll('.number');  
  
    inputsNumber.forEach(function(input) {  
        input.addEventListener('change', function() {  
            // Especifica que value será um número decimal (base 10)  
            var value = parseInt(this.value, 10);  
            // Valores menores que o permitido viram 1, assim como os  
            if (value < 1) {  
                this.value = 1;  
            } else if (value > 50) {  
                this.value = 50;  
            }  
        });  
    });  
}
```

Os trechos referentes à proteção de submissão das apostas e proteção do intervalo dos números apostados são os fornecidos acima. Todos os trechos podem ser encontrados no arquivo **bet.js**, note que apesar do último caso estar sendo inicializado por pela função *validateNumber()*, ela ainda é inicializada pela *DOM*.

A mostrar registos de 0 - 24 (45 total, A consulta demorou 0,0002 segundos.)

select * from bet;

Perfil

[Editar em linha]

[Editar]

[Explicar SQL]

[Criar código PHP]

[Atualizar]

1

>

>>

Mostrar tudo

Número de registos:

25

Filtrar registos:

Pe

Opções extra

bet_id

user_id

registration_id

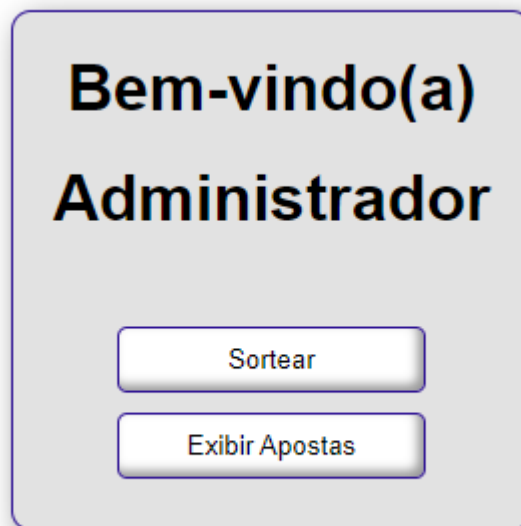
edition_id

number

Acima está um exemplo de como o Banco de Dados ficou após dois usuários apostadores realizarem alguns palpites.

Note que para cada registro de aposta *“registration_id”* há cinco linhas contendo os números da aposta, uma alternativa era criar cinco colunas onde cada uma comportaria um valor apostados, tornado a tabela cinco vezes menor, essa opção não foi escolhida pois aumentaria o nível de complexidade na hora da manipulação dos dados.

FUNCIONALIDADE 3 – Listar apostas



Ao entrar no sistema com a conta administrador haverá duas opções, iniciar um novo sorteio ou exibir as apostas da edição atual. Ao clicar em *“Exibir Apostas”* o operador será redirecionado para a página de exibição.

Registro	Nome	CPF	N1	N2	N3	N4	N5
1000	Arthur	12345678900	16	24	14	2	27
1001	Arthur	12345678900	7	16	12	47	2
1002	Arthur	12345678900	41	27	37	32	17
1003	Arthur	12345678900	42	49	44	39	17
1004	Arthur	12345678900	3	26	7	23	8
1005	Arthur	12345678900	50	3	17	43	36
1006	Maria	98765432100	41	8	32	28	14
1007	Maria	98765432100	42	35	19	38	25
1008	Maria	98765432100	46	13	30	6	18
1009	Maria	98765432100	30	16	9	3	38
1010	Maria	98765432100	35	2	7	16	18
1011	Maria	98765432100	46	11	30	49	47
1012	Maria	98765432100	27	2	25	30	23
1013	Maria	98765432100	20	5	8	3	37
1014	Maria	98765432100	18	26	50	42	7
1015	Maria	98765432100	17	23	5	43	41
1016	Maria	98765432100	28	47	21	8	33
1017	Maria	98765432100	20	25	29	15	43
1018	Maria	98765432100	11	26	10	27	5

Aqui o administrador poderá ter uma visão completa de todas as apostas realizadas para a edição corrente do sorteio.

Registro	Nome	CPF	N1	N2	N3	N4	N5
----------	------	-----	----	----	----	----	----

Note que após realizar o sorteio de uma edição, todos estes registros não mais constaram nesta tabela, dando espaço para as apostas da nova edição, se encontrando dessa maneira logo após um sorteio ser concluído.

[show.js:7](#)

```
▼ Object 1
  ▶ 1: {numbers: Array(5), registration_nr: 1000, name: 'Arthur', cpf:
  ▶ 2: {numbers: Array(5), registration_nr: 1001, name: 'Arthur', cpf:
  ▶ 3: {numbers: Array(5), registration_nr: 1002, name: 'Arthur', cpf:
  ▶ 4: {numbers: Array(5), registration_nr: 1003, name: 'Arthur', cpf:
  ▶ 5: {numbers: Array(5), registration_nr: 1004, name: 'Arthur', cpf:
  ▶ 6: {numbers: Array(5), registration_nr: 1005, name: 'Arthur', cpf:
  ▶ 7: {numbers: Array(5), registration_nr: 1006, name: 'Maria', cpf:
  ▶ 8: {numbers: Array(5), registration_nr: 1007, name: 'Maria', cpf:
  ▶ 9: {numbers: Array(5), registration_nr: 1008, name: 'Maria', cpf:
  ▶ 10: {numbers: Array(5), registration_nr: 1009, name: 'Maria', cpf:
  ▶ 11: {numbers: Array(5), registration_nr: 1010, name: 'Maria', cpf:
  ▶ 12: {numbers: Array(5), registration_nr: 1011, name: 'Maria', cpf:
  ▶ 13: {numbers: Array(5), registration_nr: 1012, name: 'Maria', cpf:
  ▶ 14: {numbers: Array(5), registration_nr: 1013, name: 'Maria', cpf:
  ▶ 15: {numbers: Array(5), registration_nr: 1014, name: 'Maria', cpf:
  ▶ 16: {numbers: Array(5), registration_nr: 1015, name: 'Maria', cpf:
  ▶ 17: {numbers: Array(5), registration_nr: 1016, name: 'Maria', cpf:
  ▶ 18: {numbers: Array(5), registration_nr: 1017, name: 'Maria', cpf:
  ▼ 19:
    cpf: "98765432100"
    name: "Maria"
    ▶ numbers: (5) [11, 26, 10, 27, 5]
    registration_nr: 1018
    ▶ [[Prototype]]: Object
    ▶ [[Prototype]]: Object
  >
```

Se preferir, o administrador poderá observar os resultados pelo console, para a maioria dos navegadores é possível abri-lo pressionando a tecla *F12* e clicando em *console*.

```
$sql = "SELECT bet.*
        , userr.name
        , userr.cpf
        , registration.registration_nr
FROM bet
LEFT JOIN userr
    ON userr.user_id = bet.user_id
LEFT JOIN registration
    ON registration.registration_id = bet.registration_id
WHERE bet.edition_id = (
    SELECT MAX(edition_id)
    FROM edition
)
ORDER BY userr.name";

$stmt = $mysqli->prepare($sql);

if ( ! $stmt->execute() ) {
    error_log("Erro no SQL: " . $mysqli->error . " " . $mysqli->errno);
    echo json_encode(["error" => "Erro de envio 6.1"]);
    exit();
}
```

O trecho acima exibe a query responsável pela seleção dos campos que serão exibidos para o administrador. Os detalhes da manipulação dos dados podem ser encontrados nos arquivos “show.php” e “show.js”.

FUNCIONALIDADE 4 – Finalizar apostas e executar o sorteio

Como mencionado anteriormente, apenas o administrador terá acesso ao botão “Sorteio”, esta é a opção que para a fase de apostas da edição corrente e imediatamente da início a fase de sorteio.

```
// Primeiramente cria um array ordenado de 1 até 50
$orderedNumbers = range(1, 50);
// Bagunça a ordem dos elementos no array
shuffle($orderedNumbers);
// Com as posições randomizadas, seleciona os cinco primeiros, começando do index zero
$drawnNumbers = array_slice($orderedNumbers, 0, 5);
```

O trecho acima mostra as regras para sortear cinco números aleatórios distintos, estas três linhas podem ser encontradas no arquivo **raffle.php**, mais detalhes sobre o processo de sorteio podem ser encontrados em comentários dentro do código.

```
$sql = "INSERT INTO edition (year)
      VALUES (?)";

$stmt = $mysqli->stmt_init();
if ( ! $stmt->prepare($sql) ) {
    error_log("Erro no SQL: " . $mysqli->error . " " . $mysqli->errno);
    echo json_encode(["error" => "Erro de envio 5.7"]);
    exit();
}

$stmt->bind_param("i", $newYear);

if( ! $stmt->execute()){
    $mysqli->rollback();
    error_log("Erro no SQL: " . $mysqli->error . " " . $mysqli->errno);
    echo json_encode(["error" => "Erro de envio 5.8"]);
    exit();
}

$mysqli->commit();
$stmt->close();

echo json_encode($output);
exit();
```

Após a execução do sorteio e a manipulação dos dados que serão retornados para o JavaScript, o código finaliza adicionando uma nova edição de sorteio ao sistema.

```
64
65  /* TESTAR SORTEIO
66  $drawnNumbers = [1, 2, 3, 4, 5];
67  */
68
```

Para fins de testes, pode-se remover o comentário da linha apresentada no arquivo **raffle.php** para forçar os valores sorteados.

✓ A mostrar registos de 0 - 1 (2 total, A consulta demorou 0,0003)

```
select * from edition;
```

☐ Perfil [[Editar em linha](#)] [[Editar](#)] [[Explicar SQL](#)] [[Criar código](#)]

☐ Mostrar tudo | Número de registos: [Filtrar](#)

Opções extra

				edition_id	year			
<input type="checkbox"/>		Editar		Copiar		Apagar	1	2024
<input type="checkbox"/>		Editar		Copiar		Apagar	2	2025

Adicionado uma nova edição ao Banco de Dados logo após o término de um sorteio.

FUNCIONALIDADE 5 – Fim da apuração

```
const drawnNumbersFormatted = data.drawnNumbers.reduce((accumulator, number, index) => {
  // Calcula em que linha os número devem ser colocados (index 0 até 9 é linha 1, index 10 até 19 é
  const currentLine = Math.floor(index / 10);
  // Ao chegar no momento de ter uma nova linha, isso garante que haverá um array disponível
  if ( ! accumulator[currentLine] ) {
    accumulator[currentLine] = [];
  }
  // Adição do zero na frente dos numerais com um dígito para melhor alinhamento na exibição
  const numberFormatted = String(number).padStart(2, '0');
  accumulator[currentLine].push(numberFormatted);
  return accumulator;
}, []).map(line => line.join(', ')).join('\n'); // O array de arrays é mapeado, separando os elementos
document.getElementById('drawn-numbers').innerHTML = drawnNumbersFormatted.replace(/\n/g, '<br>'); //
```

Logo após o sorteio o administrador será redirecionado para uma tela contendo todos os principais dados do sorteio, o arquivo **raffle.js** contém diversos comentários para explicar seu funcionamento, o trecho acima mostra provável manipulação de dados mais complexa do arquivo, responsável por capturar todos os números sorteados e criar dinamicamente o HTML da página.

Número de Sorteios Extras: 25

Número de Ganhadores: 0

Números Sorteados

06, 44, 43, 18, 41, 27, 48, 36, 05, 12
11, 13, 40, 29, 30, 47, 04, 37, 09, 07
42, 32, 33, 19, 22, 01, 03, 23, 14, 24

Vencedores

Sem Vencedores

Estatísticas

Número	Quantidade
7	4
17	4
30	4
2	4
16	4
3	4
27	4
8	4
47	3
41	3
25	3
42	3
26	3

Número de Sorteios Extras: 21**Número de Ganhadores: 1****Números Sorteados**

31, 18, 22, 30, 06, 34, 05, 07, 08, 24
11, 23, 01, 38, 50, 27, 16, 46, 29, 45
44, 15, 40, 37, 33, 35

Vencedores

Registro: 1049

Nome: Arthur

CPF: 12345678900

Estatísticas

Número	Quantidade
17	6
42	6
23	6
21	5
16	5
50	5
41	5
25	5
30	5
20	5
14	5
35	4
13	4

Após o término do sorteio, será exibido na tela os resultados, independente de haver ou não vencedores. Como pode-se notar, o administrador possuirá as informações de **Número de Sorteios Extras**, **Número de Ganhadores**, **Números Sorteados**, **Dados Pessoais dos Vencedores** e **Estatística das Apostas**.

Os Dados Pessoais dos Vencedores exibirão os números de registros das apostas vencedoras, o nome e o CPF dos ganhadores, assim como as Estatísticas das Apostas mostraram a quantidade que cada número foi escolhido para aquela edição do sorteio.

Número de Sorteios Extras: 0

Número de Ganhadores: 2

Números Sorteados

01, 02, 03, 04, 05

Vencedores

Registro: 1060

Nome: Arthur

CPF: 12345678900

Registro: 1061

Nome: Arthur

CPF: 12345678900

Estatísticas

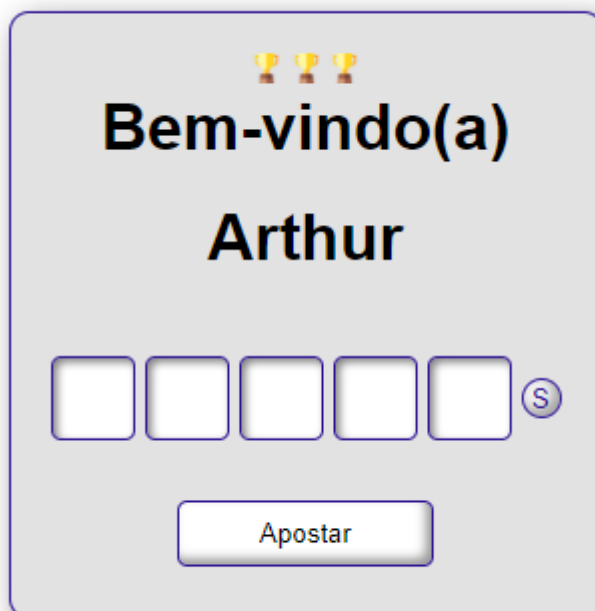
Número	Quantidade
1	2
2	2
3	2
4	2
5	2

Exemplo de resultado com sorteio forçado, feito para testar se um mesmo apostador poderia ganhar duas vezes na mesma edição de sorteio caso fizesse dois palpites iguais.

```
raffle.js:6
▼ Object 1
  countWinners: 0
  ▶ drawnNumbers: (30) [6, 44, 43, 18, 41, 27, 48, 36, 5, 12, 1:
  iteration: 25
  ▶ quantityNumbers: (42) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}
  ▼ winners: Array(0)
    length: 0
    ▶ [[Prototype]]: Array(0)
    ▶ [[Prototype]]: Object
>
```

Assim como na funcionalidade de exibir as apostas correntes, o sorteio também possui a alternativa de analisar os dados via console.

FUNCIONALIDADE 6 – Premiação



Three gold trophy icons

Bem-vindo(a)

Arthur

S

Apostar

Após a realização dos testes com o usuário apostador *Arthur*, houve um total de três vitórias ao decorrer das edições de sorteios, para cada vitória um ícone de troféu é adicionado ao seu perfil pessoal, para um total de no máximo dez troféus.

A mostrar registros de 0 - 4 (5 total, A consulta demorou 0,0003 segundos.)

select * from userr;

Perfil

Editar em linha

Editar

Explicar SQL

Criar código PHP

Atualizar

Mostrar tudo

Número de registros:

25

Filtrar registros:

Pesquisar esta tabela

Ordenar pela chave:

Nenhum

Opções extra

				user_id	cpf	name	password_hash	lvl_access	wins
<input type="checkbox"/>				1	00000000000	Administrador	\$2y\$10\$s.Kaa6PBMMafo1GgFSSwPuTuJ.tLIVMepNjC0D2kTCS...	1	0
<input type="checkbox"/>				2	12345678900	Arthur	\$2y\$10\$0kt6LOAcpwKjTWYBXq9/kes.K3ysTfpgIGK03Q/GbXo...	2	3
<input type="checkbox"/>				3	96325874100	Maria	\$2y\$10\$O9MhnHy6HH3rzcpRG5hECuK6a2epTICK43BLX7GhLvX...	2	0
<input type="checkbox"/>				4	14785236900	Xavier	\$2y\$10\$yYGZAIUIW0Jcw84LQ8sQ.eEY3.2I4YxAA/x6Biw1/pS...	2	0
<input type="checkbox"/>				5	00123456789	Julio	\$2y\$10\$dbOvJ2BLZZzV9uQT2nK0leqdUQR.lpeNeqe96cvRuPg...	2	0

Cada usuário por padrão, inicia sua conta com o valor zero na coluna *wins* da tabela *userr*, para cada nova vitória este valor é incrementado em um.

```
$newWins = $currentWins + 1;

$sqlUpdate = "UPDATE userr
              SET wins = ?
              WHERE cpf = ?";

$stmtUpdate = $mysqli->prepare($sqlUpdate);
if ( ! $stmtUpdate ) {
    error_log("Erro no SQL: " . $mysqli->error . " " . $mysqli->errno);
    echo "Erro de envio 4";
    exit();
}

$stmtUpdate->bind_param("is", $newWins, $cpf);
if ( ! $stmtUpdate->execute() ) {
    $mysqli->rollback();
    error_log("Erro no SQL: " . $mysqli->error . " " . $mysqli->errno);
    echo "Erro de envio 5.5";
    exit();
}

$stmtUpdate->close();
```

Esse valor de vitórias é incrementado no Banco de Dados através do trecho acima, essa parte do código pode ser encontrada no arquivo **raffle.php**.

NOTA – Informações adicionais estão contidas no README

AUTOAVALIAÇÃO

Você concluiu a implementação de 100% das funcionalidades solicitadas?

(X) Sim () Não

Para as 6 principais funcionalidades solicitadas, como você avalia a sua solução?

Marque um 'X'.

	Inexistente/ Insuficiente	Pouco satisfeito(a)	Satisfeito(a)	Muito satisfeito(a)
Funcionalidade 1				X
Funcionalidade 2				X
Funcionalidade 3				X
Funcionalidade 4			X	
Funcionalidade 5			X	
Funcionalidade 6			X	

Principais dificuldades

Como eu já havia feito entregas de projetos utilizando as linguagens descritas neste documento, esse fator não foi um grande problema. Contudo, o já citado arquivo “raffle.js” demandou bem mais tempo do que eu esperava devido a sua complexidade, principalmente com relação aos números sorteados.

A manipulação de um array de arrays feita por um `reduce()` e ainda mapeada por um `map()` levou muito tempo para funcionar, exigindo a interrupção da codificação para se dedicar a pesquisa, no fim a solução foi obtida da maneira mais comum, muitos testes e depurações. Eu estive atribuindo a responsabilidade de criar HTMLS dinâmicos para o arquivo “util.js”, mas devido a alta complexidade que encontrei no “raffle.js” eu resolvi manter toda a responsabilidade nele.

A segunda dificuldade também veio em virtude da complexidade de um arquivo, nesse caso o “bet.php”. Ocorreu que muitas funções estavam sendo atribuídas a ele, tornando-o muito pesado, como tenho mais facilidade em PHP do que JavaScript, separa-lo em três arquivos diferentes não foi uma tarefa tão complexa, pode-se notar que atualmente ele está dividido em “bet-name.php”, “bet-trophy.php” e “bet-process”.

Próximo do fim do projeto notei que estava ocorrendo um bug na inserção no Banco de Dados, quando um usuário apostador concluía um registro, em vez do comportamento esperado dois registros eram inseridos, ambos contendo os mesmos números escolhidos. Esse bug foi algo trivial de resolver, mas era algo tão sutil que

demandou bastante tempo até encontra-lo, exigindo muita depuração. O que estava acontecendo era que o arquivo "bet.js" possui um trecho dentro da DOM que chama o fetch do arquivo PHP, contudo, havia uma função que já foi apagada, sendo chamada pela DOM, está também fazia um fetch no mesmo arquivo PHP, chamando ele duas vezes. Como havia muitos locais onde o erro poderia estar ocorrendo e a função era responsável pela validação dos dados, foi difícil de enxergar a causa do bug.

Por fim, a última dificuldade foi a única que infelizmente não fui capaz de resolver por completo. Pode-se notar que alguns botões de redirecionamento têm certo delay ao clicar, podendo levar de dois a cinco segundos para seu funcionamento ser liberado. Além disso, por vezes é necessário clicar uma ou duas vezes nos botões para ativar suas funcionalidades.

Em partes eu consegui solucionar a demora, originalmente estava levando de quatro a oito segundos para o funcionamento ser liberado, parte disso foi resolvido dividindo funções entre arquivos de processamento, mas mesmo testando tais botões com arquivos praticamente vazios, algum delay ainda estava ocorrendo. Ponderei que pudesse ser questão de cache, mas mesmo testando em navegadores limpos o delay ainda ocorria, tentei remover as tags HTML <a> de dentro das tags <button>, mas os resultados foram inconclusivos, por fim me indaguei se os arquivos de processamento não estavam sendo finalizados, mas mesmo matando os arquivos logo após sua execução o problema permanecia.

Com isso, posso garantir que os botões de redirecionamento funcionam adequadamente, mas talvez seja preciso aguardar alguns segundos para que sejam liberados.

Desempenho Geral

No geral gostei bastante de meu desempenho neste projeto, todas funcionalidades e elementos que eu gostaria de implantar foram atribuídos ao programa com sucesso, a maioria dos bugs que surgiam foram corrigidos rapidamente, os tratamentos de erros e questões de segurança parecem estarem fazendo um papel decente e num contexto geral fiquei satisfeito com a organização, arquitetura e limpeza do código.

Muito me agrada a conclusão desse trabalho, visto que a alguns anos atrás eu havia tentando realizar o desafio técnico da Dell, porém não tem muita coisa que eu possa elogiar de meu desempenho da época, aquele código havia sido desenvolvido apenas em Java, contendo alguns bugs, ausência de interface gráfica e nem todas funcionalidades tinham sido finalizadas. Independente do resultado deste desafio, foi uma grande satisfação conseguir deixar o código exatamente da forma que eu o imaginei quando li ele pela primeira vez.

Obrigado por participar deste processo seletivo.
Salve o documento em PDF com o seu nome.