



**UNIVERSITATEA TEHNICĂ “GH ASACHI” IAȘI FACULTATEA  
AUTOMATICĂ ȘI CALCULATOARE SPECIALIZAREA  
CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI**

**DISCIPLINA BAZE DE DATE**

# **Organizarea concursurilor de matematica din punct de vedere financiar**

**Coordonator,**

Prof. Mironeanu Catalin

**Student,**

Baciu Claudia – 1310A

## Cuprins documentatie

	<b>Titlu capitol</b>	<b>Pagina</b>
1)	Descrierea proiectului .....	3
2)	Descrierea tehnologiilor folosite – Front-end .....	3
3)	Descrierea tehnologiilor folosite – Back-end .....	4
4)	Structura tabelelor .....	4
5)	Inter-relationarea entitatilor .....	5
6)	Aspecte legate de normalizare .....	6
7)	Descrierea constrangerilor .....	7
8)	Conectarea la baza de date .....	8
9)	Operatia de tranzactie .....	9
10)	Capturi de ecran cu interfata aplicatiei .....	10
11)	Link-uri utile .....	13

## 1. Descrierea proiectului

În cadrul concursurilor, utilizarea corectă a fondurilor disponibile furnizate de către sponsori reprezintă o componentă foarte importantă, deoarece folosirea acestora eficient constituie nivelul de calitate a concursului: modul de organizare, facilitățile care sunt oferite participanților, condițiile de examinare și valoarea premiilor castigate de cei merituosi. Toate aceste aspecte pot aduce în viitor popularitate concursului, mai mulți participanți, fiind posibil să fie recunoscut de firme și de universități.

Scopul aplicației este acela de a le oferi persoanelor responsabile cu organizarea concursurilor de matematică (și nu numai) o unealtă utilă, prin intermediul căreia să poată urmări ușor detalii despre evenimentele, participanții și premiile din cadrul unui concurs, dar să producă și modificări la nivel de costuri sau atribuirea premiilor participanților câștigători.

Într-un concurs, principalii factori financiari care trebuie luați în considerare sunt:

- a) Costul fiecărui eveniment din cadrul concursului – mese festive, evenimentul de deschidere, cazare studenți, excursie pentru participanți și profesori, închiriere sală de examen, etc. Fiecare din acestea presupune plăți pentru diferite subprobleme din cadrul evenimentului
- b) Costul fiecărui tip de premiu oferit câștigătorilor – există cazuri în care mai mulți participanți ocupă același loc și atunci este decizia organizatorilor dacă ambii vor primi același premiu (presupune cost în plus pentru achiziționarea unui premiu în plus) sau o altă soluție, în funcție de buget, precum declararea unui nou tip de premiu care valorează jumătate din valoarea inițială și oferit acel premiu ambilor participanți (împartirea premiului la două persoane).
- c) Suma de bani de care dispun organizatorii pentru a pune la punct toate detaliile legate de concurs. Această sumă este provenită din diferite sponsorizări de la firme sau instituții care susțin financiar organizarea concursului.

Suma de bani rămasă după desfășurarea concursului în sine reprezintă profitul. Pentru ca un concurs să fie organizat bine, profitul rămas nu trebuie să fie nici prea mare (înseamnă că potențialul maxim al concursului nu a putut fi atins, întrucât au rămas bani care puteau fi investiți), dar nici foarte mic (în niciun caz negativ), deoarece munca depusă de organizatori nu va putea fi recompensată în totalitate.

## 2. Descrierea tehnologiilor folosite – Front-end

Aplicația este una de tip Web și poate fi folosită de pe orice tip de dispozitiv cu conexiune la internet și un browser. În realizarea părții de Front-end a proiectului s-au utilizat următoarele limbaje:

- a) HTML
- b) CSS
- c) Javascript

Pentru ușurarea realizării interfeței a fost folosit Bootstrap versiunea 4.5.3, un framework gratuit, open-source care utilizează ca limbaj de bază CSS și optional JavaScript

pentru a creea aplicatii web responsive. Bootstrap ofera o multitudine de template-uri pentru butoane, formulare, meniuri de navigare si multe altele. Proiectul a avut ca baza un template de website open-source de tip pagina de administrator. Aceasta poate fi previzualizat si downloandat gratuit de pe link-ul urmator:

<https://startbootstrap.com/template/sb-admin>

Template-ul utilizat a fost ales in principal pentru faptul ca acesta furnizeaza tabele deja programate pentru a facilita cautarea de elemente, sortarea elementelor si dispunerea inregistrarilor pe pagini, dar si pentru faptul ca ofera moduri de a afisa informatiile sub forma de chart.

Interfata este una usor de folosit, intrucat orice pagina este la putine click-uri distanta una fata de cealalta, iar operatiile pe elemente sunt intuitive.

### 3. Descrierea tehnologiilor folosite – Back-end

Pentru partea de back-end a aplicatiei, a fost utilizat un server Apache HTTP v3.2.4 prin intermediul programului Xampp, v 7.3.25. Principalul limbaj in care a fost realizata aplicatia este PHP v7.3.25. Am ales acest limbaj deoarece se potriveste pentru proiectele web unde informatiile sunt raspandite in mai multe pagini si se utilizeaza formulare care sunt completate si trimise pe alte pagini prin metode de tip POST si GET, dar si pentru ca este foarte usor de integrat in codul html al unei pagini.

Pentru utilizarea bazei de date, am instalat extensia oci8 pentru limbajul PHP.

- a) Instalarea acestuia pe masinile cu sistem de operare Linux se face utilizand comanda: `pecl install oci8-2.2.0`.
- b) Instalarea pe masina care ruleaza Windows OS, se gaseste urmarind tutorialul aflat la link-ul urmator:

<https://gist.github.com/jeanne007/ffe6db1b6d539544d634006d25aa32c3>

Extensia ofera o multitudine de functii prin intermediul carora se poate comunica cu o baza de date de tip Oracle precum: `oci_connect` (pentru conectare la baza de date), `oci_parse` si `oci_execute` (pentru a rula comenzi de tip sql), posibilitate de rulare folosind statements. Accesul se face prin intermediul unui client Oracle.

### 4. Structura tabelor

Informatiile de care avem nevoie in aplicatie sunt cele legate de :

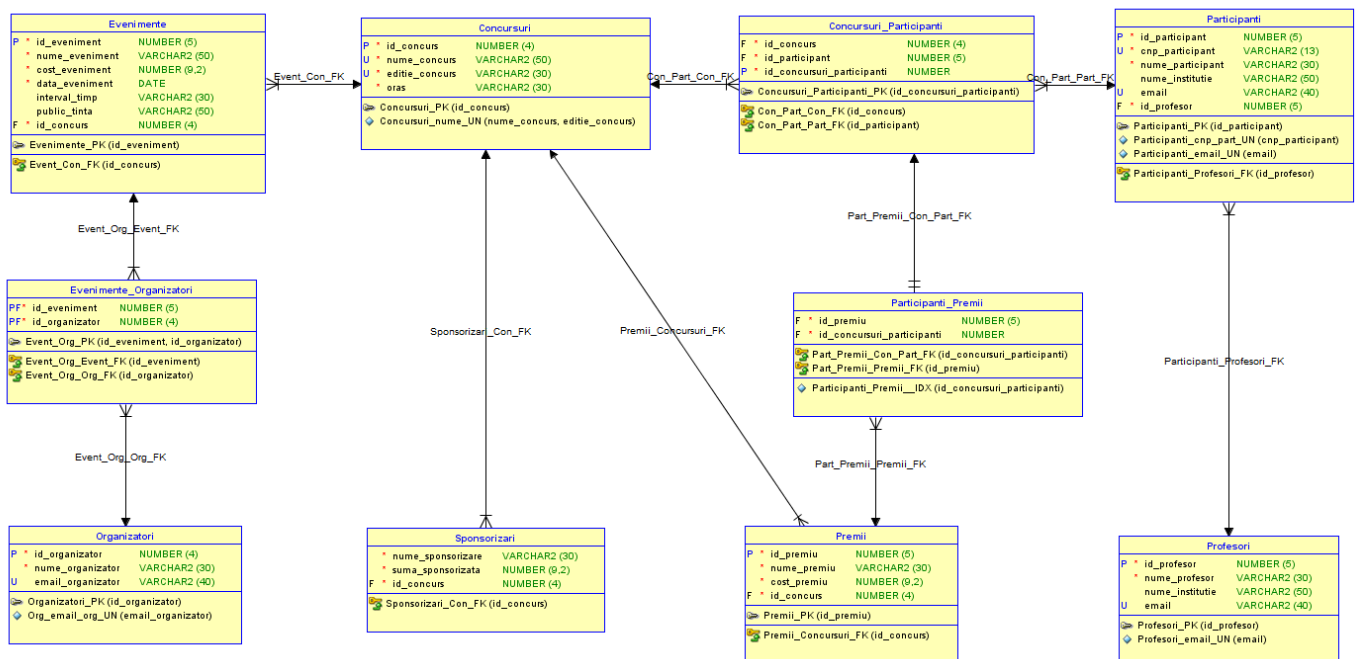
- **echipa de organizatori:** ne intereseaza sa stim care sunt persoanele responsabile cu planificarea concursurilor, astfel incat sa existe posibilitatea de a le asina fiecaruia un set de evenimente de care sa se ocupe.
- **evenimentele:** sunt componentele de baza din care este organizat un concurs, fiecaruia fiindu-I asociat un cost, data cand va avea loc, intervalul de timp, grupul tinta si denumirea acestuia.
- **concursurile:** reprezinta punctul central al proiectului, fiecare avand o lista de evenimente asociata, o lista de premii care pot fi obtinute, un numar de sponsorizari si o lista de participanti impreuna cu profesorul lor insotitor. Acestea pot avea loc in mai multe editii, in diferite orase.

- **premiile:** sunt in bani sau obiecte in functie de locul obtinut de catre participant la concursul corespunzator, avand un cost proportional cu locul obtinut asociat.
- **sponsorizarile:** firmele partenere sau diferite institutii sustin financiar concursurile, reprezentand o sursa importanta de bani care vor fi depuse pentru derularea in siguranta a concursului (premii si evenimente).
- **participantii:** sunt elevi sau student care decid sa se inscrie la diferite concursuri. Acestia beneficiaza de evenimentele din concurs, dar si de premiile oferite
- **profesorii insotitori:** fiecare participant are un profesor insotitor care participa de asemenea la evenimente, si este responsabil de acestia.

Tabelele care s-au utilizat pentru a descrie atat entitatile enumerate mai sus, cat si relatiile dintre acestea sunt urmatoarele:

- **Organizatori**
- **Evenimente**
- **Concursuri**
- **Sponsorizari**
- **Premii**
- **Participantii**
- **Profesori**
- **Concursuri\_Participanti** – precizeaza la ce concursuri participa fiecare competitor
- **Participantii\_Premii** – pentru fiecare pereche participant castigator de premiu – concurs ii va atribui un premiu
- **Evenimente\_Organizatori** – asigura relatia de multi la multi dintre entitatea Evenimente si entitatea Organizatori

Diagrama E-R a bazei de date arata in felul urmatoar



## 5. Inter-relationarea entitatilor

In proiectarea acestei baze de date s-au identificat urmatoarele tipuri de relatii: 1:1, 1:n, n:1 si n:m.

### a) Relatii 1:n

- Intre tabelele **concursuri** si **evenimente**, deoarece orice concurs are in cadrul sau organizate mai multe evenimente, dar un eveniment este specific doar pentru un anumit concurs si nu exista concursuri care sa impartaseasca acelasi eveniment.
- De asemenea, un concurs beneficiaza de mai multe sponsorizari, astfel intre tabelele **concursuri** si **sponsorizari** este o relatie de 1:n, dat fiind ca o sponsorizare reprezinta o actiune de finantare din partea unei firme sau institutii pentru un singur concurs.
- Pentru participantii inscrisi la concurs care au obtinut locuri mari in clasament, concursul pune la dispozitie mai multe tipuri de premii, fiecare avand atribuite un cost specific. Astfel, intre tabelele **concursuri** si **premii** exista o relatie de 1:n. Din lista tipurilor de premii existente in cadrul unui concurs, organizatorii vor oferi castigatorilor o parte din ele sau, in unele cazuri, pot atribui de doua ori acelasi premiu participantilor care ocupa acelasi loc.
- Intre tabelele **premii** si **participanti** exista indirect o relatie de tip n:m justificata de ideea ca un participant poate castiga mai multe premii din cadrul diferitelor concursuri, iar un premiu poate fi oferit mai multor oameni castigatori. Aceasta relatie este modelata prin intermediul relatiei 1:n dintre participanti si participanti\_premii, a doua reprezentand o entitate care leaga o pereche (concurs, participant) de premiul primit.
- Alta relatie din proiect reprezinta cea dintre **profesori** si **participanti**. Orice participant are ca insotitor un cadru didactic. Acesta are responsabilitatea de a se asigura de siguranta participantilor insotiti. Cum un profesor poate insotii mai multi participanti, relatia dintre cele doua tabele este de 1:n.

### b) Relatii m:n

- Evenimentele sunt structurate si puse la punct de o echipa de mai multi organizatori care isi impart atributiile pentru a gestiona diferite parti ale concursului. Astfel intre tabelele **evenimente** si **organizatori** exista o relatie de n:m, astfel incat un eveniment poate fi organizat de mai multi **organizatori**, iar un organizator poate participa la aranjarea mai multor **evenimente**. Aceasta relatie este simulata prin intermediul tabelii **evenimente\_organizatori**.
- Intre tabelele **concursuri** si **participanti** exista o relatie de n:m, deoarece un concurs se desfasoara cu mai multi participanti, iar de-a lungul timpului, un participant poate sa se inscrie la mai multe concursuri. Legatura dintre cele doua tabele este simulata prin intermediul tabelii **concursuri\_participanti**. Aceasta pe langa cele doua chei straine din tabelele **concursuri**, respectiv **participanti**, prezinta si un id specific fiecarei perechi cu rol de cheie primara.

### c) Relatii 1:1

- Intre tabelele **concursuri\_participanti** si **participanti\_premii** este stabilita o relatie de 1:1. Problema care a intervenit in proiect si a dus la luarea acestei decizii a fost faptul ca daca se realiza direct conectiunea dintre premii si participanti ca relatie de tip n:m, ducea la posibilitatea ca in sistem sa existe persoane castigatoare de premii de la concursuri la care n-au participat. Astfel, prin configuratia curenta, ne asiguram ca doar participantii de la un anumit concurs vor beneficia de premiile oferite de concurs, in cazul in care obtin un loc semnificativ.

## 6. Aspecte legate de normalizare

Baza de date a fost normalizata, deoarece indeplinesc urmatoarele conditii:

- a) Toate tabele respecta conditiile primei forma normale:
  - un atribut contine valori atomice din domeniul sau (si nu grupuri de astfel de valori)
  - nu contine grupuri care se repeta
- b) Toate tabelele respecta conditiile celei de a doua forme normale:
  - este prima forma normala
  - toate attributele non-cheie depind de toate cheile candidat
- c) Toate tabelele respecta a treia forma normala:
  - este in a doua forma normala
  - toate attributele non-cheie sunt direct (non-tranzitiv) dependente de toate cheile candidat.

## 7. Descrierea constrangerilor

Pentru a evita introducerea datelor eronate de catre utilizatori, in cadrul bazei de date s-au stabilit diferite constrangeri:

### a) Constrangeri de tip Check

Acest tip de constrangere a fost folosit pentru a nu permite introducerea de costuri negative pentru entitatile: **evenimente** (atributul *cost\_eveniment*), **premii** (atributul *cost\_premiu*) si **sponsorizari** (atributul *suma\_sponsorizata*).

O alta utilizare a constrangerilor de tip Check este pentru a asigura ca datele sunt introduse intr-un format specific:

- pentru verificarea email-ului in tabelele **organizatori** (atributul *email\_organizator*), **participant** (atributul *email*) si **profesori** (atributul *email*)
- pentru corectitudinea formatului cnp-ului unui participant in tabela **participant** (atributul *cnp\_participant*)
- pentru a verifica faptul ca numele persoanelor contin doar litere si spatii in tabelele **organizatori** (*nume\_organizator*), **participanti** (*nume\_participant*) si tabela **profesori** (*nume\_profesor*)

#### b) Constrangeri de tip NOT NULL

Acest tip de constrangere a fost folosit pentru a specifica campurile esentiale pentru o entitate. Pentru ca majoritatea atributelor din toate entitatile respecta acest tip de constrangere, le voi specifica doar pe cele care nu sunt esentiale:

- Tabelele **organizatori** (*email\_organizator*), **evenimente** (*interval\_timp*, *public\_tinta*), **participanti** (*nume\_institutie*, *email*) si **profesori** (*nume\_institutie*, *email*) contin campuri care nu sunt esentiale, descriptive.

#### c) Constrangeri de tip Primary Key

Fiecare tabel in parte (mai putin **sponsorizari**) prezinta un camp de tip index cu auto-incrementare care respecta o constrangere de tip Primary Key.

#### d) Constrangeri de tip Unique

Inafara de acele attribute care sunt Primary Key si respecta implicit constrangerea de tip Unique, in proiect exista mai multe attribute care este necesar sa fie unic: pentru tabela **participanti**, atributul *cnp\_participant*, attributele de tip *email* pentru tabelele **organizatori**, **participanti**, **profesor** si pentru tabela **concursuri** o cheie unica compusa din attributele *nume\_concurs* si *editie\_concurs*.

#### e) Constrangeri de tip Foreign Key

Aceste constrangeri sunt utilizate pentru a modela diferite tipuri de relatii dintre tabele:

- Relatii **1:1** in **participanti\_premii** (*id\_concursuri\_participanti*)
- Relatii **1:n** in **premii** (*id\_concurs*), **evenimente** (*id\_concurs*), **participanti** (*id\_profesor*)
- Relatii **n:m** in tabelele **concursuri\_participanti** (*id\_concurs*, *id\_participant*), **evenimente\_organizatori** (*id\_eveniment*, *id\_organizator*)

## 8. Conectarea la baza de date

Conectiunea cu baza de date este realizata prin intermediul functiilor puse la dispozitie de catre extensia php numita oci8. Acesta ofera functia de conectare la baza de date *oci\_connect* care primeste ca parametrii utilizatorul, parola si numele bazei de date. Deoarece este utilizata o baza de date Oracle locala prin intermediul Oracle Instant Client, la numele bazei de date se va specifica „localhost/XE”.

In proiectul realizat, s-a creat un fisier dedicat operatiilor cu baza de date (conectare, selectie, update, inserare, stergere), care poate fi gasit la adresa */bd\_proiect/dashboard/db\_connection.php*. Acest fisier contine o singura clasa numita „db\_connection” care este responsabila cu tot ce tine de baza de date. La apelarea constructorului, se va realiza conectarea la baza de date, iar la apelul destructorului, se va face deconectarea de la baza de date.

Functiile membre ale clasei se impart in 2 mari categorii: functii generale si functii de realizare comenzi sql.

Din cadrul functiilor generale fac parte functiile:



- `add_param_query` (primește ca parametru un dictionar și adaugă unei comenzi de selectie sql o clauza WHILE care include toate elementele din dictionar) cu rol în a spori flexibilitatea apelării comenzilor sql.
- `execute_select` (primește ca parametru un string care reprezintă o comandă sql) cu rol în a rula toate comenzile de sql și de a returna un vector cu obiecte care reprezintă rezultatul căutării.
- `execute_query` (primește ca parametru un string care reprezintă o comandă sql) cu rol în realizarea comenzilor de tip update, returnând dacă a fost executată sau nu comanda.

Funcțiile care realizează comenzi sql sunt următoarele:

- **Comenzi de selectare a datelor:** `select_concursuri`, `select_evenimente`, `select_optiuni_premii`, `select_sponsorizari`, `select_participanti_concurs_premiu`
- **Comenzi de evaluarea unor sume totale:** `cost_premii_oferite`, `cost_evenimente`, `suma_sponsorizari`
- **Comenzi de numarare a inregistrarilor:** `numar_participanti_concurs`, `numar_total_participanti`
- **Comenzi de update:** `update_cost_eveniment`, `update_cost_premiu`, `update_relatie_premiu_participant`
- **Comenzi de inserare:** `insert_premiu`, `insert_asociere_participant_premiu`
- **Comenzi de stergere:** `delete_premiu_transaction`, `delete_relatie_participant_premiu`

## 9. Operatia de tranzactie

În cadrul proiectului a fost implementată o tranzacție pentru stergerea unui anumit tip de premiu pentru un concurs. Fiecare concurs dispune de mai multe opțiuni de premii pe care le pot oferi castigatorilor, pe unele dintre ele în mai multe exemplare, sau renunțând la altele complet. În cazul în care organizatorul dorește să steargă un tip de premiu din baza de date, este important ca înainte de a executa stergerea aceasta, să steargă toate asocierile premiului în cauză cu participanții (din tabela **participanti\_premii**, vor fi șterse toate acele înregistrări unde `id_premiu` este premiul aflat în stergere). După această operație, este eligibilă stergerea tipului de premiu.

Mentionez că, deși există posibilitatea setării unei proprietăți a relației dintre tabelele **premiu** și **participanti\_premii** ca să reacționeze în felul descris anterior, aceasta reprezintă o oportunitate de a folosi o tranzacție în proiect.

Funcția apelată pentru realizarea tranzacției execută înainte de aceasta în plus o operație de selectie, pe baza căreia decide dacă execută sau nu stergerea în funcție de existența premiului sau nu. Scopul acestei funcții suplimentare este cel de afișare a erorilor corespunzătoare.

În cazul în care apare o problemă între executia primei și celei de a doua instrucțiuni, se va executa un rollback prin intermediul funcției puse la dispoziție de extensia OCI8 numite `oci_rollback()`.

Implementarea funcției o veți găsi în figura de mai jos.

```

301 public function delete_premiu_transaction($id_premiu, $id_concurs='-1'){
302     // obs: parametrul $id_concurs are rol de verificare, pentru a nu sterge alte premii
303     $params = array(
304         'id_premiu' => $id_premiu
305     );
306     if($id_concurs != '-1')
307         $params['id_concurs'] = $id_concurs;
308
309     // daca nu exista id-ul respectiv
310     $sql_command = "SELECT id_premiu FROM premii";
311     $sql_command .= $this->add_param_query($params);
312     $exista = $this->execute_select($sql_command);
313     if(count($exista) != 1){
314         //echo "Eroare[$id_premiu]: nu exista premiul selectat!";
315         return "Eroare[$id_premiu]: nu exista premiul selectat!";
316     }
317
318
319     // sterge peste tot unde participantii au premiu de id dat
320     $sql_command = "DELETE FROM participanti_premii WHERE id_premiu = '$id_premiu'";
321     $stmt = oci_parse($this->conn, $sql_command);
322     $r = oci_execute($stmt, OCI_NO_AUTO_COMMIT);
323
324     if(!$r){
325         //echo "Eroare[$id_premiu]: nu s-a realizat stergerea participanti_premii!";
326         return "Eroare[$id_premiu]: nu s-a realizat stergerea participanti_premii!";
327     }
328
329     $sql_command = "DELETE FROM premii WHERE id_premiu = '$id_premiu'";
330     $stmt = oci_parse($this->conn, $sql_command);
331     $r = oci_execute($stmt, OCI_NO_AUTO_COMMIT);
332     if(!$r){
333         oci_rollback($this->conn); // rollback changes to both tables
334         //echo "Eroare[$id_premiu]: nu s-a realizat stergere premiului!";
335         return "Eroare[$id_premiu]: nu s-a realizat stergere premiului!";
336     }
337
338     // Commit the changes to both tables
339     $r = oci_commit($this->conn);
340     oci_free_statement($stmt);
341     return "Success[$id_premiu]!";
342 }

```

## 10. Capturi de ecran cu interfata aplicatiei

### 1. Pagina tabel concursuri

Organizatori Concursuri

Search for...

MAIN PAGE

- Dashboard
- VIEW
- Concursuri
- Participanti

Logged in as: Organizer

## Concursuri

Dashboard / Tabela Concursuri

In aceasta pagina, puteti urmari toate concursurile care au fost organizate.

Tabela Concursuri

Show 10 entries

Nume Concurs	Editie Concurs	Oras	Numar Participanti
SEEMOUS	Editia X	Devin, Bulgaria	5 (Vezi Participanti)
Traian Lalescu	Editia XIII	Cluj	4 (Vezi Participanti)
Nume Concurs	Editie Concurs	Oras	Numar Participanti

Showing 1 to 2 of 2 entries

Previous 1 Next

(pagina concursuri\_tabele.php)

```
// returneaza toate campurile concursurilor
public function select_concursuri($stmt_param = array()){
    $sql_command = "SELECT * FROM concursuri";

    // adaugam parametrii
    $sql_command .= $this->add_param_query($stmt_param);

    return $this->execute_select($sql_command);
}
```

(script selecte concursuri)

```
// returneaza numarul de participanti pentru un anumit concurs
public function numar_participanti_concurs($id_concurs){
    $sql_command = "SELECT COUNT(*) AS numar_participanti FROM concursuri_participanti";

    // adaugam parametrii
    $sql_command .= $this->add_param_query(['id_concurs' => $id_concurs]);

    // returnam numarul de participanti
    $numar_participanti = $this->execute_select($sql_command);
    if(count($numar_participanti) == 0)
        return 0;

    return $numar_participanti[0]->NUMAR_PARTICIPANTI;
}
```

(script numarare participanti concurs)

## 2. Pagina detalii concurs

The screenshot displays the 'Organizatori Concursuri' web application. The main header includes a search bar and a user profile icon. The left sidebar shows navigation options: 'MAIN PAGE' (Dashboard), 'VIEW' (Concursuri, Participanti), and 'Logged in as: Organizator'. The main content area is titled 'Concursul "Traian Lalescu" - Editia XIII' and shows details for this specific competition. It includes a search bar for the competition name, a dropdown menu for the city (Cluj), and a button to view participants. Below this, there are four summary cards: 'Cost Premii' (650, 8.13%), 'Cost Evenimente' (5875.75, 73.45%), 'Suma Totala Sponsorizari' (8000, 100.00%), and 'Total Profit' (1474.25). A bar chart titled 'Sponsorizari Concurs' shows the number of sponsors for each category: ULLINK, blue Projects, Betfair, Fundatia EMAG, and BCR. The chart is updated yesterday at 11:59 PM. The footer includes copyright information and links to Privacy Policy and Terms & Conditions.

(pagina *detalii\_concurs.php*)

```

162 ▼ public function cost_premii_oferite($id_concurs){
163 ▼     $sql_command = "
164 ▼         SELECT SUM(p.cost_premiu) cost_total_premii
165             FROM concursuri_participanti cp JOIN participanti_premii pp ON
166             cp.id_concursuri_participanti = pp.id_concursuri_participanti
167             JOIN premii p ON
168             p.id_premiu = pp.id_premiu
169             WHERE cp.id_concurs = ".$id_concurs."
170     ";
171     $r = $this->execute_select($sql_command);
172     if(count($r) == 0)
173         return 0;
174     return $r[0]->COST_TOTAL_PREMII;
175 }
176
177
178 // cost total evenimente
179 ▼ public function cost_evenimente($id_concurs){
180 ▼     $sql_command = "
181         SELECT SUM(cost_eveniment) cost_total_evenimente FROM evenimente
182         WHERE id_concurs = ".$id_concurs."
183     ";
184     $r = $this->execute_select($sql_command);
185     if(count($r) == 0)
186         return 0;
187     return $r[0]->COST_TOTAL_EVENIMENTE;
188 }
189
190 // suma totala sponsorizari
191 ▼ public function suma_sponsorizari($id_concurs){
192 ▼     $sql_command = "
193         SELECT SUM(suma_sponsorizata) suma_totala FROM sponsorizari
194         WHERE id_concurs = ".$id_concurs."
195     ";
196     $r = $this->execute_select($sql_command);
197     if(count($r) == 0)
198         return 0;
199     return $r[0]->SUMA_TOTALA;
200 }

```

(scripturi calcul costuri premii, evenimente si suma sponsorizata)

## 11. Link-uri utile

- 1) Link template Bootstrap  
<https://startbootstrap.com/template/sb-admin>
- 2) Referinte Bootstrap  
<https://getbootstrap.com/docs/4.0/getting-started/introduction/>
- 3) Link instalare server Apache XAMPP  
<https://www.apachefriends.org/download.html>
- 4) Comanda instalare extensie oci8 pentru PHP pentru Linux  
pecl install oci8-2.2.0
- 5) Link-uri tutoriale instalare extensie oci8 pentru PHP pentru Windows 10  
<https://gist.github.com/jeanne007/ffe6db1b6d539544d634006d25aa32c3>  
<https://www.youtube.com/watch?v= CNM6ie-PwQ>
- 6) Referinte limbaj PHP  
<https://www.php.net/manual/en/langref.php>