**UNIVERSITATEA TEHNICĂ “GH ASACHI” IAŞI FACULTATEA AUTOMATICĂ ŞI CALCULATOARE SPECIALIZAREA CALCULATOARE ŞI TEHNOLOGIA INFORMAŢIEI**

**DISCIPLINA BAZE DE DATE**

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

**Coordonator,**

Prof. Mironeanu Catalin

**Student,**

Baciu Claudia – 1310A

**Cuprins documentatie**

**Titlu capitol Pagina**

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
12. **Descrierea proiectului**

In cadrul concursurilor, utilizarea corecta a fondurilor disponibile furnizate de catre sponsori repezinta o componenta foarte importanta, deoarece folosirea acestora eficient costituie nivelul de calitate a concursului: modul de organizare, facilitatile care sunt oferite participantilor, conditiile de examinare si valoarea premiilor castigate de cei merituosi. Toate aceste aspecte pot aduce in viitor popularitate concursului, mai multi participanti, fiind posibil sa fie recunoscut de firme si de universitati.

Scopul aplicatiei este acela de a le oferi persoanelor responsabile cu organizarea concursurilor de matematica (si nu numai) o unealta utila, prin intermediul careia sa poata urmari usor detalii despre evenimentele, participanti si premiile din cadrul unui concurs, dar sa produca si modificari la nivel de costuri sau atribuirea premiilor participantilor castigatori.

Intr-un concurs, principalii factori financiari care trebuie luati in considerare sunt:

1. Costul fiecarui eveniment din cadrul concursului – mese festive, evenimentul de deschidere, cazare studenti, excursie pentru participanti si profesori, inchiriere sala de examen, etc. Fiecare din acestea presupun plati pentru diferite subprobleme din cadrul evenimentului
2. Costul fiecarui tip de premiu oferit castigatorilor – exista cazuri in care mai multi participanti ocupa acelasi loc si atunci este decizia organizatorilor daca ambii vor primi acelasi premiu (presupune cost in plus pentru achizitionarea unui premiu in plus) sau o alta solutie, in functie de buget, precum declararea unui nou tip de premiu care valoreaza jumatate din valoarea initiala si oferit acel premiu ambilor participanti (impartirea premiului la doua persoane).
3. Suma de bani de care dispun organizatorii pentru a pune la punct toate detaliile legate de concurs. Aceasta suma este provenita din diferite sponsorizari de la firme sau institutii care sustin financiar organizarea concursului.

Suma de bani ramasa dupa desfasurarea concursului in sine reprezinta profitul. Pentru ca un concurs sa fie organizat bine, profitul ramas nu trebuie sa fie nici prea mare (inseamna ca potentialul maxim al concursului nu a putut fi atins, intrucat au ramas bani care puteau fi investiti), dar nici foarte mic (in nicun caz negativ), deoarece munca depusa de organizatori nu va putea fi recompensata in totalitate.

1. **Descrierea tehnologiilor folosite – Front-end**

Aplicatia este una de tip Web si poate fi folosita de pe orice tip de dispozitiv cu conectiune la internet si un browser. In realizarea partii de Front-end a proiectului s-au utilizat urmatoarele limbaje:

1. HTML
2. CSS
3. Javascript

Pentru usurinta realizarii interfetei a fost folosit Bootstrap versiunea 4.5.3, un framework gratuit, open-source care utilizeaza ca limbaj de baza CSS si 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.

1. **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.

1. Instalarea acestuia pe masinile cu sistem de operare Linux se face utilizand comanda: pecl install oci8-2.2.0.
2. 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\_exectute (pentru a rula comenzi de tip sql), posibilitate de rulare folosind statements. Accesul se face prin intermediul unui client Oracle.

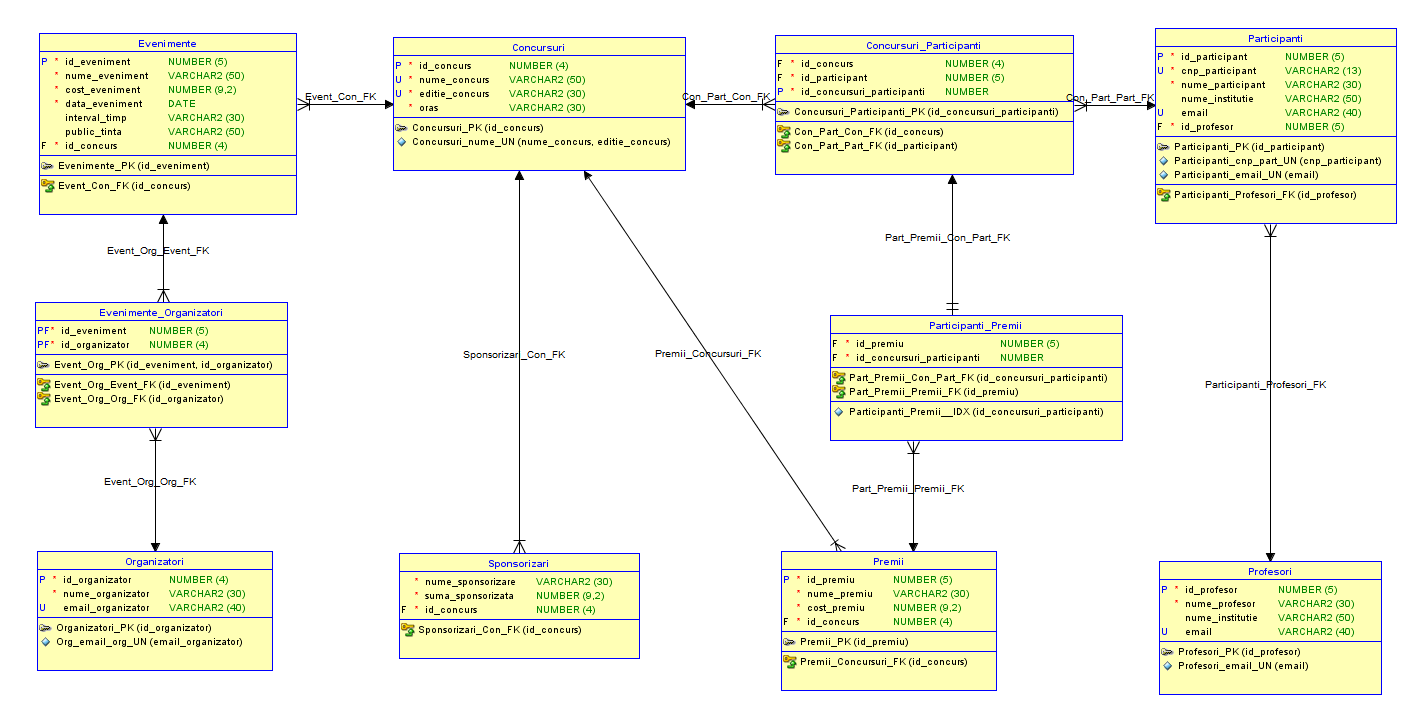
1. **Structura tabelelor**

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 asigna 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 insistutii 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**
* **Sposorizari**
* **Premii**
* **Participanti**
* **Profesori**
* **Concursuri\_Participanti** – precizeaza la ce concursuri participa fiecare competitor
* **Participanti\_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 urmator

1. **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.

1. **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 beneficieaza 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 oferii 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.

1. **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 eveminente 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 simulate prin intermediul tabelei **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 simulate prin intermediul tabelei 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.

1. **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.

1. **Aspecte legate de normalizare**

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

1. Toate tabele respecta conditiile primei forma normale:

* un atribut contine valori atomica din domeniul sau (si nu grupuri de astfel de valori)
* nu contine grupuri care se repeta

1. Toate tabelele respecta contitiile celei de a doua forme normale:

* este prima forma normala
* toate atributele non-cheie depind de toate cheile candidat

1. Toate tabelel respecta a treia forma normala:

* este in a doua forma normala
* toate atributele non-cheie sunt direct (non-tranzitiv) dependente de toate cheile candidat.

1. **Descrierea constrangerilor**

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

1. 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 persoanlor contin doar litere si spatii in tabelele **organizatori** (*nume*\_*organizator*), **participanti** (nume\_*participant*) si tabela **profesori** (*nume*\_*profesor*)

1. 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.

1. 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.

1. Constrangeri de tip Unique

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

1. 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*)

1. **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 descructorului, 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 (primeste ca parametru un dictionar si adauga unei comenzi de selectie sql o clauza WHILE care include toate elementele din dictionar) cu rol in a spori flexibilitatea apelarii comenzilor sql.
* execute\_select (primeste ca parametru un string care reprezinta o comanda sql) cu rol in a rula toate comenzile de sql si de a returna un vector cu obiecte care reprezinta rezultatul cautarii.
* execute\_query (primeste ca parametru un string care reprezinta o comanda sql) cu rol in realizarea comenzilor de tip update, returnand daca a fost executata sau nu comanda.

Functiile care realizeaza comenzi sql sunt urmatoarele:

* **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

1. **Operatia de tranzactie**

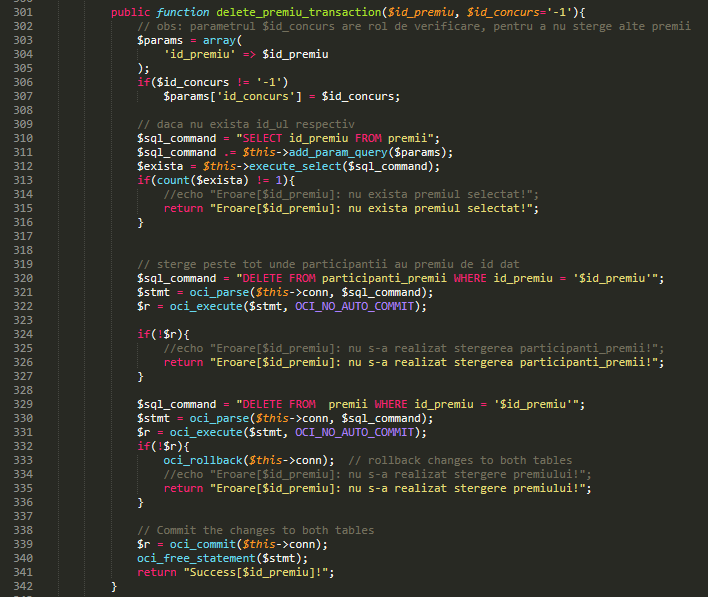
In cadrul proiectului a fost implementata o tranzactie pentru stergerea unui anumit tip de premiu pentru un concurs. Fiecare concurs dispune de mai multe optiuni de premii pe care le pot oferii castigatorilor, pe unele dintre ele in mai multe exemplare, sau renuntand la altele complet. In cazul in care organizatorul doreste sa stearga un tip de premiu din baza de date, este important ca inainte de a executa stergerea aceasta, sa stearga toate asocierile premiului in cauza cu participantii (din tabela **participanti\_premii**, vor fi sterse toate acele inregistrari unde *id\_premiu* este premiul aflat in stergere). Dupa aceasta operatie, este eligibila stergerea tipului de premiu.

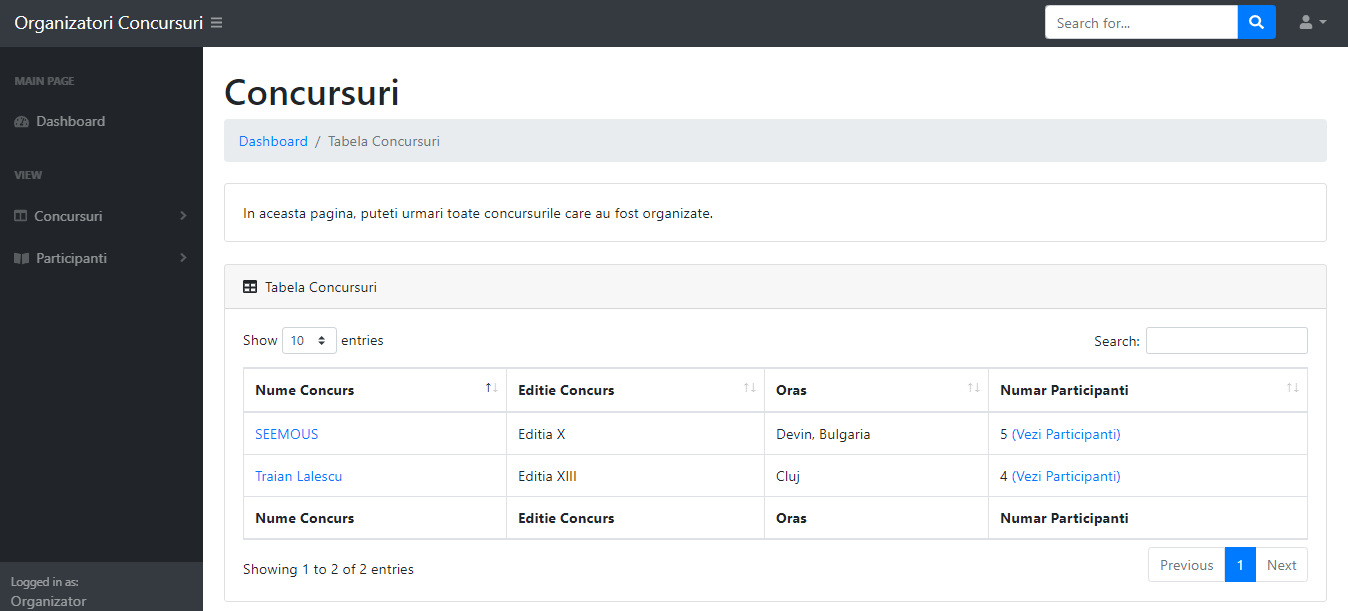
Mentionez ca, desi exista posibilitatea setarii unei proprietati a relatiei dintre tabelele **premii** si **participanti\_premii** ca sa reactioneze in felul descris anterior, aceasta reprezinta o oportunitate de a folosi o tranzactie in proiect.

Functia apelata pentru realizarea tranzactiei executa inainte de aceasta in plus o operatie de selectie, pe baza careia decide daca executa sau nu stergerea in functie de existenta permiului sau nu. Scopul acestei functii suplimentare este cel de afisare a erorilor corespunzatoare.

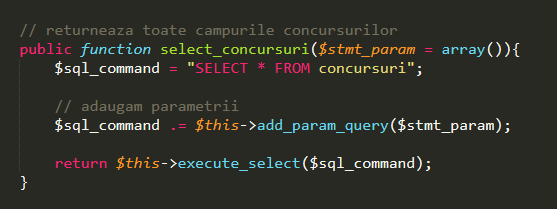
In cazul in care apare o problema intre executia primei si celei de a doua intructiuni, se va executa un roleback prin intermediul functiei puse la dispozitie de extensia oci8 numite oci\_rollback().

Implementarea functiei o veti gasi in figura de mai jos.

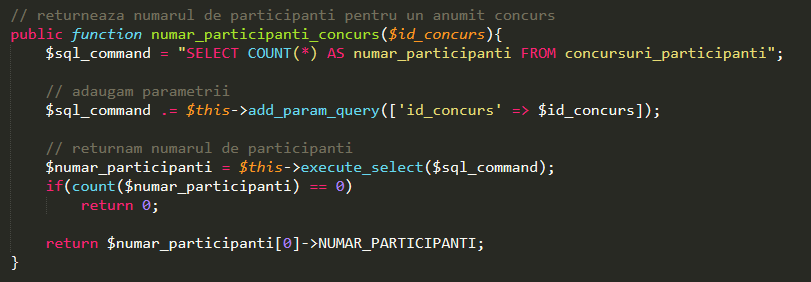


1. **Capturi de ecran cu interfata aplicatiei**
2. Pagina tabel concursuri

(pagina *concursuri\_tabele.php*)

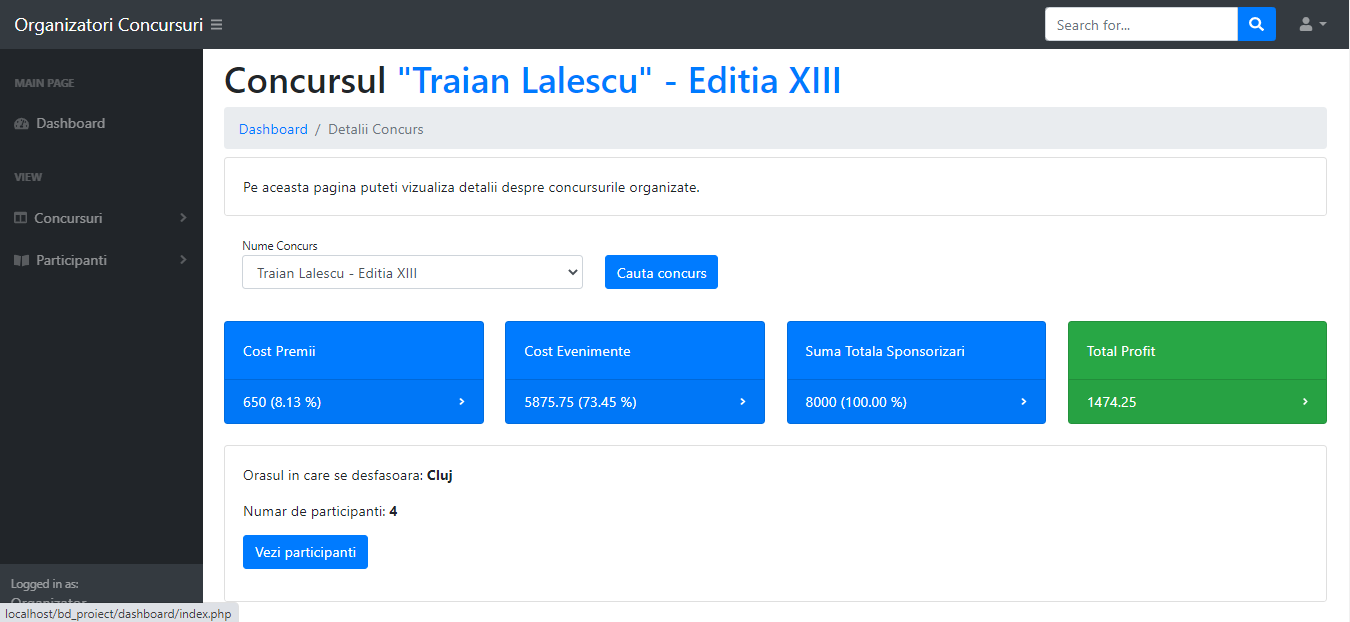


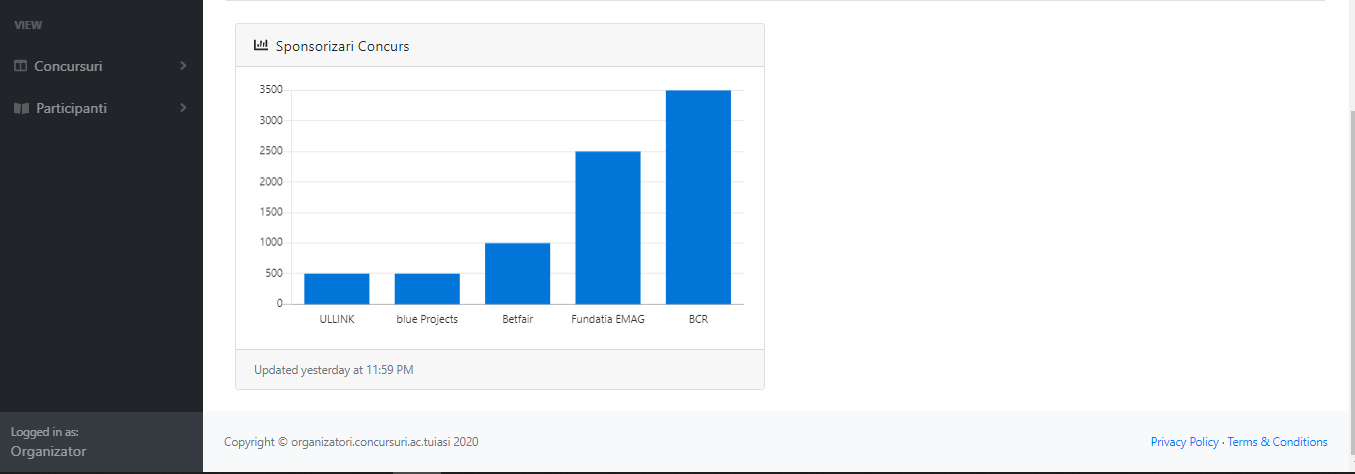
(script selecte concursuri)



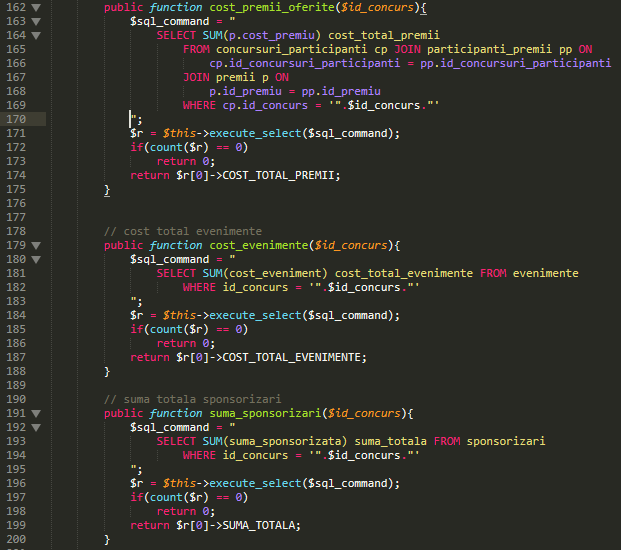
(script numarare participanti concurs)

1. Pagina detalii concurs





(pagina *detalii\_concurs.php*)



(scripturi calcul costuri premii, evenimente si suma sponsorizata)

1. **Link-uri utile**
2. Link template Bootstrap

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

1. Referinte Bootstrap

<https://getbootstrap.com/docs/4.0/getting-started/introduction/>

1. Link instalare server Apache XAMPP

<https://www.apachefriends.org/download.html>

1. Comanda instalare extensie oci8 pentru PHP pentru Linux

pecl install oci8-2.2.0

1. 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>

1. Referinte limbaj PHP

<https://www.php.net/manual/en/langref.php>