



Universitatea  
Transilvania  
din Brașov

FACULTATEA DE INGINERIE ELECTRICĂ  
ȘI ȘTIINȚA CALCULATOARELOR

# Universitatea Transilvania din Brașov

Facultatea de Inginerie Electrică și Știința Calculatoarelor

## Baze de Date

Student: Cătălin Daniel Bradac, Andreea Negru, Simona Grama

Specializarea: TST

Anul III

## Cuprins

1.	Introducere și scop .....	3
1.1.	Obiective comerciale.....	3
1.2.	Funcționalități de business .....	3
2.	Stack tehnic și justificare .....	4
3.	Arhitectură fizică.....	5
4.	Baza de date – detalii tehnice .....	5
4.1.	Chei și indici.....	5
4.2.	Indice FULLTEXT .....	6
4.3.	Audit_Log schemă .....	6
5.	Flux securitate .....	6
6.	Erori & UX.....	7
7.	Explicații cheie de cod .....	7
8.	Structura fișierelor.....	8
9.	Bibliografie.....	9

## 1. Introducere și scop

Aplicația „Zona Administrativă” este o soluție web „all-in-one” pentru secretariatul unei facultăți care dorește să înlocuiască fișierele Excel cu un sistem centralizat, auditat și ușor de operat.

### 1.1. Obiective comerciale

- Timp de onboarding < 1 oră – instalare prin simplu upload + import SQL.
- Cost zero licențe – stack open-source (PHP, MySQL, Bootstrap).
- Compliance – tabel Audit\_Log păstrează istoric complet (GDPR ready).
- Modular – fiecare entitate (Studenti, Cursuri...) este separată → ușor de dezvoltat integrări API.

### 1.2. Funcționalități de business

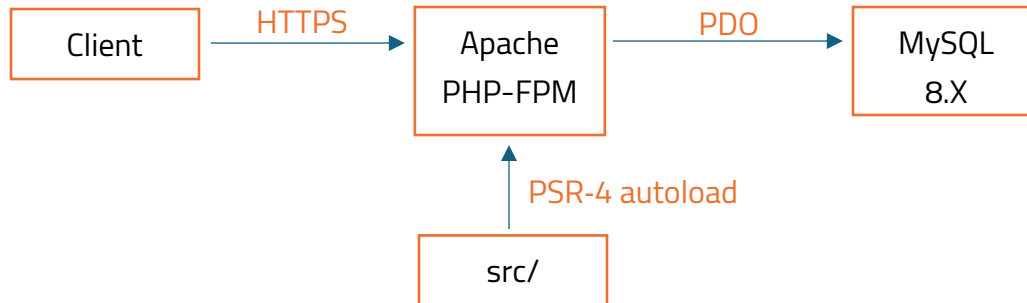
Modul	Macro-funcții	Detalii implementare
Autentificare	Login, Logout, rol	parole bcrypt, audit IP, rate-limit la nivel Apache (fișier separat).
Studenti	CRUD + căutare rapidă	Indice FULLTEXT (Nume, Prenume) ⇒ căutare $O(\log n)$ .
Cursuri	CRUD + filtrare semestru	ENUM Semestru (1,2) - motiv: economisește JOIN suplimentar.
Catalog (Înscrieri)	note unice student-curs	PK compus → integritate, tratat PDO-23000 pentru UX.
Prezențe	max 14 / semestru, grafic	Limită validată în model; view de statistici prin Attendance::stats().
Utilizatori	admin CRUD conturi	Rol ENUM + ACL (Auth::MAP).

Audit	trace all + IP	JSON payload, stocare permanentă (no trim).
Export CSV	orice listă	folosește "streamed response" pentru mem-footprint minim.

## 2. Stack tehnic și justificare

Strat	Tech	De ce?	Trade-off
Limbaaj	PHP 8.1	disponibil shared hosting, tipizare strictă, match, enum, readonly.	fără type-safety ca Rust/Go, dar time-to-market rapid.
Framework	„Hand-rolled MVC”	control maxim, zero dependențe, ușor de auditat.	lipsă ORM sofisticat; compensăm cu PDO + helperi.
DB	MySQL 8 InnoDB	FK, cascade, JSON ops, comun în cPanel.	pentru scaling >100k useri → eventual shard/Postgre.
Front	Bootstrap 5.3 + Vanilla JS	CDN, timp zero de configurare, accesibilitate out-of-box.	UI generic; se poate tema cu SCSS.
Charting	Chart.js CDN	60 kB, rendering rapid pe <canvas>.	pentru dataset >5k points → migrate la CanvasJS.
Logs	Tabel SQL + IP	compliance-ready, ușor de query pentru rapoarte GDPR.	DB crește; se arhivează anual.

### 3. Arhitectură fizică



- Front Controller : index.php unifică logging, sesiune, router.
- Controllers injecție minimă (model + ACL).
- Views sunt izolate – nu pot fi accesate direct (blocate din .htaccess).

### 4. Baza de date – detalii tehnice

#### 4.1. Chei și indici

```
CREATE TABLE Inscrieri (  
  ID_Student INT NOT NULL,  
  ID_Curs INT NOT NULL,  
  Nota_Finala DECIMAL(3,1) NOT NULL CHECK (Nota_Finala BETWEEN 1 AND 10),  
  PRIMARY KEY (ID_Student, ID_Curs),  
  FOREIGN KEY (ID_Student) REFERENCES Studenti(ID_Student) ON DELETE CASCADE,  
  FOREIGN KEY (ID_Curs) REFERENCES Cursuri(ID_Curs) ON DELETE CASCADE  
) ENGINE=InnoDB;
```

- PK compus → garantează un singur rând / combinație; query-ul de UPDATE este index-friendly.
- CHECK pe Notă → previne injection numerică.

## 4.2. Indice FULLTEXT

**ALTER TABLE Studenti ADD FULLTEXT(Nume, Prenume, Email);**

folosit în **Student::search(\$q)** pentru căutare instant.

## 4.3. Audit\_Log schemă

Col	Tip	Motiv
ID_User	INT FK	trace responsabil.
IP	VARCHAR(45)	audit trail, IPv6-ready.
Actiune	VARCHAR(20)	create/update/delete/login/logout.
Entitate	VARCHAR(30)	Studenti / Cursuri etc.
Info	TEXT	JSON cu before/after payload.
Created_at	TIMESTAMP	ordonare și analiză.

## 5. Flux securitate

1. Login → AuthController::login() → setează sesiune, scrie audit.
2. Router verifică dacă există \$ctrl->\$action – else 404 → fallback custom.
3. Controller cheamă Auth::can(\$op) → map rol-acțiune; altfel 403.
4. Model folosește prepared-statement cu binding → zero SQLi.
5. Pe delete se loghează întregul rând (SELECT ... before) pentru rollback.
6. .htaccess blochează /src, /views, .env, dumpuri etc.

## 6. Erori & UX

- Duplicate key – prindem PDOException code 23000, returnăm alert Bootstrap.
- Telefon invalid – regex server-side + feedback instant (alert-danger).
- 404/403 – cod HTTP + mesaj text (simplu, no leak path). Pentru production se configurează ErrorDocument 404 /error/404.html.

## 7. Explicații cheie de cod

Fragment	Rațiune	Alternativă
Database::getInstance()	Single PDO conn / request -> performanță	Container DI (Symfony) – overkill aici.
Auth::MAP	map static rol-op – rapid, clar	RBAC DB-driven – necesită interfață extra.
Audit::log(\$user,\$act,\$tbl,\$id,\$json)	centralizează logging	Monolog – numai dacă vrei fișier text + rotație.
.htaccess RewriteRule ^src	prevenim leak cod sursă	Mutare cod în ../app – necesită custom path.

## 8. Structura fișierelor

```
proiect.bradac.ro/
├── .htaccess                # protecții Apache (no-index, deny src/
views/)
├── autoload.php            # autoloader PSR-4 minimal (≈20 LOC)
├── config.php              # setări ini + const DB
├── index.php               # front-controller + router
├── install/ schema.sql     # script inițial BD
├── src/
│   ├── Core/
│   │   ├── Controller.php  # bază pentru toate controllerele
│   │   ├── Database.php    # singleton PDO, ATTR_ERRMODE=EXCEPTION
│   │   └── Auth.php        # map rol->acțiuni (ACL) + helper can()
│   ├── Models/
│   │   ├── Student.php     # CRUD, validare telefon, fulltext search
│   │   ├── Course.php      # CRUD + search pe semestru
│   │   ├── Enrollment.php  # CRUD; PK compus, JOIN Student/Course
│   │   ├── Attendance.php  # CRUD + stats() (absente pe săptămână)
│   │   ├── User.php        # conturi aplicație, hash parole bcrypt
│   │   └── Audit.php       # static log() - inserează în Audit_Log
│   ├── Controllers/
│   │   ├── StudentController.php # orchestrare + audit + ACL
│   │   ├── CourseController.php  # idem pentru Cursuri
│   │   └── EnrollmentController.php # Catalog înscrieri (dropdown
id-safe)
│   ├── AttendanceController.php # Prezențe (limită 14/sem)
│   ├── UserController.php       # admin: CRUD conturi
│   ├── AuthController.php       # login/logout + audit
│   ├── AuditController.php      # listare Audit_Log
│   └── ExportController.php      # CSV streaming universal
└── Views/
    ├── templates/ header.php footer.php # nav-bar + bootstrap
    ├── students/ list.php form.php
    ├── courses/ list.php form.php
    ├── enrollments/ list.php form.php
    ├── attendance/ list.php form.php
    ├── users/ list.php form.php
    ├── audit/ list.php # tabel audit
    └── auth/ login.php
```



## Legende fișiere importante

Controller – primește input GET/POST, validează, cheamă Model, decide view.

Model – interacțiune pură DB. try/catch pentru duplicate-key, returnează arrays.

View – HTML + Bootstrap; toate datele sunt escapate via htmlspecialchars. Nici un view nu conține query-uri.

## 9. Bibliografie

Nr. Titlu / Resursă	Utilizare în proiect
[1] <b>PHP Manual</b> – <a href="https://www.php.net/manual/en/">https://www.php.net/manual/en/</a>	sintaxă typed-properties, password_hash, PDO.
[2] <b>MySQL 8 Reference</b> – <a href="https://dev.mysql.com/doc/">https://dev.mysql.com/doc/</a>	constrângeri FK, CHECK, YEARWEEK(), FULLTEXT.
[3] <b>Bootstrap 5 Docs</b> – <a href="https://getbootstrap.com/">https://getbootstrap.com/</a>	grid, card, alert, responsive navbar.
[4] <b>Chart.js 4.4 Guide</b> – <a href="https://www.chartjs.org/docs/">https://www.chartjs.org/docs/</a>	generare rapidă donut & line chart pe dashboard.
[5] <b>OWASP Cheat Sheet: Secure Headers</b> – <a href="https://cheatsheetseries.owasp.org/">https://cheatsheetseries.owasp.org/</a>	setare X-Frame-Options, X-Content-Type-Options.
[6] <b>RFC 7617</b> – Basic Auth	inspirație pentru .htaccess protections.
[7] <b>PSR-4: Autoloader</b> – PHP-FIG	designul autoload.php custom.
[8] <b>Wikipedia ENUM (SQL)</b>	motivație stocare rol & semestru.

Link: <https://github.com/dbradacc/Baze-de-Date-Proiect/>