

Whitepaper – Proiect BD

Toate interacțiunile de producție cu baza de date sunt **înglobate în clasele Model** aflate sub **/src/Models/**.

proiect.bradac.ro/

└─ src/

└─ Models/

└─ Student.php

└─ Course.php

└─ Enrollment.php

└─ Attendance.php

└─ User.php

└─ Audit.php

Cum se împart responsabilitățile

Model	Locație exactă	Ce query-uri conține
Student.php	/src/Models/Student.php	SELECT * FROM Studenti ..., INSERT INTO Studenti ..., UPDATE, DELETE (tot CRUD) + FULLTEXT search
Course.php	/src/Models/Course.php	CRUD pentru tabelul Cursuri
Enrollment.php	/src/Models/Enrollment.php	CRUD pe Inscrieri (PK compus), JOIN-uri către Studenti & Cursuri
Attendance.php	/src/Models/Attendance.php	CRUD pe Prezente, plus funcția stats() (COUNT / GROUP BY săptămână)
User.php	/src/Models/User.php	CRUD conturi + lookup findByUsername()

Model	Locație exactă	Ce query-uri conține
Audit.php	/src/Models/Audit.php	un singur INSERT în Audit_Log (logare acțiuni)

*Fiecare metodă folosește **PDO prepared statements** – codul SQL este în interiorul metodelor `create()`, `update()`, `delete()`, `search()` etc.*

/src/Core/

Acest director conține clasele „de infrastructură” – adică acele componente pe care se bazează toate celelalte straturi.

Fișier	Locație exactă	Responsabilitate principală	De ce există aici
Controller.php	/src/Core/Controller.php	Clasă-bază pentru toate controllerele concrete; poate găzdui hooks comune (ex.: CSRF, rate-limit)	Centralizează logică transversală; respectăm DRY – nu repetăm cod în fiecare controller.
Database.php	/src/Core/Database.php	Singleton PDO; configurează conexiunea, charset utf8mb4, mode EXCEPTION	Evită multiple conexiuni în același request; un singur loc de schimbat când trecem la MySQL over SSL.
Auth.php	/src/Core/Auth.php	Mecanism ACL minimalist: tabele de permisiuni din map static rol → [permise] DB la fiecare request; și helper <code>can()</code>	Viteză O(1); nu încărcăm roles sunt rareori schimbate.

/src/Controllers/

Controller	Locație exactă	Tabăra corespunză toare / Rol	Operații cheie	Audit
StudentController.php	/src/Controllers/StudentController.php	„Studenti”	create, update, delete, search, regex telefon	Da (create/update/delete)
CourseController.php	/src/Controllers/CourseController.php	„Cursuri”	CRUD filtrare semestru	+ Da
EnrollmentController.php	/src/Controllers/EnrollmentController.php	„Catalog” (note)	CRUD dropdown safe, duplicate catch	cu key- Da
AttendanceController.php	/src/Controllers/AttendanceController.php	„Prezențe”	CRUD limită săpt./sem., stats() pentru grafice	+ 14 Da
UserController.php	/src/Controllers/UserController.php	„Utilizatori” (admin only)	CRUD conturi, schimbă	Da

Controller	Locație exactă	Tabăra corespunză toare / Rol	Operații cheie	Audit
			roluri, resetă parolă	
AuthController.php	/src/Controllers/AuthController.php	Login Logout	/ password_verify, redirect	Audit login & logout
AuditController.php	/src/Controllers/AuditController.php	Raport log	SELECT joint cu Utilizatori, – (self-view) pagină doar admin	
ExportController.php	/src/Controllers/ExportController.php	CSV generator	stream php://output , fputs, – escape param	

Flux intern (exemplu adăugare student)

Browser → index.php?resource=students&action=create

Router construiește new StudentController

StudentController::create()

└─ Auth::can('create') (Core/Auth.php)

└─ Validare telefon + duplicate email (try/catch PDO 23000)

└─ Student->create() (Model)

└─ Audit::log() (Model Audit)

└─ redirect index.php?resource=students

/views/ – stratul de prezentare

(HTML + Bootstrap)

proiect.bradac.ro/

└─ views/

└─ templates/

| └─ header.php

| └─ footer.php

└─ 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

└─ auth/ login.php

Subfolder	Fișiere	Rol UI	Tehnici / observații
templates/	header.php, footer.php	wrapper comun (nav- bar, Bootstrap, rolului; logout)	<i>header</i> setează link-uri dinamice pe baza variabilele sunt escapate cu htmlspecialchars.
students/	list.php	tabel cu Studenți + butoane CRUD, filtru căutare	primește array \$students din Controller; link-uri folosesc ?resource=students&action=....
	form.php	formular add/edit student	afișează \$error (duplicate email / telefon invalid); pre-populează câmpurile la edit.
courses/	list.php, form.php	idem pentru Cursuri	afișează semestru, profesor, credite.
enrollments/	list.php	Catalog: JOIN Student + Curs + Notă	butoane Editează/Șterge construite pe 1 linie (evită newline→404).
	form.php	dropdown Student, Curs, input Notă	arată mesaj duplicat „Studentul are deja notă” (prins în controller).
attendance/	list.php	tabel prezențe + statistici săptămânale	include grafic canvas (Chart.js) stat din \$stats.
	form.php	dropdown Student, Curs, data, status absent/prezent	limită 14 prezențe/sem validată în Controller.

Subfolder	Fișiere	Rol UI	Tehnici / observații
users/	list.php, form.php	management conturi la editare, câmp parolă poate fi gol → (admin only)	păstrează hash existent.
audit/	list.php	raport audit complet	tabel striped, col. Info JSON escapata, IP, user, acțiune.
auth/	login.php	form login + alert „Date incorecte”	nu include nav-bar (header.php se adaugă după autentificare).