

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

Adaptive learning pentru Moodle

propusă de

Cătălin-Dimirie Morărașu

Sesiunea: iulie, 2025

Coordonator științific

Prof. Dr. Alboaie Lenuța

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

Adaptive learning pentru Moodle

Cătălin-Dimirie Morărașu

Sesiunea: iulie, 2025

Coordonator științific

Prof. Dr. Alboaie Lenuța

Avizat,
Îndrumător lucrare de licență,
Prof. Dr. Alboaie Lenuța.

Data: Semnătura:

Declarație privind originalitatea conținutului lucrării de licență

Subsemnatul **Morărașu Cătălin-Dimirie** domiciliat în **România, jud. Iași, mun. Iași, Sos. Rediu, nr. 6A, bl. 482E, et. 3, ap. 15**, născut la data de **10 octombrie 2003**, identificat prin CNP **5031010226726**, absolvent al Facultății de informatică, **Facultatea de informatică** specializarea **informatică**, promoția 2018, declar pe propria răspundere cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art. 143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul **Adaptive learning pentru Moodle** elaborată sub îndrumarea domnului **Prof. Dr. Alboaie Lenuța**, pe care urmează să o susțin în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului ei într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diplomă sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data:

Semnătura:

Declarație de consimțământ

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul **Adaptive learning pentru Moodle**, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test, etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de informatică.

De asemenea, sunt de acord ca Facultatea de informatică de la Universitatea "Alexandru-Ioan Cuza" din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Absolvent **Cătălin-Dimirie Morărașu**

Data:

Semnătura:

Cuprins

Motivație	2
Introducere	3
1 Studiu de Caz	5
2 Tehnologiile Utilizate	6
2.1 Moodle	6
2.2 XAMPP	7
2.3 MariaDB	8
2.4 GeminiAPI	9
2.5 PHP	9
3 Detalii de implementare	10
3.1 Arhitectura generală	10
3.2 Structura generală a extensiei Moodle	11
3.2.1 Fișierul <i>version.php</i>	11
3.2.2 Directoarele <i>lang/en</i>	12
3.2.3 Fișierul <i>lib.php</i>	12
3.2.4 Directorul <i>db</i>	12
3.3 Definirea tabelelor din baza de date	13
3.4 Paginile principale ale extensiei	15
3.4.1 Pagina <i>mod_form.php</i>	15
3.4.2 Pagina <i>view.php</i>	15
3.4.3 Pagina <i>review.php</i>	16
3.4.4 Pagina <i>learn.php</i>	17
3.4.5 Pagina <i>start.php</i>	18
3.4.6 Pagina <i>test.php</i> și <i>init_test.php</i>	18

3.4.7	Pagina <i>results.php</i>	19
3.4.8	Pagina <i>available_modules.php</i>	19
3.4.9	Pagina <i>finish.php</i>	20
3.5	Interacțiunea cu Gemini API	20
3.5.1	Generarea modulelor de învățare	20
3.5.2	Generarea testelor grilă	21
4	Cazuri de Utilizare	23
4.1	Use Case Diagram	24
4.2	Cazuri de utilizare pentru profesor	24
4.2.1	Crearea activității	24
4.2.2	Revizuirea modulelor generate	25
4.2.3	Vizualizarea modulelor și urmărirea progresului studenților . . .	26
4.2.4	Ștergerea activității	26
4.3	Cazuri de utilizare pentru student	26
4.3.1	Începerea activității	26
4.3.2	Completarea testului inițial	27
4.3.3	Vizualizarea modulelor	28
4.3.4	Completarea testului pentru a trece la următorul modul	29
4.3.5	Vizualizarea rezultatului obținut la test	29
4.3.6	Vizualizarea tuturor modulelor disponibile	30
4.3.7	Finalizarea activității	30
	Concluzii	32
	Bibliografie	33

Motivație

În contextul actual al sistemului educațional, nevoia de personalizare și adaptare a procesului de învățare este tot mai importantă. Metodele tradiționale de predare nu mai sunt suficiente și nu țin pasul cu diversitatea nevoilor unui student, diferitele stiluri de învățare și nu în ultimul rând evoluția tehnologiei. În același timp, sistemele existente de e-learning, deși oferă acces facil la o gamă largă de resurse educaționale, nu reușesc întotdeauna să țină cont de particularitățile fiecărui student. Ca soluție pentru această problemă a apărut învățarea adaptivă, un concept ce presupune ajustarea dinamică a conținutului și parcursului educațional în funcție de performanța și nevoile fiecărui student, aceasta devenind o soluție tot mai atractivă în mediul academic.[2]

Moodle, una dintre cele mai utilizate sisteme de gestionare a învățării la nivel global, oferă deja un cadru flexibil pentru dezvoltarea de extensii care să îmbunătățească întreg procesul educațional pentru studenți, profesori și instituții. Cu toate acestea, crearea materialelor didactice rămâne un mare consumator de timp pentru profesori, deoarece în continuare sunt realizate manual. Astfel, generarea automată a conținutului educațional și a testelor de evaluare pe baza unor documente deja existente ar putea reuși semnificativ acest efort.

Motivația lucrării este dublă. În primul rând se urmărește îmbunătățirea experienței de învățare a studentului prin crearea unui mediu dinamic, care se adaptează la nivelul fiecăruia. În al doilea rând, se încearcă automatizarea și eficientizarea muncii profesorului, astfel progresul metodelor de predare este susținut de tehnologie și de inteligența artificială.

Introducere

Învățarea adaptivă este o abordare care utilizează tehnologia și metodologii pentru a adapta procesul de învățare în funcție de nevoile, ritmul și preferințele fiecărui individ. Conceptul central al teoriei este că învățarea personalizată și adaptivă poate crește implicarea și eficacitatea învățării. În același timp, flexibilitatea, feedback-ul în timp util și utilizarea datelor sunt esențiale pentru ajustarea strategiilor de învățare. [1]

Într-o eră ce se caracterizează prin continuă dezvoltare și evoluție, metodele tradiționale de învățare adesea nu reușesc să satisfacă eficient nevoile tuturor cursanților. De exemplu, în sălile de clasă mari este dificil să se adapteze materialul lecțiilor în ritmul de învățare al fiecărui student. Acest lucru evidențiază necesitatea unei abordări mai adaptive și personalizate pentru a îmbunătăți rezultatele învățării.[1]

Elementele cheie ce trebuie urmate pentru a crea un mediu de învățare adaptiv sunt: flexibilitatea, feedback-ul și utilizarea datelor pentru ajustarea strategiilor de învățare și identificarea nevoilor și preferințelor de învățare ale fiecărui student. În același timp, dezvoltarea de conținut și strategii de învățare care pot fi ajustate dinamic pe baza datelor și a feedback-ului obținut în timpul procesului de învățare este esențială. Totodată, este necesară implementarea de tehnologii care sprijină învățarea adaptivă, cum ar fi platformele de e-learning care utilizează algoritmi pentru a adapta materialele lecțiilor, iar toți acești factori trebuie susținuți de o curriculă flexibilă ce permite satisfacerea nevoilor studenților. [1]

Această lucrare propune dezvoltarea unei extensii pentru platforma Moodle care integrează tehnologii de procesare a limbajului natural și de generare automată de conținut educațional. Prin încărcarea unui document în format PDF, extensia va extrage informațiile și pe baza acestora va structura materialul în module de învățare, iar pentru fiecare modul va genera teste. Accesul studentului la modulele ulterioare este condiționat de finalizarea și promovarea testului de la modulul curent, implementând astfel un parcurs educațional secvențial și adaptiv. Scopul principal este de a

îmbunătăți rezultatele educaționale și a experienței de învățare a studenților, reducând astfel decalajul de performanță prin personalizare. Totodată, ajustarea ritmului de învățare și evaluarea continuă a progresului și a performanței sunt trăsături esențiale ale învățării adaptive, iar extensia propusă permite acest lucru prin testele generate automat după finalizarea fiecărui modul.[1]

Capitolul 1

Studiu de Caz

Capitolul 2

Tehnologiile Utilizate

2.1 Moodle

Moodle este sistemul de management al învățării (LMS) pentru care este concepută această extensie. Moodle este o aplicație web scrisă în PHP [3], de tip open-source, care are ca scop oferirea unui mediu unificat de învățare profesorilor și studenților. Aceasta a luat naștere dintr-un proiect de cercetare doctorală (PhD) condus de Martin Dougiamas, cu ajutorul lui Peter C. Taylor, la Curtin University of Technology. Scopul principal al proiectului de cercetare a fost de a explora cum software-ul de internet poate sprijini cu succes epistemologii de predare și învățare bazate pe construcționismul social. Întrebarea principală a cercetării a fost clară: ce tipuri de structuri web și interfețe ajută sau, dimpotrivă, împiedică implicarea activă a participanților într-un dialog reflexiv, într-o comunitate de învățare? Accentul a fost pus pe sprijinirea lecturii deschise, a reflecției critice și a unei scrieri constructive, care să pornească din experiențele personale ale cursanților. Pe baza acestor observații, dezvoltarea platformei Moodle este orientată constant de această analiză, fiind gândită ca un instrument care sprijină și îmbunătățește procesele de învățare reflexivă în comunitate. [8]

Acest model open-source permite ca modificările făcute de utilizatori să fie adesea integrate în proiectul principal, permițând software-ului să evolueze conform valorilor comunității de utilizatori. Designul Moodle a fost gândit specific pentru a fi compatibil, flexibil și ușor de modificat. Este scris în limbajul PHP și este construit modular, utilizând tehnologii comune. Această abordare modulară a fost adoptată inițial pentru a permite modificarea rapidă a interfețelor ca răspuns la analiză și interese de cercetare, dar acum permite și altor programatori să modifice și să extindă codul. [8]

Adaptibilitatea este o caracteristică importantă pentru cei ce aleg să folosească Moodle. Spre deosebire de software-ul restricționat de licențe care limitează personalizarea, Moodle permite accesul la codul sursă și modificarea acestuia, această flexibilitate fiind văzută ca un mare avantaj. Capacitatea de a personaliza Moodle a fost un motiv pentru adoptare la instituții precum Otago Polytechnic din Noua Zeelandă și Dublin City University (DCU).[9]

Distribuția Moodle are trei componente principale: codul sursă ce rulează pe un server web, o baza de date relațională (MariaDB) destinată stocării datelor și un spațiu de depozitare pentru toate fișierele folosite (folderul moodledata). Cursurile, activitățile și resursele sunt stocate în baza de date, iar extensiile instalate sunt stocate în folderul moodledata. Extensia creată intră în această categorie ca un modul de activitate. Moodle definește un modul de activitate ca fiind o extensie în care studentul interacționează cu alți studenți sau cu profesorul. Într-o activitate, studenții pot contribui direct la ceva propus de profesor, cum ar fi o resursă în forma unui fișier sau a unei pagini web.[3] Extensia este formată dintr-un set de fișiere PHP, precum version.php, lib.php, view.php, etc. și fișiere CSS și JavaScript pentru stilizare și interactivitate. Moodle apelează funcțiile extensiei în momentul în care utilizatorul accesează pagina acesteia în interfața web. Prin utilizarea API-ului Moodle, extensia are parte de mecanisme standard pentru gestionarea bazei de date, controlul accesului și a permisiunilor și de integrare, instalare și actualizare ușoară.

2.2 XAMPP

Pentru dezvoltarea, testarea locală a extensiei și pentru rularea Moodle, a fost utilizat XAMPP. XAMPP este un pachet software gratuit care conține Apache, MariaDB, PHP și Perl, „XAMPP este cel mai popular mediu de dezvoltare PHP”[4]. În contextul dezvoltării extensiei, Apache servește ca server web pentru aplicația Moodle și răspunde la cererile HTTP ale utilizatorilor. Am ales XAMPP deoarece facilitează o configurare rapidă și simplă a mediului de dezvoltare deoarece nu este necesară instalarea separată a fiecărei componentă în parte.

2.3 MariaDB

MariaDB este sistemul de baze de date relațional folosit pentru stocarea datelor aplicației Moodle. MariaDB a fost creat inițial ca „fork al MySQL”, menținând compatibilitatea cu acesta la nivel de protocol și dialect SQL[4]. Principalele motive pentru alegerea MariaDB ca sistem de baze de date pentru Moodle sunt:

- **Performanță și scalabilitate:** MariaDB este recunoscută pentru performanța sa îmbunătățită și pentru facilitarea unei scalări mai eficiente a aplicațiilor, ceea ce este esențial pentru Moodle, având în vedere numărul mare de utilizatori și volumele de date gestionate. Teste comparative au demonstrat că MariaDB depășește MySQL în special în mediile cloud, permițând un debit mai mare de tranzacții și o latență mai mică în accesarea datelor pentru fire de execuție multiple ce rulează în paralel. [11]
- **Motoare de stocare suplimentare:** Pe lângă motoarele de stocare de date din MySQL, MariaDB oferă și altele noi, cum ar fi XtraDB, MariaDB ColumnStore și Aria, care îmbunătățesc performanța și flexibilitatea în gestionarea datelor. Aceste motoare permit optimizarea stocării și accesului la date, ceea ce este benefic pentru Moodle, având în vedere diversitatea tipurilor de date și a volumelor de informații gestionate. [11]
- **Caracteristici avansate pentru medii cu sarcini ridicate:** MariaDB oferă posibilitatea de a gestiona peste 200.000 de utilizatori conectați simultan. În același timp, aceasta are capacitatea de a cripta jurnalele binare și tabelele temporare. [11]
- **Suport pentru Clustering și toleranța la erori:** Un atribut important al MariaDB este capacitatea de a crea clustere de baze de date care permit gestionarea distribuției sarcinilor pe nodurile serverului de baze de date. Utilizarea acestei funcționalități îmbunătățește toleranța la erori, asigurând o redistribuire a traficului în caz de eșec al unui nod. Studiile arată că utilizarea acestei caracteristici poate îmbunătăți semnificativ performanța mediului Moodle. [11]
- **Optimizare pentru configurații specifice:** Pentru sistemele bazate pe Windows, MariaDB este considerată configurația optimă pentru instalarea Moodle. Această variantă oferă avantaje de clustering al bazei de date și echilibrare a sarcinii, asigurând calitate optimă, viteză ridicată și absența problemelor de securitate. Deși

Apache/MySQL este considerată optimă pentru distribuțiile Linux, MariaDB are potențialul de a înlocui MySQL în acest context. [11]

2.4 GeminiAPI

Gemini face parte din familia de modele generative de la Google DeepMind/Vertex AI. Aceste modele de inteligență artificială pot procesa intrări multimodale, text și imagini, și generează conținut corect. Gemini este „cel mai capabil și general model AI” creat de Google până în prezent, optimizat în variante precum Ultra, Pro și Nano pentru sarcini de complexități diferite. Gemini este accesibil prin API-ul Google Cloud Vertex AI, care permite aplicațiilor externe să trimită prompt-uri și să primească răspunsuri textuale generate. [5]

Domeniul Educational este unul dintre cele mai importante domenii de aplicare pentru inteligența artificială, iar aceasta are datorită de a transforma modul în care se predă și se învață, elevii numărându-se printre cei mai entuziaști utilizatori ai instrumentelor de inteligență artificială generativă. Deși nu există un reper robust și general pentru a evalua modelele AI pentru învățare, evaluările existente se concentrează adesea pe sarcini educaționale cum ar fi acuratețea la examene sau identificarea greșelilor. Predarea eficientă necesită mai mult decât suma acestor capacități individuale, necesită o înțelegere a cum și când sunt utilizate în practică. [13]

2.5 PHP

Limbajul de programare folosit este PHP sau Hypertext Preprocessor. PHP este un limbaj de scriptare open-source, executat pe server, utilizat la scară largă pentru dezvoltarea de aplicații web dinamice. PHP poate genera conținut HTML dinamic, poate gestiona formulare web, poate interacționa cu sisteme de fișiere și poate efectua operații „Create, Read, Update, Delete” pe baze de date. De asemenea, acesta permite efectuarea de apeluri HTTP către API-uri externe, facilitând comunicarea cu servicii precum Gemini API, folosit pentru generarea modulelor de învățare. [7]

Capitolul 3

Detalii de implementare

Scopul extensiei create este de prelua un fișier PDF încărcat de profesor într-un formular din Moodle și de a genera automat module de învățare și teste grilă, teste cu patru variante de răspuns și doar una corectă, pe baza informațiilor din fișier. Mediul de dezvoltare folosit este XAMPP, cu un server Apache, limbajul de programare PHP, o bază de date MariaDB, și platforma Moodle instalată local. Generarea conținutului educațional pe baza materialului încărcat se face cu ajutorul API-ului Gemini de la Google, care oferă un model de inteligență artificială capabil să extragă și să reformuleze text pentru a genera module de învățare și să creeze întrebări cu un singur răspuns corect pentru testele ce facilitează trecerea individului la următorul nivel.

3.1 Arhitectura generală

Arhitectura sistemului utilizează modelul client-server. Clientul este reprezentat de interfața Moodle, iar aceasta diferă în funcție de rolul utilizatorului. Pentru profesor, extensia va fi vizibilă sub forma unei activități Moodle și va putea fi adăugată ca orice altă activitate, cum ar fi Quiz, Lecție sau orice altă activitate preinstalată de configurația standard. După selectarea extensiei, profesorul va avea parte de un formular unde poate încărca documentul în format PDF, iar extensia va crea automat modulele în urma completării acesteia. După generarea modulelor profesorul selectează dacă le salvează sau le generează încă o dată. În urma salvării acestora, profesorul le poate vizualiza și este informat de progresul studenților ce au început această activitate, printr-un tabel unde este trecut numele studentului, modulul la care se află și ora la care a început modulul. Pentru student, interfața permite începerea activității, unde

studentul susține un test pentru a de la ce modul este asignat pentru a-și începe studiul, vizualizarea modului de învățare curent și a testului generat pe baza căruia îi este facilitată trecerea la următorul modul. Serverul este aplicația Moodle încărcată local pe care este instalată extensia. Backend-ul extensiei se ocupă cu preluarea fișierului PDF, gestionarea permisiunilor de acces, încărcarea informațiilor în baza de date și comunicarea cu API-ul Gemini, folosit pentru generarea conținutului din cadrul modulelor și a testelor.

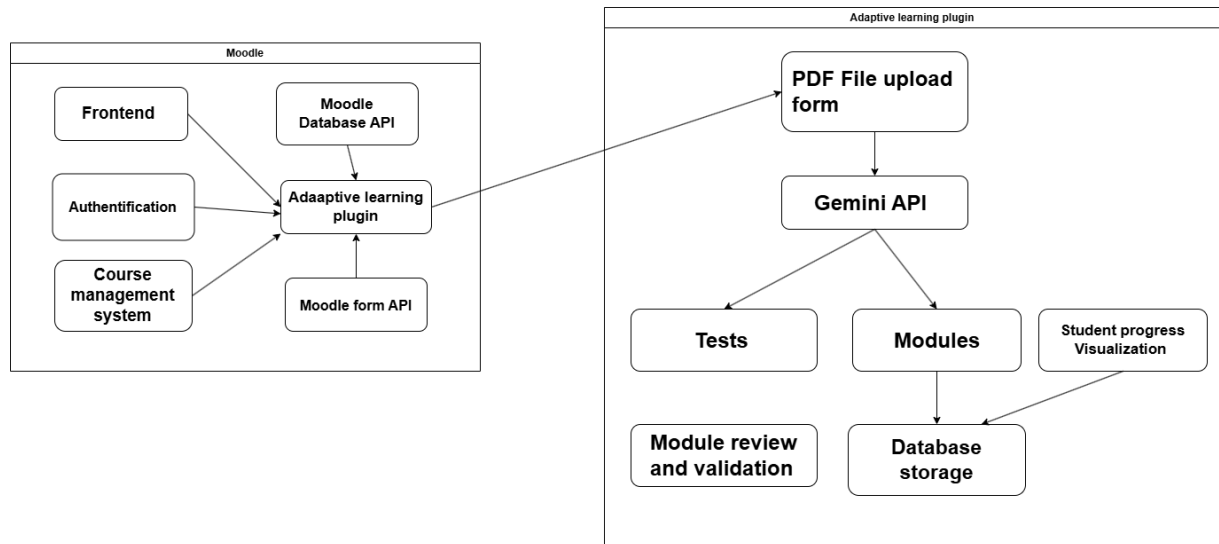


Figura 3.1: Arhitectura generală a extensiei

3.2 Structura generală a extensiei Moodle

Extensia dezvoltată este un plugin de tip *activity* pentru Moodle, parte a categoriei *mod*. Conform standardelor Moodle de dezvoltare a extensiilor, fiecare modul de activitate trebuie să se regăsească în directorul */mod/* și respecte convenția de denumire *mod_[modname]*, în cazul acesta denumindu-se *mod_adaptivelearning*. [15] Toate extensiile moodle sunt structurate într-un mod specific, având un set de fișiere și directoare necesare pentru a putea fi detectate de Moodle și pentru a funcționa corect.

3.2.1 Fisierul *version.php*

Primul fișier este *version.php*, fișier ce conține metadate despre extensie. Acest fișier este esențial pentru ca Moodle să recunoască extensia, să o instaleze și să o ac-

tualizeze. Acesta conține informații precum versiunea extensiei, versiunea Moodle necesară pentru a rula extensia, numele și descrierea acesteia, maturitatea și o listă de dependențe de alte extensii.[14]

3.2.2 Directoarele *lang/en*

Directoarele *lang/en* sunt fișiere de limbă în care trebuie să existe un fișier *adaptivelearning.php* ce conține toate stringurile necesare pentru a traduce extensia în limba engleză. Aceste stringuri sunt folosite pentru a afișa text în interfața utilizatorului, iar ele sunt accesate prin API-ul *get_string()* al Moodle, ce primește ca parametri un identificator string și componenta unde se regăsește și returnează textul corespunzător din fișierul de limbă.[14]

3.2.3 Fișierul *lib.php*

Fișierul *lib.php* acționează ca un punct de legătură între Moodle și extensia noastră. Acesta conține funcțiile pentru a integra extensia în Moodle, cum ar fi funcțiile de adăugare, actualizare și ștergere a unei instanțe a extensiei, în cazul nostru a unei activități de tip *adaptivelearning*. În același timp, în fișier se regăsesc și funcțiile folosite în cadrul altor fișiere, de exemplu metodele create pentru comunicarea cu Gemini API și cele de parsare a răspunsurilor primite.[14]

3.2.4 Directorul *db*

Directorul *db* conține fișierele necesare pentru a crea tabelele din baza de date, ce sunt folosite pentru a stoca informațiile generate de extensie. Fișierul *install.xml* este folosit pentru a defini structura tabelor, cum ar fi numele coloanelor, tipurile de date și relațiile dintre tabele. Acest fișier este folosit de Moodle pentru a crea tabelele în baza de date la instalarea extensiei. Fișierul *upgrade.php* conține pașii de actualizare a bazei de date și include schema nouă a tabelor, schimbările aduse setărilor și alte modificări realizate în timpul actualizării extensiei. Pentru a crea o reînnoire, dezvoltatorul extensiei trebuie să folosească editorul *XMLDB*, un editor vizual pentru fișierele XML ale bazei de date, pentru a crea definițiile câmpurilor noi, să actualizeze fișierul *install.xml*, să genereze fișierul *upgrade.php* și să modifice numărul versiunii în fișierul *version.php*. Ultimul fișier necesar din directorul *db* este *access.php*, ce definește permi-

siunile necesare pentru a accesa extensia. Acest fișier conține o listă de capacități, fiecare cu un nume unic și o descriere, ce sunt folosite pentru a controla accesul la diferite funcționalități ale extensiei. Aceste capacități sunt apoi asociate cu rolurile din Moodle, cum ar fi profesor sau student, pentru a controla cine poate accesa extensia și ce acțiuni poate efectua.[14]

3.3 Definirea tabelelor din baza de date

Moodle utilizează un sistem independent de SGBD, XMLDB, pentru definirea tabelelor din baza de date. Acest sistem permite dezvoltatorilor să definească structura tabelelor, relațiile dintre acestea și să efectueze modificări fără a fi necesară scrierea manuală a interogărilor SQL. Fișierul *db/install.xml* este folosit pentru a defini tabelele și câmpurile, cheile și comentariile necesare pentru extensie, iar Moodle generează automat interogările SQL necesare pentru a crea tabelele în baza de date. Tabelele definite sunt următoarele:

- *adaptivelearning*: Acest tabel stochează instanțele activității.
- *adaptivelearning_modules*: Acest tabel stochează modulele generate din fișierul PDF încărcat de profesor.
- *adaptivelearning_progress*: Acest tabel stochează progresul studenților în cadrul activității.

Tabelul *adaptivelearning* are rolul de a stoca informațiile despre activitatea creată de profesor în cadrul unui curs Moodle și conține următoarele câmpuri: *id*, *course*, *name*, *description*, *createdby*, *timecreated*. Câmpul *id* este de tip *int* și este cheia primară, autoincrementată, a tabelului. Câmpul *course* este de tip *int* și reprezintă ID-ul cursului în cadrul căruia este creată activitatea, legând activitatea de tabelul *course* din Moodle. Câmpul *name* este de tip *char(255)* și reprezintă numele activității, iar câmpul *description* este de tip *text* și reprezintă descrierea generală a activității. Câmpul *createdby* este de tip *int* și reprezintă ID-ul utilizatorului care a creat activitatea. Câmpul *timecreated* este de tip *int* și reprezintă timpul la care a fost creată activitatea, stocat în format Unix timestamp.

Tabelul *adaptivelearning_modules* are rolul de a stoca modulele de învățare generate de API-ul Gemini pe baza fișierului PDF încărcat de profesor și conține următoare-

le câmpuri: *id*, *activityid*, *moduleindex*, *title*, *content*, *timecreated*. Câmpul *id* este cheia primară, autoincrementată, a tabelului. Câmpul *activityid* este de tip *int* și reprezintă ID-ul activității la care este asociat modulul, legându-se de tabelul *adaptivelearning*. Câmpul *moduleindex* este de tip *int* și reprezintă indexul modulului în cadrul activității, pentru a păstra ordinea modulelor. Câmpul *title* este de tip *char(255)* și reprezintă titlul modului, iar câmpul *content* este de tip *text* și reprezintă conținutul modulului generat de Gemini API. Câmpul *timecreated* este de tip *int* și reprezintă timpul la care a fost creat modulul.

Tabelul *adaptivelearning_progress* are rolul de a stoca progresul studenților în cadrul activității și conține următoarele câmpuri: *id*, *userid*, *moduleid*, *status*, *attemptedat*. Câmpul *id* este cheia primară, autoincrementată, a tabelului. Câmpul *userid* este de tip *int* și reprezintă ID-ul utilizatorului care a început activitatea, legându-se de tabelul *user* din Moodle. Câmpul *moduleid* este de tip *int* și reprezintă ID-ul modulului la care se află studentul în cadrul activității, legându-se de tabelul *adaptivelearning_modules*. Câmpul *status* este de tip *char(50)* și reprezintă starea progresului studentului. Acest câmp poate avea două valori posibile: *inprogress*, pentru modulul actual la care se află studentul și urmează să dea test pentru a înregistra progres, sau *completed*, pentru modulele pe care studentul le-a finalizat prin susținerea și promovarea testului. Câmpul *attemptedat* este de tip *int* și reprezintă timpul la care studentul a început modulul.

Am ales această structură a tabelelor pentru a permite stocarea eficientă a informațiilor despre activitățile create de profesori, modulele generate și progresul studenților. Am separat responsabilitățile fiecărui tabel pentru a permite o gestionare mai ușoară a datelor și pentru a facilita interogările ulterioare și le-am legat între ele prin chei externe pentru a asigura integritatea datelor. Această structură permite extensibilitate, astfel încât se pot adăuga noi funcționalități și tabele în viitor fără a afecta funcționarea actuală a extensiei. Totodată, în cadrul tabelului *adaptivelearning_progress* am implementat o coloană *status*, cu cele două valori posibile, și nu am stocat doar modulele la care studentul se află în prezent, ci și cele pe care le-a finalizat pentru a putea implementa o pagină dedicată studentului, unde poate accesa toate modulele disponibile, iar această implementare necesită un număr mai redus de interogări.

3.4 Paginile principale ale extensiei

3.4.1 Pagina *mod_form.php*

Prima pagina din cadrul extensiei este cea a formularului de configurare a activității *mod_form.php*, la care are acces doar profesorul pentru a adăuga o instanță de tip *adaptivelearning* în cadrul unui curs Moodle. Această pagină este accesibilă din meniul de adăugare a activităților și resurselor, iar profesorul poate completa numele activității, descrierea acesteia și poate încărca un fișier PDF pe care dorește să îl folosească ca sursă de informații pentru generarea modulelor de învățare de către Gemini API. Formularul a fost creat folosind clasa *moodleform*, ce oferă funcționalități pentru a crea formulare în Moodle. În urma salvării formularului, va fi apelată funcția *add_instance()* din fișierul *lib.php*, care va crea o instanță a activității în baza de date și va salva informațiile completate de profesor. După apelul acestei funcții, extensia va redirecționa profesorul la pagina *view.php*. Într-o extensie Moodle completarea formularului din interfață este direct legată de funcția *add_instance()* din fișierul *lib.php*, care este responsabilă de crearea unei instanțe a activității, datorită mecanismului intern al Moodle. După salvarea formularului, Moodle validează datele introduse, creează un obiect *\$data* cu valorile trimise și apelează funcția *add_instance()*, cu parametrii *\$data* și *\$mform*, din *lib.php*, funcție ce inserează înregistrarea activității în tabelul *adaptivelearning*. Pentru salvarea fișierului PDF încărcat de profesor, extensia folosește funcția *file_save_draft_area_files()* din Moodle și apelează funcția *adaptivelearning_generate_modules_from_pdf()*, funcție creată, responsabilă cu preluarea fișierului și comunicarea cu Gemini API pentru generarea modulelor de învățare. În urma adăugării instanței, extensia va redirecționa profesorul la pagina *view.php*. [17]

3.4.2 Pagina *view.php*

Pagina *view.php* este pagina principală a activității, unde profesorul poate vizualiza informațiile despre activitate, modulele generate și progresul studenților. Înainte de afișarea conținutului paginii, extensia verifică dacă modulele activității există în baza de date, iar dacă nu există, va redirecționa utilizatorul la pagina de *review.php* unde profesorul salvează modulele generate. Totodată, extensia verifică dacă utilizatorul are permisiunea de a accesa pagina, verificând dacă acesta are permisiunea de a adăuga o instanță. Dacă utilizatorul nu are permisiunea de a adăuga o instanță,

înseamnă că este student, iar acesta va fi redirecționat la pagina *learn.php* cu ajutorul funcției *redirect()*. Pagina fiind destinată doar profesorilor, va avea afișate toate modulele, titlul și conținutul acestora, folosind funcția *get_records()* din Moodle, ce are ca parametrii numele tabelului și condițiile de filtrare și returnează toate înregistrările din tabel. În cazul de față am selectat toate înregistrările din tabelul *adaptivelearning_modules* unde câmpul *activityid* este egal cu ID-ul activității curente. Totodată, extensia va afișa și progresul studenților în cadrul activității, folosind funcția *get_records_sql()* din Moodle, funcție ce permite executarea interogărilor SQL personalizate și returnează rezultatele sub formă de obiecte. Am ales această funcție deoarece aveam nevoie de o interogare SQL în care să dau join între tabelele *adaptivelearning_progress* și *user* pentru a identifica utilizatorilor și între tabelele *adaptivelearning_modules* și *adaptivelearning_progress* pentru a identifica modulele la care se află fiecare student. În tabelul rezultat vor fi afișate numele studentului, prin funcția *fullname()* din Moodle care primește ca parametru un obiect de tip *user*, titlul modulului curent, și ora la care studentul a început modulul. Vor fi afișați doar studenții care au un modul cu statusul *inprogress* asupra unui modul din cadrul activității curente, pentru a nu afișa studenții care au finalizat toate modulele. În cazul în care nu există studenți care să fi început activitatea sau toți studenții care au început activitatea deja au terminat-o, extensia va afișa un mesaj corespunzător folosind funcția Moodle *notification()*.

3.4.3 Pagina *review.php*

Pagina *review.php* este pagina unde profesorul poate vizualiza modulele generate de Gemini API și poate decide dacă le salvează sau le generează încă o dată. Această pagină este accesibilă doar profesorului, prin redirecționare de la pagina *view.php* după completarea formularului, formular la care doar profesorul are acces. Răspunsul primit de la Gemini API este salvat în sesiune și este folosit pentru a afișa modulele generate în cadrul paginii. O sesiune este o stocare temporară pe server care reține date între mai multe cereri HTTP ale aceluiași utilizator. PHP creează automat un ID de sesiune care este trimis browserului sub forma unui cookie. Browserul trimite înapoi acest ID la fiecare cerere, iar PHP îl folosește pentru identificarea datelor salvate ale celui utilizator. Profesorul are două butoane disponibile, *Save* și *Regenerate*. Dacă profesorul apasă butonul *Save*, extensia va salva modulele în baza de date, folosind funcția *insert_record()*, ce are ca parametru un obiect de tip *stdClass* cu atributele necesare pen-

tru a crea o înregistrare în tabelul *adaptivelearning_modules*. Dacă profesorul apasă butonul *Regenerate*, extensia va apela funcția *adaptivelearning_generate_modules_from_pdf()* din fișierul *lib.php*, care va trimite din nou fișierul PDF la Gemini API, iar întreg procesul de salvare a modulelor va fi repetat cu noile module generate.

3.4.4 Pagina *learn.php*

Pagina *learn.php* este pagina destinată studenților, unde aceștia pot începe activitatea de învățare. Această pagină este accesibilă doar studenților, iar această verificare este realizată în pagina *view.php* prin verificarea permisiunilor de creare a unei instanțe a activității, iar doar cei ce nu au această permisiune vor fi redirecționați la *learn.php*. În cadrul acestei pagini, studentul va vedea titlul și conținutul modului curent, adică cel la care studentul are statusul *inprogress* în tabelul *adaptivelearning_progress*. Înainte de a afișa conținutul modulului, extensia verifică dacă studentul are ultimul modul din cadrul activității cu statusul *completed*, folosind funcția *record_exists_sql()* din Moodle, funcție ce permite executarea interogărilor SQL și returnează un boolean în funcție de existența înregistrării. Interogarea SQL folosită realizează un join între tabelele *adaptivelearning_progress* și *adaptivelearning_modules*, verificând ID-ul activității curente, ID-ul studentului, statusul *completed* și dacă modulul este ultimul din cadrul activității. Pentru a obține ID-ul ultimului modul din cadrul activității, realizez înainte o nouă interogare SQL prin intermediul funcției *get_field_sql()*, funcție ce returnează valoarea unui câmp dintr-o interogare SQL. Această interogare selectează ID-ul modulului cu indexul maxim din tabelul *adaptivelearning_modules* pentru activitatea curentă. Dacă ultimul modul are statusul *completed*, studentul va fi redirecționat la pagina *finish.php*, iar în cazul în care nu are ultimul modul completat, extensia va afișa conținutul modulului curent. Pentru a obține modulul curent, apelez funcția *get_record_sql()* și folosesc o interogare SQL în care dau joint între tabelele *adaptivelearning_modules* și *adaptivelearning_progress* verificând ID-ul activității curente, ID-ul studentului și statusul *inprogress*. Dacă studentul nu are niciun modul cu statusul *inprogress*, acesta este redirecționat la pagina *start.php*. Studentul are acces la două butoane, *Take test* și *See available modules*. Butonul *Take test* redirecționează studentul la pagina *test.php*, unde va putea începe testul pentru a trece la următorul modul, iar butonul *See available modules* redirecționează studentul la pagina *available_modules.php*, unde acesta poate vedea toate modulele disponibile din cadrul activității, adică toate

modulele cu statusul *completed* sau *inprogress*.

3.4.5 Pagina *start.php*

Pagina *start.php* este pagina de start a activității, unde studentul poate începe activitatea de învățare. Această pagină este accesibilă doar studenților, deoarece pentru a ajunge la ea, trebuie să fii redirecționat de la pagina *learn.php*. Pagina conține o descriere a activităților de tip *adaptivelearning* și un buton de start al activității. După apăsarea butonului, studentul este redirecționat la pagina *init_test.php* și este creată o înregistrare în tabelul *adaptivelearning_progress*, folosind funcția *insert_record()*, cu statusul *inprogress* pentru primul modul din cadrul activității.

3.4.6 Pagina *test.php* și *init_test.php*

Paginile *test.php* și *init_test.php* sunt implementate aproape identic deoarece pagina *init_test.php* folosește aceeași structură cu cea a paginii *test.php*. Ambele pagini sunt accesibile doar studenților și afișează un test generat cu ajutorul funcției *adaptivelearning_generate_quiz_from_module()* din fișierul *lib.php*, funcție ce comunică cu Gemini API. Diferențele provin din rolul fiecărei pagini. Pagina *init_test.php* este folosită pentru a vedea la ce modul va fi distribuit studentul. Prima oară va primi un test pentru primul modul, iar dacă îl promovează va primi testul pentru următorul modul și va fi creată o înregistrare în tabelul *adaptivelearning_progress* cu statusul *inprogress* pentru modulul următor, iar înregistrarea pentru modulul curent va fi actualizată cu statusul *completed*. Dacă studentul nu promovează testul, înregistrarea pentru modulul curent va rămâne cu statusul *inprogress* și studentul este redirecționat la pagina *learn.php*. Totodată, studentul are acces la butonul *skip* care îl redirecționează la pagina *learn.php* fără a mai edita înregistrarea deja creată în baza de date, iar verificarea răspunsurilor date se face în cadrul aceleiași pagini. Pagina *test.php* este folosită pentru a afișa testul generat de Gemini API destinat promovării la următorul modul. Verificarea răspunsurilor se face prin redirecționarea la pagina *results.php* după apăsarea butonului *Submit answers*. O altă diferență între cele două pagini este modul în care testul afișat este considerat promovat. În pagina *init_test.php*, pentru a putea trece la testul următorului modul studentul trebuie să obțină minim 75% întrebări cu răspuns corect. Deoarece nu mereu testul generat are un număr de întrebări multiplu de 4, am implementat o funcție care calculează procentul de întrebări corecte și dacă acesta este mai mare sau egal cu

75%, studentul trece la următorul modul. Pentru a promova testele din pagina *test.php*, studentul trebuie să obțină minim 50% + 1 întrebări corecte. Ambele tipuri de teste sunt formulare HTML care conțin întrebările și răspunsurile sunt amestecate prin intermediul metodei *shuffle()* din PHP, iar pentru fiecare răspuns este creat un input de tip *radio*.

3.4.7 Pagina *results.php*

Pagina *results.php* este pagina unde studentul poate vedea rezultatele testului de promovare la următorul modul. După ce studentul completează testul generat, formularul HTML trimite datele prin metoda POST către pagina *results.php*. Pagina are acces și la răspunsul primit de la Gemini API, ce conține întrebările și răspunsurile corecte și verifică dacă răspunsul studentului este același cu cel corect oferit de API. Răspunsurile corecte sunt stocate într-un array construit pe baza răspunsului primit de la Gemini API pentru a putea fi comparate cu răspunsurile studentului. În urma verificării răspunsurilor, extensia va calcula procentul de întrebări corecte și va afișa un mesaj, folosind funcția *notification()* din Moodle, pentru a informa studentul dacă a promovat sau nu testul. Dacă studentul a promovat testul, extensia va actualiza înregistrarea din tabelul *adaptivelearning_progress* pentru modulul curent, setând statusul la *completed* și va crea o nouă înregistrare cu statusul *inprogress* pentru următorul modul. În ambele situații în care studentul fie a promovat, fie nu a promovat testul, extensia va redirecționa studentul la pagina *learn.php* prin intermediul butonului *Continue learning*.

3.4.8 Pagina *available_modules.php*

Pagina *available_modules.php* este pagina unde studentul poate vedea toate modulele disponibile din cadrul activității. Modulele disponibile sunt cele cu statusul *completed* sau *inprogress*. Afișarea modulelor se face printr-o interogare SQL cu join între tabelele *adaptivelearning_modules* și *adaptivelearning_progress*, selectând ID-ul modulului, titlul, conținutul acestuia și statusul progresului studentului, verificând ID-ul activității curente, ID-ul studentului și statusul *completed* sau *inprogress*. Dacă studentul nu are niciun modul disponibil, extensia va afișa un mesaj folosind funcția *notification()*.

3.4.9 Pagina *finish.php*

Pagina *finish.php* este pagina finală a activității, unde studentul este înștiințat că a finalizat toate modulele din cadrul activității. Pagina afișează un mesaj pentru student și un buton de redirectionare la pagina *available_modules.php* pentru a putea vedea toate modulele disponibile.

3.5 Interacțiunea cu Gemini API

Extensia utilizează Gemini API pentru două funcționalități: generarea modulelor de învățare pe baza fișierului PDF încărcat de profesor și generarea testelor grilă pentru fiecare modul. Pentru a putea utiliza Gemini API, extensia trebuie să aibă un token de autentificare, ce este obținut prin intermediul unui cont Google Cloud Platform.

3.5.1 Generarea modulelor de învățare

Trimiterea fișierului PDF este realizată prin intermediul metodei *adaptivelearning_generate_modules_from_pdf()*, ce are ca parametru fișierul PDF încărcat în formularul de configurare a activității. Fișierul este citit din *file storage* folosind funcția *get_file_storage()* din Moodle, iar conținutul binar este codificat în Base64 pentru a fi trimis la Gemini API. În cadrul acestei funcții este contruit payload-ul JSON necesar pentru a trimite fișierul PDF în care este inclus și promptul de generare a modulelor de învățare. Apoi se face apelul HTTP POST către endpoint-ul Gemini API, folosind funcția *file_get_contents()*, ce trimite datele către Gemini API și returnează răspunsul primit. Răspunsul este apoi decodat din JSON și se apelează funcția de parsare a răspunsului pentru a extrage modulele de învățare generate. Promptul pentru generarea modulelor de învățare a evoluat pentru a obține rezultate mai bune și pentru a extrage informații relevante din fișierul PDF. Primul prompt a fost:

Împarte acest fișier PDF în module de învățare.

Acest prompt a fost prea general și nu oferea suficiente informații despre modul în care ar trebui să fie structurate modulele. Chiar dacă a fost realizată o împărțire bună a modulelor, conținutul fiecărui modul era prea scurt și era sub forma unui cuprins a ce se regăsește în acesta.

Următorul prompt încercat a fost:

Împarte acest fișier PDF în 3 module de învățare, fiecare modul va avea un titlu și un text care va explica fiecare lucru din acel modul.

Acest prompt a oferit rezultate mai bune deoarece cuprindea toate informațiile necesare, dar formularea utilizată nu era cea potrivită, deoarece răspunsul primit povestea fiecare informație în loc să fie structurată ca un modul de învățare.

Următorul prompt încercat a fost:

Împarte acest fișier PDF în 3 module de învățare, fiecare modul va avea un titlu și un text care va explica în detaliu fiecare lucru din acel modul. Textul fiecarui modul va fi formulat ca un profesor

Acest prompt a oferit rezultate foarte bune, deoarece a structurat modulele de învățare în mod corespunzător, fiecare modul având un titlu și un conținut detaliat, dar adăugarea formulării textului ca un profesor, a făcut ca răspunsul să fie formulat ca un dialog pe care profesorul îl are cu o clasă de elevi sau studenți.

Promptul final folosit pentru generarea modulelor de învățare este:

*Împarte acest fișier PDF în 3 module de învățare, fiecare modul va avea un titlu și un text care va explica în detaliu fiecare lucru din acel modul. Textul fiecarui modul va fi formulat cu un limbaj academic. Răspunsul va fi structurat astfel:
*titlu modul1*modul text1*titlu modul2*modul text2... și fara new line.*

Acest prompt oferă toate informațiile necesare pentru a obține modulele de învățare structurate corespunzător, fiecare modul având un titlu și un conținut detaliat, iar răspunsul este structurat într-un mod ușor de procesat. Răspunsul primit de la Gemini API este un string ce conține modulele separate prin *, acest răspuns fiind apoi procesat de funcția `adaptivelearning_parse_modules()`, ce împarte stringul în module și extrage titlul și conținutul fiecărui modul.

3.5.2 Generarea testelor grilă

Generarea testelor grilă se face prin intermediul funcției `adaptivelearning_generate_quiz_from_module()`, ce are ca parametru conținutul modulului curent pentru care se dorește generarea testului. Payload-ul JSON este construit similar cu cel pentru generarea modulelor de învățare, cu un prompt diferit, specific pentru generarea testelor grilă. Promptul folosit pentru generarea testelor grilă este:

*Folosind informații din text, generează un test de tip grilă cu întrebări relevante pentru studenți. Fiecare întrebare să aibă 1 răspuns corect și 3 răspunsuri greșite. Formatul răspunsului va fi exact așa: doar-textul-întrebării*răspuns-corect*răspuns-greșit*răspuns-greșit*răspuns-greșit*/*

Acest prompt oferă toate informațiile necesare pentru a obține un test de tip grilă structurat corespunzător, fiecare întrebare având un răspuns corect și trei răspunsuri greșite. Am ales să folosesc doi delimitatori, * pentru delimitarea întrebării de răspunsuri și */ pentru a delimita fiecare întrebare, deoarece acest lucru permite o procesare mai ușoară a răspunsului primit. Întrebările generate cuprind întreg modulul, iar răspunsurile sunt relevante pentru conținutul acestuia.

Capitolul 4

Cazuri de Utilizare

Definirea cazurilor de utilizare oferă o perspectivă clară asupra funcționalităților oferite de extensie și a interacțiunilor dintre utilizator și sistem. Extensia a fost proiectată pentru a automatiza și îmbunătăți procesul educațional, prin structurarea conținutului în module parcurse progresiv. Punctul de plecare al studentului este ales prin intermediul unui test inițial ce verifică cunoștințele acestuia asupra subiectului. Deciziile privind trecerea de la un modul la altul sunt luate tot în funcție de performanța studentului, determinată cu ajutorul testelor generate automat din conținutul fișierului încărcat de profesor. Acest comportament oferă o experiență de învățare personalizată și eficientă.

Cazurile de utilizare acoperă două categorii de utilizatori:

- **Profesor:** are rolul de a crea activitatea, de a încărca materialul pe baza căruia este generat conținutul educațional, de a revizui modulele generate și de a urmări progresul studenților.
- **Student:** are rolul de a accesa activitatea, de a completa un test inițial, de a urma modulele în ordine și de a rezolva noi teste destinate progresării către următorul modul.

4.1 Use Case Diagram

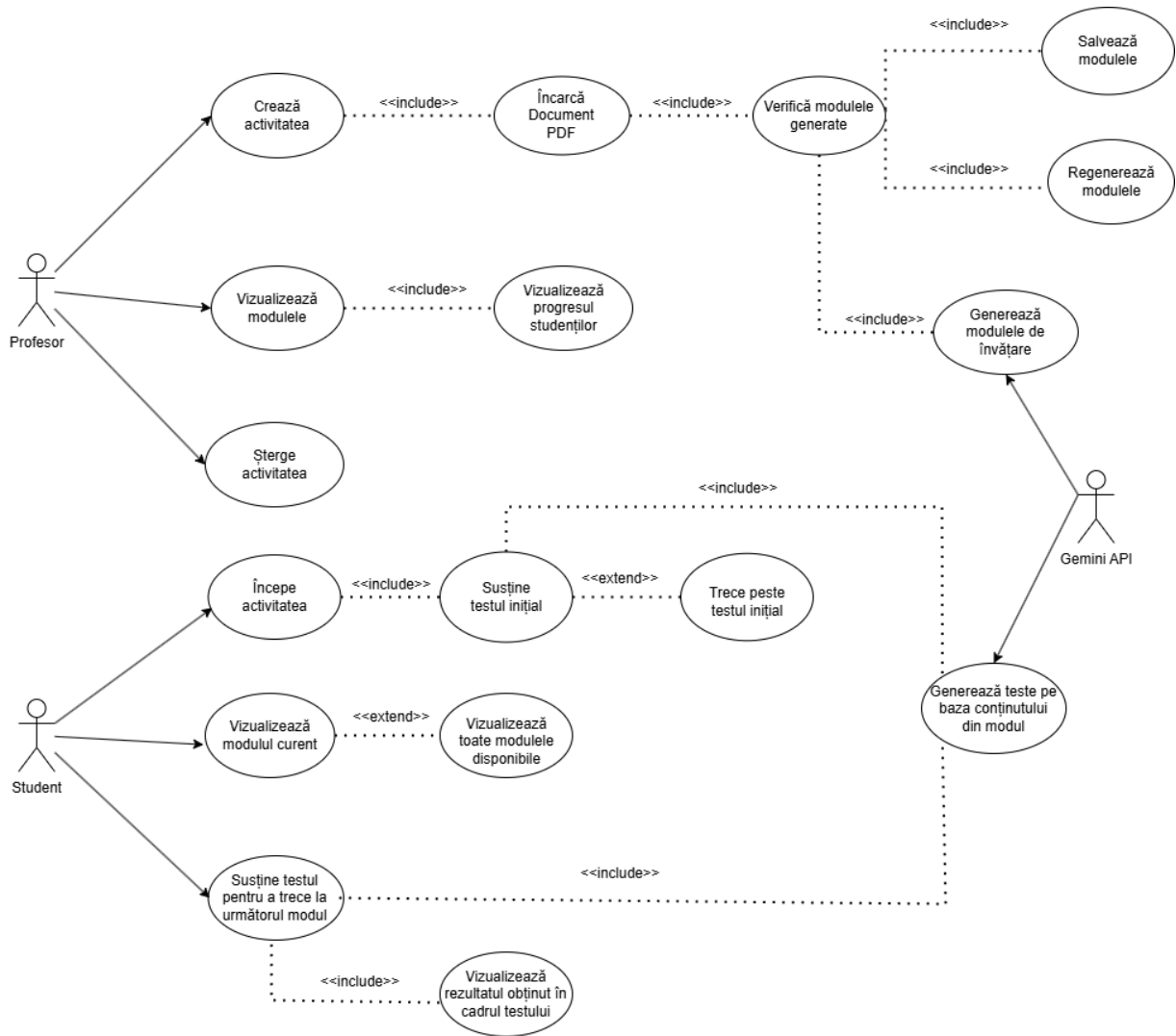


Figura 4.1: Use Case Diagram

4.2 Cazuri de utilizare pentru profesor

4.2.1 Crearea activității

Profesorul are posibilitatea de a adăuga o activitate nouă de tip *Adaptive Learning* în cadrul unui curs. Activitatea poate fi adăugată în orice secțiune a cursului și se găsește în lista de activități disponibile. Pentru a crea o activitate, profesorul trebuie să completeze un formular în care specifică numele activității, descrierea acesteia și să încarce un fișier care conține materialul educațional. Acest fișier va fi folosit ulterior pentru generarea automată a modulelor de învățare.

Name ! Romania de la statul totalitar la statul de drept

Description ! Această lecție urmărește tranziția României de la regimul comunist totalitar

Upload a PDF document Maximum file size: Unlimited, maximum number of files: 1, maximum total size: 10 MB

Files

PDF

Romania de la

Accepted file types:

PDF document .pdf

☐ Send content change notification ?

Save and return to course Save and display Cancel

Activate Windows

Figura 4.2: Crearea activității de tip Adaptive Learning

4.2.2 Revizuirea modulelor generate

După ce profesorul a încărcat fișierul, extensia va procesa conținutul, generând module de învățare cu ajutorul Gemini API. Profesorul are posibilitatea de a previzualiza modulele generate și are opțiunea de a le regenera, dacă consideră că acestea nu sunt conforme cu așteptările sau dacă dorește să facă modificări, și opțiunea de a le salva și publica studenților în cazul în care este mulțumit de acestea prin intermediul a două butoane, *Save modules* și *Regenerate modules*.

Modul 2: Transformări Politice, Economice și Culturale în Perioada Totalitară

Constituția RPR din 1948, inspirată din modelul sovietic, a instaurat un sistem politic dominat de PMR, cu o aparentă separare a puterilor în stat, dar cu o concentrare efectivă a puterii în mâinile primului-secretar al partidului. Naționalizarea industriei, colectivizarea agriculturii și planificarea centralizată au marcat transformarea radicală a economiei românești. Rezistența la colectivizare a fost înăbușită violent, iar intelectualii și opoziții regimului au fost persecutați și închiși. În plan cultural, regimul a impus controlul strict asupra educației, mass-media și artei, promovând o ideologie marxist-leninistă și cenzurând orice formă de disidență. Securitatea, poliția politică a regimului, a jucat un rol central în reprimarea oricărei forme de opoziție. Epoca Ceaușescu, care a început în 1965, a fost marcată inițial de o relativă liberalizare, urmată de o întoarcere la un control totalitar și de instaurarea unui cult al personalității liderului. Politica economică dezastruoasă și represiunea brutală au condus la o criză profundă a societății românești.

Modul 3: Tranziția către Democrație și Integrarea Europeană

Revolta din decembrie 1989, declanșată la Timișoara, a marcat sfârșitul regimului Ceaușescu și începutul tranziției României către democrație. După executarea soților Ceaușescu, conducerea provizorie a țării, reprezentată de Frontul Salvării Naționale (FSN), și-a asumat obiective majore: trecerea la economia de piață, democratizarea societății și integrarea în structurile euro-atlantice. Reînființarea partidelor istorice și apariția unor noi formațiuni politice au marcat revenirea la pluralismul politic. Adoptarea Constituției din 1991 a pus bazele unui stat de drept, garantând drepturile și libertățile fundamentale ale cetățenilor. Tranziția economică, marcată de privatizare și de liberalizarea piețelor, s-a confruntat cu numeroase dificultăți, inclusiv creșterea șomajului și inegalitățile sociale. Alegerile libere din 1990, câștigate de Ion Iliescu și FSN, au marcat începutul unei noi ere în istoria României. Guvernele post-comuniste s-au confruntat cu provocări complexe, inclusiv gestionarea moștenirii regimului totalitar, reconstrucția economică și consolidarea democrației. Integrarea României în NATO și Uniunea Europeană a reprezentat o etapă importantă în procesul de modernizare și de ancorare a țării în valorile occidentale.

Save modules Regenerate modules

Figura 4.3: Revizuirea modulelor generate

4.2.3 Vizualizarea modulelor și urmărirea progresului studenților

După salvarea modulelor, profesorul este redirecționat către pagina de vizualizare a modulelor, pagină ce poate fi accesată și ulterior din meniul cursului. Pe această pagină, profesorul poate vedea lista modulelor generate, cu titlul și conținutul lor. De asemenea, acesta are acces la un tabel unde poate vedea progresul studenților. Tabelul conține numele complet al studentului, în format *Prenume Nume*, Titlul și numărul modulului la care se află și ziua, data și ora la care a început modulul.

Modul 3: Economia românească în perioada postdecembristă

Perioada postdecembristă a fost marcată de tranziția de la economia centralizată la economia de piață. Acest proces a fost însoțit de fenomene economice negative, cum ar fi inflația galopantă, scăderea nivelului de trai, scăderea producției industriale, închiderea unor întreprinderi, creșterea șomajului și migrația inversă de la oraș la sat. Tranziția a fost îngreunată de lipsa de experiență în economia de piață și de corupție. Disparitățile dintre mediul urban și rural s-au accentuat, în special în ceea ce privește nivelul de educație. În mediul rural, accesul la educație a scăzut dramatic, mulți locuitori rămânând fără nicio formă de instruire.

Students Progress

Full name	Current Module	Last Attempt
Catalin Morarasu	Modul 1: Economia românească în perioada interbelică	Friday, 13 June 2025, 2:57 PM

Figura 4.4: Vizualizarea modulelor și urmărirea progresului studenților

4.2.4 Ștergerea activității

Profesorul are posibilitatea de a șterge activitatea de tip *Adaptive Learning* din curs. Această acțiune va elimina toate modulele generate și progresul studenților asociați cu această activitate. Ștergerea activității se realizează similar cu ștergerea oricărei alte activități din Moodle, prin intermediul butonului *Delete* disponibil în meniul de administrare al cursului, meniu accesibil doar utilizatorilor cu rol de profesor sau administrator ce se află în modul editare.

4.3 Cazuri de utilizare pentru student

4.3.1 Începerea activității

Pentru student, activitățile de tip *Adaptive Learning* sunt afișate în lista de activități a cursului. Când studentul accesează activitatea pentru prima dată, acesta este redirecționat la pagina de start a activității, unde vede titlul activității, și o descriere

a activităților de tip *Adaptive Learning*. Butonul *Start Learning* este disponibil pentru a începe activitatea și pentru a trece la testul inițial.

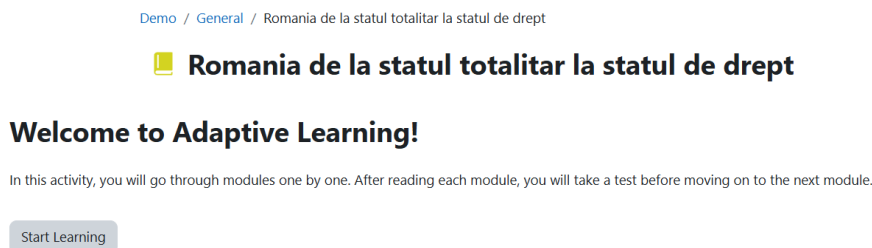


Figura 4.5: Pagina de start a activității de tip Adaptive Learning

4.3.2 Completarea testului inițial

După ce studentul apasă butonul *Start Learning*, acesta este redirecționat către pagina de test inițial ce conține întrebări din materialul educațional încărcat de profesor. Testul inițial este generat automat de Gemini API și are rolul de a evalua cunoștințele studentului asupra subiectului. Prima oară studentul primește un test cu întrebări doar din primul modul, iar dacă acesta obține un scor mai mare sau egal cu 75% va putea accesa testul următorului modul. Dacă studentul nu obține un scor suficient de mare, acesta va rămâne la același modul și va fi redirecționat către pagina de vizualizare a modulului. În cadrul fiecărui test, studentul are posibilitatea de a trece peste test cu ajutorul butonului *Skip test*, caz în care va fi redirecționat către pagina de vizualizare a modulului pentru care era destinat testul. Testul nu are un număr fix de întrebări, API-ul generând câte întrebări consideră că sunt necesare pentru a evalua cunoștințele studentului și a cuprinde toate conceptele importante din modulul respectiv, iar fiecare întrebare are patru variante de răspuns, dintre care doar una este corectă. După ce studentul a răspuns la toate întrebările, acesta apasă butonul *Submit test* pentru a trimite răspunsurile.

Test de inițializare: Modul 1 - Contextul instaurării regimului totalitar în România

Întrebarea 1: În ce an a preluat PCR puterea deplină în România?

- ☐ 1948
- ☐ 1947
- ☐ 1946
- ☐ 1945

Întrebarea 2: Cine a condus guvernul instaurat în martie 1945?

- ☐ Gheorghe Gheorghiu-Dej
- ☐ Regele Mihai I
- ☐ Ion Antonescu
- ☐ Petru Groza

Întrebarea 3: Ce eveniment a marcat proclamarea Republicii Populare Române?

- ☐ Fuziunea PCR cu PSD
- ☐ Abdicarea forțată a Regelui Mihai I
- ☐ Instaurarea guvernului Petru Groza
- ☐ Alegerile din 1946

Întrebarea 4: Cum s-a numit partidul format prin fuziunea PCR cu PSD?

- ☐ Partidul Comunist Român (PCR)
- ☐ Partidul Social Democrat (PSD)
- ☐ Partidul Muncitoresc Român (PMR)
- ☐ Partidul Socialist Român

Întrebarea 5: Ce model politic a implementat PMR în România?

- ☐ Monarhie constituțională
- ☐ Republică parlamentară
- ☐ Democrație populară
- ☐ Democrație liberală

Submit answers

Skip test

Figura 4.6: Pagina de test inițial

4.3.3 Vizualizarea modulelor

După completarea testului inițial, studentul este redirecționat către pagina de vizualizare a modulului curent, pagina fiind acceseabilă și ulterior din meniul cursului. Pe această pagină, studentul poate vedea titlul și conținutul modulului curent. De asemenea, studentul are la îndemână două butoane: *Take Test* și *See available modules*. Butonul *Take Test* îl redirecționează către pagina de test pentru modulul curent, iar butonul *See available modules* îl redirecționează către pagina de vizualizare a tuturor modulelor disponibile, adică cele prin care deja a trecut și cel curent.

Demo / General / Romania de la statul totalitar la statul de drept

Romania de la statul totalitar la statul de drept

Modul 1: Contextul Internațional și Instaurarea Regimului Totalitar în România

Contextul internațional postbelic, marcat de ascensiunea URSS și de înțelegerile dintre Marile Puteri, a avut un impact decisiv asupra destinului României. Influența sovietică a facilitat ascensiunea Partidului Comunist Român (PCR), care a orchestrat o preluare treptată a puterii. Prin intermediul unor tactici precum infiltrarea guvernului, controlul ministerelor cheie și falsificarea alegerilor din 1946, comuniștii au marginalizat forțele politice democratice. Abdicarea forțată a Regelui Mihai I în 1947 a marcat sfârșitul monarhiei și instaurarea Republicii Populare Române (RPR), un stat satelit al URSS. Sub conducerea lui Gheorghe Gheorghiu-Dej, PCR, fuzionat cu Partidul Social Democrat (PSD) pentru a forma Partidul Muncitoresc Român (PMR), a implementat modelul sovietic de "democrație populară".

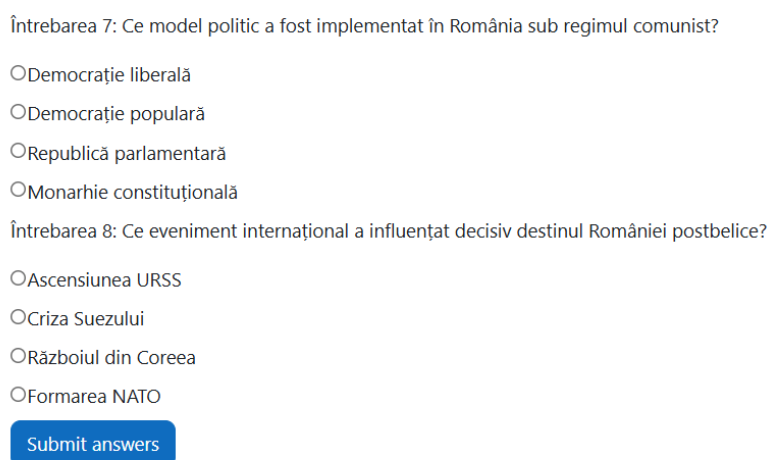
Take Test

See available modules

Figura 4.7: Pagina de vizualizare a modulului curent

4.3.4 Completarea testului pentru a trece la următorul modul

După ce studentul a citit conținutul modulului curent, acesta poate accesa testul asociat cu modulul respectiv prin intermediul butonului *Take Test*. Testul este generat automat de Gemini API și conține întrebări din materialul educațional al modulului curent. Similar cu testul inițial, acesta nu are un număr fix de întrebări, iar fiecare întrebarea are patru variante de răspuns, dintre care doar una este corectă. După ce studentul a răspuns la toate întrebările, acesta apasă butonul *Submit test* pentru a trimite răspunsurile și este redirecționat la pagina unde poate vedea rezultatul obținut la test.



Întrebarea 7: Ce model politic a fost implementat în România sub regimul comunist?

- ☐Democrație liberală
- ☐Democrație populară
- ☐Republică parlamentară
- ☐Monarhie constituțională

Întrebarea 8: Ce eveniment internațional a influențat decisiv destinul României postbelice?

- ☐Ascensiunea URSS
- ☐Criza Suezului
- ☐Războiul din Coreea
- ☐Formarea NATO

[Submit answers](#)

Figura 4.8: Pagina de test pentru modulul curent

4.3.5 Vizualizarea rezultatului obținut la test

După ce studentul a trimis răspunsurile la test, acesta este redirecționat către pagina de vizualizare a rezultatului obținut. Aici, studentul vede toate întrebările, răspunsurile sale și răspunsurile corecte. De asemenea studentul vede și scorul obținut la test, iar dacă acesta este mai mare sau egal cu 50% + 1, studentul promovează la următorul modul. Indiferent de rezultatul obținut, butonul *Continue learning* trimite studentul la pagina de vizualizare a aceluiași modul dacă nu a promovat, sau la pagina de vizualizare a următorului modul dacă a promovat.

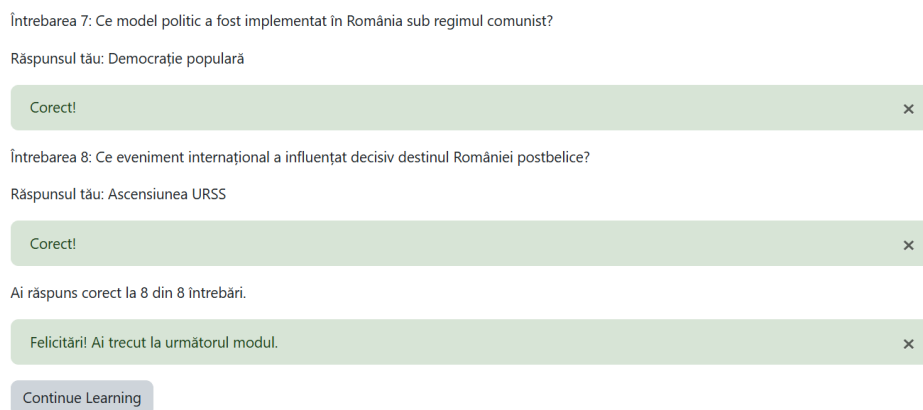


Figura 4.9: Pagina de vizualizare a rezultatului obținut la test

4.3.6 Vizualizarea tuturor modulelor disponibile

Pe pagina de vizualizare a modulului curent, studentul are posibilitatea de a accesa pagina de vizualizare a tuturor modulelor disponibile prin intermediul butonului *See available modules*. Pagina afișează toate modulele prin care studentul deja a trecut, precum și cel curent. Fiecare modul are titlul și conținutul său, iar pentru a se întoarce la modulul curent studentul poate apăsa butonul *Back*.



Figura 4.10: Pagina de vizualizare a tuturor modulelor disponibile

4.3.7 Finalizarea activității

După ce studentul a parcurs toate modulele disponibile, acesta este trimis la pagina de finalizare a activității, unde poate vedea un mesaj de felicitare și un buton care îl redirecționează la pagina de vizualizarea a tuturor modulelor disponibile, în cazul

de față toate modulele din cadrul activității. Studentul poate accesa această pagină și ulterior, din meniul cursului.

[Demo](#) / [General](#) / Economie Rurala/Urbana

Economie Rurala/Urbana

Congratulations!

You finished all the modules from this activity. Well done!

[See all modules](#)

Figura 4.11: Pagina de finalizare a activității

Concluzii

.

Bibliografie

- 1 D. B. A. Prof. Dr. Yoesoep Edhie Rachmad, *Adaptive Learning Theory*, 2024, doi: 10.17605/OSF.IO/VFZ38.
- 2 P. Brusilovsky și E. Millán, *User Models for Adaptive Hypermedia and Adaptive Educational Systems*, în *The Adaptive Web*, vol. 4321, P. Brusilovsky, A. Kobsa, și W. Nejdl, Ed., în *Lecture Notes in Computer Science*, vol. 4321. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 3–53. doi: 10.1007/978-3-540-72079-9_1.
- 3 *Moodle architecture - MoodleDocs* [Online]. Disponibil la: https://docs.moodle.org/dev/Moodle_architecture
- 4 *MariaDB Knowledge Base*, MariaDB KnowledgeBase [Online]. Disponibil la: <https://mariadb.com/kb/en/>
- 5 *Introducing Gemini: our largest and most capable AI model*, Google [Online]. Disponibil la: <https://blog.google/technology/ai/google-gemini-ai/>
- 6 *Generate content with the Vertex AI Gemini API — Generative AI on Vertex AI*, Google Cloud [Online]. Disponibil la: <https://cloud.google.com/vertex-ai/generative-ai/docs/model-reference/inference>
- 7 *PHP Introduction* [Online]. Disponibil la: https://www.w3schools.com/php/php_intro.asp
- 8 M. Dougiamas și P. Taylor, *Moodle: Using Learning Communities to Create an Open Source Course Management System*, prezentat la EdMedia + Innovate Learning, Association for the Advancement of Computing in Education (AACE), 2003, pp. 171–178 [Online]. Disponibil la: <https://www.learntechlib.org/primary/p/13739/>
- 9 E. Costello, *Opening up to open source: looking at how Moodle was adopted in higher education*, *Open Learning: The Journal of Open, Distance and e-Learning*, vol. 28,

nr. 3, pp. 187–200, nov. 2013, doi: 10.1080/02680513.2013.856289.

- 10 *XAMPP Installers and Downloads for Apache Friends* [Online]. Disponibil la: <https://www.apachefriends.org/>
- 11 I. Poghosyan, *ADVANTAGE OF INSTALLING MOODLE LEARNING MANAGEMENT SYSTEM UNDER IIS/MARIADB OVER APACHE/MYSQL*, Main Issues Of Pedagogy And Psychology, vol. 11, nr. 2, Art. nr. 2, nov. 2024, doi: 10.24234/miopap.v11i2.49.
- 12 G. Lang, T. Triantoro, și J. Sharp, *Large Language Models as AI-Powered Educational Assistants: Comparing GPT-4 and Gemini for Writing Teaching Cases*, Journal of Information Systems Education, vol. 35, nr. 3, pp. 390–407, sep. 2024, doi: <https://doi.org/10.62273/YCIJ6454>.
- 13 L. Team și colab., *Evaluating Gemini in an arena for learning*, 30 mai 2025, arXiv: arXiv:2505.24477. doi: 10.48550/arXiv.2505.24477.
- 14 *Common files — Moodle Developer Resources* [Online]. Disponibil la: <https://moodledev.io/docs/5.1/apis/commonfiles>
- 15 *Activity modules — Moodle Developer Resources* [Online]. Disponibil la: <https://moodledev.io/docs/5.1/apis/plugin/types/mod>
- 16 *Using XMLDB - MoodleDocs* [Online]. Disponibil la: https://docs.moodle.org/dev/Using_XMLDB
- 17 *Moodle PHP Documentation: moodleform_mod Class Reference* [Online]. Disponibil la: https://phpdoc.moodledev.io/4.1/d2/d19/classmoodleform_mod.html
- 18 *PHP: Sessions - Manual* [Online]. Disponibil la: <https://www.php.net/manual/en/book.session.php>