

Proiect 2 MIP

Prompturi AI

Prompt 1: Structura aplicației

Generează o aplicație .NET MAUI numită **ChimieQuiz** care permite testarea cunoștințelor de chimie prin întrebări grilă.

Aplicația trebuie să conțină:

- o pagină principală (Dashboard) pentru alegerea capitolului
- o pagină de quiz care afișează întrebări grilă
- o pagină de rezultate care afișează scorul final

Aplicația trebuie scrisă în **C# și XAML**, cu navigare simplă între pagini și date încărcate din fișiere JSON locale.

Prompt 2: Modelele de date

Creează modelele de date necesare pentru o aplicație de quiz la chimie în .NET MAUI: model **Question** cu:

- Id
- Chapter
- Text
- Options (exact 5 variante)
- CorrectIndex (0–4)
- model **QuizAnswer** care stochează:
 - QuestionId
 - SelectedIndex
 - IsCorrect

Include reguli de validare pentru date și oferă un exemplu de fișier JSON cu întrebări.

Prompt 3: Încărcarea întrebărilor din JSON

Implementează mecanismul de încărcare a întrebărilor dintr-un fișier JSON local (chimie_questions.json) folosind MauiAsset.

Întrebările trebuie:

- deserializate în obiecte Question
- disponibile pentru filtrare după capitol
- reutilizabile atât în QuizPage, cât și în ResultsPage

Prompt 4: DashboardPage (pagina principală)

Creează **DashboardPage**, pagina principală a aplicației:

- afișează capitolele de chimie disponibile (extrase din JSON)
- permite utilizatorului să selecteze un capitol
- include un buton „Start Quiz”
- la apăsarea butonului, navighează către QuizPage și transmite capitolul selectat

Prompt 5: Logica de rulare a quizului

Implementează logica principală a quizului:

- afișează întrebările una câte una
- menține indexul întrebării curente
- stochează răspunsul selectat pentru fiecare întrebare

- NU afișează feedback imediat (corect/greșit)
- la final, trimite toate răspunsurile către pagina de rezultate

Prompt 6: QuizPage (interfața utilizator)

Construiește interfața QuizPage:

- afișează textul întrebării
- afișează 5 butoane pentru răspunsuri (A–E)
- evidențiază răspunsul selectat
- afișează progresul (ex: 2 / 5)
- include buton „Next” pentru navigare între întrebări

Prompt 7: Calcularea rezultatelor

Implementează logica de evaluare a quizului:

- compară răspunsurile utilizatorului cu CorrectIndex

Calculează:

- numărul de răspunsuri corecte
- scorul final
- procentajul obținut
- identifică întrebările greșite pentru afișare separată

Prompt 8: ResultsPage (pagina de rezultate)

Creează ResultsPage care:

- afișează scorul și procentajul

listează întrebările greșite cu:

- textul întrebării
- răspunsul utilizatorului
- răspunsul corect

include butoane:

- „Retry wrong answers”
- „Back to Dashboard”

Prompt 9: Reluarea întrebărilor greșite

Implementează funcționalitatea „Retry wrong answers”:

- permite refacerea quizului doar cu întrebările greșite
- transmite ID-urile întrebărilor greșite către QuizPage
- reîncarcă doar acele întrebări
- recalculează rezultatele după reluare

Prompt 10: Stilizare și consistență UI

Aplică un stil vizual unitar aplicației:

- paletă de culori roșu / roz / portocaliu
- butoane mari, colțuri rotunjite
- layout aerisit și lizibil
- stiluri reutilizabile pentru butoane, texte și containere