

Prompt Engineering — Experiența echipei în dezvoltarea aplicației *DressUp Web Application*

În cadrul acestui proiect, am lucrat în echipă de 5 la o aplicație web de tip dress-up, unde utilizatorii își pot personaliza avatarul, crea outfit-uri, participa la concursuri de modă și interacționa prin like-uri și comentarii. Platforma a fost dezvoltată în ASP.NET cu C# și Visual Studio, iar pentru managementul task-urilor am folosit Jira.

Pe parcursul dezvoltării, am apelat destul de des la modele de limbaj de tip LLM — în special **ChatGPT**, **Gemini**, **GitHub Copilot** și **Groq**, fiecare ajutându-ne în feluri diferite.

Cum am folosit LLM-urile în proiect

ChatGPT

Probabil cel mai folosit model pe parcursul proiectului a fost **ChatGPT**.

- Pe parcursul dezvoltării aplicației, am folosit ChatGPT ca ajutor pentru înțelegerea mai bună a anumitor concepte din ASP.NET Core MVC și pentru a obține sugestii de rezolvare a unor erori întâlnite în cod. Ne-a ghidat mai ales în structurarea anumitor metode din controller, în utilizarea UserManager și în înțelegerea legăturii dintre View și Controller.
- Am avut destule situații în care build-ul nu mergea sau apăreau erori după ce făceam **push/pull/merge din GitHub**. De exemplu, ni s-a întâmplat să avem conflicte pe fișiere după ce lucram pe aceleași, dar am reușit să evităm asta lucrând pe branch-uri. Am pus erorile în ChatGPT și de multe ori ne-a explicat clar ce s-a întâmplat și cum s-o rezolvăm.
- Tot cu ajutorul lui am reparat un bug: **"Edit Profile" Does Not Save Changes**. Practic, formularul de editare afișa datele corect, dar nu le salva în baza de date. Am descoperit că problema era din cauza faptului că proprietatea *UserAchievements* din modelul User era obligatorie, iar formularul de editare profil nu o popula. Soluția a

fost să facem *UserAchievements* opțional în model, iar după asta *ModelState.IsValid* a trecut, datele s-au salvat corect și redirect-ul a funcționat.

GitHub Copilot

Am folosit **GitHub Copilot** direct în **Visual Studio 2022**, pe partea de completări de cod a fost fantastic. De fiecare dată când lucram la metode repetitive, Copilot oferea sugestii de cod pe care le puteai accepta rapid. A fost de mare ajutor mai ales în controllere, paginile de administrare pentru concursuri și fișierele html și css, stilizarea devenind mult mai plăcută atunci când nu trebuie să știi exact ordinea atributelor.

Ce am observat e că, spre deosebire de ChatGPT, Copilot nu explică ce face, dar când ai nevoie de completări rapide, fără să ieși din IDE, e perfect.

Gemini

Am folosit **Gemini (Google)** în special pentru zona de **User Interface/User Experience** și text content. De exemplu, atunci când am lucrat la pagina de home și la secțiunea de concursuri, Gemini ne-a ajutat să generăm texte de prezentare atractive, sugestii de categorii pentru outfit-uri și chiar descrieri pentru avataruri publice.

De asemenea, am cerut idei de cum ar trebui organizat dulapul pe categorii de haine (accesorii, bluze, pantaloni) și cum să arătăm în mod clar articolele disponibile într-un mod responsive pe orice device.

Groq (pentru chatbot)

Pentru **chatbot-ul de personalizare a outfit-urilor**, ne-am dorit ceva rapid și flexibil. Aici am ales să lucrăm cu **Groq**, care e recunoscut pentru viteza de răspuns a LLM-urilor. Am configurat un flux conversațional în care utilizatorii pot primi sugestii de haine sau accesorii în funcție de tema concursului sau sezon. De exemplu, dacă un utilizator întreba „Ce outfit să aleg pentru concursul de vară?”, chatbot-ul returna combinații recomandate pe baza item-urilor din dulap. Groq a fost super util pentru că oferea răspunsuri aproape instant și putea procesa rapid variante multiple de outfit-uri.

Concluzie

Experiența noastră cu LLM-urile a fost una pozitivă: ne-au ajutat să economisim timp, să înțelegem mai bine problemele apărute și să venim cu soluții rapide. Tot LLM-urile ne-au

ajutat să scriem **teste automate** pentru funcțiile de creare, ștergere și editare de outfit-uri și pentru verificarea sistemului de like-uri.

- **ChatGPT** a fost de ajutor pentru debugging și generare de cod cu explicații.
- **Copilot** a accelerat scrierea de cod în Visual Studio.
- **Gemini** ne-a oferit idei de design și conținut text.
- **Groq** a fost ideal pentru chatbot-ul rapid și interactiv.