

## ANEXA 1: Prompt-uri IAG

ChatGPT-3.5	Gemini	Copilot
Q-1. Cunoști principiile de inginerie software și cum se aplică acestea în cadrul proiectelor de dezvoltare software a aplicațiilor web?		
<p>Da, pot discuta principiile de inginerie software și cum se aplică acestea în proiectele de dezvoltare a aplicațiilor web.</p> <p>1. <b>**Principiul modularității**</b>: Acesta se referă la împărțirea aplicației în module independente, care pot fi dezvoltate, testate și întreținute separat. În cadrul aplicațiilor web, aceasta se traduce adesea prin utilizarea unor framework-uri sau biblioteci pentru gestionarea diferitelor aspecte ale aplicației, cum ar fi interfața cu utilizatorul, logica de afaceri și accesul la date.</p> <p>2. <b>**Principiul separării preocupărilor (Separation of Concerns - SoC)**</b>: Acest principiu se referă la separarea diferitelor aspecte ale aplicației pentru a facilita dezvoltarea, testarea și întreținerea. În cadrul unei aplicații web, acest lucru poate însemna separarea logicii de prezentare (HTML, CSS, JavaScript) de logica de afaceri și de accesul la date.</p> <p>3. <b>**Principiul deschiderii/închiderii (Open/Closed Principle)**</b>: Acest principiu afirmă că o clasă trebuie să fie deschisă pentru extensie, dar închisă pentru modificare. În cadrul aplicațiilor web, acest lucru poate însemna utilizarea de design patterns și framework-uri care facilitează extinderea funcționalității fără a modifica codul existent.</p> <p>4. <b>**Principiul unicității responsabilității (Single Responsibility Principle - SRP)**</b>: Acesta afirmă că o clasă ar trebui să aibă o singură responsabilitate și să o îndeplinească bine. În cadrul dezvoltării web, aceasta poate însemna crearea de clase și funcții care se ocupă doar de un aspect specific al aplicației.</p> <p>5. <b>**Principiul încapsulării**</b>: Acest principiu implică încadrarea datelor și funcțiilor asociate într-un singur</p>	<p>Da, cunosc principiile ingineriei software și modul în care se aplică în proiectele de dezvoltare software a aplicațiilor web. Iată o prezentare generală:</p> <p>Principii fundamentale:</p> <ul style="list-style-type: none"><li>• Separarea preocupărilor: Implementarea diverselor aspecte ale unei aplicații web (prezentare, logică, date) în module separate și bine definite.</li><li>• Modularitate: Dezvoltarea de componente independente și reutilizabile care pot fi combinate pentru a crea diverse funcționalități.</li><li>• Coerență: Asigurarea compatibilității și interoperabilității dintre diferitele componente ale aplicației web.</li><li>• Testabilitate: Scrierea codului în mod modular și testabil pentru a facilita identificarea și remedierea erorilor.</li><li>• Performanță: Optimizarea codului și a infrastructurii pentru a obține timpi de răspuns rapizi și o utilizare eficientă a resurselor.</li><li>• Securitate: Implementarea de măsuri de securitate adecvate pentru a proteja datele și a preveni accesul neautorizat.</li></ul> <p>Aplicare la dezvoltarea web:</p> <ul style="list-style-type: none"><li>• Arhitectura web: Utilizarea de modele de arhitectură web bine definite, cum ar fi MVC sau REST, pentru a organiza componentele aplicației.</li><li>• Cadre de dezvoltare: Utilizarea de cadre de dezvoltare web populare, cum ar fi Django, Spring Boot sau Ruby on Rails, pentru a facilita și a accelera procesul de dezvoltare.</li><li>• Metodologii Agile: Implementarea de metodologii Agile, cum ar fi Scrum sau Kanban, pentru a gestiona proiectele de dezvoltare web în mod iterativ și incremental.</li><li>• Controlul versiunilor: Utilizarea de sisteme de control al versiunilor, cum ar fi Git, pentru a urmări modificările</li></ul>	<p>Desigur! Ingineria software este un domeniu esențial în dezvoltarea aplicațiilor software și se bazează pe principii și practici riguroase. Iată câteva aspecte importante:</p> <p>1. Definirea Ingineriei Software:</p> <ul style="list-style-type: none"><li>◦ Ingineria software se referă la procesul de dezvoltare, gestionare și mentenanță a software-ului. Acest proces se desfășoară în conformitate cu un plan sistematic și bine stabilit.</li><li>◦ Ingineria software acoperă toate aspectele producției de software și se bazează pe discipline precum științele aplicate, ingineria științelor aplicate, management, managementul de proiect, managementul calității, ergonomia software și ingineria de sistem.</li></ul> <p>2. Rolul Ingineriei Software în Proiectele de Dezvoltare a Aplicațiilor Web:</p> <ul style="list-style-type: none"><li>◦ Ingineria software este crucială în crearea și menținerea aplicațiilor web. Principiile sale se aplică în fiecare etapă a dezvoltării:</li><li>▪ Analiza și Specificarea Cerințelor: Inginerii software identifică și documentează cerințele funcționale și non-funcționale ale aplicației web.</li><li>▪ Design și Arhitectură: Se creează modele de design și se planifică structura aplicației.</li><li>▪ Implementare și Testare: Codul este scris și testat pentru a asigura funcționalitatea corectă.</li><li>▪ Managementul Dezvoltării: Se urmărește progresul, se gestionează resursele și se respectă termenele limită.</li><li>▪ Asigurarea Calității: Se aplică teste și se verifică calitatea software-ului.</li><li>▪ Mentenanță și Îmbunătățiri: Software-ul este actualizat și optimizat pe parcursul utilizării.</li></ul> <p>3. Principii de Inginerie Software:</p> <ul style="list-style-type: none"><li>◦ Separarea Preocupărilor: Divizarea funcționalităților în module separate pentru a facilita dezvoltarea și întreținerea.</li></ul>

ChatGPT-3.5	Gemini	Copilot
<p>obiect, limitând accesul la acestea din exteriorul obiectului. În cadrul aplicațiilor web, acest lucru poate fi realizat prin utilizarea de clase și module care expun interfețe clare și restricționează accesul la detalii interne.</p> <p>Aceste principii, împreună cu alte concepte de inginerie software, cum ar fi testarea unitară, integrarea continuă și livrarea continuă, sunt fundamentale în dezvoltarea aplicațiilor web robuste, scalabile și ușor de întreținut.</p> <p>Scor răspuns: 3/5</p> <p>Răspunsul se referă la dezvoltarea efectivă a aplicației și nu la principiile proiectului, deși unele se aplică în ambele situații.)</p>	<p>codului și a facilita colaborarea.</p> <ul style="list-style-type: none"> <li>• Testare: Efectuarea de teste amănunțite ale aplicației web, inclusiv teste unitare, teste de integrare și teste de acceptare, pentru a asigura calitatea și fiabilitatea.</li> </ul> <p>Resurse suplimentare:</p> <ul style="list-style-type: none"> <li>• Principii de inginerie software: [URL-ul nevalid a fost eliminat]</li> <li>• Dezvoltare web: <a href="https://ro.wikipedia.org/wiki/Dezvoltare_web">https://ro.wikipedia.org/wiki/Dezvoltare_web</a></li> <li>• Metodologii Agile: <a href="https://en.wikipedia.org/wiki/Agile_software_development">https://en.wikipedia.org/wiki/Agile_software_development</a></li> <li>• Controlul versiunilor: <a href="https://en.wikipedia.org/wiki/Version_control">https://en.wikipedia.org/wiki/Version_control</a></li> <li>• Testarea software: <a href="https://en.wikipedia.org/wiki/Software_testing">https://en.wikipedia.org/wiki/Software_testing</a></li> </ul> <p>Rețineți că această listă nu este exhaustivă și că principiile specifice de inginerie software aplicate pot varia în funcție de complexitatea și specificul fiecărui proiect de dezvoltare web.</p> <p>Scor răspuns: 5/5</p>	<ul style="list-style-type: none"> <li>◦ Modularitate: Construirea software-ului din componente independente.</li> <li>◦ Abstractizarea: Crearea de niveluri de abstractizare pentru a ascunde detalii complexe.</li> <li>◦ Anticiparea Schimbării: Proiectarea software-ului pentru a face față modificărilor viitoare.</li> <li>◦ Dezvoltare Incrementală: Construirea software-ului în etape mici și adăugarea graduală de funcționalități.</li> <li>◦ Consistența: Aplicarea unor standarde și reguli uniforme în întregul proiect.</li> </ul> <p>4. Rolul Inginerului Software:</p> <ul style="list-style-type: none"> <li>◦ Inginerul software aplică aceste principii în structura și procesul de dezvoltare a software-ului.</li> <li>◦ El sau ea se concentrează pe crearea unui produs software de calitate, care să îndeplinească cerințele și să ofere o experiență eficientă utilizatorilor.</li> </ul> <p>În concluzie, ingineria software este esențială pentru proiectele de dezvoltare a aplicațiilor web și contribuie la crearea unor produse fiabile și performante. 🚀</p> <p>Scor răspuns: 4/5</p>
<p>Q-2. Având în vedere principiile anterioare care sunt pașii pe care i-ai parcurge pentru a dezvolta o aplicație web cu următoarele cerințe? Aplicația va fi folosită de trei categorii de utilizatori: studenți, secretar și administrator. În afară de utilizatorii cu rol administrator care se vor conecta pe bază de utilizator și parolă, ceilalți utilizatori se vor conecta cu Google Sign In. Adresele de email ale utilizatorilor vor fi încărcate la instalarea aplicației de către administrator pe baza unui fișier Excel. Acesta va putea, din lista de adrese să desemneze care utilizatori vor îndeplini ulterior rolul de secretar sau secretară din lista de utilizatori. Utilizatorii de tip student vor înregistra cereri de eliberare adeverințe prin completarea unui formular Google unde vor introduce doar motivul cererii și adresa de email. Informațiile din formular vor fi, ulterior, consultate de către aplicație și încărcate într-o bază de date separată. Utilizatorii cu rol secretar se vor conecta tot pe baza Google SignIn însă la o interfață proprie în care vor vedea o listă cu cererile înregistrate de studenți și va putea aproba sau respinge cererea. În cazul respingerii cererii va trebui să furnizeze un motiv. Administratorii vor avea, și ei, acces la interfața secretarului și, în plus, vor putea să introducă detaliile de identificare ale facultății, să introducă fișierul Excel pentru lista inițială de utilizatori și să realizeze operațiuni de salvare a bazei de date și de reinițializare a întregului sistem.</p>		
<p>Pentru a dezvolta o astfel de aplicație web, urmând principiile de inginerie software menționate anterior, iată pașii pe care i-aș urma:</p> <p>1. <b>**Analiza cerințelor și proiectarea sistemului**</b>:</p> <ul style="list-style-type: none"> <li>- Identificarea și definirea tuturor cerințelor menționate în descrierea aplicației.</li> </ul>	<p>Pașii pentru dezvoltarea aplicației web:</p> <ol style="list-style-type: none"> <li>1. Planificare și proiectare: <ul style="list-style-type: none"> <li>• Definirea detaliată a cerințelor funcționale și non-funcționale.</li> <li>• Proiectarea arhitecturii aplicației web, a fluxurilor de lucru și a interfețelor de utilizator pentru studenți, secretari și administratori.</li> <li>• Identificarea instrumentelor și</li> </ul> </li> </ol>	<p>Desigur! Dezvoltarea unei aplicații web cu cerințele specificate implică mai mulți pași. Iată cum aș aborda această sarcină:</p> <ol style="list-style-type: none"> <li>1. Definirea Cerințelor și Analiza: <ul style="list-style-type: none"> <li>◦ Începe prin a înțelege în detaliu cerințele aplicației. Ai deja o descriere generală, dar trebuie să clarifici aspecte precum funcționalitățile exacte, fluxurile de</li> </ul> </li> </ol>

ChatGPT-3.5	Gemini	Copilot
<ul style="list-style-type: none"> <li>- Proiectarea arhitecturii sistemului, includerea separării preocupărilor și modularizării.</li> <li>- Definirea bazelor de date necesare și a relațiilor între acestea.</li> <li>- Proiectarea interfeței utilizatorului pentru fiecare tip de utilizator.</li> </ul> <p>2. <b>**Implementarea autentificării și autorizării**</b>:</p> <ul style="list-style-type: none"> <li>- Implementarea autentificării cu Google Sign-In pentru utilizatorii non-administratori.</li> <li>- Implementarea autentificării bazate pe utilizator și parolă pentru administratori.</li> <li>- Implementarea mecanismelor de autorizare pentru a asigura că fiecare utilizator are acces doar la funcționalitățile corespunzătoare rolului său.</li> </ul> <p>3. <b>**Încărcarea inițială a utilizatorilor și a datelor despre facultate**</b>:</p> <ul style="list-style-type: none"> <li>- Dezvoltarea unei funcționalități pentru administratori pentru a încărca adresele de email ale utilizatorilor din fișierul Excel.</li> <li>- Implementarea unei funcționalități pentru introducerea detaliilor de identificare ale facultății de către administratori.</li> </ul> <p>4. <b>**Implementarea funcționalității pentru înregistrarea cererilor de eliberare a adeverințelor**</b>:</p> <ul style="list-style-type: none"> <li>- Dezvoltarea unui formular Google pentru utilizatorii de tip student pentru a înregistra cererile.</li> <li>- Implementarea unei interfețe pentru procesarea și stocarea datelor din formular în baza de date.</li> </ul> <p>5. <b>**Interfața secretarului pentru aprobarea/rejetarea cererilor**</b>:</p> <ul style="list-style-type: none"> <li>- Crearea unei interfețe pentru secretari pentru a vedea lista cererilor înregistrate.</li> <li>- Implementarea funcționalității de aprobare sau respingere a cererilor de către secretari.</li> <li>- Asigurarea că secretarii pot furniza motive în cazul respingerii cererilor.</li> </ul> <p>6. <b>**Interfața administratorului pentru administrarea sistemului**</b>:</p> <ul style="list-style-type: none"> <li>- Dezvoltarea unei interfețe pentru administratori pentru a accesa funcționalitățile specifice, cum ar fi gestionarea utilizatorilor și a datelor de identificare ale facultății.</li> </ul>	<p>tehnologiilor necesare (framework web, biblioteci, baze de date).</p> <p>2. Dezvoltare:</p> <ul style="list-style-type: none"> <li>• Implementarea front-end-ului studenților și secretarilor folosind HTML, CSS și JavaScript.</li> <li>• Implementarea back-end-ului cu framework-ul web ales (ex: Django, Spring Boot) pentru gestionarea logicii aplicației, a conexiunii la baza de date și a integrării cu Google Sign In.</li> <li>• Crearea interfeței administratorului cu posibilitatea de a configura detaliile facultății, de a încărca fișierul Excel cu utilizatori și de a administra baza de date.</li> <li>• Implementarea funcționalităților de completare a formularului Google de către studenți și de vizualizare/aprobare/respingere a cererilor de către secretari.</li> <li>• Asigurarea securității aplicației prin implementarea de autentificare și autorizare.</li> </ul> <p>3. Testare:</p> <ul style="list-style-type: none"> <li>• Efectuarea de teste unitare, de integrare și de acceptare pentru a verifica corectitudinea funcționalităților aplicației.</li> <li>• Testarea performanței și a scalabilității aplicației.</li> <li>• Remedierea erorilor identificate în faza de testare.</li> </ul> <p>4. Implementare:</p> <ul style="list-style-type: none"> <li>• Instalarea aplicației web pe un server web.</li> <li>• Importarea datelor inițiale din fișierul Excel.</li> <li>• Configurarea accesului pentru utilizatori (studenți, secretari, administratori).</li> </ul> <p>5. Mentenanță:</p> <ul style="list-style-type: none"> <li>• Monitorizarea performanței aplicației.</li> <li>• Remedierea erorilor care pot apărea după lansare.</li> <li>• Efectuarea de actualizări și îmbunătățiri ale aplicației.</li> </ul> <p>Conșiderații suplimentare:</p> <ul style="list-style-type: none"> <li>• Implementarea unui sistem de notificări prin email pentru a informa studenții cu privire la stadiul cererilor lor.</li> <li>• Generarea de rapoarte statistice pentru administratori.</li> <li>• Asigurarea accesibilității</li> </ul>	<p>lucru și interacțiunile dintre utilizatori.</p> <ul style="list-style-type: none"> <li>◦ Identifică entitățile principale: studenți, secretari și administratori.</li> </ul> <p>2. Autentificarea Utilizatorilor:</p> <ul style="list-style-type: none"> <li>◦ Pentru utilizatorii cu rol de administrator, implementează autentificarea pe bază de utilizator și parolă.</li> <li>◦ Pentru ceilalți utilizatori (studenți și secretari), folosește Google Sign-In. Acest lucru simplifică procesul de autentificare și permite utilizatorilor să se conecteze cu conturile Google.</li> </ul> <p>3. Încărcarea Adreselor de Email din Fișier Excel:</p> <ul style="list-style-type: none"> <li>◦ Administratorul poate încărca adresele de email ale utilizatorilor dintr-un fișier Excel.</li> <li>◦ Aceste adrese vor fi stocate într-o bază de date sau altă formă de stocare.</li> </ul> <p>4. Roluri și Permisii:</p> <ul style="list-style-type: none"> <li>◦ Administratorul poate desemna utilizatorii care vor îndeplini rolul de secretar sau secretară.</li> <li>◦ Atribuire permisiuni specifice fiecărui rol (de exemplu, dreptul de a aproba sau respinge cereri).</li> </ul> <p>5. Înregistrarea Cererilor de Eliberare Adeverințe:</p> <ul style="list-style-type: none"> <li>◦ Studenții pot completa un formular Google cu motivul cererii și adresa de email.</li> <li>◦ Informațiile din formular vor fi stocate temporar într-o bază de date separată.</li> </ul> <p>6. Interfața Secretarului:</p> <ul style="list-style-type: none"> <li>◦ Secretarii se vor conecta cu Google Sign-In la o interfață dedicată.</li> <li>◦ Vor vedea o listă cu cererile înregistrate de studenți.</li> <li>◦ Pot aproba sau respinge cererile, furnizând un motiv în cazul respingerii.</li> </ul> <p>7. Interfața Administratorului:</p> <ul style="list-style-type: none"> <li>◦ Administratorii au acces la interfața secretarului.</li> <li>◦ În plus, ei pot:</li> </ul> <ul style="list-style-type: none"> <li>▪ Introduce detaliile de identificare ale facultății.</li> <li>▪ Încărca fișierul Excel pentru lista inițială de utilizatori.</li> <li>▪ Realiza operațiuni de salvare a bazei de date și reinițializare a întregului sistem.</li> </ul> <p>8. Implementarea Bazei de Date:</p>

ChatGPT-3.5	Gemini	Copilot
<p>- Implementarea funcționalității de salvare a bazei de date și de reinițializare a sistemului.</p> <p>7. <b>**Testarea și optimizarea**</b>:</p> <p>- Testarea fiecărei funcționalități pentru a asigura funcționarea corectă și fără erori.</p> <p>- Optimizarea performanței și a securității aplicației.</p> <p>- Efectuarea testelor de integrare pentru a asigura funcționarea corectă a întregului sistem.</p> <p>8. <b>**Implementarea și lansarea în producție**</b>:</p> <p>- Asigurarea că toate funcționalitățile sunt complet funcționale și gata pentru producție.</p> <p>- Implementarea sistemului pe un server web și lansarea în producție.</p> <p>- Monitorizarea și întreținerea continuă a aplicației după lansare.</p> <p>Răspuns ChatGPT (4/5) - a crescut calitatea răspunsului din contextul suplimentar furnizat.</p>	<p>aplicației pentru utilizatorii cu dizabilități.</p> <p>Resurse utile:</p> <ul style="list-style-type: none"> <li>• [URL-ul nevalid a fost eliminat]</li> <li>•</li> </ul> <p><a href="https://en.wikipedia.org/wiki/Web_development">https://en.wikipedia.org/wiki/Web_development</a></p> <ul style="list-style-type: none"> <li>• [URL-ul nevalid a fost eliminat]</li> </ul> <p>Notă: Această descriere este o prezentare generală a pașilor implicați în dezvoltarea aplicației web. Detaliile specifice pot varia în funcție de complexitatea și specificul proiectului.</p> <p>Răspuns Gemini (5/5)</p>	<ul style="list-style-type: none"> <li>◦ Creează o bază de date pentru a stoca informațiile relevante (adrese de email, cereri, roluri etc.).</li> <li>◦ Asigură că baza de date este securizată și eficientă.</li> </ul> <p>9. Testarea și Optimizarea:</p> <ul style="list-style-type: none"> <li>◦ Testează fiecare funcționalitate pentru a verifica că totul funcționează corect.</li> <li>◦ Optimizează performanța aplicației și asigură că nu există vulnerabilități de securitate.</li> </ul> <p>10. Documentarea și Implementarea:</p> <ul style="list-style-type: none"> <li>◦ Documentează fiecare pas și funcționalitate.</li> <li>◦ Implementează aplicația pe un server web și asigură-te că este disponibilă pentru utilizatori.</li> </ul> <p>În timpul dezvoltării, asigură-te că respecti cele mai bune practici de inginerie software și că ai în vedere securitatea și scalabilitatea aplicației. 🚀</p> <p>Răspuns Copilot (4/5)</p>
<p>Q-3. Pentru dezvoltarea aplicației am decis să folosesc următoarele tehnologii: Google pentru autentificarea utilizatorilor non-administratori conform cerințelor, Google Forms pentru preluarea cererilor în baza de date a aplicației, Next.JS, Nest.JS, Sqlite3 și TypeORM. În acest scenariu poți crea codul necesar care să fie folosit la autentificarea utilizatorilor de tip secretar știind că Google ca autentifica utilizatorul pe baza adresei de email iar în baza de date a aplicației vom avea o coloană rol în tabelul cu utilizatori?</p>		
<p>Da, cu siguranță! Iată un exemplu de cod pentru autentificarea utilizatorilor de tip secretar folosind Google Sign-In și tehnologiile menționate:</p> <pre>``typescript // Importăm bibliotecile și modulele necesare import { Injectable } from '@nestjs/common'; import { InjectRepository } from '@nestjs/typeorm'; import { Repository } from 'typeorm'; import { User } from './user.entity'; // presupunând că avem o entitate User definită în aplicație  @Injectable() export class AuthService {   constructor(     @InjectRepository(User)</pre>	<p>Codul pentru autentificarea utilizatorilor de tip secretar:</p> <p>1. Implementarea front-end-ului:</p> <ul style="list-style-type: none"> <li>• Folosind Next.JS, creai o pagină de login unde utilizatorii secretari vor introduce adresa lor de email.</li> <li>• Implementați Google Sign In pe pagina de login.</li> <li>• După autentificarea cu Google, obțineți adresa de email a utilizatorului din obiectul de răspuns Google.</li> </ul> <p>2. Verificarea rolului utilizatorului:</p> <ul style="list-style-type: none"> <li>• Folosiți Nest.JS ca server API pentru a primi adresa de email de la front-end.</li> <li>• Implementați o funcție care se conectează la baza de date Sqlite3 folosind TypeORM.</li> <li>• Interogați baza de date pentru a obține rolul utilizatorului asociat adresei de email.</li> </ul>	<p>Desigur! Pentru a crea codul necesar pentru autentificarea utilizatorilor de tip secretar, vom folosi Google Sign-In pentru a autentifica utilizatorii pe baza adresei de email. De asemenea, vom adăuga o coloană "rol" în tabelul cu utilizatori din baza de date a aplicației.</p> <p>Iată pașii pe care îi vom urma:</p> <ol style="list-style-type: none"> <li>1. Configurarea Autentificării Google:       <ul style="list-style-type: none"> <li>◦ În primul rând, asigură-te că ai configurat autentificarea Google pentru aplicația ta. Acest lucru implică crearea unui proiect în Google Cloud Console și activarea serviciului Google Sign-In.</li> <li>◦ Obține cheile API necesare pentru a face apeluri către serviciul Google Sign-In.</li> </ul> </li> <li>2. Implementarea Autentificării Google în Aplicație:</li> </ol>

ChatGPT-3.5	Gemini	Copilot
<pre> private readonly userRepository: Repository&lt;User&gt;, ) {} // Metodă pentru autentificarea utilizatorului de tip secretar folosind Google Sign-In                                 async authenticateWithGoogle(email: string): Promise&lt;User&gt; {     // Verificăm dacă utilizatorul există     în baza de date     const user = await this.userRepository.findOne({ email });     // Dacă utilizatorul nu există, returnăm null     if (!user) {         return null;     }     // Dacă utilizatorul există și are rolul de secretar, returnăm obiectul utilizatorului     if (user.role === 'secretar') {         return user;     }     // Dacă utilizatorul există dar nu are rolul de secretar, returnăm null     return null; } }, ... </pre> <p>Acest exemplu presupune că avem deja configurată integrarea cu Google pentru autentificarea utilizatorilor non-administratori și că avem o entitate User definită în aplicație, cu o coloană `role` care indică rolul utilizatorului (secretar, student, etc.). Această metodă `authenticateWithGoogle` verifică dacă utilizatorul există în baza de date și are rolul de secretar. Dacă da, returnează obiectul utilizatorului; în caz contrar, returnează `null`. Această metodă ar putea fi apoi folosită în cadrul unui serviciu de gestionare a autentificării pentru a verifica și autoriza accesul utilizatorilor de tip secretar în aplicație.</p> <p>Răspuns ChatGPT (3/5)</p>	<ul style="list-style-type: none"> <li>• Returnați un răspuns JSON către front-end care conține rolul utilizatorului.</li> </ul> <p>3. Redirecționarea utilizatorului:</p> <ul style="list-style-type: none"> <li>• Pe baza rolului obținut din baza de date, redirecționați utilizatorul către interfața specifică: <ul style="list-style-type: none"> <li>◦ Secretar: Redirecționați către interfața de vizualizare/aprobare/respingere a cererilor.</li> <li>◦ Non-secretar: Afișați un mesaj de eroare care specifică că utilizatorul nu are rolul necesar.</li> </ul> </li> </ul> <p>Exemplu de cod:</p> <p>JavaScript</p> <pre> // Front-end (Next.JS) const handleLogin = async (email) =&gt; {     // Autentificare Google     const googleResponse = await googleSignIn();     // Obținerea adresei de email     const email = googleResponse.profileObj.email;     // Trimiterea adresei de email către API     const response = await fetch("/api/auth/secretary", {         method: "POST",         body: JSON.stringify({ email }),     });     // Decodarea răspunsului JSON     const { role } = await response.json();     // Redirecționarea utilizatorului     if (role === "secretary") {         router.push("/secretary");     } else {         // Afișarea unui mesaj de eroare         alert("Utilizatorul nu are rolul necesar!");     } } </pre> <p>// Back-end (Nest.JS)</p> <pre> @Injectable() export class AuthService {     constructor(         @InjectRepository(User) private readonly userRepository: UserRepository,     ) {}     async authenticateSecretary(email: string): Promise&lt;{ role: string }&gt; {         const user = await </pre>	<ul style="list-style-type: none"> <li>◦ În codul tău, vei utiliza biblioteca sau SDK-ul Google Sign-In pentru a permite utilizatorilor să se autentifice cu conturile Google.</li> <li>◦ Când un utilizator se autentifică, vei primi un token de autentificare care conține informații despre utilizator (cum ar fi adresa de email).</li> </ul> <p>3. Verificarea Adresei de Email și Atribuirea Rolului:</p> <ul style="list-style-type: none"> <li>◦ După ce primești tokenul de autentificare, verifică adresa de email asociată.</li> <li>◦ În baza de date a aplicației, adaugă o coloană “rol” în tabelul cu utilizatori. Această coloană poate avea valori precum “student”, “secretar” sau “administrator”.</li> <li>◦ Pe baza adresei de email, atribuie utilizatorului rolul corespunzător.</li> </ul> <p>4. Exemplu de Cod (Pseudo-Cod):</p> <ul style="list-style-type: none"> <li>◦ Mai jos este un exemplu de pseudo-cod pentru autentificarea utilizatorilor de tip secretar:</li> </ul> <pre> // Exemplu de cod în Node.js folosind Express și Passport.js (pentru autentificare) const express = require('express'); const passport = require('passport'); const GoogleStrategy = require('passport-google- oauth20').Strategy; const app = express(); // Configurare Google Strategy passport.use(new GoogleStrategy({     clientID: 'YOUR_CLIENT_ID',     clientSecret: 'YOUR_CLIENT_SECRET',     callbackURL: '/auth/google/callback' // Ruta de callback }),     (accessToken, refreshToken, profile, done) =&gt; {         // profile.emails[0].value conține adresa de email         const userEmail = profile.emails[0].value;         const userRole = getUserRoleFromDatabase(userEmail); // Funcție pentru a obține rolul din baza de date          // Salvare utilizator în sesiune done(null, { email: userEmail, role: userRole });     } </pre>

ChatGPT-3.5	Gemini	Copilot
	<pre> this.userRepository.findOne({ where: { email } });      if (!user) {         throw new Error("Utilizatorul nu a fost găsit!");     }     return { role: user.role }; } } // Controler API (Nest.JS) @Controller("/api/auth") export class AuthController {     constructor(private readonly authService: AuthService) {}     @Post("/secretary")     async authenticateSecretary(@Body() { email }: { email: string }) {         return await this.authService.authenticateSecretar y(email);     } } </pre> <p>Notă:</p> <ul style="list-style-type: none"> <li>• Acest cod este un exemplu simplificat și poate necesita adaptări specifice în funcție de implementarea specifică a aplicației.</li> <li>• Asigurați-vă că implementați măsuri de securitate adecvate pentru a proteja datele utilizatorilor.</li> </ul> <p>Resurse suplimentare:</p> <ul style="list-style-type: none"> <li>• [URL-ul nevalid a fost eliminat]</li> <li>• [URL-ul nevalid a fost eliminat]</li> <li>• [URL-ul nevalid a fost eliminat]</li> </ul> <p>Răspuns Gemini (4/5)</p>	<pre> )); // Rute pentru autentificare app.get('/auth/google',     passport.authenticate('google', { scope: ['profile', 'email'] }) ); app.get('/auth/google/callback',     passport.authenticate('google', { failureRedirect: '/login' }),     (req, res) =&gt; {         // Redirecționează către pagina principală sau altă pagină         res.redirect('/');     } ); // Funcție pentru a verifica dacă utilizatorul are rolul de secretar function isSecretary(req, res, next) {     if (req.user &amp;&amp; req.user.role === 'secretar') {         return next();     }     res.status(403).send('Acces interzis.');</pre> <p>}</p> <p>// Exemplu de rută protejată pentru secretari</p> <pre> app.get('/secretary-dashboard', isSecretary, (req, res) =&gt; {     res.send('Bun venit, secretar!'); }); // Alte rute și configurații... app.listen(3000, () =&gt; {     console.log('Serverul rulează pe portul 3000'); }); </pre> <p>5. Observații:</p> <ul style="list-style-type: none"> <li>◦ În exemplul de cod, <code>getUserRoleFromDatabase</code> este o funcție fictivă care ar trebui să interogheze baza de date pentru a obține rolul utilizatorului pe baza adresei de email.</li> <li>◦ Asigură-te că gestionezi corect erorile și că protejezi rutele relevante (cum ar fi panoul secretarului) pentru a permite accesul doar utilizatorilor cu rolul</li> </ul> <p>Răspuns Copilot (5/5)</p>
<p>Q-4. Folosind aceleași tehnologii menționate mai sus scrie un cod pentru o rutină automată prin care, la final de zi, la ora 17:00 toți utilizatorii cu rol administrator să primească un raport Excel cu toate adeverințele emise în ziua respectivă. Coloanele sunt: numele și prenumele solicitantului, adresa de email, motivul cererii, stare: dacă a fost aprobată sau nu, numărul cererii: în cazul în care fost aprobată sau textul „respinsă” dacă a fost respinsă și motivul.</p>		

ChatGPT-3.5	Gemini	Copilot
<p>Pentru a realiza această funcționalitate, putem folosi un mix de tehnologii precum Nest.js pentru gestionarea serverului și a cronului, TypeORM pentru interacțiunea cu baza de date, și o bibliotecă pentru generarea de fișiere Excel, cum ar fi `exceljs`. Iată un exemplu de cum ai putea realiza această rutină automată:</p> <ol style="list-style-type: none"> <li>Mai întâi, instalează biblioteca `exceljs`:  <pre>``bash npm install exceljs ``</pre> </li> <li>Apoi, scrie o metodă într-un serviciu Nest.js care va fi apelată la ora specificată pentru a genera raportul Excel:</li> </ol> <pre>``typescript // Importăm bibliotecile și modulele necesare import { Injectable } from '@nestjs/common'; import * as ExcelJS from 'exceljs'; import { InjectRepository } from '@nestjs/typeorm'; import { Repository } from 'typeorm'; import { Cerere } from './cerere.entity'; // presupunând că avem o entitate Cerere definită în aplicație @Injectable() export class RaportService {   constructor(     @InjectRepository(Cerere)     private readonly cerereRepository: Repository&lt;Cerere&gt;,   ) {}   async generateDailyReport() {     // Obținem toate cererile emise în ziua respectivă     const today = new Date();     today.setHours(0, 0, 0, 0); // setăm ora la 00:00:00     const cereri = await this.cerereRepository.find({       where: {         data: today,       },     });      // Creăm un nou workbook Excel     const workbook = new ExcelJS.Workbook();     const worksheet =</pre>	<p>Cod pentru trimiterea automată a raportului zilnic:</p> <ol style="list-style-type: none"> <li>Implementarea funcției de generare a raportului:       <ul style="list-style-type: none"> <li>Folosiți Nest.JS pentru a crea o funcție care se conectează la baza de date SQLite3 folosind TypeORM.</li> <li>Interogați baza de date pentru a obține toate cererile de adeverințe din ziua respectivă.</li> <li>Generați un fișier Excel cu coloanele: nume, prenume, email, motiv, stare (aprobată/respinsă), număr cerere/motiv respingere.</li> <li>Salvați fișierul Excel pe server.</li> </ul> </li> <li>Implementarea programării cron:       <ul style="list-style-type: none"> <li>Folosiți o bibliotecă de programare cron (ex: cron) pentru a configura o sarcină care să ruleze zilnic la ora 17:00.</li> <li>Apelați funcția de generare a raportului din cadrul sarcinii cron.</li> </ul> </li> <li>Trimiterea emailului:       <ul style="list-style-type: none"> <li>Folosiți o bibliotecă de trimitere emailuri (ex: Nodemailer) pentru a trimite fișierul Excel ca atașament către toți utilizatorii cu rol administrator.</li> <li>Includeți un mesaj clar și concis în corpul emailului care descrie conținutul raportului.</li> </ul> </li> </ol> <p>Exemplu de cod:</p> <pre>JavaScript // Back-end (Nest.JS)  @Injectable() export class ReportService {   constructor(     @InjectRepository(User) private readonly userRepository: UserRepository,     @InjectRepository(Request) private readonly requestRepository: RequestRepository,   ) {}   async generateDailyReport() {     const today = new Date();     const requests = await this.requestRepository.find({       where: {         createdAt: {           gte: new Date(today.getFullYear(), today.getMonth(), today.getDate(), 0, 0, 0),           lte: new Date(today.getFullYear(),</pre>	<p>Desigur! Pentru a crea o rutină automată care să trimită un raport Excel la finalul zilei, vom folosi Node.js împreună cu Google Sheets API pentru a accesa și actualiza datele într-un fișier Google Sheets. Vom crea un script care să ruleze la ora 17:00 și să genereze raportul solicitat.</p> <p>Iată un exemplu de cod pentru această rutină:</p> <pre>// Instalează pachetele necesare folosind npm install googleapis google-auth-library const { google } = require('googleapis'); const fs = require('fs'); // Autentificare folosind fișierul JSON cu credențiale const credentials = require('./path/to/your/credentials.json'); const SCOPES = [   'https://www.googleapis.com/auth/spreadsheets.readonly' ]; const TOKEN_PATH = 'token.json'; // Funcție pentru a citi datele din Google Sheets async function readDataFromSheet() {   {     const auth = await authorize(credentials);     const sheets = google.sheets({ version: 'v4', auth });     try {       const response = await sheets.spreadsheets.values.get({         spreadsheetId: 'YOUR_SPREADSHEET_ID',         range: 'Sheet1!A2:F', // Intervalul de celule cu date (fără antet)       });       const rows = response.data.values;       if (rows.length) {         // Procesează datele și generează raportul Excel         const report = generateExcelReport(rows);         // Trimite raportul către utilizatorii cu rol de administrator         sendReportToAdmins(report);       } else {         console.log('Nu s-au găsit date în Google Sheets.');  </pre>

ChatGPT-3.5	Gemini	Copilot
<pre> workbook.addWorksheet('Raport cereri'); // Adăugăm antetul worksheet.addRow(['Nume', 'Prenume', 'Adresă de email', 'Motiv cerere', 'Stare', 'Număr cerere', 'Motiv respingere']);  // Adăugăm fiecare cerere în fișierul Excel cereri.forEach(cerere =&gt; { worksheet.addRow([ cerere.umeSolicitant, cerere.prenumeSolicitant, cerere.emailSolicitant, cerere.motiv, cerere.aprobata ? 'Aprobată' : 'Respinsă', cerere.nrCerere    'N/A', cerere.motivRespins    'N/A', ]); }); // Generăm fișierul Excel const buffer = await workbook.xlsx.writeBuffer(); // Aici poți trimite fișierul Excel pe email către toți administratorii sau îl poți salva pe server // De exemplu, poți folosi o bibliotecă pentru trimiterea de email- uri, cum ar fi @nestjs-modules/mailer // sau poți salva fișierul pe server folosind Node.js // Exemplu de trimitere email: // mailerService.sendMail({ // // to: 'administrator@example.com', // subject: 'Raport cereri zilnic', // attachments: [ // { // content: buffer, // filename: 'raport_cereri.xlsx', // contentType: 'application/vnd.openxmlformats- officedocument.spreadsheetml.sheet', // }, // ], // }); } }, ... </pre> <p>3. Folosește un modul Nest.js pentru a programa execuția rutinei la ora specificată. Poți folosi `nestjs-schedule` pentru aceasta:</p>	<pre> today.getMonth(), today.getDate(), 23, 59, 59), }, }, }); const data = requests.map((request) =&gt; ({ name: request.user.name, surname: request.user.surname, email: request.user.email, reason: request.reason, status: request.approved ? "Aprobată" : "Respinsă", requestNumber: request.approved ? request.id : request.rejectionReason, })); const excel = generateExcel(data); await saveFile(excel, "daily- report.xlsx"); const users = await this.userRepository.find({ where: { role: "administrator" } }); const emails = users.map((user) =&gt; user.email); await sendEmail(emails, "Raport zilnic adevărat", "Vă rugăm să găsiți atașat raportul zilnic cu adevărurile emise.", "daily- report.xlsx"); } // Funcție de programare cron const cronJob = new cron.CronJob("0 17 * * *", async () =&gt; { await reportService.generateDailyReport(); }); cronJob.start(); Notă: • Acest cod este un exemplu simplificat și poate necesita adaptări specifice în funcție de implementarea specifică a aplicației. • Asigurați-vă că configurați serverul de email corect și că aveți permisiunea de a trimite emailuri. Resurse suplimentare: • [URL-ul nevalid a fost eliminat] • [URL-ul nevalid a fost eliminat] • [URL-ul nevalid a fost eliminat] </pre> <p>Răspuns Gemini (4/5)</p>	<pre> datelor din Google Sheets:', err); } } // Funcție pentru a autoriza utilizatorul async function authorize(credentials) { // Implementează logica de autorizare // ... // Returnează obiectul de autentificare return auth; }  // Funcție pentru a genera raportul Excel function generateExcelReport(data) { // Implementează logica de generare a raportului Excel // ... // Returnează raportul sub formă de fișier Excel return excelReport; }  // Funcție pentru a trimite raportul către utilizatorii cu rol de administrator function sendReportToAdmins(report) { // Implementează logica de trimitere a raportului // ... console.log('Raportul a fost trimis către administratori.');</pre> <p>// Rulează funcția la ora 17:00  setTimeout(() =&gt; {    readDataFromSheet();  }, 17 * 60 * 60 * 1000); // 17:00 în  milisecunde  // Alte funcții și configurații...  Asigură-te că înlocuiești valorile  corespunzătoare pentru  YOUR_SPREADSHEET_ID și  implementezi logica specifică pentru  generarea raportului Excel și  trimiterea acestuia către  administratori. Acest exemplu este o  bază de plecare și poate fi adaptat la  nevoile tale specifice. 🚀</p> <p>Răspuns Copilot (4/5 – prea general)</p>



ChatGPT-3.5	Gemini	Copilot
<pre> ``typescript // Importăm biblioteca necesară import { Module } from '@nestjs/common'; import { ScheduleModule } from '@nestjs/schedule'; import { RaportService } from './raport.service';  @Module({     imports: [ScheduleModule.forRoot()],     providers: [RaportService], }) export class AppModule {} `` </pre> <p>Acest cod va programa execuția metodei `generateDailyReport` din `RaportService` la ora 17:00 în fiecare zi. Notă: asigură-te că serviciul rulează continuu pentru a permite programarea task-ului. Acesta este doar un exemplu de cod. Va trebui să ajustezi și să adaptezi aceste fragmente la structura și nevoile specifice ale aplicației tale. De asemenea, asigură-te că ai configurat corect modulele și serviciile Nest.js și că baza de date este conectată și configurată în mod corespunzător.</p> <p>Răspuns ChatGPT (5/5)</p>		
<p>Q-5. În contextul acestui proiect ce sugestii poți aduce pentru a crește productivitatea activității utilizatorilor cu rol de secretar și a face experiența de lucru cât mai plăcută?</p>		
<p>Există câteva sugestii pe care le poți lua în considerare pentru a crește productivitatea utilizatorilor cu rol de secretar și pentru a îmbunătăți experiența lor de lucru:</p> <ol style="list-style-type: none"> <li><b>**Interfață intuitivă și prietenoasă**</b>: Asigură-te că interfața utilizatorului pentru secretari este intuitivă și ușor de navigat. Folosește design-ul de experiență a utilizatorului (UX) pentru a crea fluxuri de lucru eficiente și pentru a minimiza confuzia.</li> <li><b>**Funcționalități de căutare și filtrare**</b>: Oferă secretarilor opțiuni de căutare și filtrare pentru a găsi rapid cererile pe care trebuie să le</li> </ol>	<p>Sugestii pentru a crește productivitatea secretarilor:</p> <ol style="list-style-type: none"> <li>Interfață intuitivă și ușor de utilizat: <ul style="list-style-type: none"> <li>Creați o interfață clară și organizată, cu o navigare intuitivă și un design plăcut.</li> <li>Folosiți elemente vizuale și pictograme pentru a facilita înțelegerea funcționalităților.</li> <li>Implementați o funcție de căutare rapidă pentru a găsi cu ușurință cererile de adevărințe.</li> </ul> </li> <li>Automatizarea sarcinilor repetitive: <ul style="list-style-type: none"> <li>Automatizați procesul de validare a cererilor prin implementarea de</li> </ul> </li> </ol>	<p>Desigur! Pentru a crește productivitatea și a face experiența de lucru a secretarilor cât mai plăcută, iată câteva sugestii:</p> <ol style="list-style-type: none"> <li>Comunicare Eficientă: <ul style="list-style-type: none"> <li>Folosiți comunicarea verbală și scrisă pentru a vă coordona cu colegii, clienții și managerii. Răspundeți prompt la e-mailuri și asigurați-vă că mesajele sunt clare și concise.</li> <li>Utilizați instrumente de comunicare precum Slack sau Microsoft Teams pentru a colabora eficient cu echipa.</li> </ul> </li> <li>Gestionarea Timpului: <ul style="list-style-type: none"> <li>Planificați-vă activitățile</li> </ul> </li> </ol>

ChatGPT-3.5	Gemini	Copilot
<p>gestioneze. Aceasta poate include căutarea după nume de solicitant, dată a cererii sau statusul acesteia.</p> <p>3. <b>**Notificări și alerte**</b>: Implementează sistem de notificări și alerte pentru a informa secretarii despre cererile noi sau despre cererile care necesită acțiune imediată. Acest lucru îi va ajuta să rămână la curent cu activitățile lor fără a fi nevoie să verifice în mod constant aplicația.</p> <p>4. <b>**Automatizarea sarcinilor repetitive**</b>: Identifică sarcinile repetitive pe care le au secretarii și încearcă să le automatizezi pe cât posibil. De exemplu, poți implementa un sistem de răspuns automatizat pentru cererile standard sau poți folosi tehnici de inteligență artificială pentru a prelua și procesa informații din cereri.</p> <p>5. <b>**Integrarea cu alte instrumente și servicii**</b>: Dacă secretarii folosesc deja alte instrumente sau servicii în cadrul activităților lor zilnice, încearcă să integrezi aplicația ta cu acestea pentru a simplifica fluxul de lucru. De exemplu, poți integra calendarul Google pentru a programa întâlniri sau poți sincroniza datele cu un sistem de gestionare a documentelor.</p> <p>6. <b>**Feedback și îmbunătățiri continue**</b>: Colectează feedback-ul utilizatorilor și folosește-l pentru a îmbunătăți constant aplicația. Înțelege nevoile și problemele cu care se confruntă secretarii și adaptează funcționalitățile aplicației în consecință.</p> <p>7. <b>**Suport tehnic și formare**</b>: Oferă secretarilor acces la suport tehnic și la resurse de formare pentru a-i ajuta să înțeleagă și să folosească eficient aplicația. Asigură-te că există documentație clară și instrucțiuni pentru utilizare disponibile pentru ei.</p> <p>Implementarea acestor sugestii ar putea contribui semnificativ la creșterea productivității și a satisfacției utilizatorilor cu rol de secretar în cadrul proiectului tău de dezvoltare a aplicației web.</p> <p>Răspuns ChatGPT (5/5)</p>	<p>reguli predefinite.</p> <ul style="list-style-type: none"> <li>• Creați șabloane de email personalizabile pentru a răspunde rapid la solicitări comune.</li> <li>• Integrați un sistem de notificare pentru a informa secretarii cu privire la cererile noi sau actualizate.</li> </ul> <p>3. Acces facil la informații:</p> <ul style="list-style-type: none"> <li>• Creați o bază de cunoștințe cu informații utile și instrucțiuni detaliate.</li> <li>• Oferiți acces la o bibliotecă de documente și formulare relevante.</li> <li>• Implementați un sistem de chat intern pentru a facilita comunicarea cu administratorii.</li> </ul> <p>4. Personalizarea experienței:</p> <ul style="list-style-type: none"> <li>• Permiteți secretarilor să personalizeze interfața în funcție de preferințe.</li> <li>• Oferiți posibilitatea de a configura notificări și alerte personalizate.</li> <li>• Implementați un sistem de feedback pentru a colecta opinii și sugestii de la utilizatori.</li> </ul> <p>5. Îmbunătățirea securității:</p> <ul style="list-style-type: none"> <li>• Implementați autentificare multi-factor pentru a proteja accesul la interfața secretarilor.</li> <li>• Criptați datele sensibile din baza de date.</li> <li>• Respectați regulile GDPR privind protecția datelor personale.</li> </ul> <p>6. Oferirea de instruire:</p> <ul style="list-style-type: none"> <li>• Creați materiale de instruire detaliate pentru a familiariza secretarii cu funcționalitățile aplicației.</li> <li>• Oferiți sesiuni de instruire online sau offline pentru a facilita învățarea.</li> <li>• Creați o platformă de suport cu resurse utile și tutoriale video.</li> </ul> <p>7. Implementarea de sondaje:</p> <ul style="list-style-type: none"> <li>• Realizați sondaje periodice pentru a evalua satisfacția secretarilor cu privire la experiența de utilizare a aplicației.</li> <li>• Colectați feedback și sugestii pentru îmbunătățirea funcționalităților.</li> <li>• Implementați modificările solicitate de utilizatori pentru a crește satisfacția și productivitatea.</li> </ul> <p>Considerații suplimentare:</p> <ul style="list-style-type: none"> <li>• Asigurați-vă că serverul web are o performanță bună pentru a evita</li> </ul>	<p>zilnice și prioritizați sarcinile importante.</p> <ul style="list-style-type: none"> <li>◦ Folosiți tehnici precum metoda Pomodoro (17 minute de lucru concentrat urmate de 3 minute de pauză) pentru a menține concentrarea și a preveni oboseala.</li> </ul> <p>3. Automatizarea și Șabloane:</p> <ul style="list-style-type: none"> <li>◦ Utilizați șabloane pentru e-mailuri, documente și alte sarcini repetitive.</li> <li>◦ Automatizați procesele folosind instrumente precum Zapier sau IFTTT.</li> </ul> <p>4. Organizarea Spațiului de Lucru:</p> <ul style="list-style-type: none"> <li>◦ Mențineți biroul curat și organizat. Acest lucru vă va ajuta să găsiți rapid documente și să vă concentrați mai bine.</li> <li>◦ Etichetați dosarele și fișierele în mod coerent pentru a facilita căutarea.</li> </ul> <p>5. Îmbunătățirea Abilităților Tehnice:</p> <ul style="list-style-type: none"> <li>◦ Învățați să utilizați instrumente software precum Microsoft Office, Google Workspace și alte aplicații relevante.</li> <li>◦ Dezvoltați-vă abilitățile de tastare și de luare a notițelor.</li> </ul> <p>6. Gestionarea Stresului și Pauzele:</p> <ul style="list-style-type: none"> <li>◦ Faceți pauze regulate pentru a vă relaxa și a vă reîncărca.</li> <li>◦ Practicați tehnici de reducere a stresului, cum ar fi respirația profundă sau meditația.</li> </ul> <p>7. Formare Continuă:</p> <ul style="list-style-type: none"> <li>◦ Participați la cursuri sau seminarii pentru a vă îmbunătăți abilitățile și a vă menține la curent cu noile tehnologii și tendințe.</li> </ul> <p>8. Feedback și Îmbunătățire Constantă:</p> <ul style="list-style-type: none"> <li>◦ Solicitați feedback de la colegi și superiori pentru a vă îmbunătăți continuu performanța.</li> <li>◦ Fiți deschiși la schimbare și adaptați-vă la nevoile organizației.</li> </ul> <p>În final, amintiți-vă că productivitatea și satisfacția la locul de muncă depind și de un echilibru sănătos între muncă și viața personală. Asigurați-vă că vă acordați timp pentru relaxare și reîncărcare. 🌟</p>

ChatGPT-3.5	Gemini	Copilot
	<p> timpii de răspuns mari.</p> <ul style="list-style-type: none"><li>• Oferiți asistență tehnică promptă și eficientă în caz de probleme cu aplicația.</li><li>• Implementați o politică de securitate clară și transparentă.</li></ul> <p>Aplicând aceste sugestii, veți putea crește semnificativ productivitatea secretarilor și le veți oferi o experiență de lucru mai plăcută și mai eficientă.</p> <p>Răspuns Gemini (5/5)</p>	<p>Răspuns Copilot (4/5)</p>

<p>Q-6: Aici sunt requirement-urile unei aplicatii pentru un proiect la facultate. Fa-mi o lista pe scrut cu tot ce ar trebui sa faca aplicatia.</p>
<p>A: ChatGPT 4.0</p> <p><b>Lista cerințelor pentru aplicația "Sistem de generare a adeverințelor electronice pentru studenți"</b></p> <p><b>1. Prezentare generală</b></p> <ul style="list-style-type: none"><li>• <b>Obiective :</b><ul style="list-style-type: none"><li>• Simplificarea procesului de obținere a adeverințelor pentru studenți.</li><li>• Reducerea timpului necesar eliberării adeverințelor.</li><li>• Asigurarea corectitudinii adeverințelor generate.</li></ul></li><li>• <b>Funcționalități generale :</b><ul style="list-style-type: none"><li>• Generarea adeverinței la primirea unei cereri electronice.</li><li>• Validarea adeverinței de către secretariat.</li><li>• Informarea studentului despre starea cererii prin email.</li><li>• Alocarea unui număr de înregistrare unic adeverinței.</li><li>• Generarea de rapoarte zilnice sau lunare privind adeverințele emise.</li></ul></li></ul> <p><b>2. Categori de utilizatori ai aplicației</b></p> <p><b>2.1. Rol: Student</b></p> <ul style="list-style-type: none"><li>• <b>Acces aplicație :</b><ul style="list-style-type: none"><li>• Autentificare prin "Sign In With Google" cu adresa de email asociată facultății.</li></ul></li><li>• <b>Utilizarea aplicației :</b><ul style="list-style-type: none"><li>• Completarea unui formular Google Forms cu scopul adeverinței.</li><li>• Validarea și sanitizarea datelor introduse în formular.</li><li>• Identificarea pe baza adresei de email folosite la autentificare .</li></ul></li></ul> <p><b>2.2. Rol: Secretariat</b></p>

- **Acces aplicație :**

- Autentificare prin "Sign In With Google" cu adresa de email asociată facultății.
- Posibilitatea de deconectare din aplicație.

- **Utilizarea aplicației :**

- Acces la lista de cereri în așteptare.
- Introducerea manuală a numărului de înregistrare pentru prima cerere aprobată zilnic.
- Configurarea opțională a sintaxei numărului de înregistrare.
- Respingerea cererilor neaprobrate și arhivarea automată a acestora cu motivarea respingerii.
- Editarea scopului adevărții.
- Listarea cererilor aprobate.
- Vizualizarea cererilor aprobate și neaprobrate.
- Emiterea de rapoarte în format XLSX.
- Trimiterea de email-uri de informare către studenți și secretare.
- Editarea și gestionarea datelor studenților .

### 2.3. Rol: Administrator

- **Acces aplicație :**

- Autentificare cu nume de utilizator și parolă.
- Posibilitatea de deconectare din aplicație.

- **Configurarea aplicației :**

- Actualizarea credențialelor de acces.
- Editarea informațiilor specifice facultății.
- Adăugarea și editarea utilizatorilor cu rol de secretar.

- **Utilizarea aplicației :**

- Încărcarea listei studenților înmatriculați prin fișiere XLSX sau CSV.
- Validarea și sanitizarea datelor din fișierele încărcate.
- Resetarea aplicației pentru un nou an universitar și generarea unui raport XLSX cu informațiile curente .

### 3. Cerințe de securitate

- Implementarea măsurilor de securitate adecvate pentru protejarea datelor studenților și asigurarea integrității adevărților .

### 4. Lansare în execuție și mentenanță

- Documentarea procesului de instalare, configurare și pornire a aplicației sub forma unui fișier PDF.
- Furnizarea codului sursă complet al aplicației sub forma unei arhive ZIP.
- Furnizarea unei copii funcționale a aplicației sub forma unei mașini virtuale Ubuntu Server sau a unui container Docker .

### 5. Model adevărții

- Generarea adevărților conform unui model prestabilit, listate câte 4 pe pagină A4.
- Personalizarea adevărților în funcție de programul de studii și alte detalii specifice .

<https://chatgpt.com/share/0896ad57-ef77-4b2a-bfde-a59eaf3a64ab>

Q-7: Fa-mi o lista cu checkboxes, ca sa o pot pune in notion

## Lista cerințelor pentru aplicația "Sistem de generare a adeverințelor electronice pentru studenți"

### 1. Prezentare generală

- ☒ Simplificarea procesului de obținere a adeverințelor pentru studenți.
- ☒ Reducerea timpului necesar eliberării adeverințelor.
- ☒ Asigurarea corectitudinii adeverințelor generate.
- ☒ Generarea adeverinței la primirea unei cereri electronice.
- ☒ Validarea adeverinței de către secretariat.
- ☒ Informarea studentului despre starea cererii prin email.
- ☒ Alocarea unui număr de înregistrare unic adeverinței.
- ☒ Generarea de rapoarte zilnice sau lunare privind adeverințele emise.

### 2. Categori de utilizatori ai aplicației

#### 2.1. Rol: Student

- ☒ Autentificare prin "Sign In With Google" cu adresa de email asociată facultății.
- ☒ Completarea unui formular Google Forms cu scopul adeverinței.
- ☒ Validarea și sanitizarea datelor introduse în formular.
- ☒ Identificarea pe baza adresei de email folosite la autentificare.

#### 2.2. Rol: Secretariat

- ☒ Autentificare prin "Sign In With Google" cu adresa de email asociată facultății.
- ☒ Posibilitatea de deconectare din aplicație.
- ☒ Acces la lista de cereri în așteptare.
- ☒ Introducerea manuală a numărului de înregistrare pentru prima cerere aprobată zilnic.
- ☒ Configurarea opțională a sintaxei numărului de înregistrare.
- ☒ Respingerea cererilor neaprobate și arhivarea automată a acestora cu motivarea respingerii.
- ☒ Editarea scopului adeverinței.
- ☒ Listarea cererilor aprobate.
- ☒ Vizualizarea cererilor aprobate și neaprobate.
- ☒ Emiterea de rapoarte în format XLSX.
- ☒ Trimiterea de email-uri de informare către studenți și secretare.
- ☒ Editarea și gestionarea datelor studenților.

#### 2.3. Rol: Administrator

- ☒ Autentificare cu nume de utilizator și parolă.
- ☒ Posibilitatea de deconectare din aplicație.
- ☒ Actualizarea credențialelor de acces.
- ☒ Editarea informațiilor specifice facultății.
- ☒ Adăugarea și editarea utilizatorilor cu rol de secretar.
- ☒ Încărcarea listei studenților înmatriculați prin fișiere XLSX sau CSV.
- ☒ Validarea și sanitizarea datelor din fișierele încărcate.
- ☒ Resetarea aplicației pentru un nou an universitar și generarea unui raport XLSX cu informațiile curente.

### 3. Cerințe de securitate

- ☒ Implementarea măsurilor de securitate adecvate pentru protejarea datelor studenților și asigurarea integrității adeverințelor.

#### 4. Lansare în execuție și mentenanță

- ☒ Documentarea procesului de instalare, configurare și pornire a aplicației sub forma unui fișier PDF.
- ☒ Furnizarea codului sursă complet al aplicației sub forma unei arhive ZIP.
- ☒ Furnizarea unei copii funcționale a aplicației sub forma unei mașini virtuale Ubuntu Server sau a unui container Docker.

#### 5. Model adeverință

- ☒ Generarea adeverințelor conform unui model prestabilit, listate câte 4 pe pagină A4.
- ☒ Personalizarea adeverințelor în funcție de programul de studii și alte detalii specifice.

Q-8: Q: I need you to identify and create the users diagram using UML code.

A: ChatGPT 3.5

@startuml

left to right direction

actor Student as S  
actor Secretariat as Sec  
actor Administrator as Admin

```
package "Application" {  
    usecase "Access Application" as UC1  
    usecase "Use Application" as UC2  
    usecase "Manage Requests" as UC3  
    usecase "Configure Application" as UC4  
}
```

S --> UC1 : "Sign In With Google"  
Sec --> UC1 : "Sign In With Google"  
Admin --> UC4 : Username & Password

S --> UC2 : Fill Google Form  
Sec --> UC2 : View Pending Requests  
Sec --> UC2 : Approve/Reject Request  
Sec --> UC2 : Edit Certificate Purpose  
Sec --> UC2 : Generate Certificate Report  
Sec --> UC2 : View Approved/Rejected Requests  
Sec --> UC2 : Edit Student Data  
Sec --> UC2 : Delete Student  
Sec --> UC2 : Add New Student  
Admin --> UC4 : Update Credentials  
Admin --> UC4 : Edit Faculty Information  
Admin --> UC4 : Upload Student List  
Admin --> UC4 : Reset Application

UC2 --> UC3 : Approve/Reject Request  
UC2 --> UC3 : Print Certificates  
UC2 --> UC3 : View Unapproved Requests  
UC2 --> UC3 : View Approved Requests  
UC2 --> UC3 : Generate Daily Report  
UC2 --> UC3 : Send Email Notification

UC3 --> UC2 : Notify Student  
UC4 --> UC3 : Export Data

@enduml

[https://www.plantuml.com/plantuml/png/XPHFZzCm4CNI\\_XIZ7hYj4Bjxue2swfyL4gogHeNpo9vc3OuTF1DAXVZksAtJaco0tEBtoxoPULOUM31ArpYb7Eq5942qzRE0iP6qsE2LGYqXGYMT8I-035L\\_H3gIOBHufYPTXBbfhBSi4VCRylb0gHRrLwm9PIEsTLPZhZ-3dmhIqp5fvAnfJSmmG91XjtWpv7PC4z3j4Fg8FXjkwLj7B5oWko6q27vlwovErRjNlvIguERcNUu0tiAiihM7znu-Mtc6nnlgHpELPlylSrf2eUuJbGQ87Xk2Lx11vaE8vcfrcu2rTQvy3UiGczwbo9yi7M13tbXVNoOSCMcW6BxJworz8IqzDO9MnWeiA8hTv-49Dbri0zC8UoHFCSa3T4jjY3BLMR4snPICkCU1VN-zbYWuqfVaIAWdnWCQ0qzqwDMNIsxDgUL8MRReU8GK\\_pNngpianXRPfcpxTW8bQBg2vjFh1yak\\_BJ79d\\_T8gTHe0Ux-diWGsaJh1v7m02WRtdaiE-xBJ6BpVqFNN9TetR4aEc8gyWPMJGBWAUIcod19IOPnbewNBF96hYLMFrBT4h1wIFNoN-St](https://www.plantuml.com/plantuml/png/XPHFZzCm4CNI_XIZ7hYj4Bjxue2swfyL4gogHeNpo9vc3OuTF1DAXVZksAtJaco0tEBtoxoPULOUM31ArpYb7Eq5942qzRE0iP6qsE2LGYqXGYMT8I-035L_H3gIOBHufYPTXBbfhBSi4VCRylb0gHRrLwm9PIEsTLPZhZ-3dmhIqp5fvAnfJSmmG91XjtWpv7PC4z3j4Fg8FXjkwLj7B5oWko6q27vlwovErRjNlvIguERcNUu0tiAiihM7znu-Mtc6nnlgHpELPlylSrf2eUuJbGQ87Xk2Lx11vaE8vcfrcu2rTQvy3UiGczwbo9yi7M13tbXVNoOSCMcW6BxJworz8IqzDO9MnWeiA8hTv-49Dbri0zC8UoHFCSa3T4jjY3BLMR4snPICkCU1VN-zbYWuqfVaIAWdnWCQ0qzqwDMNIsxDgUL8MRReU8GK_pNngpianXRPfcpxTW8bQBg2vjFh1yak_BJ79d_T8gTHe0Ux-diWGsaJh1v7m02WRtdaiE-xBJ6BpVqFNN9TetR4aEc8gyWPMJGBWAUIcod19IOPnbewNBF96hYLMFrBT4h1wIFNoN-St)

Q-9: Now the entity diagram.

A: ChatGPT 3.5

@startuml

```
entity "Student" as student {  
    + email  
    + fullName  
    + gender  
}
```

```
entity "CertificateRequest" as request {  
    + requestId  
    --  
    + purpose  
    + status  
    + approvalDate  
}
```

```
entity "Certificate" as certificate {  
    + certificateId  
    --  
    + registrationNumber  
    + issueDate  
}
```

```
entity "Secretariat" as secretariat {  
    + email  
    + fullName  
}
```

```
entity "Administrator" as admin {  
    + username  
    --  
    + password  
}
```

```
entity "Faculty" as faculty {  
    + fullName  
    + shortName  
    + currentAcademicYear  
    + deanName  
}
```

```
+ chiefSecretaryName
}
```

```
student -- request
request -- certificate
secretariat -- request
admin -- faculty
```

@enduml

[https://www.plantuml.com/plantuml/png/TL6xJWCn4Epz5LPQTF-G28H4au9Kb8kzbrYoVSS-G17Ytmdh76T5IkMPaNV6djs08cMx1nMINBscuDelE2D1\\_RSiFzstZzuB92oeL5ptaD18qUi4Sdzh1QFI9B6543m5JyKkRbHmyrjOI\\_nmeR1f5bfdXWDiu0e6uHvfkaJyXC8tean\\_x3obj4hVGxXCR4GI9hgNiixaic9MV4sv821K80IN2jP-VrQklbTBACSj8dC0VwKwwmoKeM2NGV0\\_3LHx3sU8MYMiqsF3TlypIBuD97yiw14M6KN8699uGs1J8y8TRrqIZWkNpmIRr\\_MFWpBbbmxJUfATLqR\\_MtxuaLONklS5clKajql](https://www.plantuml.com/plantuml/png/TL6xJWCn4Epz5LPQTF-G28H4au9Kb8kzbrYoVSS-G17Ytmdh76T5IkMPaNV6djs08cMx1nMINBscuDelE2D1_RSiFzstZzuB92oeL5ptaD18qUi4Sdzh1QFI9B6543m5JyKkRbHmyrjOI_nmeR1f5bfdXWDiu0e6uHvfkaJyXC8tean_x3obj4hVGxXCR4GI9hgNiixaic9MV4sv821K80IN2jP-VrQklbTBACSj8dC0VwKwwmoKeM2NGV0_3LHx3sU8MYMiqsF3TlypIBuD97yiw14M6KN8699uGs1J8y8TRrqIZWkNpmIRr_MFWpBbbmxJUfATLqR_MtxuaLONklS5clKajql)

Q-10: And now the classes diagram.

A: ChatGPT 3.5

@startuml

```
class Student {
- email: string
- fullName: string
- gender: string
+ makeCertificateRequest(): void
}
```

```
class CertificateRequest {
- requestId: int
- purpose: string
- status: string
- approvalDate: date
+ approveRequest(): void
+ rejectRequest(): void
}
```

```
class Certificate {
- certificateId: int
- registrationNumber: string
- issueDate: date
}
```

```
class Secretariat {
- email: string
- fullName: string
+ viewPendingRequests(): void
+ approveCertificateRequest(): void
+ rejectCertificateRequest(): void
+ generateCertificateReport(): void
}
```

```
class Administrator {
- username: string
- password: string
+ updateCredentials(): void
}
```



```
+ editFacultyInformation(): void
+ uploadStudentList(): void
+ resetApplication(): void
}
```

```
class Faculty {
- fullName: string
- shortName: string
- currentAcademicYear: string
- deanName: string
- chiefSecretaryName: string
}
```

Student "1" --> "0..\*" CertificateRequest  
CertificateRequest "1" --> "0..1" Certificate  
Secretariat "1" --> "0..\*" CertificateRequest  
Administrator "1" --> "1" Faculty

@enduml

[https://www.plantuml.com/plantuml/png/bLDDRrCn4BpxLrWv0LKgUiq1DG8XLK8LeYUEYpr95\\_nXrkjK5UA\\_awHkYnBFAZqzUJoxEpFUTrcLH5iCnhX0jTeRRHv9xIzZxT8Y4eULhIgSjaTaqqaufeW1t29vo0bqPYFzm7k8yeOTARxWPqFLLwzNTfVPcz-FywQSFbeUJbT-PJdf4IfDIgxZwAgahGuGbI9vH-43wPxhIN5GrE69aiELu3kS\\_e\\_6BixzHKu52hPSLKWvf-iMlq46OLnhmucgfm4tS08bONfHy6TsnxZxZEJtvww-3iQwvqd8S\\_QV8MsH8AHZhV9b9g-rZvmUWiZI3RKAIV\\_kJTcpxxBumLCh\\_Z16u96KAOoEu5a\\_acj1xw\\_I9aiyXZrGMWcPV5\\_aJpmnMw7hKWAXfz9HVC\\_VXSykVBtDeXFKDH4aNJloYEo-WiRtzW0qBRfbR1uNu7wutYlgBkpYOc6NotTsyVRy\\_CrYvgyn4sWikXYApEd6FTzxVD3EFtvxLiPS8laMW\\_a3](https://www.plantuml.com/plantuml/png/bLDDRrCn4BpxLrWv0LKgUiq1DG8XLK8LeYUEYpr95_nXrkjK5UA_awHkYnBFAZqzUJoxEpFUTrcLH5iCnhX0jTeRRHv9xIzZxT8Y4eULhIgSjaTaqqaufeW1t29vo0bqPYFzm7k8yeOTARxWPqFLLwzNTfVPcz-FywQSFbeUJbT-PJdf4IfDIgxZwAgahGuGbI9vH-43wPxhIN5GrE69aiELu3kS_e_6BixzHKu52hPSLKWvf-iMlq46OLnhmucgfm4tS08bONfHy6TsnxZxZEJtvww-3iQwvqd8S_QV8MsH8AHZhV9b9g-rZvmUWiZI3RKAIV_kJTcpxxBumLCh_Z16u96KAOoEu5a_acj1xw_I9aiyXZrGMWcPV5_aJpmnMw7hKWAXfz9HVC_VXSykVBtDeXFKDH4aNJloYEo-WiRtzW0qBRfbR1uNu7wutYlgBkpYOc6NotTsyVRy_CrYvgyn4sWikXYApEd6FTzxVD3EFtvxLiPS8laMW_a3)

Q-11: Acum diagrama pe componente.

A: ChatGPT 3.5

@startuml

```
package "Next.js Frontend" {
[Student Dashboard]
}
```

```
package "Nest.js Backend" {
[Authentication Service]
[Certificate Service]
[Request Management Service]
[Database]
}
```

```
package "External Services" {
[Google Sign-In]
[Google Forms]
[Printer]
[Email Service]
}
```

[Student Dashboard] --> [Authentication Service] : Sign In With Google  
[Student Dashboard] --> [Certificate Service] : Make Certificate Request  
[Request Management Service] --> [Database] : Store Requests  
[Certificate Service] --> [Database] : Store Certificates

[Authentication Service] --> [Google Sign-In] : Sign In  
[Request Management Service] --> [Google Forms] : Fill Form  
[Request Management Service] --> [Printer] : Print Certificates  
[Request Management Service] --> [Email Service] : Send Email Notifications

@enduml

[https://www.plantuml.com/plantuml/png/XPF1Ji9048Rl-nGJxlW07WmgO3X0Z1my60v3ExOhxQxkJWs9yTtT3bWxYQMtPORl\\_sV-IMUHCN1JLyQyOtR0Wc2oeIDVIKLO1k-ON3w1BmFmikKc9ySmnrZkFOPyPxuL5OMwIuK\\_wBRXCasGxRUMPR2fyrrerxRkAR1zRJkawa\\_qqIGjMADBmdMYLNkEZ7kCfDqNHwRWiFhzQppxFtXVL4dV5cwwShjUQUb37QNm6AnBjBmNDTfEH2p-sHkcqvl1rU1Qt63buDbo2IUxOHatGwUnnWD1ltUEnLpAHnIxX6GKzg5Zer5kbodb7itWkWBhc7iPZ8-hxv78fQqg-JNETgTBc3pLpEEyFhUCJIw7KtNZHKdMZSRCKgVzJdu0](https://www.plantuml.com/plantuml/png/XPF1Ji9048Rl-nGJxlW07WmgO3X0Z1my60v3ExOhxQxkJWs9yTtT3bWxYQMtPORl_sV-IMUHCN1JLyQyOtR0Wc2oeIDVIKLO1k-ON3w1BmFmikKc9ySmnrZkFOPyPxuL5OMwIuK_wBRXCasGxRUMPR2fyrrerxRkAR1zRJkawa_qqIGjMADBmdMYLNkEZ7kCfDqNHwRWiFhzQppxFtXVL4dV5cwwShjUQUb37QNm6AnBjBmNDTfEH2p-sHkcqvl1rU1Qt63buDbo2IUxOHatGwUnnWD1ltUEnLpAHnIxX6GKzg5Zer5kbodb7itWkWBhc7iPZ8-hxv78fQqg-JNETgTBc3pLpEEyFhUCJIw7KtNZHKdMZSRCKgVzJdu0)

Q-12: An you generate questions for gathening user stories?

A: ChatGPT 3.5

Certainly! User stories are typically derived from the perspectives of different users or stakeholders involved in the system. Here are some questions that could help gather user stories for the described system:

1. **\*\*For Students\*\***:

- What tasks do students need to accomplish within the application?
- What information do students need to provide when requesting a certificate?
- How do students want to be notified about the status of their certificate requests?
- Are there any specific preferences or requirements students have when using the application?

2. **\*\*For Secretariat\*\***:

- What steps does the secretariat take when reviewing and processing certificate requests?
- How does the secretariat want to manage and organize pending and approved requests?
  - Are there any specific criteria or guidelines the secretariat follows when approving or rejecting requests?
- What information does the secretariat need to include when generating certificate reports?

3. **\*\*For Administrator\*\***:

- What configuration options does the administrator need to manage for the application?
- How does the administrator want to handle user authentication and access control?
- Are there any specific administrative tasks or reports the administrator needs to generate or oversee?
- What actions does the administrator take to ensure the application runs smoothly and securely?

4. **\*\*General\*\***:

- Are there any security or privacy concerns that need to be addressed within the system?
- How does the system handle errors or exceptions during user interactions?
  - What external integrations or services does the system rely on, and how do they impact user workflows?
- Are there any usability or accessibility considerations that need to be taken into account for different user roles?

These questions can help uncover various user stories by eliciting specific requirements, preferences, and

expectations from different stakeholders involved in the system.

<https://chatgpt.com/share/fcbfd7f7-ac96-4fcc-93f3-2a90202a5fcd>

Q-13: Can you generate a couple of personas for secretary?

A: ChatGPT 3.5

Certainly! Here are a couple of personas for a secretary role in the described system:

1. **Maria - Experienced Secretary**:

- **Background**: Maria has been working as a secretary at the university for over 10 years. She is highly experienced in managing administrative tasks and is familiar with the processes involved in handling student requests.
- **Goals**: Maria aims to efficiently review and process certificate requests, ensuring accuracy and compliance with university regulations. She also seeks to maintain organized records and generate reports promptly.
- **Challenges**: Maria faces occasional challenges in managing a high volume of certificate requests, especially during peak times such as the end of the semester. She also needs to ensure that all requests are processed accurately and in a timely manner.
- **Preferences**: Maria prefers intuitive and user-friendly interfaces that streamline her workflow. She values clear communication and collaboration with colleagues to address any issues or discrepancies in requests.

2. **Alex - New Secretary**:

- **Background**: Alex recently joined the university as a secretary and is still learning the ropes of the administrative processes. While Alex is eager to learn and adapt, they may require additional guidance and training to become proficient in their role.
- **Goals**: Alex's primary goal is to familiarize themselves with the certificate request process and gain confidence in managing tasks independently. They aim to contribute effectively to the team and provide quality service to students.
- **Challenges**: Alex faces challenges in navigating the complex systems and procedures involved in processing certificate requests. They may encounter difficulties in prioritizing tasks and managing time effectively, especially during busy periods.
- **Preferences**: Alex appreciates clear instructions and documentation to guide them through tasks. They value supportive colleagues who are willing to provide guidance and mentorship as they learn and grow in their role.

These personas capture different levels of experience and perspectives within the secretary role, highlighting their goals, challenges, and preferences in interacting with the system.