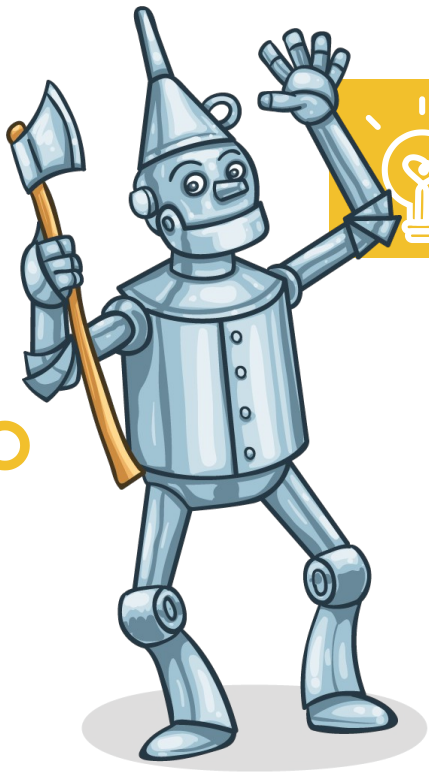


# UTILIZAREA I.A. ÎN INGINERIA PROGRAMELOR SOFTWARE

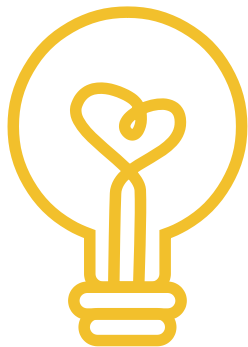
*Analiză privind oportunitățile oferite de dezvoltarea  
inteligenței artificiale  
în ingineria dezvoltării software în general și a aplicațiilor  
web în special*

## Om de tinichea din Vrăjitorul din Oz ✕



Fascinația pentru inteligența artificială este prezentă  
în gândurile și imaginația oamenilor de mulți ani.

## Uriasul din Talos



01

# Inteligența artificială & știința calculatoarelor



Lansarea ideii de dezvoltare a unei inteligențe artificiale  
care să nu poată fi deosebită de cea naturală.



01

## 1941 – Turing Test

“Computing Machinery and Intelligence”

Acest articol urma să devină piatra de temelie pentru dezvoltarea inteligenței artificiale.

02

## 1957 - Perceptron

Prima rețea neuronală – creată de către Frank Rosenblatt la **Cornell Aeronautical Laboratory**

03

## 1967 - ELIZA

Un algoritm bazat pe recunoașterea șabloanelor și înlocuire. Deși nu era decât un chat-bot simplu și nu era dotat cu inteligență reală crea o iluzie convingătoare. Precursorul chatboturilor actuali dotați cu I.A.

04

## 2018 - GPT1

Inițial s-a ținut departe de public accesul la LLM și I.A. deoarece s-au considerat prea periculoși. În 30.11.2022 s-a lansat ChatPT-3 în mod public.



# Schimbarea atenției

**Cum programăm un PC? →**

**Cum programăm un PC să se programeze pe sine?**

**De ce?**

**Sintetizare**

**Analiză**

**Acces la multiple tehnologii**

**Concentrare neîntreruptă**



# Ingineria software

Ingineria software reprezintă aplicarea metodelor ingineresti asupra stagiilor de planificare, dezvoltare și mentenanță a soluțiilor software. Acest concept a apărut ca și consecință a observării unei deteriorări continue și agravate a programelor software.



Scopul fundamental al ingineriei software este de a furniza produse software (livrabile) de calitate și la timp. În atingerea acestui scop ingineria software conturează câteva obiective pe care produsul final trebuie să le atingă:

**fiabilitate**, adică abilitatea de a își îndeplini funcțiile pentru o perioadă indefinită de timp;

**reutilizarea**, posibilitatea de a fi refolosit în tot sau în parte;

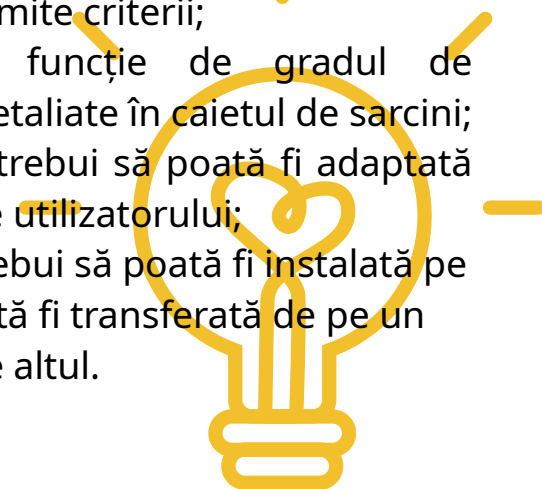
**servisabilitate**, posibilitatea de a putea fi îmbunătățit sau adaptat cu ușurință la cerințele actuale;

**testabilitate**, posibilitatea de a testa și a evalua performanțele software-ului în funcție de anumite criterii;

**corectitudine**, determinabilă în funcție de gradul de îndeplinire a caracteristicilor detaliate în caietul de sarcini;

**adaptabilitate**, soluția software ar trebui să poată fi adaptată cu un minim de efort la nevoile utilizatorului;

**portabilitate**, soluția software ar trebui să poată fi instalată pe multiple sisteme sau să poată fi transferată de pe un sistem pe altul.





# - În ce faze ale dezvoltării ne poate ajuta I.A.G.?



În faza de idee și analiză a ideii ne poate oferi capacități avansate de analiză și sinteză privind evaluarea oportunității și a riscurilor.



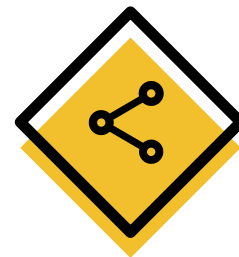
În faza de planificare ne poate oferi ajutor în stabilirea unor pași concreți în vederea creării unui plan de acțiune coerent și riguros.



În faza de proiectare ne poate ajuta cu multiple tehnici și elemente de proiectare: identificarea de entități, procese, clase, construirea de diagrame, identificarea de sarcini, etc.



În faza de dezvoltare efectivă a codului capacitatea de a utiliza în mod eficient inteligența artificială devine repede o condiție de supraviețuire în lumea IT. Este de un ajutor uriaș pentru programatori.



În faza de întreținere și testare ajută la identificarea de metode fiabile de mentenanță a software-ului și ajută la depanarea cu rapiditate a eventualelor probleme apărute în ciclul de viață a aplicației.



-v03.03- Caiet de sarcini TWAACS-SIC-... PDF

Aici sunt requirement-urile unei aplicații pentru un proiect la facultate. Fa-mi o lista pe scrut cu tot ce ar trebui sa faca aplicatia.

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

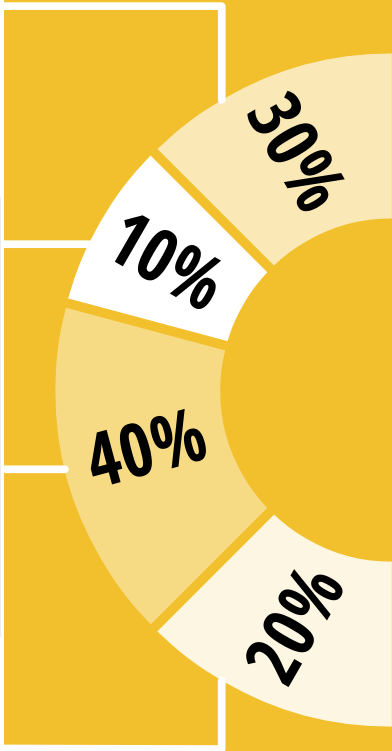
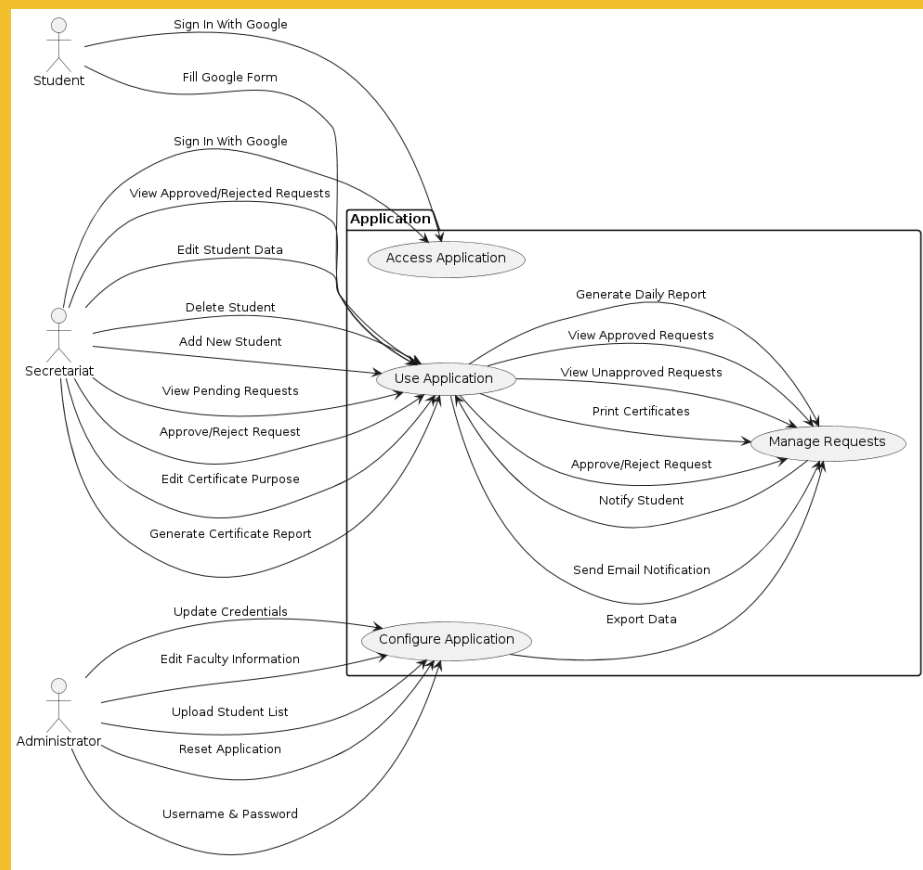
**1. Prezentare generală**

- Obiective:**
  - Simplificarea procesului de obținere a adeverințelor pentru studenți.
  - Reducerea timpului necesar eliberării adeverințelor.
  - Asigurarea corectitudinii adeverințelor generate.
- Funcționalități generale:**
  - 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 [\[1\]](#).

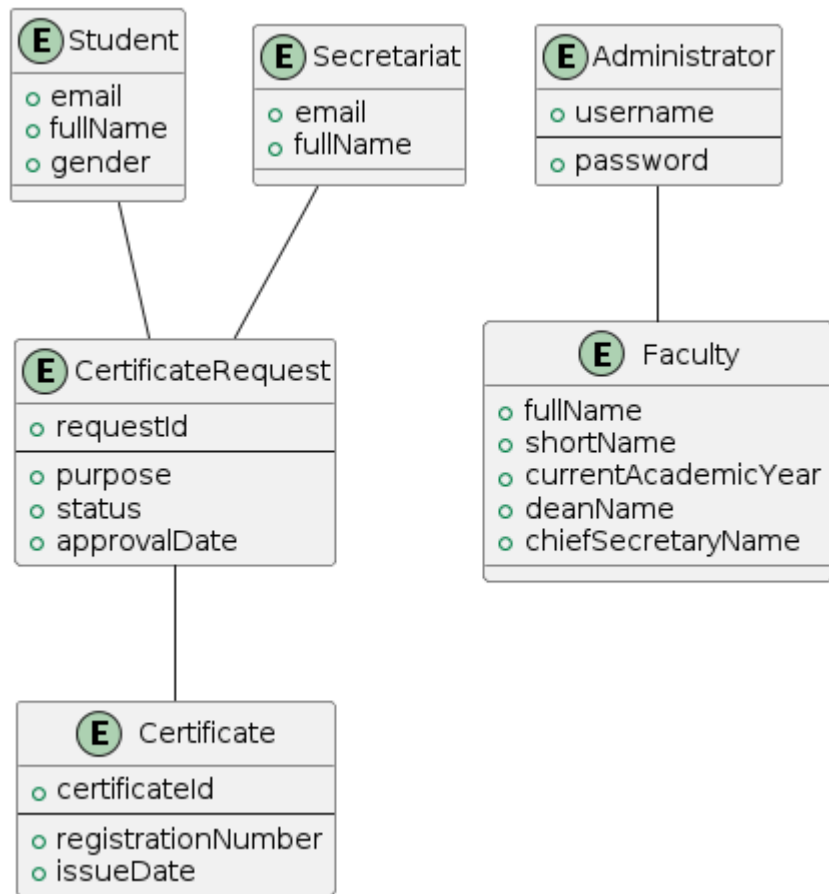
**2. Categoriile de utilizatori ai aplicației**

**2.1. Rol: Student**

- Acces aplicație:**
  - Autentificare prin "Sign In With Google" cu adresa de email asociată facultății.







I seem to have some problem with the callback



If you are encountering issues with the callback, there are several potential areas to check. Let's go through a comprehensive setup and troubleshooting guide to ensure everything is configured correctly.

### Step-by-Step Setup and Troubleshooting Guide

#### 1. Ensure Correct Callback URLs

Verify that the callback URL configured in your Google Cloud Console matches the one in your Nest.js strategy setup.

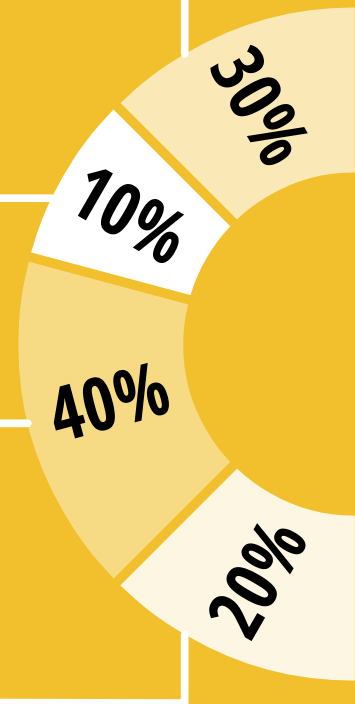
- **Google Cloud Console:** The callback URL should be `'http://localhost:3000/auth/google/callback'`.
- **Nest.js Strategy:** Ensure the callback URL is set correctly in your `'GoogleStrategy'`:

typescript

Copy code

```
import { Injectable } from '@nestjs/common';
import { PassportStrategy } from '@nestjs/passport';
import { Strategy, VerifyCallback } from 'passport-google-oauth20';

@Injectable()
export class GoogleStrategy extends PassportStrategy(Strategy, 'google') {
  constructor() {
    super({
      clientId: process.env.GOOGLE_CLIENT_ID,
      clientSecret: process.env.GOOGLE_CLIENT_SECRET,
      callbackURL: 'http://localhost:3000/auth/google/callback',
      scope: ['email', 'profile'],
    });
  }
}
```

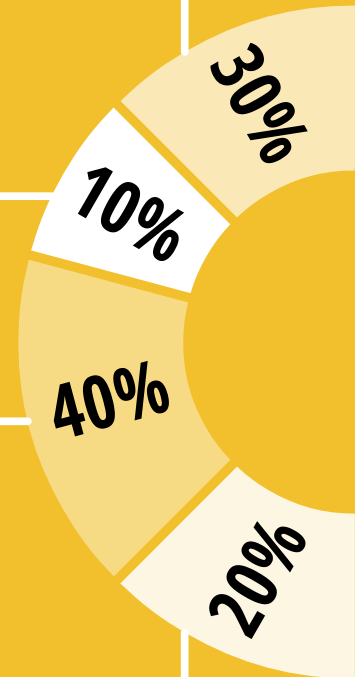




## I.A. nu pleacă nicăieri

Paradigma de dezvoltare a soluțiilor software se va schimba mai devreme decât mai târziu și trebuie să ne pregătim pentru o lume în care inteligența artificială va face munca grea – așa-numita „eavy lifting” – iar valoarea reală a aplicațiilor va proveni din valoarea lor intrinsecă pentru utilizatori – adică în ce mod rezolvă o problemă a acestora și mai puțin de calitatea dezvoltării care va fi ținută la un standard înalt de către unelte de dezvoltare pe bază de inteligență artificială.

Pentru a avea succes va trebui să învățăm să o exploatăm eficient.





Multumim!