

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI  
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

**Courseeker**

propusă de

***Dănuț Talabă***

**Sesiunea:** *feb, 2020*

Coordonator științific  
**Prof. Lect. Dr. Pistol Ionuț**

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI  
FACULTATEA DE INFORMATICĂ

# Courseeker

***Dănuț Talabă***

**Sesiunea:** *feb, 2020*

Coordonator științific  
**Prof. Lect. Dr. Pistol Ionuț**



# Cuprins

<b>1. INTRODUCERE.....</b>	<b>9</b>
<b>1. ABORDARE CONCEPTUALĂ .....</b>	<b>11</b>
1.1. ÎNVĂȚAREA.....	12
1.2. PROCESAREA LIMBAJULUI NATURAL (ENGL. <i>NATURAL LANGUAGE PROCESSING OR NLP</i> ).....	14
<b>2. SIMILARITATE .....</b>	<b>17</b>
2.1. CÂT DE IMPORTANT ESTE UN CUVÂNT? .....	17
2.2. FRECVENȚA TERMENULUI - FRECVENȚA INVERSATĂ A DOCUMENTULUI(TF- IDF) .....	18
2.3. ANALIZA TEXTULUI EXTRAS.....	21
2.4. SIMILARITATE COSINUS .....	23
<b>3. ARHITECTURA APLICAȚIEI .....</b>	<b>25</b>
3.1. BAZA DE DATE .....	26
3.2. ADMIN .....	27
3.1.1 Strapi.....	29
3.1.2. <i>La ce se referă un curs ?</i> .....	29
3.3. CLIENT .....	30
3.3.1. SCURTĂ PREZENTARE A APLICAȚIEI .....	30
3.3.2. PAGINA PRINCIPALĂ .....	30
3.3.3. PAGINA „COURSES” .....	33
3.3.4. PAGINA „CATEGORIES” .....	34
3.3.5. PAGINA CURSULUI VIZUALIZAT .....	34
3.3.6. PAGINA SIMILARITĂȚILOR .....	37
3.4. MOTIVUL UTILIZĂRII LIBRARIILOR .....	38
3.5. LIBRĂRII ȘI FRAMEWORK-URI.....	38
<b>4. CONCLUZII .....</b>	<b>39</b>
<b>5. BIBLIOGRAFIE .....</b>	<b>40</b>

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele \_\_\_\_\_

Data \_\_\_\_\_ Semnătura \_\_\_\_\_

### **DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemnatul(a)

.....

domiciliul în .....

născut(ă) la data de ....., identificat prin CNP  
....., absolvent(a) al(a) Universității „Alexandru Ioan Cuza”  
din Iași, Facultatea de ..... specializarea  
....., promoția ....., 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:

\_\_\_\_\_  
\_\_\_\_\_

\_\_\_\_\_elaborată sub îndrumarea dl. / d-na  
\_\_\_\_\_, pe care urmează să o susțină 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 său 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 diploma 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 azi, .....

Semnătură student .....

## DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Titlul complet al lucrării*”, 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.

Iași,

Absolvent *Dănut Talabă*

---

(semnătura în original)





## 1. Introducere

Era digitală ... Cum am putea defini această expresie atât de utilizată în zilele noastre?

Astăzi cam tot ceea ce ne înconjoară și mare parte din ceea ce utilizăm zi de zi, se află în concordanță cu fenomenul de **digitalizare**. Tehnologia a evoluat în ultimul deceniu mai mult ca niciodată, și dacă e să facem o retrospectivă, practic toată viața noastră, împreună cu acțiunile zilnice depind de ea. Prin urmare, chiar și fără o analiză complexă, putem spune, fără doar și poate că trăim într-o **eră digitală**, în care vrem, nu vrem depindem și suntem supuși acestui fenomen deja controversat.

Odată cu digitalizarea lumii în care trăim, un mare avantaj de care dispunem este accesul la informație, fie că vorbim aici de televiziune, radio sau internet. Televiziunea și radioul oferă informație consumatorului, însă aceasta este o informație relativ obiectivă, prelucrată pentru a fi înțeleasă și analizată așa cum vrea redactorul.

Bineînțeles, și din media putem alege informația pe care ne-o dorim, dar mult mai ușor o facem în momentul în care utilizăm internetul, având o sursă vastă de informație prelucrată și neprelucrată totodată, dar care oferă utilizatorului mai multe modalități de cunoaștere și informare.

Cu mulți ani înainte să dispunem de o tehnologie atât de avansată, oamenii utilizau ziare și publicații, lucrări științifice, cărți sau notițe ale prietenilor. Primul ziar a apărut în secolul XVII, aceasta conținea știri, anunțuri și articole publicitare, dând startul unei noi ere care urma să extindă accesul la informație, cultură și publicitate, eră care nu și-a încetat expansiunea nici până astăzi, cred eu.

Facilitându-se accesul la informație în mediul online se poate observa o ușurință în obținerea informației, cu precădere determinată de multiplele surse de documentare antrenate de motoare de căutare foarte puternice, cu capacitate de procesare impresionantă.

În prezent încă se discută aprins despre termenul “[future of work](#)”, care se referă la expansiunea viitoare a tehnologiei informației și care actualmente stârnește controverse prin utilizarea relativ nouă a inteligenței artificiale.

Privită subiectiv de unii și obiectiv de alții, observăm o dorință continuă a utilizatorilor și în general a factorului uman, de cunoaștere și dezvoltare tehnologică. Așa ajungem să realizăm că expansiunea tehnologică nu este un simplu moft al unor ultime generații, ci este de fapt o nevoie reală de dezvoltare, o nevoie de cunoaștere și de ușurare a învățării și selectării informației în funcție de diverși factori. Acest impuls uman al dorinței de cunoaștere ne implică în fenomenul de digitalizare și ne face complici, indiferent de aportul adus, fie el acela de dezvoltare de aplicații sau de utilizare a lor.

## 1. Abordare conceptuală

CourSeeker ("Course Seeker") este o aplicație care, pe baza activității utilizatorului vine în ajutorul său cu recomandări similare pentru fișierele citite. Prin urmare aplicația are rolul de a furniza și de a oferi sugestii cu privire la cursurile care prezintă un interes pentru utilizator folosindu-se de cursurile stocate într-o bază de date. Prin intermediul unor mecanisme de extracție a textelor ce aparțin cursului respectiv, texte extrase din fișiere de tip „.pdf” deoarece un curs poate fi format din unul sau mai multe fișiere, se face o prelucrare a datelor obținute pentru a putea furniza utilizatorului recomandări care pot prezenta interes.

În funcție de acest parcurs căutarea documentelor similare, în cazul nostru cursuri ale Facultății de Informatică și MIT (Massachusetts Institute of Technology<sup>1</sup>), se va face la nivelul aplicației având ca set de date fișiere încărcate de către administrator, deasemenea sistemul este capabil să detecteze un curs nou care a fost încărcat de către cei de la MIT și în felul acesta să îl descarce și totodată să se facă procesarea fișierelor din cadrul acelui curs. De exemplu dacă studentul deschide un curs de Inteligență artificială, aplicația va afișa în partea stângă cursurile care pot prezenta interes pentru utilizator.

Odată determinată și trasată o arie de interes, aceasta nu va rămâne una standard, utilizatorul putând să își modifice direcția de interes prin accesarea unor altor cursuri din diverse alte categorii. Pe măsură ce acesta își redirecționează căutările, aplicația va recalcula noi sugestii pe baza noilor direcții de interes ale utilizatorului, venind în ajutorul lui cu informa

---

<sup>1</sup> Massachusetts Institute of Technology - <https://ocw.mit.edu/index.htm>

## 1.1. Învățarea

Cum poate fi definit procesul de învățare?

Este o întrebare simplă la care cei mai mulți dintre noi am putea răspunde cu ușurință. Dacă vom consulta [DEX](#)-ul vom putea obține o definiție mai exactă asupra cuvântului învățare.

2

- 
- ♦ *A dobândi cunoștințe prin studiu, a ajunge prin muncă sistematică să cunoști o meserie, o artă, o limbă etc.; a [studia](#).*
  - ♦ *A-și întipări în minte ceva pentru a putea reproduce; a [memora](#).*
- 

Din definiția de mai sus, se evidențiază două cuvinte și anume: a studia și a memora. Capacitatea de învățare pe care omul și-a dezvoltat-o de-a lungul timpului, ne ajută astăzi să ne integrăm ușor în medii diferite pe glob, să ne specializăm pe un anumit domeniu sau să rezolvăm probleme complexe.

Dorința oamenilor de știință de a crea inteligență artificială a dus la crearea unor algoritmi de învățare și nu numai, care sunt folosiți astăzi pentru a crea inteligențe artificiale. Odată implementate în majoritatea zonei materiale utilizate în zilele noastre cum ar fi: gadget-uri, mașini, telefoane, mașini industriale etc., acești algoritmi au fost perfecționați constant, ajungând să facă parte din rutina noastră zilnică și ușurând activități cotidiene, revista Forbes <sup>3</sup> clasificând chiar un top al celor mai utile zece aplicații ce utilizează inteligența artificială în zilele noastre.

---

<sup>2</sup> <https://dexonline.ro/definitie/invatare>

<sup>3</sup> <https://www.forbes.com/sites/robertadams/2017/01/10/10-powerful-examples-of-artificial-intelligence-in-use-today/#750b7e57420d>

Având o idee despre avantajele utilizării inteligenței artificiale și a unui sistem de învățare automată, putem vorbi și despre dezavantajele unui sistem clasic care nu învață, pe care le putem clasifica succint după cum urmează:

- a. realizează calcule numeroase pentru rezolvarea unei probleme
- b. nu memorează soluția
- c. de fiecare dată, realizează aceeași secvență de calcule complexe.

Să facem un exercițiu de imaginație și să presupunem că facem parte dintr-un grup cu care ne vedem și interacționăm zilnic câte două ore. Să ne imaginăm că de fiecare dată ar trebui să ne prezentăm, să vorbim despre noi și să ne expunem interesele și pasiunile pentru a putea interacționa cu restul grupului. Ar fi destul de solicitant, plictisitor și ar deveni în final o pierdere de timp.

## 1.2. Procesarea limbajului natural (*engl. Natural Language Processing or NLP*<sup>4</sup>)

Procesarea limbajului natural nu a fost niciodată mai importantă ca în ultimii ani, dorința noastră ca și oameni de a avea access la informație într-un timp cât mai scurt, iar informația obținută să fie cât mai exactă, a împins limitele noastre, ale oamenilor, către perfecționarea unei serii de algoritmi sau de a crea unii noi pentru a satisface această nevoie.

Nu este foarte ușor să înveți o mașină toate limbile vorbite de către oameni. Aceste mașini trebuie să citească texte, să le descifreze pe baza informațiilor acumulate pe parcurs, să le înțeleagă și nu în ultimul rând să aibă sens, adică dacă avem o înlănțuire de cuvinte, acele cuvinte pot forma o propoziție care are un anumit sens. De exemplu „inteligența artificială este folosită tot mai mult” sau cu aceleași cuvinte putem avea aceleași sens și anume „inteligența tot mai folosită este artificială”; pentru noi oamenii poate avea mai multe înțelesuri, fapt încercat și în cazul mașinilor: de a înțelege aceste sensuri.

### 1.2.1 Cum funcționează procesarea limbajului natural

Prin aplicarea unor algoritmi care au fost creați să ne ajute în această direcție, se încearcă extragerea textului din documente și nu numai. Ca și date se pot folosi de asemenea fișiere de tip audio sau video, textul extras fiind într-o manieră simplistă și coerentă pentru mașină, adică se încearcă renunțarea unor cuvinte care apar foarte frecvent și semne de punctuație. În schimb acest proces este unul foarte meticulos deoarece mașina trebuie să decidă ce cuvinte vor fi scoase. Pentru a obține acest lucru este nevoie de timp și de o plajă foarte mare de date.

---

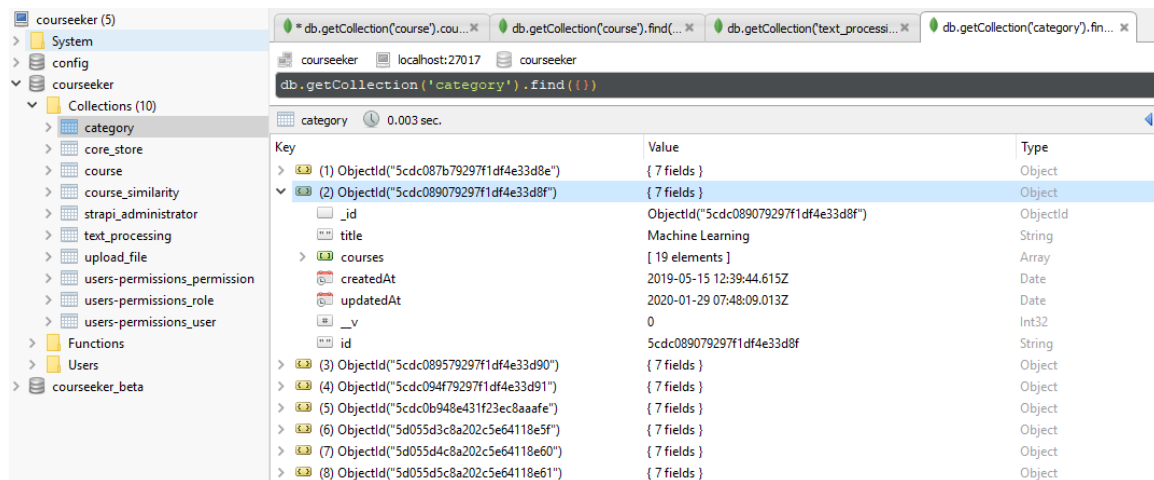
<sup>4</sup> <http://www.nltk.org/>  
<https://www.education-ecosystem.com/guides/artificial-intelligence/natural-language/history>  
<https://stanfordnlp.github.io/CoreNLP/>

Analiza datelor presupune o strategie clară și de durată. În procesarea limbajului natural sunt folosite mai multe tehnici pentru analiza, dar cele mai importante sunt sintaxa și semantica textului.

Sintaxa se referă la exemplul prezentat anterior când acea înlanțuire de cuvinte necesita un sens din punct de vedere al gramaticii. În schimb, semantica se referă la ceea vrea să exprime acea înlanțuire de cuvinte, această tehnică fiind cea mai dificilă de abordat în procesarea limbajului natural.

### 1.2.2 Utilizarea bibliotecii nltk<sup>5</sup> în aplicație

CourSeeker, reține într-o bază de date, în obiecte caracterizate prin una sau mai multe clase cunoscute, textul procesat. Analizând lingvistic textul disponibil algoritmul poate recomanda cursurile după domenii de interes precum: Tehnologii Web, Artificial Intelligence, Python etc. urmând ca fiecare curs să fie încadrat într-o clasă de interes precum cele din exemplu. Atașat în *imaginea 1*<sup>6</sup> de mai jos regăsim câteva dintre categoriile reținute în baza de date, precum și obiectele aferente informațiilor cerute.



Key	Value	Type
(1) ObjectId("5cdc087b79297f1df4e33d8e")	{ 7 fields }	Object
(2) ObjectId("5cdc089079297f1df4e33d8f")	{ 7 fields }	Object
_id	ObjectId("5cdc089079297f1df4e33d8f")	ObjectId
title	Machine Learning	String
courses	[ 19 elements ]	Array
createdAt	2019-05-15 12:39:44.615Z	Date
updatedAt	2020-01-29 07:48:09.013Z	Date
_v	0	Int32
id	5cdc089079297f1df4e33d8f	String
(3) ObjectId("5cdc089579297f1df4e33d90")	{ 7 fields }	Object
(4) ObjectId("5cdc094f79297f1df4e33d91")	{ 7 fields }	Object
(5) ObjectId("5cdc0b948e431f23ec8aaafe")	{ 7 fields }	Object
(6) ObjectId("5d055d3c8a202c5e64118e5f")	{ 7 fields }	Object
(7) ObjectId("5d055d4c8a202c5e64118e60")	{ 7 fields }	Object
(8) ObjectId("5d055d5c8a202c5e64118e61")	{ 7 fields }	Object

Imaginea 1

<sup>5</sup> <http://www.nltk.org/>

<sup>6</sup> Imaginea 1: Exemplu de categorii din tabela Categories.

Procesarea textelor va fi făcută în Python<sup>7</sup>, textul va fi extras cu ajutorul pachetului „pdfplumber<sup>8</sup>”, iar mai apoi vom folosi pachete din librăria „nltk<sup>9</sup>” precum nltk.tokenize<sup>10</sup>, nltk.corpus<sup>11</sup> dar și langdetect<sup>12</sup> pentru detectarea limbii în care a fost scris acel curs.

Pachetele menționate mai sus se vor ocupa strict de partea modelării textului pentru a-l aduce într-o formă simplistă din care se șterg semnele de punctuație și cuvintele uzuale denumite și „stop words<sup>13</sup>”.

Dacă luăm ultima frază de mai sus și o vom pune într-un document de tip pdf, iar mai apoi aplicăm algoritmi de extracție vom obține mai puține cuvinte decât fraza inițială, tocmai pentru a putea renunța la termenii care nu aduc valoare în procesul de similaritate care va fi descris ulterior.

De exemplu, am dat ca input<sup>14</sup> un document de tip pdf cu conținutul:

*„Pachetele menționate mai sus se vor ocupa strict de partea modelării textului pentru a-l aduce într-o formă simplistă din care se șterg semnele de punctuație și cuvintele uzuale denumite si „stop words<sup>15</sup>”.*

Rezultatul obținut a fost:

*„pachetele menționate ocupa strict partea modelării textului aduce într-o formă simplistă șterg semnele punctuație și cuvintele uzuale denumite stop words”*

Textele extrase pentru fiecare curs care poate conține unul sau mai multe documente ajung să fie înscrise tot în baza de date, într-o tabelă specifică care conține id-ul cursului, adresa fișierului respectiv și nu în ultimul rând textul extras.

---

<sup>7</sup> <https://www.python.org/>

<sup>8</sup> <https://pypi.org/project/pdfplumber/>

<sup>9</sup> <http://www.nltk.org/>

<sup>10</sup> <https://www.nltk.org/api/nltk.tokenize.html>

<sup>11</sup> <https://www.nltk.org/api/nltk.corpus.html>

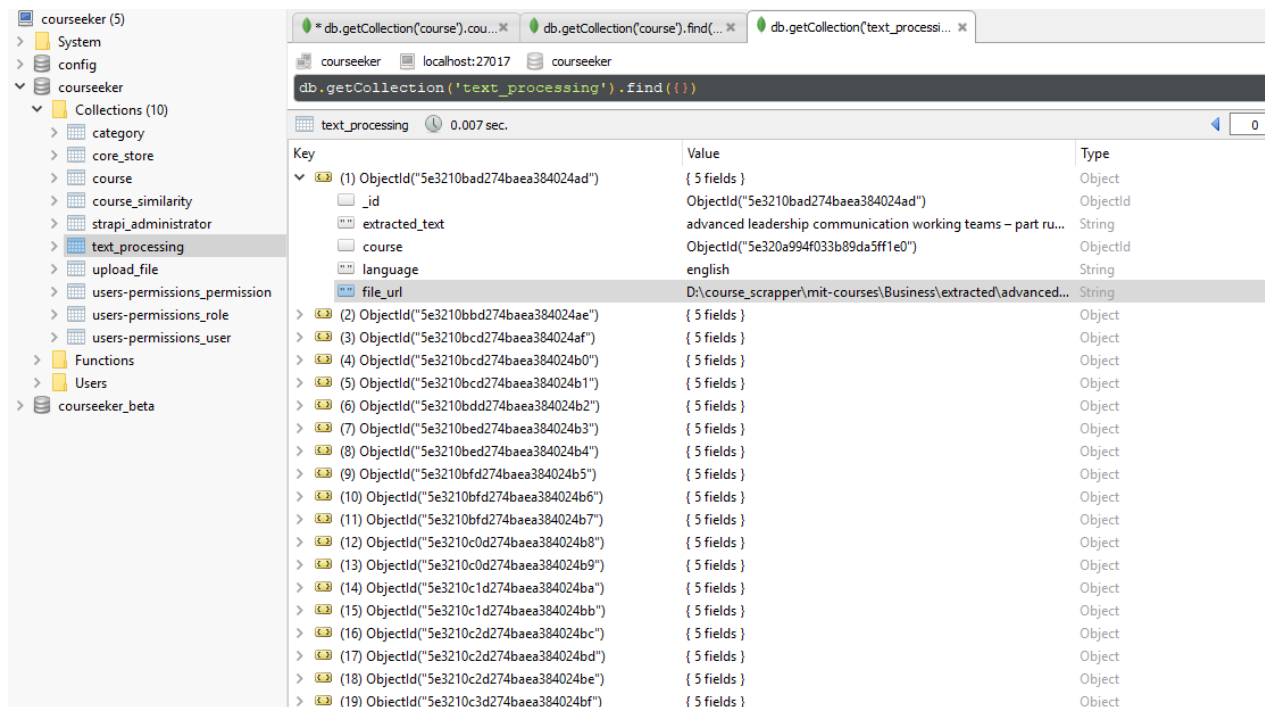
<sup>12</sup> <https://pypi.org/project/langdetect/>

<sup>13</sup> <https://pythonprogramming.net/stop-words-nltk-tutorial/>

<sup>14</sup> date de intrare pentru funcția de extragere a textului

<sup>15</sup> <https://pythonprogramming.net/stop-words-nltk-tutorial/>





Imaginea 2<sup>16</sup>

## 2. Similaritate

### 2.1. Cât de important este un cuvânt?

Un sistem de recomandări pe baza unor texte, are ca punct de plecare în special procesarea conținutului acelor texte, iar mai apoi se verifica cât de „apropiate” sunt, fiind luate două câte două. Practic similaritatea dintre două texte este să determinăm cât de apropiate din punct de vedere lexical și semantic sunt acele două texte.

Numărând de câte ori apare un cuvânt în cele două documente nu înseamnă că am calculat pe deplin similaritatea dintre acele documente; trebuie acordată atenție și contextului din care fac parte acele cuvinte. Pentru a avea similaritate semantică între două fraze, trebuie să ne focusăm asupra unei bucăți de text care face parte dintr-un grup relevant de cuvinte înrudite. O problemă des întâlnită în sistemele de recomandări care constau în calculul similarității între texte este aceea că ordinea cuvintelor contează foarte

<sup>16</sup> Imaginea 2: Exemplu pentru salvarea textului extras in baza de date .

mult. Tocmai de aceea s-a pus un accent foarte mare pe dezvoltarea algoritmilor de tip „word vectors”<sup>17</sup>.

Există numeroase tehnici de a evalua un document din punct de vedere semantic dezvoltate de-a lungul anilor, iar unele din ele ar fi:

1. Bag of Words (BoW)
2. Term Frequency - Inverse Document Frequency (TF-IDF)
4. Modele pre-antrenate de reprezentări ale cuvintelor:
  - 4.1 Word2Vec (de către Google)
  - 4.2 GloVe (de către Stanford)
  - 4.3 fastText (de către Facebook)

Pentru aplicația CourSeeker vom folosi a doua variantă pentru a construi un sistem de recomandări cu ajutorul VSM sau Vector Space Model<sup>18</sup>. Față de alte tehnici în care aflăm semantica unui cuvânt, VSM considera întregul document o colecție de cuvinte, fiecare cuvânt fiind o entitate independentă, această colecție putându-se mapa cu ușurință mai apoi într-un vector multi-dimensional. Acești vectori de cuvinte sunt rezultatul aplicării algoritmului TF-IDF în aplicația CourSeeker iar mai apoi această matrice sau vector multi-dimensional va fi folosit de către metodă de calcul a similarității dintre două cursuri, mai multe detalii la capitolul 2.2, pagina 16.

## 2.2. Frecvența termenului - Frecvența inversată a documentului(TF-IDF)

Frecvența termenului - Frecvența inversată a documentului, este un algoritm care ne oferă o statistică a unuia sau mai multe documente ce alcatuiesc un [corpus](#)<sup>19</sup>. Corpusul este argumentul pe baza caruia algoritmul va trebui să evalueze toate cuvintele din fiecare document și să stabilească importanța fiecăruia. Importanța unui

---

<sup>17</sup> vectori de cuvinte

<sup>18</sup> <https://nlp.stanford.edu/IR-book/html/htmledition/the-vector-space-model-for-scoring-1.html>

<sup>19</sup> <https://dexonline.ro/definitie/corpus>

cuvânt crește odată cu frecvența sa în document, dar va scădea luând în calcul frecvența sa în corpus.

Frecvența termenului sau TF, reprezintă de fapt numărul de apariții ale unui termen (cuvânt) într-un document. De exemplu dacă avem un document cu 1000 de cuvinte și acesta la rândul lui conține termenul „licență” de 10 ori, atunci TF pentru cuvântul licență este bazată pe următoarea formula de calcul:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (\text{formula de calcul}^{20})$$

Frecvența termenului „licență” în document este:

$$TF_{\text{licență}} = 10/1000 = 0.01 \quad ^{21}$$

În a doua etapă, Frecvența inversată a documentului sau IDF, reprezintă importanța sau numărul de apariții al termenului în întreg corpusul, prin urmare dacă se dă un corpus alcătuit din 2000 de documente și cuvântul „licență” apare în 100 de documente atunci vom obține următorul rezultat:

$$IDF_{\text{licență}} = \log(2000/100) = 1.30 \quad ^{22}$$

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|} \quad ^{23}$$

$|D|$  reprezintă numărul total de documente din cadrul corpusului. Numitorul fracției,  $|\{j : t_i \in d_j\}|$ , reprezintă numărul de documente în care apare termenul. Dacă termenul

---

<sup>20</sup> [http://disi.unitn.it/~bernardi/Courses/DL/Slides\\_11\\_12/measures.pdf](http://disi.unitn.it/~bernardi/Courses/DL/Slides_11_12/measures.pdf)

<sup>21</sup> Frecvența termenului „licență” utilizând formula de calcul

<sup>22</sup> Frecvența inversată a documentului utilizând formula de calcul

<sup>23</sup> <https://nlp.stanford.edu/IR-book/html/htmledition/inverse-document-frequency-1.html>  
[http://disi.unitn.it/~bernardi/Courses/DL/Slides\\_11\\_12/measures.pdf](http://disi.unitn.it/~bernardi/Courses/DL/Slides_11_12/measures.pdf)

nu apare în corpus, numărătorul va fi deîmpărțit la zero, astfel se recomanda ca numitorul să fie adunat cu 1, ca în exemplul următor  $1 + |\{j : t_j \in dj\}|$ .

Folosind cele 2 rezultate de mai sus TFlicență și IDFlicență vom afla importanța termenului „licență” din corpus. Formula după care se face calculul este următoarea:

$$\text{TF-IDF} = \text{TF} * \text{IDF}^{24}$$

Ponderea pentru cuvântul (termenul) “licență” este:

$$\text{TF-IDF} = 0.01 * 1.30 = 0.13.$$

### Exemplu 1

Mai jos avem alt exemplu mai concludent folosind două fraze, utilizând bucăți din codul aplicației pentru a obține rezultatul algoritmului TF-IDF.

*„doc1 = 'Prezentarea lucrării de licență în fața comisiei de examinare este cel puțin la fel de importantă ca și redactarea acesteia.'”*

*doc2 = „'Lucrarea de licență este partea cea mai importantă a examenului de încheiere a studiilor universitare – ciclul I'”*<sup>25</sup>

---

<sup>24</sup> <https://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html>

<sup>25</sup> Documentele folosite pentru a folosi TF-IDF

```

1 from sklearn.feature_extraction.text import TfidfVectorizer
2 import pandas as pd
3 from nltk.tokenize import word_tokenize
4 from nltk.corpus import stopwords
5 import string
6
7 def process(text):
8     text = ''.join([word for word in text if word not in string.punctuation])
9     text = text.lower()
10    text = ' '.join([word for word in word_tokenize(text) if word not in stop_words])
11    return text
12
13 vectorizer = TfidfVectorizer()
14 stop_words = stopwords.words('romanian')
15
16 doc1 = 'Prezentarea lucrării de licență în fața comisiei de examinare este cel puțin la fel de importantă ca și redactarea acesteia.'
17 doc2 = 'Lucrarea de licență este partea cea mai importantă a examenului de încheiere a studiilor universitare - ciclul I'
18
19 sparse_matrix = vectorizer.fit_transform([process(doc1), process(doc2)])
20
21 doc_term_matrix = sparse_matrix.todense()
22 df = pd.DataFrame(doc_term_matrix,
23                  columns=vectorizer.get_feature_names(),
24                  index=['document1', 'document2'])

```

Imaginea 3<sup>26</sup>

Rezultatul algoritmului este:

	acesteia	ciclul	comisiei	examenului	examinare	fața	importantă	licență	lucrarea
document1	0.31603	0.000000	0.31603	0.000000	0.31603	0.31603	0.224858	0.224858	0.000000
document2	0.00000	0.353278	0.00000	0.353278	0.00000	0.00000	0.251360	0.251360	0.353278
	lucrării	partea	prezentarea	puțin	redactarea	studiilor	universitare	încheiere	și
document1	0.31603	0.000000	0.31603	0.31603	0.31603	0.000000	0.000000	0.000000	0.31603
document2	0.00000	0.353278	0.00000	0.00000	0.00000	0.353278	0.353278	0.353278	0.00000

### 2.3. Analiza textului extras

Făcând un simplu raport cu TF-IDF putem constata cât de important este un termen sau cât de rar este el printre ceilalți termeni din corpus. Cu cât valoarea crește cu atât termenul respectiv va fi mai important, cu cât valoarea scade cu atât termenul respectiv apare de mai multe ori.

Raportat la aplicația curentă CourSeeker, se utilizează termenul descris mai sus corpus, ca fiind format din totalitatea mulțimii cuvintelor reprezentate sub forma unui vector. Fiecare document va fi parsat prin intermediul unei metode Python specifice denumită succint “extract\_text” conținută de clasa “TextProcessor”, clasă care

<sup>26</sup> Imaginea 3Ș Codul pentru generarea ponderilor returnate de algoritmul TF-IDF

utilizează mai multe pachete Python pentru a procesa documentul. În urma parsării se obține un vector de cuvinte, care va fi mai apoi înserat în baza de date sub forma unui tip de date specific și anume – Strâng<sup>27</sup>.

Înainte de a fi inserat și odată formată mulțimea cuvintelor utilizate, în cadrul fiecărui curs se face o rafinare a cuvintelor de legătură, cuvinte care nu aduc aport în calculul celui mai frecvent cuvânt, prin eliminarea acestora.

În *imaginea 2<sup>28</sup> de pe pagina 14*, se poate vedea o structură a obiectului din baza de date care corespunde unui curs. Se observă câmpul “extracted\_text” care reține mulțimea cuvintelor returnate de metoda descrisă mai sus.

```
{
  "_id" : ObjectId("5cdc097d79297f1df4e33d92"),
  "title" : "Introducere",
  "content" : "Introducere",
  "createdAt" : ISODate("2019-05-15T12:43:41.200Z"),
  "updatedAt" : ISODate("2019-06-16T22:22:41.181Z"),
  "_v" : 0,
  "author" : ObjectId("5cdc072cfad08613e0976a0e"),
  "id" : "5cdc097d79297f1df4e33d92",
  "extracted_text" : "Dr. Sabin Buraga profs.info.uaic.ro/~ busaco / Tehnologii Web chestiuni organizatorice Dr. Sabin Buraga profs.info.uaic.ro/~ busaco / Web URI develop ent POST design XML CGI proxy N - tier HTTP WSDL browser DOM resource XSS SID validation application CSS meta - dat SOA servic e JSON framework WS data format GET SSI module SOAP representation SQL injection DTD cookie SAX push Ajax tag Comet mash - MVC social MIME PI schema REST deployment HTML model XPath session etc . ? - \\ _ / Dr. Sabin Buraga profs.info.uaic.ro/~ busaco / Dac nu ti nu - nicio problem ! https //imgur.com/PG 4 B l fJ unii vor cunoa te la finalul acestui curs al ii ... in prima sesiune Dr. Sabin Buraga profs.info.uaic.ro/~ busaco / Materia va fi divizata i n cuno tin e de baz obligatorii cuno tin e mai avansate op ionale Dr. Sabin Buraga profs.info.uaic.ro/~ busaco / Subiecte abordate concepte arhitectura WWW protocolul HTTP + cookie - limbaajul de marcare HTML + foli de stiluri CSS laborator inginerie Web C et al . programare Web servere de aplica exemplu PHP modelarea datelor via XML XPath DTD DOM SAX SimpleXML pentru HTML XML de la SOA la servicii Web SOAP REST mash - ups Dr. Sabin Buraga profs.info.uaic.ro/~ busaco / Evaluare proiect clasa M max . 10 puncte / clasa B max . 7 puncte P S S laborator A S il laborator S demo sesiune nu sunt permise framework - uri teste scrise T l T 2 = 2.5 puncte T 3 = 5 puncte S 9 S l4 curs Dr. Sabin Buraga profs.info.uaic.ro/~ busaco / Evaluare punctaj final P * 0.2 + A * 0.2 + S * 0.5 + T * 0.2 cu T = sum T l 3 P 5 & A 5 & S 5 P A P < S A < S T nu se 2 puncte penalitate la P A / sau S reevaluat profs.info.uaic.ro/~ busaco / teach / courses /web/web - exam.html Dr. Sabin Buraga profs.info.uaic.ro/~ busaco / Situl disciplinei Tehnologii Web desf urare reguli resurse contact http //tinyurl.com/tehnologii - web Dr. Sabin Buraga profs.info.uaic.ro/~ busaco / Titularul disciplinei Tehnologii Web Dr. Sabin Buraga 1974 2021 pasionat de tehnologii Web din 1996 al cursului Tehnologii Web 2000 autor al primelor tehnologiilor Web 2001 - ului semantic 2004 doctorat Cum Laude in tehnologii Web 2004 laureat al Academiei Române 2005 co - autor peste 10 alte volume de specialitate Profesor Bologna 2017 profs.info.uaic.ro/~ busaco / Dr. Sabin Buraga profs.info.uaic.ro/~ busaco /"
}
```

Corpusul reprezentat ca în exemplul din imaginea de mai sus este trimis ca parametru în aplicație către metoda “calculateSimilarity” care va utiliza algoritmul TF-IDF,folosindu-se de un pachet Python „[TfidfVectorizer](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html)<sup>29</sup>” .

<sup>27</sup> <https://ocw.cs.pub.ro/courses/ii/lab/laborator1>

<sup>28</sup> Modul de stocare al unui curs in baza de date si obiectul continut de acesta.

<sup>29</sup> [https://scikit-learn.org/stable/modules/generated/sklearn.feature\\_extraction.text.TfidfVectorizer.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html)

## 2.4. Similaritate cosinus

### 1.1.1. Importanța algoritmului

Similaritatea cosinus, este o modalitate de a calcula similaritatea a două texte, folosindu-se cosinusul unghiului dintre doi vectori A și B.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad 30$$

Continuând ideea din capitolul 2.1 de pe pagina 13, știm că algoritmul TF-IDF ne returnează ca și rezultat un vector multi-dimensional în care regăsim o pondere atribuită fiecărui termen. O altă modalitate de calcul al similarității dintre două documente pe lângă modalitatea descrisă mai sus la capitolul 2.4.1 este [Distanța Euclidiană](#)<sup>31</sup>. Din nefericire, această soluție nu este una tocmai eficientă datorită faptului că un articol poate fi o copie a altuia cu conținut multiplicat, iar atunci acest algoritm ne sugerează că aceste două documente sunt similare când de fapt sunt două articole diferite. Tocmai de aceea algoritmul Similaritate cosinus are o abordare diferită și normalizează vectorii documentelor, explicația o puteți găsi mai jos în următorul capitol.

### 2.4.1. Similaritatea textului

Similaritate cosinus normalizează toți vectorii, pentru fiecare document, astfel fiecare vector va avea o magnitudine de 1 unitate menținându-se astfel rația dintre cuvinte.

Prin urmare, aplicând formula pe cei doi vectori A și B, dacă aceștia au aceeași orientare atunci au similaritate cosinus de 1, dacă ei vor fi orientați la 90 ° unul față de celălalt atunci vor avea o similaritate 0 , iar dacă cei doi vectori sunt diametral opuși au o asemănare -1, independent de magnitudinea lor. Cu cât unghiul dintre cei doi vectori este mai mic, cu atât similaritatea cosinus este mai

---

<sup>30</sup> Formulă preluată de pe <https://www.machinelearningplus.com/nlp/cosine-similarity/e>

<sup>31</sup> <http://mathonline.wikidot.com/the-distance-between-two-vectors>

mare. Bineinteles, nu este cea mai buna metoda de a construi un sistem de recomandari, dar simplitatea și capacitatea de procesare a documentelor este impresionantă.

### Exemplu 2

Outputul pentru cele două documente folosite în Exemplul 1 de la pagina 16 utilizând funcția

```
from sklearn.metrics.pairwise import cosine_similarity
```

va fi următorul:

$[1, 0.11304078] [0.11304078, 1]$ <sup>32</sup>

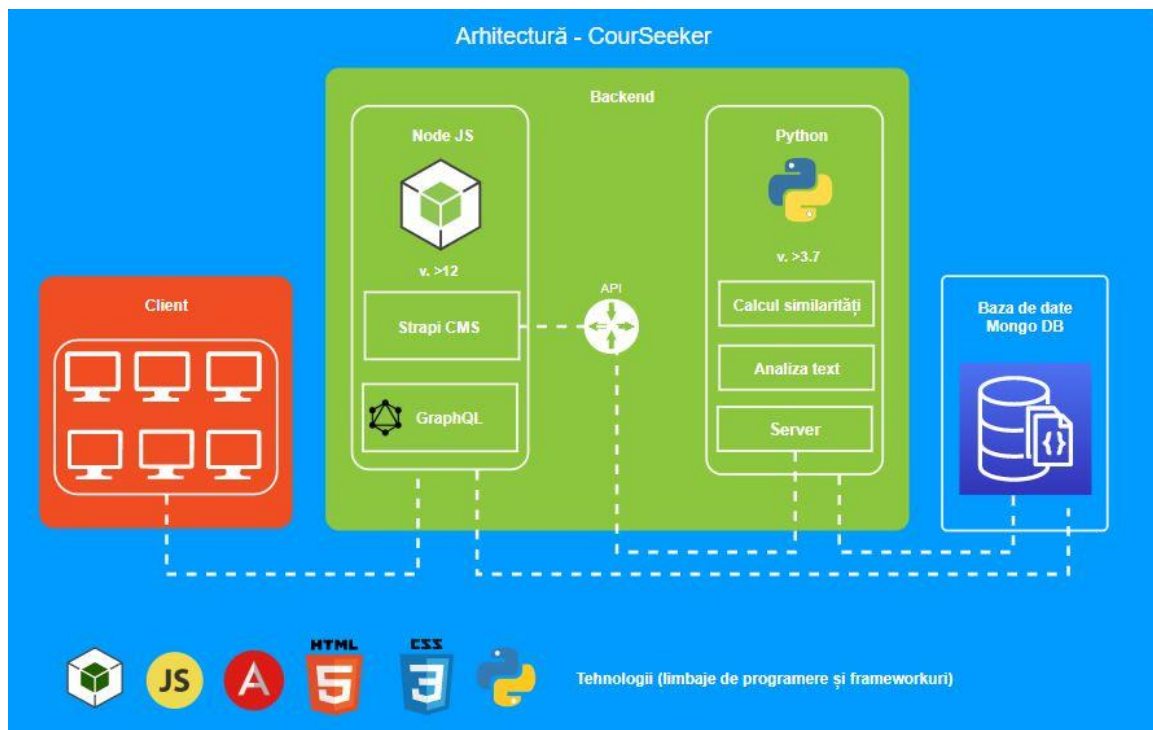
---

<sup>32</sup> Rezultatul funcției cosine\_similarity



### 3. Arhitectura aplicației

Aplicația este împărțită în 3 părți: partea de server, partea de client și partea în care se realizează stocarea datelor. Informația stocată în baza de date este legătura dintre NodeJS <sup>33</sup>și Python<sup>34</sup>, în timp ce partea de client intra în posesia informației pe baza unui API (specificat în notele paginii 17) generat de admin.



Imagine 3<sup>35</sup>

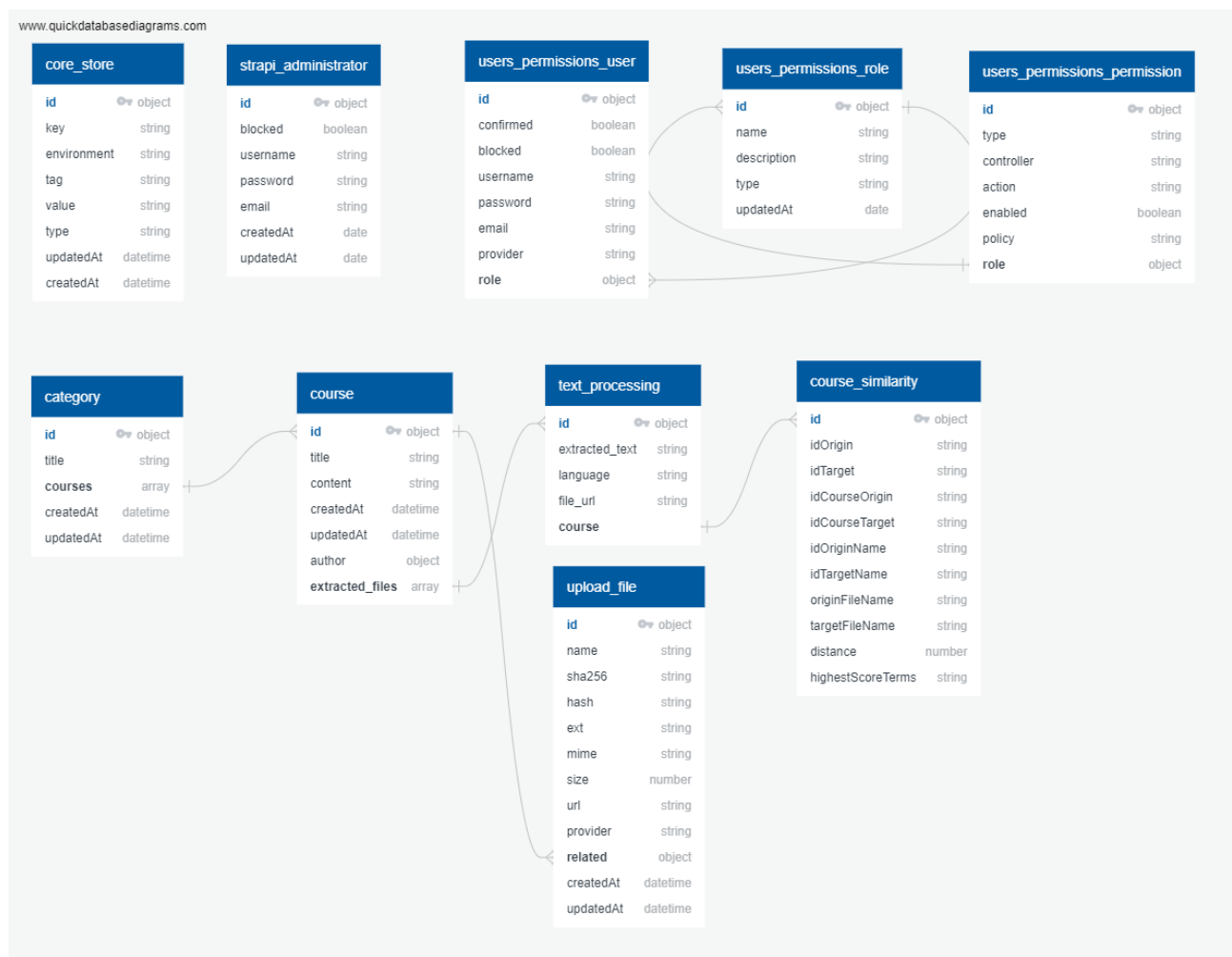
<sup>33</sup> <https://nodejs.org/en/>

<sup>34</sup> <https://www.python.org/>

<sup>35</sup> Arhitectura aplicației

### 3.1. Baza de date

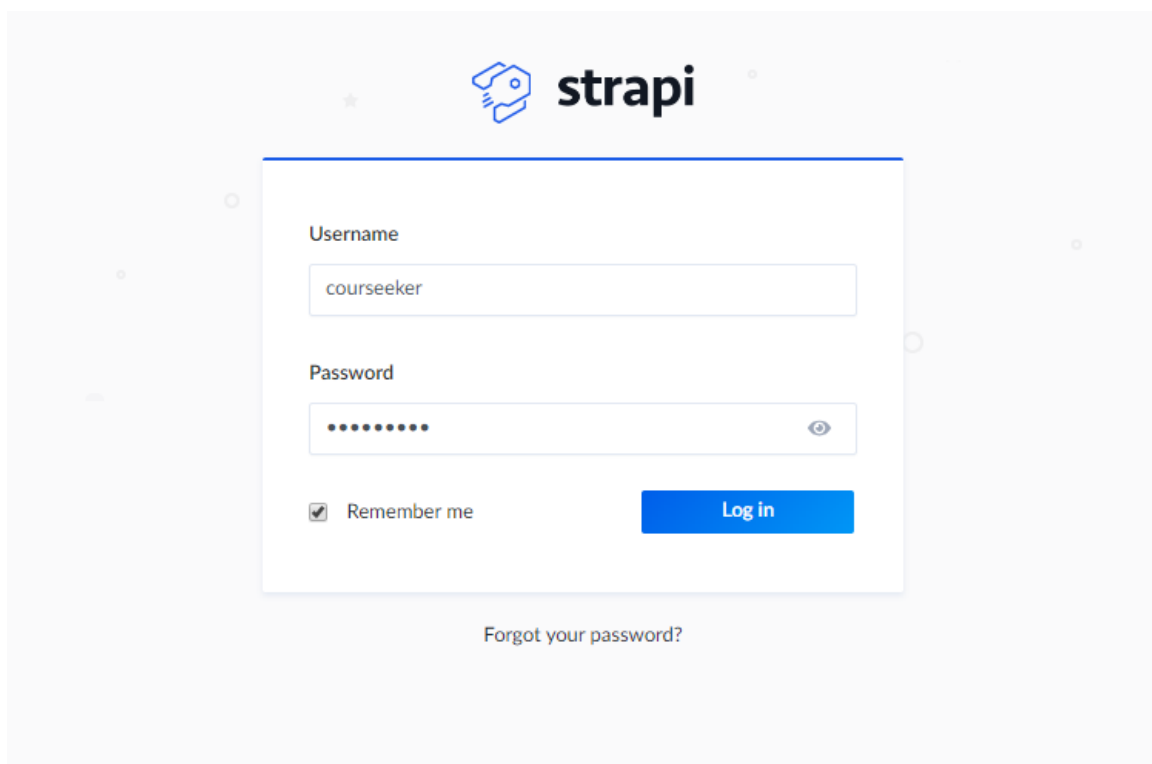
Baza de date folosită este MongoDB<sup>36</sup>, este o bază de date foarte versatilă, potrivită noilor tipuri de aplicații de pe piață. Fiind foarte flexibilă, timpul de dezvoltare al aplicației scade și nu în ultimul rând are o performanță destul de bună pentru interogări. MongoDB vine cu un concept destul de diferit față de celelalte baze de date pe care le folosim mai des, și anume că este bazată pe colecții de documente, un document putând conține unul sau mai multe câmpuri față de următorul document din aceeași colecție. Mai jos puteți vedea și infrastructura bazei de date.



<sup>36</sup> Baza de date

### 3.2. Admin

Accesul se face printr-o pagină de logare, vezi  *imaginea 3*<sup>37</sup> prezentată mai jos. După acest pas utilizatorul care va trebui să dețină și un rol de administrator și va fi redirecționat către panoul de administrare în care va putea accesa cursurile, categoriile din care cursurile fac parte și utilizatorii înregistrați în aplicație. Rolul de administrator presupune control absolut asupra datelor salvate în baza de date Mongo DB<sup>38</sup>.



Imaginea 3.1<sup>39</sup>

Partea de administrare are la bază NodeJs care va „colabora” cu serverul de Python pentru încărcarea datelor în baza de date, analiza textelor dar și pentru calcularea similarităților dintre fișiere. Pe partea de client despre care vom vorbi un

<sup>37</sup> Pagina de logare in sectiunea Admin.

<sup>38</sup> <https://www.mongodb.com/>

<sup>39</sup> Pagina de autentificare pe partea de admin

pic mai jos, datele sunt accesate prin GraphQL <sup>40</sup>, un tool <sup>41</sup> foarte puternic care poate face interogări în baza de date pe query<sup>42</sup>-uri foarte mari într-un timp foarte scurt.

```
export const SIMILAR_COURSES_QUERY_TO = gql`
query SimilarCourses2($name: String!) {
  courseIsOrigin: similarities(
    where: { idOriginName: $name }
    sort: "distance:desc"
    limit: 10
  ) {
    idOrigin
    idTarget
    idCourseOrigin
    idCourseTarget
    idOriginName
    idTargetName
    originFileName
    targetFileName
    highestScoreTerms
    distance
  },
  courseIsTarget: similarities(
    where: { idTargetName: $name }
    sort: "distance:desc"
    limit: 10
  ) {
    idOrigin
    idTarget
    idCourseOrigin
    idCourseTarget
    idOriginName
    idTargetName
    originFileName
    targetFileName
    highestScoreTerms
    distance
  }
}
`;
```

Imaginea 3.1.1 <sup>43</sup>

---

<sup>40</sup> <https://graphql.org/learn/>

<sup>41</sup> <https://ro.bab.la/dictionar/engleza-romana/tool>

<sup>42</sup> <https://ro.bab.la/dictionar/engleza-romana/query>

<sup>43</sup> Exemplu interogare cursuri similare

### 3.1.1 [Strapi](#)<sup>44</sup>

Strapi este un Headless CMS <sup>45</sup>(Content Management System), open source care ne pune la dispoziție o interfață intuitivă și de mare ajutor, alături de o funcționalitate „elastică” care te ajută să îți organizezi informația pusă la dispoziția utilizatorului în așa fel încât poți să adăuga oricând, să ștergi sau să creezi legături între entități. Deasemenea poți crea API<sup>46</sup>-uri care pot fi folosite în partea de client a aplicației, parte care va fi descrisă mai pe larg în capitolul 3.2.

Strapi este într-o continuă dezvoltare și vine la pachet cu mai multe unelte potrivite pentru un CMS, cu fiecare versiune fiind tot mai stabil , fapt care ne ajută și din punct de vedere al securității. Ideea de a face un CMS de la zero nu ne ajută în mod exclusiv deoarece că orice altă aplicație lansată în producție poate fi vulnerabilă. Astfel, e posibil că atacurile cibernetice ulterioare asupra aplicației să nu aibă câștig de cauză.

Odată autentificat, adminul are access și la o secțiune de „Roles & Permission” (Roluri și Permisuni), o secțiune în care adminul poate da drepturi unui utilizator sau poate suspenda anumite drepturi.

### 3.1.2. La ce se referă un curs ?

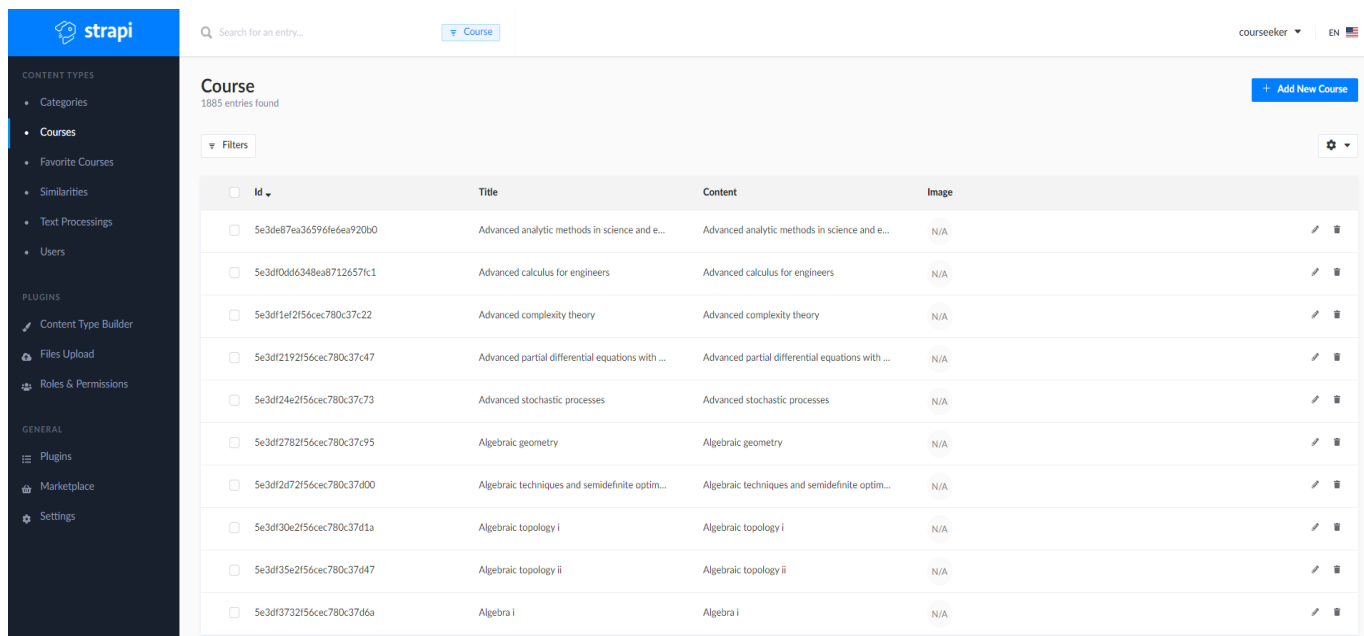
Imaginea de mai jos ne prezintă pagina cursurilor încărcate de utilizatori. Structura unui curs este identificat prin 4 caracteristici: Title (titlu), Content (conținut), Image (imaginea), Files (cursul său fișierul care urmează să fie încărcat).

---

<sup>44</sup> <https://strapi.io/>

<sup>45</sup> <https://www.gatsbyjs.org/docs/headless-cms/>

<sup>46</sup> API – este o funcționalitate care creează un punct de access între 2 entitati, in cazul nostru între cea de Client și cea de Admin.



Imaginea 4<sup>47</sup>

### 3.3. Client

#### 3.3.1. Scurtă prezentare a aplicației

Partea de client și partea de analiză a textului prezentată la punctul 2.2, realizată în Python, sunt punctele cheie ale aplicației. Desigur, are o importanță majoră și partea de administrare unde se încarcă cursurile de către administrator. Partea de client afișează datele într-un mod simplist și clar pentru ca utilizatorul să poată găsi informația într-un mod cat mai eficient.

#### 3.3.2. Pagina principală

Pe prima pagină a aplicației se poate observa în partea de sus meniul aplicației, care conține 3 secțiuni printre care „Home”, „Categories” și „Courses”. Pe pagina de home avem un subtitlu „Latest Courses” care ne va furniza ultimele cursuri încărcate de către administrator sau de către sistem atunci când caută automat după cursuri și categoriile aferente.

<sup>47</sup> Imaginea 4 reprezintă pagina cursurilor încărcate în Admin.

Designul aplicației este unul simplist, orientat către utilizabilitate și nu către complexitate, este scalabil pe toate dispozitivele, începând de la telefoane mobile până la echipamente de tip desktop. CourSeeker va permite utilizatorului să vizualizeze cursuri cu ușurință dar și cursurile similare cu acesta. Stilurile aplicației au fost scrise în [SASS](https://sass-lang.com/)<sup>48</sup>, apoi partea de compilare a stilurilor este făcută de librăria Webpack<sup>49</sup>.

Pentru a avea o variantă pentru producție se va face compilarea folosindu-se [NodeJS](https://nodejs.org/en/)<sup>50</sup>(se va instala și pachetul [npm](https://www.npmjs.com/)<sup>51</sup>) și Angular<sup>52</sup> rulând comanda:.

```
$ npm run build
```

---

<sup>48</sup> <https://sass-lang.com/>

<sup>49</sup> <https://webpack.js.org/>

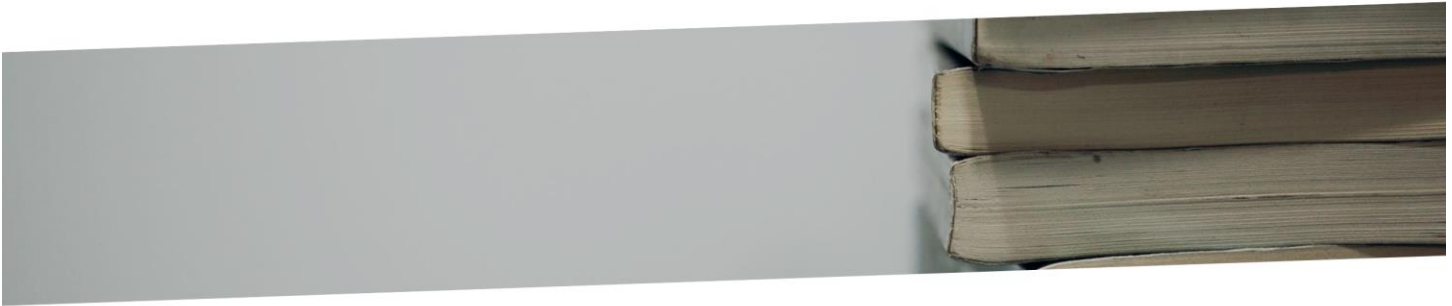
<sup>50</sup> <https://nodejs.org/en/>

<sup>51</sup> <https://www.npmjs.com/>

<sup>52</sup> <https://angular.io/>

# Discover new courses to improve your skills.

Search through hundreds of courses



## Latest courses

artificial intelligence - 2017  
 bioinformatics  
 business  
 fine arts  
 humanities  
 investigarea criminalitatii  
 informatică  
 mathematics  
 other

Investigarea criminalitatii informatice - ICI02  
 by system / Jun 17, 2019

Investigarea criminalitatii informatice - ICI03  
 by system / Jun 17, 2019

Investigarea criminalitatii informatice - ICI05  
 by dtalaba / Jun 17, 2019

Investigarea criminalitatii informatice - ICI06  
 by dtalaba / Jun 17, 2019

Investigarea criminalitatii informatice - ICI07  
 by system / Jun 17, 2019

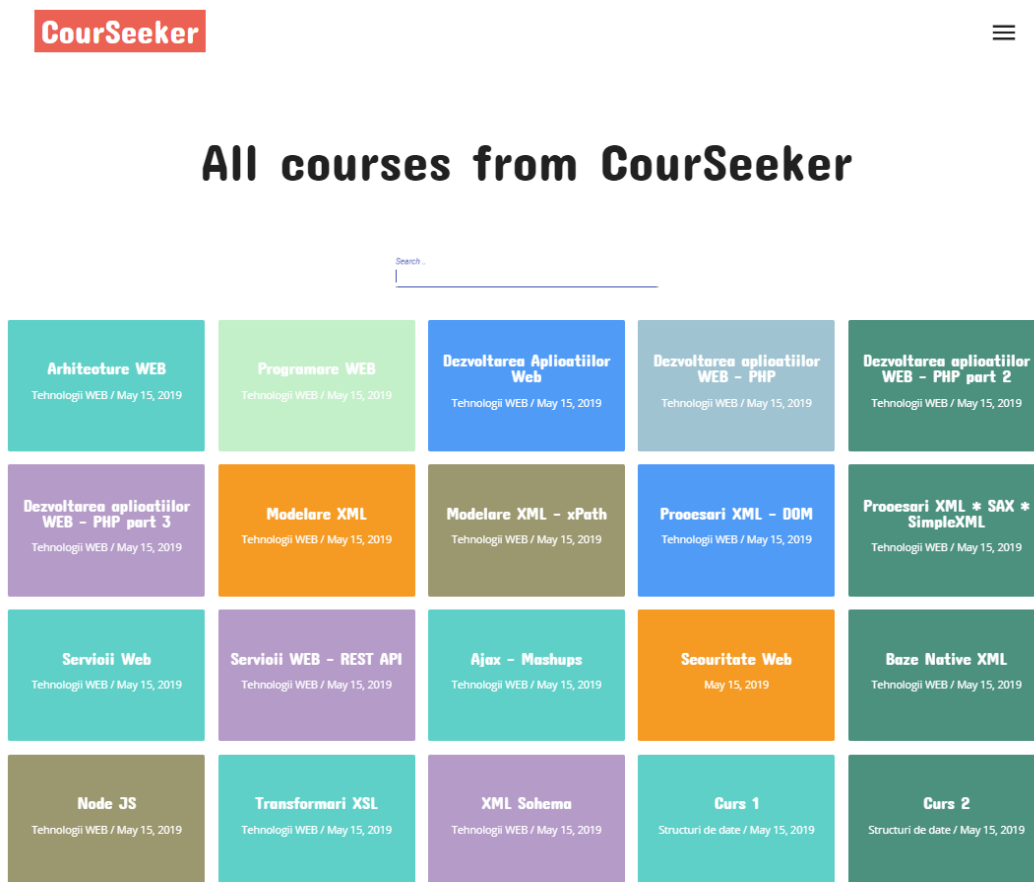
Imaginea 5<sup>53</sup>

<sup>53</sup> Imaginea 5 reprezintă pagina principală a aplicației



### 3.3.3. Pagina „Courses”

Pe această pagină utilizatorul poate căuta un anumit curs dintr-o listă lungă de cursuri, nefiind filtrate după categorii sau oricare alte criterii. Cursurile se vor încărca pe pagina în timp ce utilizatorul va naviga spre josul paginii web.

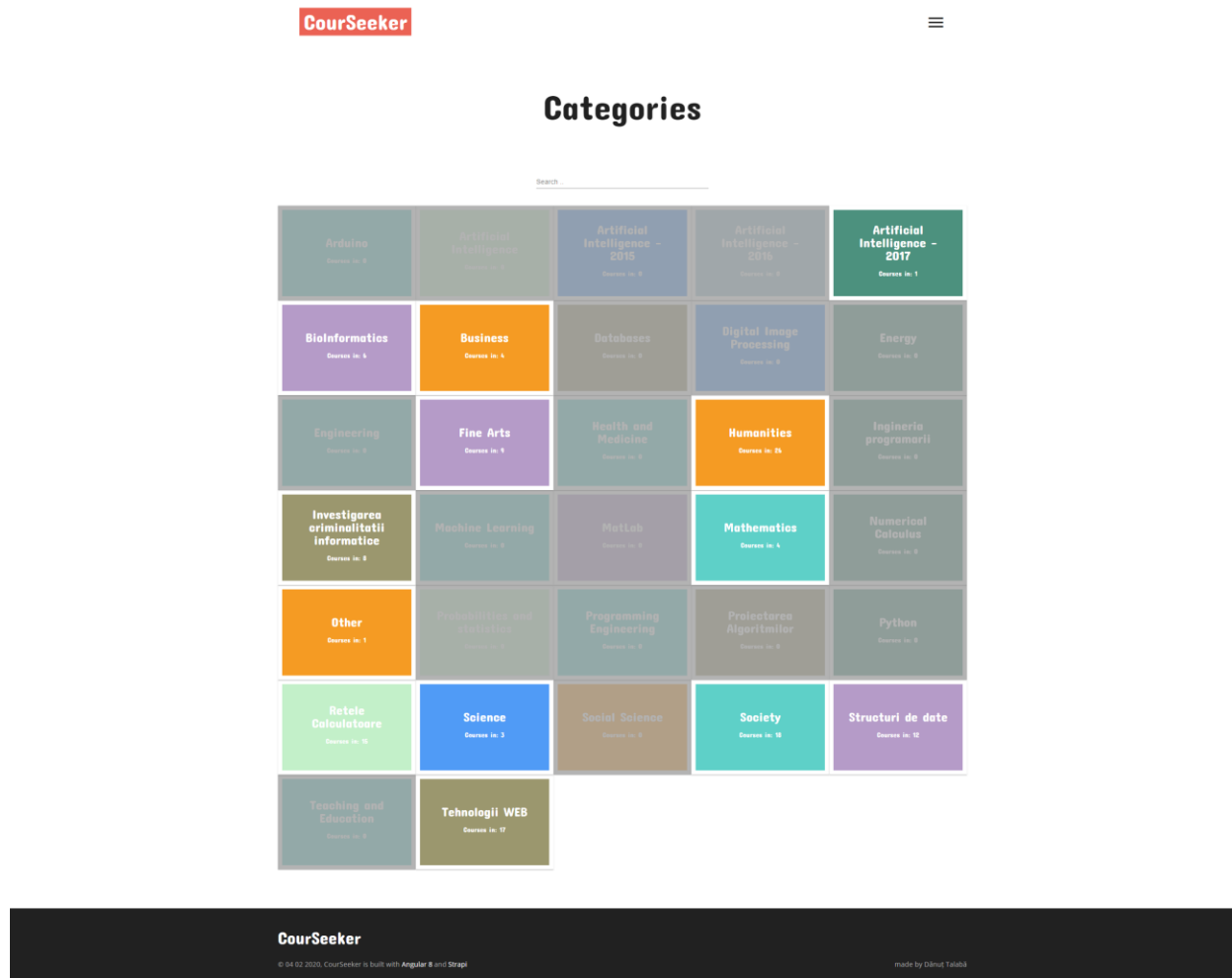


Imaginea 3.2.3<sup>54</sup>

<sup>54</sup> Pagina „Courses” a aplicației

### 3.3.4. Pagina „Categories”

Acest modul cuprinde toate categoriile din care poate face un curs, cursul poate să aibă una sau mai multe categorii.



Imaginea 3.2.4<sup>55</sup>

### 3.3.5. Pagina cursului vizualizat

Pe această pagină vor fi afișate toate fișierele care fac parte din cursul respectiv. De notat este că similaritățile cu celelalte cursuri se vor calcula în funcție de fiecare fișier, ele

<sup>55</sup> Pagina „Categories” a aplicație

luându-se două câte două, astfel se va urma procesul descris la punctul 2.2. Fișierele vor fi ordonate după subcategorii, dacă este cazul. De exemplu pentru cursurile luate de pe site-ul MIT <sup>56</sup> categoriile afișate vor fi exact ca cele de pe site-ul celor de la MIT, pentru a eficientiza procesul de căutare.

CourSeeker

Ambient Intelligence

Course information

You can find more about this course following the next topics from MIT:

Projects

ss\_scene\_paper.pdf

az\_augment\_obj.pdf

az\_airoi\_airoi.pdf

az\_airoi\_update.pdf

az\_scene\_paper.pdf

az\_augment\_obj.pdf

az\_airoi\_airoi.pdf

az\_airoi\_update.pdf

az\_scene\_paper.pdf

ci\_augment\_obj.pdf

ci\_airoi\_airoi.pdf

ci\_airoi\_update.pdf

ci\_scene\_paper.pdf

ss\_augment\_obj.pdf

ss\_final\_airoi.pdf

ss\_airoi\_airoi.pdf

ss\_airoi\_update.pdf

ss\_scene\_paper.pdf

Assignments

az\_week5.pdf

az\_week6.pdf

az\_week7.pdf

az\_week8.pdf

az\_week4.pdf

az\_week5.pdf

az\_week6.pdf

az\_week7.pdf

az\_week8.pdf

ci\_week2.pdf

ci\_week3.pdf

ci\_week4.pdf

ci\_week5.pdf

ci\_week6.pdf

ci\_week7.pdf

ci\_week8.pdf

ss\_comment\_week2.pdf

ss\_comment\_week3.pdf

ss\_comment\_week4.pdf

ss\_comment\_week5.pdf

ss\_comment\_week6.pdf

ss\_comment\_week7.pdf

ss\_comment\_week8.pdf

az\_week4.pdf

Lecture Notes

week1\_om\_intro.pdf

week2\_om\_int\_aug.pdf

week3\_ci\_hipoint.pdf

week3\_om\_context.pdf

week4\_pash\_singh.pdf

week5\_az\_ubicom.pdf

week5\_om\_ubicom.pdf

week5\_salikad1.pdf

week5\_salikad2.pdf

week6\_om\_recores.pdf

week8\_az\_airoi.pdf

week8\_ss\_airoi.pdf

week8\_ss\_airoi.pdf

week8\_ss\_airoi.pdf

Or you can find more about this course [here](#).

Posted by system at Jan 30, 2020

Lecture notes

Ambient Intelligence

Analysis text

Show extracted text from all files

CourSeeker

© 04/02/2020, CourSeeker is built with Angular 8 and Strapi

made by Ghinut Tabak

## Imaginea 3.2.5.1

<sup>56</sup> <https://ocw.mit.edu/courses/>

35

În partea de jos a paginii se poate observa textul extras în faza de analiză, a se vedea imaginea 3.2.5.2.

---

## **Lecture notes**

Advanced writing seminar

---

## **Analysis text**

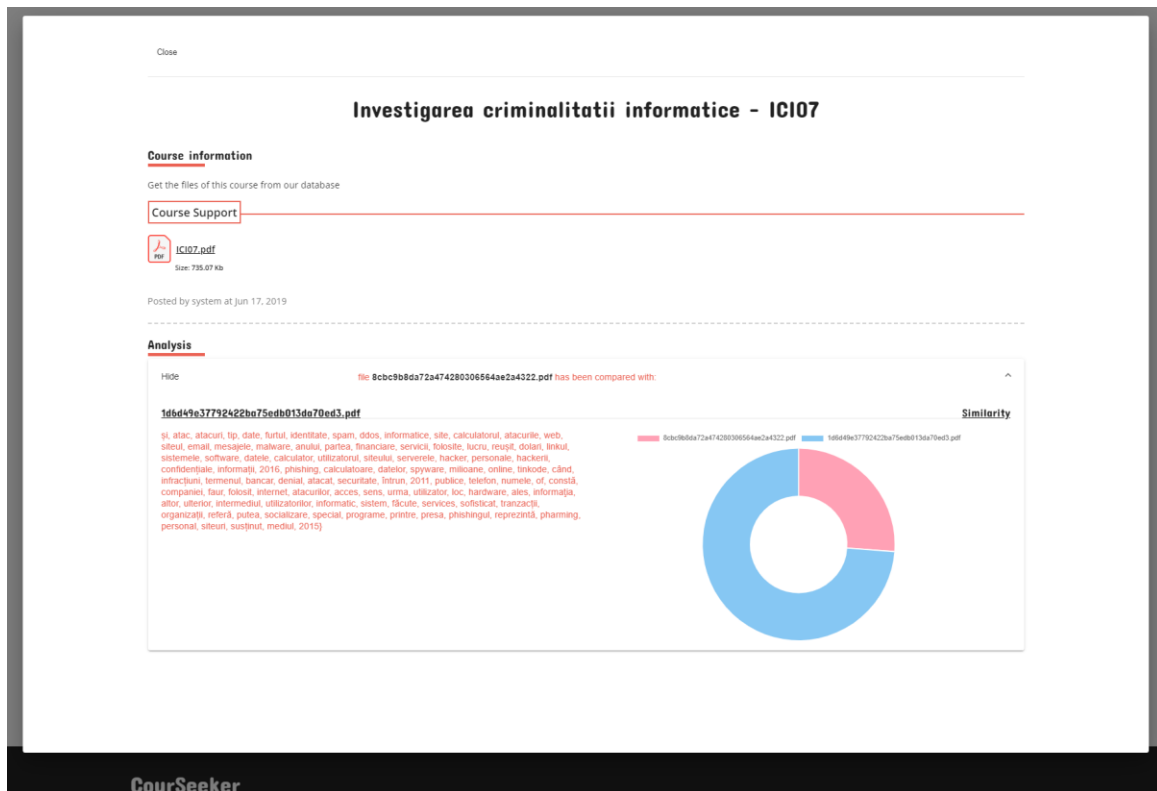
Hide extracted text from all files

guidelines grading papers 1 paper receive score 16 score 6 highest 2 use passive voice acceptable social science technical writing long interfere intrude upon obscure meaning passage active voice tends read better engaging 3 grading writing content still content likely add strength essay 46 point papers considered upperhalf six point essay characterized following excellent organization ideas clarity conciseness virtually errorfree grammar usage five point essay display features 6point essay slightly weaker clarity concision organization four point essay characterized following basic competence grammar usage lacks structural organizational sophistication 5 6 point essay 13 point papers considered lowerhalf three point essay characterized following overly formulaic organization lacking organization problems grammar usage vague wordy construction excessive use passive voice excessive nominalization two point essay compound problems three point essay display consistent flaws syntax diction grammar spelling significant flaws organization lack overall coherence usage terms without prior definition one point essay compound weaknesses two point essay

Imaginea 3.2.5.2

### 3.3.6. Pagina similarităților

Ajungând în punctul acesta utilizatorul va avea access la cursuri similare cu cel citit anterior. Cursurile vor fi afișate prin intermediul unui modal, acesta fiind un avantaj, deoarece utilizatorul are posibilitatea de a naviga printre cursuri mai ușor. Din punct de vedere al stilurilor și așezarea în „pagină”, informațiile din acest modal sunt afișate asemănător cu cele de la punctul 3.2.5. Pe partea de analiză nu va mai exista textul fișierelor din acel curs, abordarea va fi un pic diferită pentru a da utilizatorului mai mult context cu privire la ceea ce i-a fost recomandat. Prin urmare vom avea în partea dreaptă procentajul obținut în urma calculării similarității dintre cele două fișiere, iar în partea stângă vor fi afișați cei mai importanți o sută de termeni din corpusul analizat de către TFIDF prezentat la punctul 2.2. Toate aceste detalii se pot vedea și în imaginea de mai jos.



### 3.4. Motivul utilizării librăriilor

Aplicațiile pe parte de client și backend au evoluat mult în ultima vreme. Dacă acum câțiva ani aplicațiile erau scrise folosind HTML, CSS și foarte puțin JavaScript, iar pe partea de backend erau folosite limbaje mai populare precum „PHP”, „ASP.NET” sau „Python”, de asemenea găsim librării într-un număr foarte limitat pentru ele, odată cu evoluția limbajelor de programare s-au pus bazele unor librării și frameworkuri care au ajuns să fie dezvoltate în comunități IT. Avantajul major în această ecuație este faptul că multe din aceste frameworkuri sau librării sunt open-source, ceea ce înseamnă că pot fi folosite de către oricine. Folosirea unei librării sau a unui framework nu presupune doar instalarea și utilizarea acestora, presupune integrare cu aplicația existentă, schimbări de arhitectură pentru a întruni scopul aplicației respective și nu în ultimul rând, alocarea timpului de development în alte zone ale aplicației care necesită mai multă atenție și în care nu se poate refolosi cod.

### 3.5. Librării și Framework-uri

- a. Mongo DB
- b. Node JS
  - 1. Strapi
  - 2. Npm
- c. Angular 8
- d. Webpack
- e. Python
  - 1. nltk
  - 2. flask
  - 3. pdfplumber
  - 4. pandas
  - 5. PyPDF2
  - 6. Scipy

## 4. Concluzii

În această lucrare am prezentat o aplicație care recomandă utilizatorului cursuri similare cu domeniul său de interes. Aceasta are momentan ca punct de plecare aproximativ 300 de cursuri din Facultatea de Informatică dar și cursuri din afara țării, ca MIT, de unde s-a înregistrat un număr de aproximativ 800 de cursuri.

Poate deveni un factor de interes pentru majoritatea studenților, întrucât conține și cursurile predate în anii anteriori și există posibilitatea de a face ușor o retrospectivă a unui curs, de a-și stabili domeniul de interes sau de a recomanda unui alt coleg un anumit curs printr-un simplu link.

În același timp consider că aplicația CourSeeker are ca punct forte inclusiv unicitatea ideii, aceasta survenind din experiența proprie în căutarea unor anumite materiale de studiu pentru o anumită materie, sau de ce nu, căutarea unor documentații sau texte similare cu anumite interese, activitate care ar ușura cu siguranță procesul de învățare sau aprofundare.

Ca orice aplicație, intervin desigur și obiective viitoare în dezvoltarea acesteia care să îi asigure o utilizare facilă și ușoară. Între aceste obiective poate fi menționată și funcționalitatea de căutare a unor cursuri similare pe baza unui alt curs încărcat de un utilizator, idee care ar favoriza utilizarea aplicației. Totodată se poate menționa și extinderea domeniilor de interes cu materiale de studiu și cursuri din cadrul cât mai multor universități și facultăți.

Se poate spune că punctul de plecare al aplicației poate reprezenta un domeniu de interes atât pentru studenți/elevi, cât și pentru profesori, facilitând accesul la informații academice ale cât mai multor universități și oferind în același timp procente ale similarităților expuse în mod grafic, concludent.

Astfel putem conchide că utilizând procesarea de limbaj natural și domeniul Inteligență Artificială, aspecte extrem de des menționate în ultima vreme, aplicația CourSeeker își propune să utilizeze tehnologii moderne și de viitor, pentru facilitarea căutării de cursuri în mediul academic, pe baza unui domeniu de interes.

## 5. Bibliografie

- <https://dexonline.ro/d>
- <https://ai.stanford.edu/~nilsson/MLBOOK.pdf>
- [http://disi.unitn.it/~bernardi/Courses/DL/Slides\\_11\\_12/measures.pdf](http://disi.unitn.it/~bernardi/Courses/DL/Slides_11_12/measures.pdf)
- <http://www.tfidf.com/>
- <https://nodejs.org/en/>
- <https://strapi.io/>
- <https://reactjs.org/>
- <http://nlp.town/blog/sentence-similarity/>
- <https://www.machinelearningplus.com/nlp/cosine-similarity/>
- [https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature\\_extraction.text](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_extraction.text)
- <https://scikit-learn.org/0.19/modules/classes.html#module-sklearn.metrics.pairwise>
- <https://pythonhosted.org/PyPDF2/>
- <https://www.nltk.org/>
- <https://textextract.readthedocs.io/en/stable/>
- <https://docs.python.org/3/>
- <https://valor-software.com/ng2-charts/>