

Colegiul Național „Spiru Haret” Târgu Jiu

- Cleverly -

DOCUMENTAȚIE TEHNICĂ



|  |  |
| --- | --- |
| *Coordonator:*  Prof. Dumitru-Flavius Boian | *Elevi:*  Ana-Natalia Loga  Anais-Valentina Spafiu |

Târgu Jiu, 2025

**Cuprins**

1. INTRODUCERE
2. ANALIZA PIEȚEI
3. IMPLEMENTAREA APLICAȚIEI
4. ECHIPA
5. CODUL SURSĂ
6. BIBLIOGRAFIE
7. INTRODUCERE
8. **Descriere generală**

**Cleverly** este o aplicație web interactivă și inteligentă destinată tuturor celor care își doresc să învețe și să înțeleagă informații eficient. Prin intermediul Inteligenței Artificiale, aplicația generează flashcard-uri și teste pe baza materialelor de studiu introduse de către utilizator, fie în format PDF, fie ca text sau imagine.

1. **Scopul aplicației**

Ușurarea și eficientizarea învățării prin intermediul flashcard-urilor, temporizatorului Pomodoro și teste de tip quiz.

1. ANALIZA PIEȚEI

**Cleverly** se încadrează în categoria aplicațiilor utilitare pentru educație. Pe piață există aplicații precum Quizlet, dar necesită introducere manuală a informației și nu oferă integrare cu AI.

**Cleverly** oferă elemente inovatoare menite să eficientizeze procesul de învățare și înțelegerea informației precum:

* generare automată a unor teste de tip quiz și flashcard-uri inteligente;
* interfață ușor de folosit;
* temporizator Pomodoro integrat;

Proiectul a fost dezvoltat în mai multe etape:  
1. Analiza nevoii utilizatorului;  
2. Implementare backend (Node.js);  
3. Implementare frontend (HTML, CSS, JAVASCRIPT);  
4. Testare pe mai multe fișiere PDF, imagini și texte;

**O imagine care conține text, captură de ecran, Font, număr

Conținutul generat de inteligența artificială poate fi incorect.**

Imagine 1

1. IMPLEMENTAREA APLICAȚIEI
2. **Proiectarea arhitecturală**

Aplicația este structurată pe două părți principale:

* Frontend: Am utilizat HTML, CSS și JavaScript pentru o interfață stilizată, ușor de accesat și flexibilă.
* Backend: Am utilizat Node.js împreună cu Express pentru a construi un server rapid și scalabil. De asemenea, am integrat API-ul Google Gemini 2.0 Flash AI pentru a beneficia de capabilități avansate de inteligență artificială. Am utilizat această tehnologie deoarece oferă un timp de răspuns redus și un nivel ridicat de acuratețe în procesarea cererilor utilizatorilor, ceea ce a îmbunătățit considerabil experiența generală în aplicație.

1. **Tehnologii folosite**

* Node.js, Express – server și API handling;
* Google Gemini 2.0 flash API – integrare AI;
* HTML, CSS, JS – interfață intuitivă.

1. **Stabilitatea aplicației**

Aplicația funcționează eficient cu fișiere PDF mari și cu texte lungi. Pe viitor dorim să implementăm o modalitate pentru aplicație să înțeleagă și scrisul de mână. Nu există pierderi de memorie, fișierele fiind șterse după procesare.

1. **Securitatea aplicației**

* Fișierele sunt validate prin extensie și dimensiune;
* Nu sunt salvate date personale.

1. **Testarea produsului**

Testele au fost realizate pe:

* Diferite fișiere PDF;
* Imagini clare și neclare cu text tipărit;
* Texte aleatoare de pe website-uri precum Wikipedia.

Funcționalitățile frontend au fost testate pe mai multe browsere și dispozitive, iar conținutul flashcard-urilor și testelor de tip quiz a fost de asemenea verificat.

1. **Folosirea unui sistem de versionare**

Aplicația este versionată utilizând sistemul de control al versiunilor Git. Codul este organizat într-o structură clară, împărțită pe directoare specifice (frontend, backend, utils, etc.), ceea ce facilitează gestionarea și înțelegerea proiectului.

Am utilizat ramuri (branches) separate pentru dezvoltarea de funcționalități noi, remedieri de erori și testare. Astfel, modificările au fost integrate treptat și controlat în ramura principală (main), reducând riscul apariției erorilor majore.

O imagine care conține text, captură de ecran, software, afișaj

Conținutul generat de inteligența artificială poate fi incorect.

Imagine 2

De asemenea, folosirea Git a permis:

* 1. colaborarea eficientă în echipă;
  2. urmărirea clară a istoricului modificărilor;
  3. revenirea rapidă la versiuni stabile în caz de necesitate;

Această abordare a contribuit la un proces de dezvoltare mai sigur, mai organizat și mai ușor de întreținut.

🔗 Link GitHub: <https://error404girls.github.io/Cleverly/>

1. ECHIPA

**Responsabilitățile au fost împărțite astfel:**Dezvoltarea aplicației a fost împărțită eficient între membrii echipei, fiecare contribuind cu expertiza proprie pe segmentele principale ale proiectului:

Ana Natalia Loga s-a ocupat de partea de frontend, realizând designul interfeței și implementând componentele vizuale folosind HTML, CSS și JavaScript. Tot ea a dezvoltat și mecanismul de temporizare din aplicație. Designul a fost gândit pentru a fi accesibil și estetic, asigurând o experiență de utilizare plăcută pe diferite dispozitive.

Anais Valentina Spafiu a fost responsabilă de partea de backend, construind arhitectura serverului cu ajutorul Node.js și Express. De asemenea, ea a integrat în aplicație Inteligența Artificială prin API-ul Google Gemini 2.0 Flash, aducând funcționalități avansate de procesare și generare a conținutului. Acest lucru a permis aplicației să răspundă în mod inteligent la solicitările utilizatorilor, asigurând performanță.

Această colaborare a permis o diviziune clară a sarcinilor și o dezvoltare coerentă, în care ambele componente – frontend și backend – au fost integrate armonios într-o aplicație complet funcțională.

1. CODUL SURSĂ

Codul sursă a fost scris în Visual Studio Code, după care toate fișierele au fost introduse în GitHub. La anumite părți din codul din fișierele JavaScript a fost folosit ChatGPT sau Claude pentru optimizare.

O imagine care conține text, captură de ecran

Conținutul generat de inteligența artificială poate fi incorect.

Imagine 3

O imagine care conține text, captură de ecran

Conținutul generat de inteligența artificială poate fi incorect.

Imagine 4

O imagine care conține text, captură de ecran

Conținutul generat de inteligența artificială poate fi incorect.

Imagine 5

O imagine care conține text, captură de ecran, software

Conținutul generat de inteligența artificială poate fi incorect.

Imagine 6

O imagine care conține captură de ecran, text, software

Conținutul generat de inteligența artificială poate fi incorect.

Imagine 7

|  |  |
| --- | --- |
| Imagine 1 | Generatorul de flashcard-uri |
| Imagine 2 | Organizarea fișierelor în GitHub |
| Imagine 3 | Porțiune cod JS pentru generarea testelor de tip quiz |
| Imagine 4 | Node.js pentru backend instalat prin terminal |
| Imagine 5 | Porțiune de cod privind integrarea Google Gemini 2.0 flash |
| Imagine 6 | Porțiune de cod JS pentru Pomodoro |
| Imagine 7 | Porțiune de cod JS pentru generarea flashcard-urilor |

**BIBLIOGRAFIE**

https://github.com/the-beast-coder/AI-Flashcards/blob/main/app.py

https://fontawesome.com/search?q=next&o=r&s=solid&ip=classic

https://www.w3schools.com/js/js\_async.asp

https://www.w3schools.com/howto/howto\_css\_next\_prev.asp

https://cdnjs.cloudflare.com/ajax/libs/pdf.js/3.4.120/pdf.min.js

https://fonts.google.com/selection/embed

https://cdnjs.cloudflare.com/ajax/libs/pdf.js/3.4.120/pdf.min.js

https://cdnjs.cloudflare.com/ajax/libs/pdf.js/3.4.120/pdf.worker.min.js

https://cdn.jsdelivr.net/npm/tesseract.js@4/dist/tesseract.min.js