

Lebanese University

Faculty of Engineering III

Done by:

Mohammad Karaki 6393

Mohammad Atwi 6328

Supervisor:

Dr. Mohammad Aoude

**Introduction:**PDF files have long established themselves as a standard format in both digital education and various professional environments, primarily because of their ability to retain consistent formatting and their platform-independent nature. Whether it's lecture notes, technical reports, official forms, or detailed research papers, PDFs are widely utilized to ensure that content appears the same across different devices and systems. However, while this consistency is beneficial, it can also present notable challenges—especially when users need to interact more deeply with the content. This becomes particularly problematic in cases where the PDF consists of scanned images or contains non-editable text. In such instances, tasks like copying text for translation, summarization, or simple note-taking become tedious, inefficient, and often frustrating.  
  
To address these limitations, our project introduces a comprehensive, web-based platform specifically designed to improve the overall usability of PDF documents. Users can upload virtually any PDF file and utilize a suite of integrated tools that enable a variety of functions. These include text extraction through Optical Character Recognition (OCR), machine translation using both traditional rule-based and advanced AI-driven methods, automated summarization, and even the generation of quizzes based on the document’s content. With these capabilities, the platform aims to significantly streamline tasks related to learning, information retrieval, and cross-language communication—especially within academic, multilingual, or professional settings.  
  
**Technical Stack:**  
On the frontend, we utilized Next.js, a powerful framework built on React. It allowed us to build a fast, interactive, and search-engine-friendly user interface. It efficiently manages routing, server-side rendering, and page generation while delivering a smooth user experience across various screen sizes and devices. For styling, we integrated Tailwind CSS, which enabled rapid and consistent design with utility-first classes. Additionally, we used Radix UI for accessible and customizable UI primitives that enhanced the interactivity and consistency of our components.  
  
The backend of our application is powered by Flask, a lightweight and flexible Python framework. It handles incoming API requests, manages interactions with the AI and NLP models, and connects various components of the platform together. Additionally, we used Flask-CORS to enable Cross-Origin Resource Sharing, ensuring seamless communication between the frontend and backend during both development and production.  
  
Our system incorporates a number of pre-trained AI models and Natural Language Processing tools to power its functionality. For translation, we integrated the Helsinki-NLP/opus-mt-en-fr model, which enables robust neural machine translation from English to French and is easily adaptable to other language pairs. Summarization is handled by distilbart, a distilled version of the BART transformer fine-tuned for text summarization, allowing us to condense large chunks of content effectively. For question answering, we utilized deepset’s roberta-base-squad2 model, which excels in extracting answers from provided passages. To generate new questions from existing text, we employed the potsawee/t5-base-question-generation model, which creates meaningful and relevant questions automatically.  
  
Supporting these models are essential libraries and tools like Hugging Face Transformers, which provides streamlined access to a wide variety of pre-trained models. We use PyTorch as the underlying deep learning framework for these models. EasyOCR and pytesseract are both integrated for OCR tasks, giving users the ability to extract text from scanned or image-based documents. PyMuPDF enables detailed PDF parsing and content extraction, while python-docx allows us to convert or export text to Word document format when needed.  
  
**Ethical Consideration:**  
  
Artificial Intelligence—particularly Natural Language Processing—presents a number of ethical challenges. These include concerns around bias, privacy, misuse, and reliability. NLP models can sometimes reflect harmful or unintended biases present in their training data or be misused to spread misinformation. Recognizing these risks, we have taken several steps to design our system responsibly.  
  
Our platform only processes documents explicitly uploaded by users and does not generate creative or open-ended content. It functions solely as a utility to assist users in processing and interpreting existing text. Importantly, we do not collect, store, or analyze any user data, which helps ensure a high standard of privacy. The AI models we use are open-source, widely vetted, and consistently produce dependable results within their intended scope. Since our toolset is limited to well-defined tasks—such as translating known text or summarizing documents—the potential for harmful or unethical use is minimized. Therefore, we believe the platform operates within a safe, fair, and responsible framework.  
  
**AI vs. Rule-Based Methods:**  
  
After conducting numerous tests and trials, it became apparent that AI-based solutions outperformed rule-based systems by a considerable margin in several key areas. This was especially true in complex scenarios like OCR for low-quality scans, contextual summarization, and quiz generation based on nuanced text. AI models demonstrated an ability to understand intent, structure, and context that traditional logic-based systems could not replicate.  
  
However, these advantages come at a cost. AI models—particularly large transformer architectures—require significant computational resources and can be slow to execute on lower-end devices such as standard laptops. Nevertheless, when deployed on high-performance machines or cloud infrastructure, these delays are substantially reduced, allowing AI to shine where accuracy and sophistication are prioritized. In scenarios where computing resources are limited, rule-based methods can still serve as effective backups, especially for basic or well-scoped tasks.  
  
**Rule-Based Logic:**  
  
Despite the capabilities of AI, we developed several rule-based components to serve as functional alternatives or fallback mechanisms. For OCR, we implemented Tesseract-OCR, a rule-based engine developed by Google. While not as accurate as AI-powered OCR, it performs reasonably well on clean and well-formatted text.  
  
For summarization, our rule-based approach retains the first and last sentences of each paragraph—based on the assumption that these are typically the most contextually important—and selects a few additional sentences from the body depending on relevance and length. This provides a condensed version of the original content while preserving key information.  
  
In terms of quiz generation, our rule-based method is quite basic, primarily due to time limitations. It converts declarative sentences into questions by flipping the subject and verb order, serving as a proof of concept for more advanced future development.  
  
For translation, we created a simple dictionary containing 2,000 English-to-French word pairs. During processing, each word in the input is checked against this dictionary, and if a match is found, it is substituted directly. While limited in scope, this rule-based approach offers quick results for predictable and controlled input.  
  
**Future Work and Conclusion:**  
  
Looking ahead, we see many opportunities for improving and expanding the platform. One key enhancement would be enabling users to apply summarization and quiz generation after translating a document. This would streamline the user workflow when working with foreign-language PDFs and allow for more dynamic interactions across language barriers.  
  
Although we started with English-to-French translation as a prototype, the system is flexible and can be extended to support additional languages with minimal changes. This would open the door to broader international use and make the tool more inclusive. On the rule-based side, refining our current implementations could yield faster and more efficient performance for specialized cases. A promising direction may involve hybrid models that combine the adaptability of AI with the speed and simplicity of rule-based logic.  
  
In conclusion, this project has been a rewarding and educational experience. It allowed us to explore the entire development lifecycle—from creating a frontend in Next.js to building and deploying a backend in Flask, and integrating sophisticated AI tools to handle real-world content. We gained hands-on experience in project planning, task delegation, and problem-solving in a team setting. The technical and conceptual insights from this work lay a strong foundation for future projects in the growing field of intelligent web applications.