

# Text Summarization Application

---

## Project Overview

This is a Text Summarization application built using **Streamlit** and **Sumy**, designed to summarize both text input and PDF files. The application provides users with four different summarization methods to choose from, allowing flexibility and comparison between different techniques.

## Project Objectives

The primary objective of this project is to automatically generate concise summaries from long texts or PDF documents, enabling users to grasp the essential points efficiently. The application offers multiple summarization methods, each employing different NLP algorithms.

## Data Input

The application supports two main input methods:

1. **Manual Text Entry:** Users can directly paste the text they want to summarize.
2. **PDF File Upload:** Users can upload PDF files, and the application will extract the text automatically.

## Summarization Methods

The application provides four popular text summarization techniques:

1. **LSA (Latent Semantic Analysis):** Identifies important sentences based on latent topics in the text.
2. **Luhn:** Focuses on sentences with a high frequency of significant words.
3. **LexRank:** Uses a graph-based approach to rank sentences by their importance.
4. **TextRank:** Similar to LexRank but differs in scoring based on sentence relationships.

## Tools & Libraries

The project uses the following libraries and tools:

- **Streamlit:** For creating an interactive web application.
- **Sumy:** A library for extracting automatic text summaries.
- **PyMuPDF (fitz):** To extract text from PDF files.
- **Python:** The primary programming language used.

## Installation

To run the application locally, make sure you have Python installed and then install the required libraries:

## How to Run

Start the Streamlit application using the command: The application will open in your default web browser.

## How to Use

1. Run the application.
2. Enter text manually or upload a PDF file.
3. Select a summarization method from the drop-down menu.
4. Choose the number of sentences for the summary.
5. Click on "🔗 **Summarize Text**" to generate the summary.
6. Download the summarized text using the "📄 **Download Summary**" button.

## Output & Features

- Displays text statistics before and after summarization (words, characters, sentences).
- Shows the summarized text with a success message.
- Provides a download option for the summarized output.
- Interactive and responsive UI with gradient background and animated buttons.

## Example Usage

- Summarizing long articles or reports.
- Extracting key points from research papers or academic texts.
- Condensing lengthy PDF documents into concise summaries.

## Challenges & Solutions

Challenge:

- Handling PDFs with complex formatting.
- Extracting meaningful summaries from large texts.

Solution:

- Used **PyMuPDF (fitz)** for accurate text extraction.
- Implemented multiple summarization techniques to give users flexibility in choosing the most suitable one.

## Future Improvements

- Adding support for more summarization algorithms like BERT and GPT-based models.
- Enabling multi-language summarization.
- Incorporating visualization of text statistics (word clouds, graphs).

## Contributors

This project was developed by:

- **Ziad Hamada**
- **Omar Hassan** We worked together to build a user-friendly and efficient text summarization tool using NLP techniques.