# Explain My Report - Code Walkthrough

This document explains each line of code used in the 'Explain My Report' project. The application uses Gradio, PyPDF2, and Hugging Face Transformers to create a simple web interface that takes a PDF medical report as input and generates a simplified, patient-friendly explanation.

|  |  |
| --- | --- |
| Code Section | Explanation |
| import gradio as gr | Imports the Gradio library, which is used to create an interactive web interface for the AI model. |
| from PyPDF2 import PdfReader | Imports PdfReader, which allows extraction of text content from PDF files. |
| from transformers import AutoTokenizer, AutoModelForCausalLM | Imports Hugging Face's AutoTokenizer and AutoModelForCausalLM. These are used to load the model and tokenizer for text generation. |
| model\_name = 'microsoft/phi-2' | Specifies the pretrained small language model (SLM) to be used for generating simplified text. |
| tokenizer = AutoTokenizer.from\_pretrained(model\_name) | Downloads and initializes the tokenizer for the chosen model, converting text into tokens. |
| model = AutoModelForCausalLM.from\_pretrained(model\_name) | Downloads the model weights and prepares the model for inference (text generation). |
| def simplify\_report(pdf\_file): | Defines a function that processes the uploaded PDF file and returns a simplified report. |
| reader = PdfReader(pdf\_file.name) | Opens and reads the uploaded PDF file using PyPDF2. |
| text = ''.join([page.extract\_text() for page in reader.pages]) | Extracts text from all pages of the PDF and combines them into a single string. |
| prompt = f'Explain this medical report in simple, patient-friendly language:\n\n{text}\n\nSimplified Explanation:' | Creates a clear instruction (prompt) for the model, telling it to simplify the extracted report text. |
| inputs = tokenizer(prompt, return\_tensors='pt') | Converts the text prompt into numerical tokens the model can understand. |
| outputs = model.generate(\*\*inputs, max\_new\_tokens=250) | Uses the model to generate up to 250 new tokens (words) based on the prompt. |
| result = tokenizer.decode(outputs[0], skip\_special\_tokens=True) | Decodes the model's output tokens back into human-readable text. |
| return result | Returns the final simplified explanation text to display in the app. |
| demo = gr.Interface(...) | Creates the Gradio interface. It defines inputs (file upload), outputs (text box), and app layout. |
| demo.launch() | Launches the Gradio app locally, opening a web interface at http://127.0.0.1:7860. |