```python
import streamlit as st
from PyPDF2 import PdfReader
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Function to extract text from PDF
def extract_text_from_pdf(file):
    pdf = PdfReader(file)
    text = ""
    for page in pdf.pages:
        text += page.extract_text()
    return text

# Function to rank resumes based on job description
def rank_resumes(job_description, resumes):
    # Combine job description with resumes
    documents [job_description] + resumes
    vectorizer = TfidfVectorizer().fit_transform(documents)
    vectors vectorizer.toarray()

    # Calculate cosine similarity
    job_description_vector = vectors[0]
    resume_vectors = vectors [1:]
    cosine_similarities = cosine_similarity ([job_description_vector],
resume_vectors).flatten()

    return cosine_similarities

# Streamlit app
st.title("AI Resume Screening & Candidate Ranking System")
# Job description input
st.header("Job Description")
job_description= st.text_area ("Enter the job description")

# File uploader
st.header("Upload Resumes")
uploaded_files = st.file_uploader("Upload PDF files", type=["pdf"],
accept_multiple_files=True)

if uploaded_files and job_description:
    st.header("Ranking Resumes")

    resumes = []
    for file in uploaded_files:
        text = extract_text_from_pdf(file)
        resumes.append(text)

    # Rank resumes
    scores = rank_resumes(job_description, resumes)
    # Display scores
    results = pd.DataFrame({"Resume": [file.name for file in uploaded_files], "Score": score })
    results = results.sort_values(by="Score", ascending=False)

    st.write(results)
```