Chapter 1: Introduction

This project aims to develop a resume ranking system using Natural Language Processing (NLP). It extracts text from resumes and ranks them based on similarity to a given job description.

Chapter 2: Project Overview

2.1 Objective

The objective of this project is to automate the process of resume screening by matching resumes against a job description using NLP techniques.

2.2 Scope

The system takes multiple resumes as input, extracts their text, and calculates a similarity score with respect to the job description. The resumes are then ranked accordingly.

Chapter 3: Technology Stack

The following technologies were used in this project:

- Python
- spaCy (NLP processing)
- pdfminer (PDF text extraction)
- Pandas (Data handling)

Chapter 4: Implementation

4.1 System Workflow

The system follows these steps:

1. Load the job description.
2. Extract text from resumes.
3. Compute similarity scores.
4. Rank resumes based on scores.
5. Save results to a CSV file.

4.2 Code Structure

4.2.1 Extracting Text from PDF

Text extraction from resumes is done using the pdfminer library.

4.2.2 Calculating Similarity Score

The similarity between resumes and job descriptions is calculated using the spaCy NLP model.

4.2.3 Ranking Resumes

The extracted text is compared with the job description, and resumes are ranked based on similarity scores.

Chapter 5: Code Explanation

5.1 Enhanced Python Code

The following is the improved version of the resume ranking system:

```
import os
import pandas as pd
import spacy
from pdfminer.high_level import extract_text

# Load an optimized NLP model
nlp = spacy.load("en_core_web_lg")

def extract_text_from_pdf(pdf_path):
    try:
        return extract_text(pdf_path)
    except Exception as e:
        print(f"Error extracting text from {pdf_path}: {e}")
        return ""

def calculate_match_score(resume_text, job_doc):
    resume_doc = nlp(resume_text)
    return resume_doc.similarity(job_doc) if resume_text.strip() else 0

def process_resumes(resume_folder, job_description):
    scores = []
    job_doc = nlp(job_description)

    for filename in os.listdir(resume_folder):
        if filename.endswith(".pdf"):
```

```
        file_path = os.path.join(resume_folder, filename)
        text = extract_text_from_pdf(file_path)
        score = calculate_match_score(text, job_doc)
        scores.append((filename, score))

    ranked_resumes = sorted(scores, key=lambda x: x[1], reverse=True)
    results_df = pd.DataFrame(ranked_resumes, columns=["Resume", "Score"])

    results_df.to_csv("ranked_resumes.csv", index=False)
    return results_df

if __name__ == "__main__":
    folder_path = "resumes"
    job_desc = "We are looking for a Python developer with experience in machine learning and NLP."

    results = process_resumes(folder_path, job_desc)
    print(results)
```

Chapter 6: Results and Analysis

After running the system, resumes are ranked based on their similarity scores. The results are saved in 'ranked_resumes.csv', which can be reviewed for further analysis.

Chapter 7: Conclusion

This project successfully automates resume screening using NLP. The system efficiently extracts text, computes similarity scores, and ranks resumes accordingly, making the hiring process more efficient.