# Project Report: Music Recommendation System

## Introduction

This project focuses on building a Music Recommendation System using Python. The system suggests songs similar to a user-selected song based on the text content of the song's metadata. The application utilizes Flask for web development, NLP techniques for text processing, and machine learning for generating recommendations.

## Methodology

The implementation process began with preparing and cleaning the dataset. This involved the following steps:

1. Data Preparation:\*\* The `songdata.csv` file containing song metadata was loaded and a random sample of 5000 rows was taken. Unnecessary columns such as 'link' were dropped, and the text was preprocessed to remove special characters and convert it to lowercase.

2. Tokenization and Stemming: Using the `nltk` library, text data was tokenized and stemmed to reduce words to their base forms.

3. Feature Extraction: A Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer was used to transform the text data into numerical features. Cosine similarity was calculated to determine the closeness between song texts.

4. Model Serialization: The processed data and similarity matrix were saved as pickle files (`df.pkl` and `similarity.pkl`) for later use.

5. Web Application Development: A Flask application was built to provide a user interface. The web app allowed users to select a song and view recommendations.

## Code Summary

The project code includes the following main components:

- Data Processing: Reading and cleaning the dataset, applying NLP techniques, and calculating cosine similarity.

- Recommendation Function: A function that takes a song title as input and returns a list of 20 similar songs.

- Flask Web App: The app includes two routes—`/` for the home page and `/recom` for displaying recommendations. HTML templates with Bootstrap styling were used for the front end.

-Pickle Serialization: The processed data and similarity matrix were saved as pickle files to speed up the recommendation process.

## Conclusion

This project demonstrates the integration of machine learning and web development to build a functional Music Recommendation System. It highlights the importance of preprocessing text data, using cosine similarity for recommendations, and implementing a user-friendly interface with Flask. Future improvements could include incorporating user feedback and expanding the dataset for more diverse recommendations.