SENIOR DESIGN PROJECT END-TERM PRESENTATION

RHYTHM COMPOSER : MUSIC GENERATOR

Supervised By: Dr. Suprava Devi

Group No.#:L15
Saurav Kumar – 2141011106
A Durga Madhab Patro – 2141016231
Raushan Raj – 2141018023
Dipanshu Kumar Mahato - 2141013040

Department of Computer Sc. and Engineering
Faculty of Engineering & Technology (ITER)
Siksha 'O' Anusandhan (Deemed to be) University
Bhubaneswar, Odisha

Presentation Outline

Introduction

Project Overview and Problem Statement

- Rhythm Composer is an AI-powered music generation system that combines advanced natural language processing and deep learning to democratize music creation.
- This technique first generates lyrics using transformer models, then converts them into music with audio generation algorithms.
- Rhythm Composer enables non-musicians to create music while assisting professionals with inspiration and prototyping.
- Rhythm Composer requires a text prompt (music theme), optional style parameters (genre, mood, tempo), and a duration setting for generating music.

❑ Problem Statement

- **Incoherent Structure** – AI struggles with logical musical progression
- **Lack of Emotion** – Generated music feels robotic and lacks depth.
- **Repetitive Patterns** – AI often produces monotonous, predictable tunes.
- **No Artistic Intent** – Lacks cultural and personal expression.

Objectives & Motivations

- Objectives:
  - **Democratize Music Creation** – Enable users with no musical background to create compositions using AI.
  - **Enhance Musical Expression Through AI** – Ensure AI-generated music retains human-like emotional depth and structure.

Motivations

- To improve the structure and emotional depth of AI-generated music using transformer-based models.
- To reduce repetition and make music generation more dynamic and creative.
- To enable personal and cultural expression through AI-assisted composition.
- To bridge the gap between AI and the creative arts, making technology a tool for artistic expression.
- To provide an accessible, low-cost alternative to traditional music production for both beginners and professionals.

Background & Related Work

Literature Survey

Topic	Authors	Published	Purpose	Problem Addressed	Benefits	Limitations	Model Used
Transformer architecture [1]	Vaswani, et al.	June 2017	Introduced self-attention mechanisms to revolutionize sequence-to-sequence tasks, making it ideal for language and music generation.	Addresses challenges in handling long-range dependencies in sequences, improving efficiency in text and music processing.	Significantly enhances language modeling and music generation by capturing complex dependencies over long sequences	High computational and memory requirements for long sequences; lacks inductive bias for locality (unlike RNNs/CNNs), which can be inefficient for tasks with local structure.	Transformer (original architecture with self-attention mechanism)
Music Transformer: Addressing long-term structure in music generation [2]	Huang, et al.	January 2018	Highlighted the importance of capturing long-term musical patterns in generative models.	Resolved the challenge of maintaining coherence in AI-generated music over extended sequences.	Produced more structured, human-like compositions, ensuring musical coherence over long durations.	Requires large amounts of training data; struggles with capturing expressive nuances like dynamics or emotion in music beyond structure.	Music Transformer (an adaptation of Transformer architecture optimized for music generation)
GPT family of models for text generation [3]	Radford, et al.	June 2018	Demonstrated that large-scale language models can produce coherent and creative text, providing the foundation for our lyric generation	Tackled the challenge of generating human-like text with deep contextual understanding.	Enabled high-quality, contextually aware lyric generation, improving creative writing automation.	Can produce factually incorrect or biased outputs; lacks fine-grained controllability over generated content and coherence over very long passages.	GPT (Generative Pretrained Transformer) series, including GPT-1, GPT-2, and GPT-3

Literature Survey

Topic	Authors	Publishe d	Purpose	Problem Addressed	Benefits	Limitations	Model Used
Jukebox: End-to-end music generation with vocals [4]	Dhariwal, et al.	April 2020	Explored neural networks for raw audio music generation, including singing voices. .	Tackled the complexity of end-to-end music creation, including harmonization and vocal synthesis	Opened possibilities for AI-generated music with human-like vocals, expanding the scope of AI composition	High computational requirements and long inference times; limited control over specific lyrical or musical structure, often generating incoherent vocals.	Jukebox (Transformer + VQ-VAE)
Advances in text-to-audio models like AudioLM and suno-ai/bark [5]	**AudioLM Authors:** Borsos, et al. **Suno AI's Bark Authors:** Shulman, et al.	2022	Explored high-quality audio synthesis from text descriptions.	Addressed limitations in generating high-fidelity audio content from textual inputs.	Enabled realistic sound synthesis, improving applications in AI-generated music and speech.	Models like Bark can produce inconsistent prosody or unnatural transitions; AudioLM's reliance on pretrained models may limit generalization across languages or genres.	AudioLM (Audio Tokenizer + Language Model) Bark (GPT-style Transformer**)**
MusicLM: High-quality music generation from text descriptions [6]	Agostinelli, et al.	January 2023	Demonstrated text-to-music transformation, validating AI's ability to compose music based on textual prompts.	Addressed the gap in conditioned music generation, improving coherence and structure.	.Enhanced accessibility and usability of AI-powered music composition for artists and creators.	Struggles with fine control over musical elements like melody or rhythm; potential copyright concerns with generated content.	MusicLM (w2v-BERT + SoundStream + MuLan)

Literature Survey

Topic	Authors	Published	Purpose	Problem Addressed	Benefits	Limitations	Model Used
MusicGen: Simple and Controllable Music Generation [7]	Jean, et al.	June 2023	Developed a transformer-based model capable of generating music from text prompts and optional melody inputs..	Existing systems lacked flexibility in incorporating melodic guidance or structured control over output.	Allowed users to guide music generation using both text and melody, improving structure and musicality.	Limited by the model's training dataset diversity; struggles with generating culturally nuanced or genre-blending compositions.	MusicGen (Transformer Decoder + EnCodec)
MusiCoder: Controllable Symbolic Music Generation with Codebook Tokens[8]	Tian, et al.	October 2023	Introduced a controllable framework using codebook tokens to generate symbolic music with user-defined attributes..	Existing models often lack user control over rhythm, key, and melody, leading to generic results..	Enabled users to define specific musical attributes (e.g., tempo, scale), improving control and personalization.	Focuses on symbolic music, not raw audio; lacks realism and requires additional conversion steps for audio playback.	MusiCoder (VQ-VAE + Codebook Transformer)
Noise2Music: Text-to-Music Generation with Latent Diffusion[9]	Jang, et al.	February 2024	Proposed a latent diffusion model for generating high-quality music directly from text prompts.	Traditional text-to-music methods lack audio fidelity and semantic alignment between text and sound.	. Achieved better alignment between the prompt and musical output, offering greater fidelity and genre control.	Requires large training data and struggles with long-form composition and complex emotional tones.	Noise2Music (Latent Diffusion Model)

Improvements Over Existing Solutions

❑ Uniqueness of the work

- ➢ End-to-end pipeline from text prompt to complete musical composition.
- ➢ Integration of separate lyric and music generation phases.
- ➢ The Open-source approach enabling customization and extension It can help musician as well as the non-musician .
- ➢ It ensures consistency in the quality of the musical outputs. This is particularly useful when multiple artists are collaborating on the same project as it ensures that everyone has access to the same creative foundation.
- ➢ This technique is highly scalable, allowing users to generate music compositions of various lengths and complexities. This is particularly useful in time-sensitive projects where manually composing music could take a significant amount of time.

Proposed Solution & Architecture

Schematic Layout

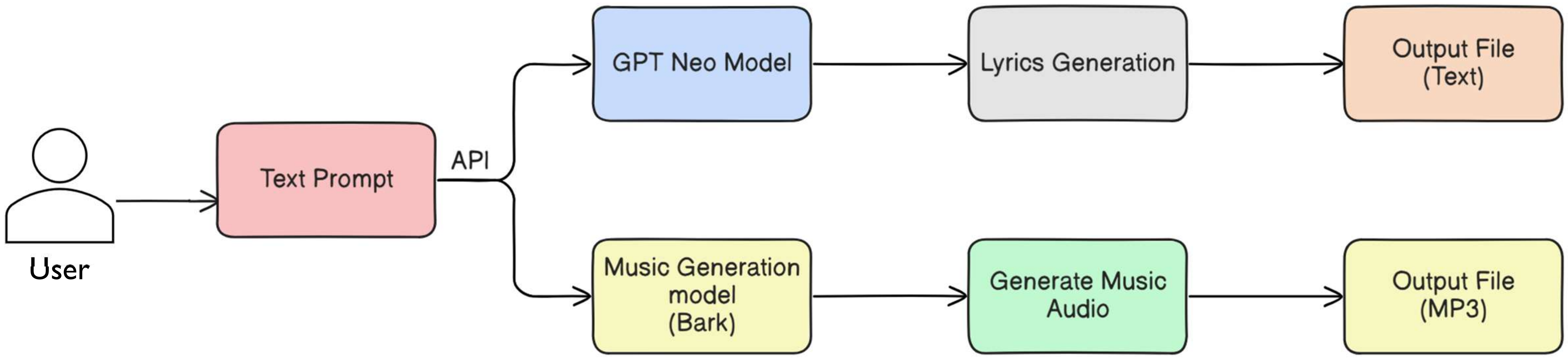


Figure 1: Schematic Layout for the Rhythm Composer

Proposed Solution & Architecture

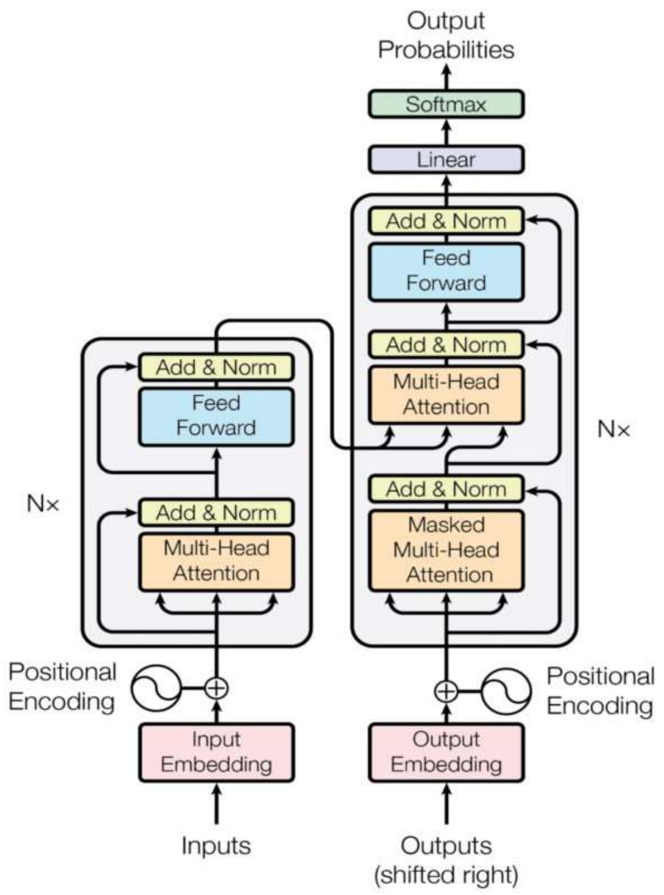


Figure 2: Architecture of Transformer Model

Proposed Solution & Architecture

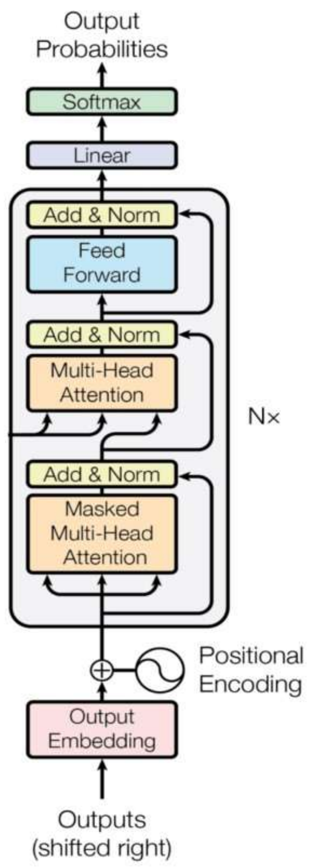


Figure 3: Architecture of GPT Model

Proposed Solution & Architecture

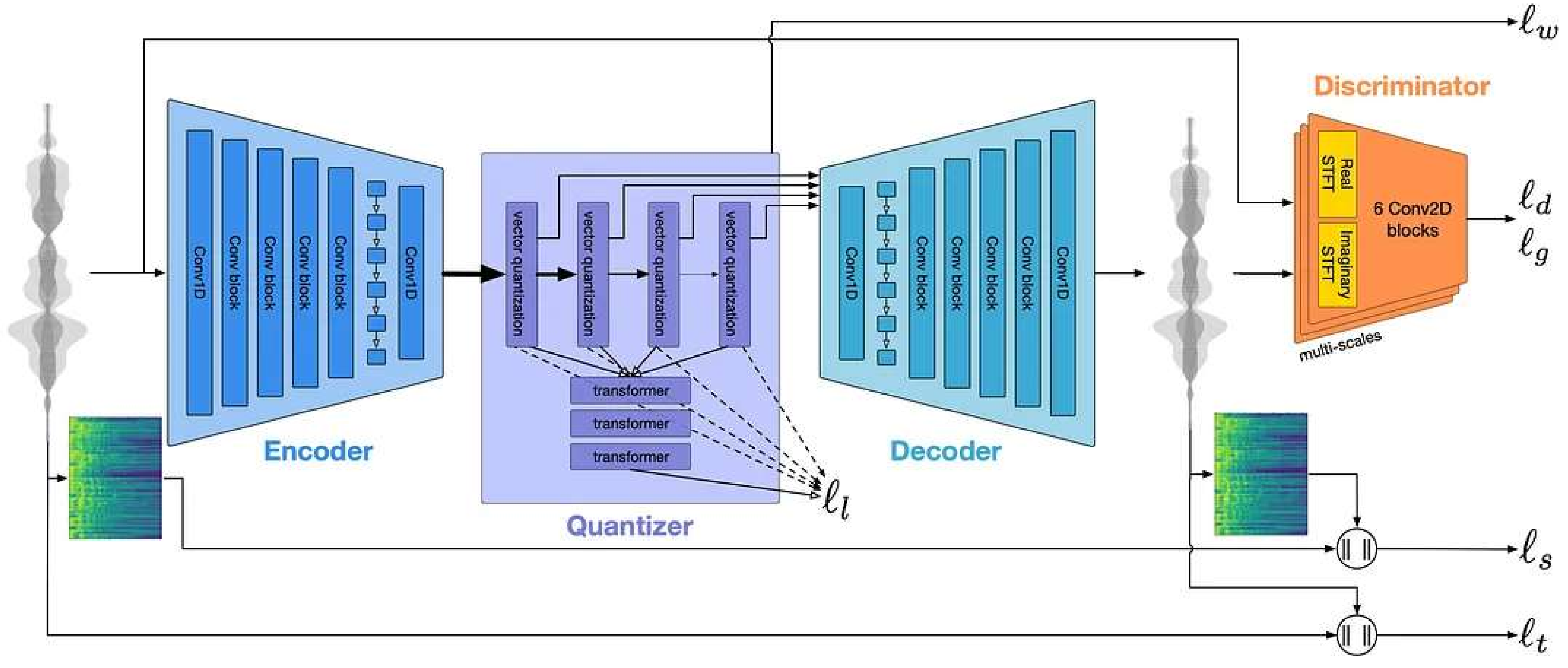


Figure 4: Architecture of Bark Model(Encoder-Decoder)

Key Components & Modules

- **Lyric Generation Model** – Uses **GPT-Neo** to generate structured lyrics from user text prompts.
- **Music Generation Model** – Converts lyrics into musical compositions using **deep learning audio synthesis**.
- **Transformer-Based Architecture** – Utilizes **self-attention mechanisms** to enhance context understanding.
- **ML-Driven Music Structuring** – Aligns lyrics with **verse-chorus patterns, melody progression, and rhythm**.
- **Music Generation Model** – Uses **Bark** Model to generate the music audio from user text prompts.

Methods and Algorithms used

- **Python**: A programming language for AI and web development.
- **GPT-Neo**: A text-generation AI model.
- **Transformers Architecture**: Enhances text processing with self-attention.
- **FastAPI**: A fast web framework for APIs.
- **Text-to-Music Generation**: Converts text prompts into lyrics and music.
- **Music Structuring**: Aligns lyrics with musical patterns.
- **Composition Approaches**: Uses structured song elements.
- **Neural Architecture**: Ensures AI-generated music follows composition rules.

Results and Analysis

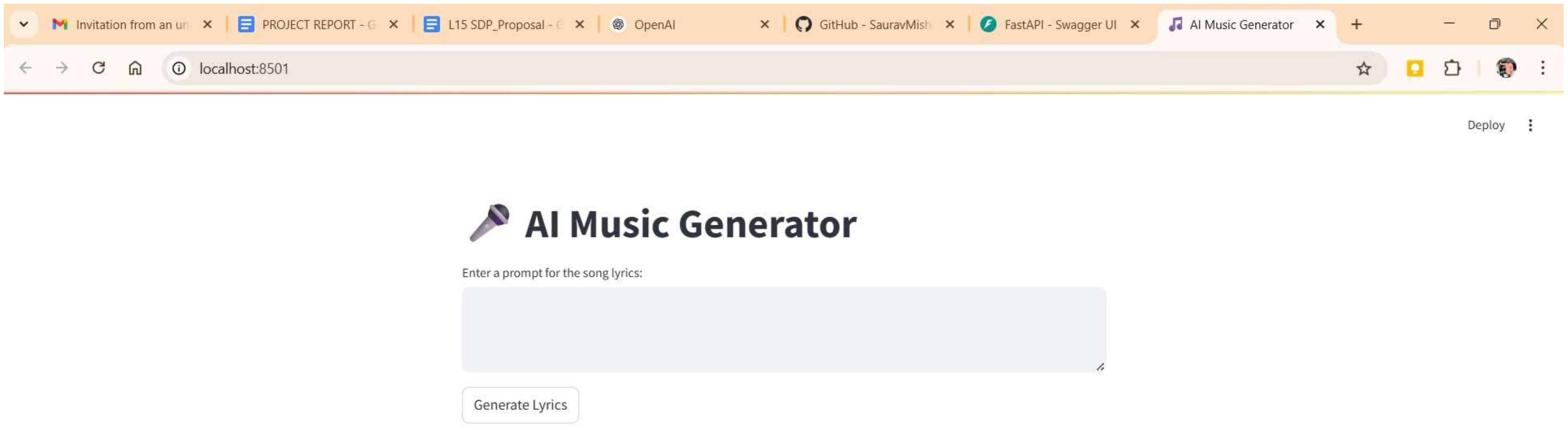
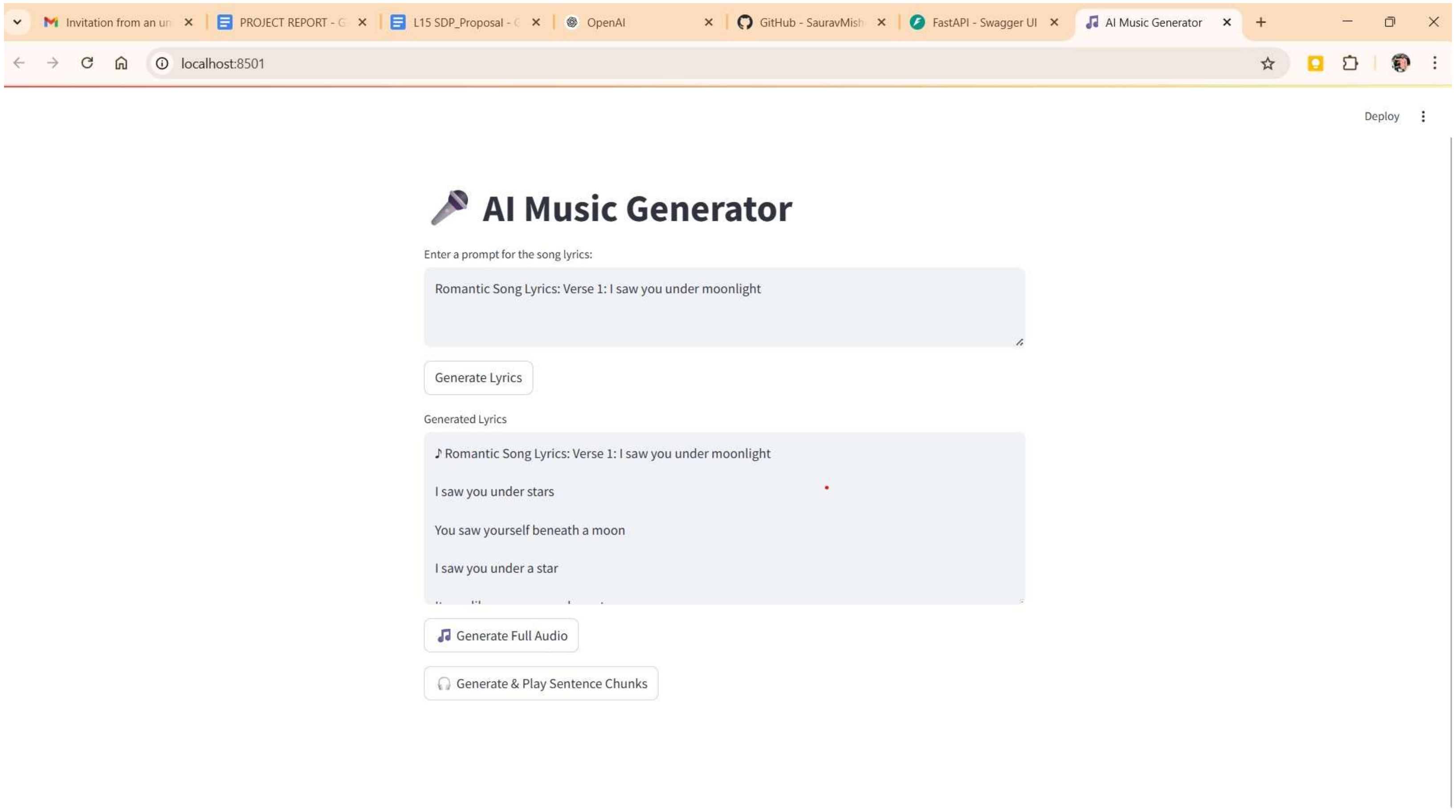


Figure 5: User Interface of the AI Music Generator

Results and Analysis



🎤 AI Music Generator

Enter a prompt for the song lyrics:

Romantic Song Lyrics: Verse 1: I saw you under moonlight

Generate Lyrics

Generated Lyrics

♪ Romantic Song Lyrics: Verse 1: I saw you under moonlight

I saw you under stars

You saw yourself beneath a moon

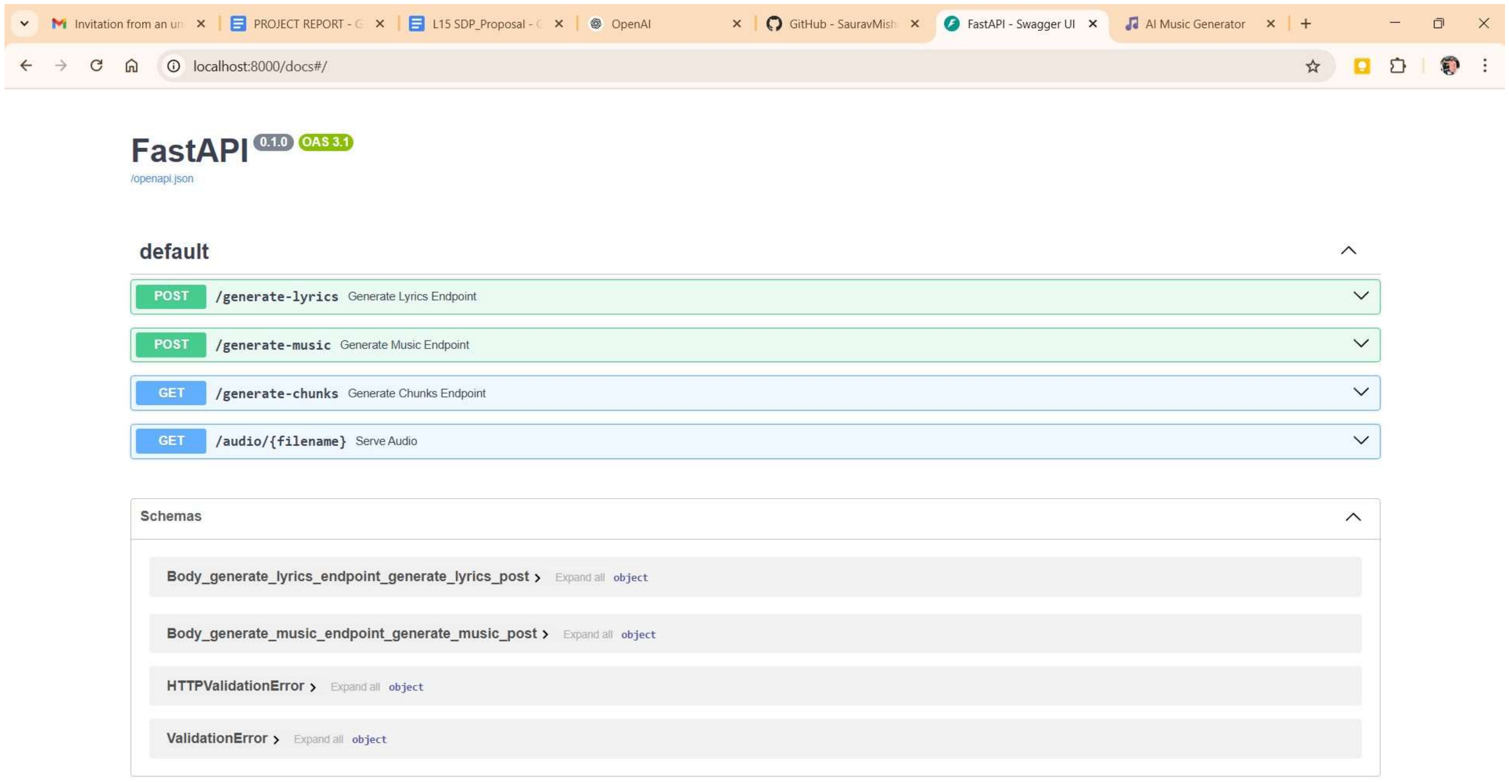
I saw you under a star

Generate Full Audio

Generate & Play Sentence Chunks

Figure 6 : User Interface of the AI Music Generator with prompt

Results and Analysis



Invitation from an un
PROJECT REPORT - G
L15 SDP_Proposal - C
OpenAI
GitHub - SauravMish
FastAPI - Swagger UI
AI Music Generator

localhost:8000/docs#/

FastAPI 0.1.0 OAS 3.1
/openapi.json

default

POST `/generate-lyrics` Generate Lyrics Endpoint

POST `/generate-music` Generate Music Endpoint

GET `/generate-chunks` Generate Chunks Endpoint

GET `/audio/{filename}` Serve Audio

Schemas

Body_generate_lyrics_endpoint_generate_lyrics_post › Expand all object

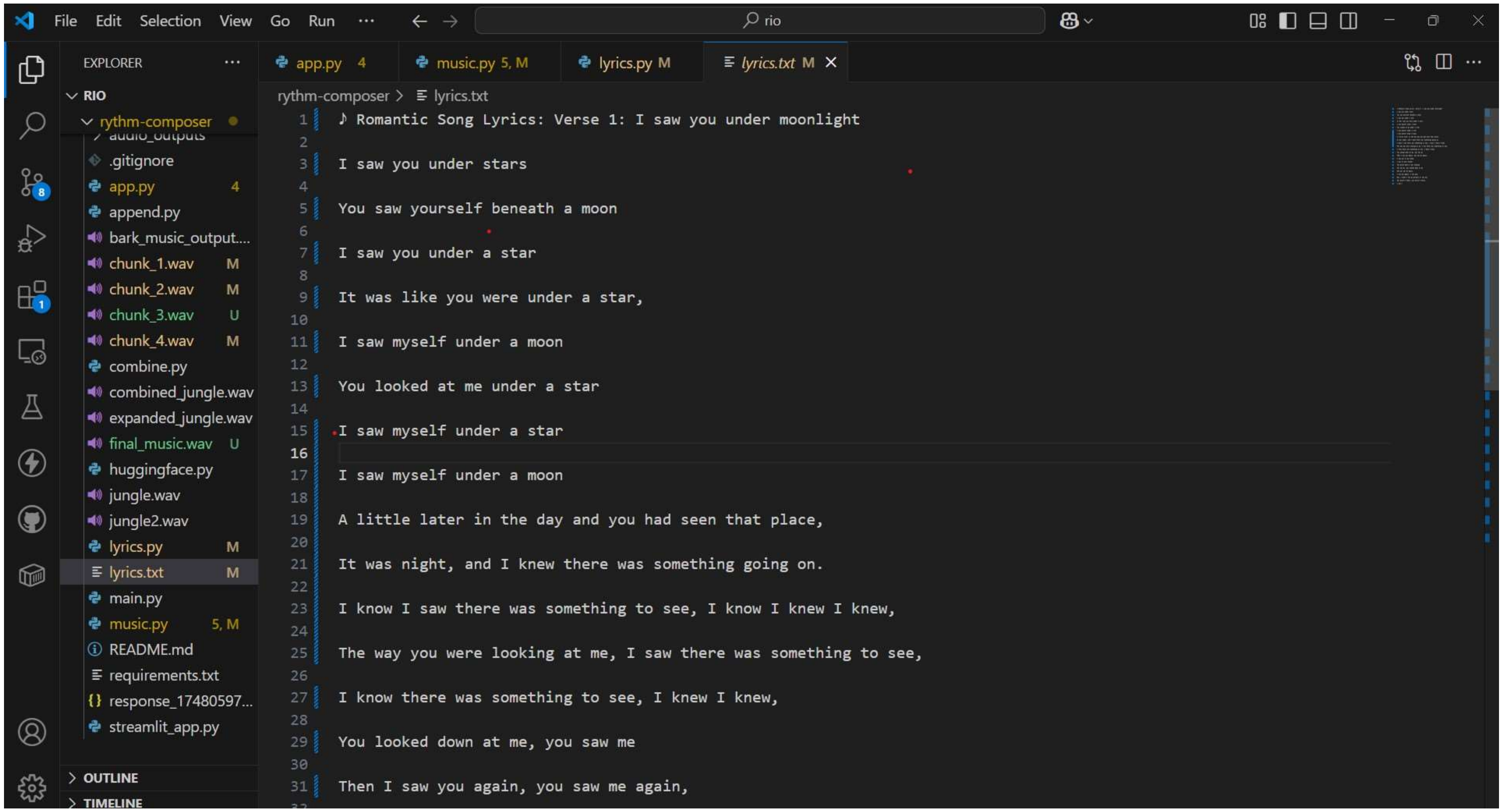
Body_generate_music_endpoint_generate_music_post › Expand all object

HTTPValidationError › Expand all object

ValidationError › Expand all object

Figure 7: API End Points for Rhythm Composer

Results and Analysis



```
♪ Romantic Song Lyrics: Verse 1: I saw you under moonlight

I saw you under stars

You saw yourself beneath a moon

I saw you under a star

It was like you were under a star,

I saw myself under a moon

You looked at me under a star

I saw myself under a star

I saw myself under a moon

A little later in the day and you had seen that place,

It was night, and I knew there was something going on.

I know I saw there was something to see, I know I knew I knew,

The way you were looking at me, I saw there was something to see,

I know there was something to see, I knew I knew,

You looked down at me, you saw me

Then I saw you again, you saw me again,
```

Figure 8 : Output of Lyrics

Results and Analysis

```
Microsoft Windows [Version 10.0.19045.5737]
(c) Microsoft Corporation. All rights reserved.

(env) G:\rythm-composer>
```

Figure 9: Generated Music Audio

Comparison with Existing Models

Feature / Model	Bark (Suno AI)	Jukebox (OpenAI)	AudioLM (Google)	MusicLM (Google)	MusicGen (Meta)	Noise2Music (NAVER)
Architecture	GPT-style Transformer	VQ-VAE + Transformer	Audio tokenizer + Language model	w2v-BERT + MuLan + SoundStream	Transformer + EnCoder	Latent Diffusion Model
Vocal Capability	Yes (speech synthesis, expressive voices)	Yes (singing vocals)	Limited (not intended for vocals)	No (instrumentals only)	No	No
Text-to-Audio Quality	High (speech/music)	Medium (some artifacts)	Very high	Very high	High	Very high
Control & Customization	Low–Medium (limited prompt control)	Low (genre/lyrics based)	Medium (prompt tuning)	Medium (text semantics)	High (text + musical control)	High (genre, style tags)
Training Data	Proprietary audio datasets	1.2M songs	Large-scale audio-text pairs	Large audio-caption pairs	Meta's internal music dataset	Noisily paired text-music datasets

Comparison with Existing Models

	Bark (Suno AI)	Jukebox (OpenAI)	AudioLM (Google)	MusicLM (Google)	MusicGen (Meta)	Noise2Music (NAVER)
Realism of Output	High (specially speech)	High (vocals, raw music)	Very high	Very high	High	Very high
Use Case Fit for Music	Medium (not optimized for full tracks)	High (end-to-end music)	Medium	High	High	High
Latency / Speed	Fast (real-time)	Very Slow (hours per song)	Moderate	Moderate	Fast	Moderate
Open Source Availability	Yes (via Hugging Face)	No	No	No	Yes	Yes
Main Limitation	Less control over music; better for speech	Long generation time, coherence issues	Limited melody structure	No singing or vocals	Limited long-form coherence	Needs paired data, lacks emotion modeling

Conclusion

❑ Key Findings:

- Rhythm Composer showcases how AI can enhance music creation accessibility.
- It transforms simple text prompts into structured, emotionally rich compositions.
- Utilizes models like GPT-Neo for generating lyrics.
- Employs machine learning algorithms for audio generation.
- Enables non-musicians and professionals alike to create music quickly and creatively.
- Demonstrates AI's role as a tool for artistic expression.
- Bridges the gap between technology and creativity.

Conclusion

❑ **Potential Extension:**

- **Faster Output Generation**: Reduce track generation time for improved efficiency.
- **Multilingual Vocal Support**: Support multiple languages with correct pronunciation and emotion.
- **Improved Emotional Depth**: Enhance the music's emotional range and mood variation.
- **Interactive Features**: Enable real-time editing and user feedback.
- **Genre and Cultural Diversity**: Support a broader range of musical genres and cultures.
- **Enhanced Audio Quality**: Add realistic vocals and better instrument simulations.

Bibliography

[1] Transformer architecture introduced by Vaswani et al. (2017) Vaswani, Ashish, et al. "Attention is a
need." *Advances in neural information processing systems* 30 (2017).

[2] Music Transformer: Addressing long-term structure in music generation. Huang, C. Z. A., et al "Music transformer. arXiv preprint arXiv: 1809.04281." (2018).

[3] GPT family of models for text generation. Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

[4] Jukebox: End-to-end music generation with vocals Dhariwal, Prafulla, et al. "Jukebox: A generative model for music. " *arXiv preprint arXiv:2005.00341* (2020).

[5] Advances in text-to-audio models like AudioLM and suno-ai/bark Borsos, Zalán, et al. "Audiolm: a language modeling approach to audio generation."*IEEE/ACM transactions on audio, speech, and language processing* 31 (2023): 2523-2533.

[6] MusicLM: High-quality music generation from text descriptions. Agostinelli, Andrea, et al. "Musiclm: Generating music from text." *arXiv preprint arXiv:2301.11325* (2023).

Bibliography

[7] MusicGen: Simple and Controllable Music Generation Copet, Jean, et al. "Simple and controllable generation." arXiv preprint arXiv:2306.05284 (2023).

[8] MusicCraft: Structure-Aware Music Generation Ren, Yi, Jinzheng He, and Tao Qin. "MusicCraft: Structure-aware symbolic music generation." arXiv preprint arXiv:2309.10864 (2023).

[9] Hunyuan-Audio: Multi-level Language-Aligned Audio Generation Wu, Jiatong, et al. "Hunyuan-Audio: An audio foundation model for content creation with multiple levels of language-audio alignment." arXiv preprint arXiv:2310.02227 (2023).

Thank you