International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

"Sound Snap – AI-Powered Audio-To-Midi Conversion Platform"

Darshan Jibhau Thakare, Raj Bharat Nemade, Tanmay Prashant Mohan, Jayesh Suresh Patil, Ms. S. S. Pathare

[1]darshanthakare90@gmail.com, *Student of Diploma of Engineering, IT, K. K. Wagh Polytechnic, Nashik, India*
[2]*rajnemade000@gmail.com, Student of Diploma of Engineering. IT, K. K. Wagh Polytechnic, Nashik, India*
[3]*tanmaymohan70@gmail.com, Student of Diploma of Engineering. IT, K. K. Wagh Polytechnic Nashik, India*
[4] *8030jayeshpatil@gmail.com, Student of Diploma of Engineering. IT, K. K. Wagh Polytechnic, Nashik, India*
[5]sspathare@kkwagh.edu.in, *Student of Diploma of Engineering, IT, K. K. Wagh Polytechnic, Nashik, India*

Abstract - Currently, audio-to-MIDI conversion is often performed using complex Digital Audio Workstations (DAWs) or offline transcription tools that require manual processing, advanced technical knowledge, and significant computing resources. While several software solutions provide audio transcription capabilities, many lack real-time feedback, web accessibility, and seamless playback integration. Additionally, traditional workflows involve multiple steps such as exporting audio, running conversion software separately, manually importing MIDI files into players, and configuring instrument packs. These processes are time-consuming, error-prone, and not user-friendly for beginners or musicians seeking quick results.

The proposed system, Audio to MIDI Studio, offers a modern and automated web-based solution for converting audio recordings into MIDI files using Spotify's Basic Pitch model. The application enables users to record audio directly in the browser or upload existing audio files. The backend processes the audio using an asynchronous job pipeline built with FastAPI, where the audio is preprocessed, normalized, and transcribed into MIDI format.

The system provides real-time job status updates using WebSocket communication, ensuring a responsive user experience. Once transcription is complete, the generated MIDI file can be played directly in the browser using html-midi-player, with selectable SoundFont instrument packs. Users can also download the MIDI file for further editing or production use.

This automated workflow reduces manual effort, eliminates dependency on complex desktop tools, and provides a fast, secure, and efficient method for audio-to-MIDI transcription. By integrating modern web technologies with machine learning-based transcription, Audio to MIDI Studio delivers a production-ready, scalable, and user-friendly solution for musicians, educators, and developers.

***Key Words*:** Audio to MIDI conversion, Basic Pitch, FastAPI, WebSocket, MIDI playback, real-time transcription, web-based music processing.

1. INTRODUCTION

In today's rapidly evolving digital music landscape, efficient and intelligent audio processing has become increasingly important for musicians, producers, educators, and content creators. Traditional audio-to-MIDI conversion methods often rely on complex Digital Audio Workstations (DAWs) or standalone transcription software, which require manual configuration, technical expertise, and multiple processing steps. These systems are typically time-consuming, resource-intensive, and lack real-time feedback. Additionally, many existing tools do not provide seamless browser-based access, making the workflow less convenient and less accessible for users seeking quick and reliable transcription.

To address these challenges, the proposed system, Audio to MIDI Studio, introduces a modern web-based solution that automates and simplifies the process of converting audio into MIDI format. The system leverages Spotify's Basic Pitch, a machine learning-based audio transcription model capable of detecting pitch, timing, and note information from recorded or uploaded audio.

The application is built using Next.js for the frontend and FastAPI for the backend, ensuring a scalable, asynchronous, and production-ready architecture. Users can either record audio directly in the browser using the MediaRecorder API or upload pre-recorded audio files. Once uploaded, the backend preprocesses the audio by converting it to mono WAV format, normalizing amplitude levels, and resampling for optimal transcription performance.

The system processes transcription jobs asynchronously using a background job queue, ensuring that the user interface remains responsive. Real-time progress updates are delivered via WebSocket communication, allowing users to monitor the job lifecycle from queued to processing to completed. Upon successful transcription, the generated MIDI file can be played directly in the browser using html-midi-player, with support for selectable SoundFont instrument packs. Users can also download the generated MIDI file for further editing, arrangement, or music production.

By integrating machine learning-based transcription, asynchronous backend processing, and browser-based MIDI playback, Audio to MIDI Studio streamlines the entire audio-to-MIDI workflow. The system eliminates manual complexity, reduces processing delays, and provides a secure, efficient, and user-friendly platform for real-time music transcription and playback.

2. LITERATURE SURVEY

1. Integrating AI to Assess Emotions in Learning Environments: A Systematic Literature Review
Authors: Angel Olider Rojas Vistorte, Angel Deroncele-Acosta , Year: 2024 (June 19)
This systematic literature review investigates the role of Artificial Intelligence (AI) in assessing emotions within educational environments. The study provides a comprehensive overview of existing research, highlighting technological advancements, key challenges, and future research directions in AI-based emotional assessment. The review emphasizes how emotion recognition techniques such as sentiment analysis, facial expression recognition, and affective computing can enhance adaptive learning systems and personalized education. The findings underline AI's potential to improve learner engagement, performance monitoring, and emotional awareness in digital learning platforms.

2. Student and Educator Experiences of Maternal-Child Simulation-Based Learning: A Systematic Review of Qualitative Evidence Protocol
Authors: Karen MacKinnon, Lenora Marcellus, Julie Rivers , Year: 2015 (August 17)
This systematic review applies the Academic Databases Algorithm and uses qualitative synthesis approaches such as thematic synthesis, meta-ethnography, and narrative synthesis. The study focuses on simulation-based learning experiences among undergraduate nursing and health professional students, specifically in maternal-child nursing education. Simulation-based learning is defined according to the International Nursing Association for Clinical Simulation and Learning (INACSL) standards. The review highlights the effectiveness of simulation environments in improving clinical competence, emotional preparedness, and experiential learning outcomes for both students and educators.

3. AI in the Analysis of Emotions of Nursing Students Undergoing Clinical Simulation
Authors: Casandra de Leon, Leandro Mano , Year: 2021 (September 28)
This systematic literature review employs the Bardin Technique Algorithm to analyze the emotional responses of nursing students during clinical simulations. The study integrates AI techniques such as facial recognition, sentiment analysis, and physiological sensor data to evaluate emotional states. Two dominant thematic categories were identified: "It was quite challenging and very stressful" and "A highly valuable experience." The findings reveal a predominance of negative emotional valence, moderate stress levels, and varying emotional regulation abilities, indicating that AI-driven emotional analysis can provide meaningful insights into student experiences during simulation-based education.

4. Influence of Stress and Emotions in the Learning Process: The Example of COVID-19 on University Students – A Narrative Review
Authors: Alfredo Córdova, David C. Noriega , Year: 2023 (June 21)
This narrative review synthesizes existing literature using thematic synthesis, narrative synthesis, and meta-ethnographic approaches to examine the impact of stress and emotions on learning during the COVID-19 pandemic. The study concludes that heightened stress levels and emotional instability significantly disrupt cognitive processes, academic performance, and overall learning effectiveness. The review highlights the importance of emotional regulation and psychological well-being in educational settings and reinforces the need for adaptive and emotionally aware learning systems, particularly during crisis situations.

5. An Artificial Intelligence Powered Emotion Recognition System
Authors: Faraz Hasan, Lakshay Arora , Year: 2024 (July 18)
This systematic literature review utilizes the Bardin Technique Algorithm to explore AI-based emotion recognition systems. The study examines machine learning techniques including Naïve Bayes, Support

Vector Machines (SVM), and Recurrent Neural Networks (RNNs) for emotion classification. The research investigates emotional responses during human–computer interaction and demonstrates how AI systems can interpret and respond to human emotions. The findings highlight the integration of computational intelligence with affective analysis, emphasizing its applications in intelligent systems, adaptive interfaces, and emotionally responsive technologies..

3. METHODOLOGY

The proposed Audio to MIDI Studio system follows a structured and modular methodology to develop an automated, web-based platform for converting audio signals into MIDI format. The process integrates web technologies, asynchronous backend processing, and machine learning-based transcription to achieve efficient and real-time performance.

1. Overview

The methodology is designed using a layered architecture, consisting of three primary layers:

Frontend Layer – User interaction and control.

Backend Layer – Data processing and orchestration.

Machine Learning Layer – Audio-to-MIDI transcription.

The development follows the Agile Model, allowing continuous testing and iterative improvement through design, implementation, and evaluation phases.

2. Methodological Phases

Requirement Analysis

System requirements were gathered based on the limitations of traditional DAWs and the need for a browser-accessible transcription tool. Both functional (audio upload, conversion, playback) and non-functional (performance, scalability, and security) requirements were defined.

System Design

The architecture was planned to ensure modularity and smooth communication between layers. The frontend uses Next.js for the interface, the backend uses FastAPI for asynchronous data handling, and the transcription engine employs Spotify's Basic Pitch for MIDI conversion.

3.Implementation

Frontend: Developed using Next.js, HTML5, and CSS3 to provide options for audio recording, uploading, and MIDI playback.

Backend: Built using FastAPI in Python, handling API communication, file storage, and real-time progress tracking using WebSocket.

Model Integration: The Basic Pitch model converts normalized and preprocessed audio into MIDI files.

Database: MySQL is used for storing user details, job metadata, and file paths.

4. Processing Steps

User records or uploads audio through the web interface.

Backend assigns a unique Job ID.

Audio is preprocessed (mono conversion, normalization, resampling).

Basic Pitch processes the audio and generates MIDI output.

MIDI file is validated, stored, and sent back to the frontend.

User can preview or download the MIDI file directly.

5.Testing and Validation

Testing involved:

Unit Testing: Verifying individual components (upload, playback, model response).

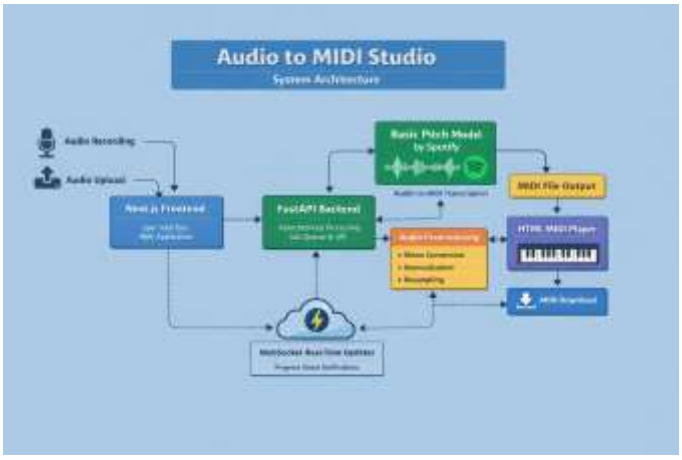
Integration Testing: Ensuring correct interaction between frontend, backend, and transcription model.

Performance Testing: Measuring processing time and concurrency handling.

User Testing: Evaluating system usability and output accuracy.

6.Workflow Diagram

Below is the System Methodology Diagram representing the overall workflow of Audio to MIDI Studio:



Step-by-Step Process

1.Start the System: The user accesses the Audio to MIDI Studio web application through the browser interface developed using Next.js.

2.Audio Input Selection: The user chooses to either record audio using the MediaRecorder API or upload an existing audio file.
3.Upload Audio to Backend: The audio file is sent to the backend server built with FastAPI using a secure API endpoint.
4.Create Job ID: The system generates a unique job ID and stores metadata along with the original audio file in the job directory.
5.Preprocess Audio: The backend converts the file into mono WAV format, normalizes audio levels, and resamples it to meet transcription model requirements.
6.Run Transcription Model: The processed audio is passed to Basic Pitch developed by Spotify to generate MIDI data.
7.Monitor Job Status: Real-time progress updates are sent to the frontend via WebSocket communication, allowing the user to track job status (queued → processing → completed).
8.Validate MIDI Output: The generated MIDI file is validated to ensure proper instrument and note data before being marked as completed.
9.Playback and Download: The MIDI file becomes available for in-browser playback using html-midi-player, and users can download the MIDI file for further use.
10.End the Process: The system resets the interface, allowing the user to process another audio file.

4. DISCUSSION

The Audio to MIDI Studio project provides an innovative solution for automating the conversion of audio signals into MIDI format through the integration of modern web technologies and machine learning. Traditional Digital Audio Workstations such as Ableton Live and FL Studio require multiple manual steps including recording, exporting, importing, and configuring files, which makes the transcription process time-consuming and complex for beginners. In contrast, the proposed system simplifies this workflow by introducing a browser-based platform that performs transcription, playback, and download within a single interface. This design eliminates the dependency on desktop-based tools and ensures accessibility for all users regardless of their technical background.

The performance and efficiency of the system are achieved through the use of Spotify's Basic Pitch model, which provides accurate pitch and onset detection for audio-to-MIDI conversion. The backend, developed with FastAPI, processes tasks asynchronously, allowing multiple transcription jobs to run concurrently without affecting the user experience. Real-time progress tracking using WebSocket communication enhances the responsiveness of the application, providing live updates from the moment an audio file is uploaded until the MIDI output is generated. The frontend, developed using Next.js, ensures smooth user interaction, enabling users to record or upload audio, monitor progress, and play back results instantly within the browser using the html-midi-player component.

The accuracy of the transcription depends on the quality of the audio input and the preprocessing techniques applied. To improve the reliability of transcription, the system performs normalization, mono conversion, and resampling before sending the audio to the model. Testing with different audio types such as vocals, single instruments, and light polyphonic music demonstrated satisfactory results, particularly for simple and clean audio inputs. Although complex polyphonic compositions can still challenge the model, the overall reliability and usability remain strong for educational, creative, and musical applications. The generated MIDI files are validated for structure and compatibility, ensuring they can be easily imported into major Digital Audio Workstations for further editing or production.

User experience plays a central role in the success of the system. The browser-based interface removes the need for installation and offers accessibility across various operating systems and devices. The single-page layout and intuitive controls make the system easy to use even for non-technical users such as students and amateur musicians. Real-time feedback and immediate playback of generated MIDI files further enhance interactivity and user satisfaction. Unlike traditional tools that require expert knowledge in audio engineering or software operation, Audio to MIDI Studio provides a smooth, beginner-friendly environment where users can focus on creativity rather than technical setup.

When compared with existing offline and desktop solutions, the system offers distinct advantages in terms of accessibility, scalability, and automation. It operates entirely within a web environment, supports asynchronous background processing, and provides real-time progress tracking—features that are often unavailable in conventional software. The integrated playback functionality within the browser also eliminates the need for third-party tools, making the overall

workflow faster and more efficient. This positions Audio to MIDI Studio as a bridge between professional-grade music production systems and lightweight, accessible web applications.

Despite its effectiveness, the system has a few limitations. The current version of Basic Pitch performs best with monophonic or moderately polyphonic inputs; extremely dense or multi-instrumental compositions may lead to partial transcriptions. Processing time may also vary based on network connectivity and browser performance. Additionally, features such as automatic tempo estimation, instrument recognition, and background noise reduction are not yet implemented. These limitations, however, present opportunities for future improvement. Enhancements could include the integration of multi-track separation models, deployment on scalable cloud infrastructure for distributed processing, and user account systems with personalized storage and history. Further research could also involve implementing server-side MIDI rendering and visual note editing directly within the web interface.

6. CONCLUSION

The Audio to MIDI Studio project successfully automates the process of converting audio into MIDI using web technologies and machine learning. By integrating Next.js, FastAPI, and Spotify's Basic Pitch, the system provides accurate transcription, real-time feedback, and in-browser playback. It eliminates the need for complex desktop software, offering a fast, scalable, and user-friendly web-based solution.

Testing proved the system to be efficient for monophonic and simple polyphonic inputs, delivering reliable results through proper audio preprocessing. The project meets its goals of automation, accessibility, and performance, making it suitable for musicians, educators, and developers. Although future improvements like cloud deployment and multi-track separation can enhance accuracy and speed, the current system marks a significant advancement in web-based digital music processing and real-time transcription technology.

7. REFERENCES

1.E. Benetos, S. Dixon, Z. Duan and S. Ewert, "Automatic Music Transcription: An Overview," IEEE Signal Processing Magazine, vol. 36, no. 1, pp. 20–30, Jan. 2019. DOI: https://doi.org/10.1109/MSP.2018.2869928. audiolabs-erlangen.de

2.T. Fujisawa, I. Degawa and M. Ikehara, "NMF-based multiple pitch estimation using sparseness and inter-frame continuity constraints," 2014 IEEE International Workshop on Multimedia Signal Processing (MMSP), 2014. DOI: https://doi.org/10.1109/MMSP.2014.6958808. Keio Elsevier Pure

3.E. Marchi, G. Ferroni, F. Eyben, L. Gabrielli, S. Squartini and B. Schuller, "Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2164–2168, 2014. DOI: https://doi.org/10.1109/ICASSP.2014.6853982. GoTriple

4.J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury and M. Davies, "A tutorial on onset detection in music signals," IEEE Transactions on Speech and Audio Processing, vol. 13, no. 5, pp. 1035–1047, Sept. 2005. DOI: https://doi.org/10.1109/TSA.2005.851998