

Binauralize Your Beats

Mir Jeffres, Tucker Alexander, Kai White, Duncan Hanson

Binaural Beats

- Two frequencies close together played in different ears create BEAT FREQUENCIES
- Brain Entrainment
- Sleep Induction
- Memory Improvement
- Anxiety Treatment
- Stress Relief
- Mood Improvement
- Lessen dependency on pain medication



healthline

brainwave **STATES**

DELTA

0.1 Hz | 4 Hz

Deep sleep | Pain relief | Anti-aging | Healing

THETA

4 Hz | 8 Hz

REM sleep | Deep relaxation | Meditation | Creativity

ALPHA

8 Hz | 13 Hz

Relaxed focus | Stress reduction | Positive thinking | Fast learning

BETA

13 Hz | 30 Hz

Focused attention | Cognitive thinking | Problem solving | Active

GAMMA

30 Hz & above

High-level cognition | Memory recall | Peak awareness

healthline

Goal Statement



To create a **website** that can generate two 1-30 Hz spaced frequencies in each ear following the bassline of an existing song in order to create a binaural beating sensation. This is for the purpose of **reducing stress and anxiety, inducing sleep, or improving mood or long-term memory** based on the user's **desired mental health goals**. The website will also **teach users about how binaural beats work**.

Audience: Anyone who likes to listen to music and wants to modify their current mental state while doing so. The website will make their uploaded music into a tool to focus, sleep, relax, or otherwise change their brain activity with the power of binaural beats.

Currently there are no web apps that generate binaural beats that are easy for users to understand without prior knowledge of what the different brain wave bands do, nor are there web apps that turn your music into binaural beat music. If a consumer wanted this feature in their music they would have to do it themselves in a DAW.

Design Overview

- Required Hardware Specifications:
 - Web application minimizes the hardware requirements a user will need
 - Python requirements outlined in requirements.txt
- Frameworks and other Dependencies:
 - Flask based app
 - Opens in a local Python server
 - Demucs source separation library
 - SciPy pitch shifting and filtering



Back End

- Python and Flask
- Takes in the separated bass stem as well as the full song, frequency shifts the bass in one ear, and mixes the audio in stereo
 - Frequency shifting function uses a Hilbert transform, which imparts the same phase shift for all frequencies. This helps preserve the quality of sound in our output
 - Low pass filter cleans up the extracted stem, further preserving audio quality

Original Audio + Original Bass



Original Audio + Shifted Bass



Front End

- HTML, Javascript, and CSS
- Three pages:
 - Landing: five buttons to choose the frequency band and one button to learn more
 - Upload: dropdown menu that communicates with backend, upload audio files to be binauralized and play/download the output file
 - Education: additional resources and explanations of research/use of binaural beats
- Navigation bar allows user to intuitively navigate to other pages from any page
- Communicates with backend via XHR requests from the dropdown menu

Limitations

- Currently hosted locally, so users need many Python libraries, which are outlined in requirements.txt
- The audio upload can only receive .mp3 and .wav files
- Slow audio processing

Live Demo of the Progress we Made!

On Mir's computer:

<http://143.215.93.120:8100/>

Reminder of Measure of Success

- The website is easy to navigate and the content is explained well enough that anyone with an interest in the topic could understand
- The sound processing is not distracting to the user while creating the intended beating effect
- The site can pull music from a user-uploaded file and implement the desired binauralization
 - Bassline is pitch shifted in one or both ears to create a binaural beat sensation for the listener
 - Beat frequency depends on the desired brain effect selected on the landing page

Where To Go From Here

- Create a URL for the web app (Heroku)
 - All files are hosted through Flask right now
- Improve wait times through refactoring and looking further into Demucs documentation
 - Potentially add an estimated wait time indicator or an accurately scaled loading bar
- Volume selector
- Connect landing page selectors to the dropdown on upload page
- Connecting to Spotify/YouTube for easier music accessibility
- Marketing for the application to draw in a user-base
 - Social media based marketing can be extremely effective these days
- Collect feedback from users to gather ideas for further improvement