Industry Project: Final Report

Affective State Change via Procedurally Generated Haptics

by Vincent **Göke** | Moritz **Sendner** | JM **Santiago** III

in collaboration with Daniel **Shor** from **Innovobot Labs Inc.**

Research Library (Zotero): https://www.zotero.org/groups/5177784/industry_project/library

GitHub Project Page (MVP code): https://github.com/NesR0M/Industry_Project/tree/main

🤖 Live Page (MVP): https://react-midi.netlify.app/

Project Summary

⚠️ **2 Pages**, focusing on the **Application context, Problem statement and motivation, Goal of the project,** and **Project plan and process steps**

Application Context

For our Industry Project, we explored various directions, including artificial intelligence (AI), to procedurally generate long-form audio, additionally focusing on their musical patterns. This audio is intended for playback through a haptic actuator, aimed at inducing a state of relaxation leading to a state of flow. Following initial research and conceptualization, we developed a suitable prototype for producing this long-form audio. With this project, we are contributing to the field of emotional well-being applications aimed at users who want to improve their emotional state during concentrated productivity activities or for later relaxation.

Problem Statement and Motivation

In our fast-paced modern world, many people struggle with stress, lack of concentration, and emotional imbalance. This project addresses the need for a solution that can conveniently and effectively help users achieve their desired emotional states, such as relaxation and flow during or before the start of a productive work session. The motivation behind this project is to utilize generative AI and haptics to create an accessible emotional enhancement tool that meets the various needs of users in their daily lives and provides valuable insights into the field of emotional state change technology development.

Goal of the Project

Our main goal was to investigate the emotional states of flow and relaxation, design a concept for inducing these states, and construct a basic proof-of-concept prototype to evaluate our tool's effectiveness. Additionally, we aimed to explore potential embodiments for this technology. This prototype aims to leverage AI-generated long-form audio, played through a haptic speaker, to induce various affective states. The goal of this application is to improve users' emotional well-being, especially during study or work sessions, by offering personalized and guided audio-based haptic experiences that promote states such as relaxation and flow.

Initial Project Plan

Our initial approach was to only focus on flow:

1. Phase 1: Initial Research
    a. Desk Research:
        i. Understand the concept of flow.
        ii. Explore how AI can be utilized.
    b. Practical Research:
        i. Analyze existing music playlists and scientifically supported applications (like Endel) for soundscapes to identify essential characteristics for effective flow states.
        ii. Investigate existing generative AIs for audio creation and their limitations.
2. Phase 2: Concept Creation
    a. Based on the findings, create a concept (using only AI, partly AI and code, or no AI, only standard code).
    b. Design a haptic feedback system to deliver the customized waveforms to the user.
    c. Explore how this can be achieved in non-obtrusive ways for long periods.
    d. Explore possible embodiments like a neck pillow with built-in haptics, and what actuator with what resonance is working the best for this purpose.

3. Phase 3: Concept Testing
    a. Conduct qualitative user testing to assess perceived effectiveness.
4. Phase 4: Build MVP for our Industry Partner
    a. Integrate the AI-generated audio and haptic feedback system into a simple MVP mobile app using a front-end framework with parameters that can be adjusted in real time.
    b. The app should allow users to select a desired emotional state and then generate a personalized audio composition with haptic feedback.
5. Phase 5: Evaluation (If Time Permits)
    a. Conduct a second round of qualitative user testing to determine the effectiveness of the prototype.
    b. A sample comparative study may involve looking into the participant's productivity with or without the app
    c. The study will also collect feedback from participants on the app's overall experience.

Actual Project Plan

The project plan unfolded as follows:

1. Phase 1: Initial Research
    a. Desk Research:
        i. Research to understand the concept of flow.
        ii. Explored how AI can be utilized.
    b. Practical Research:
        i. We analyzed existing music playlists and scientifically supported applications (like Endel) for soundscapes to identify essential characteristics for effective flow states.
        ii. Investigated existing generative AIs for audio creation and their limitations.

2. Phase 2: Concept Creation
    a. We developed a concept for a flow-inducing soundwave that used fast-paced rhythmic beats (similar to a techno song) to induce a flow-like rhythmic state.
3. Phase 3: Concept Testing
    a. Based on our initial concept, we created a flow soundwave for testing with haptic actuators provided by our industry partner, using a very limited Python script (no GUI, no AI, hardcoded, no adjustments possible), also provided by our industry partner.
    b. Additionally, we tested different types of embodiments (oen neck and one back pillow and various placements of haptic actuators on the body).
    c. We conducted a qualitative user study with a simple breath-time-reducing relaxation soundwave provided by our industry partner and our flow-inducing soundwave. The relaxation soundwave worked well, but the flow soundwave was perceived as unpleasant. Multiple potential embodiment preferences were shared with us.
4. Phase 4: Second Research Phase
    a. Based on our user testing results, we conducted a second research phase and found that inducing flow alone is too difficult due to multiple internal and external triggers that we cannot control. Therefore, we reconceptualized our framework to use a well-received relaxation baseline pattern to induce relaxation and create a state of control, making it easier to induce a flow state.
5. Phase 5: New Concept Creation
    a. For our new relaxation-based framework, we developed a concept that included two components:
        i. A “baseline”, which is a code-created amplitude-modulated sine tone that gradually decreases in frequency and wavelength (tempo), resulting in a calming effect that gradually reduces the breathing rate from 11 to 6 breaths per minute.

    ii. “Sparkles,” which are AI-generated nuanced notes created based on the baseline to add texture and variety to the soundscape, preventing it from becoming monotonous and enhancing its richness and interest.

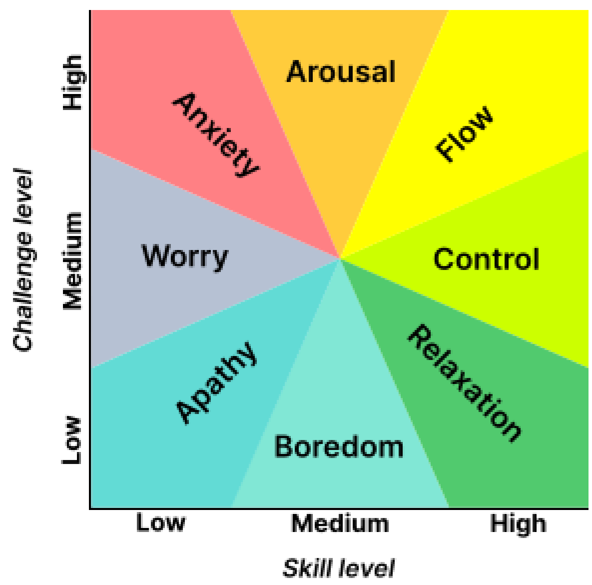
6. Phase 6: Build MVP for Our Industry Partner
    a. We developed a simple web-based MVP using the front-end framework React, with the potential for later adaptation to React Native for mobile app development. The “baseline” structure was generated using code, and the “sparkles” were generated using GPT-4. The soundwaves were generated using Tone.js. Additionally, we built an RNBO instrument designed for playback, which was not used in the React prototype anymore.

Project Outcome

Description of the Deliverable

Initial Concept and Evolution

Our project began with the goal of developing a mobile application aimed at general users in the work environment, focusing on improving emotional well-being through procedurally generated audio signals that are translated into haptic feedback via audio-haptic actuators. At the beginning of our project, our industry partner Daniel Shor from Innovobot gave us the choice of exploring relaxation, flow, or sleep. As relaxation is already a very well-known area, our team decided to explore the flow state instead, a concept from Mihály Csíkszentmihályi.



*https://en.wikipedia.org/wiki/File:Challenge_vs_skill.svg

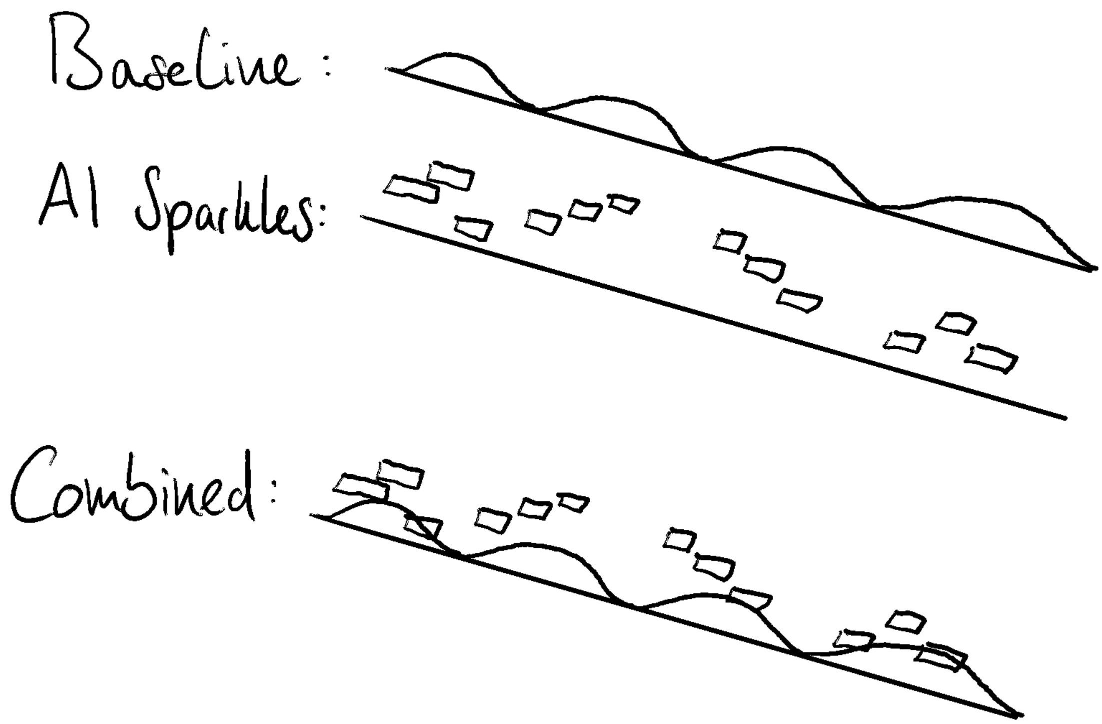
The conducted research during this phase was guided by our industry partner. After understanding the context and problem statement, we conducted our first preliminary pilot user tests with three different haptic actuators which also came from our industry partner. The goal of the user testing was to test out a created flow soundwave based on our research and to explore potential embodiments for a haptic device that would interface with the application. The knowledge gained from these tests, combined with research into AI models for audio generation, led to a significant shift from developing an application that triggers a flow state to one that triggers relaxation for a better flow state. For this purpose, we carried out a second research phase, as outlined in our project plan above.

Final Deliverable: Web Application

Aside from focusing on a relaxation tool we also changed the target group of our tool, we decided on developing a more specialized, testing tool, catering not just to general users but specifically to haptic designers and audio professionals. This was influenced by our research findings and early development insights. We built a web application-based MVP that allows haptic designers to create procedurally generated audio-based haptics, composed and controlled by MIDI data, to make the import and export of compositions more viable. This platform offers insight into possible future implementations of generative AI to create haptic experiences.

For this purpose, we had to update our original research plan, to incorporate:

1. **Desk Research**: Conducting thorough research, investigating the emotional states of relaxation to flow from the psychological perspective, to gain further insights on how these emotional states could be established more reliably. Additionally continuing research on current AI capabilities in audio and haptic generation, and studying the impact of these technologies on emotional states.
2. **Prototype Development**:
    - **Tech Stack**: Developing a React-based web application that incorporates MIDI creation and playback with Tone.js as an open-source version and in the end of the project with an alternative approach of cycling74's RNBO.json Max 8 instrument patches, that give options to create more versatile instruments, used to play the MIDI, resulting in these audio-based haptic experiences when played on the audio-haptic actuators.
    - **Sound Composition**: Creating the prototype with two key components, as suggested by the industry partner: a baseline and sparkles.
        - *Baseline*: Developing a modulated sine wave with amplitude modulation to mimic a calming breathing pattern, trying to lower the breath rate with a change of BPM and frequency pitch of the sine wave audio composition.
        - *Sparkles*: Integrating AI-generated notes using GPT-4 with few shotting to enhance the baseline and create a richer auditory experience by adding suitable harmonies, based on scientific frequency calculations and musical scale analysis.



3. **Deployment**: Launching the web app for web server application usage, allowing users to experience and provide feedback on the audio-haptic compositions.

Key Features

1. **Procedural Generation Capabilities**: Users can set specific characteristics and parameters to generate audio compositions. This is powered by a blend of AI algorithms and manual controls, providing both precision and creativity in design.
2. **Customizable Parameters**: Haptic designers have the flexibility to manipulate a range of parameters, including frequency, amplitude, and modulation settings, allowing for the creation of diverse and nuanced haptic feedback for instant testing.
3. **Real-Time Previews**: The application provides real-time rendering, enabling designers to hear and feel the haptic output as they adjust settings, ensuring the end product aligns with their creative vision.

4. **Integration of Tone.js and RNBO.js**: By utilizing advanced web audio technologies, the application provides robust and high-quality audio output, which is crucial for the precise development of haptic feedback. The RNBO instrument would be optimized for haptic actuators.
5. **AI-Enhanced Creativity**: By using GPT-4 LLM to generate MIDI compositions via a few shots to create the note structure, the application offers new and dynamic possibilities in haptic design.

Application Context and Usage

Our web application MVP offers a prototyping platform for professionals to swiftly generate relaxing haptic sine waves, as well as to explore, innovate, and enhance AI prompts for 'sparkles' feedback. It has potential applications across various industries, including gaming, therapy, and office environments, wherever there is a need for inducing relaxation.

Detailed Overview of Intermediary Results

The project commenced with an extensive phase of desk research, which revolved around the latest advancements in AI, audio, and haptic technology. Our initial idea revolved around establishing a guide for the concept of induced flow state, a highly focused and immersive mental state, through the use of specifically designed audio and haptic feedback. This concept was grounded in the hypothesis that certain auditory and tactile stimuli could effectively influence the user’s emotional and cognitive state, thereby facilitating easier entry into the flow state.

We initially aimed to address both relaxation and flow states. However, it became evident that encompassing both affective states within the project's scope and time constraints was overly ambitious. Therefore, we made a strategic decision to narrow our focus solely on relaxation. This refinement allowed for a more in-depth and targeted approach, ensuring the quality and feasibility of the project within the allocated timeframe.

Understanding Flow State

The first step was to delve deeply into the characteristics necessary to induce a flow state. This involved:

1. **Research on Flow Inducers**: The team conducted independent research, guided by the industry partner and haptic expert Daniel Shor, to understand the specific elements like Beats Per Minute (BPM), Breaths Per Minute (BrPM), haptic frequency sweet spots, audio-haptic actuator specifics, and other sonic characteristics that could potentially induce flow or at least potentially influence our cognitive processing. A big part of the flow state establishment is also the intrinsic motivation mindset of the user, which can have various variables influencing once's capabilities to engage into a flow state.
2. **Few-shotting an LLM**: The next step involved experimenting with a generative pre-trained transformer (GPT-4) model to generate usable music compositions in midi JSON form. This approach involved fine-tuning GPT's understanding of music theory. During these experiments, we encountered some limitations in the length of the response. Therefore, we also investigated ocal Large Language Models (LLM) as a promising alternative. However, this model proved to be impractical in the context of our project. It gave good and long responses but needed too much computing power to run as a simple application on the web or as an app.

Exploration of Haptic Development

Parallel to the audio component, we also looked into developing haptic feedback mechanisms, which included:

1. **SuperCollider Program Template**: A program could be designed in SuperCollider, an environment and programming language for real-time audio synthesis, where variables could be manipulated to create specific waveforms that would be translated into tactile feedback. However, we decided against using SuperCollider due to its complexity.
2. **Integration with Interhaptics Haptic Composer**: We also explored methods to integrate these waveforms into Interhaptics Haptic Composer, an industry-known tool that would allow the translation of audio signals into haptic feedback. After a short test phase, we decided against this tool as the existing app was only available on Android devices and was unfortunately very buggy.
3. **Implementation on Haptic Devices**: The final step in this segment was figuring out how to run these haptics on a physical haptic device, turning the

digital signals into actual tactile experiences.

Those approaches did not result in a fruitful manner, so we decided to first approach the audio-haptic actuators closer, to determine their characteristics and suitability four our cause.

Exploration of Audio Generation

At the beginning, we investigated various generative AI models and services. This exploration aimed to identify models that could seamlessly integrate with our existing Python script.

1. **Audiocraft + Gradio**
    - Utilized Meta's open-source music and audio-generative AI models.
    - Incorporated Gradio to run a web app, simplifying user interaction.
    - Found limitations in the variety of available models (primarily demo models), constraining the breadth of audio generation.
2. **Magenta's MelodyRNN**
    - Explored open-source melody-generative AI models using RNN-LSTM.
    - The model generates melodies in a step-by-step process, akin to a Large Language Model (LLM).
    - Availability limited to a Python library with no interactive playground or demo for user testing.
3. **Stability AI's Dance Diffusion**
    - Examined a collection of audio-generating machine learning models from HarmonAI under Stability AI.
    - Considered exploring more of HarmonAI's work.
    - Access to the model was restricted to demos on Google Colab, presenting challenges for direct integration.
4. **Non-AI Audio Creation**
    - As an alternative, a template was planned in MAXSP, a visual programming language for music and multimedia, to manually create audio waveforms.

Preliminary User Testing

As mentioned previously, we also conducted preliminary user testing to determine the effectiveness of our conducted desk research. For that, we tested two soundwaves one from our industry partner for relaxation and one flow state-inducing wavelength along with various haptic actuators and their potential embodiments. To make sure our first study on flow states was going the right way, this testing phase was very important. It helped us see what users liked and whether different designs of our devices worked well at work. We made two types of haptic devices (for the neck and back) and tested them in three situations.

1. **Prototypes**
    - **Neck Pillow**: Integrated with two wired haptic motors.
    - **Back Pillow**: Featured one wireless haptic motor.
2. **Testing Scenarios**
    - **First Prototype without Music**: Users experienced a 10-minute audio session (low intensity) and then took a 1-hour break. They maintained a diary for before and after the test.
    - **First Prototype with Music**: Similar to the first scenario, but with the inclusion of music.
    - **Second Prototype without Music:** A 10-minute audio session (high intensity) was followed by a 1-hour break, with diary entries before and after the test.
    - **Second Prototype with Music**: Similar to the first scenario, but with the inclusion of music.
3. **Final Interview**
    - After completing the testing scenarios, users participated in a final interview. This provided valuable insights into their experiences, preferences, and the overall impact of the haptic devices in conjunction with the audio application.

Shift in Target Audience and Platform

During the early stages of development, our team envisioned creating a mobile application aimed at general users. This concept was driven by the desire to make affective state changes accessible to a broad audience. However, as the project evolved, we encountered several challenges and opportunities that prompted a shift in our approach.

1. **Identifying a Niche Need**: Our research and early feedback highlighted a specific need within the haptic design community – a tool that allows for the creation of procedurally generated audio-based haptics. This realization steered our focus towards a more niche but impactful area.
2. **Recognizing the Individuality of Flow States:** Flow state establishment is a more complicated and not so easily inducible state. There are various factors like goal establishment, and mood similarities that help transition into flow easier (e.g. relaxation → flow = easier than stress → flow) and distraction is a big factor in taking someone out of their flow state. These findings led us to the realization, that a mixed approach with an initial, guided relaxation phase to set goals for the work session and creating an emotional baseline could potentially ease the user's engagement and transition into a productive flow state.
3. **Redefining the Deliverable**: In light of our new direction, we transitioned from developing a user-centered mobile application to creating a web application tailored for haptic designers. This shift was not only aligned with the identified niche need but also offered greater flexibility and potential for professional use.
4. **Proof of Concept Goal**: The change in our target audience and platform was also influenced by the project's revised goal to serve as a proof of concept. Given the time constraints and the complex nature of integrating AI with audio and haptic feedback, a web application for haptic designers presented a more viable and focused approach. It allowed us to demonstrate the core functionalities and potential of our concept, setting the foundation for future enhancements and expansions.

We then came up with an iterated idea for our project, adapting to new insights and practical considerations. While the original focus was on creating a tool to induce flow state through audio-based haptic feedback, the revised concept shifted towards a more feasible and research-informed relaxation to “create a base for flow" approach. Here's an overview of the key elements of the iterated idea:

Our Guiding Star 🌟

The guiding principle of the revised project was the assumption that accessing flow states is more achievable from a relaxed state. This approach was based on the understanding that relaxation could serve as a preparatory phase, helping users to set goals and structure their thoughts, thereby reducing anxiety. This structured and calm mindset was seen as a potential foundation for more effectively inducing flow states.

Revised Framework

The revised approach was structured around two main components: Baseline and Sparkles.

Baseline Audio

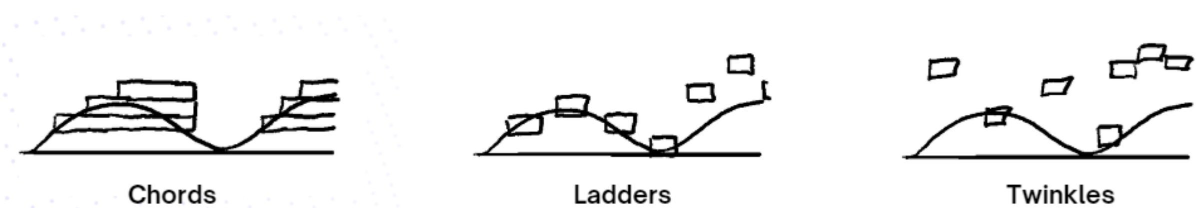
- **Utilizing the Existing Python script**: We planned to use the Python script provided by our industry partner for creating the baseline audio. However, this was later revised into Javascript using Tone.js as a base library for producing the sounds. For the future, we imagined an alternative version and substituted the Tone.js instrument with a more versatile RNBO web-audio export patch.
- **User Control Over Inputs**: We planned that a future version of the application would enable users to dynamically shift inputs according to an emotional grid, influencing the baseline audio to match the desired state and their complexity preferences to target the ideal engagement factor and have it as non-intrusive as possible.
- **Integration of the Max 8's RNBO Object:** By creating a javascript RNBO instrument patch, we would be able to play the generated MIDI on different instrument types to allow for rapid prototyping of waveforms and testing on different actuators with adjustable instrumentalization.
- **Two Different Approaches:** For the creation of baseline audio, two approaches were possible. The basic concept is to lower the notes played while simultaneously manipulating the tempo to have the last third of the

generated audio on the target frequency for the Breaths Per Minute (BrPM), scientifically backed with relaxation.

- Linear MIDI + Instrument Pitch: The first utilizes linear MIDI creation that plays an instrument which will be pitch shifted to have the gradual detune effect for the relaxation. For example, one can choose a starting frequency where an associated note will be chosen as the baseline note and determining the musical scale of the composition. This composition can then be played linearly given the pitch of the instrument, playing the same tone and a melody suitable to that base tone) will be manipulated to gradually lower the compositions overall frequency. The tempo will be separately slowed and has no effect on the pitch logic of this approach.
- Ladder MIDI: The second utilizes the MIDI ladder concept, where fixed notes are played. The notes are associated with a certain frequency pitch, e.g. concert pitch of A4=440Hz, and the change in the frequency parameter towards lower and slower notes will be nudged to the closest note associated with the current value. This concludes in a MIDI composition, that when reaching a pivotal frequency threshold, the next generated note will be a lower one.

Dynamic Sparkles

- **AI-Generated Audio**: The plan included using GPT-4 to dynamically generate audio that would layer over the baseline audio, aiding in inducing relaxation. These "sparkles" can be differentiated into Chords, Ladders and Twinkles:



Chords are Triads, Ladders are chords that are split and played up and down and Twinkles are short notes with no predefined sequence.

- **Incorporating Natural Sound Characteristics**: We considered integrating sounds found in nature (such as bird sounds or flowing water) to enrich the audio experience. This could later be implemented in future work on the prototype, considering technological advancements in AI Audio generation like the audio sound effect generation model “Audiobox” by Meta.

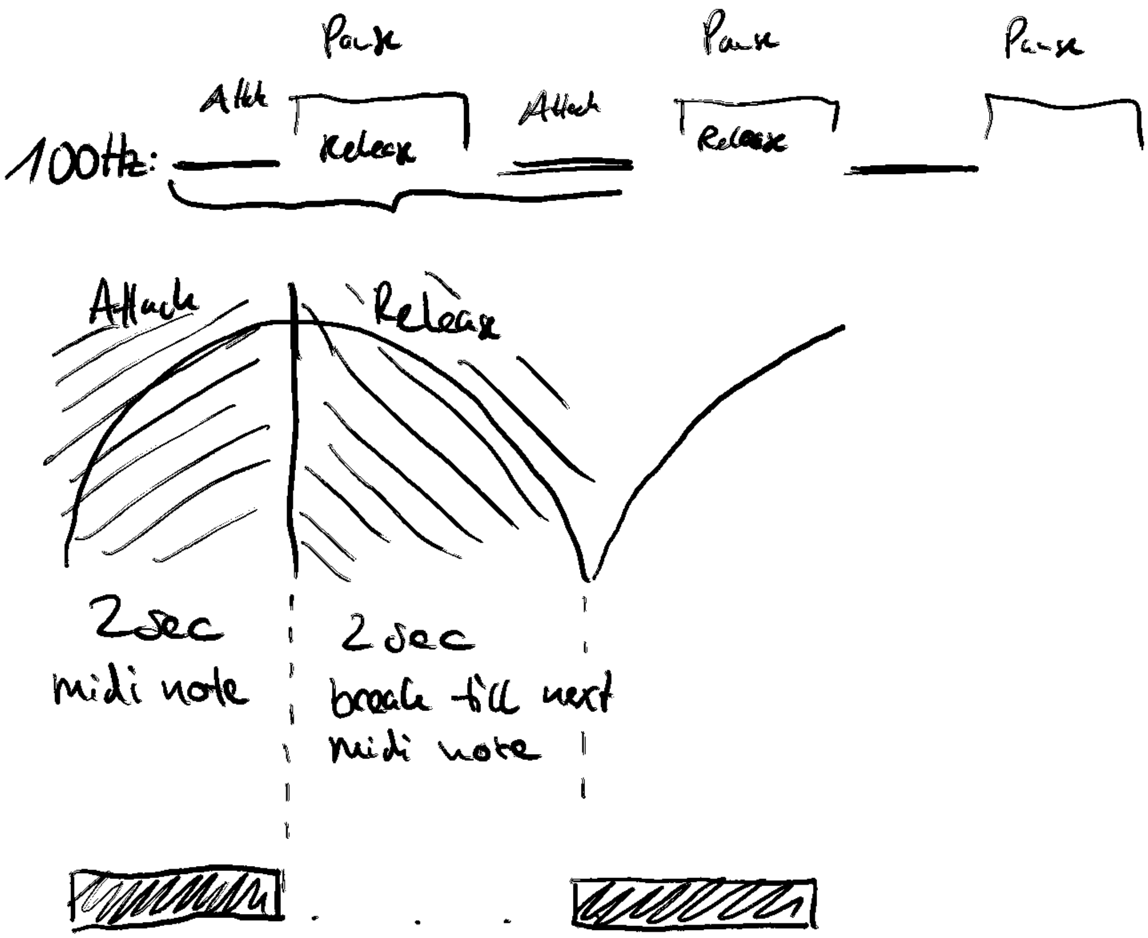
Outcomes of the Shift

- **Enhanced Focus on Quality**: By narrowing our scope to establish the first step into relaxation and targeting haptic designers, we were able to concentrate our resources on developing an MVP.
- **Increased Technical Depth**: By switching to a web application, we allow for a more modular approach thanks in part to the framework we’re utilizing. By using the React Javascript library as a base, we allow for the creation of components that can be rearranged based on the needs of the professional. This also allows for the use of an extensive list of Javascript libraries such as OpenAI, Tone.js, RNBO.js, Bootstrap, etc.
- **Foundation for Future Expansion**: The proof of concept established with this web application lays a solid groundwork for future development. It provides a platform that can be improved and potentially expanded to include additional affective states and different context use cases apart from relaxation enhancement and stress reduction in work settings.

Building the MVP

As previously mentioned, our MVP was a React-based prototype. We anticipated that this would facilitate the transition to a mobile app or enable it to function as a web application if required later. Additionally, this approach allowed us to utilize sound libraries like Tone.js and paved the way for future integration of a digital instrument (specifically designed for haptics) created using software called RNBO. The design of the Baseline and Sparkles as separate code components makes it possible to easily swap these elements, further enhancing the application's versatility.

The baseline was created only through code on parameters like frequency given by the user.
The up and down modulated sine wave effect was created through attack and release filters on single midi notes:



For the “sparkles”, we created prompts to teach GPT-4 how to compose musical notes in the way we wanted them (example for the twinkles prompt):

```
export const promptTwinkles =
`I need assistance in producing AI-generated text that I convert

1. **Note Specification**: Use pitch values corresponding to fre
2. **Integration of Silence**: Include silences, denoted by a tu
```

```
3. **Musical Elements**:
1. **Preferred Frequencies**: 40-60 Hz are a pleasant range for
2. **Consonance and Dissonance**: The papers extend the concept
3. **Individual Variability**: While certain frequencies and int
4. **Rhythmic and Smooth Patterns**: Preferred are non intrusive
5. **Resonance and Neural Entrainment**: The concept of resonanc
- **Pitch Range**: Emphasize pitches that align with or build on
    - **Rhythm and Contour**: Develop rhythms and contours refle
    - **Consonance and Dissonance**: Apply principles of consona
Syntax:
{
// the transport and timing data
header: {
name: String,                     // the name of the first empty
                                  // which is usually the song n
tempos: TempoEvent[],             // the tempo, e.g. 120
timeSignatures: TimeSignatureEvent[],  // the time signature, e

PPQ: Number                       // the Pulses Per Quarter of t
                                  // this is read only
},

duration: Number,                   // the time until the last n

// an array of midi tracks
tracks: [
{
  name: String,                   // the track name if one was g

  channel: Number,                // channel
                                  // the ID for this channel; 9
                                  // reserved for percussion
  notes: [
    {
      midi: Number,               // midi number, e.g. 60
      time: Number,               // time in seconds
```

```
      ticks: Number,              // time in ticks
      name: String,               // note name, e.g. "C4",
      pitch: String,              // the pitch class, e.g. "C",
      octave : Number,            // the octave, e.g. 4
      velocity: Number,           // normalized 0-1 velocity
      duration: Number,           // duration in seconds betweer
    }
  ],

  // midi control changes
  controlChanges: {
    // if there are control changes in the midi file
    '91': [
      {
        number: Number,           // the cc number
        ticks: Number,            // time in ticks
        time: Number,             // time in seconds
        value: Number,            // normalized 0-1
      }
    ],
  },

  instrument: {                   // and object representing the
    number : Number,              // the instrument number 0-12
    family: String,               // the family of instruments,
    name : String,                // the name of the instrument
    percussion: Boolean,          // if the instrument is a perc
  },
}
]
} here is a filled out example: "{
"header": {
"keySignatures": [],
"meta": [],
"name": "",
"ppq": 240,
```

```
"tempos": [
  {
    "bpm": 120,
    "ticks": 0
  }
],
"timeSignatures": [
  {
    "ticks": 0,
    "timeSignature": [
      4,
      4
    ],
    "measures": 0
  }
]
},
"tracks": [
{
  "channel": 0,
  "controlChanges": {
    "7": [
      {
        "number": 7,
        "ticks": 0,
        "time": 0,
        "value": 1
      }
    ]
  },
  "pitchBends": [],
  "instrument": {
    "family": "bass",
    "number": 33,
    "name": "electric bass (finger)"
  },
```

```
    "name": "",
    "notes": [
      {
        "duration": 0.26875,
        "durationTicks": 129,
        "midi": 45,
        "name": "A2",
        "ticks": 0,
        "time": 0,
        "velocity": 0.6141732283464567
      },
      {
        "duration": 0.25625000000000001,
        "durationTicks": 123,
        "midi": 48,
        "name": "C3",
        "ticks": 360,
        "time": 0.75,
        "velocity": 0.6377952755905512
      },
      {
        "duration": 0.26875000000000004,
        "durationTicks": 129,
        "midi": 50,
        "name": "D3",
        "ticks": 480,
        "time": 1,
        "velocity": 0.6220472440944882
      },
      {
        "duration": 0.1333333333333333,
        "durationTicks": 64,
        "midi": 52,
        "name": "E3",
        "ticks": 600,
        "time": 1.25,
```

```
      "velocity": 0.5354330708661418
    },
    {
      "duration": 0.7333333333333334,
      "durationTicks": 352,
      "midi": 45,
      "name": "A2",
      "ticks": 840,
      "time": 1.75,
      "velocity": 0.6456692913385826
    },
    {
      "duration": 0.4791666666666665,
      "durationTicks": 230,
      "midi": 43,
      "name": "G2",
      "ticks": 1200,
      "time": 2.5,
      "velocity": 0.4881889763779528
    },
    {
      "duration": 0.28125,
      "durationTicks": 135,
      "midi": 45,
      "name": "A2",
      "ticks": 1440,
      "time": 3,
      "velocity": 0.6299212598425197
    },
    {
      "duration": 0.16041666666666643,
      "durationTicks": 77,
      "midi": 45,
      "name": "A2",
      "ticks": 1920,
      "time": 4,
```

```
      "velocity": 0.6929133858267716
    },
    {
      "duration": 0.16041666666666643,
      "durationTicks": 77,
      "midi": 48,
      "name": "C3",
      "ticks": 2280,
      "time": 4.75,
      "velocity": 0.6062992125984252
    },
    {
      "duration": 0.25416666666666643,
      "durationTicks": 122,
      "midi": 50,
      "name": "D3",
      "ticks": 2400,
      "time": 5,
      "velocity": 0.7086614173228346
    },
    {
      "duration": 0.2020833333333334,
      "durationTicks": 97,
      "midi": 52,
      "name": "E3",
      "ticks": 2520,
      "time": 5.25,
      "velocity": 0.5905511811023622
    },
    {
      "duration": 0.760416666666667,
      "durationTicks": 365,
      "midi": 45,
      "name": "A2",
      "ticks": 2760,
      "time": 5.75,
```

```
      "velocity": 0.5905511811023622
    },
    {
      "duration": 0.47916666666666696,
      "durationTicks": 230,
      "midi": 48,
      "name": "C3",
      "ticks": 3120,
      "time": 6.5,
      "velocity": 0.6062992125984252
    },
    {
        "duration": 0.1875,
        "durationTicks": 90,
        "midi": 45,
        "name": "A2",
        "ticks": 3360,
        "time": 7,
        "velocity": 0.6141732283464567
      }
  ],
  "endOfTrackTicks": 3839
}
]
};"

Here is another example: {
"header": {
"keySignatures": [],
"meta": [],
"name": "",
"ppq": 240,
"tempos": [
  {
    "bpm": 120,
    "ticks": 0
```

```
    }
  ],
  "timeSignatures": [
    {
      "ticks": 0,
      "timeSignature": [
        4,
        4
      ],
      "measures": 0
    }
  ]
  },
  "tracks": [
  {
    "channel": 0,
    "controlChanges": {
      "7": [
        {
          "number": 7,
          "ticks": 0,
          "time": 0,
          "value": 1
        }
      ]
    },
    "pitchBends": [],
    "instrument": {
      "family": "bass",
      "number": 33,
      "name": "electric bass (finger)"
    },
    "name": "",
    "notes": [
      {
        "duration": 0.26875,
```

```
      "durationTicks": 129,
      "midi": 45,
      "name": "A2",
      "ticks": 0,
      "time": 0,
      "velocity": 0.6141732283464567
    },
    {
      "duration": 0.25625000000000001,
      "durationTicks": 123,
      "midi": 48,
      "name": "C3",
      "ticks": 360,
      "time": 0.75,
      "velocity": 0.6377952755905512
    },
    {
      "duration": 0.26875000000000004,
      "durationTicks": 129,
      "midi": 50,
      "name": "D3",
      "ticks": 480,
      "time": 1,
      "velocity": 0.6220472440944882
    },
    {
      "duration": 0.1333333333333333,
      "durationTicks": 64,
      "midi": 52,
      "name": "E3",
      "ticks": 600,
      "time": 1.25,
      "velocity": 0.5354330708661418
    },
    {
      "duration": 0.7333333333333334,
```

```
      "durationTicks": 352,
      "midi": 45,
      "name": "A2",
      "ticks": 840,
      "time": 1.75,
      "velocity": 0.6456692913385826
    },
    {
      "duration": 0.47916666666666665,
      "durationTicks": 230,
      "midi": 43,
      "name": "G2",
      "ticks": 1200,
      "time": 2.5,
      "velocity": 0.4881889763779528
    },
    {
      "duration": 0.28125,
      "durationTicks": 135,
      "midi": 45,
      "name": "A2",
      "ticks": 1440,
      "time": 3,
      "velocity": 0.6299212598425197
    },
    {
      "duration": 0.16041666666666643,
      "durationTicks": 77,
      "midi": 45,
      "name": "A2",
      "ticks": 1920,
      "time": 4,
      "velocity": 0.6929133858267716
    },
    {
      "duration": 0.16041666666666643,
```

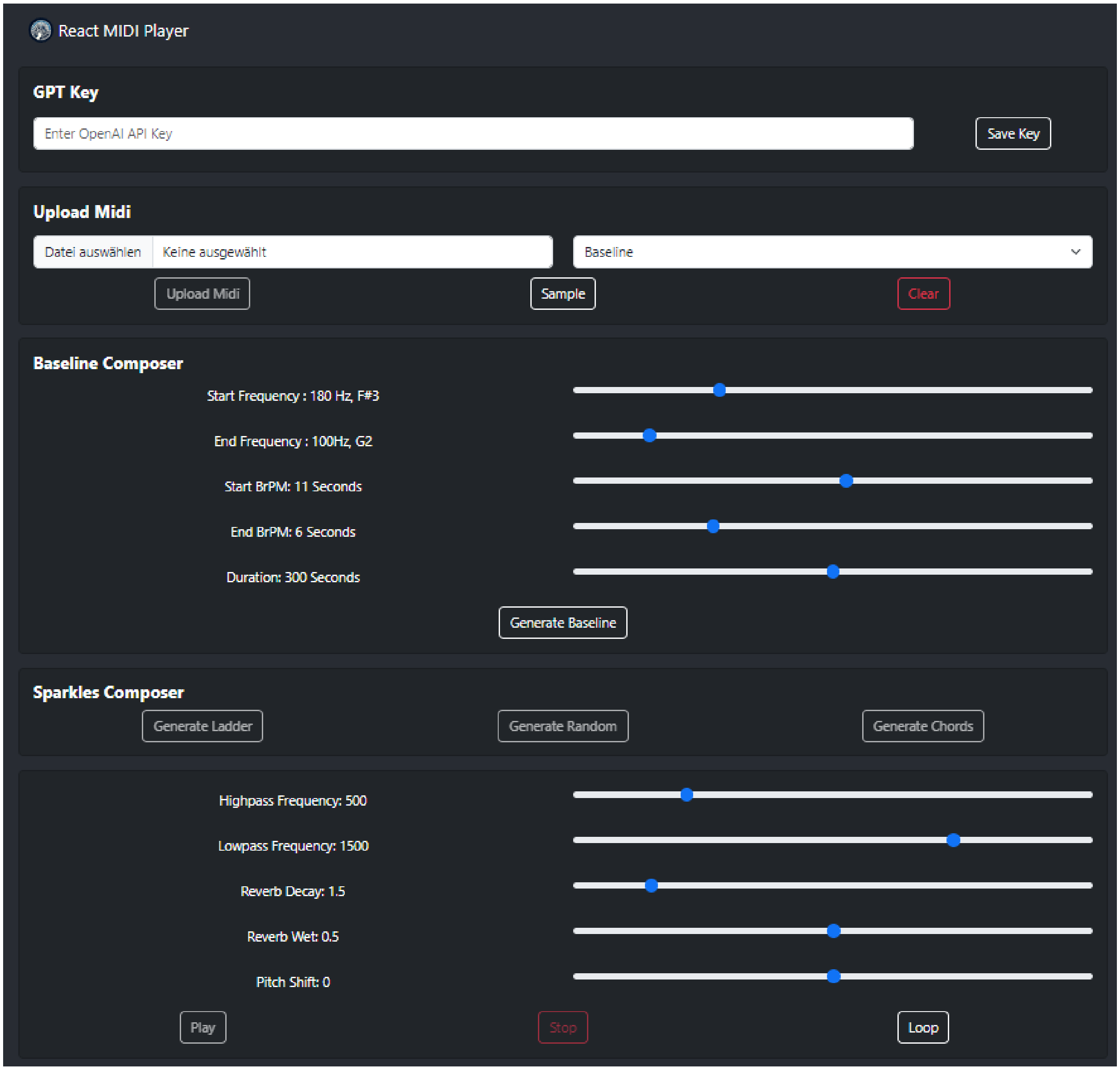
```
      "durationTicks": 77,
      "midi": 48,
      "name": "C3",
      "ticks": 2280,
      "time": 4.75,
      "velocity": 0.6062992125984252
    },
    {
      "duration": 0.25416666666666643,
      "durationTicks": 122,
      "midi": 50,
      "name": "D3",
      "ticks": 2400,
      "time": 5,
      "velocity": 0.7086614173228346
    },
    {
      "duration": 0.2020833333333334,
      "durationTicks": 97,
      "midi": 52,
      "name": "E3",
      "ticks": 2520,
      "time": 5.25,
      "velocity": 0.5905511811023622
    },
    {
      "duration": 0.760416666666667,
      "durationTicks": 365,
      "midi": 45,
      "name": "A2",
      "ticks": 2760,
      "time": 5.75,
      "velocity": 0.5905511811023622
    },
    {
      "duration": 0.47916666666666696,
```

```
      "durationTicks": 230,
      "midi": 48,
      "name": "C3",
      "ticks": 3120,
      "time": 6.5,
      "velocity": 0.6062992125984252
    },
    {
      "duration": 0.1875,
      "durationTicks": 90,
      "midi": 45,
      "name": "A2",
      "ticks": 3360,
      "time": 7,
      "velocity": 0.6141732283464567
    }
  ],
  "endOfTrackTicks": 3839
}
]
}
5. **Individual Variability**: Recognize the variation in indivi

The goal is to create melodies that are not only musically coher
```

The answer was a list of notes in a Tone.js readable MIDI JSON format that were converted into playable MIDI again.

Analysis of Project Outcome



The final deliverable of our project is the web application designed for haptic designers.

1. **AI-Driven Procedural Audio Generation**: One of the application's core explorations was to generate unique audio compositions with LLM AI capabilities. By few-shotting the GPT model with a JSON structure, which was parsable by our code, we were able to create playable MIDI files with a LLM. This potentially offers a fresh perspective for establishing an AI promptable playground for haptic designers. Our experiment aimed at the creation of

harmonic soundscapes that possibly could add to the relaxation-inducing baseline and therefore enhance the personal relaxation experience.

2. **Customizable Haptic Feedback Parameters**: Designers have extensive control over haptic feedback parameters like frequency, tempo, and duration, essential for crafting diverse and nuanced haptic experiences suited to relaxation scenarios.
3. **Responsive User Interface**: The interface is intuitively designed, allowing easy navigation, parameter adjustments, and previews of haptic designs. The application is also responsive across different devices, ensuring flexibility and accessibility for designers using various platforms.
4. Basic **Sound Editing Tools**: The application includes tools for basic audio editing, such as amplitude modulation and frequency adjustments, and allows overlaying multiple soundtracks for rich sound design that could translate into interesting haptics.
5. **Integration with Tone.js and RNBO.js**: Utilizing these advanced libraries, the app provides robust and high-fidelity audio capabilities, crucial for professional audio-based haptic design.
6. **Custom MIDI Files**:

- **Upload Capability**: One of the application's key features is the ability for users to upload their custom MIDI files for both the sparkles and baseline components. This functionality greatly enhances the customization potential, enabling designers to bring their unique creative visions into their haptic designs.
- **Download Option**: Additionally, the application provides the capability to download the final MIDI file. This feature is particularly valuable as it allows for external customization and further manipulation of the audio composition. Designers can use this downloaded file in other software or platforms, offering even more flexibility and creative control in their design process.

7. **FX Features**:

Adding these additional audio effects can significantly alter the final haptic experience.

- **High-Pass and Low-Pass Filters**: These filters enable designers to fine-tune their audio by controlling the frequency range that is let pass the filters. This is crucial for crafting the desired auditory texture and higher the quality of the auditory signal from the perspective of audio engineering, freeing up headroom for potentially more detailed and desirable haptic sensations.
- **Reverb Effects**: Adds depth and space to the audio, allowing designers to simulate different environmental acoustics for a more immersive haptic experience.
- **Delay Effects:** Adds little pulses that mirror the first “big impact” of played haptic audio. This can contribute to more complex haptic sensations while requiring less compositional effort.

> ⚠️ Add more filters as necessary

1. **Responsive Design for Various Devices**: The react application is responsive across different devices, ensuring flexibility and accessibility for designers using various platforms. The core thought was to be able to connect a

Bluetooth device and just play the audio generated by the website through the connected Bluetooth speaker, functioning as an audio-haptic actuator.

The project outcome, though divergent from the initial concept, stands as a testament to our team's ability to navigate complex technological challenges and deliver a product that significantly advances the field of experimental haptic design and AI-audio integration.

Comparison of Initial Goal vs. Final Outcome

Initial Goal

The original aim of the project was to develop a mobile app that could induce a state of flow through the integration of live AI-generated long-form audio translated into haptic feedback when played on a suitable audio-haptic actuator. This concept was grounded in the belief that certain auditory and tactile stimuli, when orchestrated effectively, could significantly influence a user's cognitive and emotional state, facilitating the achievement of a highly focused and immersive flow state. Key elements of this initial goal included:

1. **Research on Flow State Inducers**: In-depth exploration of the audio characteristics necessary to induce flow, such as BPM, frequency, and others.
2. **AI Model Training**: Utilizing and fine-tuning GPT models to generate suitable audio content.
3. **Haptic Feedback Development**: Creating a SuperCollider program template for waveform generation and exploring the integration of these waveforms into haptic devices.
4. **Optional Audio Development**: Considering both AI-driven and non-AI methods for additional audio generation, with tools like Audiocraft and MAXSP.
5. **Focus on flow State Characteristics**: Emphasizing various audio elements like frequency, amplitude, and rhythm to induce flow state.

Final Outcome

The project's final outcome, however, shifted significantly from this initial goal. The revised project, while still rooted in the use of audio-haptic feedback, leans now towards a more practical and achievable objective. The key changes in the final outcome included:

1. **Relaxation instead of Flow:** The project shifted focus towards the idea of using relaxation as a precursor to achieving a flow state. This concept was based on the assumption that a relaxed state could set the stage for a more effective transition into flow.
2. **Infrastructure Development**: Rather than completing the entire tool within the project's timeframe, the team focused on building the foundational system that could facilitate the future realization of the concept.
3. **Baseline and Dynamic Audio Components**: The project emphasized developing a system that balanced a hardcoded, yet individually adjustable, stable baseline haptic audio and conducting the first experiments for enriching the experience with subtle variations of dynamic, AI-generated melodies and harmonies.
4. **Practical Adjustments**: Given the scope and resource limitations, the project strategically adapted its goals, concentrating on creating an insightful MVP that is a scalable and adaptable system rather than a fully realized product.

Analysis of the Shift

The shift from the initial goal to the outcome can be attributed to several factors:

- **Resource and Time Constraints**: The ambitious nature of the original goal, coupled with the time and resource limitations, necessitated a more pragmatic approach.
- **Technical Feasibility**: As the project progressed, the technical challenges of integrating AI-driven audio with haptic feedback became more apparent, leading to a refocus on building a foundational system.
- **Expert Consultation and Market Viability**: Feedback from experts and market analysis highlighted the need for a more focused and validated approach, leading to our flow-through relaxation concept.
- **Adaptability to Insights**: The team's willingness to adapt to new insights and practical considerations allowed for a more achievable and strategically sound outcome, with ultimately interesting learnings, research insights and outlooks for future projects.

Main Roles and Work Distribution

The project's success was underpinned by the distinct yet interconnected roles of each team member. Their diverse expertise contributed to the multifaceted nature of the project, ensuring thorough research, innovative design, and effective implementation. Below is a detailed overview of each member's roles and contributions.

1. **Vincent Göke**
    - **Role**: Sound Designer & Research Lead
    - **Contributions**:
        - Conducted extensive research on affective audio experiences and their potential translation into haptics, current research underlining haptics' role in affective state change, with a focus on relaxation and flow states and psychological background for understanding the scientific state of research for these two states.
        - Led the initial user testing for haptic embodiments, specifically integrating a haptic speaker in a neck pillow. This was initialized to also test the three different audio-haptic actuators that Daniel Shor gave us as tangible testing hardware.
        - Provided expert consultation on all audio-related matters, ensuring the audio output was in line with project goals. Additionally proposed the MIDI framework to provide the best possible flexibility for potential readjustments of technical implementations.
        - Engaged in musical and haptic prompt engineering for GPT-4, tailoring it to meet specific conditions required for AI MIDI composition creation of the project.
        - Designed the RNBO.js instrumentalization solution for tone generation and gradual detuning, extending the application's audio generation capabilities.
2. **JM Santiago III**
    - **Role**: React Developer
    - **Contributions**:

    - Conducted research on generative AI solutions for audio and haptic generation.
    - Participated in the initial user testing phase, focusing on the integration of a haptic speaker in a neck pillow.
    - Led the design and development of the React-based web application, ensuring a user-friendly and functional interface.
    - Managed the deployment of the final web application, overseeing its launch and functionality.
    - Responsible for documenting the entire project process, including each milestone and report, ensuring a comprehensive record of the project's evolution.

3. **Moritz Sendner**
    - **Role**: Framework Designer
    - **Contributions**:
        - Focused on researching flow-state awareness and the influence of haptics on this state.
        - Designed the final framework of the project that combines the initial baseline sound with the dynamic sparkles sound, the basis of the web application.
        - Prepared and tested various materials for haptic embodiments while addressing all issues related to the haptic speakers, enhancing their performance and reliability.
        - Enhanced the initial Python script for baseline audio generation, contributing to the foundational elements of the application.
        - Very active developer in the reacted-based solution using Tone.js for the baseline generation