**"MoodTune: Visualizing and Predicting Moods Based on Music Listening Behavior"**

**Group members: Rob Ranieri, Peter Lin, Gwen Seymour**

**1. Project Overview**

The proposed project aims to develop an interactive dashboard that visualizes users’ music listening behavior and predicts their moods using machine learning models. By analyzing listening patterns, audio features, and contextual data, the dashboard provides insights into how music correlates with and potentially influences mood over time.

**2. Objectives**

* **Visualize** music listening behavior (frequency, time of day, genre, tempo, etc.)
* **Integrate ML models** to predict user mood from audio features and contextual metadata
* **Provide recommendations** or mood analysis feedback in an engaging dashboard format
* **Analyze trends** between music types and self-reported emotional states

**3. Data Sources**

* **Spotify API**: Track metadata (e.g., tempo, key, danceability, valence, energy, etc.), listening history
* **DEAM (Database for Emotional Analysis of Music):** To track emotion recognition in music research
* **Self-reported mood data**: User-tagged emotions (via survey, form, or mood logging)

**4. Key Features**

* 📈 **Dashboard Components**:
  + Listening frequency over time
  + Genre and mood heatmaps
  + Energy vs Valence trends
  + Time-of-day listening behavior
  + Personalized insights
* 🤖 **Machine Learning Integration**:
  + **Classification model** (Random Forest / Logistic Regression) to predict mood from music features
  + **Clustering** (K-Means) to group similar mood profiles
  + **Recommendation system** to suggest tracks for mood regulation
* 🧠 **Emotion Modeling**:
  + Map audio features to emotion labels (happy, sad, relaxed, angry, etc.)
  + Use valence-arousal mapping to interpret moods

**5. Tools & Technologies**

* **Python**: Data processing, modeling (Pandas, NumPy, Scikit-learn)
* **Spotify API / Spotipy**: Data ingestion
* **PyQt:** Desktop GUI development and interactive dashboard
* **VisualStudio:** Interactive prototype
* **PostgreSQL / MongoDB / Spark**: Data storage
* **Seaborn / Plotly**: Advanced visualizations

**6. Timeline**

**Week 1 – Data & Modeling (Foundation)**

1. Set up **Spotify API** access using Spotipy
   * Pull sample **listening history** and audio features
   * Load and explore **DEAM dataset**
2. Create a **self-reported mood input mechanism**
   * CSV form, Google Form, or in-app tagging mockup
3. Build a **data preprocessing pipeline**
   * Clean and merge DEAM, Spotify, and mood tags
   * Feature engineering (e.g., normalize tempo, one-hot genres)
4. Train **ML classification model** (Random Forest / Logistic Regression)
   * Input: Spotify audio features
   * Output: Mood labels (happy, sad, angry, etc.)
   * Evaluate performance (accuracy, F1-score)
5. Apply **K-Means clustering** for mood profiles
   * Group tracks or users into mood archetypes
6. Analyze **correlations and trends**
   * Genre vs mood
   * Valence/Energy scatter plots
   * Time-of-day listening patterns

**Week 2 – PyQt GUI & Dashboard Integration**

1. Set up base **PyQt project structure**
   * Main window, navigation tabs (e.g., Overview, Trends, Predictions)
2. Integrate **listening frequency chart**  
   Create **genre + mood heatmaps** in PyQt using Plotly or matplotlib
3. Embed **valence vs energy scatter plot**  
   Add **time-of-day listening behavior** chart
4. Load trained **ML model into PyQt**  
   Allow user to select a track and **predict mood** in real time
5. Build a simple **recommendation engine**
   * Based on predicted mood, suggest Spotify tracks (from local or API)
6. Polish GUI: Add dropdowns, user filters, error handling  
   Enable refresh & reload of data sources
7. Final testing and **demo preparation**
   * Package app with dummy data
   * Record walk-through or create slides/screenshots
   * Push to GitHub

**7. Success Metrics**

* **Model accuracy / F1-score** in mood prediction
* **User engagement** with dashboard (views, time spent)
* **Visual clarity** and interpretability of insights
* **Relevance** of recommendations or mood analysis

**8. Target Audience**

* Individual users interested in mood-aware music insights
* Mental health researchers exploring music’s emotional impact
* Streaming platforms seeking user emotion personalization

**9. Potential Extensions**

* Real-time mood tracking with wearable integration (e.g., Fitbit, Apple Watch)
* Natural Language Processing (NLP) to include lyrics analysis
* Mood-aware playlist generation using reinforcement learning