Release-1 Report

Project Overview

Code Mood is a fun, developer-first VS Code extension that aims to detect a developer's emotional state based on their code and responds with themes, fonts, music, and humor tailored to that mood. It blends productivity tools with entertaining, mood-driven dynamics to make coding more expressive and engaging.

Features Implemented

1. Mood Detection Based on Code

- Analyzes code structures, error patterns, comments, and complexity to determine the coder's current mood.

2. Mood-Aligned Fonts

- Fonts dynamically update to reflect the detected mood. For example:
    - Comic Sans for chaotic/frustrated coding.
    - Consolas for calm focus.

3. Mood-Aligned Themes

- Color themes adjust in real time. For example:
    - Dark+ for frustration.
    - Monokai for a focused mood.
    - Solarized for an anxious mood.

4. Code Smell Detection

- Identifies poor coding practices such as:
    - Long methods
    - Methods with too many parameters
    - Deeply nested blocks
    - Large classes

    - Duplicated codes

---

5. Sarcastic Comments on Code Smells

- Sarcastic ghost-style comments are injected for each smell.
    - Examples:
        - `"# Because having 200 lines in one function is totally maintainable.`
        - `"# That unused variable? It's just here for emotional support.`

---

6. Corresponding Mood-Based Music

- Auto-plays background music (audio-only) depending on the mood of the developer.
- Some of the playlists include:
    - Chill Lo-fi
    - Angry metal riffs
    - Dramatic movie soundtrack.

---

7. Per-File Timer Tracking

- Each file opened in the editor is tracked for time spent.
- Helps identify coding hotspots or zones of procrastination.

---

8. Analytics & Flow Chart

- Timer data is visualized via:
    - Coding activity graphs
    - Time-distribution per file
    - Focus streak detection

Methodology & Techniques Used

- The extension integrates **Mistral-7B-Instruct** via **Hugging Face's Inference API** to analyze code for emotional cues. A tailored system prompt drives the model to infer developer mood from coding style. **Lightweight, high-performance architecture** ensures fast predictions. Periodic re-evaluation updates the mood, triggering dynamic changes in **VS Code themes and fonts**.

- For code smell detection, it employs **AST-based static analysis** using Python's built-in ast module. A custom CodeSmellDetector class extends ast.NodeVisitor, overriding key visit methods to inspect definitions and control flow for known smells.

- Humorous feedback is generated by sending code smells to **Mistral-7B-Instruct** with a sarcasm prompt. The response is mapped to line numbers and displayed via **ghost-style inline decorations** using the **Diagnostics API**. Hover support adds context, and asynchronous execution keeps the editor responsive.

- The extension uses editor listeners like onDidOpenTextDocument and onDidChangeActiveTextEditor to track per-file coding time. Timestamps are stored in startTime and totalTime maps. On command, durations are computed and formatted into a **Chart.js pie chart** inside a WebviewPanel. Metrics like **productivity score** are derived from this time data.

- To play music based on mood, a **WebviewPanel** is created with UI controls. When a command is received via onDidReceiveMessage, the extension spawns a **VS Code terminal**, runs a shell command that uses **yt-dlp** to extract audio and **pipes it into ffplay**. This enables real-time, online music playback.

Technologies Used

- **Visual Studio Code API** – Used to build the extension's UI and logic through commands, decorations, diagnostics, and terminal integration.
- **JavaScript (Node.js)** – Core language for implementing features with modular CommonJS structure.
- **Chart.js** – Integrated into webviews for visualizing time data using interactive pie charts.
- **yt-dlp + ffplay** – Terminal-based tools used for extracting and streaming music audio based on mood.

- **VS Code Webview API** – Powers the custom music panel and time chart interface within the editor.
- **Hugging Face Inference API** – Utilized for detecting developer mood through lightweight sentiment analysis.
- **Diagnostics & Decoration API** – Used to highlight code smells and add inline ghost comments with a sarcastic tone in Python files.

Contributions

- **Jyothiraditya**
  - Productivity score generation based on tracked time per file
  - Font change based on mood and slides design
- **Sonith**
  - Enhance extension with code smell detection and update dependencies
  - Added work time tracking per files and generation of pie chart
- **Mokshith**
  - Added ghost comments for code smells
  - Developer's mood detection based on code
- **Pavan**
  - Extension command for playing music
  - Theme change based on mood and documentation
- **Sasaank**
  - Added code smell detection functionality
  - Integrated Mistral AI for sarcastic comments and updated diagnostics
- **Madhav**
  - Squiggle diagnostics for code smells
  - Mood based music system