

1.Introduction:

- ❖ Project title: rhythmic tune
- ❖ Team id:25TMID37533
- ❖ Team leader: Gayathri B
 - Email ID: Gayathrikavya2532007@gmail.com
 - Team members:
 - A.Anitha
 - P.dharani
 - S.Merlin
 - Email ID:
 - Dharanipandurangan46@gmail.com
 - samuvelmerlin7@gmail.com
 - anitha27arumugam@gmail.com

2. project overview:

“Rhythmic Tunes” is a project where you explore the role of rhythm in music. You’ll analyse different rhythmic styles, create your own beat patterns, and present them either through live performance, coding, or digital tools. The focus is on understanding how rhythm drives the energy and mood of music.

❖ Features of Rhythmic Tunes

1. Beat (Pulse)

The steady, repeating pulse in music.

Acts like a “heartbeat” that listeners can tap or clap along to.

2. Tempo (Speed)

The speed of the beats (fast, slow, moderate).

Measured in BPM (beats per minute).

3. Time Signature

The pattern in which beats are grouped (e.g., 4/4, 3/4, 6/8).

4. Accent & Stress

Some beats are stronger (accented) than others.

5. Syncopation

Placing stress on unexpected beats or off-beats.

Common in jazz, funk, and hip-hop.

3. architecture: of Rhythmic Tunes

1. Foundation Layer - Beat (Pulse)

The core unit of rhythm.

Like the heartbeat of music, always steady.

Everything is built on this repeating pulse.

2. Structural Layer - Time Signature

Organizes beats into groups (measures/bars).

Examples:

4/4 (most common in pop, rock).

$\frac{3}{4}$ (waltz style).

6/8 (folk and ballads).

3. Pattern Layer - Rhythm Cycle

Beats arranged into a cycle (loop).

Repetition gives rhythm its recognizable identity.

Example: Indian tala (16 beats - Teental), Western drum loop.

4.set-up instructions

❖ Prerequisites for Rhythmic Tunes



1. Prerequisites

Before setting up, ensure you have the following:

Hardware:

A computer/laptop (Windows, macOS, or Linux)

Minimum 4 GB RAM (8 GB recommended)

Speakers or headphones

Optional: MIDI keyboard or digital drum pad

Software:

Python (3.8 or above) if you're coding with rhythm libraries

A Digital Audio Workstation (DAW) like FL Studio, Ableton, GarageBand, or LMMS (free)

Audio libraries/tools (depending on approach):

Pygame or pydub (Python-based beat generation)

Sonic Pi (for live coding music)

MIDI drivers if using instruments

❖ Installation steps:

```

Import pygame, time
Pygame.mixer.init()
Kick = pygame.mixer.Sound("kick.wav")
Snare = pygame.mixer.Sound("snare.wav")
Hihat = pygame.mixer.Sound("hihat.wav")
BPM = 120
Beat_duration = 60 / BPM
Kick_pattern = [1, 0, 0, 0]
Snare_pattern = [0, 0, 1, 0]
Hihat_pattern = [1, 1, 1, 1]
Print("Playing rhythmic tune...")
For _ in range(8):
    For I in range(4):
        If kick_pattern[i]: kick.play()
        If snare_pattern[i]: snare.play()
        If hihat_pattern[i]: hihat.play()          Time.sleep(beat_duration)
Print("Finished.")

```

5.Folder structure:

```

Import pygame
Import time
Import random
# Initialize mixer
Pygame.mixer.init()
# Load drum sounds
Kick = pygame.mixer.Sound("kick.wav")
Snare = pygame.mixer.Sound("snare.wav")
Hihat = pygame.mixer.Sound("hihat.wav")
# Tempo (beats per minute)
BPM = 100
Beat_duration = 60 / BPM
# Number of bars to play
Bars = 8
Print("🎵 Playing random rhythmic tune...")

```

```

For bar in range(bars):
    Print(f"Bar {bar+1}")
    # Generate random 4-beat pattern for each instrument
    Kick_pattern = [random.choice([0, 1]) for _ in range(4)]
    Snare_pattern = [random.choice([0, 1]) for _ in range(4)]
    Hihat_pattern = [1, 1, 1, 1] # hi-hat always plays
    For l in range(4):
        If kick_pattern[l]: kick.play()          If snare_pattern[l]:
            snare.play()
        If hihat_pattern[l]:
            hihat.play() Time.sleep(beat_duration)

```