# ■ Project Roadmap – Mood-Based Music Web App

## Step 1 – Setup the Web App

- Create a simple project folder (moodify/).
- Add index.html, style.css, and script.js.
- Make sure you can open it in a browser.
- Test: Page loads with 'Hello World'.

## Step 2 – Enable Camera Access

- Use JavaScript + tag to capture live video from your webcam.
- Code: navigator.mediaDevices.getUserMedia({ video: true }).
- Show the video stream on the page.
- Test: You should see yourself on the page.

## Step 3 – Add Emotion Detection (face-api.js)

- Import face-api.js via CDN.
- Load pretrained models (Face detection model, Face expression recognition model).
- Detect emotions in real time from webcam feed.
- Show detected emotion on the screen (like 'You look happy ■').
- Test: See live emotion updates.

## Step 4 – Connect Emotion → Song Mapping

- Create a playlist object in script.js.
- When emotion is detected, pick a matching song.
- Use HTML player to play the song.
- Test: If emotion = 'sad', it should play motivation.mp3.

## Step 5 – Build a Spotify-Like UI

- Use CSS to make it look like a music app.
- Navbar with app name (Moodify ■).
- Main area: live camera + detected mood.
- Bottom: Music player (song name, play/pause button, progress bar).
- Test: UI should look clean and responsive.

## Step 6 – Add Motivation Layer

- If 'sad' or 'angry' is detected → show a motivational quote along with music.
- Example: Sad ■ → 'Cheer up! Better days are coming ■'.
- Angry ■ → 'Breathe in, breathe out. Stay calm ■'.

## Step 7 – Polish & Improve

- Add multiple songs per mood (shuffle/random).
- Auto-refresh emotion detection (update mood if your face changes).
- Add a button to manually select mood if detection fails.

## Step 8 – (Optional, Advanced) Spotify API Integration

- Create a Spotify Developer account.
- Get API keys and use Spotify Web API.
- Instead of local MP3, play real Spotify playlists for each mood.