Lesson Plan — Mood2Emoji: Kid-Safe Text-Mood Detector

Created by Harsha C

Age Group: 12–16 years
Duration: 60 minutes
Tools: Django (Backend), React (Frontend)

Lesson Objectives:
1. Introduce students to text mood detection concepts.
2. Demonstrate how frontend (React) and backend (Django) interact.
3. Teach safe content filtering and kid-friendly design.
4. Show how to classify simple emotions using rule-based or TextBlob logic.

Topics Introduced:
1. Basics of sentiment analysis and mood detection.
2. Frontend-backend communication through APIs.
3. Django REST Framework for building APIs.
4. React components and input/output handling.
5. Safe word filtering and responsible app design.

60-Minute Lesson Flow:
1. 0–10 min — Introduction to mood detection and examples of emojis representing emotions.
2. 10–20 min — Explain how Django handles backend logic (TextBlob / rule-based approach).
3. 20–30 min — Demonstrate API creation in Django and response structure.
4. 30–45 min — Build a React interface for input and displaying mood emoji.
5. 45–55 min — Test the app with positive, neutral, and negative sentences.
6. 55–60 min — Recap key learning points and discuss responsible AI & safe content.

Activity:
Students will write short sentences expressing different emotions (e.g., 'I love pizza', 'I failed my test').
The app will detect the mood and display a matching emoji and message. Teachers can enable 'Teacher Mo
diagram of how text flows from frontend to backend.

Learning Outcomes:
1. Understand basic sentiment analysis concepts.
2. Learn how frontend and backend communicate via REST APIs.
3. Gain exposure to Python (Django) and JavaScript (React) integration.
4. Recognize the importance of safe and age-appropriate app design.
5. Build curiosity about ethical AI and responsible technology use.

End of Lesson Plan