Lesson Plan: Build a Mood2Emoji app
Age: 12–16 | Duration: 60 minutes

Learning goals:
- Understand how short pieces of text can be mapped to 'moods'.
- Learn about simple rule-based NLP and an introductory sentiment tool (TextBlob).
- Build and test a tiny web app with Streamlit.

Materials:
- Computer with internet and Python 3.9+
- Code editor (VS Code recommended)
- Project files (app.py, requirements.txt)

Timeline (60 minutes):
0–5 min: Introduction & objectives
5–15 min: Quick primer on text and emotion (what is sentiment?) + demo of app
15–30 min: Guided coding — walk through app.py (input, filter, analyzer, output)
30–45 min: Student activity — modify rule list / test edge cases / create 5 sentences
45–55 min: Share & discuss results, pitfalls, and ethics (why kid-safety matters)
55–60 min: Quick quiz & wrap-up (3 min reflection)

Topics introduced (in detail):
1. Text input & tokenization — Splitting text into words and basic cleaning.
2. Rule-based methods — Using word lists to decide mood.
3. Simple statistical sentiment — TextBlob polarity from -1 to +1.
4. Safety & filtering — Why we filter bad words, and how to handle unknown text safely.
5. Web apps with Streamlit — Simple UI components: text_input, checkbox, form, graphviz.

Activity instructions:
- Give students 5 example sentences each (mix of happy/neutral/sad/inappropriate).
- Let them run the app and collect results in a small table (emoji + explanation).
- Ask each group to find 2 sentences that the app mis-classified and suggest rule changes.

Learning outcomes (what students will be able to do):
- Explain the difference between rule-based and simple sentiment methods.
- Run a Streamlit app and test simple NLP logic.
- Reason about kid-safety and apply a basic filter.

Assessment:
- Observe students successfully running app (manual check).
- Small exit quiz: 3 questions about sentiment, rule-based limits, and safety.