🚀 Final Internship Project

NLP-Powered Sentiment Analysis API

📌 Objective

Build and deploy a **Neural Network-based Sentiment Classifier** that predicts whether a given text (e.g., customer review, tweet, or message) is **positive** or **negative**. Expose the trained model through a **REST API** to accept text input and return predictions.

🗀 Dataset Options

Choose one dataset (recommended: IMDb for simplicity):

- **IMDb Movie Reviews Dataset** – 50k labeled reviews (positive/negative)

(*IMDb dataset can be loaded directly via Keras:*)

```
from tensorflow.keras.datasets import imdb
```

🛠 Steps to Follow

1. Data Preprocessing

- Load dataset and split into training/testing sets.
- Tokenize text and convert to integer sequences.
- Pad sequences to fixed length (e.g., 200–300 tokens).

2. Model Building (Neural Network)

- Use an **Embedding layer** for word vector representations.
- Suggested structure:
  - Embedding → LSTM (or GRU / 1D CNN or Transformers) → Dense → Softmax (2 classes).
- Compile with:
  - Loss: `binary_crossentropy`
  - Optimizer: `adam`
  - Metrics: `accuracy`

3. Training & Evaluation

- Train the model on training data.
- Evaluate on test set.
- Report accuracy, F1-score, and confusion matrix.

4. Model Saving

- Save trained model (`sentiment_model.h5`).
- Save tokenizer (`tokenizer.pkl`).

5. API Development (Flask or FastAPI)

- Create endpoints:
  - `GET /health` → API status check.
  - `POST /predict` → Accepts JSON input like:
  - `{ "text": "I really loved this movie!" }`

    Returns:

    ```
    { "predicted_label": "positive", "confidence": 0.93 }
    ```

6. Testing the API

- Use **curl** or **Postman** to test requests.
- Provide example queries and responses.

📝 Deliverables

- Training notebook/script
- API implementation (`app.py`)

⏰ Deadline

Friday, 3rd October at 2:00 PM