**🧾 Pseudocode for Fake News Detection AI System**

**1. Load & Preprocess Dataset**

BEGIN

LOAD dataset from CSV file

COMBINE 'title' and 'text' columns into a single feature column called 'content'

FOR each article IN dataset:

CONVERT text to lowercase

REMOVE punctuation, numbers, special characters

REMOVE stopwords

OPTIONAL: Apply stemming or lemmatization

LABEL encode the 'label' column (FAKE = 0, REAL = 1)

SPLIT dataset into training set and test set (e.g., 80/20)

END

**2. Feature Extraction**

**a. Baseline model: TF-IDF**

BEGIN

INITIALIZE TF-IDF Vectorizer with max\_features = N

FIT TF-IDF on training content

TRANSFORM both training and test content into TF-IDF vectors

END

**b. Deep learning model: BERT**

BEGIN

INITIALIZE tokenizer (e.g., from BERT or custom tokenizer)

TOKENIZE text content in training and test sets

PAD sequences to max length

CREATE attention masks (if using transformers)

END

**3. Model Training**

**Option A: Logistic Regression (Baseline)**

BEGIN

INITIALIZE LogisticRegression classifier

TRAIN classifier on TF-IDF training vectors and labels

PREDICT labels on test set

END

**Option B: LSTM or CNN (Deep Learning)**

BEGIN

DEFINE LSTM model architecture:

- Embedding Layer

- LSTM Layer

- Dense Output Layer (Sigmoid/Softmax)

COMPILE model with binary cross-entropy loss and Adam optimizer

TRAIN model on tokenized sequences and labels

PREDICT labels on test set

END

**Option C: BERT or Transformer Model**

BEGIN

LOAD pre-trained BERT model with classification head

PREPARE input\_ids and attention\_masks from tokenizer

FINE-TUNE BERT model on training set

PREDICT probabilities or labels on test set

END

**4. Evaluation (classification score)**

BEGIN

CALCULATE metrics: Accuracy, Precision, Recall, F1-score

DISPLAY confusion matrix

PLOT training and validation loss/accuracy (if applicable)

ANALYZE false positives and false negatives

END

**5. Chatbot / UI Integration**

BEGIN

BUILD a user interface (e.g., Streamlit or Flask)

ALLOW user to input a news article or headline

PREPROCESS and TOKENIZE the input

PASS the input to the trained model

DISPLAY the predicted result: "FAKE" or "REAL"

END