# NewsGuard AI

# User Guide

This document provides instructions for both end-users and developers. Part 1 explains how to use the NewsGuard AI application. Part 2 provides technical instructions for developers or markers to replicate the project's results and run the system locally.

## 

## Part 1: How to access the system

Welcome to NewsGuard AI. This section provides a straightforward guide on how to use the interactive web application to analyze the authenticity of a news article. Follow these simple steps to input an article's title and text, initiate the analysis, and interpret the results, which include the classification, a confidence score, and detailed AI-powered explanations.

Note: The user guide includes commands for both Windows and macOS. However, as both team members operated exclusively on Windows, the macOS commands provided have been thoroughly researched but not practically tested.

1. Download the provided ZIP folder and extract all files into a single directory. The zip folder contains:

* main\_app.py
* requirements.txt
* download\_nltk.py
* 2.13.0
* calibration\_models.pkl
* complete\_fake\_news\_model.h5
* isotonic\_calibrator.pkl
* newsguard.log
* tokenizer.pkl
* utils.py
* The .streamlit directory (contains the API key)

1. Install Python version 3.11 to maintain compatibility with the system’s dependencies. Open Command Prompt and execute the command: “**winget install python.python 3.11**”. On MacOS Terminal execute the command “brew install python@3.11”

A computer screen shot of a program

AI-generated content may be incorrect.

1. Install all required Python libraries listed in requirements.txt by running the following command inside the directory containing the extracted files: **“py -3.11 -m pip install -r requirements.txt”.** For MacOS execute “python3.11 -m pip install -r requirements.txt”

A screenshot of a computer screen

AI-generated content may be incorrect.

1. Next, download the necessary NLTK tokenizer data package punkt\_tab by running the provided script:**“py -3.11 download\_nltk.py”.** For MacOS execute “python3.11 download\_nltk.py”

A black screen with white text

AI-generated content may be incorrect.

1. Launch the Streamlit application with the command: **“py -3.11 -m streamlit run main\_app.py”**. This will open the AI system interface in your default web browser. For MacOS Terminal, use the command “python3.11 -m streamlit run main\_app.py”

A screen shot of a computer

AI-generated content may be incorrect.

1. Be aware that the first time you launch the system, it may take a few minutes to load all models and initialize due to the size and complexity of the AI components. Subsequent launches will be faster as the data is cached locally.
2. **Access the Application**: Once you complete step 5, the Streamlit web interface should automatically open. If not, open the local URL (e.g., <http://localhost:8501>) provided after running the application as described in step 5.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Enter Article Details**:
   1. In the Article Title field, type or paste the headline of the news article you wish to analyze.
   2. In the Article Text area, paste the full body content of the article. The system is optimized for English-language articles with at least 10 words.
2. **Analyze**: Click the **"Analyze Article"** button to begin the process. You will see a progress bar and status updates as the system works.  
   A screenshot of a video

   AI-generated content may be incorrect.
3. **Review the Classification**:
   1. The system will display a prominent card with the classification result:

REAL, FAKE, or UNCERTAIN

* 1. The card also shows the model's Confidence percentage for its prediction.

A red rectangular sign with white text

AI-generated content may be incorrect.

1. **Explore the Analysis Tabs**:
   1. **🤖 AI Explanation**: This tab provides insight into the model's decision-making process using LIME.
      1. **Feature Importance Chart**: A bar chart shows the top words that influenced the prediction, colored green for "REAL" and red for "FAKE".

A screen shot of a graph

AI-generated content may be incorrect.

* + 1. **Highlighted Text**: The original article text is displayed with these influential words highlighted, giving a clear, visual explanation of the AI's reasoning

A screenshot of a computer

AI-generated content may be incorrect.

* 1. **📰 Verification Sources**: This tab helps with external fact-checking.
     1. If an article is classified as ‘REAL’, the system extracts key topics and searches for related articles from credible sources using the NewsAPI

A screenshot of a computer

AI-generated content may be incorrect.

* + 1. Each found article is presented in a card with an AI-generated summary, the source, publication date, and a link to the original article.
    2. If the article is classified as ‘FAKE’, this section will advise that no search was performed to avoid amplifying misinformation.

A screenshot of a computer

AI-generated content may be incorrect.

## Part 2: Replicating the Project for Verification

This section provides technical instructions for a developer or marker to set up the project environment, replicate the model development process, and run the final web application locally. Following these steps will allow for the verification of all results claimed in the project report

**1. Regenerating Model Artifacts and Results (Google Colab):**

This process uses the provided Google Colab notebook to perform data analysis, preprocessing, model training, and evaluation, which generates the key artifacts and results presented in the report.

**Prerequisites:**

* A Google Account to use Google Colab.
* The GoogleColab.ipynb file from the project submission.
* The WELFake\_Dataset.csv file. This can be downloaded from Kaggle: [WELFake Dataset.](https://www.kaggle.com/datasets/saurabhshahane/fake-news-classification)

**Steps:**

1. **Open Google Colab:**
   1. Navigate to [colab.research.google.com.](https://colab.research.google.com/)
   2. Go to File > Upload notebook and select the ICT304\_Assignment1\_Main\_LSTM\_Final.ipynb file.
2. **Enable GPU Acceleration**:
   1. In the Colab menu, navigate to **Runtime > Change runtime type**.
   2. Select **T4 GPU** from the "Hardware accelerator" dropdown menu and click **Save**. This is crucial for training the CuDNNLSTM model efficiently

A screenshot of a computer

AI-generated content may be incorrect.

1. **Upload the Dataset**:
   1. In the left-hand sidebar, click the Files icon.
   2. Click the Upload to session storage button and select the WELFake\_Dataset.csv file you downloaded. Wait for the upload to complete.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Execute the Notebook:**
   1. To regenerate all results from scratch, go to the menu and select Runtime > Run all.
   2. The notebook is structured sequentially and will perform the following key stages:
      1. **EDA and Preprocessing**: Cleans the data, performs exploratory data analysis, and prepares the text for the model.
      2. **Model Training**: Uses Bayesian optimization to find the best hyperparameters and then trains the final CuDNNLSTM model.
      3. **Model Evaluation**: Evaluates the trained model on the test set and generates the performance metrics (Accuracy, Precision, Recall, F1-Score, AUC) and visualizations (Confusion Matrix, ROC Curve) presented in the report.
      4. **Probability Calibration**: Analyzes and applies Isotonic Regression to calibrate the model's confidence scores, generating the reliability diagrams
      5. **LIME Analysis**: Conducts an analysis on a sample of the data to demonstrate model interpretability and check for biases.
   3. **Verify Outputs**:
      1. As the cells run, the notebook will output the same tables, charts, and metrics found in the project report
      2. Upon completion, the notebook will save the necessary files to the Colab environment's session storage. You can download them by right-clicking on each file in the Files sidebar and selecting Download.
      3. **Key files generated**:
         1. **complete\_fake\_news\_model.h5**: This HDF5 file stores the fully trained Bidirectional CuDNNLSTM neural network, containing the architecture, weights, and configuration required to classify news articles.
         2. **tokenizer.pkl**: This pickle file contains the trained tokenizer object that converts raw text into numerical sequences (tokens) that the neural network can process, based on a vocabulary learned from the training data.
         3. **isotonic\_calibrator.pkl**: This pickle file saves the fitted Isotonic Regression model, which adjusts the raw prediction probabilities from the main model to provide more accurate and reliable confidence scores.
      4. These are required for running the local web application.

**2. Running the Final Application Locally (Streamlit)**

This section details how to run the interactive web application on a local machine using the pre-trained model artifacts.

**Prerequisites:**

* Python version 3.11 is required for compatibility with the project's dependencies. Using any other python version will lead to errors while accessing libraries like tensorflow.

**Required Files:**

Ensure the following files from the project submission are in a single directory

* main\_app.py
* utils.py
* requirements.txt
* download\_nltk.py
* complete\_fake\_news\_model.h5 (or the one you generated)
* tokenizer.pkl (or the one you generated)
* isotonic\_calibrator.pkl (or the one you generated)
* The .streamlit directory (contains the API key)

**Setup and Execution Steps:**

1. **Install Python 3.11**: If you do not have Python 3.11, install it. On Windows, you can use the following command in the Command Prompt: “**winget install python.python.3.11”**

A computer screen with white text

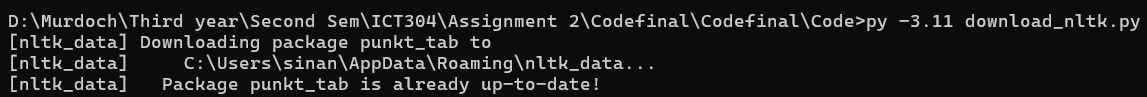
AI-generated content may be incorrect.

1. **Install Dependencies**: Navigate to the project directory in your terminal or Command Prompt and run the following command to install all required libraries from the ‘requirements.txt’ file: **“py -3.11 -m pip install -r requirements.txt”**

A screenshot of a computer program

AI-generated content may be incorrect.

1. **Download NLTK Data**: The application requires the 'punkt' data package from NLTK. Run the provided helper script to download it: **“py -3.11 download\_nltk.py”**



1. **Set Up API Key (Required for Verification Feature)**:
   1. The application uses the NewsAPI to fetch related articles for verification.
   2. **Locate the Secrets File**: In the project submission folder, navigate to the .streamlit directory and find the secrets.toml file.
   3. **Confirm the Content**: This file has been pre-configured for you with the necessary API key in the following format:

**NEWSAPI\_KEY = "YOUR\_API\_KEY\_HERE"**

* 1. **Note:** The application will run without this file, but the "Verification Sources" feature will be disabled. The core fake news classification and AI explanation will still function correctly.

1. **Launch the Application:** Run the following command from the project directory to start the Streamlit server: **“py -3.11 -m streamlit run main\_app.py”**
2. **Access the System:** The command will output a Local URL (e.g., http://localhost:8501). Open this URL in your web browser to interact with the NewsGuard AI system. Note that the initial launch may take a few moments to load the AI models into memory