

INTERNSHIP REPORT

Abstract

This project presents WriteWise, an AI-powered essay and assignment feedback tool built using Streamlit, BERT (CoLA), and T5-small. The system analyzes student writing, evaluates grammatical acceptability, generates improvement suggestions, and provides an automated grade based on readability, vocabulary complexity, and structural features. The project demonstrates how modern Natural Language Processing (NLP) models can be used to support academic writing enhancement and assist learners with real-time feedback.

Internship Objective

- To understand and implement fundamental NLP workflows using pre-trained models.
- To build an interactive Streamlit web application for text analysis.
- To apply AI techniques for grammar evaluation, suggestion generation, and readability scoring.
- To gain hands-on experience in Python, transformers, and model deployment basics.
- To complete a guided virtual internship project and develop a usable academic tool.

Internship Activities

During this internship, the following technical tasks were completed:

1. Studying NLP basics: tokenization, transformers, acceptability classification, summarization.
2. Learning model loading and inference using HuggingFace's `transformers`

library.

3. Designing the Streamlit UI for user interaction.
4. Integrating BERT CoLA for grammar/style acceptability checking.
5. Integrating T5-small for text improvement generation.
6. Implementing readability, vocabulary, and structural checks.
7. Creating an automated grading rubric.
8. Testing the application with multiple sample essays.
9. Documenting results and preparing the final report and demo.

Introduction

Academic writing evaluation typically requires manual review, which is time-consuming and subjective.

This project introduces *WriteWise*, an AI-based assistant that automates:

- grammar analysis
- clarity evaluation
- paraphrased suggestions
- readability scoring
- vocabulary complexity measurement
- overall grading

Using pre-trained models (BERT CoLA and T5-small), the system provides instant feedback, making it useful for students, educators, and early learners.

Project Analysis

1 Problem Statement

Students often struggle with grammar, structure, and clarity in academic writing. Manual feedback is slow. A fast, automated system is needed.

2 Proposed Solution

Use transformer-based models to analyze text and generate structured feedback:

- BERT CoLA → Acceptability & basic grammar inference
- T5-small → Suggest improvements/paraphrases
- textstat → Readability scoring
- Custom logic → Vocabulary analysis & grading

3 System Workflow

User → Streamlit UI → Text Input → NLP Models →
Analysis + Suggestion + Grade → Output back to UI

Technologies Used

Programming

- Python 3.10
- Streamlit UI

NLP Models

- BERT: `textattack/bert-base-uncased-CoLA`
- T5-small for text generation

Libraries

- `transformers`
- `torch`
- `textstat`
- `sentencepiece`
- `regex`

Tools

- VS Code
- Git & GitHub for version control

Code Implementation

1 Model Loading Code:

This section loads the pre-trained NLP models from Hugging Face.

- **BERT CoLA** → Checks grammar/acceptability
- **T5-small** → Generates improvement suggestions

```
def load_models():
    models = {}

    # BERT CoLA for Grammar Acceptability
    try:
        bert_tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
        bert_model = AutoModelForSequenceClassification.from_pretrained("textattack/bert-base-uncased-CoLA")
        bert_pipeline = pipeline("text-classification", model=bert_model, tokenizer=bert_tokenizer)
        models['bert_pipeline'] = bert_pipeline
    except Exception as e:
        models['bert_pipeline'] = None
        models['bert_error'] = str(e)

    # T5-small for text improvement
    try:
        t5_tokenizer = AutoTokenizer.from_pretrained("t5-small")
        t5_model = AutoModelForSeq2SeqLM.from_pretrained("t5-small")
        t5_pipeline = pipeline("text2text-generation", model=t5_model, tokenizer=t5_tokenizer)
        models['t5_pipeline'] = t5_pipeline
    except Exception as e:
        models['t5_pipeline'] = None
        models['t5_error'] = str(e)

    return models

models = load_models()
```

Purpose:

Downloads and initializes the models used for grammar evaluation and suggestions.

2 Analysis Function:

Uses BERT to evaluate whether each sentence is grammatically acceptable.

```
def analyze_text(text):
    bert_pipe = models.get('bert_pipeline')
    if bert_pipe is None:
        return None, models.get('bert_error', 'BERT pipeline not available.')

    sentences = split_sentences(text)
    results = []
    for sent in sentences:
        try:
            result = bert_pipe(sent[:512])[0]
        except Exception as e:
            result = {"label": "ERROR", "score": 0.0, "error": str(e)}
        results.append({"sentence": sent, "result": result})
    return results, None
```

Purpose:

Returns **LABEL_0** (acceptable) or **LABEL_1** (grammatical issue) with a confidence score.

3 Suggestion Function:

Uses T5 to paraphrase and improve the essay.

```
def suggest_improvements(text, num_return_sequences=1):
    t5_pipe = models.get('t5_pipeline')
    if t5_pipe is None:
        return None, models.get('t5_error', 'T5 pipeline not available.')

    prompt = f"improve: {text}"
    try:
        outs = t5_pipe(prompt, max_length=512, num_return_sequences=num_return_sequences, num_beams=5)
    except Exception as e:
        return None, str(e)

    suggestions = [o.get('generated_text', '') for o in outs]
    return suggestions, None
```

Purpose:

Creates a rewritten, cleaner version of the given text

4 Streamlit UI Structure:

```
text_input = st.text_area(
    "Enter your essay or assignment text here:",
    key="essay_text"
)

if st.button("Get Feedback"):
    if text_input.strip():

        with st.spinner("Analyzing text..."):
            analysis, analysis_error = analyze_text(text_input)
            suggestions, suggestion_error = suggest_improvements(text_input)
            grade_info = grade_text(text_input)

            if grade_info["num_paragraphs"] <= 1:
                st.write("Insufficient paragraph structure.\n\n"
                    "[Paragraph structure refers to the way paragraphs are organized within the text.]")

            st.subheader("Analysis:")
            if analysis is None:
                st.error(f"Error loading BERT model: {analysis_error}")
            else:
                for item in analysis:
                    result = item["result"]
                    label = result.get("label", "")
                    score = result.get("score", "")
                    st.write(
                        f"Text: {label}, Confidence: {score}\n\n"
                        "[LABEL_1 = grammatically incorrect]\n\n"
                        "[LABEL_0 = acceptable sentence]\n\n"
                        "[Confidence Score shows model certainty]"
                    )

            st.subheader("Suggestions for Improvement:")
            if suggestions is None:
                st.error(f"Error loading T5 model: {suggestion_error}")
            else:
                st.write(suggestions[0])

            st.subheader("Grade:")
            st.write(f"Your grade is: {grade_info['grade']}")

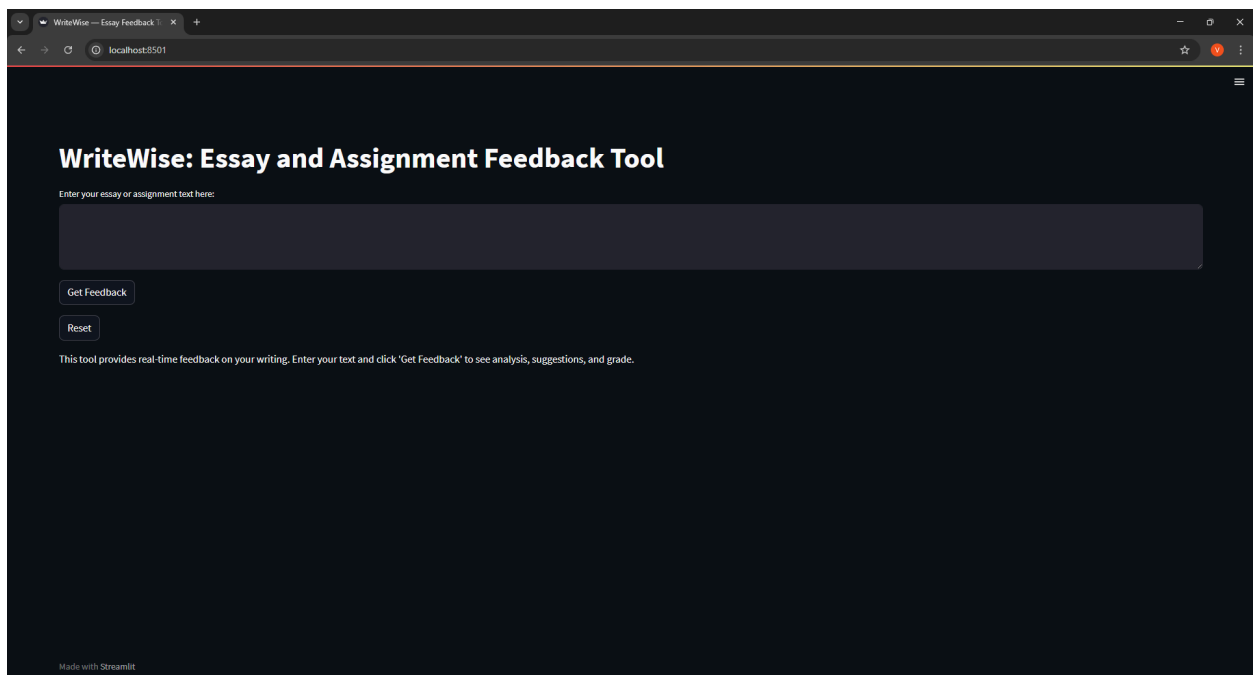
    else:
        st.warning("Please enter some text to get feedback.")
```

Renders the web interface where users input text and view results.

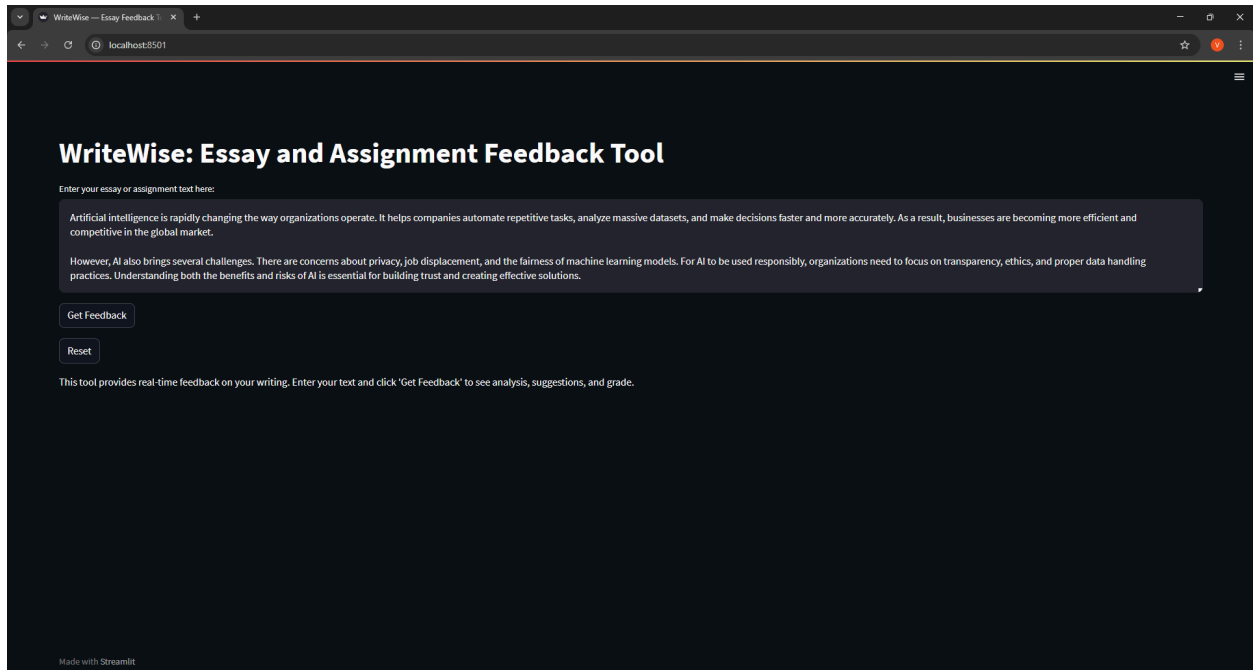
Purpose: Connects all backend functions to an interactive frontend.

Screenshots

Homepage Screenshot

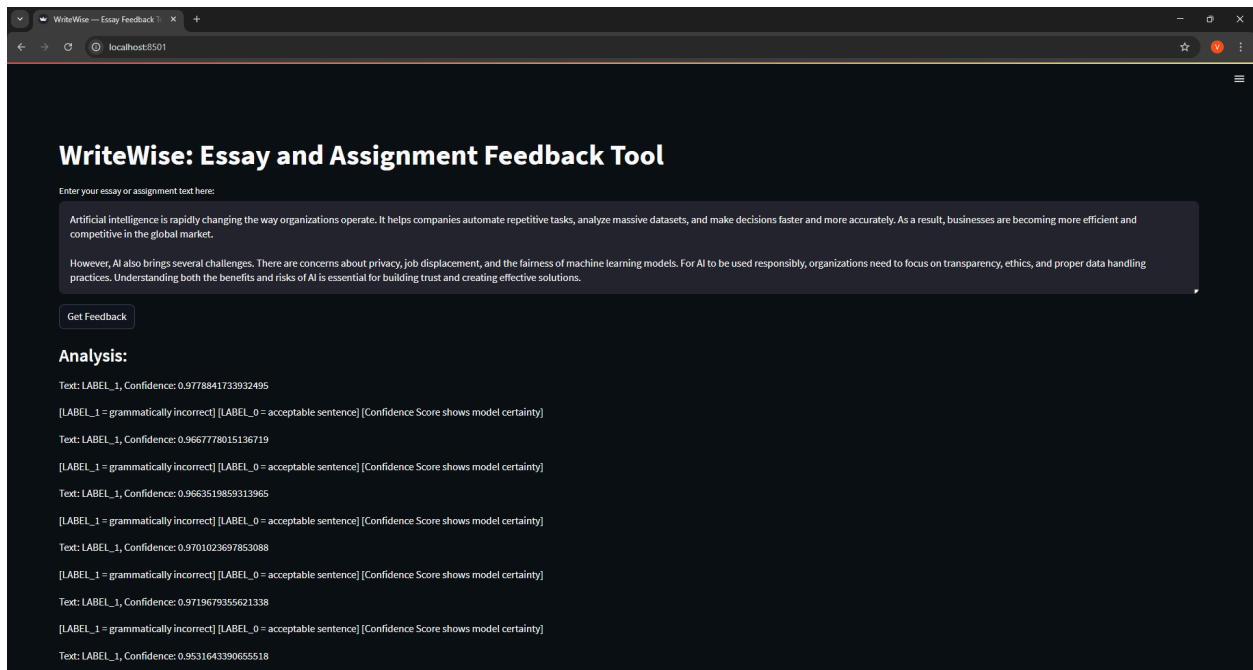


Input Sample Screenshot

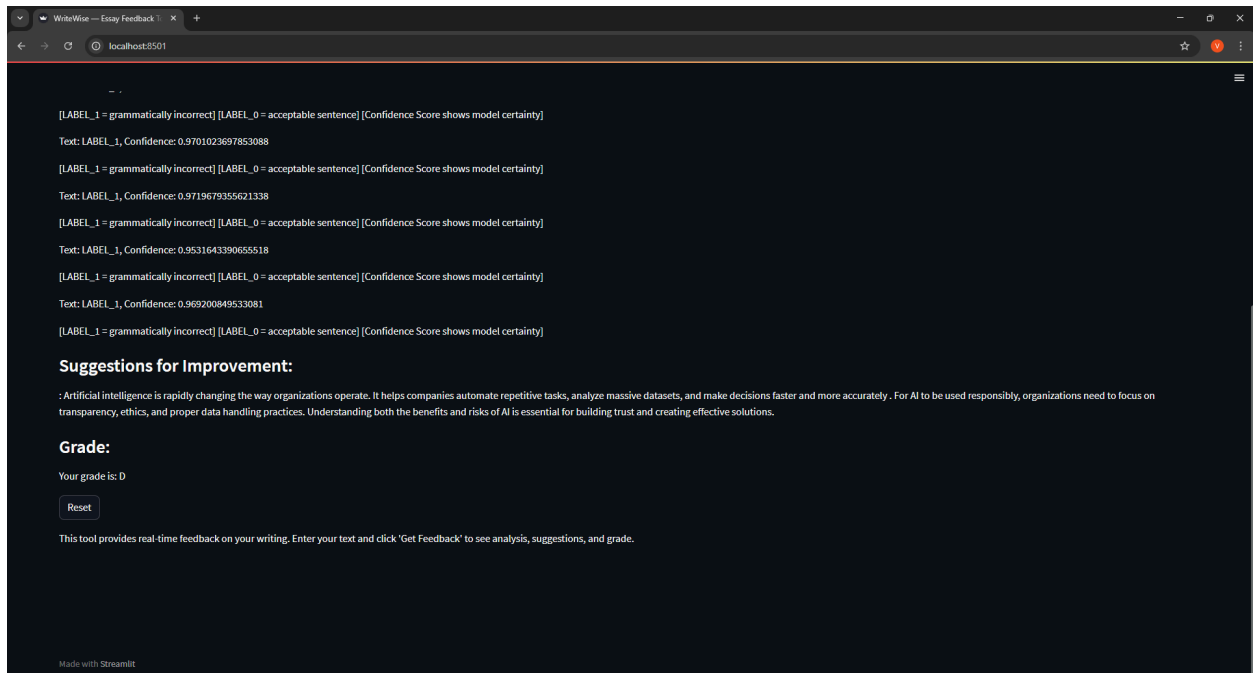


Output Full :

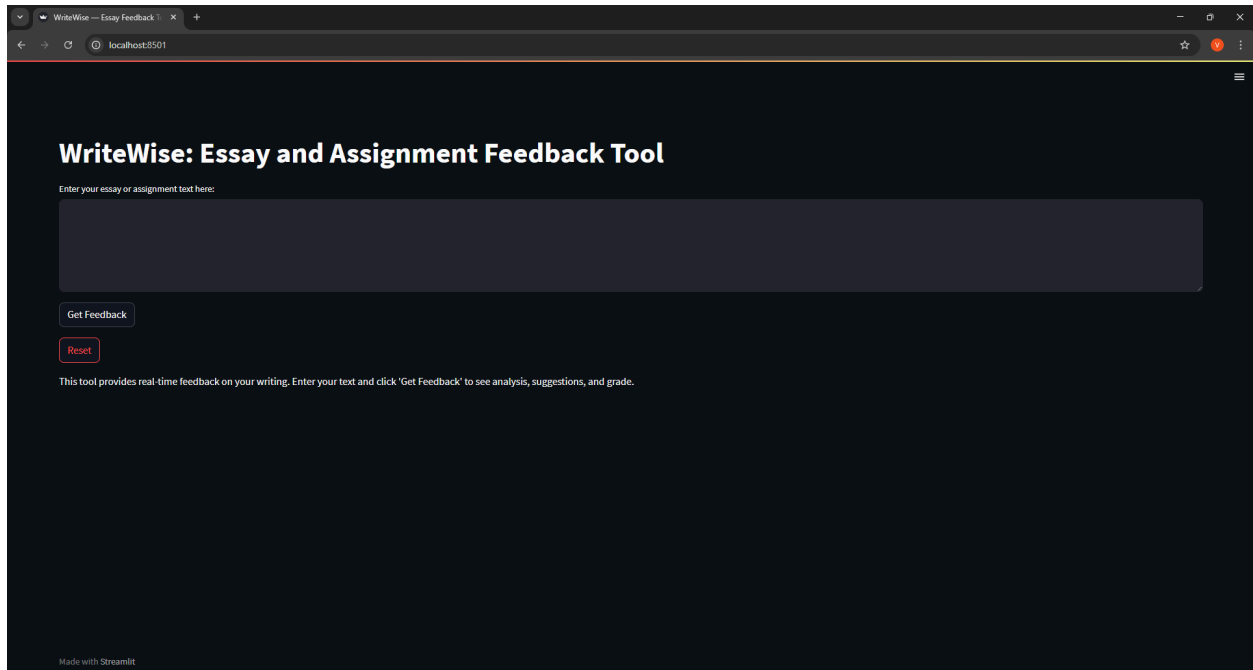
Part 1:



Part 2:



Reset Screen



Conclusion

The WriteWise application successfully demonstrates how AI and NLP can improve academic writing by automating grammar evaluation, readability analysis, and suggestion generation. Through this internship, I gained practical experience in Python development, transformer models, and user-interface integration. The project outcomes align with the internship objectives, resulting in a functional and user-friendly writing assistant that can support students in enhancing the quality of their assignments.

Author: Veerannavaru Vaijinath

SBID: SWUID20250267526

Project: WriteWise - Essay Feedback Tool

Platform: SmartInternz