

# Shakespeare Q&A System: Bringing the Bard to Life

Explore an innovative Natural Language Processing project that transforms Shakespeare's complete works into an interactive question-answering system. This presentation covers data preprocessing, model fine-tuning, and deployment using cutting-edge NLP techniques.



**Shubham Gaur**



# Data Preprocessing: From PDF to Structured Text

1

## Extract Text

Use PyMuPDF to extract raw text from Shakespeare's complete works PDF.

2

## Clean Text

Remove headers, footers, and non-ASCII characters using regular expressions.

3

## Segment Text

Divide text into Acts and Scenes for structured access.







# Model Selection and Fine-Tuning

## Model Choice

Selected distilbert-base-uncased-distilled-squad for Question Answering tasks.

## Data Preparation

Created dataset using Acts and Scenes as contexts with generated questions.

## Fine-Tuning

Used Hugging Face's Trainer API with early stopping and evaluation strategies.

# System Design: Building the Q&A Pipeline

- 1** Load Model  
Load fine-tuned model and tokenizer from saved directory.
- 2** Create Pipeline  
Use Hugging Face's pipeline API for Question Answering.
- 3** Process Input  
Take user question and selected text as inputs.
- 4** Generate Answer  
Return answer based on the model's understanding of the context.





# Asking Questions: Interacting with Shakespeare



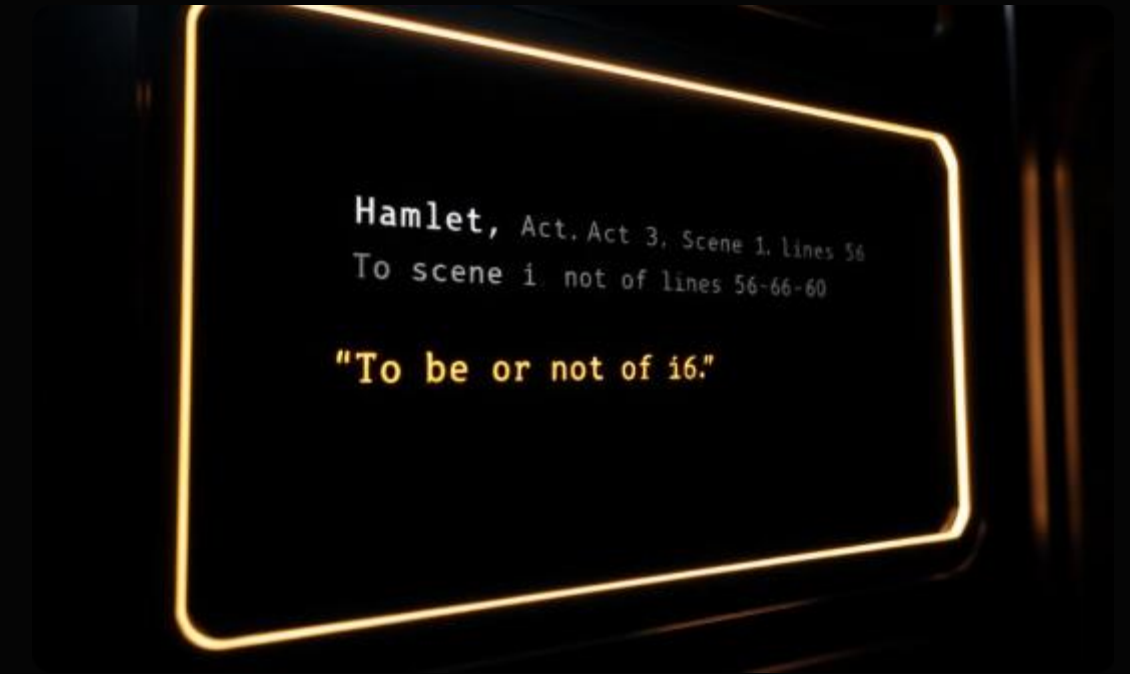
## Function Design

ask\_question function interfaces with the Q&A pipeline.



## Input

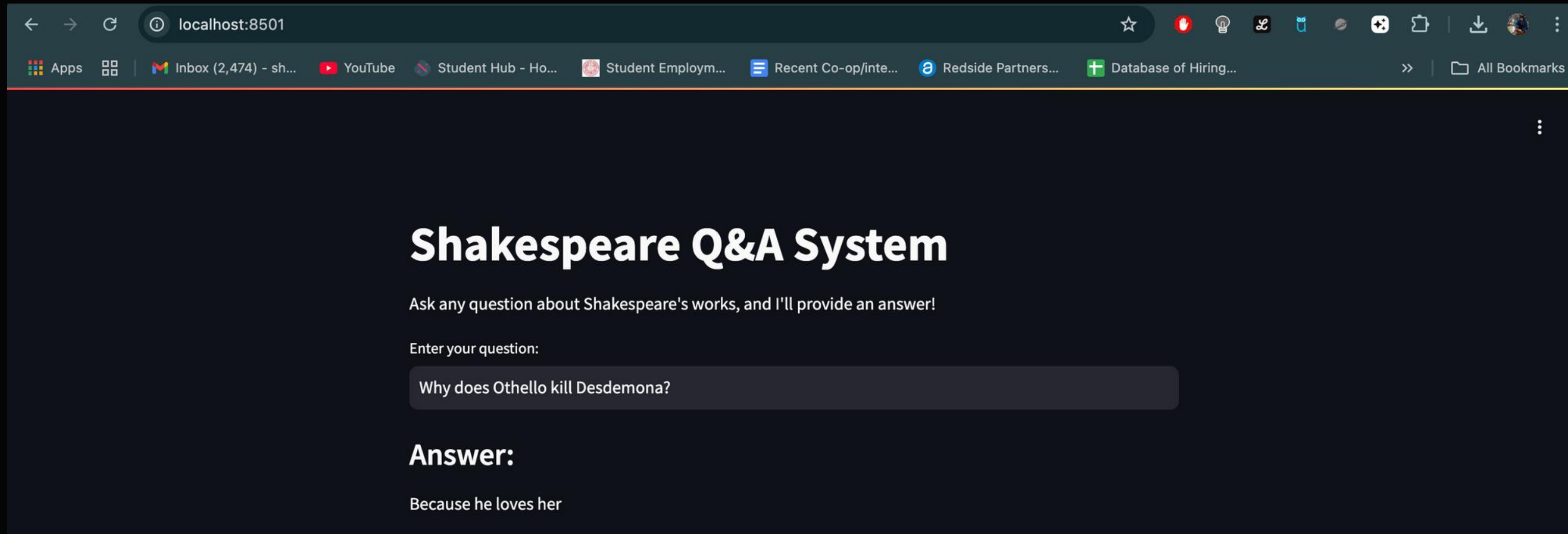
User question & scene text are inputted.



## Output

Relevant answer from Shakespeare's text.

# Web-Based Interface: Streamlit Integration



**1** Install Streamlit  
Quickly set up Streamlit using the command: ``pip install streamlit``.

**2** Design the UI  
Create a user-friendly interface with intuitive dropdown menus and input fields.

**3** Integrate the Q&A System  
Connect the Streamlit frontend to the Q&A backend for seamless interaction.

**4** Deployment  
Launch the app locally or deploy it to Streamlit Cloud for wider access.



# Setup and Execution: Getting Started



## Python Setup

Install Python 3.8+ and required libraries.



## Clone Repository

Get the code from GitHub.

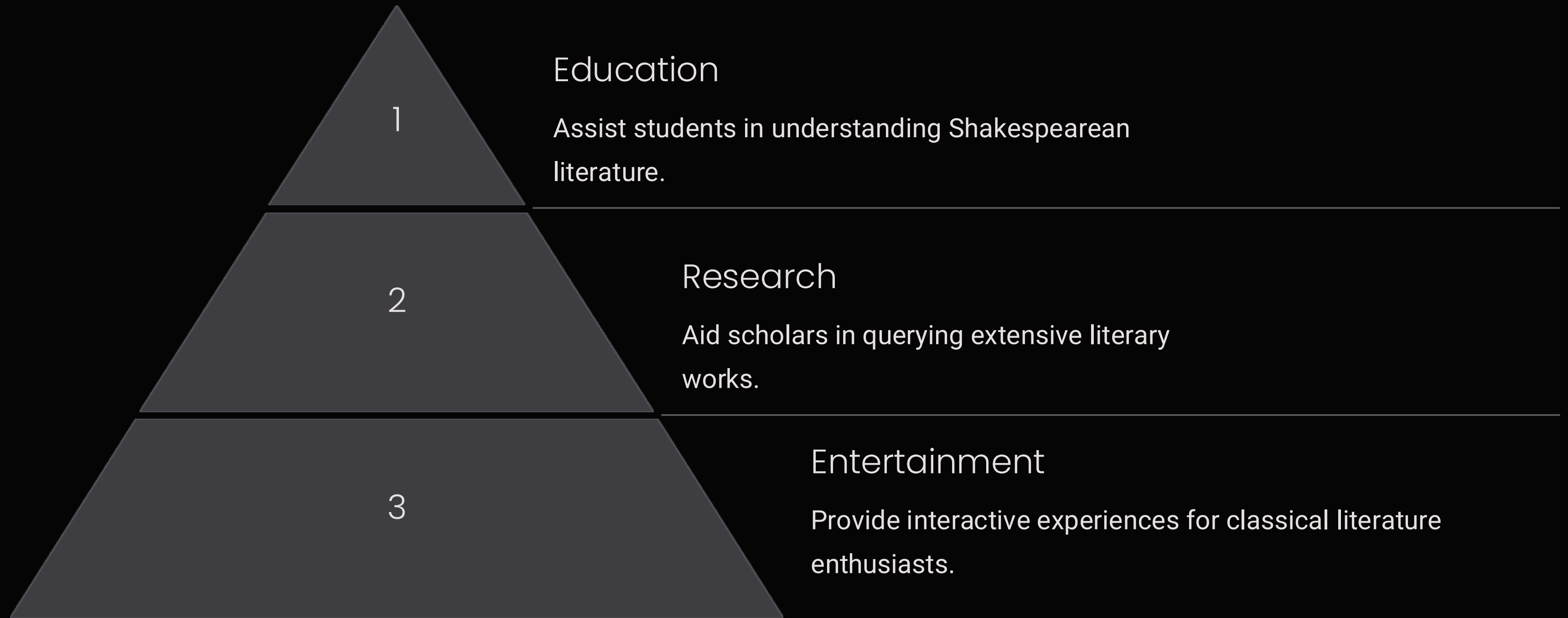


## Run Application

Execute `streamlit run qa_app.py` in terminal.



# Business Insights: Applications and Potential





# Conclusion: Bringing Shakespeare to the Digital Age

1M+

Words Processed

Entire Shakespearean corpus analyzed and made interactive.

24/7

Availability

Continuous access to Shakespearean knowledge through the Q&A system.

∞

Potential Applications

Limitless possibilities for education, research, and entertainment.

