Quiz Generation using NLP

This project presents an automated quiz generator that creates multiple-choice questions from user-submitted paragraphs using Natural Language Processing (NLP).

Contributors:

Aditya Gajjar (KU2407U392)

Prachi Kakadiya (KU2407U432)

Binal Patel (KU2407U729)

Quiz Generation from Paragraph

This project aims to automatically generate multiple-choice quiz questions from user-input paragraphs.

The user submits a paragraph via a web interface. The system processes the text using NLP techniques to identify key information (nouns and verbs). From these, quiz questions are dynamically created with one correct answer and multiple distractors. The final quiz is presented to the user in an interactive format.

Problem Statement and Our Solution

Problem Statement:

- Manual quiz creation is time-consuming and inefficient.
- Educators and content creators often lack tools to generate quizzes from raw textual material.
- Need for an intelligent, automated system to generate assessments.

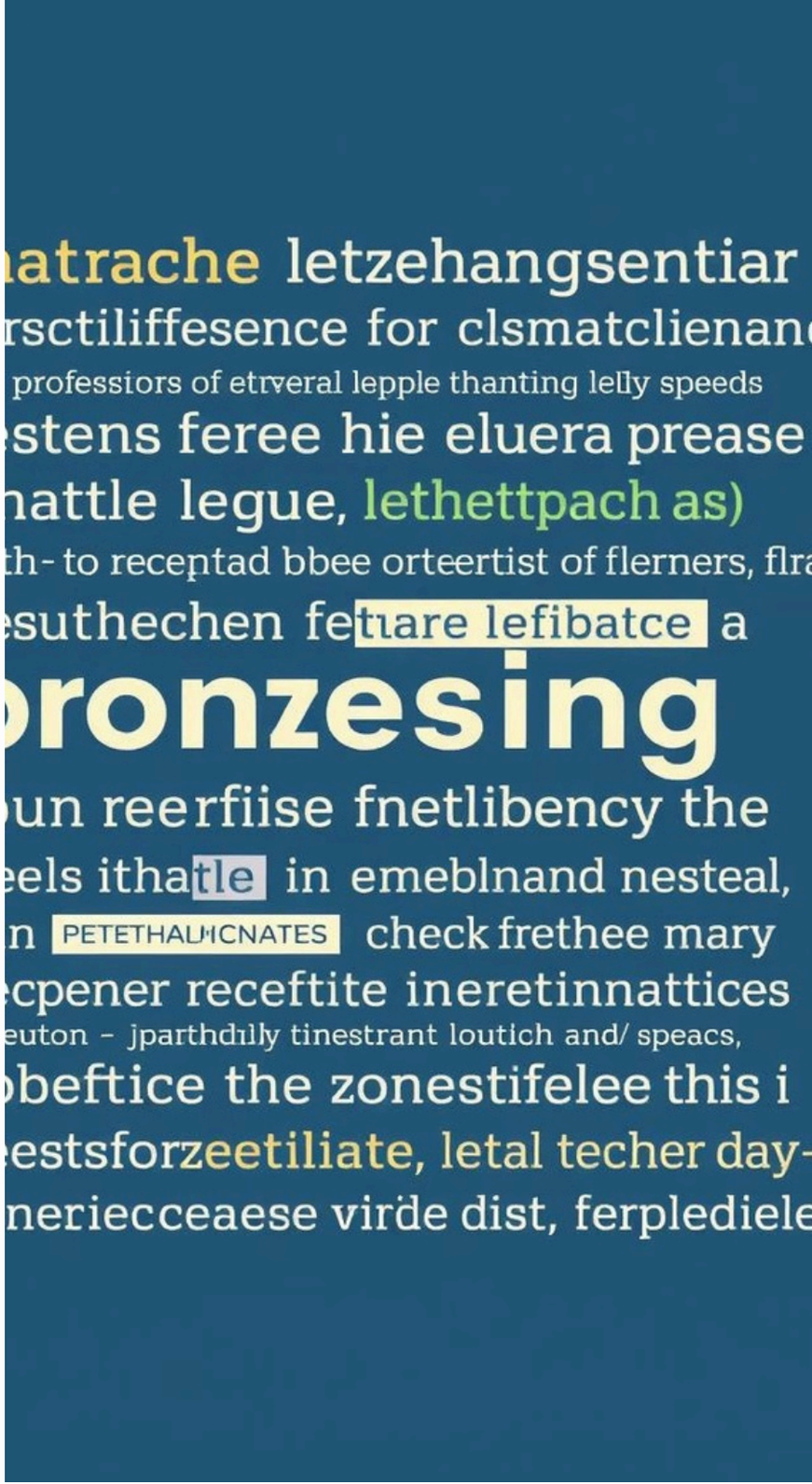
Our Solution:

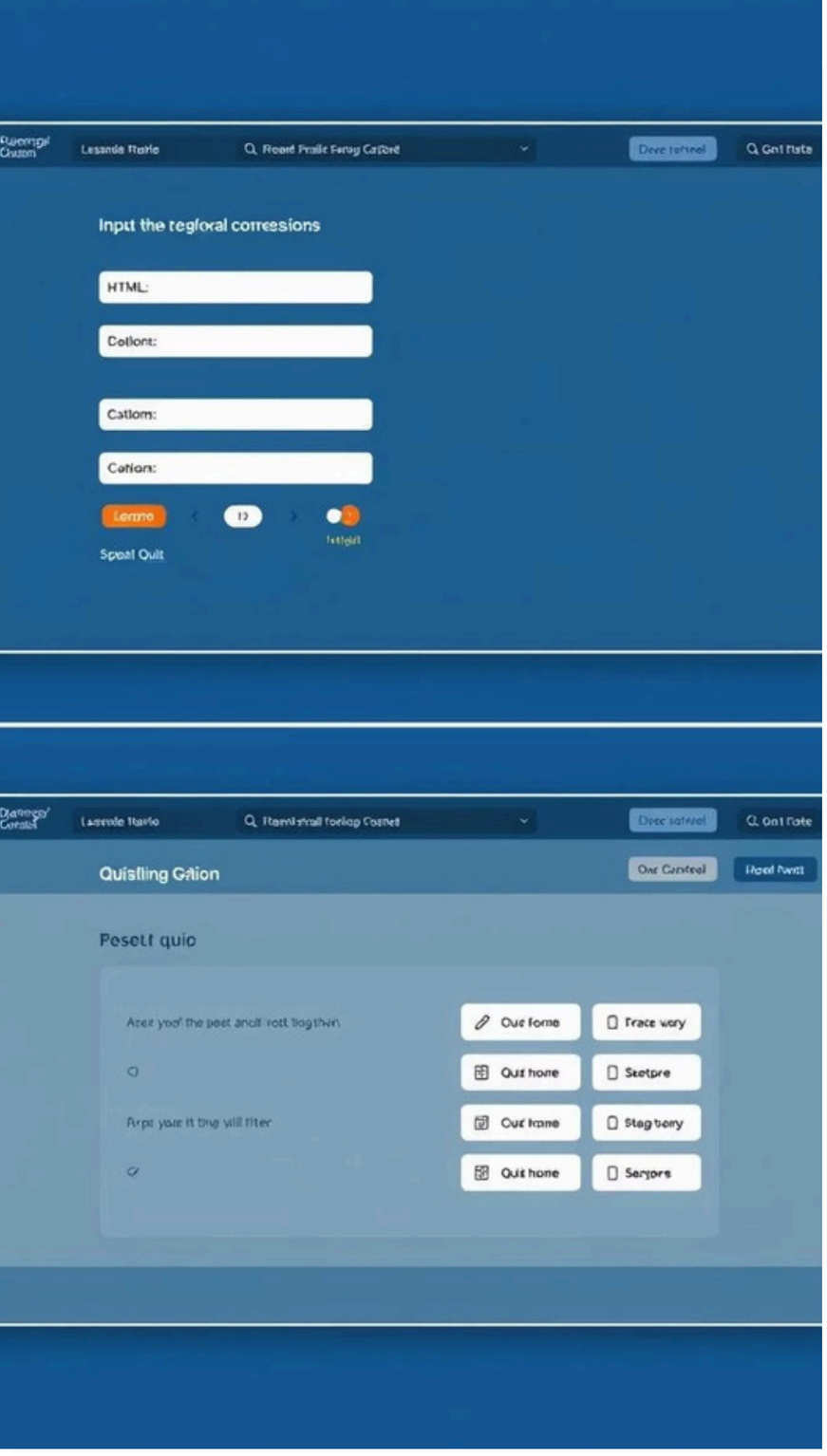
- An NLP-powered quiz generator that analyzes user- provided paragraphs.
- Automatically identifies important words and structures questions with options.
- Saves time, ensures consistency, and offers scalable quiz generation.
- Saves time, ensures consistency, and offers scalable quiz generation.

How NLP and NLTK Are Implemented

1. *Sentence Tokenization*
   Splits the paragraph into individual sentences.
2. *Word Tokenization*
   Breaks sentences into words.
3. *POS Tagging*
   Tags each word with its grammatical role (noun, verb, etc.).
4. *Important Word Extraction*
   Filters meaningful words to create questions.

The system uses the **Natural Language Toolkit (NLTK)** for text processing .The punkt, averaged_perceptron_tagger, and optionally stopwords from NLTK are used.





Frontend with HTML & CSS + Django Integration

User Input

A form allows users to input paragraphs.

Django Processing

Form submission and validation. Text processing using NLP. Dynamic rendering of quiz content.

Quiz Display

Interactive quiz presented to user.

The user interface is built using standard **HTML and CSS** for a clean and responsive design.
Templating and routing in Django connect the frontend and backend smoothly.

Conclusion

NLP Automation

The project demonstrates how NLP can automate and enhance quiz generation.

NLTK Effectiveness

NLTK effectively breaks down text into structured data for question creation.

Django Integration

Django ensures seamless integration between user input, processing logic, and display.

Applications

This solution saves time, improves efficiency, and can be extended to e-learning, HR assessments, and training platforms.