**Question and Multiple-Choice Question (MCQ) Generation from Text using Flask Framework**

**Introduction**:

This project aims to develop a web application using Flask that generates multiple-choice questions (MCQs) from text input provided by the user. The application utilizes natural language processing (NLP) techniques to extract meaningful information from the text and formulate questions along with answer choices. Users can either upload a PDF or TXT file containing the text or more than one file. The generated questions with corresponding MCQs are then displayed to the user, allowing them to test their understanding of the text.

**Features**:

* **Input Options**: Users can upload a PDF or TXT file containing the text. File can be uploaded more than one.
* **Question** **Generation**: The application processes the text using spaCy to extract sentences and identify nouns, which are used to generate Questions with MCQs.
* **MCQ** **Formatting**: Each generated MCQ includes a question stem, answer choices, and the correct answer.
* **Random** **Selection**: The application randomly selects sentences from the text to generate a diverse set of questions.
* **Answer Validation:** Users can select their answers for each MCQ, and the correct answer is displayed for validation.

**Technologies Used:**

* **Flask**: Python web framework used for building the web application.
* **Bootstrap**: Front-end framework for designing responsive web pages.
* **spaCy**: Python library for NLP tasks such as tokenization, part-of-speech tagging, and named entity recognition (NER).
* **PyPDF2**: Python library for reading and extracting text from PDF files.

**Project Structure:**

* **app.py:** Flask application script containing the main logic for processing text, generating MCQs, and handling HTTP requests.
* **index.html:** HTML template for the home page where users can upload files or manually enter text.
* **mcqs.html:** HTML template for displaying the generated MCQs to the user.

**Workflow**:

* User visits the web application.
* User uploads a PDF or TXT file containing the text.
* The application processes the text and generates MCQs based on the input text.
* Generated MCQs are displayed to the user, who can select their answers for each question.
* The correct answer for each MCQ is displayed for validation.

**Conclusion**:

The Flask-based web application provides a user-friendly interface for generating MCQs from text input. It leverages NLP techniques to automatically formulate questions, making it a valuable tool for educators, students, and anyone interested in testing their comprehension of textual content. Additionally, the project demonstrates the integration of various technologies to create a functional and interactive web application.