The Superior University, Lahore

Assignment-I (Fall 2023)

Course Title:	Programming for AI				Course Code:	CAI601410	Credit Hours: 4
Instructor:	Prof. Rasikh Ali				Programme Name:	BSDS	
Semester:	4th	Batch:	F23	Section:	BSDSM-4A	Date:	3rd May, 2025
Time Allowed:					Maximum Marks:		
Student's Name:	Asaad Iqbal				Reg. No.	SU92-BSDSM-F23-020	

Lab Project
1: AI Quiz Generator

Project

AI Quiz Generator

Overview

The **AI Quiz Generator** is a Flask-based web application that automatically generates multiple-choice quiz questions from Wikipedia or custom input text. It uses NLP techniques with pretrained transformer models and Sentence Transformers to create high-quality questions and plausible distractors. The frontend is crafted using **HTML**, **Tailwind CSS**, and **JavaScript** for an engaging and responsive user experience.

Features

- Generate multiple-choice questions (MCQs) from any topic.
- Uses Wikipedia for automatic context extraction.
- Accepts user-uploaded text or manual input.
- Three difficulty levels: Easy, Medium, Hard.
- Download quizzes as PDF or JSON.
- Intelligent distractor generation using SentenceTransformers.
- Handles numeric and named-entity answers.
- Clean frontend with animations and responsive design.

Technologies Used

Backend (Python/Flask):

- **Flask**: Web framework.
- **Transformers**: For question generation and answering.
- **SentenceTransformers**: Semantic similarity-based distractor generation.
- **NLTK**: For tokenization and POS tagging.
- **word2number**: Converts words like "three" to 3.
- **inflect**: Converts numbers like 3 to "three".
- **wikipedia**: Extracts summaries from Wikipedia.
- **FPDF** and **reportlab**: PDF generation.
- **flask-caching**: Caches generated results.

Frontend:

- **HTML**: Layout and structure.
- **Tailwind CSS**: Utility-first styling framework.
- **JavaScript**: Handles client-side logic like event listeners, theme toggle, and previews.

Folder Structure

AI_Quiz_Generator/

├── static/

│ ├── css/style.css

│ └── js/app.js

├── templates/

│ └── index.html

├── app.py

├── quiz_generator.py

├── requirements.txt

Working

Create a virtual environment:

```
python -m venv venv
```

Activating virtual requirement

```
source venv/bin/activate  # Windows: venv\Scripts\activate
```

Install dependencies:

```
pip install -r requirements.txt
```

Run the application:

```
python app.py
```

Visit http://localhost:5000 in your browser.

Usage Instructions

- Enter a topic (e.g., "Photosynthesis") or paste custom content.
- Choose difficulty and number of questions.
- Click **Generate**.
- Preview the quiz.
- Download as PDF or JSON.

Backend Logic

app.py

- Loads NLP models: valhalla/t5-base-e2e-qg for question generation and roberta-base-squad2 for answering.
- Initializes SentenceTransformer for distractor generation.
- Uses wikipedia to extract content if topic is selected.
- Creates quiz JSON, PDF, and HTML views.
- Uses sentence embedding similarity to create meaningful distractors.
- Includes numeric perturbation logic using handle_numeric_answer().
- Extracts answers and generates questions based on text chunks.

Frontend Overview

HTML (index.html)

- Layout with form fields for topic/content input.
- Sections for previewing quiz.

- Buttons to generate, download, and toggle themes.

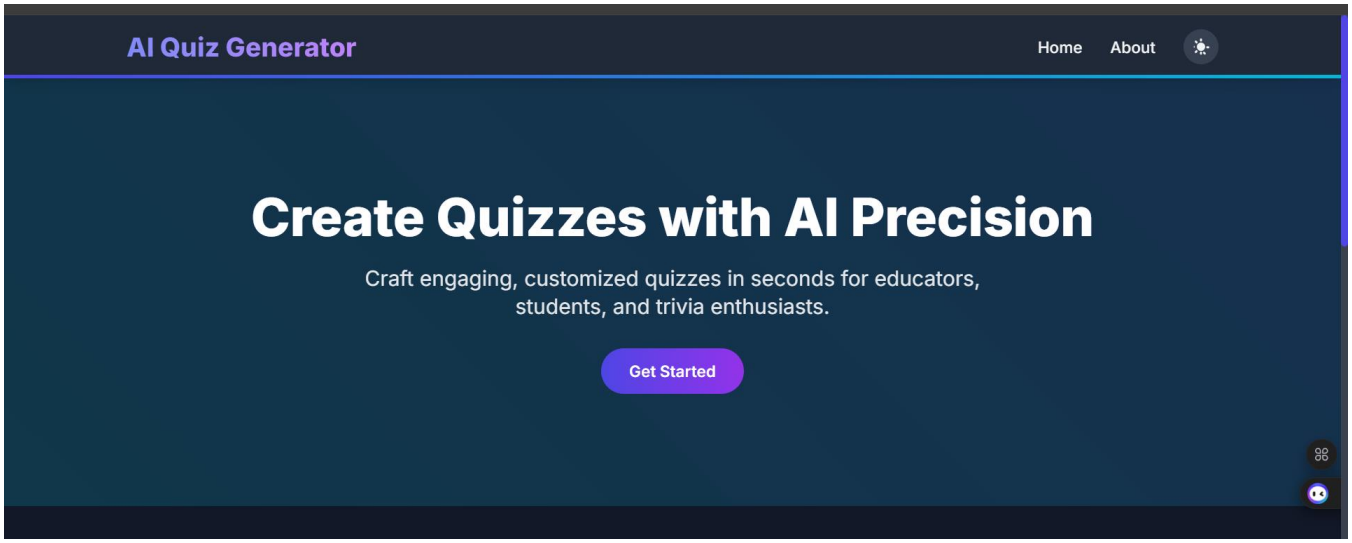
Tailwind CSS (style.css)

- Gradient background animation.
- Styled cards, buttons, alerts.
- Responsive and animated components using @apply, @keyframes.

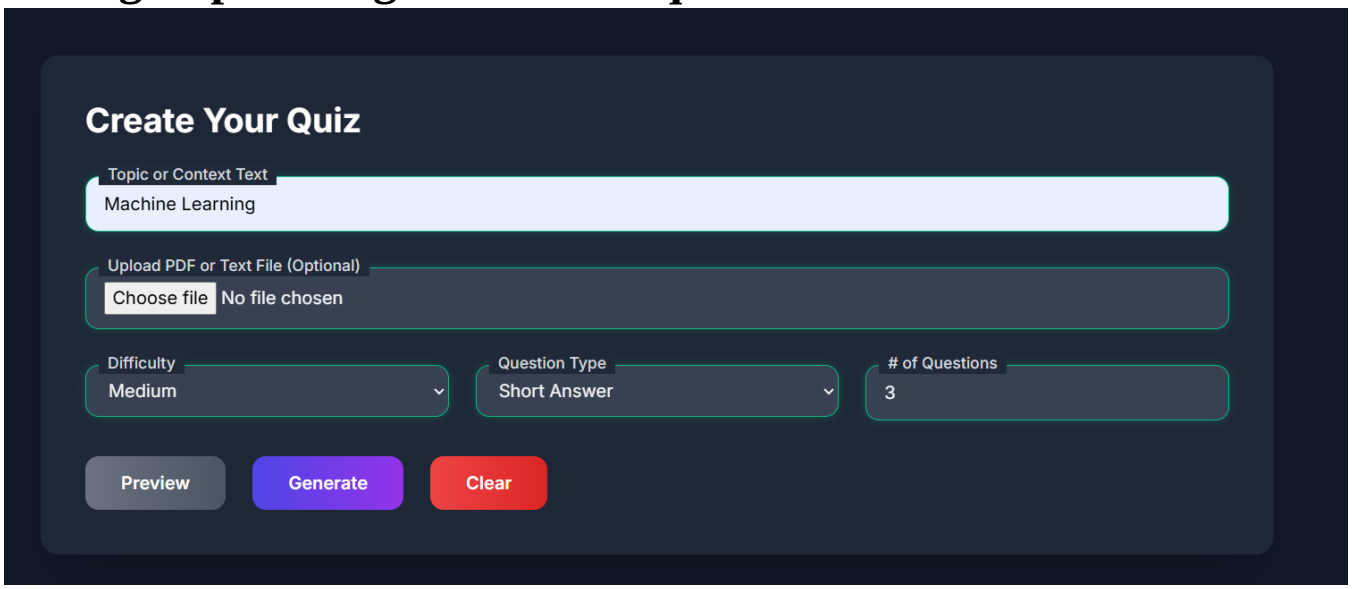
JavaScript (app.js)

- Handles form submission via fetch API.
- Updates preview area dynamically.
- Controls download buttons and theme toggles.

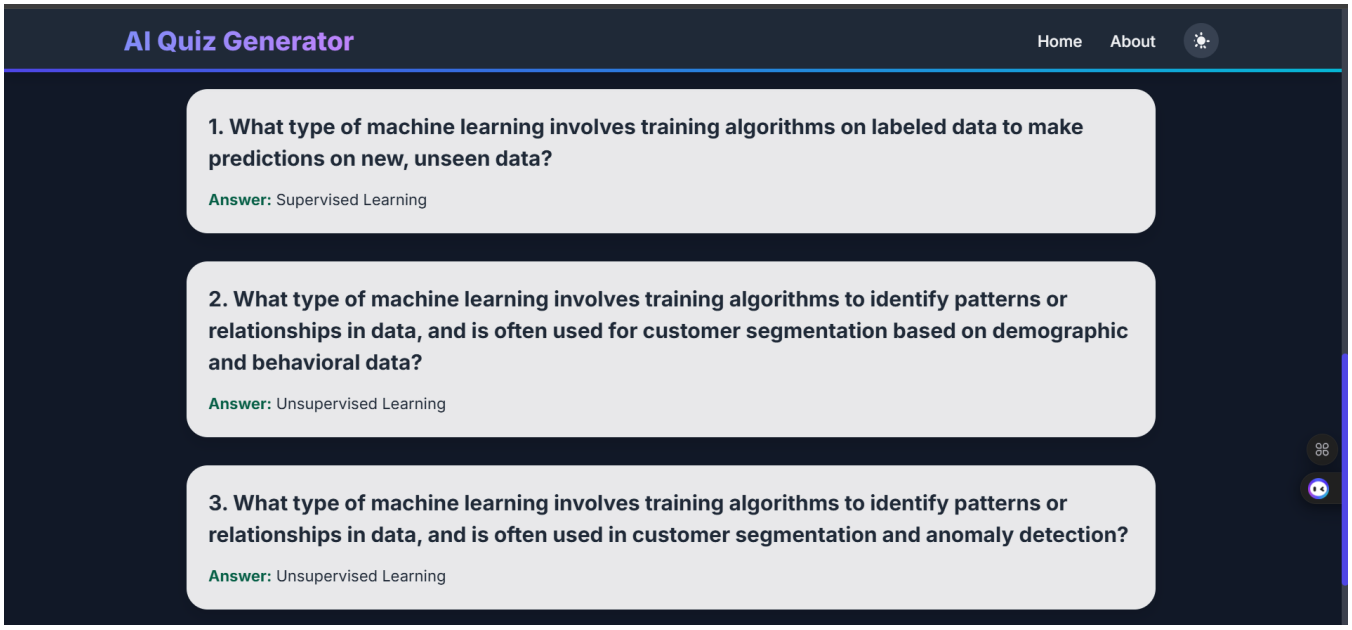
Interface:



Giving Topic and generation of quiz:



Generating Quiz:



Save Options: