Artificial Intelligence Capstone Project

Project Overview:-

The **AI Study Pal** is a simplified web-based application designed for students to apply the Artificial Intelligence Course Curriculum topics (Python, ML, DL, NLP). It creates a user-friendly study assistant that generates basic study plans, summarizes short texts, creates simple practice quizzes, and provides encouraging feedback in text format, it focuses purely on educational support, ensuring context-aware, believable, and adaptable study aids without complex features like voice output or advanced models.

Core Features Designed:

- **AI-Generated Study Plans**: Creates basic study schedules based on user inputs (subject, study hours) tailored for academic scenarios (e.g., exam prep, homework).
- **Scenario-Based Customization**: Allows users to adjust plans and quizzes for specific subjects (e.g., math, science).
- **Automated Quiz System**: Generates simple multiple-choice quizzes with easy/medium difficulty levels.
- **AI Motivational Feedback Generator**: Produces short, encouraging text feedback (e.g., "Good job on math!") to motivate users.
- **Believability and Realism**: Ensures plans, quizzes, and feedback are practical and relevant, with downloadable schedules as proof-like outputs.
- **Resource Suggestion System**: Recommends basic study resources (e.g., web links) based on subject.

Objectives:

- Develop a simple AI-powered study assistant with contextual awareness for academic needs.
- Enable users to generate study plans and quizzes for various subjects.
- Integrate automated quiz generation and basic resource suggestions.
- Ensure outputs are natural, believable, and easy to use, with options to save plans.
- Generate study aids in text format with downloadable schedules.

Implementation Details:

1. **Python and Data Setup**:
    - **Feature**: Collect a small dataset of educational texts (e.g., Wikipedia snippets) and user inputs (subject, hours). Use Pandas for data handling and Matplotlib for basic visualizations.
    - **Tasks**: Clean text data (remove duplicates, lowercase), perform simple EDA (e.g., count subjects), and store user inputs.
    - **Output**: Cleaned dataset, basic visualization (e.g., subject pie chart).
2. **Machine Learning for Quiz Generation**:
    - **Feature**: Generate simple quizzes using logistic regression to classify questions as easy/medium based on text features (Bag-of-Words). Use K-means to group similar topics for resource suggestions.
    - **Tasks**: Train logistic regression for question difficulty, cluster topics, and evaluate with accuracy and F1-score. Use basic hyperparameter tuning.
    - **Output**: Quiz generator with 5-10 questions, basic resource suggestions.
3. **Deep Learning for Summarization**:
    - **Feature**: Summarize short texts (e.g., 200 words to 50 words) using a basic neural network in Keras. Generate short feedback (e.g., “Keep it up!”) using pre-trained embeddings.
    - **Tasks**: Build a simple neural network for summarization and use GloVe embeddings for feedback.
    - **Output**: Summarized texts, short feedback messages.
4. **NLP for Study Tips**:
    - **Feature**: Process texts with NLP (NLTK) for keyword extraction (tokenization). Generate basic study tips (e.g., “Review key terms daily”).
    - **Tasks**: Implement tokenization and extract top keywords for tips.
    - **Output**: Short study tips based on input text.
5. **Web Deployment**:
    - **Feature**: Deploy a Flask web app for user interaction. Users input subject and hours; app outputs study plan, quiz, summary, tips, and schedule.
    - **Tasks**: Create a simple web interface for inputs and outputs, generate CSV schedules.
    - **Output**: Flask web app, downloadable schedules (CSV).

Tools and Technologies:

- **Programming Language**: Python
- **Libraries**:
    - Data: Pandas
    - Visualization: Matplotlib
    - ML: scikit-learn
    - DL: Keras
    - NLP: NLTK
    - Web: Flask
- **Dataset**: Small set of educational texts (e.g., Wikipedia snippets), user inputs (JSON/CSV).
- **Environment**: Anaconda, Jupyter Notebooks

Evaluation Metrics

- **ML Models**: Accuracy, F1-score for quiz difficulty classification.
- **DL Models**: Basic human evaluation for summary and feedback quality.
- **Web App**: Usability, output clarity (schedules).

Deliverables

- Cleaned dataset and simple visualizations.
- ML-based quiz generator and resource suggestions.
- DL-based text summarizer and feedback system.
- NLP-generated study tips.
- Flask web app with study plans, quizzes, summaries, tips, and downloadable schedules.
- Final report and presentation on methodology and results.

Why This Project?

- **Curriculum Fit**: Covers Python, ML (logistic regression, K-means), DL (basic neural network), NLP, and web deployment, aligning with core course topics.
- **Simplicity**: Streamlined for beginners, focusing on basic models and minimal features.
- **Engagement**: Helps students with their studies, making it relevant and motivating.
- **Believability and Realism**: Produces practical study aids with clear, downloadable outputs.

- **Scalability**: Simple enough for solo work, with room to add complexity for groups.
- **Ethics**: Positive educational focus, avoiding any deceptive elements.

Enzyme Office Space,
Backside of Star Bazaar, Sector 7,
HSR Layout, Bengaluru,
Karnataka - 560102

0-80621-81856
support@launched.org.in