# **I. Project Plan: RAG Implementation and Vector DB Usage**

## **Key Features**

### 1.1. **RAG Implementation**

#### Applied to PDFs, audio, video, and text files.

### 1.2. **Vector Database Implementation**

#### Also used with PDFs, audio, video, and text files.

## **Points to Remember**

### 2.1. **User Interface (UI)**

#### Create an interactive UI for user engagement.

### 2.2. **User Control**

#### Allow users to control whether the LLM is automated or manually operated.

### 2.3. **LLM Selection**

#### Let users choose the LLM, e.g., Grammarly or ChatGPT-like tools.

### 2.4. **Memory and Custom Settings**

#### Retain user-specific settings, such as:

#### **Coding Preferences:** Favoring for-loops, multiple statements in one line, Python over C, remembering file allocation and hardware setup.

#### **Credentials:** Storing user-specific details like name, USN, roll number.

#### **Language Preferences:** Remembering user’s age, demographics, etc.

### 2.5. **Privacy Considerations**

#### Comply with privacy laws and ensure data protection.

### 2.6. **Data Export**

#### Allow users to export their data for use in other systems.

### 2.7. **Reference Projects**

#### Explore similar projects available on GitHub for reference.

## **Project Details**

### 3.1. **Open Source**

#### This is an open-source project, not a commercial one.

### 3.2. **Scalability**

#### Smaller versions can run locally.

#### Larger, scalable versions can be hosted on the cloud.

#### Scalability should be a high priority.

## **Areas of Work**

### 4.1. **Front-End**

#### Technologies: JavaScript, Node.js, HTML, CSS.

### 4.2. **Back-End**

#### Key Areas: LLM, Generative AI, Database Management, RAG, Vector DB.