**Installation Guide for Document Summarizer**

**Overview**

This guide provides step-by-step instructions to install and set up the **Document Summarizer** project on a new system. The application is a Flask-based web tool that summarizes documents, converts text to speech, translates content, and performs analytics using AI techniques. It requires Python, MongoDB, and several Python packages.

**System Requirements**

* **Operating System**: Windows, macOS, or Linux
* **Python**: Version 3.8 or higher
* **MongoDB**: Version 4.0 or higher (local installation)
* **Internet Connection**: Required for downloading packages and external resources
* **Storage**: At least 2 GB free space
* **Software**: Terminal (Command Prompt, PowerShell, or Bash), Text Editor (e.g., VS Code), Web Browser

**Step-by-Step Installation**

**Step 1: Install Python**

1. **Download Python**:
   * Visit [python.org/downloads](https://www.python.org/downloads/).
   * Download the latest version (e.g., Python 3.12) for your operating system.
2. **Install Python**:
   * Run the installer.
   * Check the box **"Add Python to PATH"** during installation.
   * Click "Install Now" and follow the prompts.
3. **Verify Installation**:
   * Open a terminal (Command Prompt on Windows, Terminal on macOS/Linux).
   * Type python --version or python3 --version and press Enter.
   * You should see output like Python 3.12.0. If not, ensure PATH is configured correctly.

**Step 2: Install MongoDB**

1. **Download MongoDB**:
   * Visit [mongodb.com/try/download/community](https://www.mongodb.com/try/download/community).
   * Download the Community Server for your operating system.
2. **Install MongoDB**:
   * Run the installer.
   * Follow the setup wizard, keeping default settings (e.g., install MongoDB as a service).
   * Note the data directory (default: C:\data\db on Windows) and ensure it exists.
3. **Start MongoDB**:
   * Open a terminal.
   * On Windows: Run mongod (ensure the terminal is in the MongoDB bin directory or add it to PATH).
   * On macOS/Linux: Run brew services start mongodb-community (if using Homebrew) or mongod.
   * Verify by opening another terminal and typing mongo. You should enter the MongoDB shell.
4. **Troubleshooting**:
   * If mongod fails, check the data directory permissions or create it manually.

**Step 3: Set Up the Project Environment**

1. **Clone the Repository**:
   * Install Git if not already installed:
     + Download from [git-scm.com](https://git-scm.com/downloads).
     + Follow the installation wizard.
   * Open a terminal and navigate to your desired directory (e.g., cd D:\Projects).
   * Run:

text

CollapseWrap

Copy

git clone https://github.com/yourusername/document-summarize.git

cd document-summarize

(Replace yourusername with your GitHub username or use a local copy.)

1. **Create a Virtual Environment**:
   * Run:

text

CollapseWrap

Copy

python -m venv venv

* + Activate it:
    - Windows: venv\Scripts\activate
    - macOS/Linux: source venv/bin/activate
  + You should see (venv) in the terminal prompt.

**Step 4: Install Dependencies**

1. **Create requirements.txt**:
   * If not present, create a file named requirements.txt in the project directory.
   * Copy the following content into requirements.txt:

text

CollapseWrap

Copy

Flask==3.0.3

flask\_pymongo==2.3.0

werkzeug==3.0.3

flask\_cors==4.0.0

gTTS==2.5.3

deep\_translator==1.11.4

pdfplumber==0.11.3

python-docx==1.1.2

spacy==3.7.5

newspaper3k==0.2.8

pymongo==4.8.0

1. **Install Packages**:
   * In the terminal (with the virtual environment activated), run:

text

CollapseWrap

Copy

pip install -r requirements.txt

1. **Install spaCy Model**:
   * Run:

text

CollapseWrap

Copy

python -m spacy download en\_core\_web\_sm

* + This downloads the English language model required for NLP processing.

**Step 5: Configure the Project**

1. **Create Upload Folder**:
   * The code creates a static/uploads folder automatically, but verify it exists:
     + Navigate to static in the project directory.
     + If uploads is missing, create it manually.
2. **Set Environment Variables**:
   * Create a .env file in the project root.
   * Add:

text

CollapseWrap

Copy

FLASK\_APP=app.py

FLASK\_ENV=development

* + (Optional: Add API keys if using external services for translation/speech.)

**Step 6: Run the Application**

1. **Start the App**:
   * In the terminal (with the virtual environment activated), run:

text

CollapseWrap

Copy

python app.py

* + You should see output indicating the Flask server is running on http://127.0.0.1:5001.

1. **Access the App**:
   * Open a web browser and go to http://localhost:5001.
   * The registration page should load. Follow the on-screen prompts to register and log in.

**Step 7: Test the Application**

1. **Register and Log In**:
   * Enter a name, email, username, and password on the registration page.
   * Log in with the same credentials.
2. **Summarize a Document**:
   * On the home page, paste a URL (e.g., a news article) or upload a PDF/DOC/DOCX file.
   * Click "Summarize" and verify the summary appears.
3. **Test Features**:
   * Translate the summary using the language dropdown.
   * Click "Speak" to listen to the summary.
   * Navigate to /features, /about, and /profile to ensure all pages load.
4. **Update Profile**:
   * On the profile page, modify your name, email, or phone number and click "Update Profile" to save changes.

**Troubleshooting**

* **Python Not Found**: Ensure Python is added to PATH. Reinstall if necessary.
* **MongoDB Not Running**: Check the terminal for mongod errors. Start MongoDB manually if needed.
* **Module Not Found**: Re-run pip install -r requirements.txt and verify all packages.
* **spaCy Error**: Ensure the en\_core\_web\_sm model is downloaded.
* **File Upload Fails**: Verify the static/uploads folder is writable.
* **Server Errors**: Check the terminal logs for exceptions and address them (e.g., missing dependencies).