VoiceLink Intelligent Meeting Assistant

1. Project Structure

This is the project structure of our VoiceLink Intelligent Meeting Assistant:

```
VoiceLink Intelligent Meeting Assistant
   ├─ assets
   — арр
    ├─ data
    - static
    | ├─ background.jpg
      ├─ templates
    | ├── home.htnl
    | ├─ meetingAssistant.html
     ├── meetingHistory.html
    | └── meetingNote.html
    ├─ __init__.py
    backendRoutes.py
     |-- meetingInfo.py
   └─ voiceRoutes.md
   — run.py
   ├─ requirements.txt
   -- README.md
   - README.pdf
   - REPORT.md
   └─ REPORT.pdf
```

- run.py: The entry file for launching the application.
- data folder: It contains records of past meetings.
- static folder: It contains some static image styles.
- **templates** folder: This is where the html code for all front-end pages is stored.
- python files in the app folder: Back-end routing and interface code files.
- requirements.txt: Python packages that the project depends on.
- **README.md/README.pdf**: The readme file with instructions on how to run the program.
- **REPORT.md/REPORT.pdf**: Project presentation report in English.

2. How to run the Source Code

2.1. Prerequisites

Before running the project, make sure you have the following software installed:

- Python 3.6 or later.
- pip (Python package management tool).

2.2. Installation dependency

Install the required Python package using the requirements.txt file:

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

2.3. Running the Application

• Use the cd command to switch to the project root directory:

```
cd /path/to/VoiceLink-Intelligent-Meeting-Assistant
```

• Run the startup file in the project root directory:

```
python run.py
```

• The Flask application has been successfully started and is running on the local server, Visit [http://127.0.0.1:5000:

```
PS D:\2024_2025\SpeechRecognition\VoiceLink-Intelligent-Meeting-Assistant> python run.py
  * Serving Flask app 'app'
  * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
  * Running on all addresses (0.0.0.0)
  * Running on http://127.0.0.1:5000
  * Running on http://100.80.130.253:5000
Press CTRL+C to quit
  * Restarting with watchdog (windowsapi)
  * Debugger is active!
  * Debugger PIN: 340-782-134
```

• Then you go to the home page of our project, and now you can use it:

