Air Aware Smart Air Quality Prediction System

presented by -Likhitha Karri-

1. Flask API

- Flask is a simple Python framework used to create APIs and web apps.
- It allows you to create URLs (routes) and write Python code that runs when someone visits that URL.

Example:

```
from flask import Flask
app = Flask(__name__)

@app.route("/hello")
def hello():
    return "Hello!"
```

2. How to import Flask

```
from flask import Flask, request, jsonify
```

- Flask → create app
- request → read data (payload)
- jsonify → send JSON response

3. Fast API

- Fast API is a modern, fast Python framework to build APIs with automatic documentation.
- It is faster than Flask and supports async features.

Example:

```python
from fastapi import FastAPI
app = FastAPI()

@app.get("/hello")
def hello():
    return {"msg": "Hello!"}
```

4. Payload

- Payload = data sent by the client to the server.
- Usually sent as JSON in the request body.

Example JSON Payload:

```json
{
  "name": "Likhitha",
  "age": 21
}
```

5. Postman

- Postman is a tool used to test APIs.
- You can send GET/POST requests, add payloads, headers, and check responses.

6. WebSocket

- WebSocket is a communication method where both client and server can send messages anytime (real-time).
- Used for chat apps, live updates, notifications.

7. Streaming

- Streaming means sending data in small parts instead of sending everything at once.
- Used for live logs, video/audio, real-time updates.

8. Flashing (Flask Flash Messages)

Flashing is used to show one-time messages to the user like:
✓ “Login successful”
✓ “Saved successfully”

Example:

```
flash("Data saved!")
```

1. git add .

- Adds all changed files in the current folder to the staging area.

👉 Means: “Git, track ALL my changes.”

2. git add main.py

- Adds only one file (main.py) to the staging area.

👉 Means: “Track only this file.”

3. git commit -m "message"

- Saves the staged changes with a message.

👉 Means: “Create a checkpoint with this message.”

Example:

- git commit -m "fixed bug"

4. git push

- Sends your committed changes to the remote repository (GitHub).

👉 Means: “Upload my work to GitHub.”

5. git branch

- Shows all local branches.

👉 Means: “Show branches saved on my computer.”

6. git branch --all

- Shows local + remote branches.

👉 Means:

- Local branches
- Remote branches (from GitHub)

7. git fetch --all

- Downloads information about all remote branches, but does NOT merge anything.

👉 Means:
"Check if GitHub has new updates, but don't apply them yet."

8. git pull

- **Use:** Download the latest changes from GitHub and merge them into your local branch

👉 "Get updates + merge."

9. git checkout -b "sh_new"

- **Use:** Create a new branch and switch to it.

👉 "Make a new branch named *sh_new*."

10. git checkout "sh_new"

- **Use:** Switch to an existing branch.

👉 "Move to sh_new branch."

11. python -m venv venv

- **Use:** Create a virtual environment named venv.

👉 "Separate Python environment for this project."

12. python -m venv shakthi

- **Use:** Create a virtual environment named shakthi.
  (Same as above but different name.)

13. requirements.txt

- **Use:** A file that lists all Python packages your project needs.

👉 Helps others install the same dependencies.

14. pip install -r requirements.txt

- **Use:** Install all packages listed in requirements.txt

👉 "Install everything needed for the project."

15. deactivate

- **Use:** Exit the virtual environment.

👉 Go back to the normal system Python.

AI / ML Concepts

✅ AI Models

Programs trained to perform tasks like chat, prediction, or image generation.

Platforms/Companies

- Google → Gemini models
- OpenAI → GPT models
- Groq → Ultra-fast inference hardware + LLMs
- Microsoft → Copilot, Azure OpenAI services

LLM Families

✅ Gemini

- Google's family of Large Language Models.

✅ GPT

- OpenAI's Large Language Models (ChatGPT).

✅ LLaMA

- Meta's open-source LLM family.

✅ Copilot

- Microsoft's AI assistant built on LLMs.

LLM Basics

✅ LLM (Large Language Model)

- An AI model trained on massive text data to understand and generate human language.

AI Platforms

✅ Google AI Studio

- Google's platform to build apps with Gemini models.

✅ OpenAI Platform

- Dashboard to use GPT models, APIs, playgrounds.

✅ Groq Platform

- Super-fast inference platform to run LLMs.

Rate/Token topics

Rate Limit

- 100 requests/mUse: Maximum number of API calls you can make per minute/hour.
- Example:in → after that API will block temporarily.

Token Limit

- Use: Maximum text the model can handle in one request (input + output).

Example:

- GPT-4o: 128k tokens

Chat Completion Component

- The part of an API that allows conversation-style messages.

Example:

```
{
  "model": "gpt-4",
  "messages": [
    {"role": "user", "content": "Hello"}
  ]
}
```

✅ Prompt

A prompt is the *input or instruction* you give to an AI model.

Example:
"Explain machine learning in simple words."

✅ ML (Machine Learning)

Machine Learning is a method where computers learn from data without being explicitly programmed.

Three main types of Machine Learning

✅ 1. Supervised Learning

The model learns from labeled data (input + correct output).

👉 **Example:**

- Email → “spam” or “not spam”
- Image → “cat” or “dog”

Types inside Supervised Learning:

✓ Regression

Predict continuous values
Examples:

- House price prediction
- Temperature prediction

✓ Classification

Predict categories/labels
Examples:

- Spam or not spam
- Disease or no disease

2. Unsupervised Learning

The model learns from unlabeled data (no correct answers given).

👉 **Example:**

- Grouping similar customers
- Finding patterns in data

Most common technique:

✓ **Clustering**
Example:

- Grouping customers by buying behavior

3. Reinforcement Learning

- The model learns by trial and error using rewards and punishments.

Examples:

- Training robots
- Game-playing AI (Chess, Go)
- Self-driving cars

✅ Embedding

Definition:
Embedding is a way of converting text into numbers (vectors) so that a computer/AI model can understand the meaning.

Simple example:
"cat", "dog", "lion" → their embeddings will be close to each other because meanings are similar.

Use cases:

- Search
- Recommendations
- Chatbots
- Similarity matching

✅ NLTK (Natural Language Toolkit)

Definition:
NLTK is a Python library used for text processing and NLP tasks.

It helps in:

- Tokenization (splitting text into words)
- Stemming
- Lemmatization
- Removing stopwords
- Basic NLP experiments

Example:

```
import nltk
from nltk.tokenize import word_tokenize


word_tokenize("I love AI")
```