

# Infosys Springboard – Project Documentation

**Name:** Raunit Raj Singh

**Batch:** Infosys Springboard – Batch 6

**Project Name:** AirAware – Smart Air Quality Prediction System

**Duration:** 45 Days (Starting from 10 Nov 2025)

---

## Day 1 – Project Introduction & Objective

**Project Name:** AirAware – Smart Air Quality Prediction System

### Overview:

AirAware is a smart system designed to monitor and predict air quality using data analysis and machine learning. It uses environmental data such as **PM2.5, CO<sub>2</sub>, NO<sub>2</sub>, SO<sub>2</sub>** etc. to predict the air quality level.

### Objective:

- Collect real air quality dataset of Maharashtra from Kaggle.
- Data contains **PM2.5, CO<sub>2</sub>, NO<sub>2</sub>, SO<sub>2</sub>, Temperature, Humidity** etc.
- Performed **data cleaning, normalization, and missing value handling**.

**Dataset:** Kaggle – Maharashtra Air Quality Dataset

---

## Day 2 – Team Allocation

**Team:** Team 2

- Raunit Raj Singh
  - Fariha
  - Likhita
  - Sahil
  - Sneha
- 

## Day 3 – APIs (Backend Basics)

**API Meaning:** Application Programming Interface – allows communication between systems.

### Types of APIs we use:

- **FastAPI** – Fast, modern Python API framework.
- **Flask API** – Lightweight Python web framework.

- **Postman** – Tool to test API endpoints.

### Important API Methods:

- **GET** – Fetch data
- **POST** – Send data
- **PUT** – Update data
- **DELETE** – Delete data

### Example:

---

## Day 4 – Git & Version Control

### Git Commands:

```
git add .
git add main.py
git commit -m "message"
git push
git branch
git branch --all
git fetch --all
git pull
git checkout -b "sh_new"
git checkout "sh_new"
```

### Steps to Upload Project to GitHub:

1. Download & install Git.
2. Create GitHub repo.
3. Copy GitHub URL.
4. Open terminal → type:

```
git clone <url>
```

### Python Environment Setup:

```
python -m venv venv
python -m venv shakthi

# activate
venv\Scripts\activate

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

deactivate

### requirements.txt:

Used to store all Python libraries.

---

## Day 5 – Database (DBMS Basics)

### What is Database?

A structured collection of data (tables).

### What is DBMS?

Database Management System – software to manage databases.

### CRUD Operations:

- Create
- Read
- Update
- Delete

### Structured Data Example:

ID	Name	Age

### Unstructured Data Example:

- Images
- Videos
- Audio

### Normalization:

- **1NF:** No repeating columns.
- **2NF:** Remove partial dependency.
- **3NF:** Remove transitive dependency.
- **BCNF:** Stronger version of 3NF.
- **4NF & 5NF:** Advanced forms of normalization.

### Keys:

- **Primary Key:** Unique identifier.
  - **Foreign Key:** Links two tables.
-

# Day 6 – AI / ML Concepts

## AI Models Providers:

- Google
- OpenAI
- Groq
- Microsoft

## Famous Models:

- Gemini
- GPT
- LLaMA
- Copilot

## LLM (Large Language Model):

Models trained on vast text data.

## Platforms:

- Google AI Studio
- OpenAI Developer Platform
- Groq

## Important Terms:

- **Rate Limit** – Maximum requests allowed.
- **Token Limit** – Maximum text size.
- **Prompt** – Instructions to AI.
- **Chat Completion API** – For responses.
- **Embedding** – Converting text → numerical vector.
- **Vector**: Collection of numerical values.

## ML (Machine Learning)

### 1. Supervised Learning

- Regression (Predict numbers)
- Classification (Predict categories)

### 2. Unsupervised Learning

- Clustering

### 3. Reinforcement Learning

- Learning based on rewards.

## Day 7 – NLTK (Natural Language Toolkit)

**NLTK: Python library for text processing.**

- Tokenization
- Stemming
- Lemmatization

Example:

```
from nltk.tokenize import word_tokenize
print(word_tokenize("Air quality is important"))
```

---

### Git Clone Steps (Full):

1. Install Git in your system.
2. Go to your GitHub repository.
3. Click on **Code** and copy the HTTPS URL.
4. Open Command Prompt or Terminal.
5. Type the command:

```
git clone <your-repo-url>
```

Example:

```
git clone https://github.com/username/airaware-project.git
```

## Day 8 & Day 9 – Project PPT & UI Presentation

**Work Done:**

- Designed **Project Presentation (PPT)** explaining AirAware.
- Added **Problem Statement, Objective, Dataset, Working, Output.**
- Created **UI design (Frontend layout)** for AirAware.
- Presented overall flow of the system.
- Explained prediction output format using charts and tables.

---

## Day 10 – ML, DL, NLP & Knowledge Based System

### Machine Learning (ML)

Machine learns from data without being explicitly programmed.

## Deep Learning (DL)

Uses multi-layered neural networks (ANN, CNN, RNN, LSTM, Transformer).

### Models:

- **ANN** – Artificial Neural Network (basic neural model)
- **RNN** – Recurrent Neural Network (sequence data)
- **CNN** – Convolutional Neural Network (image related)
- **LSTM** – Long Short-Term Memory (improves RNN)
- **Transformer** – Used in modern AI like GPT

### Gates in LSTM:

- **Forget Gate** – what to remove
- **Input Gate** – what to add
- **Output Gate** – final output

## Knowledge Based System

System that uses stored knowledge to take decisions.

---

# Day 11 – NLP (Natural Language Processing)

## Why NLP is Important?

Helps machines understand human language.

### Examples:

- Chatbots
- Translation
- Voice assistant

## NLTK Library (Natural Language Toolkit)

Used for text processing.

## Why NLTK is Useful?

- Easy to use
- Powerful text tools
- Free and open-source

### Installation:

```
pip install nltk
```

### Techniques:

- Tokenization – split sentence into words
- Stopword Removal – remove is, the, and, etc.
- Stemming – root word
- Lemmatization – proper root form
- POS Tagging – noun, verb, adjective, etc.
- NER (Named Entity Recognition) – person, place, org

### TF-IDF:

- **TF (Term Frequency)** – how often a word appears
- **IDF (Inverse Document Frequency)** – importance of word

### SVM in NLP

Used for text classification.

---

## Day 12 – SVM & Reinforcement Learning

### Support Vector Machine (SVM)

- Supervised Learning Algorithm
- Used for classification & regression
- Used in outlier detection
- Uses a **margin** (distance between data & line)

### Reinforcement Learning (RL)

- Learning by reward & punishment

### AI Agent / Agentic AI

An agent observes environment + takes action + gets reward.

---

## Day 14 – OpenAI API & Parameters

### Install OpenAI:

```
pip install openai
```

### Basic Structure:

- **client** = OpenAI()
- **model** = gpt
- **messages**: system + user
- **max\_tokens** – output size

- **temperature** – randomness (0.1 – 1.0)
- 

## Day 17 – ML Algorithms

### Logistic Regression

Formula:

$$z = w_1x_1 + w_2x_2 + \dots + b$$

Python:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

---

### Decision Tree

Example: - If weather is rainy → don't play - If sunny → play

#### Entropy Formula:

$$H(S) = - \sum p \log_2 p$$

#### Information Gain (IG):

$$IG = \text{Entropy}(\text{Parent}) - \text{Entropy}(\text{Child})$$

Python:

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()
```

---

### Random Forest

Many decision trees combined

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
```

---



## K-Nearest Neighbour (KNN)

Uses nearest neighbor to predict.

---

## Clustering

Finds centroid (center of data groups).

---

## XGBoost

Advanced boosting algorithm

```
from xgboost import XGBClassifier
model = XGBClassifier()
```

---

# Day 18 – MySQL & SQL Basics

## What is SQL?

SQL (Structured Query Language) is used to communicate with databases and manage data.

## Common SQL Commands:

- **SELECT** – get data
- **INSERT** – add new data
- **UPDATE** – modify data
- **DELETE** – remove data
- **CREATE** – create table
- **ALTER** – change table structure
- **DROP** – delete table
- **RENAME** – change table name
- **TRUNCATE** – remove all data from table

## WHERE Clause

Used to apply condition.

```
SELECT * FROM air_data WHERE pm25 > 100;
```

## Operators:

= equal to

greater than < less than = greater or equal <= less or equal <> or != not equal BETWEEN in  
a range LIKE pattern IN list of values AND match both OR match any one NOT opposite

## ORDER BY

```
SELECT * FROM air_data ORDER BY pm25 DESC;
```

## INSERT INTO

```
INSERT INTO air_data VALUES (1, 55, 'Good');
```

## UPDATE

```
UPDATE air_data SET status='Bad' WHERE pm25 > 100;
```

## MIN, MAX, COUNT, AVG, SUM

```
SELECT MAX(pm25) FROM air_data;  
SELECT COUNT(*) FROM air_data;
```

---

## LIKE Patterns

- %a → ends with a
- a% → starts with a
- %a% → contains a
- \_a% → second letter is a
- a\_% → starts with a (min 2 letters)
- a\_\_% → starts with a (min 3 letters)

---

## IN and NOT IN

```
SELECT * FROM air_data WHERE city IN ('Mumbai', 'Pune');
```

## BETWEEN

```
SELECT * FROM air_data WHERE pm25 BETWEEN 50 AND 100;
```

## Alias

```
SELECT pm25 AS AirQuality FROM air_data;
```

---

## Joins

**INNER JOIN** – common data

```
SELECT * FROM city INNER JOIN air_data ON city.id = air_data.id;
```

**LEFT JOIN** – all left table + matched right

```
SELECT * FROM city LEFT JOIN air_data ON city.id = air_data.id;
```

**RIGHT JOIN** – all right table + matched left

**CROSS JOIN** – all combinations

---

## UNION and UNION ALL

```
SELECT city FROM table1  
UNION  
SELECT city FROM table2;
```

Note: UNION removes duplicates, UNION ALL keeps them.

---

## GROUP BY

```
SELECT city, AVG(pm25) FROM air_data GROUP BY city;
```

## HAVING

Used with GROUP BY

```
SELECT city, AVG(pm25)  
FROM air_data  
GROUP BY city  
HAVING AVG(pm25) > 80;
```

# Day 19 – Project PPT Preparation

## Work Done:

- Prepared complete **Project PPT** including:
  - Introduction
  - Objective
  - Dataset details
  - Data preprocessing
  - Machine learning workflow
  - Results & charts
  - Designed clean slides for presentation.
  - Added diagrams for better understanding.
- 

# Day 20 – Milestone 2 Presentation

## Topics Covered:

- PPT Presentation
  - Basic Chatbot Presentation
  - AirAware Air Quality Project Presentation
- 

# Day 23 – Python Basics

## What is Python?

Python is a high-level, easy, powerful programming language used for AI, ML, Web Development, Automation, etc.

## Python Basic Concepts:

- Variables
- Data Types (int, float, string, list, tuple, dict)
- If-else
- Loops (for, while)
- Functions
- Input/Output
- Comments (#)
- Lists and dictionaries

## Example:

```
x = 10
y = 20
print(x + y)
```

---

## Day 24 – Python Built-in Methods (String Methods)

### Common Built-in Methods:

- **lower()** – converts to lowercase
- **upper()** – converts to uppercase
- **split()** – splits text into list

### More String Methods:

- **strip()** – removes extra spaces
- **join()** – joins elements
- **replace()** – replaces text
- **startswith()** – checks starting text
- **endswith()** – checks ending text
- **find()** – finds position of substring
- **isdigit()** – checks if numeric
- **isalpha()** – checks if alphabets only

### Example:

```
text = "  hello python  "
print(text.strip())
```

---

## Day 26 & Day 27 – Python Concepts (Core Understanding)

### Python Key Features

- Easy to learn and read
- Interpreted language
- Platform independent
- Large standard library

### Built-in & Platform Independent

Python has many built-in functions and works on Windows, Linux, and Mac.

### PEP 8

PEP 8 is Python's coding style guide. **Why important?** It makes code clean, readable, and standard.

## Data Types

- **String** – text data
- **List** – ordered, mutable
- **Tuple** – ordered, immutable
- **Set** – unordered, unique values
- **Dictionary** – key-value pairs

## Python Memory Management

Python uses automatic memory allocation and garbage collection.

## Indentation in Python

Indentation defines code blocks instead of braces {}.

## Interpreted Language

Python code runs line by line using an interpreter.

## Namespace

Namespace is a space where variable names are stored.

## List is Mutable

List values can be changed.

## List vs Tuple

- List → mutable
- Tuple → immutable

## Set Usage

Used to store unique values and remove duplicates.

## Dictionary

Stores data in key-value pairs.

## Merge Dictionaries

Using update() method.

## Remove Duplicates from List

Convert list to set.

## **Flatten Nested List**

Convert nested list into single list.

## **Shallow vs Deep Copy**

- Shallow copy shares reference
- Deep copy creates new object

## **Slicing**

Used to access part of list or string.

## **Reverse List**

Using reverse() or slicing.

## **Frozen Set**

Immutable version of set.

## **is vs ==**

- is → checks memory
- == → checks value

## **Stack and Queue**

Stack → LIFO Queue → FIFO

## **Functional Programming**

Uses functions as first-class objects.

---

# **Day 28 & Day 29 – Functional Consultant & Software Lifecycle**

## **Functional Consultant / Business Analyst**

Understands business needs and converts them into requirements.

## **Documents**

- Functional Specification Document (FSD)
- Technical Specification Document (TSD)

## **Prototype**

Early model of the application.

## **Software Phases**

- Development
- Quality
- Production

## **Testing / Quality Assurance**

Ensures application works correctly.

## **Test Script**

Written based on functional specifications.

## **Test Results**

Records pass or fail status.

## **Administration Areas**

- Network
- Database
- Application
- Security

## **Other Teams**

- Product / Project Management
- Sales & Marketing
- Human Resources
- Documentation Team
- Support Team

---

# **Day 30 – DSA & Sorting Algorithms**

## **Data Structures & Algorithms (DSA)**

DSA helps to store, organize, and process data efficiently.

## **Sorting Algorithms (with Number Examples)**

### **1. Bubble Sort**

Compares adjacent elements and swaps them.



**Example:** Input: [5, 3, 1, 4] Pass 1 → [3, 1, 4, 5] Pass 2 → [1, 3, 4, 5] Output: [1, 3, 4, 5]

**Simple Code:**

```
def bubble_sort(arr):
    for i in range(len(arr)):
        for j in range(len(arr)-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
```

---

## 2. Selection Sort

Selects the smallest element and places it at correct position.

**Example:** Input: [4, 2, 6, 1] Step 1 → [1, 2, 6, 4] Step 2 → [1, 2, 6, 4] Output: [1, 2, 4, 6]

---

## 3. Insertion Sort

Inserts elements into sorted part.

**Example:** Input: [8, 3, 5, 2] Step 1 → [3, 8, 5, 2] Step 2 → [3, 5, 8, 2] Output: [2, 3, 5, 8]

---

## 4. Merge Sort

Divides array and merges in sorted order.

**Example:** Input: [7, 2, 6, 3] Divide → [7,2] [6,3] Merge → [2,7] [3,6] Final → [2,3,6,7]

---

## 5. Quick Sort

Uses pivot element.

**Example:** Input: [5, 1, 9, 3] Pivot = 5 Left → [1,3] Right → [9] Output → [1,3,5,9]

---

## 6. Heap Sort

Uses heap data structure.

**Example:** Input: [9, 4, 7, 1] Heap → [9,7,4,1] Sorted → [1,4,7,9]

---

## Day 30 – Milestone 3 Review

On Day 30, ma'am checked whether the project work was **75% completed or not**. Some teams completed their presentations. By the time those presentations finished, the class time was over. So, the remaining teams could not present. Ma'am informed that the remaining presentations would be continued later.

---

## Day 31 – Remaining Presentations

On Day 31, the remaining teams gave their presentations. Ma'am evaluated their work and **marks were awarded** to the teams. The session was completed successfully.

---

## Day 32 – Error Checking & Suggestions

Some teams faced errors in their projects. Ma'am checked those errors carefully. She provided suggestions to fix them. Teams were asked to make the required corrections.

---

## Day 33 – Coding Tasks & Feedback

Ma'am assigned some coding tasks. We were asked to write the required code. The written code was checked. Ma'am gave feedback and corrections wherever needed.

---

## Day 34 – Q-Learning Algorithm (Reinforcement Learning)

### Introduction to Q-Learning

Q-Learning is a reinforcement learning algorithm used to find the best action an agent should take in a given state to maximize rewards. It learns by trial and error and improves decisions over time.

### Q-Value

A Q-value represents the expected future reward for taking a specific action in a given state. Higher Q-value means a better action.

### Q-Table

Q-values are stored in a table called a Q-table. Each row represents a state and each column represents an action. The table is updated repeatedly during learning.

## Rewards and Episodes

- Reward is the feedback received after an action.
- An episode is one complete run from start to end.
- The agent moves step by step based on rewards.

## Temporal Difference (TD) Update Formula

$Q(s,a) \leftarrow Q(s,a) + \alpha [ R + \gamma Q(s',a') - Q(s,a) ]$  Where: -  $\alpha \rightarrow$  learning rate -  $\gamma \rightarrow$  discount factor -  $R \rightarrow$  reward

## Bellman Equation

$$Q(s,a) = R(s,a) + \gamma \max_{a'} Q(s',a')$$

## Applications of Q-Learning

- Stock trading
- Recommendation systems
- Traffic signal control
- Robotics

---

# Day 35 – Backpropagation in Neural Networks

## What is Backpropagation?

Backpropagation is an algorithm used to train neural networks by reducing the error between predicted and actual output. It updates weights using gradient descent and the chain rule.

## Main Steps

1. Forward Pass
2. Error Calculation (MSE)
3. Backward Pass
4. Weight Update

## Sigmoid Function

$$\sigma(x) = 1 / (1 + e^{-x})$$

## Advantages

- Improves accuracy
- Works for deep networks
- Learns complex patterns

## Challenges

- Vanishing gradient
- Exploding gradient
- Overfitting

---

## Day 36 – Path Finding & Unsupervised Learning

### BFS (Breadth First Search)

BFS is used to find the shortest path in an unweighted graph. It uses a queue and explores level by level.

### Applications of BFS

- Maze solving
- Network routing
- Shortest path problems

### GMM (Gaussian Mixture Model)

An unsupervised clustering algorithm based on Gaussian distributions. Used in image and speech recognition.

### PCA (Principal Component Analysis)

Used to reduce dimensions while preserving important information. Improves performance and visualization.

---

## Day 37 & Day 38 – Project Development Time

Ma'am allotted class time for group project development. Teams focused on: - Continuing project development - Fixing bugs - Improving UI - Implementing pending features Ma'am supervised and cleared doubts.

---

## Day 39 – Milestone 4 Review

On Day 39, ma'am conducted the **Milestone 4 review**. Each group presented their work related to the milestone.

---

## Day 40 – Milestone 4 Review Continuation

On Day 40, ma'am continued the **Milestone 4 review**. Each group presented their final work and progress.

---