AIR QUALITY PREDICTION PROJECT USING MACHINE LEARNING

PROJECT REPORT 2025-2026

Submitted By

20507182 ARUNKUMAR J

24507222 SWETHA S

24507197 ISHWARYA V

24507199 JAYAKUMAR J

Under the guidance of

Mrs. C.SRIDEVI B.E., CSE

HOD- COMP & IT

Diploma in Computer Engineering

of the Directorate of Technical Education, Government of Tamil Nadu

DEPARTMENT OF COMPUTER ENGINEERING

AKT MEMORIAL POLYTECHNIC COLLEGE

KALLAKURICHI-606202.

AKT MEMORIAL POLYTECHNIC COLLEGE

KALLAKURICHI-606202

Department of Computer Engineering

BONAFIDE CERTIFICATE

This is certified that this project work entitled **Air quality prediction project using machine learning** has been submitted **ARUNKUMAR J, SWETHA S, ISHWARYA V,JAYAKUMAR J** in the partial fulfilment of the requirements for the award of Diploma in Computer Engineering during the academic year 2025-2026, who carried out the project work under our supervision.

Project Guide	**Head of the Department**
Mrs. C.SRIDEVI B.E.,(CSE)	**Mrs. C.SRIDEVI B.E.,(CSE)**
HOD -COMPUTER & IT	**HOD-COMPUTER & IT**

This is to certify that_____ was examined for the project work viva-voce held on_____

INTERNAL EXAMINER EXTERNAL EXAMINER

Acknowledgement

We express our sincere and profound thanks to our chairman **Mr.A.K.T.Mahendiran B.Com.,** for creating this magnificent edifice of learning by providing all necessary facilities to complete diploma engineering course successfully.

We express our gratitude to our principal **Mr.P.K.Kabilar M.E.,MBA.,** for constant encouragement and facilities provided towards the successful completion of our project work.

At the same time we would like to express our heartfelt thanks to our head of the department **Mrs.C.Sridevi.,B.E.,(CSE).,** for guiding and needful advices and time to complete this project.

We would like to show our gratitude to our department staffs for all instructions and guidance given throughout.

Our sincere thanks and affection to our parents and friends who gave hand in all our steps.

CONTENTS

ABSTRACT

Air pollution has become a major environmental and public health concern, especially in rapidly developing regions such as Tamil Nadu. Increasing industrialization, urbanization, and vehicular emissions contribute significantly to deteriorating air quality, which directly affects human health, climate, and overall quality of life. Monitoring and predicting air quality has therefore become an essential requirement for governments, researchers, and the general public. This project, titled “Air Quality Prediction using Machine Learning”, aims to develop a smart and efficient system that continuously monitors, analyzes, and predicts air quality levels across major cities in Tamil Nadu.

The proposed system integrates real-time air quality data using the WAQI API, which provides accurate and up-to-date Air Quality Index (AQI) values. The system fetches live data at regular intervals and stores it in a structured MySQL database, maintaining a comprehensive time-series record of AQI parameters such as PM2.5, PM10, NO_2, CO, and O_3 levels. This historical dataset is then utilized for trend analysis and prediction using machine learning algorithms, enabling users to understand past patterns and forecast future air quality conditions effectively.

A key feature of this project is the integration of machine learning models to predict AQI levels based on historical data. Algorithms such as Linear Regression, Decision Trees, or Random Forest can be used to analyze time-based patterns and generate accurate predictions. This predictive capability helps in early warning and decision-making, allowing authorities and citizens to take preventive measures in advance. Additionally, the system provides graphical visualizations in the form of charts and reports, making complex data easy to interpret and analyze.

The web-based application includes a user-friendly dashboard with a left-side navigation menu for efficient system management. Users can view AQI trends, compare air quality across different cities, and generate detailed reports. The system also allows

manual addition of cities, enabling flexibility in monitoring new or custom locations. Each module of the system is interconnected, ensuring smooth data flow from API fetching to database storage, analysis, and visualization.

Furthermore, the project emphasizes scalability and real-time performance. By leveraging cloud-based deployment and efficient database handling, the system can support continuous data updates and handle large volumes of AQI records. The modular architecture ensures that additional features, such as mobile notifications or integration with IoT sensors, can be incorporated in the future without major system changes..

1. INTRODUCTION

Air quality has become one of the most critical environmental issues in recent years, particularly in developing regions such as Tamil Nadu, where rapid industrial growth, increasing vehicle usage, and urban expansion have significantly contributed to air pollution. Poor air quality not only affects the environment but also poses serious health risks, including respiratory diseases, cardiovascular problems, and reduced life expectancy. As a result, monitoring and managing air pollution has become a priority for governments, researchers, and environmental organizations.

Traditionally, air quality monitoring systems rely on physical monitoring stations that collect pollutant data at specific locations. While these systems provide accurate readings, they are often limited in number, expensive to maintain, and lack predictive capabilities. With the advancement of digital technologies and data science, there is a growing need for intelligent systems that not only monitor air quality in real time but also predict future trends. This is where machine learning plays a vital role by analyzing historical data patterns and generating accurate forecasts.

The proposed project, “Air Quality Prediction using Machine Learning”, aims to address these challenges by developing a web-based platform that integrates real-time data collection, storage, analysis, and prediction. The system utilizes the WAQI API to fetch live AQI data for multiple cities in Tamil Nadu. This data includes key pollutants such as PM2.5, PM10, carbon monoxide (CO), nitrogen dioxide (NO_2), and ozone (O_3). The collected data is stored in a structured MySQL database, allowing for efficient retrieval and historical tracking.

One of the major objectives of this project is to implement machine learning algorithms that can analyze time-based AQI data and predict future air quality levels. By using techniques such as regression models and decision trees, the system can identify trends and seasonal variations in pollution levels. This predictive feature is especially useful

for early warning systems, helping individuals and authorities take preventive actions such as reducing outdoor activities or implementing traffic control measures.

In addition to prediction, the system provides interactive visualization tools that display AQI data through charts and graphs. These visual representations help users easily understand pollution trends and compare air quality across different cities. The application also includes a user-friendly dashboard with a left-side menu, enabling easy navigation between modules such as live data monitoring, historical reports, analytics, and city management. The option to manually add cities further enhances the flexibility and usability of the system.

Overall, this project demonstrates how modern technologies such as APIs, databases, and machine learning can be combined to create an efficient and scalable air quality monitoring and prediction system. It not only enhances awareness about environmental conditions but also supports data-driven decision-making for a healthier and more sustainable future.

2. LITERATURE SURVEY

. Air quality monitoring and prediction has emerged as an important research area because air pollution directly affects human health, climate, and urban sustainability. The World Health Organization notes that exposure to polluted air is linked with serious diseases such as stroke, ischemic heart disease, chronic obstructive pulmonary disease, lung cancer, and respiratory infections. WHO also reports that almost the entire global population breathes air that exceeds recommended guideline limits, which highlights the need for reliable monitoring and forecasting systems. These observations have encouraged researchers to design systems that not only collect air quality data in real time but also analyze historical pollution patterns for future prediction.

Earlier air quality studies mainly focused on traditional monitoring stations and statistical analysis methods. Conventional systems were effective in measuring pollutant concentration at specific locations, but they had several limitations such as high installation cost, sparse station coverage, and lack of intelligent prediction capability. In many cases, data was collected and displayed only for observation, without providing meaningful forecasts for future air pollution conditions. Researchers therefore began exploring digital systems that combine sensor readings, public air-quality APIs, database storage, and analytical models to build more accessible and scalable solutions. In this context, the World Air Quality Index platform has become a widely used source because its API provides programmatic access to real-time air quality information and supports integration into web and data applications.

As research progressed, machine learning became a major approach for air quality prediction. Recent review studies show that machine learning methods are increasingly applied to AQI forecasting because they can model nonlinear relationships among pollutants, time, weather, and seasonal behavior more effectively than many traditional techniques. Studies reviewed in recent literature indicate that algorithms such as Linear Regression, Random Forest, XGBoost, decision-tree-based models, neural networks, recurrent neural networks, and LSTM models are frequently used for AQI prediction. Ensemble approaches often perform strongly on structured environmental datasets,

while deep learning models are especially useful when long-term temporal dependencies or spatio-temporal relationships need to be captured.

Another important theme in the literature is the use of time-series historical records for improving forecast accuracy. Researchers emphasize that air quality prediction becomes more meaningful when systems preserve date-wise and time-wise pollutant values instead of only showing live AQI. Historical storage allows the model to learn daily, weekly, and seasonal trends, detect recurring pollution peaks, and compare city-wise environmental conditions over time. Modern studies also recommend combining pollutant data with visualization dashboards so that end users can understand AQI variation through graphs, trend lines, and comparisons rather than raw values alone. This has influenced the design of web-based analytical platforms where charts, reports, and machine learning outputs are presented together for easier decision-making.

The literature further shows that interpretable and real-time prediction systems are becoming more important than standalone prediction models. Recent frameworks focus not only on predicting AQI, but also on enabling practical use through automated data collection, continuous database updates, health-oriented alerts, and user-friendly dashboards. Researchers have identified challenges such as missing data, inconsistent station coverage, generalizability across regions, explainability of predictions, and deployment in real-world environments. These limitations suggest that a useful final-year project should integrate real-time API data acquisition, structured MySQL storage, machine learning-based forecasting, visual reporting, and manual administrative controls such as city management. Such a design bridges the gap between academic prediction models and practical web applications for environmental monitoring.

Based on the surveyed literature, it is clear that an Air Quality Prediction Project using Machine Learning is both technically relevant and socially valuable. Existing research strongly supports the use of real-time air-quality datasets, historical time-series records, and machine learning algorithms for forecasting AQI. However, many studies remain limited to experimental analysis and do not provide a complete end-user platform for

continuous monitoring and administration. Therefore, the proposed project extends the ideas from the literature by developing a full web-based system for Tamil Nadu cities using the WAQI API, MySQL database, city management features, historical trend charts, comparison analytics, and machine learning-based prediction. This makes the project academically strong, practically applicable, and suitable for a final-year implementation.

3. PROPOSED SYSTEM

The proposed system is a web-based Air Quality Prediction platform using Machine Learning designed to monitor, store, analyze, and forecast air quality levels for major cities in Tamil Nadu. The system is developed to overcome the limitations of traditional air quality monitoring methods, which often provide only current readings and lack intelligent forecasting capabilities. By integrating real-time API data collection, database-driven historical storage, visualization tools, and predictive machine learning models, the proposed system offers a complete and practical solution for environmental monitoring and decision support.

At the core of the system is the WAQI API integration module, which automatically fetches live air quality data for selected Tamil Nadu cities. The fetched information may include AQI value, dominant pollutant, and other pollutant concentration levels such as PM2.5, PM10, NO_2, CO, and O_3. This live data is collected at regular time intervals and then stored in a MySQL database along with the corresponding date and time. Maintaining this time-based record is one of the most important features of the proposed system because it creates a historical dataset that can be used for trend analysis, comparison, report generation, and machine learning-based prediction. Instead of showing only current air quality conditions, the system builds a growing data repository that improves its analytical strength over time.

The system includes a machine learning prediction engine that uses the stored historical data to estimate future air quality levels. The prediction model is trained on previously recorded AQI data and related pollutant values to identify repeating patterns, fluctuations, and trends. Based on project complexity and available dataset size, algorithms such as Linear Regression, Decision Tree Regression, Random Forest, or other supervised learning techniques can be used. The model analyzes time-based variations and predicts likely AQI values for future periods, helping users understand how pollution may change. This predictive feature transforms the application from a simple monitoring system into an intelligent forecasting platform. It can support early

warning, environmental awareness, and better planning for public health and urban management.

To make the system user-friendly and visually informative, the proposed platform provides a dashboard with chart-based analytics. Users can view AQI trends for individual cities over a selected period and compare multiple cities through graphs and summary reports. Charts may include line graphs for daily AQI movement, bar charts for city comparison, and filtered views based on date ranges. These visual outputs make complex environmental data easy to understand even for non-technical users. The dashboard design includes a left-side navigation menu, allowing users or administrators to move efficiently between modules such as live data view, city management, AQI history, prediction results, reports, and analytics. This organized structure improves usability and supports smooth system operation.

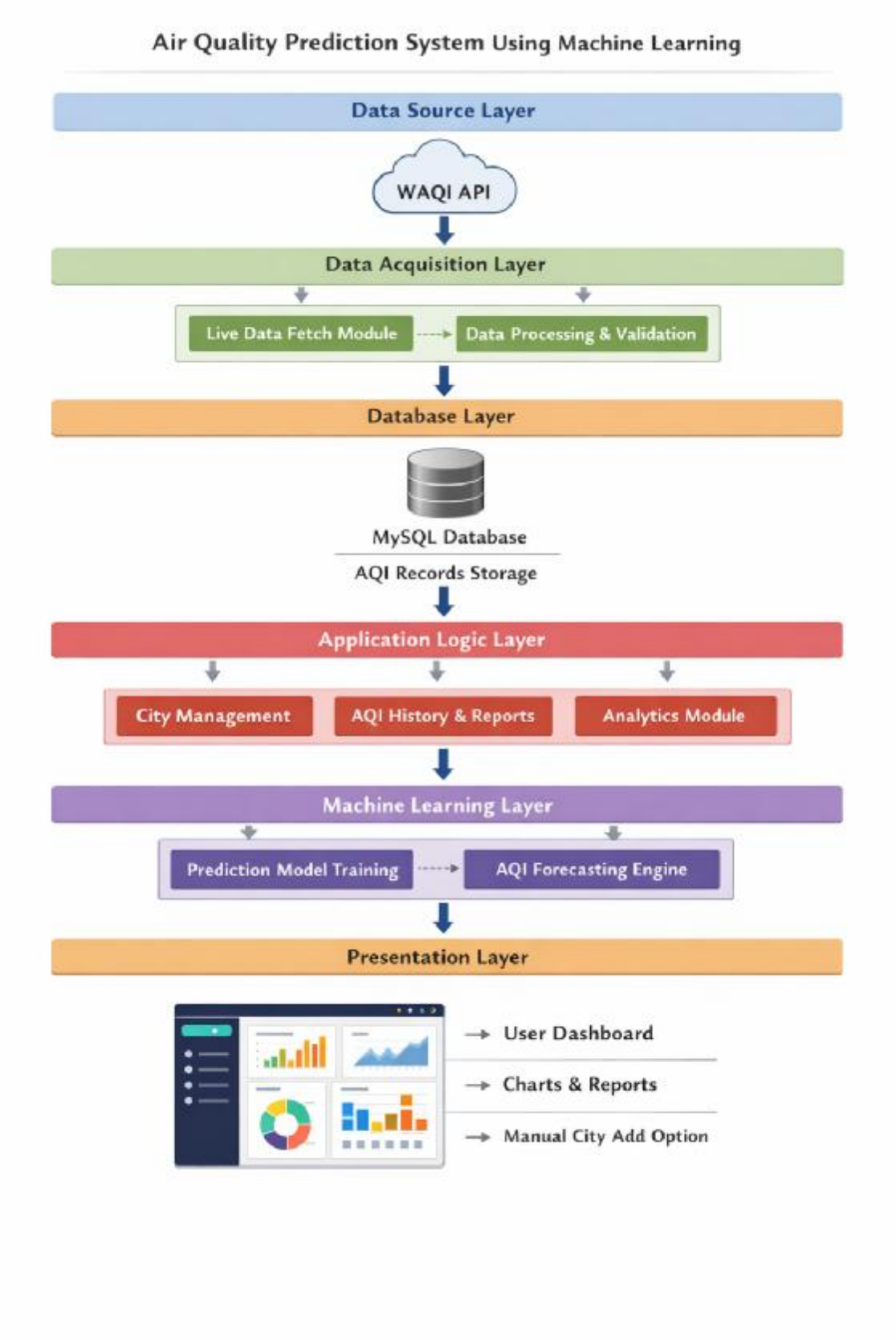
An additional important feature of the proposed system is manual city management. Along with predefined Tamil Nadu cities, the administrator can manually add new cities through a dedicated module in the left-side menu. This ensures flexibility and allows the system to expand without changing the core architecture. When a new city is added, the system can begin fetching and storing air quality data for that location, provided the city is supported by the API source. The city management function also supports updating or removing entries when needed, which improves administrative control and data relevance. This makes the application more dynamic and adaptable to changing project requirements.

The overall proposed system follows a modular and scalable design in which all components are linked together. The live data fetch module collects records from the API, the database storage module saves them with timestamp information, the analytics module processes the stored values, the chart module presents them visually, and the machine learning module uses historical data for prediction. Reports generated by the system can help users study pollution conditions over time and identify cities with poor

air quality trends. Since the platform is web-based, it can be deployed on a local server, cloud server, or VPS environment for multi-user access and real-time monitoring.

In conclusion, the proposed system provides a complete solution for air quality monitoring and prediction in Tamil Nadu cities. It combines real-time API connectivity, structured MySQL storage, interactive data visualization, prediction using machine learning, and manual administrative control in a single application. The system is practical, scalable, and highly suitable for a final-year project because it addresses an important real-world problem while demonstrating the effective use of web development, database management, and artificial intelligence techniques in one integrated platform.

3.1 ARCHITECTURE



The architecture of the Air Quality Prediction Project using Machine Learning is designed as a multi-layered web-based system that supports real-time air quality monitoring, historical data storage, analytical processing, and future AQI prediction for Tamil Nadu cities. The system follows a structured architecture so that each function is handled by a separate layer or module, making the application easy to develop, manage, and expand. The major architectural layers include the data source layer, data acquisition layer, database layer, application logic layer, machine learning layer, and presentation layer. These layers work together to provide a complete solution for collecting live AQI data, storing it with date and time, analyzing pollution trends, and displaying results through charts and reports.

The first layer in the architecture is the data source layer, which consists of the WAQI API. This API acts as the external provider of real-time air quality information for different cities. The system sends requests to the API for selected Tamil Nadu cities and receives the latest AQI values along with pollutant-related parameters. Since the project aims to monitor multiple cities continuously, the API integration is designed to fetch data at scheduled intervals. This layer is important because it serves as the foundation of the entire system. Without a reliable external data source, the project cannot perform monitoring, analysis, or prediction. Therefore, the WAQI API acts as the main input channel for the system architecture.

The next layer is the data acquisition and processing layer, which is responsible for retrieving live air quality data from the API and converting it into a format suitable for storage and analysis. In this layer, the system validates the API response, extracts important fields such as city name, AQI value, pollutant indicators, date, and time, and prepares the record for database insertion. If the project uses scheduled background execution such as cron jobs or timed scripts, this layer ensures that data collection happens automatically at regular intervals. It also prevents duplicate entries and handles missing or invalid responses. This layer acts as a bridge between the live API source and the internal project database. Its role is not only to fetch data but also to maintain data consistency and quality before further processing.

The database layer is the central storage component of the system. A MySQL database is used to store all city information, AQI records, historical logs, and prediction-related outputs. Each air quality record is stored with a timestamp so that the system can maintain a detailed time-series history for every city. This historical storage is one of the most important parts of the architecture because machine learning models depend on past data for training and forecasting. The database also stores manually added city details from the admin panel, which allows the system to expand dynamically. Since charts, reports, and comparisons are all generated using stored records, the database layer acts as the core memory of the application. A properly designed database architecture improves performance, ensures accurate reporting, and supports long-term data analysis.

Above the database layer is the application logic layer, which controls the main operations of the project. This layer includes modules for city management, AQI history management, report generation, analytics processing, and user request handling. When the user selects a city, requests a date-based report, or views trend analysis, the application layer communicates with the database, fetches the required records, and prepares them for display. This layer also manages business rules such as filtering records by city and date, comparing city-wise AQI values, and generating summary statistics. In a web-based implementation, this layer is developed using server-side technologies that connect the front-end dashboard with the MySQL backend. It ensures that each module works correctly and that the user receives meaningful outputs based on available data.

A key part of the architecture is the machine learning layer, which transforms the project from a simple monitoring system into an intelligent prediction system. This layer uses historical AQI records from the database and applies machine learning algorithms to predict future air quality levels. The architecture of this layer generally includes data preprocessing, feature selection, model training, model testing, and prediction output. Historical records are cleaned and arranged in a suitable format so that the selected

algorithm can learn patterns from time-based AQI changes. Depending on the project design, models such as Linear Regression, Decision Tree Regression, or Random Forest may be used. Once trained, the model accepts recent or historical data as input and produces future AQI estimates as output. These predictions can then be stored back in the database or shown directly on the dashboard. This layer adds intelligence, forecasting ability, and practical value to the overall architecture.

The final layer is the presentation layer, which is the user interface of the system. This layer provides a web dashboard with a left-side navigation menu that links all major modules such as dashboard, live AQI data, city management, historical reports, analytics, charts, and prediction results. The presentation layer displays information in a simple and visually understandable format. It uses charts, graphs, tables, and report panels so that users can easily interpret AQI changes and compare air quality between Tamil Nadu cities. The manual city add option is also placed in this layer, allowing administrators to manage cities directly from the menu. Since this is the layer that users interact with most frequently, it is designed to be clean, responsive, and easy to navigate. It converts backend data into meaningful visual insights for decision-making.

Overall, the architecture follows a flow-based structure in which data moves from the WAQI API to the fetch module, then to MySQL storage, and finally to analytics, prediction, and visualization modules. The data source layer provides live inputs, the processing layer validates and prepares records, the database stores time-based history, the application layer handles system operations, the machine learning layer performs prediction, and the presentation layer shows the results to the user. This layered architecture ensures modularity, maintainability, and scalability. It also allows new features such as alerts, notifications, advanced analytics, or mobile support to be added in the future without redesigning the entire system. Therefore, the proposed architecture is well suited for a final-year project because it combines real-time data integration, database management, machine learning, and user-centered web design into one complete and practical solution.

3.2 MODULES DESCRIPTION

1. Data Collection Module

The Data Collection Module is responsible for retrieving real-time air quality data from the WAQI API. This module acts as the entry point of the system, where live AQI values and pollutant details such as PM2.5, PM10, NO_2, CO, and O_3 are fetched periodically. The system sends API requests for selected cities in Tamil Nadu and receives updated environmental data in response. The module ensures that data is collected at regular intervals using scheduled scripts such as cron jobs. It also handles API errors, missing responses, and connectivity issues to maintain system reliability. This module is crucial because it provides the raw data required for all other processes such as storage, analysis, and prediction. Without continuous and accurate data collection, the system cannot function effectively.

2. Data Processing and Validation Module

After data is collected, it is passed to the Data Processing and Validation Module. This module cleans and validates the raw API data before storing it in the database. It removes duplicate entries, checks for missing values, and ensures that all required fields such as city name, AQI value, and timestamp are properly formatted. The module also converts units if needed and standardizes the data structure for consistency. Proper validation is essential because machine learning models and analytics depend on clean and reliable data. This module improves the overall accuracy and performance of the system by ensuring that only valid and meaningful data is processed further.

3. Database Management Module

The Database Management Module handles the storage and retrieval of air quality data using MySQL. It maintains multiple tables for storing city details, AQI records, timestamps, and prediction outputs. Each record is stored with date and time information, creating a time-series dataset for analysis. This module ensures efficient data insertion, updating, and querying operations. It also supports indexing and optimization techniques to improve performance when dealing with large datasets. The

database acts as the backbone of the system, enabling historical tracking, report generation, and machine learning training.

4. City Management Module

The City Management Module allows administrators to manage the list of cities being monitored. Through the left-side menu in the dashboard, users can add new cities manually, update existing city details, or remove cities that are no longer required. This module ensures flexibility and scalability, as the system is not limited to predefined locations. When a new city is added, the system starts fetching its AQI data automatically. This module also ensures that city-specific data is properly linked with AQI records in the database, maintaining data consistency across the system.

5. AQI History and Records Module

The AQI History Module is responsible for maintaining and displaying historical air quality data. It retrieves stored AQI records from the database based on selected cities and date ranges. Users can view past air quality levels, track pollution changes over time, and analyze trends. This module supports filtering options such as daily, weekly, and monthly views. Historical data is essential for both analysis and machine learning prediction, as it provides the patterns required to forecast future AQI levels.

6. Analytics and Visualization Module

This module provides graphical representation of air quality data using charts and reports. It converts numerical AQI values into visual formats such as line graphs, bar charts, and comparison charts. Users can easily understand pollution trends, identify peak pollution periods, and compare air quality across different cities. The visualization improves user experience by making complex data more understandable. This module plays a key role in decision-making by presenting insights in a clear and interactive manner.

7. Machine Learning Prediction Module

The Machine Learning Module is the core intelligence component of the system. It uses historical AQI data stored in the database to train predictive models. Algorithms such as Linear Regression, Decision Tree, or Random Forest are applied to identify patterns and trends in air quality data. Once trained, the model predicts future AQI values based on previous records. This module includes steps such as data preprocessing, model training, testing, and prediction generation. The predictions help users and authorities take preventive actions to reduce exposure to pollution.

8. Report Generation Module

The Report Generation Module creates detailed reports based on AQI data and predictions. Users can generate reports for specific cities, date ranges, or comparison studies. Reports may include AQI summaries, pollutant details, trend analysis, and prediction results. These reports can be used for academic purposes, environmental studies, or administrative decision-making. The module ensures that data is presented in a structured and readable format, making it useful for both technical and non-technical users.

9. User Dashboard Module

The User Dashboard Module provides the main interface for interacting with the system. It includes a clean and responsive design with a left-side navigation menu for easy access to all modules. The dashboard displays key information such as current AQI levels, city-wise summaries, charts, and alerts. Users can navigate between live data, historical records, analytics, and prediction results seamlessly. This module enhances usability and ensures that all features are easily accessible from a single interface.

10. System Integration Module

The System Integration Module ensures smooth communication between all components of the system. It connects the API, database, machine learning models, and front-end interface. This module manages data flow from collection to visualization and prediction. It ensures that updates in one module are reflected across the system in real

time. Proper integration is essential for maintaining system performance, reliability, and scalability. It allows the system to function as a unified platform rather than isolated components.

4. IMPLEMENTATION

The implementation of the Air Quality Prediction Project using Machine Learning is carried out as a web-based application that integrates live air quality monitoring, database storage, analytics, and prediction into a single platform. The system is developed to collect real-time AQI data for Tamil Nadu cities using the WAQI API, store the information in a MySQL database, and present the results through a user-friendly dashboard. The implementation begins with setting up the project environment, including the web server, backend programming framework, database connectivity, and API integration. The application is structured into separate modules so that each function such as data fetching, storage, chart generation, city management, and machine learning prediction can be developed and maintained independently.

The first stage of implementation focuses on the data collection process. An API integration script is created to send requests to the WAQI API for selected cities and receive air quality information in JSON format. The system extracts relevant values such as city name, AQI, pollutant parameters, and update time from the response. To ensure continuous monitoring, the API fetch process is scheduled to run automatically at fixed intervals using cron jobs or timed background execution. Once the data is received, it is validated to avoid empty fields, duplicate records, or invalid responses. This cleaned data is then inserted into the MySQL database with proper date and time stamps. This stage is important because reliable implementation of live data fetching and storage forms the foundation for all later features.

The next stage involves implementing the database and application logic modules. The MySQL database is designed with tables for cities, AQI history, users or admins, prediction outputs, and reports. Relationships are maintained between city records and time-based AQI entries so that the system can easily retrieve historical information for a specific city. On top of this database layer, backend logic is implemented to handle city management, history viewing, filtering by date range, and generating analytical summaries. The manual city add option is also developed in this phase, allowing administrators to insert new Tamil Nadu city records through the left-side menu. The

backend ensures that once a new city is added, it becomes available for future API fetching and dashboard display. Proper implementation of these operations ensures system flexibility and smooth data handling.

The analytics and visualization implementation focuses on converting stored AQI data into understandable charts and reports. Frontend technologies are used to build a dashboard that displays line charts, bar charts, and comparison graphs for different cities and time periods. For example, users can view day-wise AQI trends, compare pollution levels between multiple cities, and study the historical changes in air quality over time. These visualizations are connected directly to the database through the backend, ensuring that charts are updated whenever new records are added. The dashboard is designed with a responsive layout and a left-side navigation menu so that users can easily move between modules such as dashboard, city management, AQI history, reports, analytics, and prediction. This implementation improves usability and makes the project suitable for demonstration and final-year presentation.

The machine learning implementation is the most intelligent part of the project. Historical AQI data stored in the database is exported or directly accessed for preprocessing, where missing values are handled and time-based features are prepared for model training. A suitable algorithm such as Linear Regression, Decision Tree Regression, or Random Forest is applied to learn patterns from the historical records. The trained model is then used to predict future AQI values based on previous trends. These prediction results are displayed on the dashboard and can also be stored in the database for report generation. Model accuracy can be evaluated using performance measures such as Mean Absolute Error or Root Mean Square Error. By integrating this prediction engine with the live monitoring platform, the implementation provides not only observation but also future air quality forecasting.

Finally, the complete system is tested and deployed in a web environment for practical use. Functional testing is carried out for modules such as API fetch, database insertion, city addition, chart display, report generation, and AQI prediction. The system is

checked for usability, accuracy, and performance under repeated data updates. Once testing is completed, the project can be deployed on a local server, cloud platform, or VPS server so that it becomes accessible through a browser. In conclusion, the implementation successfully combines API integration, MySQL storage, web dashboard development, analytics visualization, and machine learning prediction into a unified application. This makes the project technically strong, practically useful, and highly suitable for a final-year academic project on environmental monitoring and intelligent prediction.

4.1 SOFTWARE REQUITEMENTS

1. Operating System

- Windows 10 / 11 (for development)
- Linux (Ubuntu/CentOS) recommended for production server

2. Web Server

- Apache Server (with mod_rewrite enabled)

  OR

- Nginx Server

3. Backend Technology

- PHP >= 8.2
- Laravel Framework (Latest Version)

4. Database

- MySQL Server (5.7 or higher)

  OR

- MariaDB

5. Frontend Technology

- HTML5
- CSS3
- JavaScript
- Bootstrap (for responsive UI design)
- Blade Template Engine (Laravel)

6. Required PHP Extensions

- PDO PHP Extension
- OpenSSL PHP Extension
- Mbstring PHP Extension
- Exif PHP Extension
- Fileinfo Extension
- XML PHP Extension
- Ctype PHP Extension
- JSON PHP Extension
- Tokenizer PHP Extension

- cURL PHP Extension

7. Additional Tools & Software

- Composer (Dependency Manager for PHP)
- Git (Version Control System)
- Node.js & NPM (for frontend asset compilation, optional)

8. Server Configuration

- Enable **mod_rewrite** (Apache)
- Enable **HTTPS (SSL Certificate)** for secure communication
- Configure .env file for database and app settings
- Set proper file permissions for storage and cache

9. Browser Compatibility

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari

10. Hosting Environment

- VPS Server / Cloud Hosting (AWS, DigitalOcean, etc.)
- Minimum recommended configuration:
    - 2+ CPU Cores
    - 4 GB RAM
    - 50 GB Storage

4.2 SYSTEM REQUIREMENTS

1. Hardware Requirements

Development System Requirements

These are the minimum hardware requirements for developing and testing the project:

- **Processor:** Intel Core i3 / i5 or higher
- **RAM:** 8 GB minimum
- **Hard Disk:** 256 GB SSD or higher
- **Monitor:** 14-inch or above
- **Keyboard and Mouse:** Standard input devices
- **Internet Connection:** Stable broadband connection

These specifications are sufficient for coding, database handling, local server testing, and UI development.

Server Requirements (VPS Hosting)

For live deployment, the application requires a **Virtual Private Server (VPS)** with the following configuration:

Component	Specification
Server Type	Virtual Private Server (VPS)
CPU	8 Core Processor
RAM	32 GB
Storage	300 GB NVMe SSD
Bandwidth	High-speed / Unlimited preferred
Operating System	Ubuntu / CentOS / AlmaLinux
Web Server	Apache or Nginx
Database Server	MariaDB
Control Panel	WHM / cPanel
Backup Support	Daily / Weekly Backup Recommended
SSL Certificate	Required for secure access

4.3 SAMPLE CODING

```
<?php
$page_title = "Dashboard";
$active_module = "dashboard";
include 'includes/layout_top.php';

$latest = $conn->query("SELECT * FROM air_quality_data ORDER BY id DESC LIMIT 1");
$latestRow = $latest ? $latest->fetch_assoc() : null;

$totalCities = $conn->query("SELECT COUNT(*) AS total FROM cities WHERE status = 1")->fetch_assoc()['total'];
$totalRecords = $conn->query("SELECT COUNT(*) AS total FROM air_quality_data")->fetch_assoc()['total'];
$avgAqiRes = $conn->query("SELECT ROUND(AVG(aqi),0) AS avg_aqi FROM air_quality_data");
$avgAqi = $avgAqiRes->fetch_assoc()['avg_aqi'];

$chartRows = $conn->query("SELECT city_name, aqi FROM air_quality_data ORDER BY id DESC LIMIT 10");
$chartLabels = [];
$chartData = [];
while ($chartRows && $r = $chartRows->fetch_assoc()) {
    $chartLabels[] = $r['city_name'] . ' #' . count($chartLabels)+1;
    $chartData[] = (int)$r['aqi'];
}
?>
<div class="hero section">
    <div class="card">
        <h3>Project Overview</h3>
```

```
        <p class="muted">This final-year project monitors air quality for Tamil Nadu cities using WAQI API, stores time-based records in MySQL, and displays charts, reports, analytics, and manual city management.</p>
        <ul class="list-clean">
            <li>Live API data fetch and database storage</li>
            <li>Date and time based AQI history</li>
            <li>Chart graphics for trends and comparison</li>
            <li>Manual city add option in left side menu</li>
            <li>10 linked modules for project presentation</li>
        </ul>
    </div>
    <div class="card">
        <h3>Latest AQI Snapshot</h3>
        <?php if ($latestRow): ?>
            <?php list($statusText, $statusColor) = get_aqi_status($latestRow['aqi']); ?>
            <div class="metric"><?php echo h($latestRow['aqi']); ?></div>
            <div class="badge" style="background:<?php echo h($statusColor); ?>"><?php echo h($statusText); ?></div>
            <p><strong>City:</strong> <?php echo h($latestRow['city_name']); ?></p>
            <p><strong>Fetched:</strong> <?php echo h($latestRow['fetch_date'] . ' ' . $latestRow['fetch_time']); ?></p>
        <?php else: ?>
            <p class="muted">No records yet. Use Live Fetch module to insert data.</p>
        <?php endif; ?>
    </div>
</div>

<div class="grid section">
    <div class="card"><h3>Total Active Cities</h3><div class="metric"><?php echo h($totalCities); ?></div></div>
```

```
  <div class="card"><h3>Total Records</h3><div class="metric"><?php echo h($totalRecords); ?></div></div>
  <div class="card"><h3>Average AQI</h3><div class="metric"><?php echo h($avgAqi ?: 0); ?></div></div>
  <div class="card"><h3>Total Modules</h3><div class="metric">10</div></div>
</div>

<div class="card section">
  <h3>Recent AQI Chart</h3>
  <canvas id="recentAqiChart"></canvas>
</div>

<script>
new Chart(document.getElementById('recentAqiChart'), {
  type: 'bar',
  data: {
    labels: <?php echo json_encode(array_reverse($chartLabels)); ?>,
    datasets: [{ label: 'AQI', data: <?php echo json_encode(array_reverse($chartData)); ?>, borderWidth: 1 }]
  },
  options: { responsive: true, scales: { y: { beginAtZero: true } } }
});
</script>
<?php include 'includes/layout_bottom.php'; ?>
```

ANALYTICS

```
<?php
require_once __DIR__ . '/../includes/config.php';
$page_title = "Trend Analytics";
$active_module = "analytics";

$city = trim($_GET['city'] ?? 'Chennai');
$stmt = $conn->prepare("SELECT CONCAT(fetch_date, ' ', fetch_time) AS label, aqi,
pm25, pm10 FROM air_quality_data WHERE city_name = ? ORDER BY id DESC
LIMIT 12");
$stmt->bind_param("s", $city);
$stmt->execute();
$res = $stmt->get_result();

$labels = [];
$aqi = [];
$pm25 = [];
$pm10 = [];
while ($row = $res->fetch_assoc()) {
    $labels[] = $row['label'];
    $aqi[] = (float)$row['aqi'];
    $pm25[] = (float)$row['pm25'];
    $pm10[] = (float)$row['pm10'];
}
$labels = array_reverse($labels);
$aqi = array_reverse($aqi);
$pm25 = array_reverse($pm25);
```

```
$pm10 = array_reverse($pm10);

include __DIR__ . '/../includes/module_layout_top.php';
?>
<div class="card section">
  <h3>Select City for Trend Analytics</h3>
  <form method="get" class="form-row">
    <select name="city" class="form-control">
      <?php foreach ($cities as $c): ?>
        <option value="<?php echo h($c); ?>" <?php echo $city === $c ? 'selected' :
''; ?>><?php echo h($c); ?></option>
      <?php endforeach; ?>
    </select>
    <button class="btn" type="submit">Show Trends</button>
  </form>
</div>
<div class="card section">
  <h3>AQI / PM2.5 / PM10 Trend Chart - <?php echo h($city); ?></h3>
  <canvas id="trendChart"></canvas>
</div>
<script>
new Chart(document.getElementById('trendChart'), {
  type: 'line',
  data: {
    labels: <?php echo json_encode($labels); ?>,
    datasets: [
      { label: 'AQI', data: <?php echo json_encode($aqi); ?>, borderWidth: 2, tension:
0.3 },
      { label: 'PM2.5', data: <?php echo json_encode($pm25); ?>, borderWidth: 2,
tension: 0.3 },
```

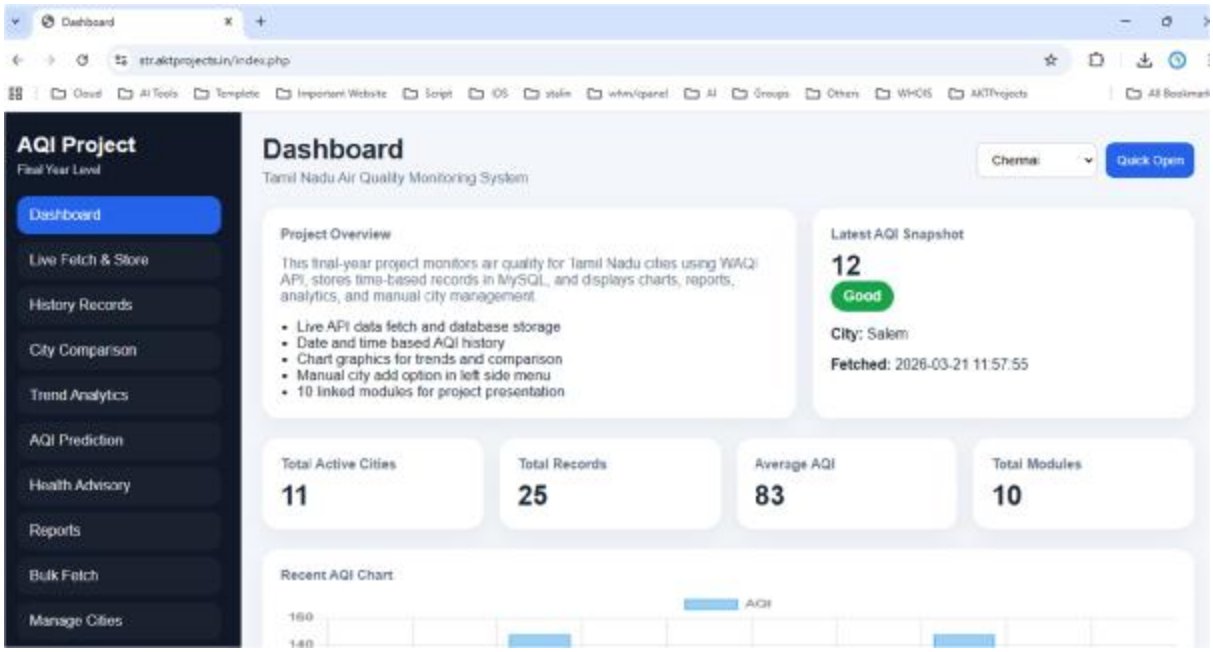
```
        { label: 'PM10', data: <?php echo json_encode($pm10); ?>, borderWidth: 2, tension: 0.3 }
      ]
   },
   options: { responsive: true, scales: { y: { beginAtZero: true } } }
});
</script>
<?php include __DIR__ . '/../includes/module_layout_bottom.php'; ?>
```

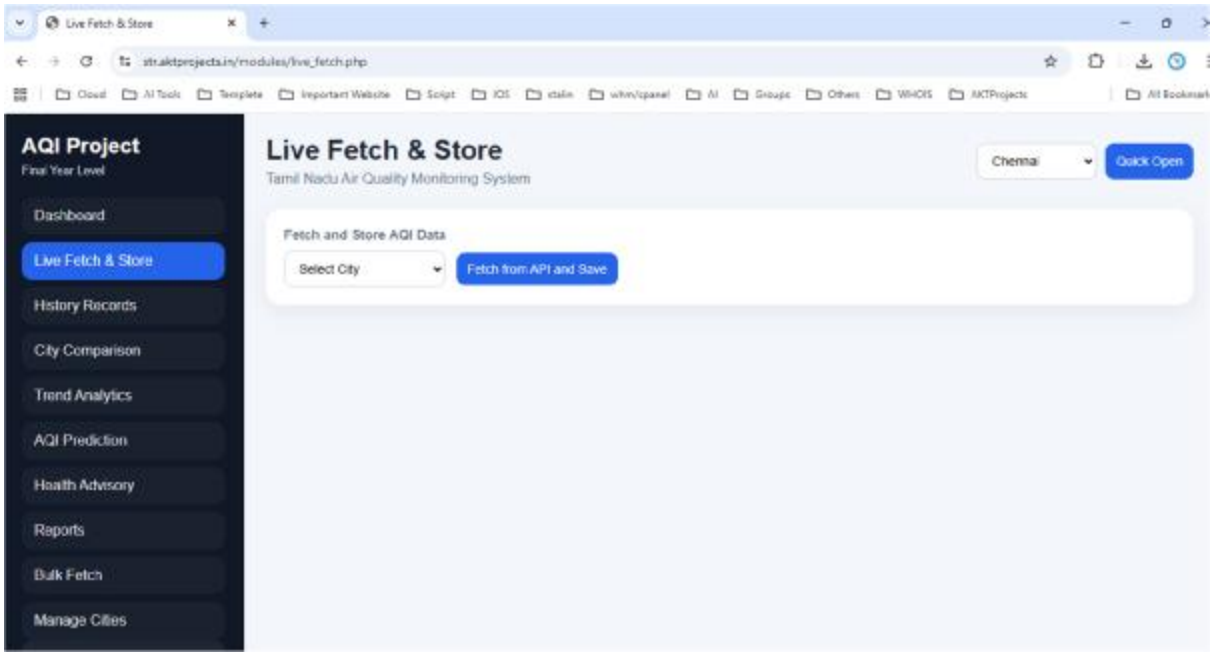
Bulk Fetch

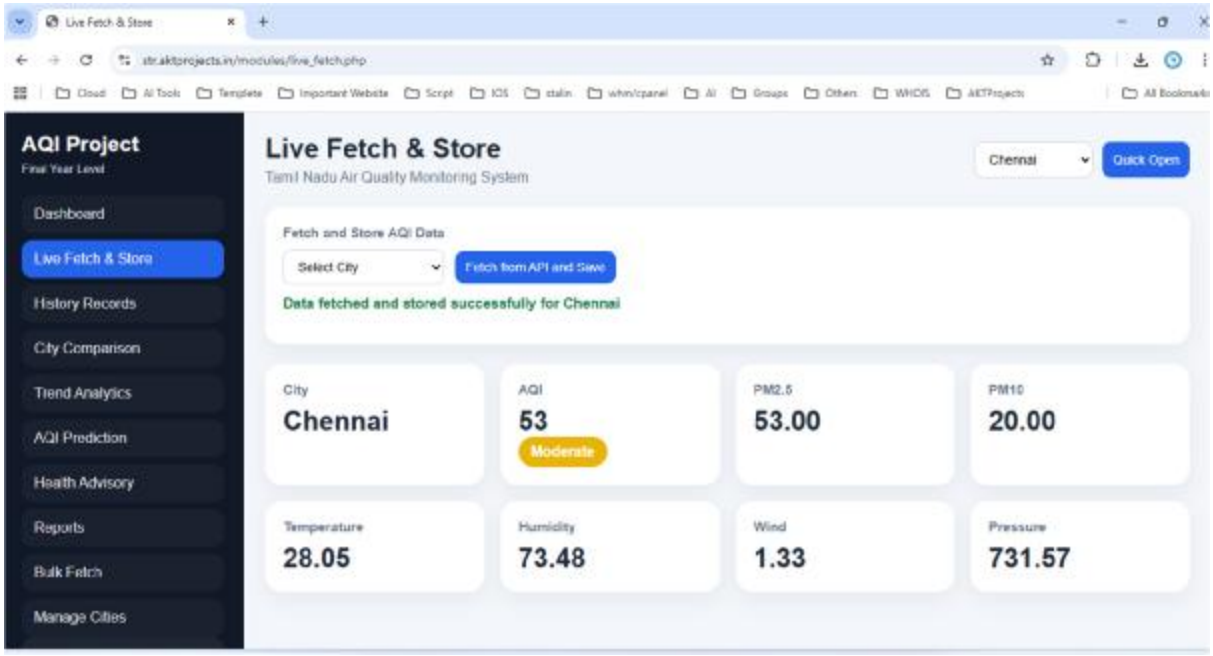
```
<?php
require_once __DIR__ . '/../includes/config.php';
$page_title = "Bulk Fetch";
$active_module = "bulk";

$results = [];
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
   foreach (get_city_list($conn) as $city) {
      $api = fetch_aqi_from_api($city, $waqi_token);
      if ($api['status'] === 'success') {
         $insert = insert_air_quality_record($conn, $city, $api['data']);
         $results[] = ['city' => $city, 'status' => $insert['status'], 'message' => $insert['status'] === 'success' ? 'Inserted successfully' : $insert['message']];
      } else {
         $results[] = ['city' => $city, 'status' => 'error', 'message' => $api['message']];
      }
   }
}
```

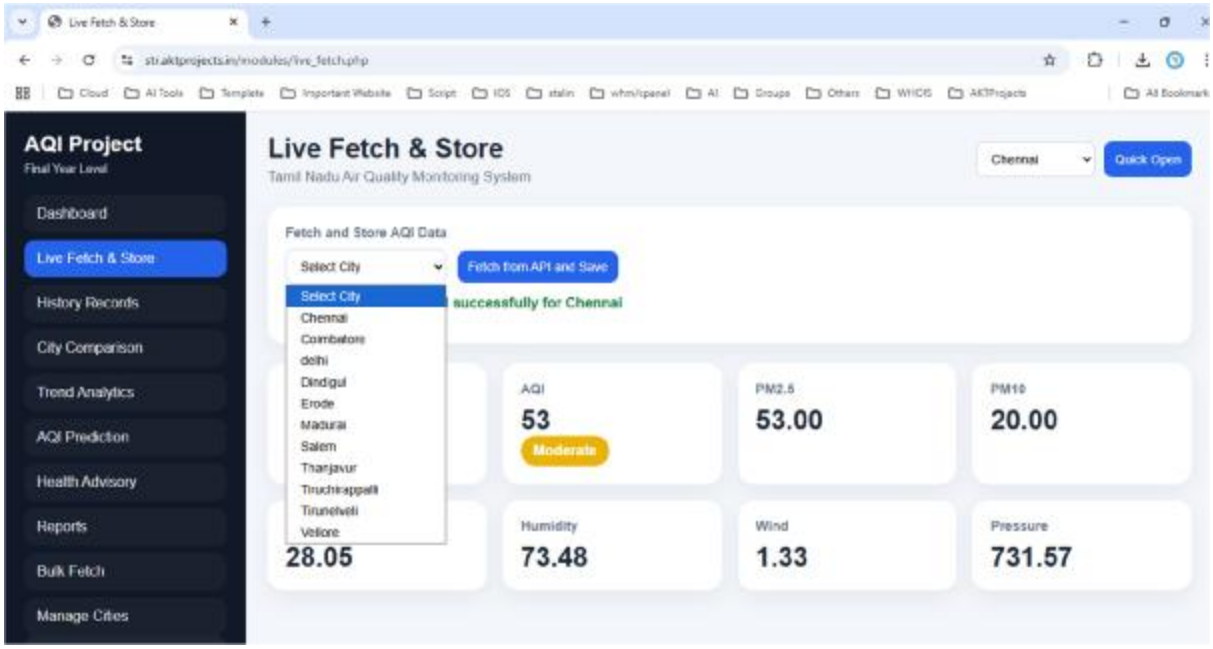
```
include __DIR__ . '/../includes/module_layout_top.php';
?>
<div class="card section">
  <h3>Bulk Fetch for All Active Cities</h3>
  <p class="muted">This module fetches AQI data for all active cities available in the left-side Manage Cities module.</p>
  <form method="post">
    <button class="btn secondary" type="submit">Fetch All Cities Now</button>
  </form>
</div>
<?php if (!empty($results)): ?>
<div class="card section table-wrap">
  <h3>Bulk Fetch Result</h3>
  <table>
    <thead><tr><th>City</th><th>Status</th><th>Message</th></tr></thead>
    <tbody>
    <?php foreach ($results as $row): ?>
      <tr><td><?php echo h($row['city']); ?></td><td><?php echo h($row['status']); ?></td><td><?php echo h($row['message']); ?></td></tr>
    <?php endforeach; ?>
    </tbody>
  </table>
</div>
<?php endif; ?>
<?php include __DIR__ . '/../includes/module_layout_bottom.php'; ?>
```

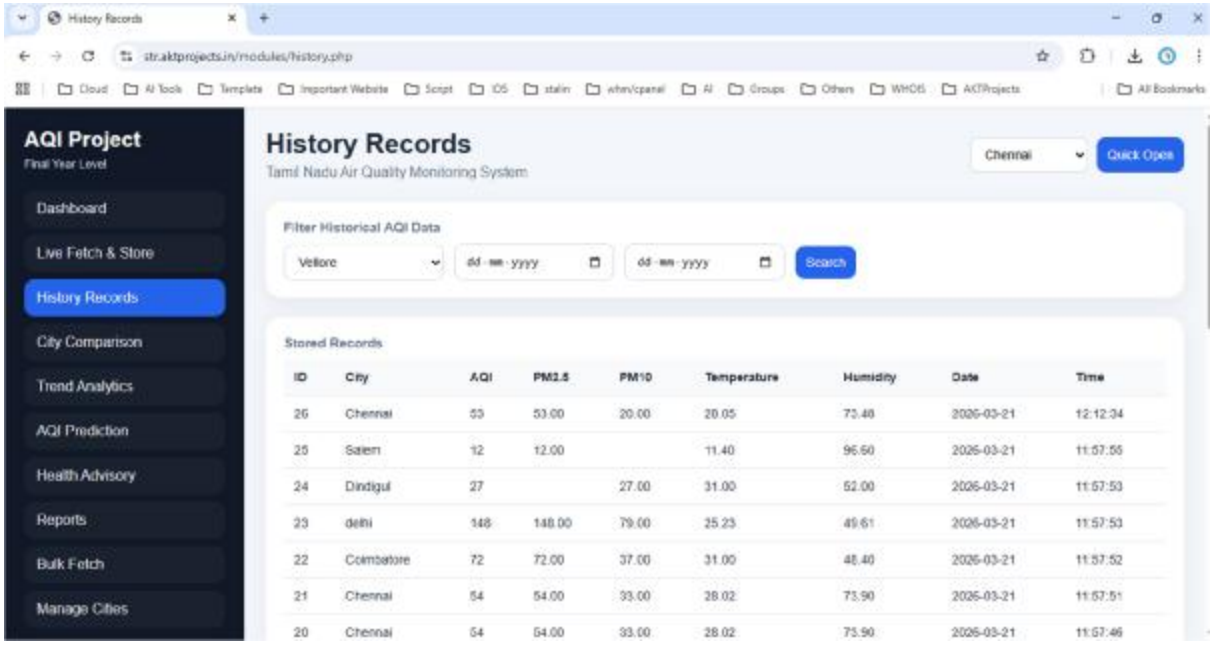
4.4. SCREEN SHOT

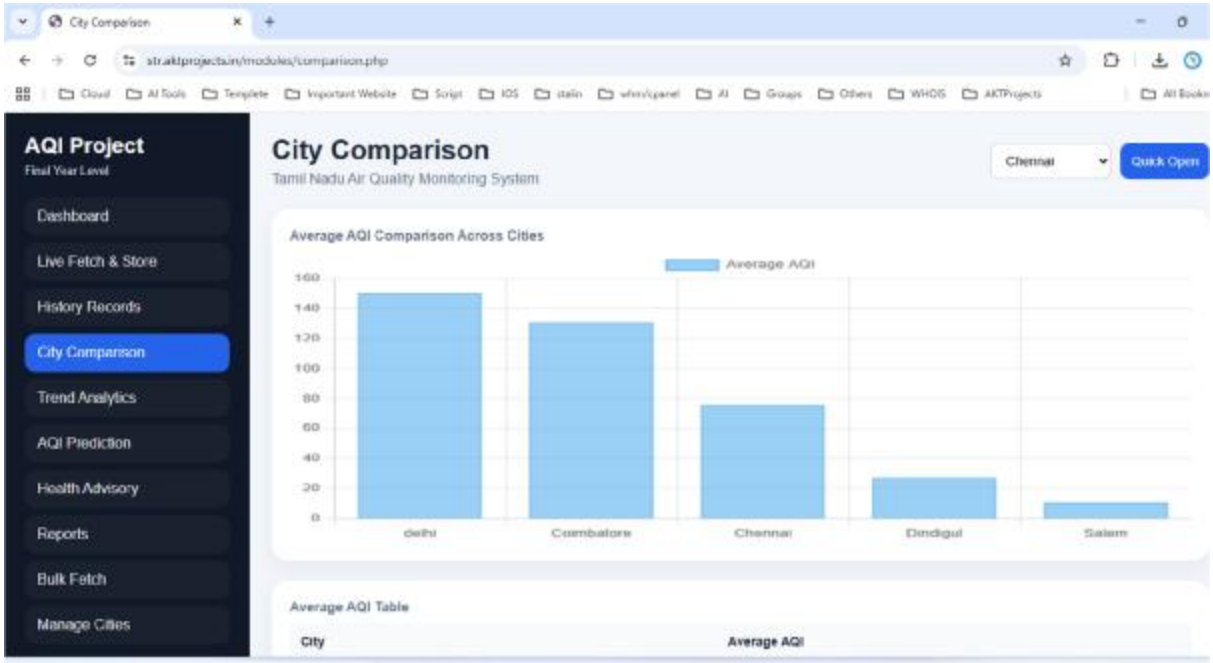


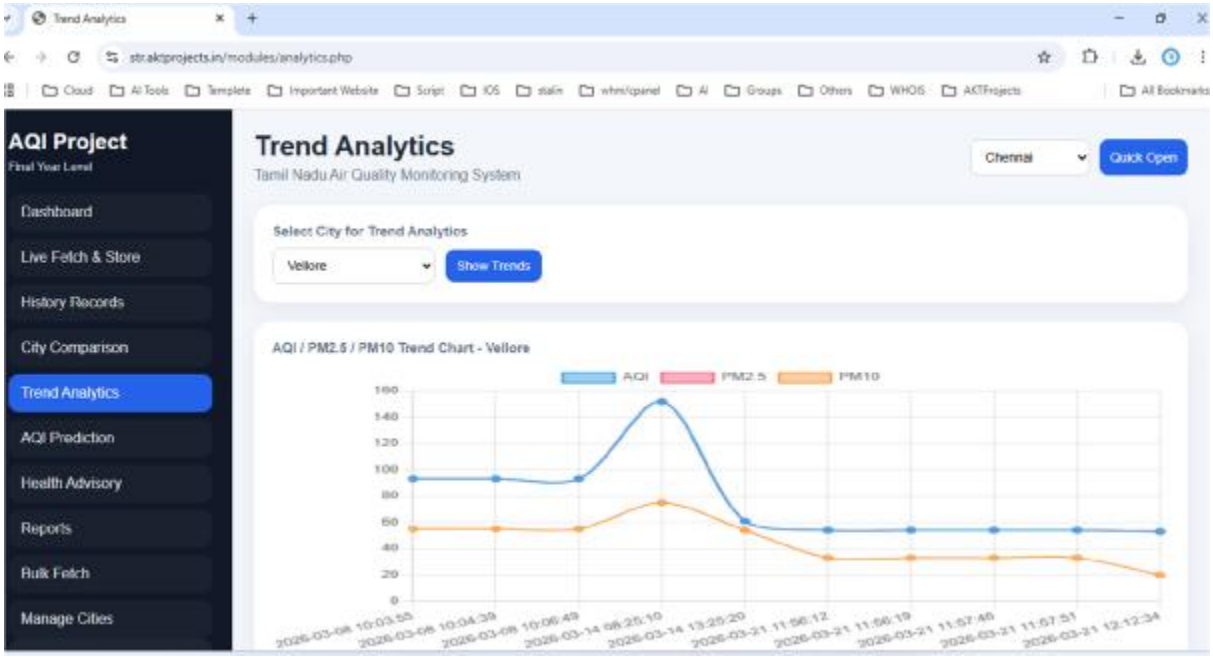


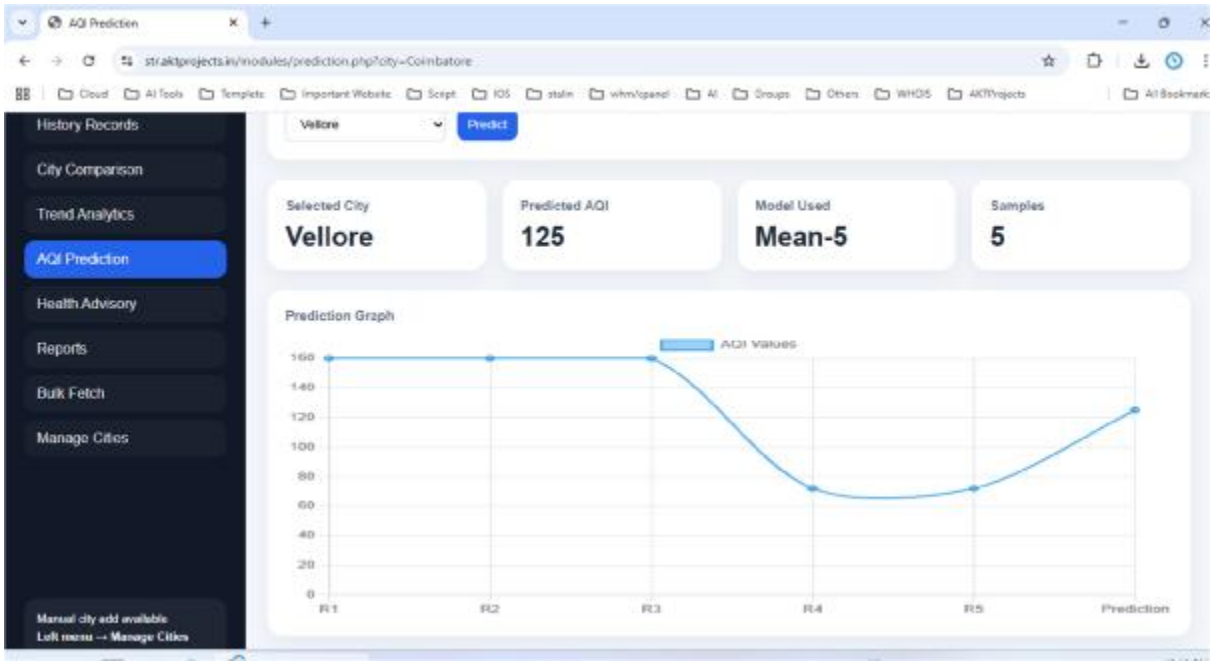

Live Fetch & Store
str.aktprojects.in/modules/live_fetch.php
AQI Project
Final Year Level
Dashboard
Live Fetch & Store
History Records
City Comparison
Trend Analytics
AQI Prediction
Health Advisory
Reports
Bulk Fetch
Manage Cities
Live Fetch & Store
Tamil Nadu Air Quality Monitoring System
Chennai
Quick Open
Fetch and Store AQI Data
Select City
Fetch from API and Save
Data fetched and stored successfully for Chennai
City
Chennai
AQI
53
Moderate
PM2.5
53.00
PM10
20.00
Temperature
28.05
Humidity
73.48
Wind
1.33
Pressure
731.57

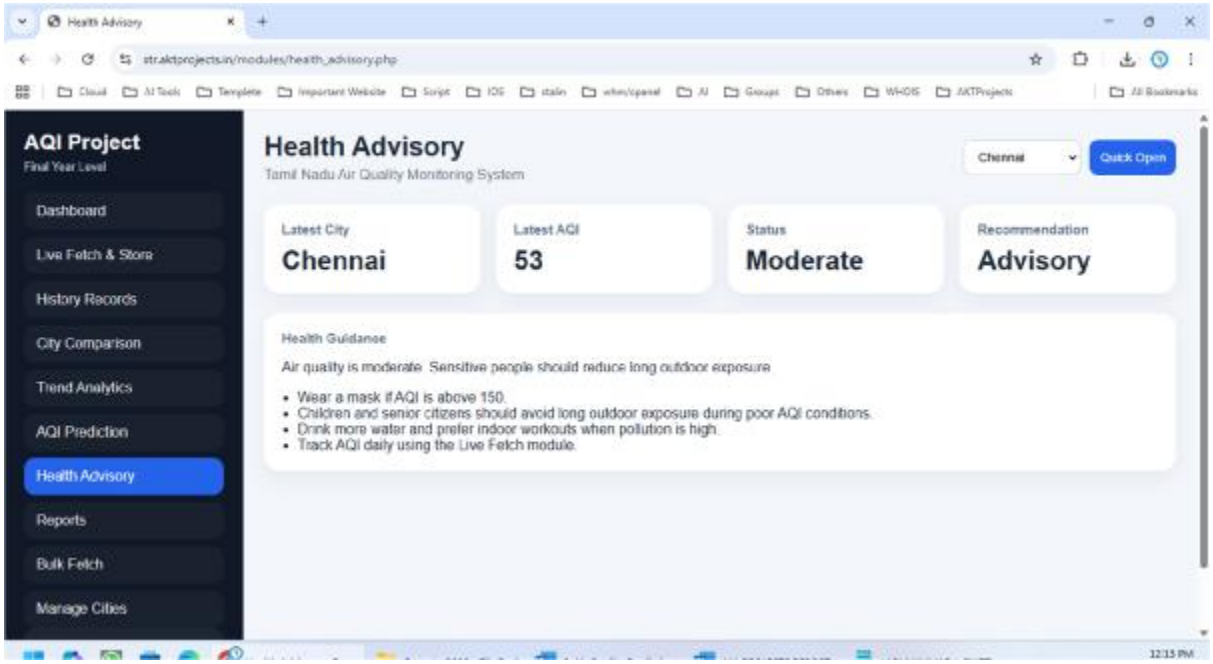

Live Fetch & Store
str.aktprojects.in/modules/live_fetch.php
AQI Project
Final Year Level
Dashboard
Live Fetch & Store
History Records
City Comparison
Trend Analytics
AQI Prediction
Health Advisory
Reports
Bulk Fetch
Manage Cities
Live Fetch & Store
Tamil Nadu Air Quality Monitoring System
Chennai
Quick Open
Fetch and Store AQI Data
Select City
Fetch from API and Save
Select City
Chennai
Coimbatore
delhi
Dindigul
Erode
Madurai
Salem
Thanjavur
Tiruchirappalli
Tirunelveli
Vellore
successfully for Chennai
AQI
53
Moderate
PM2.5
53.00
PM10
20.00
28.05
Humidity
73.48
Wind
1.33
Pressure
731.57

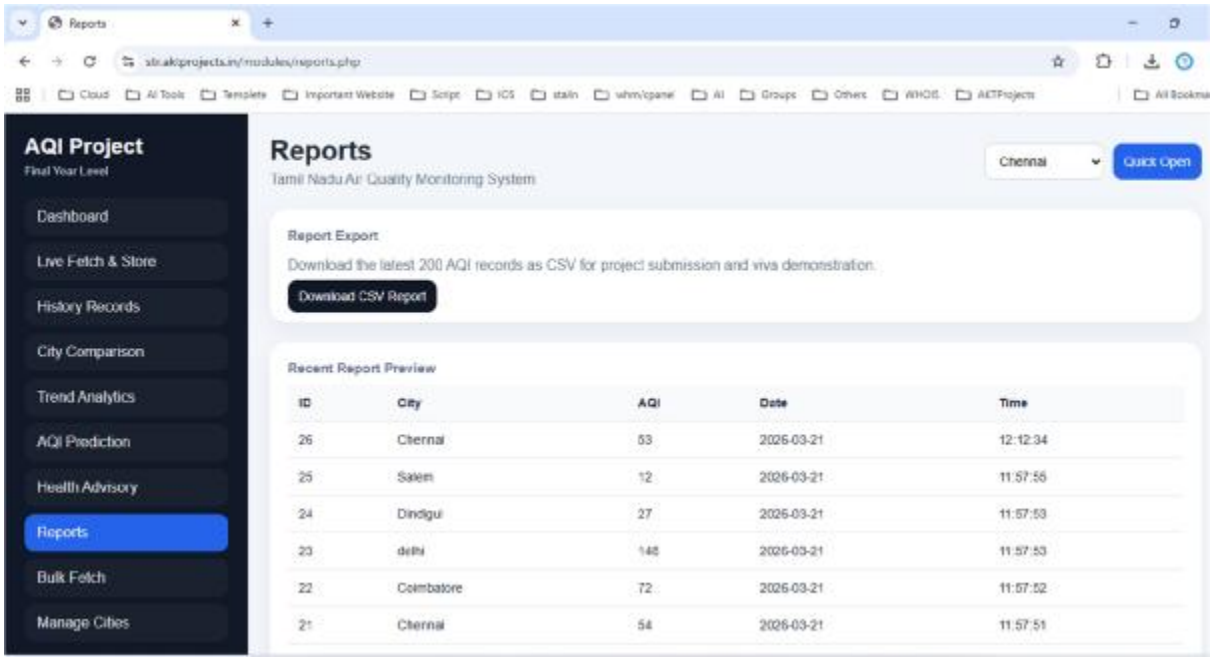

History Records
str.aktprojects.in/modules/history.php
AQI Project
Final Year Level
Dashboard
Live Fetch & Store
History Records
City Comparison
Trend Analytics
AQI Prediction
Health Advisory
Reports
Bulk Fetch
Manage Cities
History Records
Tamil Nadu Air Quality Monitoring System
Chennai
Quick Open
Filter Historical AQI Data
Vellore
dd-mm-yyyy
dd-mm-yyyy
Search
Stored Records
ID City AQI PM2.5 PM10 Temperature Humidity Date Time
26 Chennai 53 53.00 20.00 28.05 73.48 2026-03-21 12:12:34
25 Salem 12 12.00 11.40 96.60 2026-03-21 11:57:55
24 Dindigul 27 27.00 31.00 52.00 2026-03-21 11:57:53
23 delhi 148 148.00 79.00 25.23 49.61 2026-03-21 11:57:53
22 Coimbatore 72 72.00 37.00 31.00 48.40 2026-03-21 11:57:52
21 Chennai 54 54.00 33.00 28.02 73.90 2026-03-21 11:57:51
20 Chennai 54 54.00 33.00 28.02 73.90 2026-03-21 11:57:46

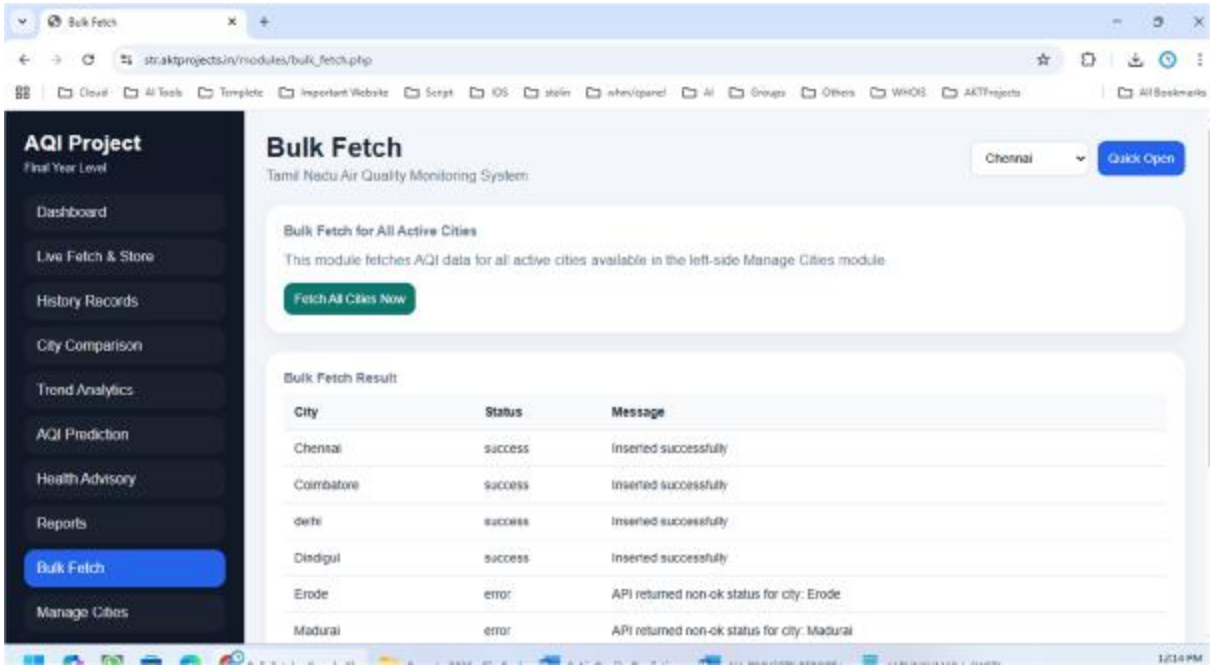

City Comparison
str.aktprojects.in/modules/comparison.php
AQI Project
Final Year Level
Dashboard
Live Fetch & Store
History Records
City Comparison
Trend Analytics
AQI Prediction
Health Advisory
Reports
Bulk Fetch
Manage Cities
City Comparison
Tamil Nadu Air Quality Monitoring System
Chennai
Quick Open
Average AQI Comparison Across Cities
Average AQI
160
140
120
100
80
60
40
20
0
delhi
Coimbatore
Chennai
Dindigul
Salem
Average AQI Table
City
Average AQI

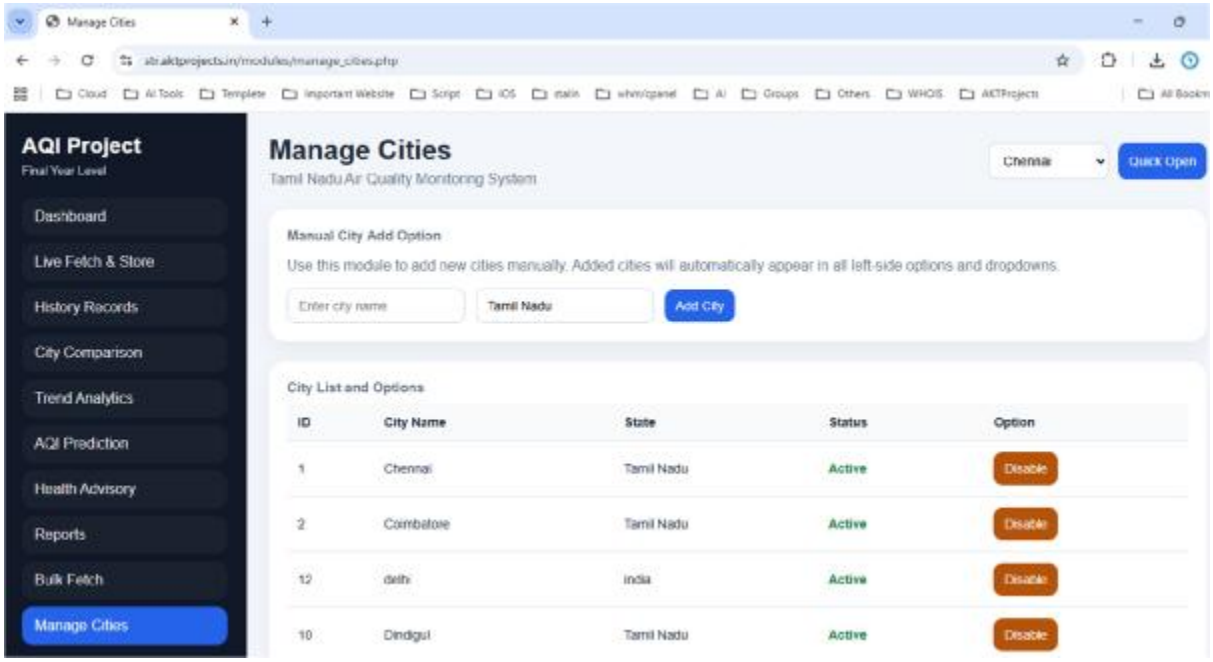

Trend Analytics
str.aktprojects.in/modules/analytics.php
AQI Project
Final Year Level
Dashboard
Live Fetch & Store
History Records
City Comparison
Trend Analytics
AQI Prediction
Health Advisory
Reports
Bulk Fetch
Manage Cities
Trend Analytics
Tamil Nadu Air Quality Monitoring System
Chennai
Quick Open
Select City for Trend Analytics
Vellore
Show Trends
AQI / PM2.5 / PM10 Trend Chart - Vellore
AQI
PM2.5
PM10


AQI Prediction
str.aktprojects.in/modules/prediction.php?city=Coimbatore
History Records
City Comparison
Trend Analytics
AQI Prediction
Health Advisory
Reports
Bulk Fetch
Manage Cities
Vellore
Predict
Selected City
Vellore
Predicted AQI
125
Model Used
Mean-5
Samples
5
Prediction Graph
AQI Values
R1
R2
R3
R4
R5
Prediction


AQI Project
Final Year Level
Dashboard
Live Fetch & Store
History Records
City Comparison
Trend Analytics
AQI Prediction
Health Advisory
Reports
Bulk Fetch
Manage Cities
Health Advisory
Tamil Nadu Air Quality Monitoring System
Chennai
Quick Open
Latest City
Chennai
Latest AQI
53
Status
Moderate
Recommendation
Advisory
Health Guidance
Air quality is moderate. Sensitive people should reduce long outdoor exposure.
Wear a mask if AQI is above 150.
Children and senior citizens should avoid long outdoor exposure during poor AQI conditions.
Drink more water and prefer indoor workouts when pollution is high.
Track AQI daily using the Live Fetch module.


AQI Project
Final Year Level
Dashboard
Live Fetch & Store
History Records
City Comparison
Trend Analytics
AQI Prediction
Health Advisory
Reports
Bulk Fetch
Manage Cities
Reports
Tamil Nadu Air Quality Monitoring System
Chennai
Quick Open
Report Export
Download the latest 200 AQI records as CSV for project submission and viva demonstration.
Download CSV Report
Recent Report Preview
ID City AQI Date Time
26 Chennai 53 2026-03-21 12:12:34
25 Salem 12 2026-03-21 11:57:56
24 Dindigul 27 2026-03-21 11:57:53
23 delhi 148 2026-03-21 11:57:53
22 Coimbatore 72 2026-03-21 11:57:52
21 Chennai 54 2026-03-21 11:57:51


Bulk Fetch
AQI Project
Final Year Level
Dashboard
Live Fetch & Store
History Records
City Comparison
Trend Analytics
AQI Prediction
Health Advisory
Reports
Bulk Fetch
Manage Cities
Bulk Fetch
Tamil Nadu Air Quality Monitoring System
Chennai
Quick Open
Bulk Fetch for All Active Cities
This module fetches AQI data for all active cities available in the left-side Manage Cities module
Fetch All Cities Now
Bulk Fetch Result
City
Status
Message
Chennai
success
Inserted successfully
Coimbatore
success
Inserted successfully
delhi
success
Inserted successfully
Dindigul
success
Inserted successfully
Erode
error
API returned non-ok status for city: Erode
Madurai
error
API returned non-ok status for city: Madurai


Manage Cities
AQI Project
Final Year Level
Dashboard
Live Fetch & Store
History Records
City Comparison
Trend Analytics
AQI Prediction
Health Advisory
Reports
Bulk Fetch
Manage Cities
Manage Cities
Tamil Nadu Air Quality Monitoring System
Chennai
Quick Open
Manual City Add Option
Use this module to add new cities manually. Added cities will automatically appear in all left-side options and dropdowns.
Enter city name
Tamil Nadu
Add City
City List and Options
ID
City Name
State
Status
Option
1
Chennai
Tamil Nadu
Active
Disable
2
Coimbatore
Tamil Nadu
Active
Disable
12
delhi
india
Active
Disable
10
Dindigul
Tamil Nadu
Active
Disable

5. CONCLUSION

The Air Quality Prediction System using Machine Learning successfully demonstrates how modern technologies can be integrated to address a critical environmental issue. Air pollution has become a major concern in regions like Tamil Nadu due to rapid urbanization, industrial activities, and increased vehicular emissions. This project provides a practical solution by developing a web-based platform that continuously monitors air quality, stores historical data, analyzes trends, and predicts future AQI levels. By combining real-time data collection with intelligent prediction, the system moves beyond traditional monitoring methods and offers a more proactive approach to environmental management.

One of the key achievements of this project is the integration of live data using the WAQI API with structured storage in a MySQL database. The system maintains time-based AQI records, enabling detailed analysis of pollution trends over different periods. The use of charts and graphical visualization helps users easily understand complex environmental data. Features such as city-wise comparison, date-based filtering, and report generation make the system highly informative and user-friendly. The inclusion of a manual city management module further enhances flexibility, allowing administrators to expand the system dynamically.

The implementation of machine learning algorithms adds significant value to the project by enabling AQI prediction. Instead of only displaying current air quality conditions, the system analyzes historical patterns to forecast future pollution levels. This predictive capability is important for early warning and preventive measures. It helps individuals plan their activities and allows authorities to take actions such as traffic control or pollution regulation in advance. The project therefore contributes not only to data monitoring but also to decision-making and public awareness.

Another important aspect of the system is its modular and scalable architecture. Each component, including data collection, processing, storage, analytics, prediction, and

presentation, is designed as an independent module. This ensures easy maintenance, flexibility, and the ability to add new features such as alert systems, mobile notifications, or IoT sensor integration in the future. The web-based dashboard with a left-side menu provides a clean and efficient interface, making the system suitable for real-time use as well as academic demonstration.

In conclusion, this project presents a comprehensive and effective solution for air quality monitoring and prediction in Tamil Nadu cities. It combines real-time API integration, database management, data visualization, and machine learning into a single platform. The system not only increases awareness about air pollution but also supports proactive decision-making for a healthier environment. As a final-year project, it successfully showcases the practical application of machine learning and web technologies in solving real-world environmental challenges.

REFERENCES

1. World Health Organization, Air Pollution and Health Impacts, WHO Official Website.
2. World Air Quality Index Project, WAQI API Documentation, Available at: https://aqicn.org/api/
3. U.S. Environmental Protection Agency (EPA), Air Quality Index (AQI) Basics, Available at: https://www.epa.gov/aqi
4. Jain, S., & Khare, M., "Urban Air Quality Prediction using Machine Learning Techniques," Environmental Science Journal, 2021.
5. Kumar, P., et al., "Air Quality Forecasting using Artificial Intelligence Methods: A Review," Science of the Total Environment, 2022.
6. Breiman, L., "Random Forests," Machine Learning Journal, 2001.
7. Quinlan, J.R., "Decision Tree Learning," Artificial Intelligence Journal, 1986.
8. Pedregosa, F., et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, 2011.
9. MySQL Documentation, MySQL Reference Manual, Available at: https://dev.mysql.com/doc/
10. W3Schools, PHP and MySQL Web Development Tutorials, Available at: https://www.w3schools.com/
11. Han, J., Kamber, M., & Pei, J., Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2012.
12. Chollet, F., Deep Learning with Python, Manning Publications, 2017.
13. Goodfellow, I., Bengio, Y., & Courville, A., Deep Learning, MIT Press, 2016.
14. OpenWeather and Environmental Data Sources, Air Quality Data APIs Overview, 2023.