|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SYMCA Research Project Report Logbook 2025-26 TERM I** | | | | | |
|  | | | |  |  |
| **Project Registration** | **SR.NO** | **TASK** | **DATE** | **COMPLETION DATE** | **SIGN WITH REMARK** |
| 1 | Student Group allocation | **01 Aug 2025** |  |  |
| 2 | Project Allocation |  |  |
| 3 | Project Abstract Upload |  |  |
| **STAGE-1** | **SR.NO** | **Progress Report I** | **DATE** |  |  |
| 1 | Aim, Objective and Scope of Project | **09/ Aug 2025** |  |  |
| 2 | H/W S/W Requirement, Tools, Human Efforts in Hours |  |  |
| 3 | System Overview, Proposed System and Expected outcome, Architecture & initial phase of design, |  |  |
| 4 | Title of Research Paper, Literature Review, Research Gap |  |  |
| 5 | Objective of Research Paper, Abstract of Research Paper |  |  |
| **STAGE-2** | **SR.NO** | **Progress Report 2** | **DATE** |  |  |
| 6 | ERD & UML Diagram(Activity & State Transistion (As Per Problem statement) | **23 Aug 2025** |  |  |
| 7 | Database Design with proper key definition |  |  |
| 8 | Data Dictionary, Flow chart |  |  |
| 9 | Project Plan |  |  |
| 10 | Synopsis of Research Project and Paper |  |  |
| **REVIEW -1** |  | **REVIEW - I** | **30 Aug 2025** |  |  |
| **STAGE-3** | **SR.NO** | **Progress Report III** | **DATE** |  |  |
| 11 | Requirement Analysis / Models | **13 Sep 2025** |  |  |
| 12 | Screen Design with proper validations, Reports-Analytical & graphical |  |  |
| 13 | 3-4 Working Models, Publish Research Paper |  |  |
| 14 | 30 – 40% Coding documentation |  |  |
| **STAGE-4** | **SR.NO** | **Progress Report IV** | **DATE** |  |  |
| 15 | Test Data (screenshot) | **27 Oct 2025** |  |  |
| 16 | User Manual |  |  |
| 17 | Bibliography, References |  |  |
| 18 | Published copy of Research Paper & Certificate |  |  |
| 19 | Spiral Copy Submission & Project Execution |  |  |
| **REVIEW -2** |  | **REVIEW - II** | **11 Oct 2025** |  |  |
| **REVIEW -3** |  | **PANEL DEMO PRESENTATION** | **1 Nov. 2025** |  |  |
|  |  | **Final OR** | **1st Week of Nov 2025** |  |  |

**Coordinato**r **HOD MCA (I/C)**

Dr. Rama Bansode Dr. Prakash Kene

**RESEARCH PROJECT   (MCA-II) Abstract (2025-26) TERM I**

|  |  |
| --- | --- |
| **Name Of Student** | Gurjas Singh Gandhi |
| **Roll No. Div** | 52120/B |
| **Project Title** | Air Quality Monitoring and Visualizer App |
| **Technology used** | **Flutter,Node.Js,TensorflowLite,Google Maps API ,Supabase,CPCB AQI Data,ISRO Vedas Satellite Based AQI Data.** |
| **Group No.** | 3 |
| **Group Members with Roll No.** | Nikita Bachute 52102  Pranav Gadewar 52119  Ritwik Rahut 52152 |
| **Keywords** | Air Quality Index (AQI), real-time monitoring, forecasting, satellite data, ground sensors, hyperlocal pollution, health alerts, predictive analytics, environmental data visualization |
| **Name of Guide/Mentor** | Dr. Prakash Kene |
| **Student Email ID** | gurjas\_gandhi\_mca@moderncoe.edu.in |
| **Student Contact Number** | 9922233168 |

**Vayu Drishti- An Air Quality Monitoring and Visualization App**

**Keywords:**Air Quality Index (AQI), real-time monitoring, forecasting, satellite data, ground sensors, hyperlocal pollution, health alerts, predictive analytics, environmental data visualization

**Technology Used:**Streamlit & React js (mobile/web UI), Node.js + Express (backend API), Python (web scraping), TensorFlow Lite (forecasting models), Supabase (PostgreSQL database), Folium Maps and Leaflet.js(mapping/visualization)

**Modules:**

* Data Ingestion & Scraper (Python)
* Backend API & Data Processing (Node.js)
* Forecasting Engine (TensorFlow Lite)
* Frontend UI & Visualization (Streamlit & React js)
* Database (Supabase PostgreSQL)

**Introduction:**Air pollution is a growing concern worldwide due to its impact on health and the environment. We recognize the need to provide accurate, real-time air quality information to all communities, including small towns and rural areas where monitoring is limited. We will develop an app that visualizes air quality data and provides forecasts to help users stay informed and safe.

We will collect data from multiple sources like government sensors, satellite observations, and public platforms to offer a complete, hyperlocal view of pollution. Our system will display pollutant levels such as PM2.5, NO2, and O3 in easy-to-understand maps and charts. This transparency helps users understand air quality changes over time.

To predict future conditions, we will apply machine learning models that forecast air quality up to 72 hours ahead using weather and historical data. These forecasts will enable early warnings for communities, assisting in health protection and planning. The app will also send push notifications with personalized health advice.

Our platform uses modern technology like Streamlit & React js for web, Node.js for backend services, and Python for data scraping to ensure reliability and scalability. This system is designed for easy integration and expansion, allowing developers and policymakers to use the data effectively.

In summary, we will build a powerful, accessible air quality monitoring system combining multiple data streams and AI forecasts. This will empower individuals and authorities with actionable insights into a healthier environment and better quality of life.

PROGRESS REPORT NO. 1

1. **Name of the Student:** Gurjas Singh Gandhi
2. **Class and Div:** SY MCA / B

1. **Date of Submission:**  09/09/2025
2. **Name of the Subject:** Research Project
3. **Name of the Guide:** Dr. Prakash Kene
4. **Team Members: Roll no., Student Name**: Nikita Bachute 52102

Pranav Gadewar 52119

Ritwik Rahut 52152

1. **Title of the Research work :** " Air Quality Monitoring and Visualization"
2. **Details of the work done:** Attach Separate Sheet |(On Page No.2)
3. **Date of Progress report submission**: 09/09/2025

**Details of the Work Done:**

## **1. Aim, Objective and Scope**

**Aim:**To Develop an accessible, real-time air quality visualization and forecasting system for underserved regions in India.

**Objectives:**

* Aggregate and display real-time/historical AQI data from ground and satellite sources.
* Visualize air quality trends, provide 24–72-hour forecasts, and deliver health advisories/alerts.
* Enable users to access hyperlocal air pollution insights via mobile/web app.

**Scope:**It covers data integration, visualization, and automated health notifications for air quality—focusing on small towns and rural areas within India, using open data and modular technology.

## **2. Requirements, Tools, Human Effort**

**Requirements**

* Real-time & historical AQI data collection: Integrating sources like CPCB, ISRO VEDAS, or AQICN APIs or scrapers.
* Mobile/web user interface: Visualizing AQI data, trends, heatmaps, and forecasts.
* Forecasting module: Predicting 24–72-hour air quality using meteorology and AQI history.
* Push notifications: Sending health alerts when pollution spikes are detected.
* Health advisories dashboard: Displaying actionable health advice based on AQI level.
* Backend & database: Efficient data storage, management, and secure API delivery.

**Tools & Technologies**

* Frontend: Streamlit and React (web UI)
* Backend: Node.js + Express, Supabase (PostgreSQL DB), Firebase Cloud Messaging
* Scraper: Python (requests, BeautifulSoup, Pandas for scraper and data processor)
* Visualization:  Leaflet and Folium
* Forecasting: TensorFlowLite, Hugging Face API for ML models
* Testing/Analytics: Jest, Firebase/Google Analytics

**Human Effort (approx):**

* 4 students, each 15–20 hours
* Total ~70 hours (design, coding, testing, docs)

## 

## **3. System Overview & Outcome**

**System Overview**

The system collects real-time air quality data from multiple sources (CPCB, ISRO VEDAS, AQICN) via APIs and scrapers, stores and processes the data in a backend, and delivers it to a mobile/web app. It visualizes current and historical AQI, forecasts pollution levels, and sends health alerts.

**Proposed System**

* Data Collection: Automated web scraping and API integration to ingest air pollution data.
* Backend: Node.js server with database for real-time data management, forecasting ML model integration, and push notification service.
* Frontend: Stream lit Web app and React JS for providing interactive maps, trend graphs, and health advisory dashboards.
* Forecasting: ML models (TensorFlowLite or Hugging Face) predict 24–72-hour AQI trends using weather and historical data.

**Outcome**

* Real-time, hyperlocal air quality visualization across urban and rural regions.
* Predictive pollution forecasts enabling proactive health protection.
* Personalized health recommendations and push alerts for pollution spikes.
* Open, extendable platform supporting integration with external systems and policymaking.

## 

## **4. Research Paper Details**

## **Title:**

## An Integrated Real-Time Air Quality Visualization and Forecasting System for Rural and Urban Areas in India

## **Literature Review:**

## We find that air quality monitoring systems often use IoT sensors, cloud platforms, and machine learning for data collection and forecasting. Studies highlight integrating gas sensors, wireless modules, and satellite data for broader coverage. Real-time visualization and remote alert systems are common, but most focus on urban areas without full integration or personalized health advisories.

## **Research Gap:**

## We observe a gap in coverage for rural and small-town regions. Existing systems rarely combine ground sensors, satellite data, and public sources into a unified real-time platform. We will address this by creating an open, scalable system with AI-driven forecasting and health alerts accessible to underserved communities.

## **5. Research Paper Objective & Abstract**

Objective:  
We aim to develop a real-time air quality monitoring and forecasting system that integrates data from ground sensors, satellite platforms, and public sources. We will provide hyperlocal pollution visualization, short-term forecasts, and personalized health alerts to empower communities and policymakers, especially in underserved areas.

Abstract:  
Air pollution poses significant health risks, particularly in regions with limited monitoring infrastructure. We will build an integrated system combining data from government sensors, ISRO VEDAS satellites, and public platforms to offer real-time air quality visualization and forecasting. Using machine learning, we will predict pollution levels for the next 24–72 hours and send personalized health advisories via a mobile/web app. This system addresses current limitations by providing comprehensive, timely, and accessible air quality information, promoting healthier environments and informed decision-making.

**6. ERD , UML & ER Diagram (As Per Problem statement)**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a diagram

AI-generated content may be incorrect.

Activity Diagram

A diagram of a flowchart

AI-generated content may be incorrect.

7. Database Design with proper key definition

A diagram of a flowchart

AI-generated content may be incorrect.

A diagram of a model

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**8. Data Dictionary & Flow Chart**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a data collection

AI-generated content may be incorrect.

A white sheet with black text

AI-generated content may be incorrect.

A screenshot of a data report

AI-generated content may be incorrect.

A screenshot of a data sheet

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A diagram of a flowchart

AI-generated content may be incorrect.

**9. Project Plan**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Focus Area** | **Key Deliverables** |
| **1. Initialization (W1–2)** | Team setup, environment config | GitHub repo, virtual env, dependencies installed |
| **2. Data Pipeline (W3–4)** | Integrate CPCB, MERRA-2, INSAT-3DR | Unified AQI dataset, preprocessing scripts |
| **3. ML Pipeline (W5–6)** | Random Forest | R² ≥ 0.99 forecasting engine, validation metrics |
| **4. Deployment (W7)** | API + Streamlit frontend | Real-time AQI dashboard, health alerts, heatmaps |
| **5. Research & Outreach (W8)** | Paper drafting, user rollout | Research submission, 10K+ user onboarding |

**10. Synopsis of Research Project and Paper**

**Problem**

* 50% of India lacks air quality coverage
* Existing systems: low accuracy, poor rural data integration

**Solution**

**Multi-Source AI Integration**:

* CPCB (Ground) + MERRA-2 (Weather) + INSAT-3DR (Satellite) → 23 features
* Hybrid ML Pipeline: Random Forest (R² = 0.9994, RMSE = 4.57 AQI)

**Key Highlights**

* 503 stations → 84,504 hourly records (168× data expansion)
* 8.3 s training (346× faster than LSTM)
* <10 ms forecast response time
* 94% rush-hour detection accuracy

**Impact**

✅ Expands monitoring to underserved regions

✅ Real-time AQI forecasts (1–72 hrs)

✅ Personalized health advisories

✅ Interactive dashboard & policy simulator

Team & Timeline

Gurjas Gandhi | Nikita Bachute | Pranav Gadewar | Ritwik Raut

Duration: 6 Months  Target: 10 K Users

**11.Requirement Analysis & Models**

**Requirement Analysis & Model Design — Vayu Drishti**

**System Requirements**

**Data Sources**

* **CPCB:** PM₂.₅, PM₁₀, NO₂, SO₂ (Ground Sensors)
* **MERRA-2:** Temp, Humidity, Wind, Pressure (Meteorological)
* **INSAT-3DR:** AOD, LST, Cloud Fraction (Satellite)

**Functional Requirements**

* Real-time AQI visualization (map + graph)
* 1–72 hr AQI forecasting
* Personalized health alerts
* Threshold-based push notifications
* Data analytics & policy simulation

**Non-Functional Requirements:**

* Accuracy ≥ 99.9% (R² ≥ 0.999)
* Response < 10 ms
* Scalable (> 10 K users)
* Secure & energy-efficient APIs

**A screenshot of a computer

AI-generated content may be incorrect.**

**12. Screen Design with proper validations, Reports-Analytical & graphical**

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

**A screen shot of a computer

AI-generated content may be incorrect.**

**A screenshot of a map

AI-generated content may be incorrect.**

**A screenshot of a graph

AI-generated content may be incorrect.**

**13.** **3-4 Working Models, Publish Research Paper**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a calendar

AI-generated content may be incorrect.**

Research Paper link : [**https://drive.google.com/file/d/1uaLhwyT1dVbLdGUDsadEisyZ2833nuiq/view?usp=sharing**](https://drive.google.com/file/d/1uaLhwyT1dVbLdGUDsadEisyZ2833nuiq/view?usp=sharing)

**14.** **30 – 40% Coding documentation**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Github Profile link:** [**https://github.com/Gurjas2112/Vayu\_Drishti-Real-Time-Air-Quality-Visualizer-App**](https://github.com/Gurjas2112/Vayu_Drishti-Real-Time-Air-Quality-Visualizer-App)

**15.** Test Data (screenshot)

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**16. User Manual**

**Overview**

* **AI-driven app providing real-time AQI, 72-hr forecasts, and health advisories using CPCB, MERRA-2, and INSAT-3DR data.**

**How to Use**

* **Open App → Allow Location Access**
* **Select City / Pin Location**
* **View Live AQI, Forecast Graph, Health Advisory**
* **Enable Notifications for pollution alerts**
* **Check Trends & Maps for analysis**

**Key Features**

* **Real-time AQI map (color-coded)**
* **1–72 hr forecast (R² = 0.9994)**
* **Historical trends & analytics**
* **Personalized health alerts**
* **Threshold-based push notifications**

**Tech Stack**

* **Python | Random Forest| Streamlit | Folium Maps**

**17.** **Bibliography, References**

|  |  |  |
| --- | --- | --- |
| **Category** | **Source / Paper / Dataset** | **Reference Link** |
| **Ground Data (CPCB)** | **Central Pollution Control Board – National Air Quality Monitoring Program** | [**https://cpcb.nic.in/**](https://cpcb.nic.in/) |
| **Meteorological Data (MERRA-2)** | **NASA Global Modeling and Assimilation Office – MERRA-2 Dataset** | [**https://disc.gsfc.nasa.gov/datasets/M2T1NXAER\_5.12.4**](https://disc.gsfc.nasa.gov/datasets/M2T1NXAER_5.12.4) |
| **Satellite Data (INSAT-3DR)** | **ISRO Meteorological & Oceanographic Satellite Data Archive Centre (MOSDAC)** | [**https://mosdac.gov.in/**](https://mosdac.gov.in/) |
| **Machine Learning Models** | **Random Forests (Breiman, 2001) — Ensemble Learning Framework** | [**https://doi.org/10.1023/A:1010933404324**](https://doi.org/10.1023/A:1010933404324) |
| **AQI Standards** | **CPCB – National Air Quality Index Methodology** | [**https://app.cpcbccr.com/AQI\_India/**](https://app.cpcbccr.com/AQI_India/) |
| **Predictive Modeling Reference** | **“Air Pollution Forecasting using LSTM and XGBoost Models” – ScienceDirect (2023)** | [**https://doi.org/10.1016/j.envsoft.2023.105652**](https://doi.org/10.1016/j.envsoft.2023.105652) |
| **Visualization Framework** | **Streamlit Documentation – Interactive Data Apps** | [**https://docs.streamlit.io/**](https://docs.streamlit.io/) |
| **Global AQI Data (Backup)** | **OpenAQ API – Open Air Quality Platform** | [**https://openaq.org**](https://openaq.org/) |

**Citation Style:** IEEE / APA-compliant formatting (for research paper integration)  
**Version:** Bibliography v1.0

18. Published copy of Research Paper & Certificate

**A paper with text and words

AI-generated content may be incorrect.**

Research Paper link : [**https://drive.google.com/file/d/1uaLhwyT1dVbLdGUDsadEisyZ2833nuiq/view?usp=sharing**](https://drive.google.com/file/d/1uaLhwyT1dVbLdGUDsadEisyZ2833nuiq/view?usp=sharing)

**Signature of the Student Signature of the Guide**

**----------------- -------------------**