24-25J-224

Intelligent Eco-Urban Monitoring System (IEMS)

Sri Lanka Institute of InformationTechnology

SLIIT UNI
THE KNOWLEDGE UNIVERSITY

OUR TEAM

24-25J-224

Dr. Samantha Rajapaksha

Head | Department of IT

Supervisor

Ms. Kaushika Kavindi

Assistant Lecturer

Co-Supervisor

Thuduwage I.M.H.G

IT21169380

Software Engineering

Arandara S. D.

IT21164330

Software Engineering

Karunarathne R. Y. D.

IT21169144

Software Engineering

Kodithuwakku C.K.

IT21156960

Software Engineering

SLIIT UNI

THE KNOWLEDGE UNIVERSITY

IEMS

INTRODUCTION

IEMS

Urban Environmental Challenges

- Rapid urbanization leads to issues like poor air quality, green space loss, noise pollution, and high vehicle emissions.

Why Address These Issues?

- Sustainable urban living is vital for residents' well-being, requiring effective monitoring and solutions.

Introduction to IEMS

- IEMS addresses these challenges with integrated monitoring and prediction, featuring EcoSensor AI, GreenVision AI, NoiseGuard AI, and Eco Go.

RESEARCH PROBLEM

IEMS

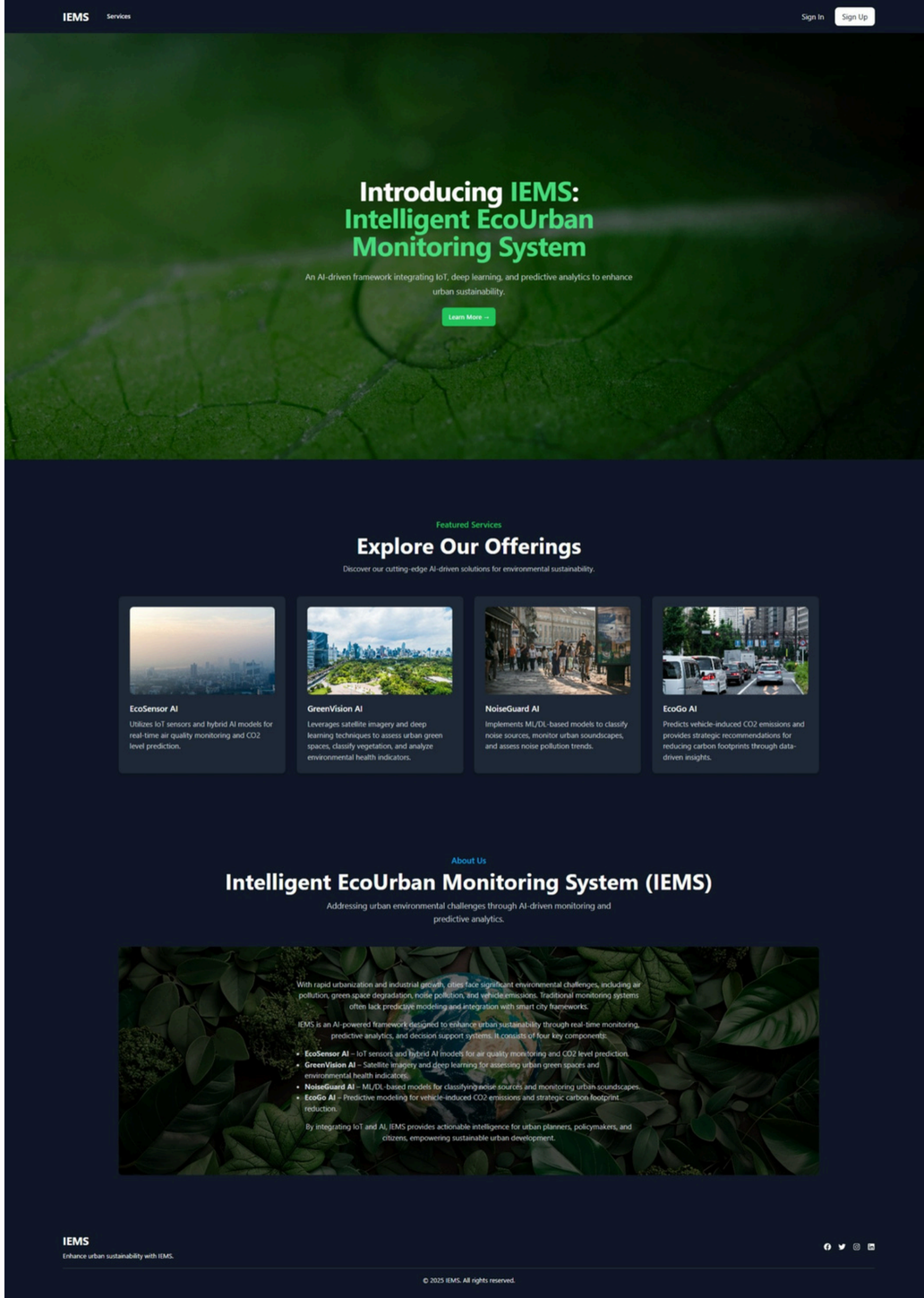
Environmental Challenges in Urban Areas

- **Air Quality Concerns**
  - Urban pollution from transport and industry includes harmful pollutants like PM2.5, NO2, and O3, posing health risks.
- **Green Space Degradation**
  - Urbanization reduces and fragments green spaces, impacting carbon capture and biodiversity.
- **Noise Pollution**
  - Noise from transport and construction causes stress and health issues like sleep disturbances.
- **Vehicle Emissions**
  - Increasing traffic contributes to air pollution and greenhouse gases, requiring emission reduction strategies.

IMPLEMENTED SOLUTION

Integrated Environmental Monitoring System (IEMS)

- **EcoSensor AI**
  - Real-time air quality monitoring and predictions.
- **NoiseGuard AI**
  - Smart noise monitoring and source identification.
- **Eco Go**
  - Predicts and reduces vehicle CO2 emissions.
- **GreenVision AI**
  - Manages green spaces using satellite imagery.



Main Objective

Enhancing Urban Sustainability

Goal

To enhance urban sustainability through effective monitoring, prediction, and mitigation of environmental challenges.

Objectives

- Optimize green spaces.
- Improve air quality.
- Reduce noise pollution.
- Lower vehicle emissions by leveraging advanced AI and IoT technologies.

IEMS

Sub Objectives

Specific Goals and Tasks

Arandara.S.D:

- Deploy IoT sensors for CO2 level monitoring.
- Train models for air quality prediction and recommendations.
- Implement adaptive sensor calibration using AI for dynamic accuracy adjustments.

Thuduwage I.M.H.G:

- Analyze satellite imagery for green space management.
- Train deep learning models for vegetation assessment.
- Provide visualization tools for data analysis and decision making.

IEMS

Sub Objectives

Specific Goals and Tasks

Karunarathne R.Y.D:

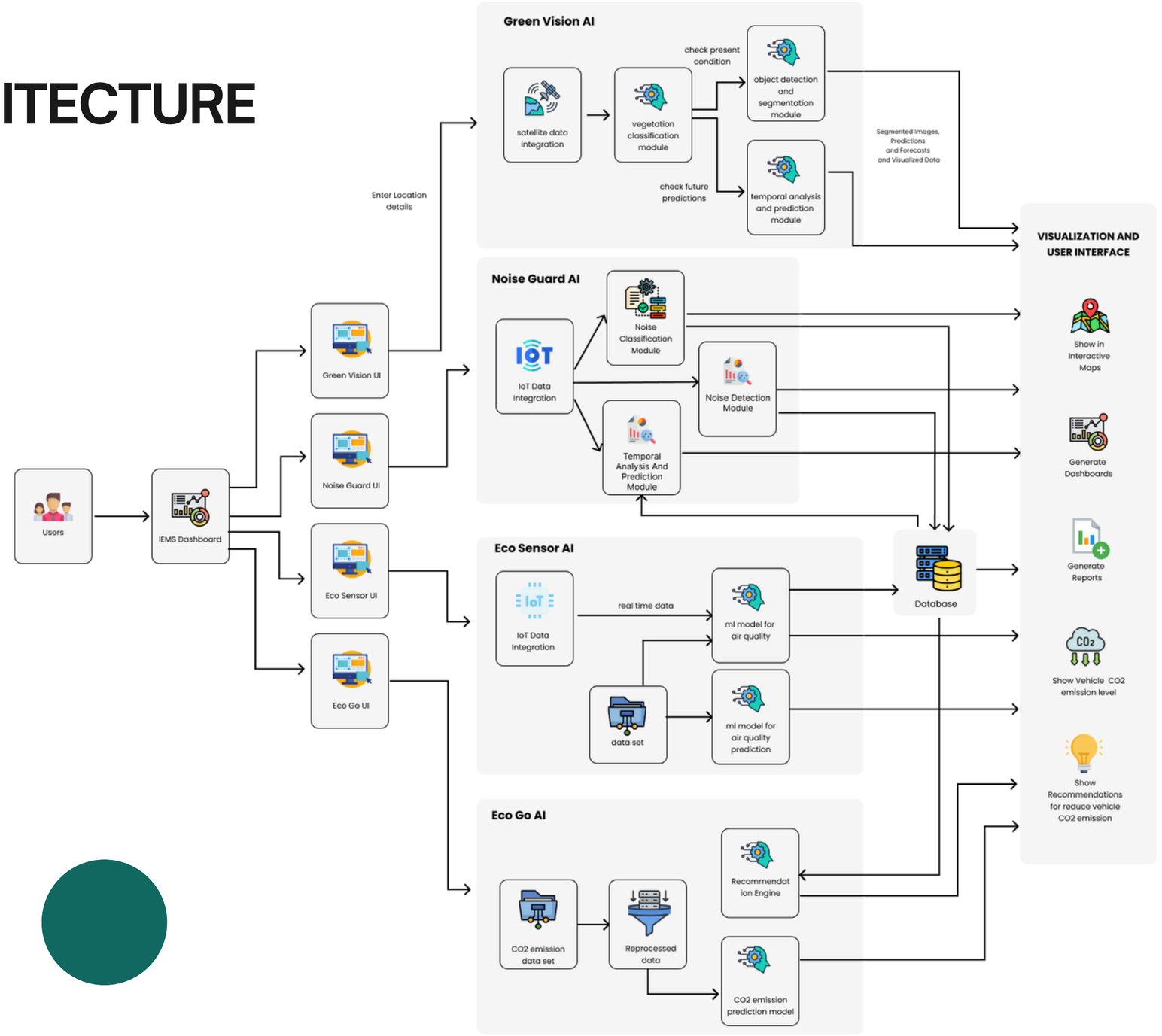
- Implement smart noise monitoring with IoT sensors.
- Classify noise sources and predict future noise levels.
- Develop alert mechanisms for noise thresholds.

Kodithuwakku C.K

- Develop a personalized CO2 emissions prediction and reduction engine.
- Train machine learning models and implement user feedback systems.
- Continuously retrain models to adapt to new data and feedback.

SYSTEM ARCHITECTURE

IEMS



Team Member 1

Green Vision AI

I.M.H.G. Thuduwage
IT21169380

GreenVision AI:
Managing Urban Green Spaces

Background

Urban green spaces are essential for reducing heat, improving air quality, and supporting biodiversity, but urban density and pollution make maintenance challenging.

- **Deep Learning Analysis**
  - AI analyzes satellite images to manage green spaces.
- **Vegetation Indices and Object Detection**
  - Assesses health and distribution of greenery.
- **Informed Decision-Making**
  - Identifies areas needing more green space.
- **Environmental Benefits**
  - Enhances biodiversity, reduces heat, and improves environmental quality.

Research Problem

Urban areas struggle to assess and manage green spaces due to fragmented data and lack of effective analysis tools. An advanced system is needed to analyze and enhance green spaces.

IT21169380 | Thuduvage I.M.H.G. | 24-25J-224

Research Gap

IEMS

Features	[1]	[2]	[3]	[4]	IEMS
Multi-Layered Vegetation Indices	Partial	✓	✗	Partial	✓
Vegetation Classification	✓	Partial	✗	Partial	✓
Object Detection and Segmentation	✗	Partial	✗	✗	✓
Temporal Analysis and Prediction	✗	✗	✗	Partial	✓
Real-Time Data Integration	✗	✗	✗	✗	✓
Visualization and User Interface	Partial	✓	Partial	Partial	✓

Obejctives

IEMS

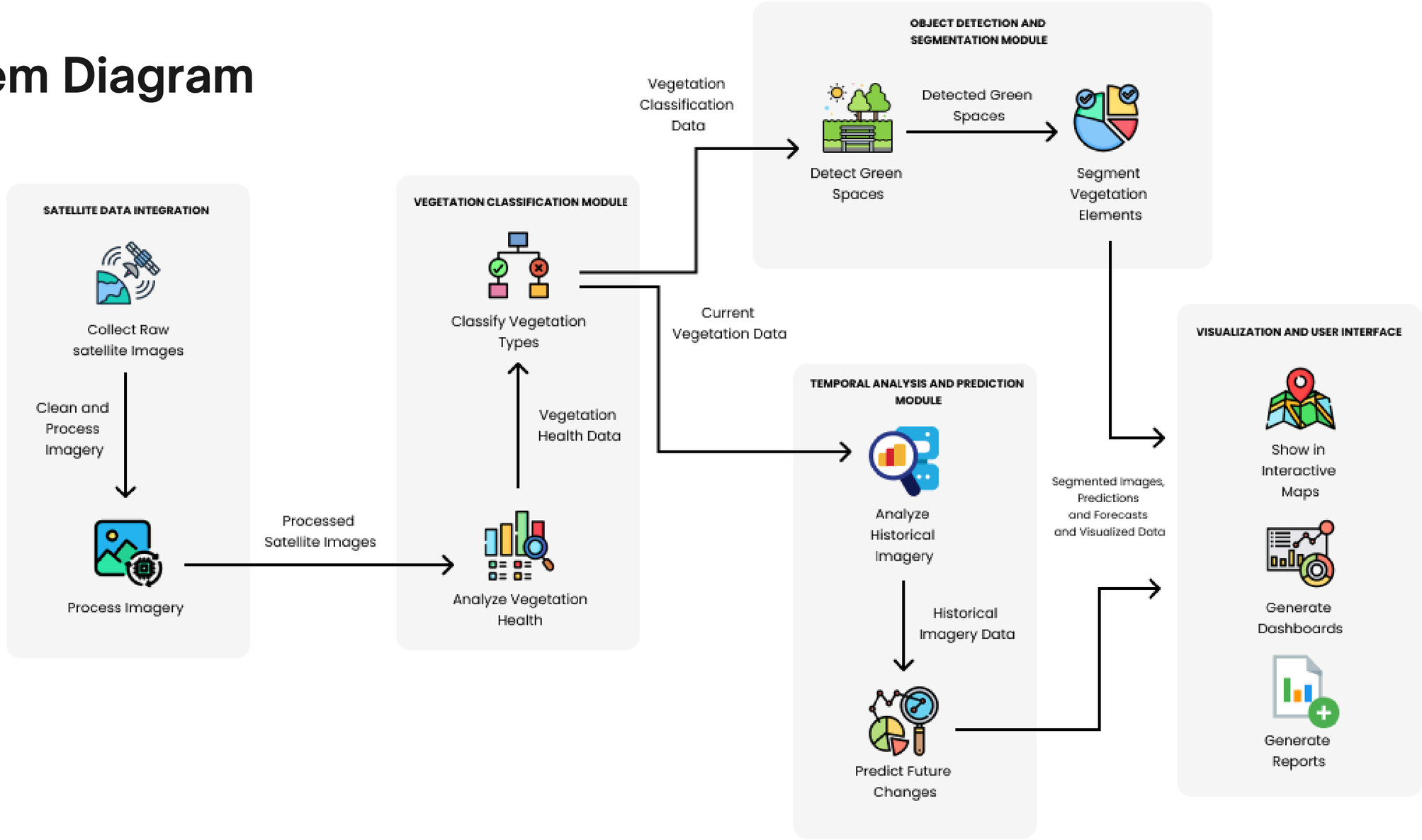
Main Objectives

- Improve urban sustainability by monitoring, predicting, and managing green spaces.
- Provide cities with advanced tools for better urban development and environmental management

Sub-Objectives

- Use satellite images and AI to manage green spaces.
- Identify areas needing more greenery based on green coverage and Temporal Analysis and Prediction.
- Guide decisions on creating new green spaces to improve biodiversity and reduce heat.

System Diagram



Technologies Used

Deep Learning

- CNN, U-Net – Tree segmentation and vegetation classification.
- LSTM – Predicting future vegetation coverage and CO2 levels.

Remote Sensing

- NDVI Calculation – Vegetation density classification.
- Satellite Imagery Processing – Green area analysis.

Backend

- Flask API, TensorFlow/Keras – Image processing and AI model inference.
- OpenCV – Image preprocessing and visualization.

Frontend & Integration

- React.js, Google Maps API – UI for selecting locations and viewing results.
- Axios – Frontend-backend communication.

Current Progress

- Developed the core backend models and Flask API for GreenVision AI.
- Created a tree segmentation model using a CNN architecture to classify vegetation areas from satellite images.
- Built a forecasting model to predict future vegetation coverage based on historical data
- Implemented image preprocessing techniques such as normalization, resizing, and mask processing for accurate segmentation.
- Developed vegetation density analysis using NDVI calculations to categorize high, medium, and low vegetation coverage (app.py).
- Created a Flask API to handle image uploads, process satellite images, and return predictions (app.py).

- Developed the frontend interface using React for an interactive user experience.
- Integrated the React app with the Flask backend, allowing users to interact with the AI model.
- Implemented a map selection feature where users can choose a location from a satellite map.
- Added functionality to capture and crop a snippet of the selected location.
- Enabled image and location data upload to the backend for analysis.
- Developed an interactive dashboard to visualize green area distribution and provide recommendations based on AI predictions.

Project Evidence

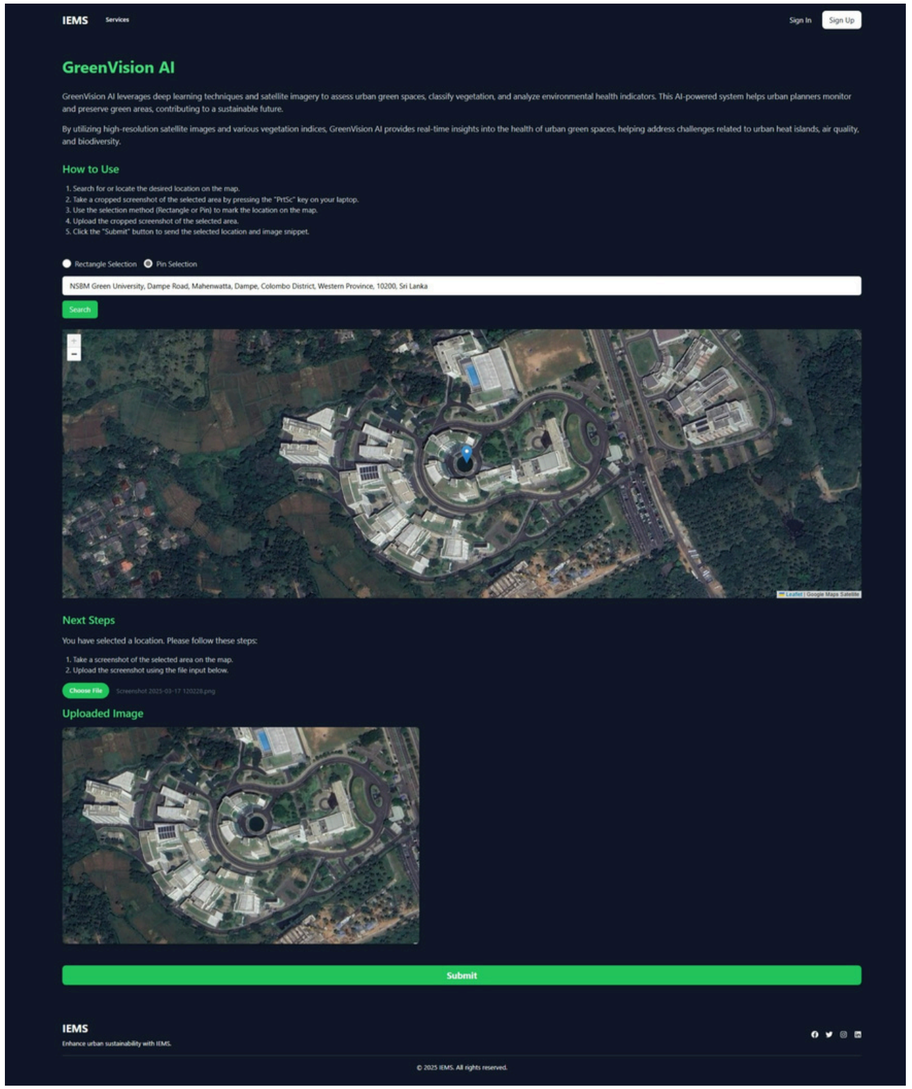
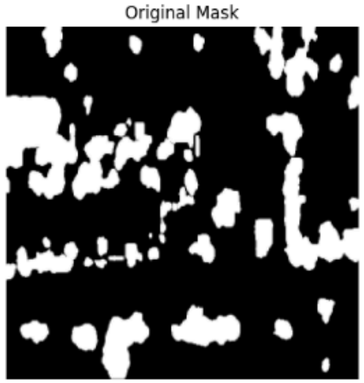
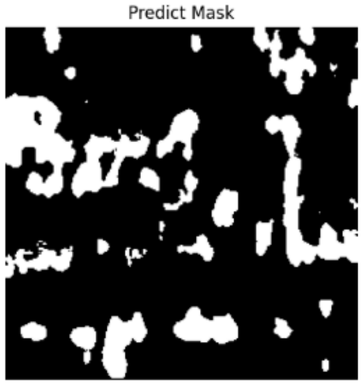
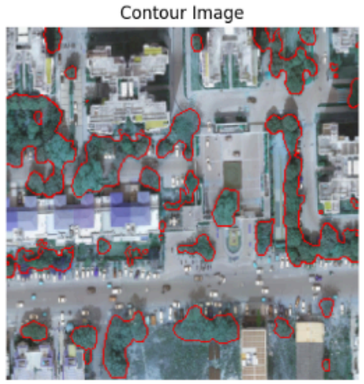
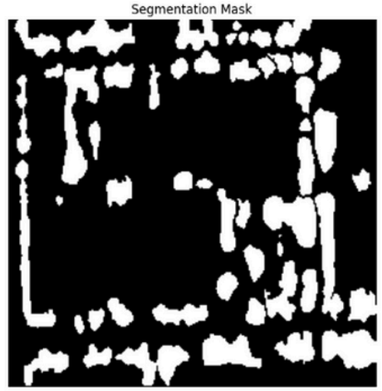


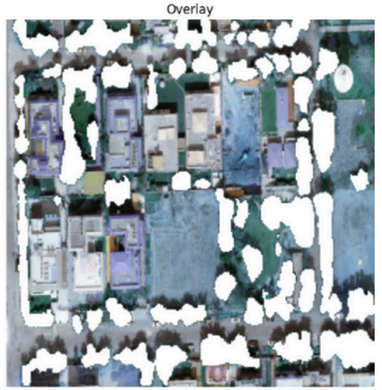
Image Masking







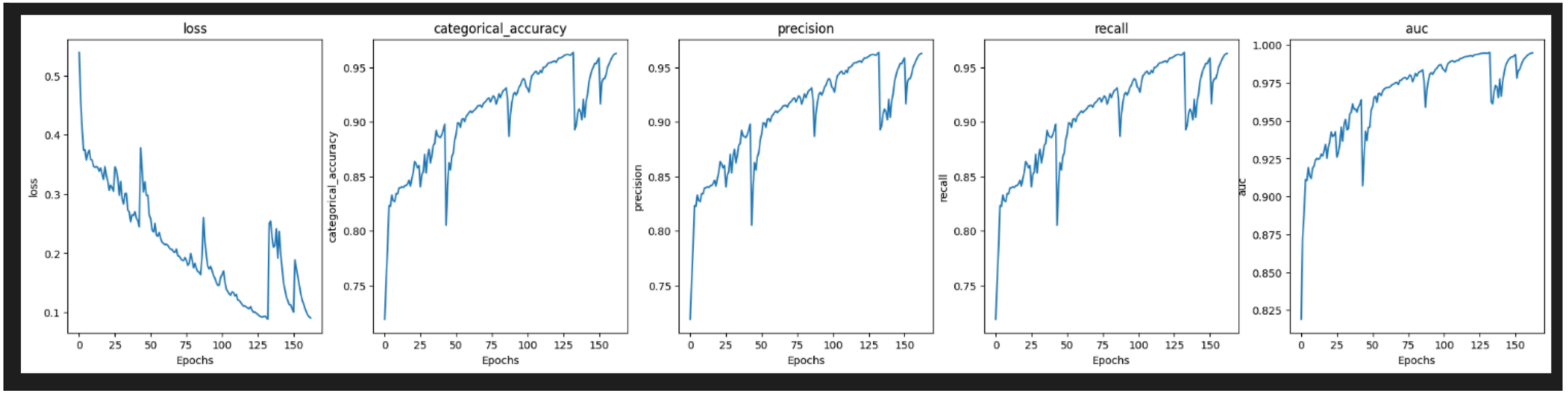




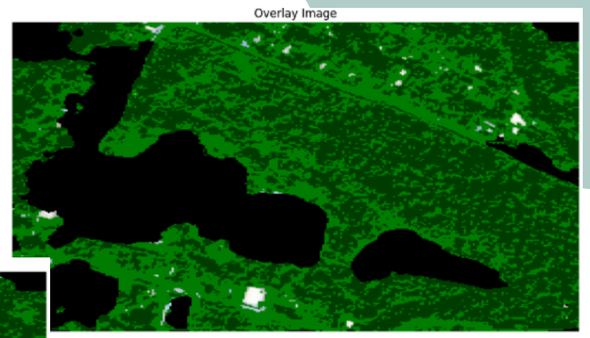
Accuracy

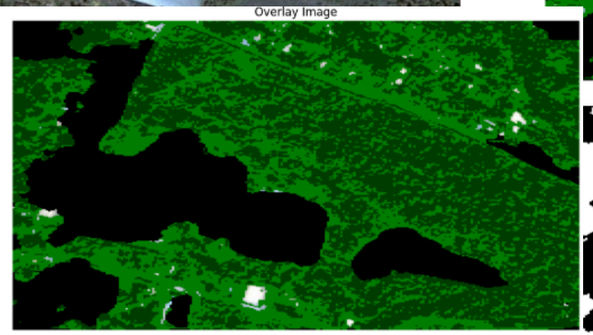
```
...
Epoch 162/200
17/17 [==============================] - 4s 214ms/step - loss: 0.0922 - categorical_accuracy: 0.9621 - precision: 0.9621 - recall: 0.9621 - auc: 0.9946
Epoch 163/200
17/17 [==============================] - 4s 215ms/step - loss: 0.0906 - categorical_accuracy: 0.9626 - precision: 0.9626 - recall: 0.9626 - auc: 0.9948
```

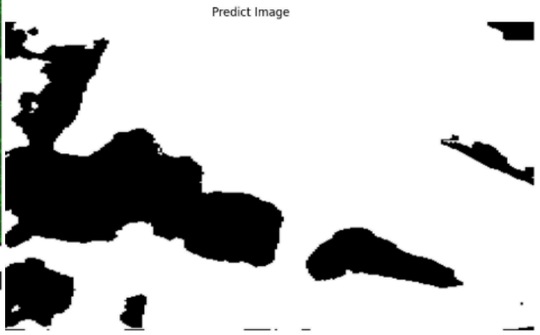
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...



Predictions








```
1/1 [==============================
1/1 [==============================
Location ID :  0
1/1 [==============================
{'forecast': {'month 1': '20.530
              'month 2': '20.530
              'month 3': '20.540
 'segmentation_results': {'conto
                          'origi
                          'overl
                          'predi
 'segmentation_stats': {'Green P
                        'High Vegetation Density Coverage': '3.35 %',
                        'Low Vegetation Density Coverage': '55.73 %',
                        'Medium Vegetation Density Coverage': '40.92 %',
                        'NDVI Score': 0.02}}
```

TOOLS & TECHNOLOGIES

IEMS

Requirements

IEMS

System Requirements

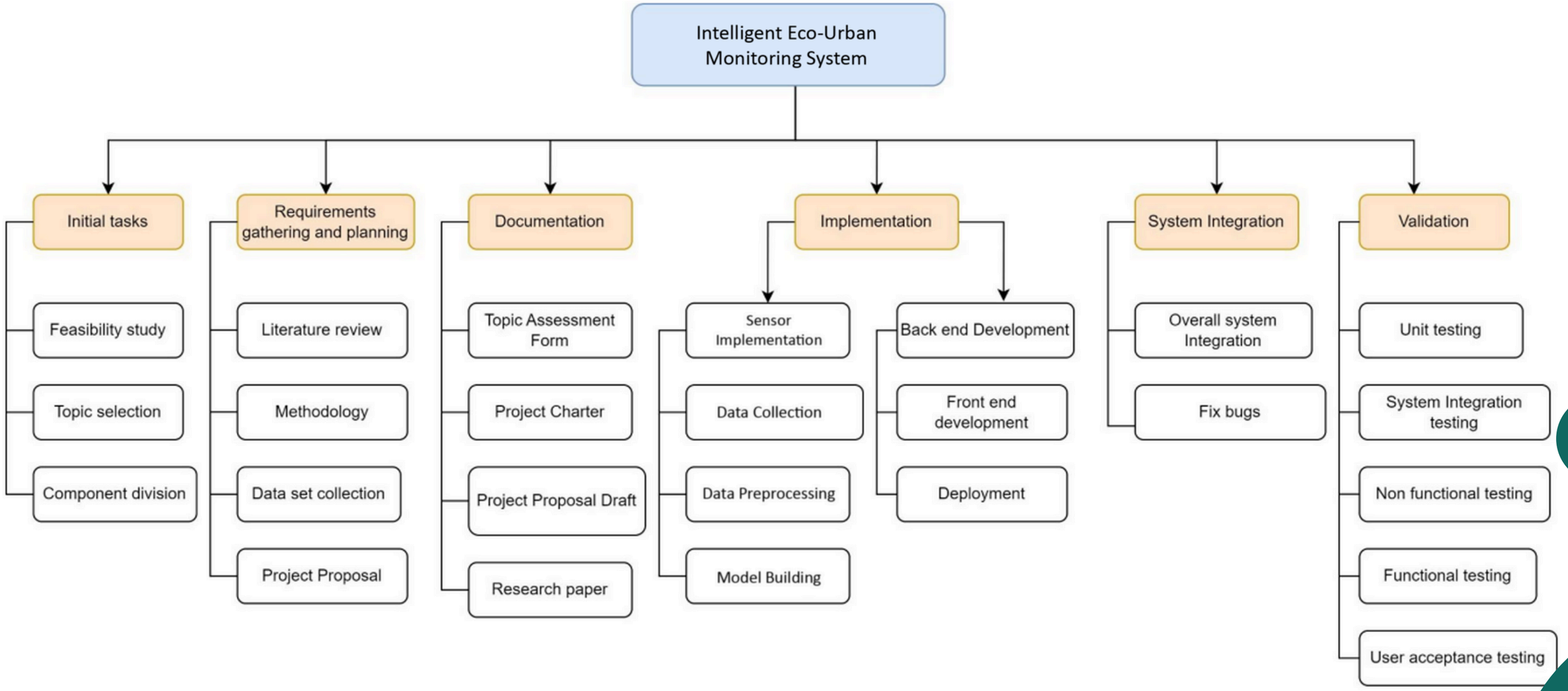
- High-resolution satellite imagery, High-Performance Computing

Software Requirements

- **Functional Requirements**
  - Data Acquisition and Preprocessing
  - Vegetation Analysis and Classification
  - Temporal Analysis and Forecasting
  - Visualization and Reporting
- **Non-Functional Requirements**
  - Performance
  - Scalability
  - Reliability
  - Usability

WORK BREAKDOWN CHART

IEMS

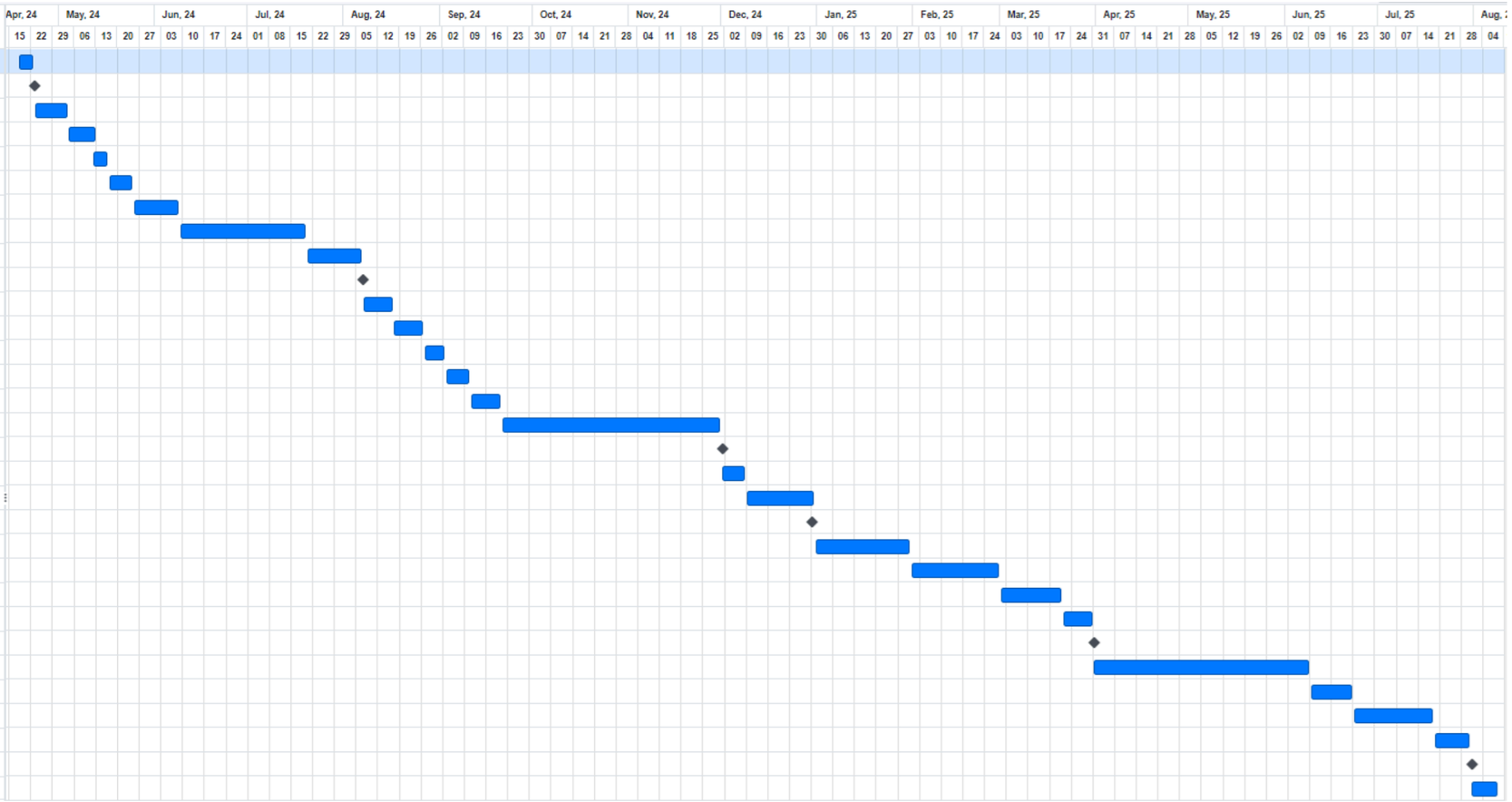


IT21169380 | Thuduvage I.M.H.G. | 24-25J-224

PROJECT TIMELINE - GANTT CHART

IEMS

ID	Name	Start Date	End Date	Duration
1	Topic selection	Apr 18, 2024	Apr 22, 2024	5 days
2	Topic selection Completed	Apr 23, 2024	Apr 23, 2024	0 days
3	Feasibility study	Apr 23, 2024	May 03, 2024	11 days
4	Requirement gathering	May 04, 2024	May 12, 2024	9 days
5	Background survey	May 12, 2024	May 16, 2024	5 days
6	Literature review	May 17, 2024	May 24, 2024	8 days
7	NON/functional requirements	May 25, 2024	Jun 08, 2024	15 days
8	Project Proposal draft	Jun 09, 2024	Jul 19, 2024	41 days
9	proposal presentation	Jul 20, 2024	Aug 06, 2024	18 days
10	proposal presentation Completed	Aug 07, 2024	Aug 07, 2024	0 days
11	Proposal report	Aug 07, 2024	Aug 16, 2024	10 days
12	Finalizing dataset	Aug 17, 2024	Aug 26, 2024	10 days
13	Finalizing methodologies	Aug 27, 2024	Sep 02, 2024	7 days
14	Data preprocessing	Sep 03, 2024	Sep 10, 2024	8 days
15	Text extraction	Sep 11, 2024	Sep 20, 2024	10 days
16	Model building	Sep 21, 2024	Nov 30, 2024	71 days
17	Model building Completed	Dec 01, 2024	Dec 01, 2024	0 days
18	Progress Presentation 1	Dec 01, 2024	Dec 08, 2024	8 days
19	Model building 2	Dec 09, 2024	Dec 30, 2024	22 days
20	Model building 2 Completed	Dec 30, 2024	Dec 30, 2024	0 days
21	Back End completion	Dec 31, 2024	Jan 30, 2025	31 days
22	Front End completion	Jan 31, 2025	Feb 28, 2025	29 days
23	First integration	Mar 01, 2025	Mar 20, 2025	20 days
24	Progress Presentation 2	Mar 21, 2025	Mar 30, 2025	10 days
25	Progress Presentation 2 Delivered	Mar 31, 2025	Mar 31, 2025	0 days
26	Integration completion	Mar 31, 2025	Jun 08, 2025	70 days
27	Unit testing	Jun 09, 2025	Jun 22, 2025	14 days
28	Integration testing	Jun 23, 2025	Jul 18, 2025	26 days
29	Final document	Jul 19, 2025	Jul 30, 2025	12 days
30	Final document Completed	Jul 31, 2025	Jul 31, 2025	0 days
31	Final Presentation and VIVA	Jul 31, 2025	Aug 08, 2025	9 days



References

[1] Urban green spaces analysis for development planning in Colombo, Sri Lanka utilizing THEOS satellite imagery - A remote sensing and GIS approach, ResearchGate, 2013. [Online]. Available: link.

[2] Application of satellite images and GIS in evaluation of green space destruction in urban area: Case study Boukan City, ResearchGate, 2012. [Online]. Available: link.

[3] Green spaces and cognitive development in primary school children, National Center for Biotechnology Information (NCBI), 2022. [Online]. Available: link.

[4] Urban green spaces, heat island effects and sustainable development: A review, IOPscience, 2021. [Online]. Available: link.

Urban green spaces and their potential to improve air quality in cities, Francis Press, 2022. [Online]. Available: link.

Team Member 2

EcoSensor AI

Arandara S. D.
IT21164330

EcoSensor AI
Air Quality Monitoring ,
Prediction and Recommondation Management

Background

Urbanization challenges air quality, vital for health and sustainability. EcoSensor AI uses IoT sensors for real-time pollutant monitoring and AI for analysis and predictions, offering recommendations to maintain optimal CO2 levels.

Air Quality Monitoring and IoT Sensor Deployment

- Strategically place IoT sensors in urban areas to collect data on air pollutants like CO2

Data Analysis and Prediction

- Use advanced machine learning models, such as Random Forest, Gradient Boosting, and LSTM networks, to analyze data and predict future air quality trends..

Recommendations for Air Quality Management

- EcoSensor AI includes predictive models that provide recommendations to maintain optimal CO2 levels and overall air quality.

Research Problem

Urbanization challenges air quality and green space management, impacting public health and the environment. Traditional systems lack real-time data, future air quality predictions, and actionable recommendations.

IT21164330 | Arandara S.D. | 24-25J-224

Research Gap

Features	[1]	[2]	[3]	[4]	IEMS
Air Quality Monitering	✓	✓	✗	✓	✓
IoT Integrated	✓	✓	✗	✓	✓
Recommandations	✗	✗	✗	✗	✓
Air Quality Prediction	✗	✗	✓	✗	✓
All in one Implement Web Application	✗	✗	✗	✗	✓
Report Generation	✗	✗	✗	✗	✓

[1] An IoT Based Air Pollution Monitoring System for Smart Cities

[2] Real-Time Air Quality Monitoring System using IoT

[3] Machine learning-based artificial intelligence method for predicting the air pollution index PM2.5

[4] An Integrated IoT and Machine Learning Approach for Environmental Monitoring and Management

Obejctives

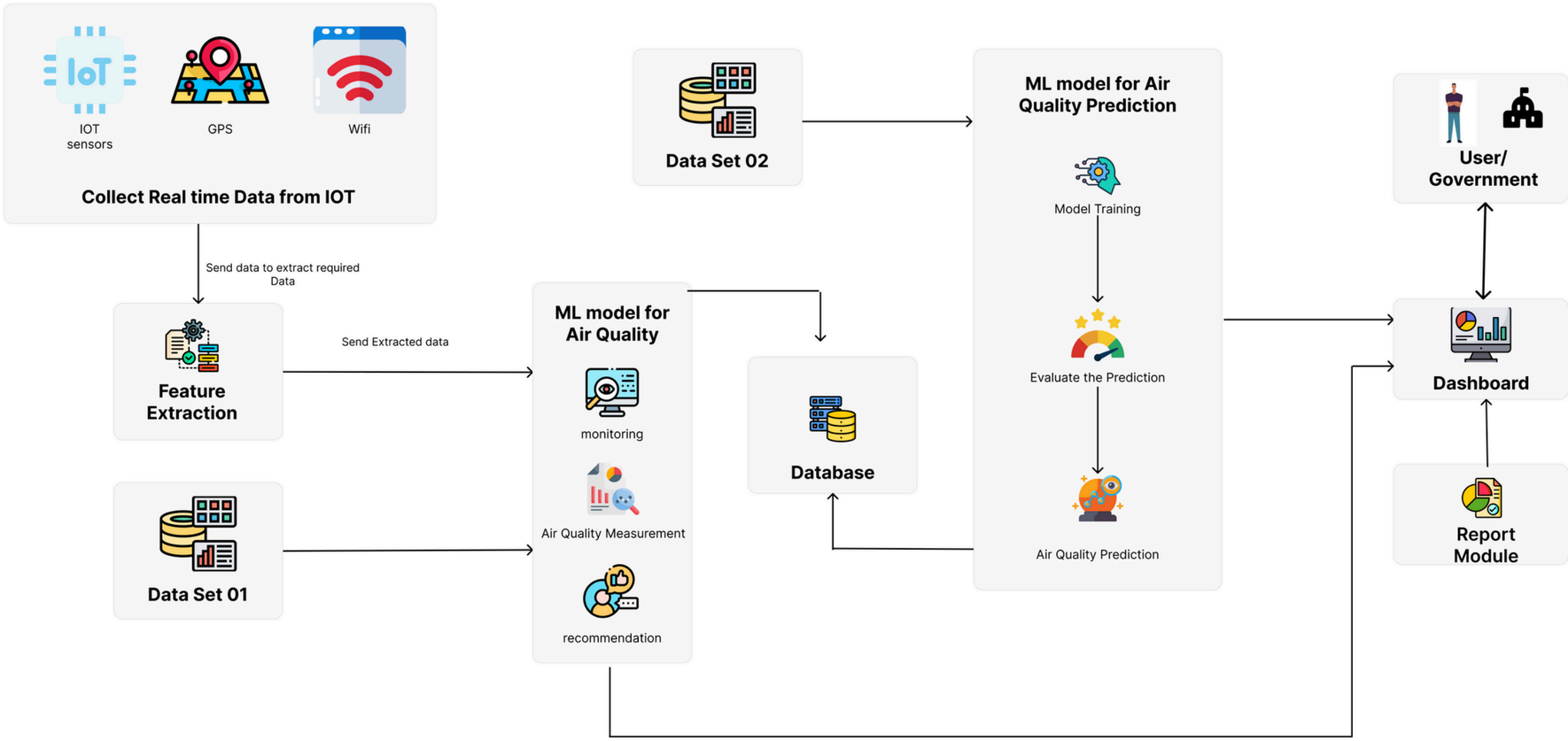
Main Objectives

- Enhance urban sustainability through effective monitoring, prediction, and management of air quality using IoT sensors and AI.
- Generate actionable recommendations for air quality management based on predictive models.

Sub-Objectives

- Deploy IoT Sensors
- Develop Predictive Models
- Provide Recommendations
- Develop User Interface

System Diagram



IT21164330 | Arandara S.D. | 24-25J-224

Technologies Used

Machine Learning:

- Supervised Models such as Random Forest, Gradient Boosting

IoT Sensors:

- ESP-32 , MQ-138 , e.t.c

Data Visualization:

- D3.js, Chart.js

Programing languages :

- python , MERN stack , Flask

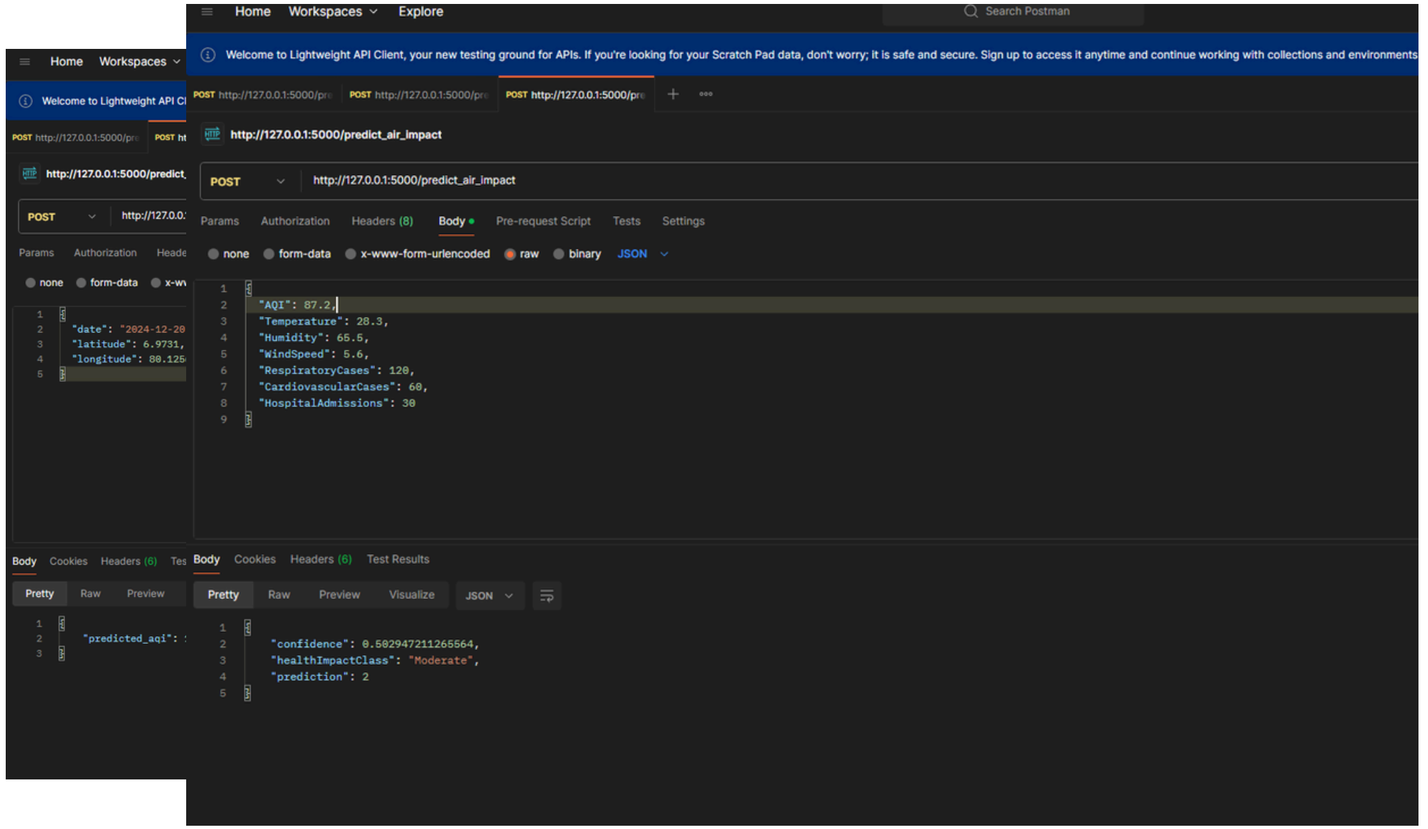
Current Progress

- Developed a Machine Learning Model for Air Quality Impact Classification
- Fine-Tuned Model Parameters using Optimization Techniques
- Implemented Backend API using Flask

- Upgraded to Deep Learning Approach for Impact Classification
- Designed and Built IoT Device for Real-time Air Quality Monitoring
- Collected & Preprocessed Data for Air Quality Prediction Model
- Developed Air Quality Prediction Model for Future Forecasting

IT21164330 | Arandara S.D. | 24-25J-224

Project Evidence



Data Distribution

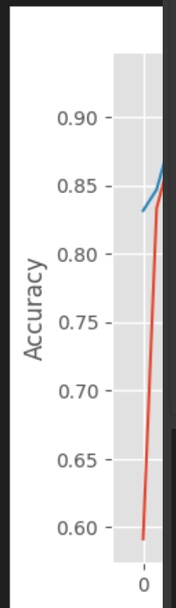
```
df.head()
```

	RecordID	AQI	PM10	PM2_5	NO2	SO2	O3	Temperature	Humidity	WindSpeed	RespiratoryCases	CardiovascularCases	HospitalAdmissions
0	1	187.270059	295.853039	13.038560	6.639263	66.161150	54.624280	5.150335	84.424344	6.137755	7	5	1
1	2	475.357153	246.254703	9.984497	16.318326	90.499523	169.621728	1.543378	46.851415	4.521422	10	2	0
2	3	365.996971	84.443191	23.111340	96.317811	17.875850	9.006794	1.169483	17.806977	11.157384	13	3	0
3	4	299.329242	21.020609	14.273403	81.234403	48.323616	93.161033	21.925276	99.473373	15.302500	8	8	1
4	5	78.009320	16.987667	152.111623	121.235461	90.866167	241.795138	9.217517	24.906837	14.534733	9	0	1

Accuracy

```
plot_acc(h
```

[40]




```
history = model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))
```

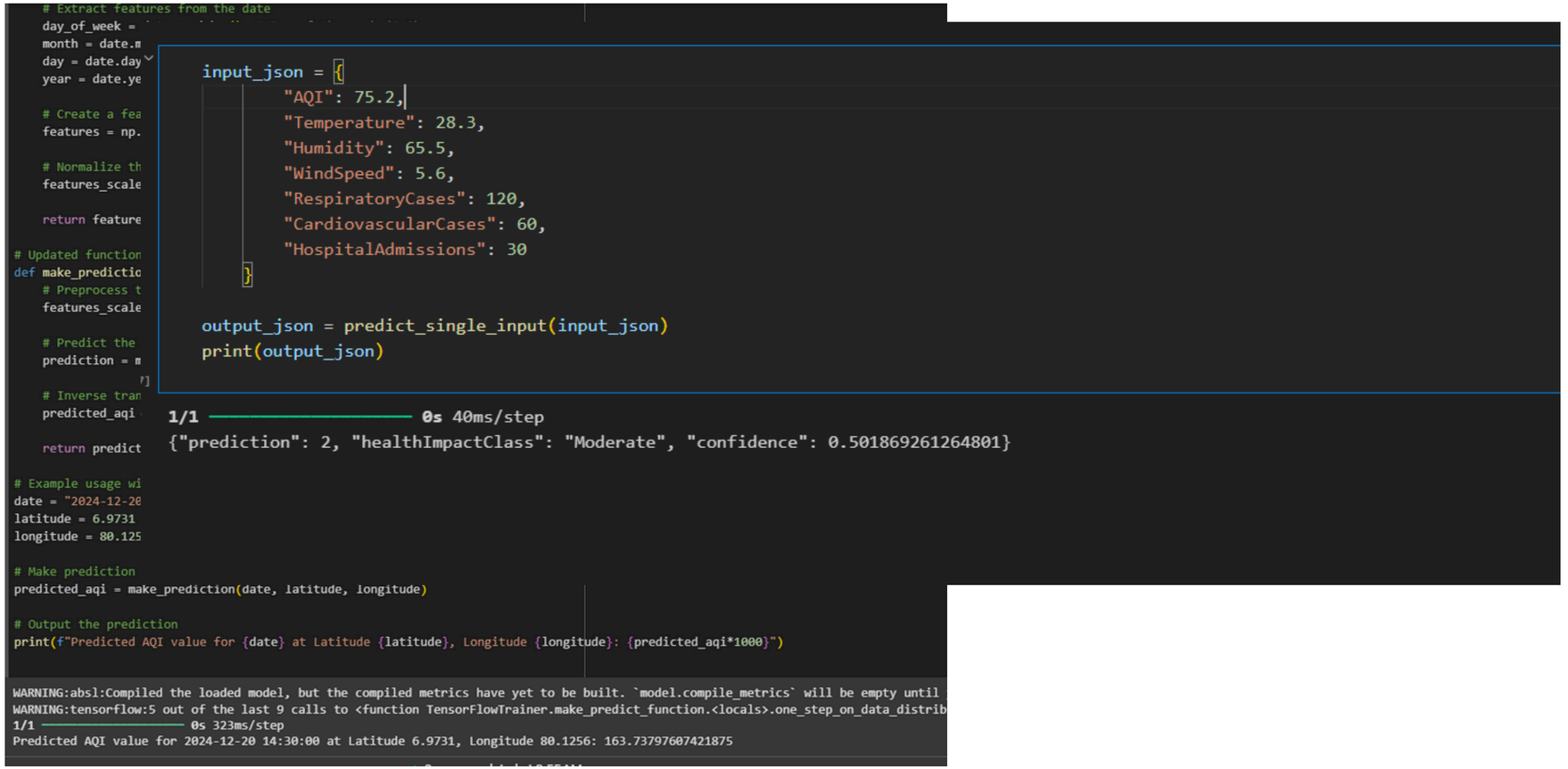
```
Epoch 1/10
21/21 ━━━━━━━━━━━━━━━━━━━━ 6s 80ms/step - loss: 0.1670 - val_loss: 0.0949
Epoch 2/10
21/21 ━━━━━━━━━━━━━━━━━━━━ 2s 57ms/step - loss: 0.0900 - val_loss: 0.0851
Epoch 3/10
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 57ms/step - loss: 0.0876 - val_loss: 0.0847
Epoch 4/10
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 55ms/step - loss: 0.0878 - val_loss: 0.0850
Epoch 5/10
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 58ms/step - loss: 0.0898 - val_loss: 0.0849
Epoch 6/10
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 57ms/step - loss: 0.0856 - val_loss: 0.0841
Epoch 7/10
21/21 ━━━━━━━━━━━━━━━━━━━━ 2s 89ms/step - loss: 0.0914 - val_loss: 0.0836
Epoch 8/10
21/21 ━━━━━━━━━━━━━━━━━━━━ 2s 72ms/step - loss: 0.0847 - val_loss: 0.0839
Epoch 9/10
21/21 ━━━━━━━━━━━━━━━━━━━━ 2s 56ms/step - loss: 0.0867 - val_loss: 0.0856
Epoch 10/10
21/21 ━━━━━━━━━━━━━━━━━━━━ 1s 57ms/step - loss: 0.0832 - val_loss: 0.0837
```

```
[ ] predictions = model.predict(X_test)
    predictions = scaler.inverse_transform(predictions)
    y_test_actual = scaler.inverse_transform(y_test.reshape(-1, 1))
    rmse = np.sqrt(mean_squared_error(y_test_actual, predictions))
    print("RMSE: ", rmse)
    r2 = r2_score(y_test_actual, predictions)
    print("R^2: ", r2)
```

```
6/6 ━━━━━━━━━━━━━━━━━━━━ 1s 79ms/step
RMSE:  0.2892256572710164
R^2:  -0.038347211078942234
```

IT21164330 | Arandara S.D. | 24-25J-224

Predictions



IT21164330 | Arandara S.D. | 24-25J-224

TOOLS & TECHNOLOGIES

IEMS

IT21164330 | Arandara S.D. | 24-25J-224

Requirements

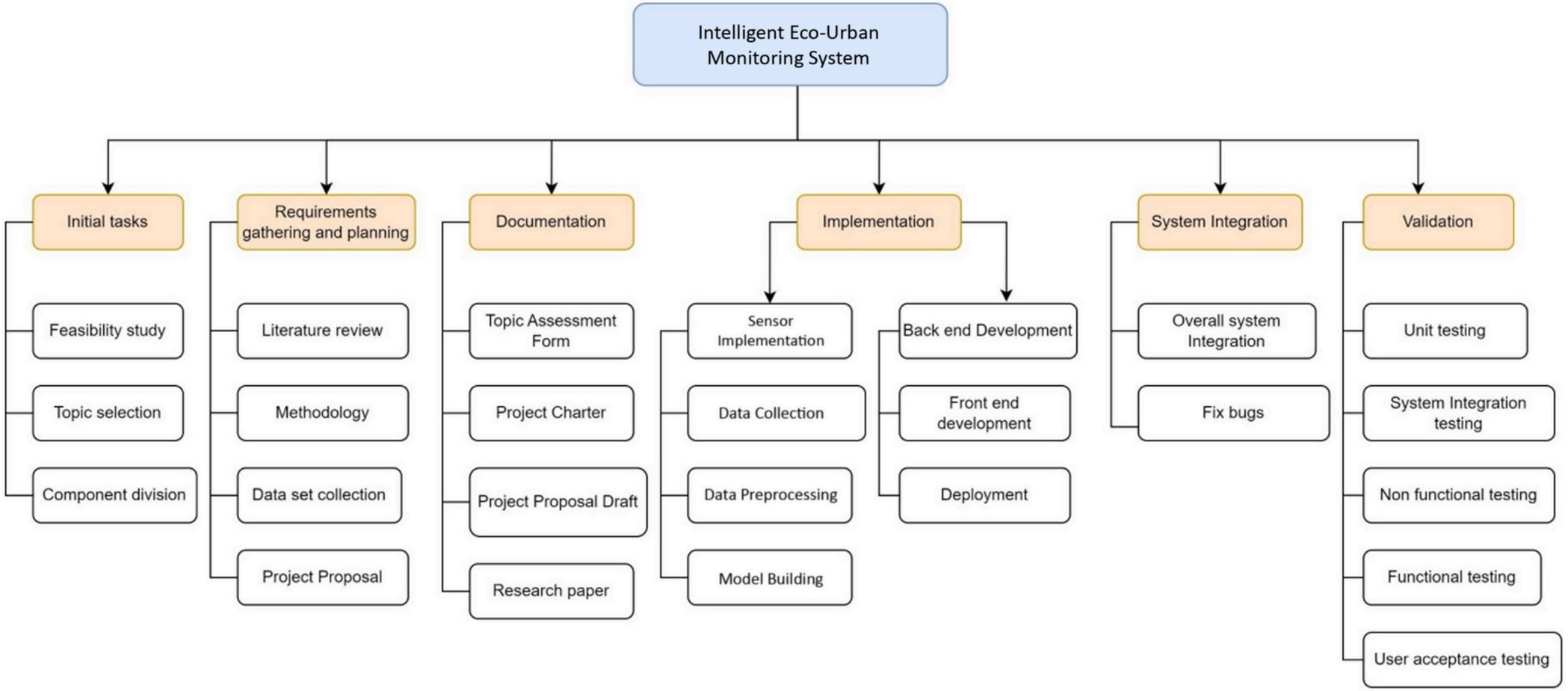
System Requirements

- Hardware: IoT sensors for CO2 and PM2.5 data collection, high-performance computing for model training.
- Software: Machine learning frameworks (TensorFlow, PyTorch), data visualization tools.

Software Requirements

- **Functional Requirements**
  - Air quality monitoring
  - Data analysis and predictions
  - Recommondations for good air quality
- **Non-Functional Requirements**
  - Performance
  - Scalability
  - Reliability
  - Usability

WORK BREAKDOWN CHART

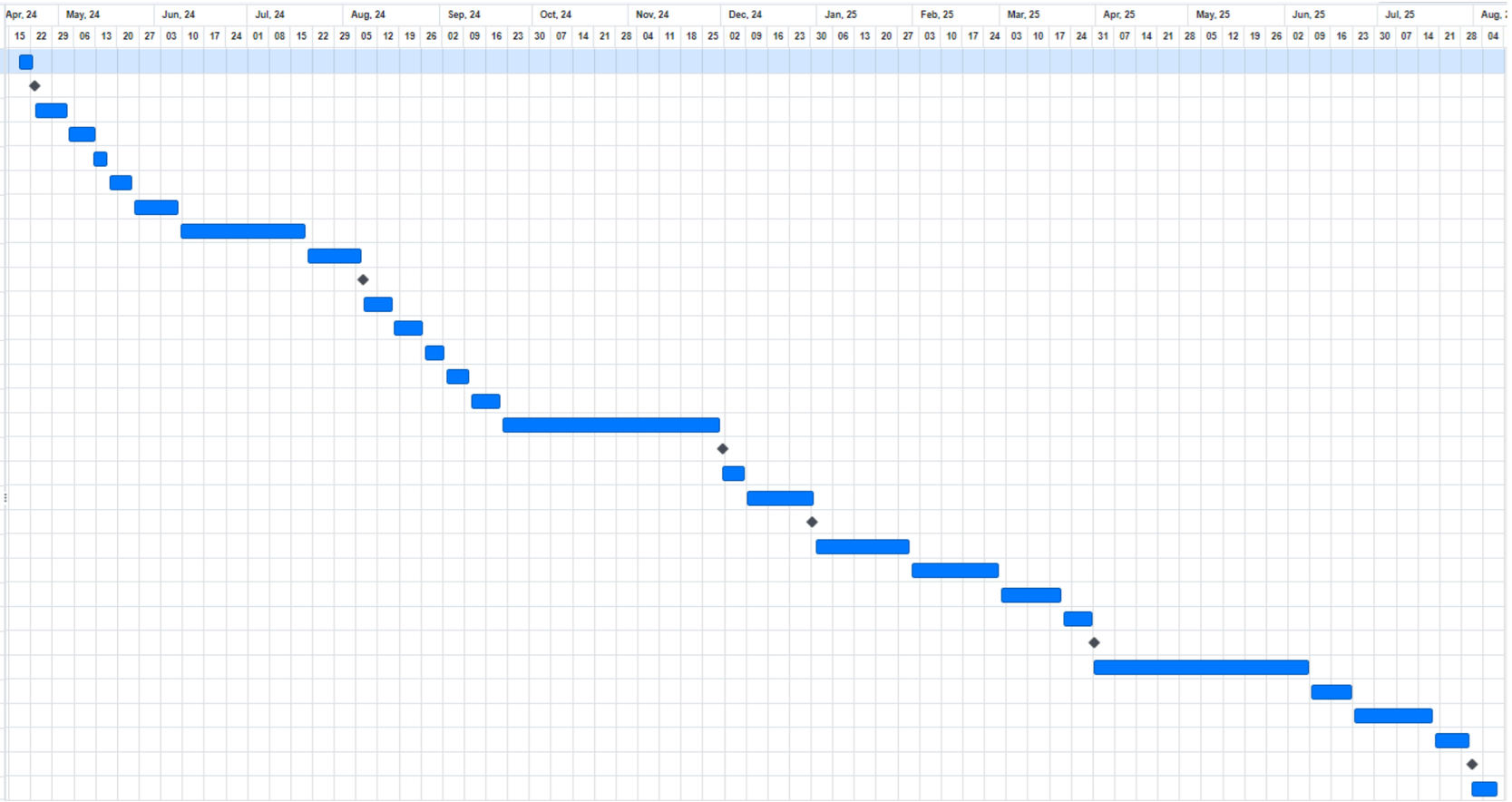


IT21164330 | Arandara S.D. | 24-25J-224

PROJECT TIMELINE - GANTT CHART

IEMS

ID	Name	Start Date	End Date	Duration
1	Topic selection	Apr 18, 2024	Apr 22, 2024	5 days
2	Topic selection Completed	Apr 23, 2024	Apr 23, 2024	0 days
3	Feasibility study	Apr 23, 2024	May 03, 2024	11 days
4	Requirement gathering	May 04, 2024	May 12, 2024	9 days
5	Background survey	May 12, 2024	May 16, 2024	5 days
6	Literature review	May 17, 2024	May 24, 2024	8 days
7	NON/functional requirements	May 25, 2024	Jun 08, 2024	15 days
8	Project Proposal draft	Jun 09, 2024	Jul 19, 2024	41 days
9	proposal presentation	Jul 20, 2024	Aug 06, 2024	18 days
10	proposal presentation Completed	Aug 07, 2024	Aug 07, 2024	0 days
11	Proposal report	Aug 07, 2024	Aug 16, 2024	10 days
12	Finalizing dataset	Aug 17, 2024	Aug 26, 2024	10 days
13	Finalizing methodologies	Aug 27, 2024	Sep 02, 2024	7 days
14	Data preprocessing	Sep 03, 2024	Sep 10, 2024	8 days
15	Text extraction	Sep 11, 2024	Sep 20, 2024	10 days
16	Model building	Sep 21, 2024	Nov 30, 2024	71 days
17	Model building Completed	Dec 01, 2024	Dec 01, 2024	0 days
18	Progress Presentation 1	Dec 01, 2024	Dec 08, 2024	8 days
19	Model building 2	Dec 09, 2024	Dec 30, 2024	22 days
20	Model building 2 Completed	Dec 30, 2024	Dec 30, 2024	0 days
21	Back End completion	Dec 31, 2024	Jan 30, 2025	31 days
22	Front End completion	Jan 31, 2025	Feb 28, 2025	29 days
23	First integration	Mar 01, 2025	Mar 20, 2025	20 days
24	Progress Presentation 2	Mar 21, 2025	Mar 30, 2025	10 days
25	Progress Presentation 2 Delivered	Mar 31, 2025	Mar 31, 2025	0 days
26	Integration completion	Mar 31, 2025	Jun 08, 2025	70 days
27	Unit testing	Jun 09, 2025	Jun 22, 2025	14 days
28	Integration testing	Jun 23, 2025	Jul 18, 2025	26 days
29	Final document	Jul 19, 2025	Jul 30, 2025	12 days
30	Final document Completed	Jul 31, 2025	Jul 31, 2025	0 days
31	Final Presentation and VIVA	Jul 31, 2025	Aug 08, 2025	9 days



IT21164330 | Arandara S.D. | 24-25J-224

References

[1] V. Tikkiwal et al., "An IoT Based Air Pollution Monitoring System for Smart Cities," ResearchGate, 2019. [Online]. Link

[2] A. Gunathilaka et al., "Real-Time Air Quality Monitoring System using IoT," Journal of Physics: Conference Series, 2021. [Online]. Link

[3] P. Wang et al., "Machine learning-based artificial intelligence method for predicting the air pollution index PM2.5" Journal of Cleaner Production, 2024. [Online]. Link

[4] S. Kumar et al., "An Integrated IoT and Machine Learning Approach for Environmental Monitoring and Management," Environmental Research, 2021. [Online] Link

Team Member 3

Noise Guard AI

Karunarathne R. Y. D.
IT21169144

Noise Guard AI

Smart Noise monitoring with IoT sensors,
Classification and Prediction

Background

Urban noise pollution harms public health and quality of life. Effective noise management is crucial for livable cities, yet current methods are inadequate due to fragmented data and limited analysis tools.

Uses AI to Analyze Noise Data for Better Management

- Leverages machine learning to analyze real-time noise levels using IoT data.

Classification and Prediction

- Uses advanced ML and DL models to classify noise sources and predict future trends based on historical data.

Informed Decision-Making

- Identifies high-noise areas and provides real time insights.

Research Problem

Urban areas struggle with managing noise pollution.

Need for a system to analyze and improve noise monitoring using advanced AI and IoT technologies.

IT21169144 | Karunarathne R. Y. D. | 24-25J-224

Research Gap

IT21169144 | Karunarathne R. Y. D. | 24-25J-224

Features	[1]	[2]	[3]	[4]	IEMS
Future Predictions	✗	✗	✗	✓	✓
Noise Classification	✓	✓	✓	✗	✓
IOT Integrated	✓	✗	✓	✓	✓
Report Generation	✗	✗	✗	✗	✓
Implement Web Application	✗	✗	✗	✗	✓
Shown in Interactive Maps	✗	✗	✗	✗	✓

[1] Urban Sound Classification using CNN

[2] A Machine Learning Driven IoT Solution for Noise Classification in Smart Cities

[3] Internet of Things for Noise Mapping in Smart Cities: State-of-the-Art and Future Directions

[4] Noise Prediction Using Machine Learning with Measurement

Obejctives

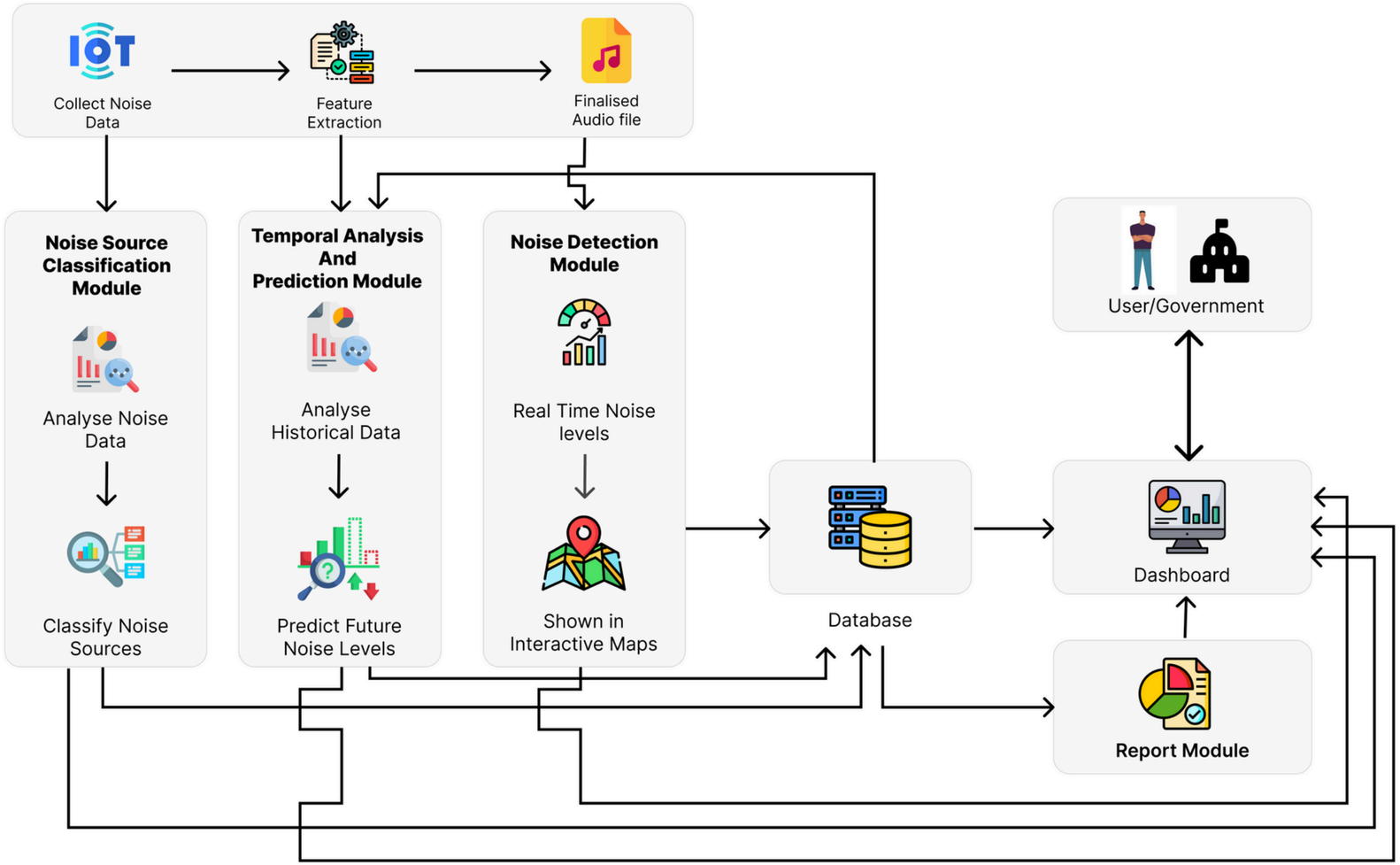
Main Objectives

- Develop a comprehensive system for real-time
  - Noise Level Monitoring
  - Noise Classification
  - Prediction.

Sub-Objectives

- Collect noise data using IoT and mobile devices.
- Classify noise sources using ML and DL models.
- Predict future noise levels based on historical data.

System Diagram



IT21169144 | Karunarathne R. Y. D. | 24-25J-224

Technologies Used

- **IoT, Mobile phone Microphone**
  - For real-time noise data collection.
  - GPS Sensor, MAX4466 Audio Sensor, ESP-32
- **ML/DL Models**
  - CNNs for classification and prediction, EfficientNetB0

1. **Data Collection:** Deploy IoT sensors in various locations and Pre Data-sets
2. **Pre-processing:** Clean and preprocess collected data.
3. **Model Training:** Train ML/DL models for classification and prediction.
4. **System Integration:** Integrate models with real-time data for monitoring.

IT21169144 | Karunarathne R. Y. D. | 24-25J-224

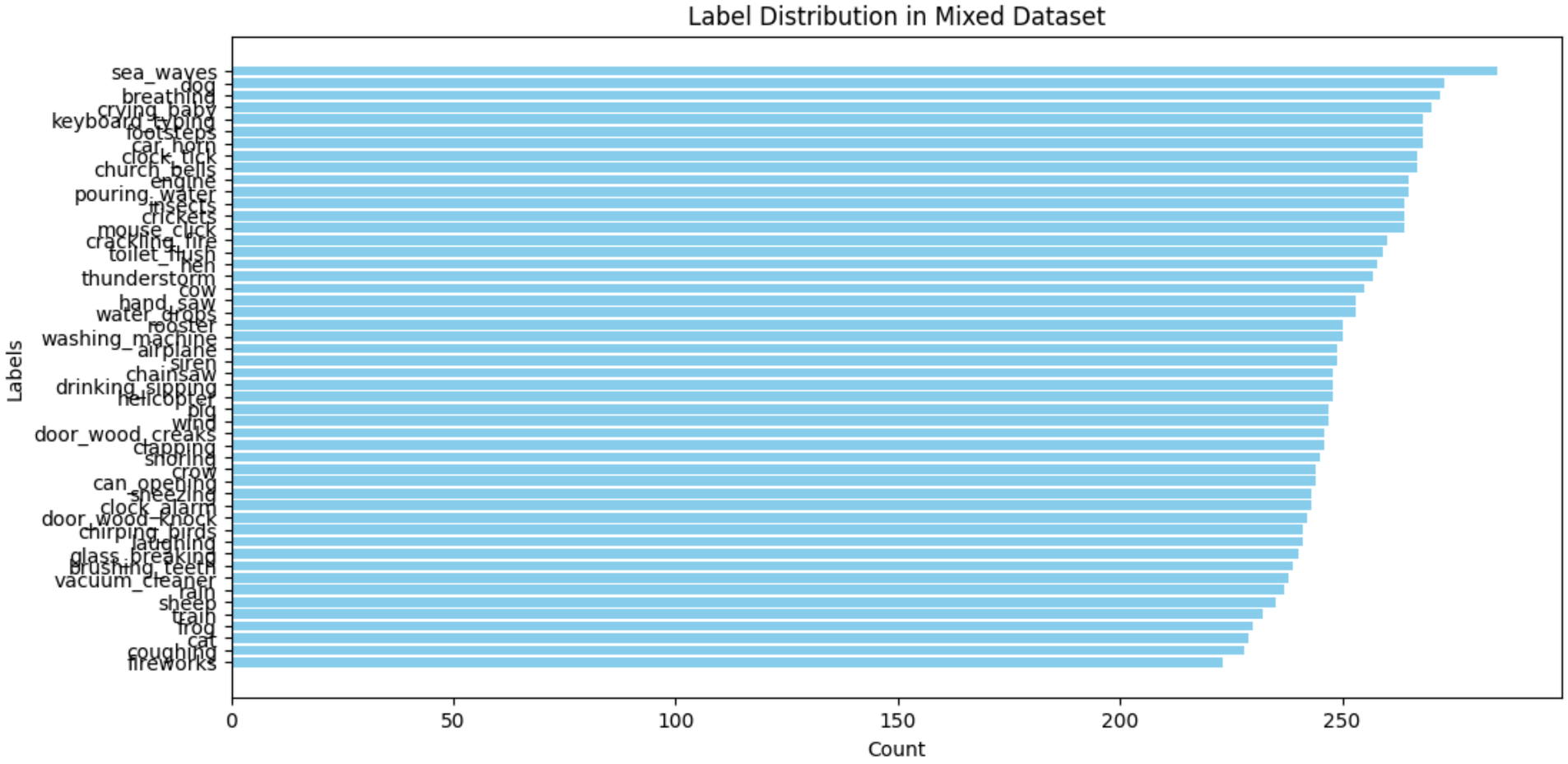
Current Progress

- Trained an initial model to predict single classification.
- Used a CNN model for audio classification.
- Converted audio into Mel spectrograms for feature extraction.
- Began fine-tuning hyperparameters to improve the accuracy and reliability of feasibility predictions.

- Implemented IoT sensors to collect real-world audio data along with geo-location metadata.
- Collected real-world audio samples and mixed them with an existing dataset to create a new dataset.
- Trained a multi-label classification model using CNN.
- Trained the model with 50 different classes for improved classification performance.
- Began fine-tuning hyperparameters to improve the accuracy and reliability of feasibility predictions.

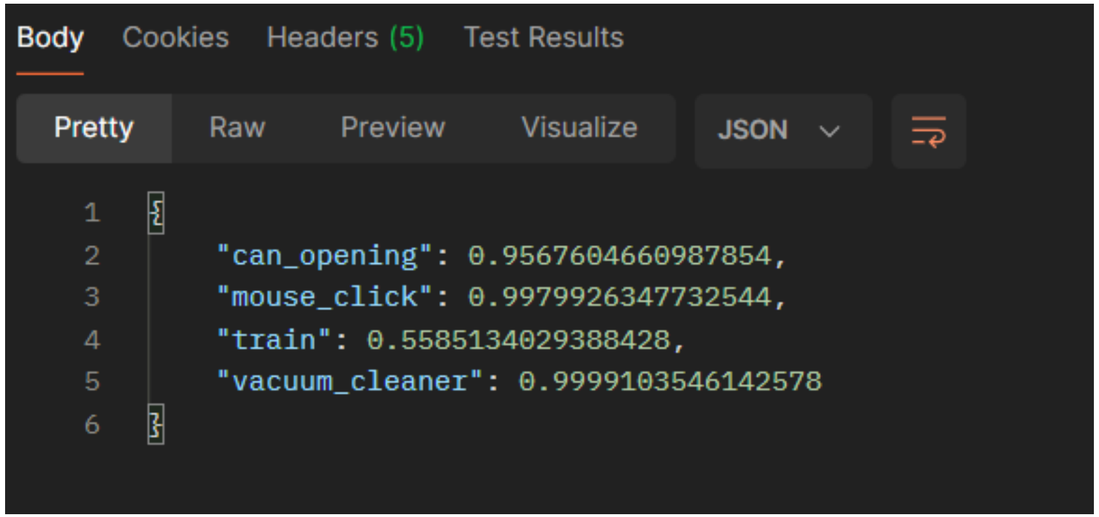
IT21169144 | Karunarathne R. Y. D. | 24-25J-224

Data Distribution



IT21169144 | Karunarathne R. Y. D. | 24-25J-224

Project Evidence



Accuracy

	precision	recall	f1-score	support
car_horn	0.84	0.81	0.83	47
engine	0.56	0.75	0.64	44
brushing_teeth	0.71	0.63	0.67	57
breathing	0.69	0.47	0.56	47
glass_breaking	0.90	0.51	0.65	53
door_wood_creaks	0.72	0.65	0.68	43
clock_tick	0.34	0.38	0.36	50
laughing	0.39	0.70	0.50	40
frog	0.77	0.75	0.76	44
chirping_birds	0.94	0.74	0.83	42
thunderstorm	0.76	0.71	0.74	59
hand_saw	0.60	0.67	0.63	54
can_opening	0.43	0.47	0.44	43
crow	0.90	0.85	0.87	52
train	0.72	0.59	0.65	49
water_drops	0.33	0.18	0.23	51
siren	0.93	0.91	0.92	47
helicopter	0.60	0.49	0.54	51
chainsaw	0.91	0.90	0.91	48
sneezing	0.70	0.50	0.58	52
snoring	0.51	0.46	0.48	46
pig	0.81	0.61	0.70	36
dog	0.79	0.52	0.63	52
rooster	0.76	0.81	0.78	47
coughing	0.77	0.47	0.58	49
keyboard_typing	0.49	0.42	0.45	59
vacuum_cleaner	0.92	0.98	0.95	58
footsteps	0.59	0.48	0.53	46
insects	0.78	0.66	0.71	47
crickets	0.87	0.81	0.84	42

Predictions

```python
with open("labels.json", "r") as f:
    loaded_labels = json.load(f)

# Load model
loaded_model = load_model("multi_label_sound_classification_model.keras", compile=False)
loaded_model.compile(optimizer=Adam(learning_rate=0.0005), loss="binary_crossentropy", metrics=["accuracy"])

def predict_audio(audio_path, model, labels, threshold=0.1):
    mel_spec = audio_to_mel_spectrogram(audio_path, target_shape=(128, 128))
    mel_spec = np.expand_dims(mel_spec, axis=-1)
    mel_spec = np.expand_dims(mel_spec, axis=0)
    y_pred = model.predict(mel_spec)[0]
    predicted_labels = {label: float(prob) for label, prob in zip(labels, y_pred) if prob > threshold}
    return dict(sorted(predicted_labels.items(), key=lambda item: item[1], reverse=True))

# Test prediction
test_audio_path = "/content/drive/MyDrive/mixed_dataset/mixed_1023.wav"
predictions = predict_audio(test_audio_path, loaded_model, loaded_labels)
print(predictions)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 7s 7s/step
{'brushing_teeth': 0.9999773502349854, 'hand_saw': 0.9985563158988953, 'mouse_click': 0.9757606983184814}
```

TOOLS & TECHNOLOGIES

IEMS

IT21169144 | Karunarathne R. Y. D. | 24-25J-224

System Requirements

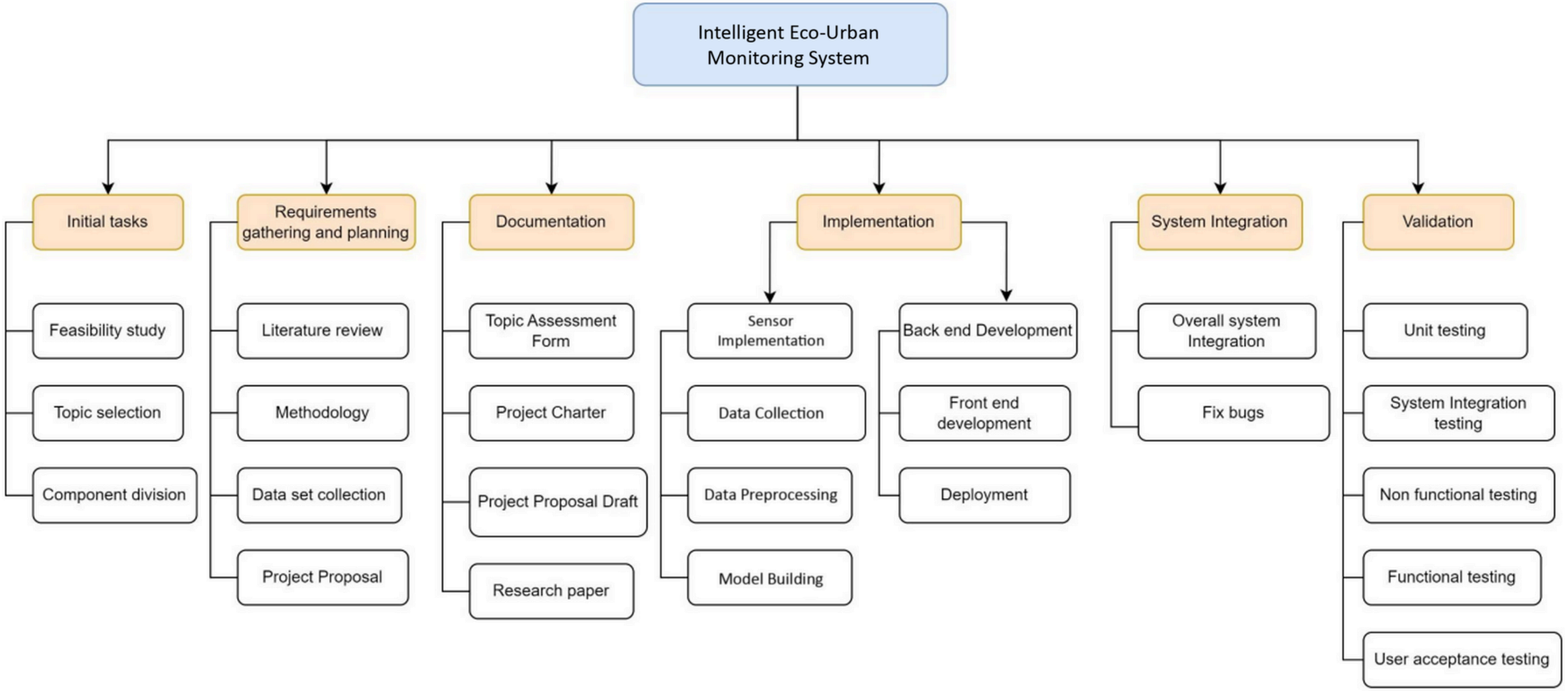
System Requirements

- **Hardware**: IoT sensors, mobile phone's microphone for noise data collection, high-performance computing for model training.
- **Software**: TensorFlow, PyTorch, data visualization tools.

Software Requirements

- **Functional Requirements**
  - Noise Data Collection
  - Noise Source Classification
  - Temporal Analysis And Prediction Module
- **Non-Functional Requirements**
  - Performance
  - Scalability
  - Reliability
  - Usability

WORK BREAKDOWN CHART

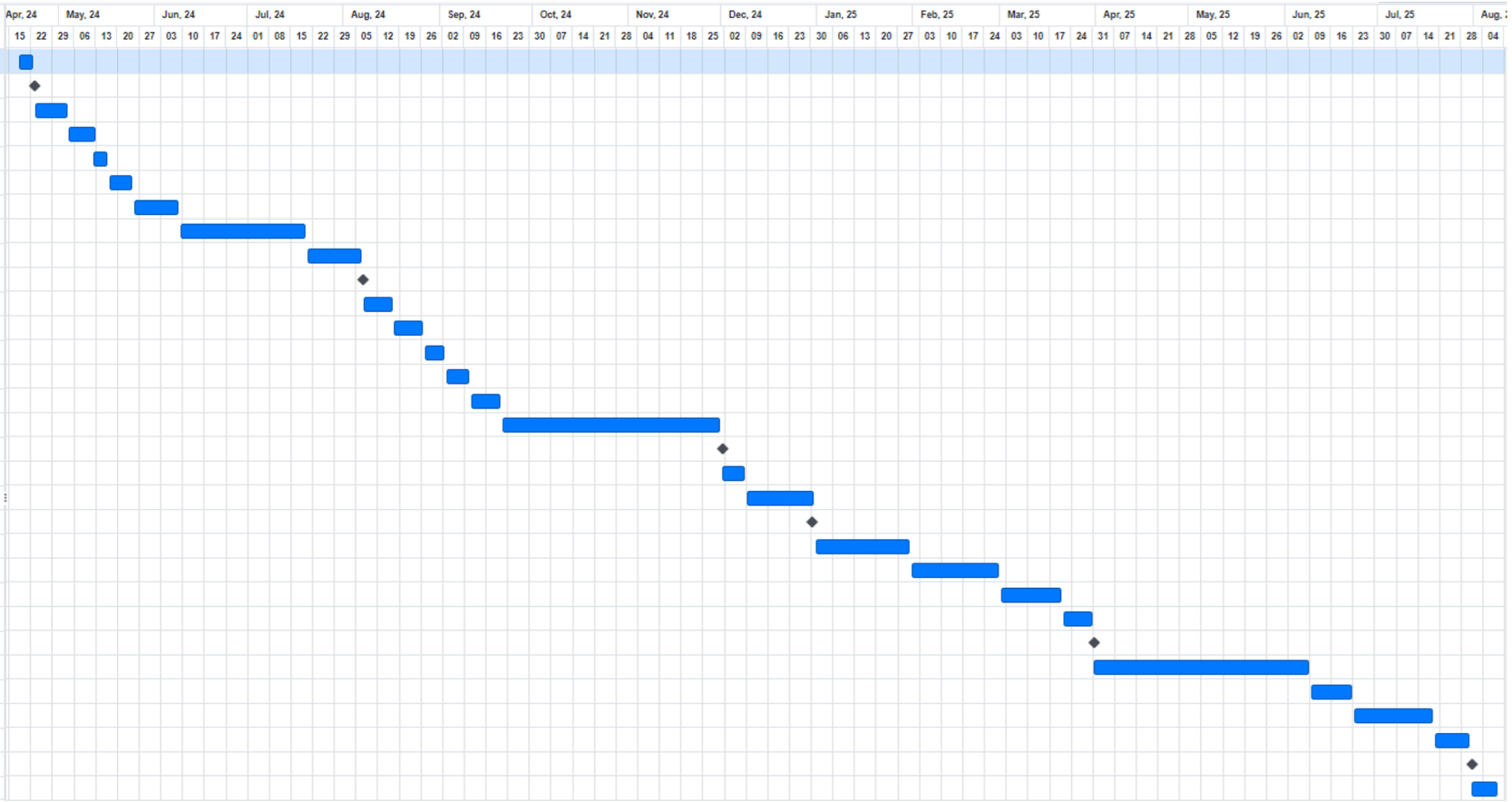


IT21169144 | Karunarathne R. Y. D. | 24-25J-224

PROJECT TIMELINE - GANTT CHART

IEMS

ID	Name	Start Date	End Date	Duration
1	Topic selection	Apr 18, 2024	Apr 22, 2024	5 days
2	Topic selection Completed	Apr 23, 2024	Apr 23, 2024	0 days
3	Feasibility study	Apr 23, 2024	May 03, 2024	11 days
4	Requirement gathering	May 04, 2024	May 12, 2024	9 days
5	Background survey	May 12, 2024	May 16, 2024	5 days
6	Literature review	May 17, 2024	May 24, 2024	8 days
7	NON/functional requirements	May 25, 2024	Jun 08, 2024	15 days
8	Project Proposal draft	Jun 09, 2024	Jul 19, 2024	41 days
9	proposal presentation	Jul 20, 2024	Aug 06, 2024	18 days
10	proposal presentation Completed	Aug 07, 2024	Aug 07, 2024	0 days
11	Proposal report	Aug 07, 2024	Aug 16, 2024	10 days
12	Finalizing dataset	Aug 17, 2024	Aug 26, 2024	10 days
13	Finalizing methodologies	Aug 27, 2024	Sep 02, 2024	7 days
14	Data preprocessing	Sep 03, 2024	Sep 10, 2024	8 days
15	Text extraction	Sep 11, 2024	Sep 20, 2024	10 days
16	Model building	Sep 21, 2024	Nov 30, 2024	71 days
17	Model building Completed	Dec 01, 2024	Dec 01, 2024	0 days
18	Progress Presentation 1	Dec 01, 2024	Dec 08, 2024	8 days
19	Model building 2	Dec 09, 2024	Dec 30, 2024	22 days
20	Model building 2 Completed	Dec 30, 2024	Dec 30, 2024	0 days
21	Back End completion	Dec 31, 2024	Jan 30, 2025	31 days
22	Front End completion	Jan 31, 2025	Feb 28, 2025	29 days
23	First integration	Mar 01, 2025	Mar 20, 2025	20 days
24	Progress Presentation 2	Mar 21, 2025	Mar 30, 2025	10 days
25	Progress Presentation 2 Delivered	Mar 31, 2025	Mar 31, 2025	0 days
26	Integration completion	Mar 31, 2025	Jun 08, 2025	70 days
27	Unit testing	Jun 09, 2025	Jun 22, 2025	14 days
28	Integration testing	Jun 23, 2025	Jul 18, 2025	26 days
29	Final document	Jul 19, 2025	Jul 30, 2025	12 days
30	Final document Completed	Jul 31, 2025	Jul 31, 2025	0 days
31	Final Presentation and VIVA	Jul 31, 2025	Aug 08, 2025	9 days



IT21169144 | Karunarathne R. Y. D. | 24-25J-224

References

[1] M. A. Basuni and S. H. Alzahrani, "Urban Sound Classification using CNN," ResearchGate, 2021. [Online]. Available: Link

[2] M. F. Khan, M. Al-Kahtani, S. H. Islam, N. Kumar, and P. W. C. Prasad, "A Machine Learning Driven IoT Solution for Noise Classification in Smart Cities," ResearchGate, 2018. [Online]. Available: link

[3] D. Gómez-Gutiérrez, J. Pascual-Gaspar, J. M. Lanza-Gutiérrez, and E. Sancristobal, "Internet of Things for Noise Mapping in Smart Cities: State-of-the-Art and Future Directions," ResearchGate, 2020. [Online]. Available: link

[4] S. A. Kumar, K. R. Sudheesh, and S. M. S. Islam, "Noise Prediction Using Machine Learning with Measurement," ResearchGate, 2020. [Online]. Available: link

Team Member 4

Eco go AI

Kodithuwakku C.K.
IT21156960

Eco go

Predicting and Reducing
CO2 Emissions from Vehicles.

Background

Eco Go, a component of the Integrated Environmental Monitoring System (IEMS), leverages advanced machine learning to predict and reduce vehicle CO2 emissions, offering personalized strategies and insights for urban emission challenges.

Predictive Emission Analysis

- Uses machine learning to predict vehicle CO2 emissions by analyzing vehicle attributes.
- Use two models to predict CO2 emissions accurately.

Personalized Recommendation Engine

- Provides tailored CO2 reduction strategies based on individual driving habits and vehicle characteristics.
- Use TOPSIS algorithm to recommend best vehicle

Research Problem

Urban areas experience significant CO2 emissions from vehicles, contributing to climate change and air pollution.

A system is needed that uses advanced machine learning to predict CO2 emissions and offer customized reduction strategies for each driver.

IT21156960 | Kodithuwakku C.K. | 24-25J-224

Research Gap

Features	[1]	[2]	[3]	[4]	IEMS
Use two ML models	✗	✗	✗	✗	✓
Adaptive Learning and Strategy Optimization	✗	✗	✓	✗	✓
Personalization	✓	✗	✗	✗	✓
Broader Vehicle and Condition Coverage	✗	✗	✗	✗	✓
Comprehensive System Integration	✓	✗	✗	✗	✓

[1] Modelling of CO2 Emission Prediction for Dynamic Vehicle Travel Behavior

[2] From lab-to-road & vice-versa

[3] Estimation of transport CO2 emissions using machine learning algorithm

[4] Models for predicting vehicle emissions

Obejctives

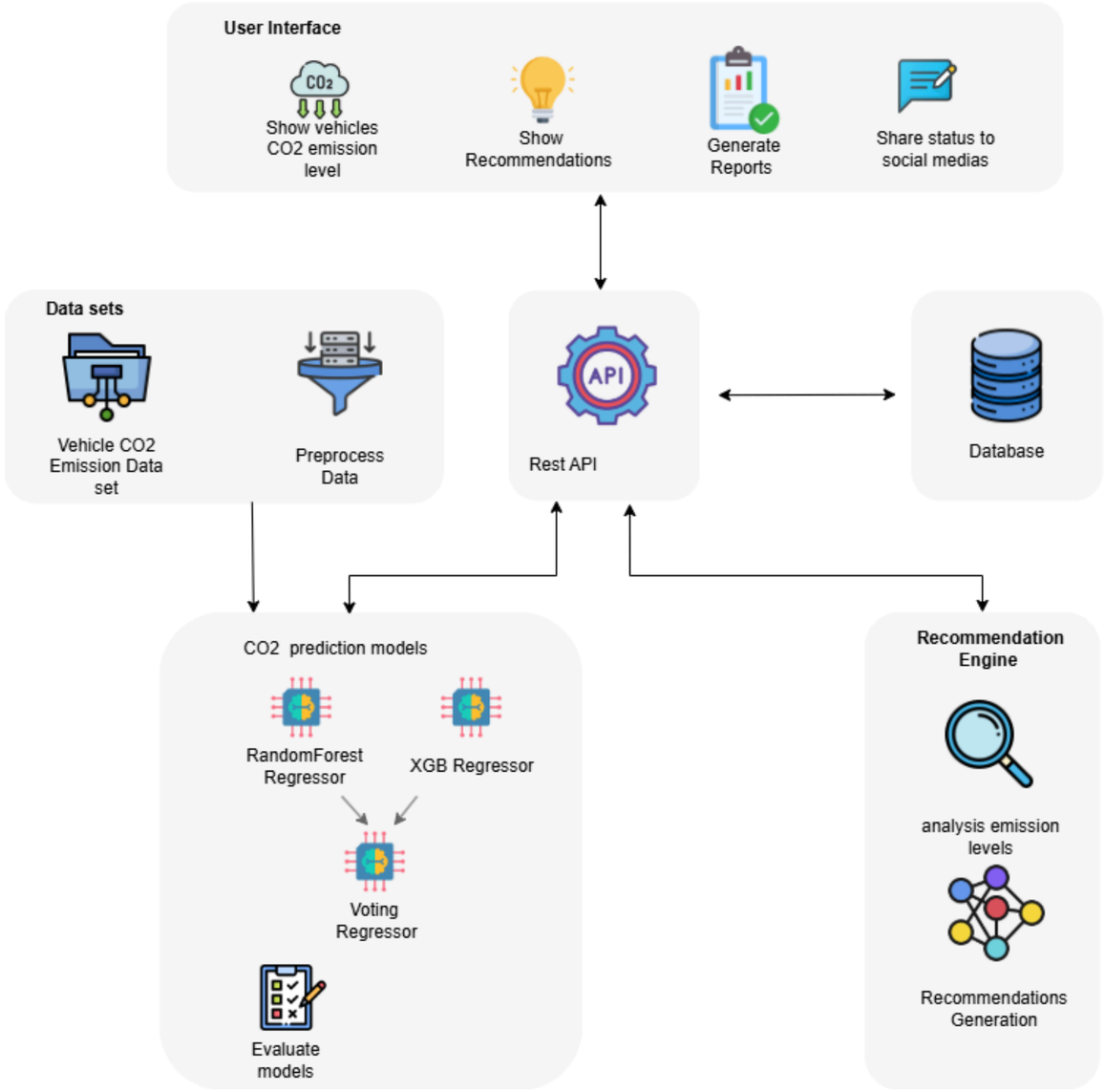
Main Objective

- Improve urban sustainability by predicting vehicle CO2 emissions and delivering personalized recommendations for effective emission reduction.

Sub-Objectives

- Predict Vehicle CO2 emission.
- Develop personalized strategies.
- Develop a recommendation engine that suggest best vehicle

System Diagram



Technologies Used

Machine Learning

- Ensemble learning models (Random Forest, Gradient Boosting)
- Metrics such as RMSE,R2
- LLM such as llama 3.3

API

- Here map

Data Visualization

- D3.js, Chart.js

Programing languages

- python , MERN stack , Flask

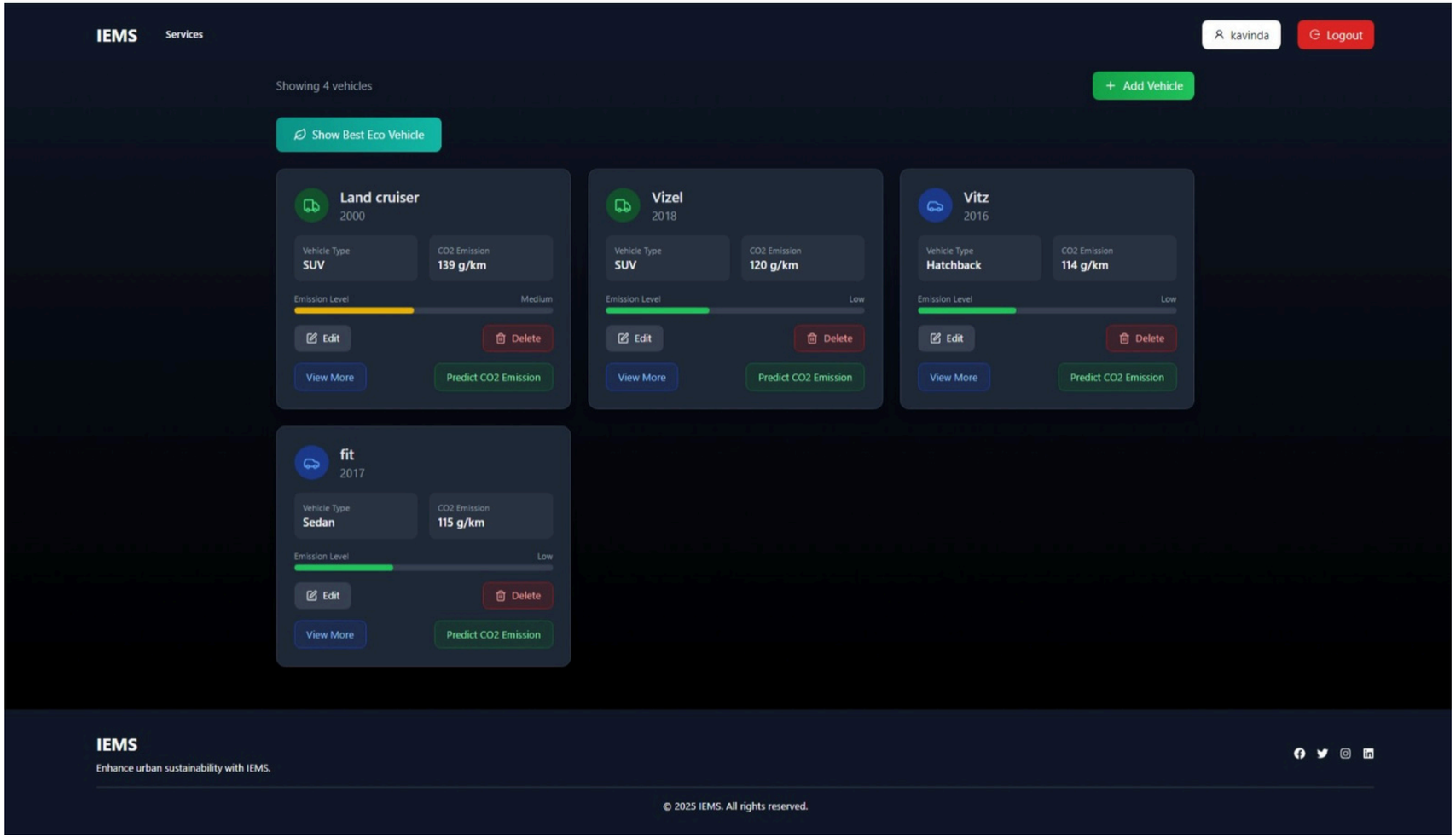
Current Progress

- Trained ML model to predict CO_2 emissions using ensemble techniques.
- Implemented Eco Vehicle Suggestion using a Multi-Criteria Decision Algorithm.
- Developed Flask backend to expose APIs for CO_2 prediction and eco vehicle suggestion.

- Implemented CO_2 emission reduction recommendations using Retrieval-Augmented Generation (RAG) with LLaMA 3.3 LLM.
- Designed and developed UIs for the EcoGo component and integrated them with the Flask backend.
- Implemented Eco Navigator Map, which provides the best routes to reduce CO_2 emissions.

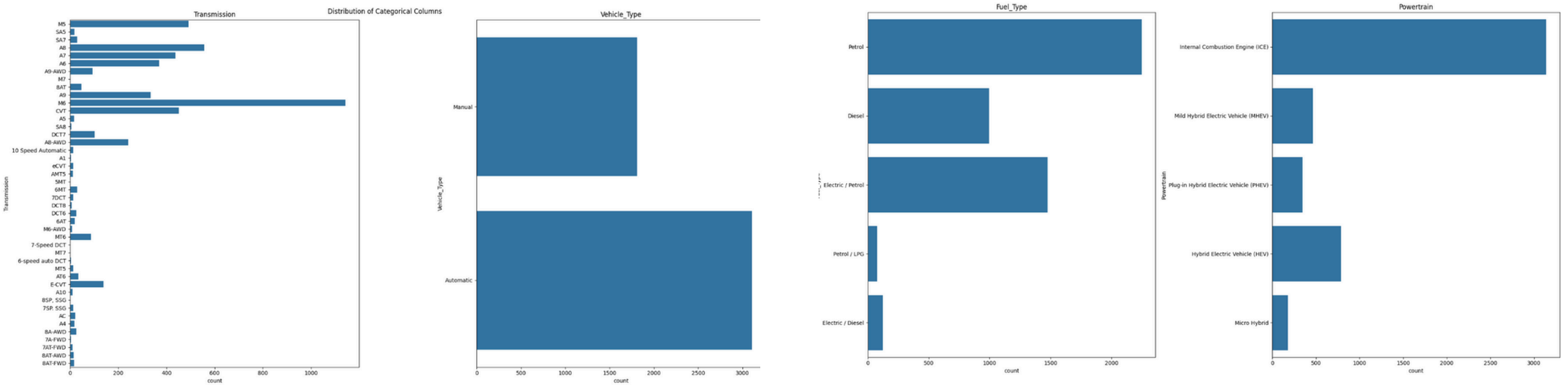
IT21156960 | Kodithuwakku C.K. | 24-25J-224

Project Evidence



IT21156960 | Kodithuwakku C.K. | 24-25J-224

Data Distribution



IT21156960 | Kodithuwakku C.K. | 24-25J-224

Accuracy

Predictions

```
def inference_co2(
                sample_json,
                cat_cols = [
                        'Transmission',
                        'Vehicle_Type',
                        'Fuel_Type',
                        'Powertrain'
                        ]):
        df = pd.DataFrame([sample_json])
        df = df.astype(str)
        for col in df.columns:
                if col in cat_cols:
                        df[col] = df[col].str.strip()
                        df[col] = encoder_co2[col].transform(df[col])
        X = df.values
        Ypred = model_co2.predict(X)
        return int(Ypred[0])
```

```
inference_co2(sample_json)
```

```
688
```

TOOLS & TECHNOLOGIES

IEMS

IT21156960 | Kodithuwakku C.K. | 24-25J-224

Requirements

IEMS

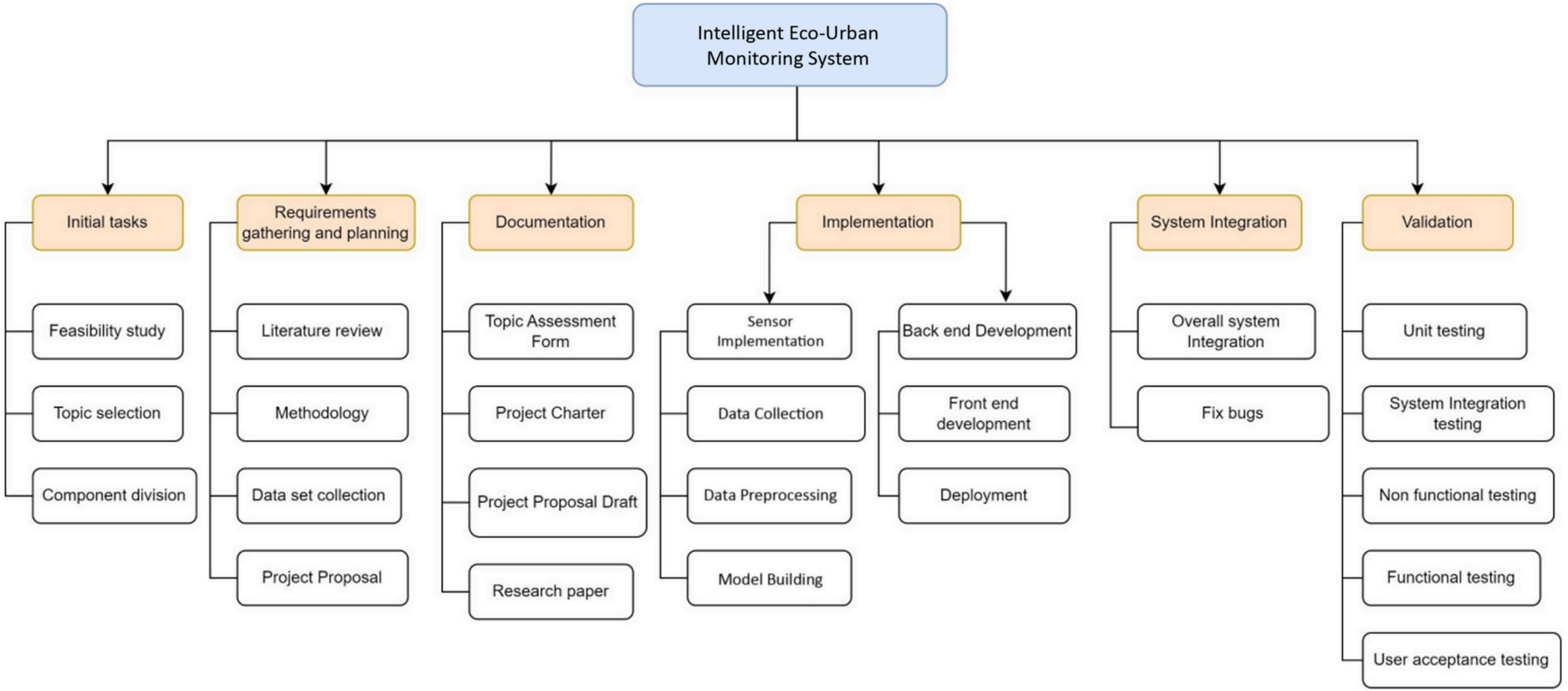
System Requirements

- High-Performance Computing

Software Requirements

- **Functional Requirements**
  - CO2 Prediction Model
  - Recommendation Engine
  - Visualization and Reporting
- **Non-Functional Requirements**
  - Performance
  - Scalability
  - Reliability
  - Usability

WORK BREAKDOWN CHART

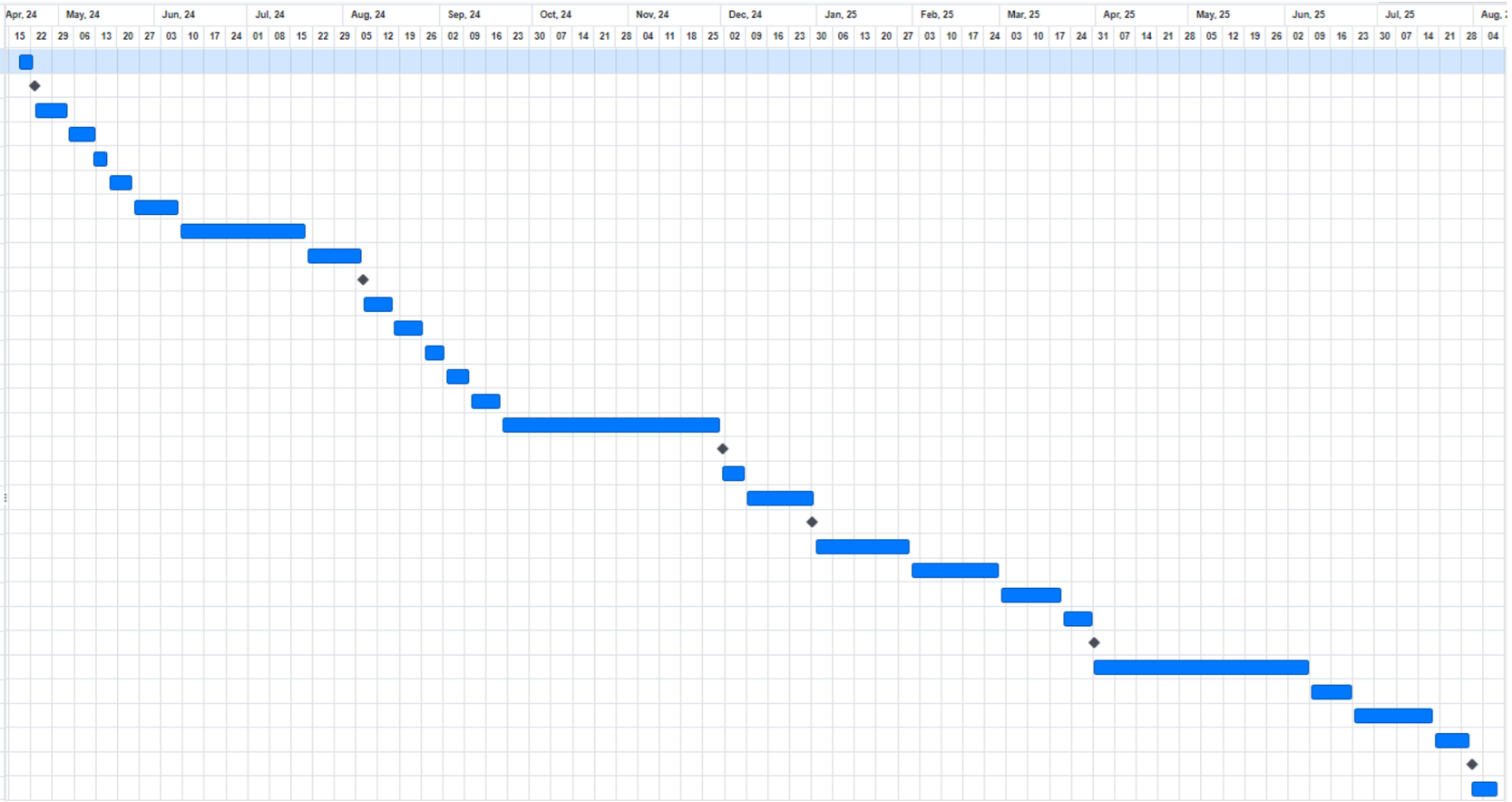


IT21156960 | Kodithuwakku C.K. | 24-25J-224

PROJECT TIMELINE - GANTT CHART

IEMS

ID	Name	Start Date	End Date	Duration
1	Topic selection	Apr 18, 2024	Apr 22, 2024	5 days
2	Topic selection Completed	Apr 23, 2024	Apr 23, 2024	0 days
3	Feasibility study	Apr 23, 2024	May 03, 2024	11 days
4	Requirement gathering	May 04, 2024	May 12, 2024	9 days
5	Background survey	May 12, 2024	May 16, 2024	5 days
6	Literature review	May 17, 2024	May 24, 2024	8 days
7	NON/functional requirements	May 25, 2024	Jun 08, 2024	15 days
8	Project Proposal draft	Jun 09, 2024	Jul 19, 2024	41 days
9	proposal presentation	Jul 20, 2024	Aug 06, 2024	18 days
10	proposal presentation Completed	Aug 07, 2024	Aug 07, 2024	0 days
11	Proposal report	Aug 07, 2024	Aug 16, 2024	10 days
12	Finalizing dataset	Aug 17, 2024	Aug 26, 2024	10 days
13	Finalizing methodologies	Aug 27, 2024	Sep 02, 2024	7 days
14	Data preprocessing	Sep 03, 2024	Sep 10, 2024	8 days
15	Text extraction	Sep 11, 2024	Sep 20, 2024	10 days
16	Model building	Sep 21, 2024	Nov 30, 2024	71 days
17	Model building Completed	Dec 01, 2024	Dec 01, 2024	0 days
18	Progress Presentation 1	Dec 01, 2024	Dec 08, 2024	8 days
19	Model building 2	Dec 09, 2024	Dec 30, 2024	22 days
20	Model building 2 Completed	Dec 30, 2024	Dec 30, 2024	0 days
21	Back End completion	Dec 31, 2024	Jan 30, 2025	31 days
22	Front End completion	Jan 31, 2025	Feb 28, 2025	29 days
23	First integration	Mar 01, 2025	Mar 20, 2025	20 days
24	Progress Presentation 2	Mar 21, 2025	Mar 30, 2025	10 days
25	Progress Presentation 2 Delivered	Mar 31, 2025	Mar 31, 2025	0 days
26	Integration completion	Mar 31, 2025	Jun 08, 2025	70 days
27	Unit testing	Jun 09, 2025	Jun 22, 2025	14 days
28	Integration testing	Jun 23, 2025	Jul 18, 2025	26 days
29	Final document	Jul 19, 2025	Jul 30, 2025	12 days
30	Final document Completed	Jul 31, 2025	Jul 31, 2025	0 days
31	Final Presentation and VIVA	Jul 31, 2025	Aug 08, 2025	9 days



IT21156960 | Kodithuwakku C.K. | 24-25J-224

References

[1] Navarajan Subramaniam, Norhakim Yusof, "Modelling of CO2 Emission Prediction for Dynamic Vehicle Travel Behavior Using Ensemble Machine Learning Technique," IEEE, 25 November 2021. [Online]. Available: link . (Accessed: Jul. 22, 2024.)

[2] S. Tsiakmakis, G. Fontaras, J. Dornoff, and V. Valverde, "From lab-to-road & vice-versa: Using a simulation-based approach for predicting real-world CO2 emissions," Energy, vol. 177, pp. 495–508, 15 February 2019. [Online]. Available: link . (Accessed: Jul. 22, 2024.)

[3] S. Li, Z. Tong, and M. Haroon, "Estimation of transport CO2 emissions using machine learning algorithm," Transportation Research Part D: Transport and Environment, vol. 122, August 2024. [Online]. Available: link. (Accessed: Jul. 23, 2024.)

[4] H. Zhong, K. Chen, C. Liu, and M. Zhu, "Models for predicting vehicle emissions: A comprehensive review," Science of The Total Environment, vol. 791, 1 May 2024. [Online]. Available: link. (Accessed: Jul. 23, 2024.)

Commercialization

Market Demand

- Market demand analysis involves understanding the need for the Intelligent EcoUrban Monitoring System (IEMS) in the market. This includes identifying the target audience, market size, growth trends, and potential market opportunities.

Target Audience

- Municipalities and city planners ,Environmental agencies , Goverment , Citizens

Market Opportunities

- Integration with existing smart city infrastructure. ,Partnership with governments for large-scale implementation , Custom solutions for different urban environments.

24-25J-224

THANK YOU !

SLIIT UNI
THE KNOWLEDGE UNIVERSITY

24-25J-224

Q & A

SLIIT UNI
THE KNOWLEDGE UNIVERSITY