JAI SHRIRAM ENGINEERING COLLEGE
TIRUPPUR – 638 660
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai
Recognized by UGC & Accredited by NAAC and NBA (CSE and ECE)

Department of Electronics And Communication Engineering

IBM – Naan Mudhalvan

Internet of Things Group 3

Phase 4 – Development part 2

Title : Noise Pollution Monitoring

NAME :C. Ranganayaki

NM ID : AU711221106027

YEAR : III

Development part 2

AI :

Define Your Problem:

- Clearly state the problem you want to solve using machine learning.
- Define your project's objectives and goals:

Data Collection:

- Gather relevant data for your project.
- Ensure data quality and pre-process it (cleaning, handling missing values, etc.).

Feature Selection/Engineering:

- Choose the most relevant features or create new ones.

Select a Machine Learning Algorithm:

- Depending on your problem (classification, regression, etc.), choose an appropriate algorithm.

Data Splitting:

- Split your data into training and testing datasets to evaluate your model's performance.

Model Training:

- Train your machine learning model on the training data.

Model Evaluation:

- Evaluate the model's performance using appropriate metrics (e.g., accuracy, F1-score, mean squared error, etc.).

Hyper parameter Tuning:

- Optimize the model's Hyper parameters for better performance.

Cross-Validation:

- Perform cross-validation to ensure the model's generalization.

Analysis:

- Analyze the model's strengths and weaknesses.
- Visualize the results and any interesting patterns.

Document Creation:

- Create a structured document summarizing all the steps and results.
- Include an introduction, problem statement, methodology, and findings.

Share for Assessment:

- Share your document with the relevant stakeholders or your assessor.

ADS:

Feature Engineering:

- Identify the relevant features and data sources that will be used to build your ADS model. Perform feature engineering to extract, transform, or create new features that can improve the model's performance.

Data Preparation:

- Clean and preprocess your data, handling missing values and outliers.
- Ensure that the data is in a suitable format for training.

Model Selection:

- Choose the appropriate machine learning or deep learning model for your ADS project. This could be a neural network, decision tree, random forest, or any other relevant model.

Model Training:

- Split your data into training and testing sets. Train your selected model on the training data. Ensure that the model is fine-tuned for your specific task.

Model Evaluation:

- Evaluate the performance of your model using relevant metrics (e.g., accuracy, precision, recall, F1-score) for the specific ADS task you working on. Consider using cross-validation for a more robust evaluation.

Analysis:

- Analyze the model's performance and identify any areas for improvement. Investigate false positives/negatives, and understand why the model is making certain predictions.

Document Creation:

- Create a detailed document summarizing your project. Include an introduction, problem statement, methodology, results, and discussion.

Visualization:

- Use visualizations to illustrate important findings or trends in the data. Visualize the model's predictions and performance metrics.

Share for Assessment:

- Share your document with relevant parties for assessment. Be sure to explain your approach, findings, and how well your model performs.

Data Collection and Preprocessing:

- Gather and clean your data, ensuring it's in a format suitable for analysis. Handle missing values, outliers, and data quality issues.

Analysis Planning:

- Define your project's objectives and the specific analyses you need to perform.

Data Exploration and Visualization:

- Use IBM Cognos to explore your data. Create visualizations such as charts, graphs, and dashboards to gain insights.

Data Analysis:

- Apply statistical and analytical techniques to uncover patterns and trends in the data. Perform hypothesis testing or any other relevant analysis as per project requirements.

Model Building (if applicable):

- If your project requires predictive modeling, build and train models using the data. Evaluate the models for accuracy and suitability.

Model Evaluation:

- Assess the performance of your models using appropriate evaluation metrics.

Visualization with IBM Cognos:

- Leverage IBM Cognos to create more advanced and interactive visualizations. Customize and design reports and dashboards for presentation.

Document Creation:

- Create a comprehensive document summarizing your project. Include an introduction, problem statement, methodology, findings, and insights.

Share for Assessment:

- Share your document with the relevant stakeholders or your assessor.

IOT

Project Requirements:

- Ensure you have a clear understanding of the project requirements. List out what functionalities your IOT platform should have and the technologies you need to use.

Choose IOT Hardware:

- Select the IOT devices and sensors that are appropriate for your project. Consider factors like compatibility and data collection capabilities.

Data Collection and Processing:

- Set up a system for data collection and processing. This typically involves using microcontrollers or development boards like

Arduino, Raspberry Pi, or specialized IOT platforms like AWS IOT.

Web Development:

- Implement a web-based dashboard to monitor and control your IOT devices. You can use technologies like HTML, CSS, JavaScript, and web development frameworks like React, Angular.

Database:

- Choose a database to store the IOT data. Options include SQL databases (e.g., MySQL) or NoSQL databases (e.g., MongoDB).

Connectivity:

- Establish a secure connection between your IOT devices and the web dashboard. This might involve using MQTT, HTTP, or other communication protocols.

Data Visualization:

- Implement data visualization tools to display data from your IOT devices. Libraries like D3.js or charting libraries can be helpful.

User Authentication and Security:

- Ensure user authentication and data security. Use technologies like OAuth or JWT for user authentication, and secure data transmission with encryption.

Testing:

- Test your IOT platform extensively to ensure that it functions as expected. Check for data accuracy, security vulnerabilities, and userfriendliness.

Documentation:

- Create comprehensive documentation that includes project details, system architecture, source code, setup instructions, and any other relevant information.

Assessment Document:

- Compile your documentation into a report or presentation format. Include an executive summary, project description, architecture diagrams, source code snippets, and testing results.

Sharing for Assessment:

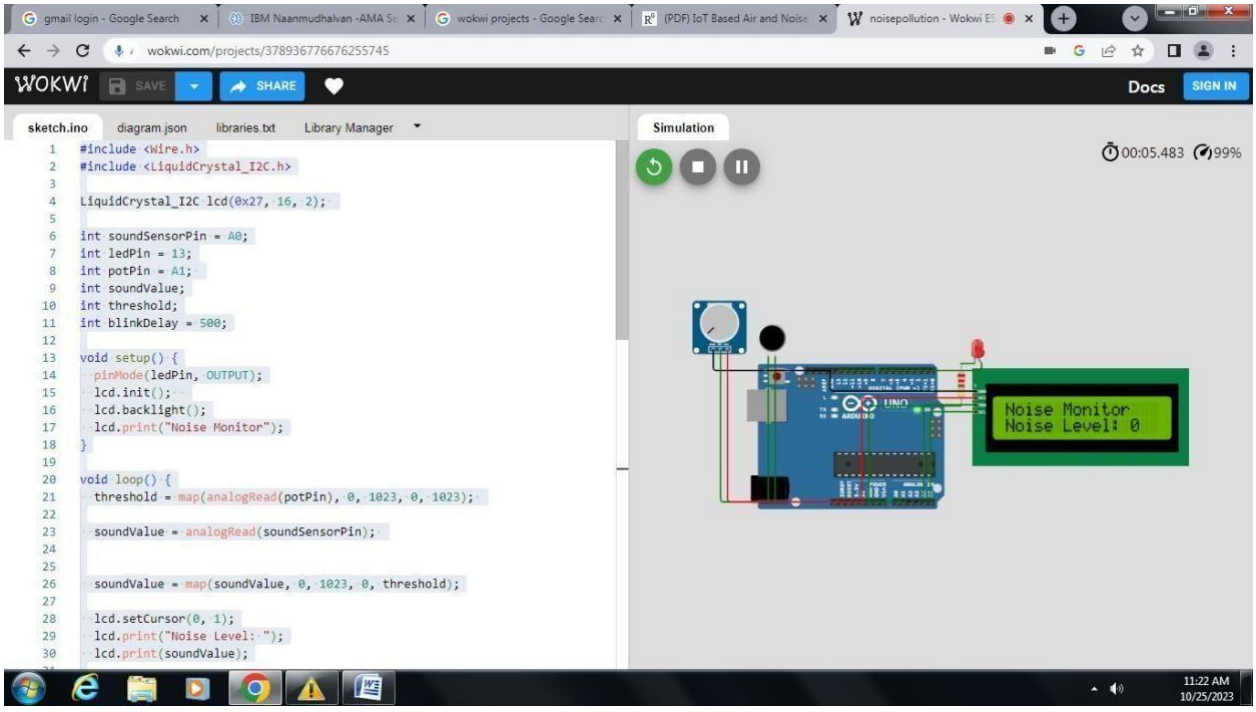
- Share your assessment document with the appropriate party, whether it's your project supervisor, teacher, or evaluator.

CODING:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h  >


LiquidCrystal_I2C lcd(0x27, 16, 2);
int soundSensorPin = A0;
int ledPin = 13; int potPin
= A1;  int soundValue; int
threshold; int blinkDelay
= 500;   void setup() {
pinMode(ledPin, OUTPUT);
```

```
lcd.init();
lcd.backlight();

lcd.Monitor"print)(; "Noise
}  void loop() {   threshold = map(analogRead(potPin),
0, 1023, 0, 1023);      soundValue
=  analogRead(soundSensorPin);



  soundValue = map(soundValue, 0, 1023, 0, threshold);
lcd.setCursor(0, 1);    lcd.print("Noise Level: ");
lcd.print(soundValue);     if (soundValue <= 200) {
digitalWrite500;   (ledPin, LOW);      blinkDelay =


  } else if (soundValue > 200 && soundValue < 300) {
digitalWrite(ledPin, HIGH);      lcd.setCursor(0, 1);
lcd.print("High Noise      ");     blinkDelay = 500;
}  else  if  (soundValue  >=  300)  {     digitalWrite( ledPin, HIGH);
lcd.setCursor(0, 1);     lcd.print("Very High Noise ");      blinkDelay
= 200;   }     delay(blinkDelay);   digitalWrite
(ledPin, LOW);    delay(blinkDelay);  }
```

SIMULATION:



CAD

Project Requirements:

- Ensure you have a clear understanding of your CAD project's requirements, including features, functionality, and any specific technical needs.

IBM Cloud Account:

- If you don't have one already, create an IBM Cloud account. You'll need this to access IBM Cloud Foundry.

Application Development:

- Start building your CAD application using the appropriate programming languages and tools for your project. Common languages for web applications are Python, or Ruby.

Database Integration:

- If your project requires data storage, integrate a suitable database service from IBM Cloud, such as IBM Db2 or Cloudant.

Application Testing:

- Thoroughly test your CAD application to ensure it functions correctly, meets your requirements, and is free of bugs and errors.

IBM Cloud Foundry Setup:

- Install the IBM Cloud CLI (Command-Line Interface) on your local machine. Log in to your IBM Cloud account using the CLI. Target the IBM Cloud Foundry environment for your application.

Deployment:

- Use the IBM Cloud CLI to deploy your CAD application to IBM Cloud Foundry.Configure environment variables, such as API keys or connection strings, to connect your application to IBM services.

Scaling and Resource Management:

- Adjust the resource allocation and scaling options as needed. IBM Cloud Foundry allows you to allocate memory and scale the application to handle varying workloads.

Security:

- Implement security measures to protect your CAD application. This includes securing sensitive data and using encryption where necessary. Ensure proper access controls and authentication methods are in place.

Monitoring and Logging:

- Set up monitoring and logging for your CAD application to track its performance and identify any issues. IBM Cloud provides tools for this purpose.

Documentation:

- Create comprehensive documentation for your CAD project, including system architecture, code documentation, and usage guides. Document the deployment process, environment configurations, and any issues faced during deployment.

Assessment Document:

- Compile all your documentation into a structured assessment document. Include project details, architecture diagrams, source code snippets, deployment instructions, testing results, and any challenges faced.

Sharing for Assessment:

- Share your assessment document with the relevant parties for evaluation. Ensure that it is well-organized and easy to understand.