

INDIAN INSTITUTE OF INFORMATION  
TECHNOLOGY SENAPATI, MANIPUR

# AIR POLLUTION DETECTOR

Project Proposal  
CS-300



*Submitted By:*

Radhe Raman Tiwari 17010115

*Supervisor:* Dr. Himangshu Sarma

Department of Computer Science & Engineering

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview . . . . .	1
1.2	Goals . . . . .	1
1.3	Specifications . . . . .	2
<b>2</b>	<b>Project Description</b>	<b>3</b>
2.1	Introduction . . . . .	3
2.2	Existing systems analysis . . . . .	3
2.3	Planning and setup . . . . .	3
2.4	Implementation . . . . .	3
2.5	Testing and Result analysis . . . . .	3
<b>3</b>	<b>Equipments</b>	<b>4</b>
3.1	Hardware . . . . .	4
3.2	Software . . . . .	4
3.3	Languages . . . . .	5

# Chapter 1

## Introduction

### 1.1 Overview

The air pollution detector is an application of the Internet of Things (IoT) and cybersecurity. This thesis project is a combination of both hardware and software that enables the collection of information about the air pollution level. This project will be having a feature of collecting data about air pollution levels after every certain period and send on the cloud with the help of Arduino Uno Ethernet shield. And Machine Learning algorithms will analyzer those data and tell the level of air pollution.

## IOT + Cybersecurity + ML = Project

### 1.2 Goals

I'm proposing this thesis project to fulfill the following six goals.

1. To collect information about the air pollution level after every certain period.
2. To whatever data has been collected, encrypt those data so that no one will be able to understand apart from the sender and receiver during communication.
3. To send encrypted information about the air pollution levels on the cloud.
4. To decrypt the information of air pollution level by Machine Learning(ML) algorithms and analyze the level of air pollution on the cloud.
5. To send the information about air pollution level on Android smartphone whatever has been analyzed by Machine Learning(ML) algorithms whenever the client needs.
6. To send encrypted information to Android clients form the cloud and Android client will decrypt the information.

### 1.3 Specifications

There are following specifications of this air pollution detector.

1. Real time access.
2. Real time result by Machine Learning algorithms.
3. Secure communication both at sender and receiver.
4. User friendly and easy to access.
5. Low cost.

## Chapter 2

# Project Description

### 2.1 Introduction

I'm planning to implement this thesis project in the following four phases.

1. Existing systems analysis.
2. Planning and setup
3. Implementation
4. Testing and Result analysis

### 2.2 Existing systems analysis

Here, I would analyze the pre-existing systems and their working procedure so that I will be able to understand how should I go.

### 2.3 Planning and setup

Here, I will plan for the project workflow what things project needs and what doesn't. This phase consists of understanding fo Git-Flow, adding version control, installation of softwares, cloning of API repository and their setup, etc.

### 2.4 Implementation

Here, I will implement the project with whatever things will be decided.

### 2.5 Testing and Result analysis

Here, I will test results and analyze the project outcomes. Whatever things will be coming will be tested with the aspected results.

## Chapter 3

# Equipments

### 3.1 Hardware

There are following hardware components that will be used in the air pollution detector and they are basically controller and sensors.

1. Arduino Uno
2. Arduino Uno Ethernet shield
3. Particulate matter(PM)2.5/10 sensor (SDS011)
4. Gas sensor (MQ135)
5. Temperature and humidity sensor (DHT11)
6. 7805 regulators (IC1 and IC2)
7. 12V battery
8. Wires and Breadboard

### 3.2 Software

There are following software that will be used in the air pollution detector.

1. Arduino IDE
2. Android Studio IDE
3. Text Editor
4. GUI Editor



### 3.3 Languages

There are following programming and interpreted languages that will be used in the air pollution detector.

1. Python3 for data transfer.
2. C/C++ for Arduino sketch.
3. XML for Android Annotation.
4. Java/Kotlin for Android Back-end.