**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Topic** | **Page** |
|  | Acknowledgement | 2 |
| 1 | Introduction | 3 |
| 2 | Relational Database | 4 |
| 3 | Programming Languages used | 4 |
| 4 | Codes for implementation | 5 |
| 5 | Conclusions | 12 |
| 6 | Other ways of air pollution management | 12 |
| 7 | References | 12 |

**ACKNOWLEDGEMENT**

This Project would never been possible to complete in time if we have not received the complete support of our respected mentor, Dr. A. Tarachand, who has constantly encouraged us to continue this project and supplied us with innumerable research papers and journals.

We are also grateful to our parents and batchmates, without whose presence, this project would be incomplete.

**Air Pollution Monitoring System**

**Abstract:**

The basic idea is to measure five pollutants: nitric oxide, nitrogen dioxide, carbon monoxide, ozone, and particulate matter. The NOx group (nitric oxide and nitrogen dioxide) is emitted from automobiles, power plants, and turbines. Carbon monoxide comes from automobile exhaust and burning fuel. Particulate matter is the result of a wide range of manmade and natural sources, while ozone is the result of reactions between chemicals already in our air. Together, these pollutants paint a comprehensive picture of air quality impacts from the interaction of human activity with natural processes.

This report presents the logic for **backend development**.

**1.INTRODUCTION**

What makes the front end of a website possible? Where is all that data stored? This is where the back end comes in. The back end of a website consists of a server, an application, and a database. A back-end developer builds and maintains the technology that powers those components which, together, enable the user-facing side of the website to even exist in the first place.

In software engineering, the terms front end and back end refers to the separation of concerns between the presentation layer (front end), and the data access layer (back end) of a piece of software, or the physical infrastructure or hardware. In the client–server model, the client is usually considered the front end and the server is usually considered the back end. A rule of thumb is that the front (or "client") side is any component manipulated by the user. The server-side (or "back end") code usually resides on the server, often far removed physically from the user.

**2. Relational Database**

RDBMS stands for Relational Database Management System.

RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.

**3. Programming languages used**

**A. SQL**

SQL stands for Structured Query Language. It is a programming language that allows interaction with **databases** by storing, manipulating and retrieving data from databases.

It is an ANSI (American National Standards Institute) standard.

* SQL can execute queries against a database
* SQL can retrieve data from a database
* SQL can insert records in a database
* SQL can update records in a database
* SQL can delete records from a database
* SQL can create new databases
* SQL can create new tables in a database
* SQL can create stored procedures in a database
* SQL can create views in a database
* SQL can set permissions on tables, procedures, and views

**B. PHP**

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose **server side scripting language** that is especially suited for web development and can be embedded into HTML. What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was.

**C. XML**

XML stands for eXtensible Markup Language. It is a software and hardware-independent tool for storing and transporting data. XML is a markup language much like HTML which was designed to store and transport data. XML is just information wrapped in tags, it does not do anything. However, it can be used for easy **transfer of data** from server side to client side.

**4.Codes for implementation:**

**1. home.php**

<?php

include("db.php");

include("new.php");

?>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<linkrel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

<!--Leaflet Links-->

<link rel="stylesheet" href="https://unpkg.com/leaflet@1.0.3/dist/leaflet.css" />

<script src="https://unpkg.com/leaflet@1.0.3/dist/leaflet.js"></script>

<style>

#map {

height:650px;

width: 100%;

}

#legend {

font-family: Arial, sans-serif;

background: #fff;

padding: 10px;

margin: 10px;

border: 3px solid #000;

}

#legend h3 {

margin-top: 0;

}

#legend img {

vertical-align: middle;

}

#valuesTable{

background-color: #6699FF;

background: rgba(0, 0, 0, 0.7);

position: fixed;

left: 1%;

top: 50px;

height: 100%;

width: 23%;

color: white;

z-index: 1;

overflow: auto;

}

#lastupdated{

position: relative;

left: 10px;

font-size: 10px;

}

#locationheader{

font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;

font-style: bold;

font-size: 25px;

position: relative;

}

.values{

font-size: 22px;

}

table{

border-collapse:separate;

border-spacing:0px 0px;

font-style: bold;

}

table.table#temprh > tbody > tr > td {

border: 0;

}

#temprh {

padding-top: 10px;

font-size: 12px;

}

.table>tbody>tr>td.pollutantName {

border-top: 0px;

}

.table-condensed>tbody>tr>td {

padding: 0.5px;

}

.table-condensed>tbody>tr>td.pollutantFull {

padding-bottom: 5px;

font-size: 10px;

}

.pollutantValue {

text-align: right;

}

</style>

<?php

$sql="UPDATEgasesSETCO=RAND()\*100,NO=RAND()\*100,NO2=RAND()\*100,O3=RAND()\*100,FineParticle s=RAND()\*100,CourseParticles=RAND()\*100 WHERE id=1"; if ($con->query($sql) === TRUE) { toXML();

}

$sql="UPDATEgasesSETCO=RAND()\*100,NO=RAND()\*100,NO2=RAND()\*100,O3=RAND()\*100,FineParticles=RAND()\*100,CourseParticles=RAND()\*100 WHERE id=2";if ($con->query($sql) === TRUE) {toXML();

}

$sql="UPDATEgasesSETCO=RAND()\*100,NO=RAND()\*100,NO2=RAND()\*100,O3=RAND()\*100,FineParticles=RAND()\*100,CourseParticles=RAND()\*100 WHERE id=3";if ($con->query($sql) === TRUE) {toXML();

}

$sql="UPDATEgasesSETCO=RAND()\*100,NO=RAND()\*100,NO2=RAND()\*100,O3=RAND()\*100,FineParticles=RAND()\*100,CourseParticles=RAND()\*100 WHERE id=4";if ($con->query($sql) === TRUE) {toXML();

}

$sql="UPDATEgasesSETCO=RAND()\*100,NO=RAND()\*100,NO2=RAND()\*100,O3=RAND()\*100,FineParticles=RAND()\*100,CourseParticles=RAND()\*100 WHERE id=5";

if ($con->query($sql) === TRUE) {

toXML();

}

else {

echo "Error updating record: " . $con->error;

}

?>

</head>

<body>

<div class="navbar navbar-inverse navbar-fixed-top" role="navigation">

<div class="container">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

</button>

</div>

<div class="collapse navbar-collapse">

<ul class="nav navbar-nav">

<li><a href="home.html" class="image\_navbar"></a></li>

<li><a href="#"><font size=5" color="yellow"><b> Air Pollution Monitoring System</b></font></a></li>

<li class="active"><a href="#">Home</a></li>

<li><a href="HowItWorks2.html" rel="m\_PageScroll2id">How It Works</a></li>

</ul>

</div>

</div>

</div>

<div id="map"></div>

<div id="legend"><h3>Legend</h3></div>

<script>

var iconBase = 'https://maps.google.com/mapfiles/kml/pushpin/';

var icons={

a:{

icon:iconBase+'red-pushpin.png',

name:'Unhealthy'

},

b:{

icon:iconBase+'ylw-pushpin.png',

name:'Moderate'

},

c:{

icon:iconBase+'grn-pushpin.png',

name:'Good'

}

};

function initMap() {

var map = new google.maps.Map(document.getElementById('map'), {

center: new google.maps.LatLng(23.8144,86.4412),

zoom: 17,

mapTypeId: 'roadmap'

});

var infoWindow = new google.maps.InfoWindow;

document.getElementById('locationheader').innerHTML ='Location';

document.getElementById('cob').innerHTML ='Blank';

document.getElementById('nob').innerHTML ='Blank';

document.getElementById('no2b').innerHTML ='Blank';

document.getElementById('o3b').innerHTML ='Blank';

document.getElementById('Fineb').innerHTML ='Blank';

document.getElementById('Courseb').innerHTML ='Blank';

downloadUrl('myxml.xml?id='+Math.random(), function(data) {

var xml = data.responseXML;

var gases= xml.getElementsByTagName('gas');

var i=0;

Array.prototype.forEach.call(gases, function(gasElem) {

var CO =parseInt(gases[i].getElementsByTagName("CO")[0].childNodes[0].nodeValue);

var NO = parseInt(gases[i].getElementsByTagName("NO")[0].childNodes[0].nodeValue);

var NO2= parseInt(gases[i].getElementsByTagName("NO2")[0].childNodes[0].nodeValue);

var O3= parseInt(gases[i].getElementsByTagName("O3")[0].childNodes[0].nodeValue);

var Fine= parseInt(gases[i].getElementsByTagName("FineParticles")[0].childNodes[0].nodeValue);

var Course= parseInt(gases[i].getElementsByTagName("CourseParticles")[0].childNodes[0].nodeValue);

var type = gases[i].getElementsByTagName("type")[0].childNodes[0].nodeValue;

var point = new google.maps.LatLng(

parseFloat(gases[i].getElementsByTagName("lat")[0].childNodes[0].nodeValue),

parseFloat(gases[i].getElementsByTagName("lng")[0].childNodes[0].nodeValue));

var avg=(CO+NO+NO2+O3+Fine+Course)/6;

console.log(avg);

if(avg>50) console.log("Hi");

i++;

var infowincontent = document.createElement('div');

var strong = document.createElement('strong');

strong.textContent = type

infowincontent.appendChild(strong);

infowincontent.appendChild(document.createElement('br'));

var text = document.createElement('text');

text.textContent = 'CO:'+CO

infowincontent.appendChild(text);

infowincontent.appendChild(document.createElement('br'));

var text = document.createElement('text');

text.textContent ='NO:'+NO

infowincontent.appendChild(text);

infowincontent.appendChild(document.createElement('br'));

var text = document.createElement('text');

text.textContent ='NO2:'+NO2

infowincontent.appendChild(text);

infowincontent.appendChild(document.createElement('br'));

var x;

if(avg>=66) x='a';

else if(avg>=33&&avg<66) x='b';

else x='c';

//var icon = customLabel[type] || {};

var marker = new google.maps.Marker({

map: map,

position: point,

//label: icon.label

icon: icons[x].icon

});

marker.addListener('click', function() {

infoWindow.setContent(infowincontent);

infoWindow.open(map, marker);

document.getElementById('locationheader').innerHTML =type;

document.getElementById('cob').innerHTML =CO;

document.getElementById('nob').innerHTML =NO;

document.getElementById('no2b').innerHTML =NO2;

document.getElementById('o3b').innerHTML =O3;

document.getElementById('Fineb').innerHTML =Fine;

document.getElementById('Courseb').innerHTML =Course;

});

});

});

var legend = document.getElementById('legend');

for (var key in icons) {

var type = icons[key];

var name = type.name;

console.log(name);

var icon = type.icon;

var div = document.createElement('div');

div.innerHTML = '<img src="' + icon + '"> ' + name;

legend.appendChild(div);

}

map.controls[google.maps.ControlPosition.RIGHT\_BOTTOM].push(legend);

}

function openIW(layerEvt) {

if (layerEvt.row) {

var content = layerEvt.row['admin'].value;

} else if (layerEvt.featureData) {

var content = layerEvt.featureData.name;

}

document.getElementById('locationheader').innerHTML =content;

document.getElementById('cob').innerHTML =content;

document.getElementById('nob').innerHTML =content;

document.getElementById('no2b').innerHTML =content;

document.getElementById('o3b').innerHTML =O3;

document.getElementById('Fineb').innerHTML =content;

document.getElementById('Courseb').innerHTML =content;

}

function downloadUrl(url, callback) {

var request = window.ActiveXObject ?

new ActiveXObject('Microsoft.XMLHTTP') :

new XMLHttpRequest;

request.onreadystatechange = function() {

if (request.readyState == 4) {

request.onreadystatechange = doNothing;

callback(request, request.status);

}

};

request.open('GET', url, true);

request.send(null);

}

function doNothing() {}

</script>

<script async defer

src="https://maps.googleapis.com/maps/api/js?key=AIzaSyD6wAa9OYZjUFgisXP6vIez963OflUcq2Q&callback=initMap">

</script>

<!-- MAP SECTION -->

<!--<div id="map"> -->

<div>

<div class="col-md-3" id="valuesTable">

<div class="col-md-3" id="valuesTable" style="padding-top: 24px;">

<div id="locationheader"></div>

<!--<div id="lastupdated">Last Updated</div>-->

<table class="table table-condensed" id="pollutants" style="padding-top: 20px">

<tr>

<td id="coa" class="pollutantName values">CO</td>

<td id="cob" class="alpha1 pollutantName pollutantValue values"></td>

</tr>

<tr>

<td class="pollutantFull">Carbon Monoxide</td>

<td class="unitlabel">ppb</td>

</tr>

<tr>

<td id="noa" class="pollutantName values">NO</td>

<td id="nob" class="alpha2 pollutantName pollutantValue values"></td>

</tr>

<tr>

<td class="pollutantFull">Nitric Oxide</td>

<td class="unitlabel">ppb</td>

</tr>

<tr>

<td id="no2a" class="pollutantName values">NO<sub>2</sub></td>

<td id="no2b" class="alpha3 pollutantName pollutantValue values"></td>

</tr>

<tr>

<td class="pollutantFull">Nitrogen Dioxide</td>

<td class="unitlabel">ppb</td>

</tr>

<tr>

<td id="o3a" class="pollutantName values">O3</td>

<td id="o3b" class="alpha1 pollutantName pollutantValue values"></td>

</tr>

<tr>

<td class="pollutantFull">O3</td>

<td class="unitlabel">ppb</td>

</tr>

<tr>

<td id="Finea" class="pollutantName values">Fine Particles</td>

<td id="Fineb" class="alpha1 pollutantName pollutantValue values"></td>

</tr>

<tr>

<td class="pollutantFull">Fine Particles</td>

<td class="unitlabel">ppb</td>

</tr>

<tr>

<td id="Coursea" class="pollutantName values">Course Particles</td>

<td id="Courseb" class="alpha1 pollutantName pollutantValue values"></td>

</tr>

<tr>

<td class="pollutantFull">Course Particles</td>

<td class="unitlabel">ppb</td>

</tr>

</table>

<div>

<p style="font-size: 15px; margin-top: 30px">Click on a location to display air quality data</p>

<p style="font-size: 12px; margin-bottom: 55px">Note: Values shown are 10 second averages, and therefore may be influenced by small fluctuations and noise</p>

</div>

</div>

<script src="HomeFront.js"></script>

</body>

</html>

**2. db.php**

<?php

$con=mysqli\_connect("localhost","USERNAME","PASSWORD","clarity");

if ($con->connect\_error) {

die("Connection failed: " . $con->connect\_error);

}

?>

**3. new.php**

<?php

//include("C:\xampp\htdocs\phpfiles\index.php");

require("phpsqlajax\_dbinfo.php");

// Start XML file, create parent node

$dom = new DOMDocument("1.0");

$node = $dom->createElement("gases");

$parnode = $dom->appendChild($node);

// Opens a connection to a MySQL server

$connection=mysql\_connect ('localhost', $username, $password);

if (!$connection) { die('Not connected : ' . mysql\_error());}

// Set the active MySQL database

$db\_selected = mysql\_select\_db($database, $connection);

if (!$db\_selected) {

die ('Can\'t use db : ' . mysql\_error());

}

function toXML(){

$data\_txt=null;

$myFile = "myxml.xml";

$fh = fopen($myFile, 'w') or die("can't open file");

$data\_txt .= '<?xml version="1.0" encoding="utf-8"?>';

$data\_txt .= '<gases>';

$query = mysql\_query("SELECT \* FROM gases");

while($values\_query = mysql\_fetch\_assoc($query))

{

$data\_txt .= '<gas>';

$data\_txt .= '<CO>' .$values\_query['CO']. '</CO>';

$data\_txt .= '<NO>' .$values\_query['NO']. '</NO>';

$data\_txt .= '<NO2>' .$values\_query['NO2']. '</NO2>';

$data\_txt .= '<O3>' .$values\_query['O3']. '</O3>';

$data\_txt .= '<FineParticles>' .$values\_query['FineParticles']. '</FineParticles>';

$data\_txt .= '<CourseParticles>' .$values\_query['CourseParticles']. '</CourseParticles>';

$data\_txt .= '<lat>' .$values\_query['lat']. '</lat>';

$data\_txt .= '<lng>' .$values\_query['lng']. '</lng>';

$data\_txt .= '<type>' .$values\_query['type']. '</type>';

$data\_txt .= '</gas>';

}

$data\_txt .= '</gases>';

fwrite($fh, $data\_txt);

fclose($fh);

}

?>

**4. phpsqlajax\_dbinfo.php**

<?php

$username="USERNAME";

$password="PASSWORD";

$database="clarity";

$con=mysqli\_connect("localhost",$username,$password,$database);

?>

**5. gases.sql**

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

CREATE TABLE `gases` (

`id` int(11) NOT NULL,

`CO` int(11) NOT NULL,

`NO` int(11) NOT NULL,

`NO2` int(11) NOT NULL,

`O3` int(11) NOT NULL,

`FineParticles` int(11) NOT NULL,

`CourseParticles` int(11) NOT NULL,

`lat` float(10,6) NOT NULL,

`lng` float(10,6) NOT NULL,

`type` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO `gases` (`id`, `CO`, `NO`, `NO2`, `O3`, `FineParticles`, `CourseParticles`, `lat`, `lng`, `type`) VALUES

(1, 83, 95, 24, 37, 12, 51, 23.815920, 86.439468, 'NLHC'),

(2, 20, 46, 68, 2, 6, 23, 23.814095, 86.440071, 'OAT'),

(3, 97, 19, 1, 47, 34, 27, 23.811882, 86.440819, 'CSEDEPT'),

(4, 33, 86, 31, 95, 85, 38, 23.814287, 86.442741, 'Director'),

(5, 36, 66, 23, 14, 4, 77, 23.814919, 86.442497, 'Library');

ALTER TABLE `gases`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=6;

**6. myxml.xml**

<?xml version="1.0" encoding="utf-8"?><gases><gas><CO>13</CO><NO>61</NO><NO2>66</NO2><O3>47</O3><FineParticles>38</FineParticles><CourseParticles>48</CourseParticles><lat>23.815920</lat><lng>86.439468</lng><type>NLHC</type></gas><gas><CO>28</CO><NO>94</NO><NO2>87</NO2><O3>52</O3><FineParticles>98</FineParticles><CourseParticles>36</CourseParticles><lat>23.814095</lat><lng>86.440071</lng><type>OAT</type></gas><gas><CO>84</CO><NO>12</NO><NO2>8</NO2><O3>3</O3><FineParticles>93</FineParticles><CourseParticles>56</CourseParticles><lat>23.811882</lat><lng>86.440819</lng><type>CSEDEPT</type></gas><gas><CO>0</CO><NO>34</NO><NO2>68</NO2><O3>39</O3><FineParticles>91</FineParticles><CourseParticles>39</CourseParticles><lat>23.814287</lat><lng>86.442741</lng><type>Director</type></gas><gas><CO>22</CO><NO>94</NO><NO2>5</NO2><O3>41</O3><FineParticles>90</FineParticles><CourseParticles>27</CourseParticles><lat>23.814919</lat><lng>86.442497</lng><type>Library</type></gas></gases>

**5. CONCLUSION**

The proposed Wireless Air Pollution Monitoring System

provides the real-time information about the level of pollution in air,

as well as provides alerts in cases of drastic change in quality of air.

This information can then be used by authorities to take prompt

actions such as evacuating people and sending emergency response

team . It uses an air quality index to categorize the various levels of

air pollution .

**6. Other ways of air pollution management**

* Use craft supplies in well-ventilated areas.
* Make sure your gas stove is well-ventilated.
* Minimize clutter.
* Remove carpeting if possible.
* Use a dehumidifier and/or air conditioner to reduce moisture
* Avoid smoking indoors.

**7. References**

* Robin Nixon ;Learning PHP ,MySQL and Javascript
* W3schools.com
* Codeacademy.com etc..