**SMART WATER IRRIGATION SYSTEM USING IOT**

**Project overview**

The Real time smart water irrigation Platform is a web-based application designed to display simulated real time transit data for water monitoring of a plant. This documentation provides an overview of the project, its objectives, and the technologies used.

**Objectives:**

**The main objectives of the project are as follows**

Simulate real-time transit data updates for water monitoring of a plant.

Demonstrate the use of web development technologies (HTML, CSS, JavaScript) to create a dynamic platform.

**HTML STRUCTURE (INDEX.HTML):**

The index.html file serves as the entry point for the application.

**Code Explanation**

<!DOCTYPE html>

<html>

<head>

<title>Smart Water Consumption Dashboard</title>

<link rel="stylesheet" type="text/css" href="styles.css">

</head>

<body>

<header>

<h1>Water Consumption Dashboard</h1>

</header>

<section class="data">

<div id="consumption-data">

<!-- Real-time water consumption data will be displayed here -->

</div>

</section>

<section class="conservation-efforts">

<h2>Water Conservation Efforts</h2>

<p>Information and tips on how to conserve water can be displayed here.</p>

</section>

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

</body>

</html>

**CSS STYLING LOGIC (STYLES.CSS):**

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

text-align: center;

}

header {

background-color: #0077b6;

color: #fff;

padding: 20px;

}

h1 {

font-size: 2rem;

}

.data {

background-color: #fff;

padding: 20px;

margin: 20px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);

}

.conservation-efforts {

background-color: #fff;

padding: 20px;

margin: 20px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);

}

**JAVA SCRIPT LOGIC (SCRIPT.JS):**

The script.js file contains JavaScript code that simulates real-time data updates.

**Code Explanation:**

// Simulated data for demonstration purposes

function getRealTimeData() {

// Replace this with actual data retrieval from IoT sensors

return Math.random() \* 100; // Random value for demonstration

}

function updateConsumptionData() {

const consumptionData = document.getElementById("consumption-data");

setInterval(() => {

const waterConsumption = getRealTimeData();

consumptionData.innerHTML = `Current Water Consumption: ${waterConsumption.toFixed(2)} gallons`;

}, 10000); // Update every 10 seconds (adjust as needed)

}

updateConsumptionData();

**PROJECT ARCHITECTURE**

**High-Level Overview**

The project consists of a simple webpage with HTML, CSS, and JavaScript for the frontend. It simulates real-time data updates without the use of actual IoT sensors. The data is generated and displayed on the page.

**Data Flow**

Data is generated and updated in real time using JavaScript. The frontend receives data updates and displays them in the designated section.

**REAL-TIME DATA SIMULATION**

**Data Source**

The project simulates data for a water monitoring of a plant. Data sources are not real IoT sensors but generated within the JavaScript code.

**Real-Time Updates**

The JavaScript code simulates real-time updates by refreshing data every 10 seconds

**USER INTERFACE:**

The user interface includes a header and a data display section, styled using CSS.

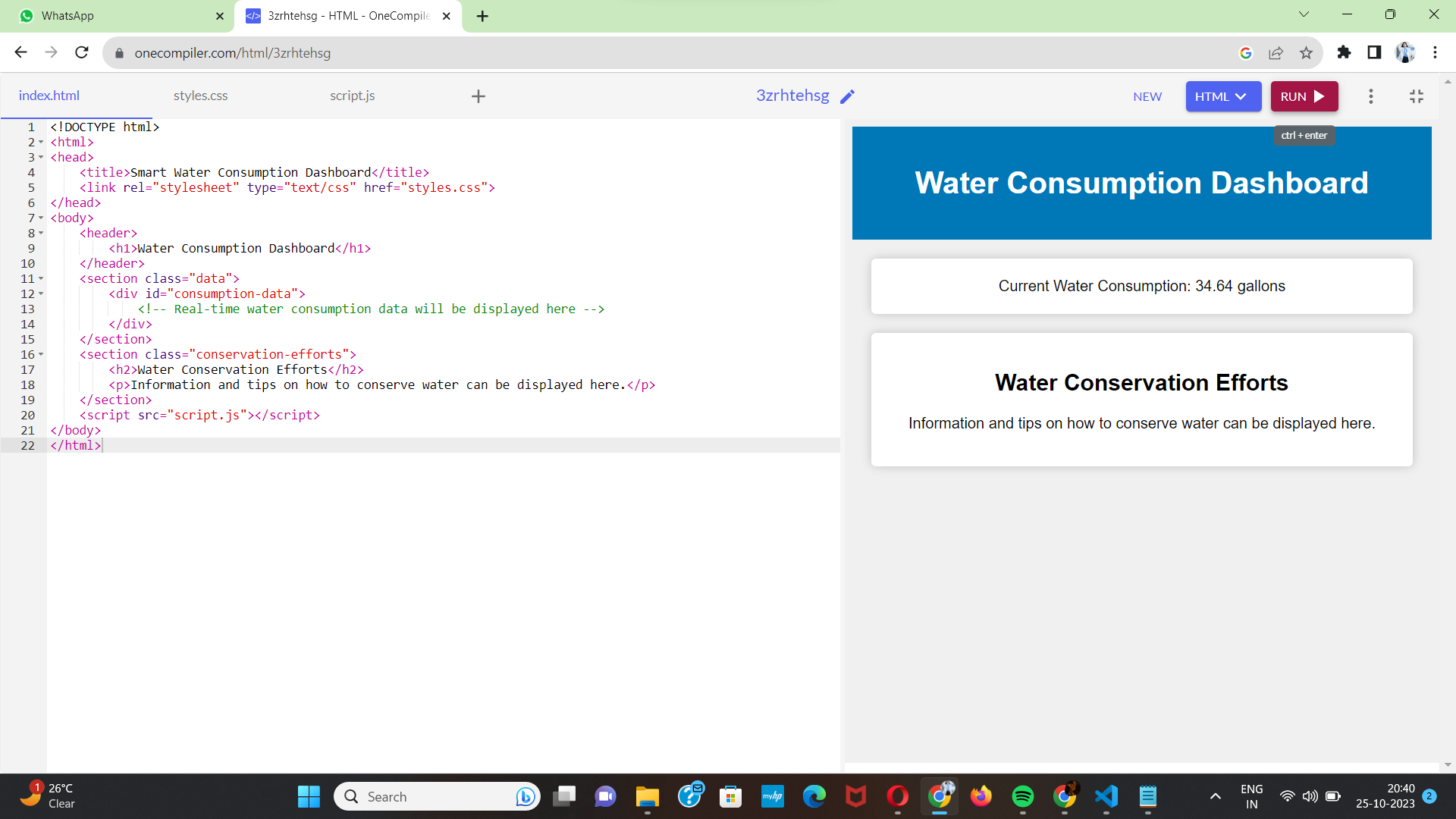
**TESTING:**

The project can be tested by opening the index.html file in a web browser. Real-time data updates are simulated, and you can observe the changes

**DEPLOYMENT:**

The project is deployed as a static website, and it can be hosted on any web server or platform capable of serving HTML, CSS, and JavaScript files.

**OUTPUT:**



**CONCLUSION:**

The Real-Time water monitoring of a plant is a simple example of using web development technologies to simulate and display real-time data updates. It serves as a basic educational resource for understanding how real-time data can be presented in a web application.