# Smart water fountain Documentation & Submission

**The objective of a smart water fountain could be to provide an efficient and convenient way for people and pets to access clean and filtered water while integrating technology to monitor usage, track water quality, and encourage hydration. This could involve features like automatic refilling, sensors to detect when the fountain needs cleaning, and the ability to connect to a mobile**

**app for control and data analysis.**

**Internet Of Things:

- Internet of Things (IOT) refers to the network of physical objects or “things” embedded with sensors, software, and connectivity that allows them to collect and exchange data over the internet. These objects can be everyday devices like smart thermostats, wearable fitness trackers, or industrial equipment like sensors in manufacturing plants.

**Smart Water Fountain features:

- Smart Water Fountains in IOT reference
- Implementing smart water fountains using IOT (Internet of Things) technology involves connecting fountains to the internet to monitor and control their functions remotely. Here's a basic reference

architecture for a smart water fountain system

**Components:

- The physical water fountain unit that provides water, typically with a pump and nozzle.

**Sensors:

- Various sensors can be incorporated, such as water level sensors, temperature sensors, and flow sensors to monitor fountain conditions.

**Micro Control/Embedded Systems:

- A device like Arduino, Raspberry Pi, or specialized IOT microcontrollers to control the fountain's operation based on sensor inputs.

**Connectivity Module:

- A Wi-Fi, Bluetooth, or cellular module to enable communication between the fountain and the IOT platform.
- A cloud-based platform like AWS IOT, Azure Cloud IOT Core to manage device data, authentication, and remote control.

**Mobile/Web Application:

- A user interface (app or web) to allow users to monitor and control the fountain remotely.

**Data Collection:

- Sensors monitor fountain parameters (water level, temperature, etc.) and send data to the microcontroller.

**Data Processing:

- The microcontroller processes sensor data and decides if any action is needed (e.g., refilling water).

**Communication:

- The microcontroller sends data to the IOT platform using the connectivity module, which is then securely transmitted to the cloud.

**Cloud processing:

- The IOT platform stores and processes the data, making it accessible for remote monitoring and control.

**User Interaction:

- Users access the mobile or web app to check the fountain's status and control its functions (e.g., turn on/off, adjust water flow).

**Smart Water Fountain Innovation:

- development in the realm of hydration and sustainability. They often incorporate technology to provide several benefits

**Filtered Water:

- Smart fountains can filter tap water, ensuring it’s clean and safe to drink, reducing the need for bottled water.

**Hydration Tracking:

- Some models offer features to track your water consumption, helping individuals stay hydrated.

**Touchless Operation:

- Many smart fountains have touchless sensors, promoting hygiene by reducing contact with surfaces.

**Refill Alerts:

- They can send notifications when it’s time to refill the water reservoir.

**Customization:

- Users can often customize water temperature and carbonation levels to their preference.

- These innovations align with sustainability and health goals while embracing technological advancements for improved water access and conservation.

Python is a high-level, versatile programming language known for its simplicity and readability. Created by Guido van Rossum and first released in 1991, Python has gained widespread popularity for its clean and easy-to-understand syntax. It supports both procedural and object-oriented programming, making it suitable for a wide range of applications.

**Python features:

- Some key features of Python include dynamic typing, automatic memory management (garbage collection), and a vast standard library that provides modules and functions for various tasks. Python is used in web development, data

analysis, scientific computing, artificial intelligence, automation, and more.

- You can write Python code in a variety of development environments, and it's widely used for both beginners and experienced programmers. Python's readability and extensive community support make it an excellent choice for anyone looking to start their programming journey or tackle complex projects

**Model Training:

*/

//define the input/output Int refill pump

= 6; // sets pin 7 as the output pin to the

Refill pump

Int Submersible pump = 8; //de clear pin 8 as

Submersible pump pin

Const Int Relay = 4;

Const int analog Pin = A0; // pin that the soil

Moisture

Sensor is attached to

Const int threshold = 350; // an arbitrary

Threshold

Level that tell the amount of water in the soil

Is enough

#define float switch 3 //de clear pin 3 as float

Switch pin

// int the DS3231 using the hardware

Interface

#include <DS3231.h>

DS3231 rtc(SDA, SCL);

Time t;

Void setup(),{ Serial. Begin(9600); //set up

Serial monitor Rtc. Begin(); // Initialize the rtc

Object

//define the input/output pins pin

Mode(Relay, OUTPUT); // set pin 4 relay pin
As output
Digital Write(Relay, LOW); pin Mode(Refill
Pump , OUTPUT); pin Mode( Submersible
Pump, OUTPUT);
Pin Mode(float switch, INPUT_PULLUP); //sets
Float switch pin which is pin 2 as input}
Void loop () { // put your main code here, to
Run
Repeatedly:
T = rtc get Time() Serial print(t. hour); Serial.
Print(“ hour(s), “); Serial. Print(t. min);
Serial .print(“ minute(s)”); serial. Print ln(“ “);
Delay (1000);
If(digital Read(float switch)== HIGH){ digital
Write( Submersible pump, LOW); //T turn off
Submersible pump digital Write(Refill pump,

```
LOW); //turn on the pump}
Else{ digital Write(Refill pump, HIGH);//turn
On the pump digital Write(submersible
Pump, HIGH ); //Turn off
Submersible pump }
Delay(1);
Int moisture level = analog Read(analog Pin);
//declare
Analogue in value as the moisture level Serial.
Print ln(moisture level);
If( moisture level > threshold) {
// t. hour is the hour to turn On and t. min is
The minutes
To turn on
// the following lines will turn on at 7:00am
To 7:15am
If (t. hour ==7 && t. min ==0) {digital
```

Write(Relay ,HIGH);

Serial print ln (“pump ON”); }

Else if (t. hour ==7 && t. min ==1) {digital Write(Relay, HIGH); Serial. Print ln(“pump ON”); }

**Evaluation:

Else if (t. hour ==7 && t. min ==2) {digital Write(Relay, HIGH); serial. Print ln(“pump ON”); }

Else if (t. hour ==7 && t. min ==3) {digital Write(Relay, HIGH); Serial. Print ln(“pump ON”); }

Else if (t. hour ==7 && t. min ==4) {digital Write(Relay, HIGH); Serial. Print ln(“pump ON”);

Else if (t. hour ==7 && t. min ==5) {digital Write(Relay, HIGH); Serial. Print ln(“pump

```
ON”); }
Else if (t. hour ==7 && t. min ==6) {digital
Write(Relay, HIGH);
Serial. Print ln(“pump ON”); }
Else if (t. hour ==7 && t. min ==8) {digital
Write( Relay ,HIGH); Serial. Print ln(“pump
ON”); }
Else if (t. hour ==7 && t. min ==9) {digital
Write(Relay, HIGH); Serial. Print ln(“pump
ON”); }
Else if (t. hour ==7 && t. min ==10) {digital
Write(Relay, HIGH); Serial. Print ln(“pump
ON”); }
Else if (t. hour ==7 && t. min ==11)
{digital Write(Relay, HIGH); Serial. Print
Ln(“pump ON”); }
Else if (t. hour ==7 && t. min ==12) {digital
```

Write(Relay, HIGH); Serial. Print ln(“pump ON”); }

Else if (t. hour ==7 && t. min ==13) {digital Write(Relay ,HIGH); Serial. Print ln(“pump ON”); }

**Build a Use smart home Automation:

Energy-efficient Smart Home Automation

**Scenario:

- John and Jane are a couple who want to make their home more energy-efficient and convenient. They have a busy lifestyle and often forget to turn off lights, adjust the thermostat, or lock doors when leaving home. They decide to implement a smart home automation system to address these issues.

**Smart Home Automation Components:

- Smart Thermostat: They install a smart thermostat that can be controlled remotely

through a mobile app. The thermostat learns their preferences and adjusts the temperature based on their schedule, optimizing energy consumption.

**Smart Lighting:

- They replace traditional bulbs with smart bulbs that can be controlled individually or as a group. Motion sensors in different rooms automatically turn off lights when no one is present and turn them on when someone enters.

**Smart Lock:

- They install a smart door lock with keyless entry. The lock can be controlled remotely and sends notifications when the door is locked or unlocked. It can also be set to lock automatically when they leave the house.

**Security Cameras:

- Smart security cameras are placed around the house. These cameras can be accessed through their smartphones, allowing them to check on their home's security while they're away.

**Voice Assistant Integration:

- They connect their smart home system to a voice assistant like Amazon Alexa or Google Assistant for voice control of various devices. For instance, they can say, "Alexa, turn off all lights

**Workflow:

- When John and Jane leave for work in the morning, the smart lock automatically locks the door behind them. The smart thermostat adjusts the temperature to an energy-saving setting.

During the day, motion sensors and occupancy detectors in each room control the smart

lighting, turning lights off when no one is in the room.

If they forget to lock the door or turn off lights, they can use their smartphones to control these devices remotely.

When they return home, the smart thermostat has already adjusted the temperature to their comfort level.

In case of any unusual activity, their security cameras send alerts to their phones for immediate action.

**Benefits:

Energy savings through automated temperature control and lighting.

Enhanced security with remote door locking and surveillance.

Convenience with voice control and remote access to devices.

Reduced environmental impact through efficient energy use.

This smart home automation use case not only makes their lives more convenient but also contributes to a more energy-efficient and secure living environment.

## **Introduction to computer vision with python:

- Computer vision is a field of artificial intelligence that focuses on enabling computers to interpret and understand visual information from the world, much like the human visual system. Python is a popular programming language for developing computer vision applications, thanks to its rich ecosystem of libraries and tools. Here's a brief introduction to computer vision in Python.

## **Install Required Libraries:

- Start by installing key Python libraries for computer vision, including OpenCV (Open Source Computer Vision Library) and NumPy. You can use pip for installation.

Python

Copy code

Pip install opencv-python numpy

Image and Video Processing: You can use OpenCV to load, display, and manipulate images and video streams. For example, you can read an image from a file:

Python

Copy code

```
Import cv2
Image = cv2.imread('image.jpg')
Cv2.imshow('Image', image)
Cv2.waitKey(0)
Cv2.destroyAllWindows()
```

**Basic Operation:

- You can perform operations like image resizing, cropping, and applying filters using Open cv functions.

**Objective Detection:

- includes pre-trained models for object detection, like Haar cascades or deep learning-based models. These can be used to detect faces, objects, and more.

**Image. Processing:

- You can apply various image processing techniques, such as filtering, thresholding, edge detection, and contour analysis to extract meaningful information from images.

- You can combine computer vision with machine learning for tasks like image

classification, object recognition, and image segmentation.

Real-time Video Processing: OpenCV allows you to process video streams from cameras, making it suitable for applications like video surveillance or gesture recognition.

**Deep Learning:

- Python libraries like Tensor Flow and PyTorch are often used for deep learning-based computer vision tasks, including image classification, object detection, and semantic segmentation.

**Community and Resources:

- Python has a large and active computer vision community, so you can find extensive documentation, tutorials, and code examples online.

To get started with computer vision in Python, consider following tutorials and documentation for OpenCV and exploring various computer vision projects to apply these concepts in practical applications.