Register No: 421221106307

Team Mentor: T. Karthika

**SMART WATER FOUNTAIN USING IOT**

**Design Thinking:**

**Identify the Problem:**

* *Define the problem you aim to solve or the purpose of the smart water fountain. Is it for a public park, office, or home use? What specific issues should it address, such as water conservation or convenience?*

**User Research:**

* *Understand the needs and preferences of potential users. Conduct surveys or interviews to gather insights about how they would like to interact with the fountain Ideation:*
* *Brainstorm and generate ideas for the smart features of the water fountain. Consider things like automatic refill, water quality monitoring, touchless operation, and mobile app integration*.

**Concept Development:**

* *Create a concept that outlines the key features and functionalities of the smart water fountain. This should include a user-friendly interface and a mobile app for control and monitoring.*

**Technology Selection:**

* *Choose IoT components such as sensors, microcontrollers, connectivity options (Wi-Fi, Bluetooth, LoRa, etc.), and a cloud platform to support data storage and analysis.Prototyping:*
* *Build a prototype of the smart water fountain to test the design and functionality. This will help you identify any issues and make improvements*.

**User Testing:**

* *Gather feedback from users by allowing them to interact with the prototype. Use this feedback to refine the design.*

**Development:**

* *Develop the final version of the smart water fountain, including the hardware and software components.*

**Integration:**

* *Integrate sensors to monitor water levels and quality, as well as controls for the fountain's operation. Ensure connectivity to the IoT platform.Mobile App Development:*
* *Create a mobile app that allows users to check water quality, remotely control the fountain, and receive alerts or notifications.*

**Security:**

* *Implement robust security measures to protect user data and the fountain's operation from potential threats.*

**Sustainability:**

* *Consider sustainability aspects, like energy efficiency and materials used in the fountain's construction.*

**Testing and Quality Assurance:**

* *Thoroughly test the smart water fountain to ensure it functions reliably and safely.*

**Deployment:**

* *Install the smart water fountain in its intended location, and provide user training if needed.Maintenance and Updates:*
* *Plan for regular maintenance, software updates, and support for the smart water fountain to keep it running smoothly.*

**Data Analysis:**

* *Collect and analyze data from the fountain's sensors to gain insights and improve its efficiency and user experience*.

**Feedback Loop:**

* *Continuously gather user feedback and iterate on the design to address any issues or add new features.*

**Components:**

* *Water Fountain Hardware:*
* *Water pump and reservoir*
* *Water quality sensors*
* *Water level sensors*
* *Microcontroller (e.g., Arduino, Raspberry Pi)*
* *Connectivity module (Wi-Fi, Bluetooth, LoRa)*
* *Touchless operation sensors (e.g., motion or proximity sensors)*

***Mobile App (if applic*able):**

* *User interface for remote control and monitoring*
* *Notifications and alerts*
* *Data visualization*

**Cloud Platform:**

* Data storage and analysis
* Remote access and control
* Security and authentication

**Power Supply:**

Power source for the fountain and IoT components (battery or electrical)

**Housing and Enclosure:**

Protecting the electronics from environmental factors

User Interface (if not using a mobile app):

**Buttons or touch panels for manual controlSecurity Measures**:

*Encryption and authentication protocols to secure data and operations.*

**Maintenance and Support System:**

*Remote diagnostics and updates for ongoing maintenance*

**Process to built**

**Gather Requirements:**

* *Identify the specific requirements for your smart water fountain. Determine the purpose, location, user needs, and desired features.*

**Design the System:**

* *Create a detailed system design, including the hardware and software components, connectivity, and power source.*

**Acquire Components:**

* *Procure the necessary components, such as water pumps, sensors, microcontrollers, connectivity modules, and a power supply.Assemble the Hardware:*
* *Build the physical structure of the water fountain, integrating the water pump, sensors, and microcontroller. Ensure proper placement of sensors for water quality and level monitoring.*

**Connectivity Setup:**

* *Configure the connectivity module (e.g., Wi-Fi or Bluetooth) to establish a connection with your chosen IoT platform.*

**Develop Firmware:**

* *Write the firmware for the microcontroller to control the water pump, collect data from sensors, and communicate with the IoT platform.Create a Mobile App (Optional):*
* *If your design includes a mobile app, develop it to provide remote control and monitoring capabilities. Ensure it's user-friendly and secure.*

**Set Up IoT Platform:**

* *Choose an IoT platform (e.g., AWS IoT, Azure IoT, or a custom solution) and set up the necessary data storage and communication infrastructure.*

**Implement Security Measures:**

* *Implement encryption and authentication protocols to secure data and communication between the fountain and the IoT platform.Testing and Calibration:*
* *Test the smart water fountain for functionality, including water quality monitoring, automatic refill, and remote control. Calibrate sensors as needed.*

**Deployment:**

* *Install the smart water fountain in its intended location, ensuring it has a stable power source and connectivity.*

**User Training:**

* *If necessary, provide user training on how to use the smart water fountain and the accompanying mobile app.*

**Maintenance and Updates:**

* *Plan for ongoing maintenance, including software updates and system checks to ensure the fountain operates correctlyData Analysis and Continuous Improvement:*
* *Collect and analyze data from the fountain's sensors to gain insights and make improvements to the system.*

**Feedback Loop:**

* *Continuously gather user feedback and iterate on the design to address any issues or add new features to enhance the user experience.*

**Steps for coding :**

*Set up your development environment with a C compiler.*

*Define your system requirements and components.*

*Write C code to control the water pump, collect sensor data, and enable communication with IoT components.*

*Ensure security measures for data protection.*

*Test the code and hardware.*

*Integrate with your chosen IoT platform.*

*Optionally, create a user interface.*

*Document your code.*

*Deploy, provide user training, and plan for maintenance and updates.*

*Analyze data for continuous improvement.*

***Program C coding:***

*#include <stdio.h>*

*#include <stdbool.h>*

*// Function to control the water pump*

*void controlWaterPump(bool pumpOn) {*

*if (pumpOn) {*

*// Code to turn the water pump on*

*printf("Water pump is turned on\n");*

*} else {*

*// Code to turn the water pump off*

*printf("Water pump is turned off\n");*

*}*

*}*

*int main() {*

*bool isPumpOn = false;*

*// Simulate a user's request to turn the pump on*

*isPumpOn = true;*

*controlWaterPump(isPumpOn);*

*// Simulate a user's request to turn the pump off*

*isPumpOn = false;*

*controlWaterPump(isPumpOn);*

*return 0;*

*}*