SMART WATER FOUNTAIN

IOT_PHASE4

Mobile App Development

To connect your IoT smart Water fountain with a mobile app, need to create APIs that allow the mobile app to interact with the backend system. Here's a step-by-step guide on how to achieve this:

1. Develop Backend APIs:

- Create a set of API endpoints on your server to handle various functionalities of the Smart Parking System, such as user authentication, parking spot availability, reservations, and payments. You can use a web framework like Express.js (Node.js) or Django (Python) to develop these APIs.

2. User Authentication:

- Allow users to register and log in to the mobile app.
- Create API endpoints for user registration and login.
- Implement token-based authentication for secure access to the app.

3. Parking Spot Availability:

- Develop an API endpoint to provide real-time information about parking spot availability.
- The mobile app can query this endpoint to display available parking spots to users

4. Reservations:

- Create APIs for reserving parking spots. When a user selects a spot and reserves it, the mobile app should send a request to the reservation API.
- Implement logic to check spot availability and confirm the reservation.
- Return a response to the mobile app with the reservation status.

5. Payment Integration:

- Integrate payment gateway APIs, such as Stripe or PayPal, for processing payments.
- Create API endpoints for initiating and verifying payments. The mobile app can call these endpoints to handle payments.

6. Real-Time Updates:

- Implement WebSocket communication to provide real-time updates on parking spot availability and reservation confirmation. When a parking spot becomes available or a reservation is confirmed, use WebSockets to push updates to the mobile app.

7. Mobile App Development:

- Develop the mobile app using a cross-platform framework like React Native or Flutter to ensure compatibility with both Android and iOS
- Implement user interfaces for registration, login, parking spot selection,reservations, and payment processing

8. API Integration:

- Use HTTP requests (e.g., GET, POST, PUT, DELETE) in the mobile app to communicate with the backend APIs.
- Handle API responses in the app to update the user interface and provide feedback to the user.

9. User Notifications:

- Implement push notifications to notify users of reservation confirmations, payment status, and other important updates.
- Utilize Firebase Cloud Messaging (FCM) for Android and Apple Push Notification Service (APNs) for iOS.

10. Testing and Debugging:

- Test the mobile app's functionality by creating test scenarios and debugging any issues that arise.
- Verify that the app can interact seamlessly with the backend APIs.

11. Deployment:

- Deploy the mobile app to app stores (Google Play Store and Apple App Store) for public use.

12. User Support and Updates:

- Provide ongoing support and maintenance for the mobile app.
- Implement updates as needed, addressing user feedback and making improvements.

STEPS

Smart water fountains, also known as intelligent or connected water fountains, are advanced drinking fountains that incorporate technology to enhance functionality, convenience, and user experience. These fountains are often found in public places, educational institutions, workplaces, or even homes. Here are some key features and components commonly associated with smart water fountains:

Touchless Operation: Many smart water fountains are equipped with touchless sensors that allow users to activate the fountain without physically touching any buttons. This feature is especially important for promoting hygiene in public spaces.

Water Quality Monitoring: Some smart fountains have sensors that continuously monitor water quality, checking for contaminants, pH levels, and temperature. Users can be alerted if water quality falls below safe standards.

Filtration System: Smart water fountains may include built-in water filtration systems, ensuring that the water is clean and safe to drink. The status of the filter is often displayed on the fountain or through a connected app.

Adjustable Settings: Users can often adjust the water temperature, flow rate, and even carbonation level, if applicable. This customization allows for a more personalized drinking experience.

Hydration Tracking: Smart fountains can track a user's water consumption and send reminders or notifications to help maintain hydration goals. This data can also be accessed through a mobile app.

Mobile App Integration: Many smart water fountains can be controlled and monitored through a dedicated mobile app. Users can adjust settings, receive notifications, and access usage statistics remotely.

Data Analytics: Data from smart fountains can be collected and analyzed to monitor usage patterns, diagnose potential issues, and optimize water fountain placement.

Voice Activation: Some smart fountains are compatible with voice assistants like Amazon Alexa or Google Assistant, allowing users to control the fountain using voice commands.

Energy Efficiency: These fountains are designed to be energy-efficient, with features such as automatic shut-off after a period of inactivity.

Maintenance Alerts: Users and maintenance staff can receive alerts when filters need replacement, or when the fountain requires maintenance or repairs.

Water Bottle Filling: Many smart fountains include a water bottle filling station, making it easy for users to refill reusable water bottles. This promotes sustainability and reduces plastic waste.

QR Code and NFC Scanning: In some cases, users can activate the fountain by scanning a QR code or using NFC technology, which is especially useful in public spaces.

UV-C Sterilization: Advanced models may include UV-C sterilization to kill bacteria and pathogens in the water, ensuring it's safe to drink.

Commercial and Residential Models: Smart water fountains are available for both commercial and residential use, with some models designed for homes or smaller office settings.

Water Dispensing Metrics: Smart fountains can provide users with data on the number of bottles saved by using the fountain instead of disposable water bottles.

Smart water fountains offer a combination of convenience, sustainability, and hygiene. They also have the potential to reduce the environmental impact of bottled water consumption, as users can access clean and filtered water directly from these fountains.

Python script for this Project:

```
import time  # Import the time module for time delays


# Define GPIO pin numbers
TRIG_PIN = 2  # GPIO pin number for the ultrasonic sensor's trigger
ECHO_PIN = 3  # GPIO pin number for the ultrasonic sensor's echo
PUMP_PIN = 4  # GPIO pin number for the water pump
LED_PIN = 5   # GPIO pin number for the LED


# Initialize components (virtual components for Wokwi)
ultrasonic_sensor = Ultrasonic(TRIG_PIN, ECHO_PIN)  # Create an ultrasonic sensor
pump = Motor(PUMP_PIN)  # Create a water pump
led = LED(LED_PIN)  # Create an LED


while True:
    # Measure distance
    distance = ultrasonic_sensor.distance_cm  # Measure distance in centimeters


    if distance > 200:  # Water level is above 200 cm
        # Make the LED blink
        led.blink(on_time=0.5, off_time=0.5)  # LED blinks with 0.5 seconds on and off time
        pump.on()  # Water pump is turned on
    else:
        # Water level is below 200 cm
        # Turn off the LED and the pump
        led.off()
        pump.off()
    # Introduce a small delay to control the loop rate
    time.sleep(0.1)  # Sleep for 0.1 seconds
```