Code

#include <SoftwareSerial.h>         // including the library for the software serial  
#define DEBUG true  
SoftwareSerial esp8266(10,11);      /\* This will make the pin 10 of arduino as RX pin and  
pin 11 of arduino as the TX pin Which means that you have to connect the TX from the esp8266  
to the pin 10 of arduino and the Rx from the esp to the pin 11 of the arduino\*/  
                                     
const int trigPin = 8;            // Making the arduino's pin 8 as the trig pin of ultrasonic sensor  
const int echoPin = 9;            // Making the arduino's pin 9 as the echo pin of the ultrasonic sensor  
// defining two variable for measuring the distance  
long duration;  
int distance;

void setup()  
{  
  Serial.begin(9600);         // Setting the baudrate at 9600  
  esp8266.begin(9600);        // Set the baudrate according to you esp's baudrate. your esp's baudrate might be different from mine  
  pinMode(trigPin, OUTPUT);   // Setting the trigPin as Output pin  
  pinMode(echoPin, INPUT);    // Setting the echoPin as Input pin  
    
  sendData("AT+RST\r\n",2000,DEBUG);            // command to reset the module  
  sendData("AT+CWMODE=2\r\n",1000,DEBUG);       // This will configure the mode as access point  
  sendData("AT+CIFSR\r\n",1000,DEBUG);          // This command will get the ip address  
  sendData("AT+CIPMUX=1\r\n",1000,DEBUG);       // This will configure the esp for multiple connections  
  sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG); // This command will turn on the server on port 80  
}

void loop()  
{  
digitalWrite(trigPin, LOW);   // Making the trigpin as low  
delayMicroseconds(2);         // delay of 2us  
digitalWrite(trigPin, HIGH); // making the trigpin high for 10us to send the signal   
delayMicroseconds(10);  
digitalWrite(trigPin, LOW);     
duration = pulseIn(echoPin, HIGH);  // reading the echopin which will tell us that how much time the signal takes to come back

distance= duration\*0.034/2;         // Calculating the distance and storing in the distance variable  
    
  if(esp8266.available())         // This command will that check if the esp is sending a message   
  {      
    if(esp8266.find("+IPD,"))  
    {  
     delay(1000);  
     int connectionId = esp8266.read()-48; /\* We are subtracting 48 from the output because the read() function returns   
                                            the ASCII decimal value and the first decimal number which is 0 starts at 48\*/  
     String webpage = "<h1>IOT Garbage Monitoring System</h1>";  
       webpage += "<p><h2>";     
       if (distance<5)  
       {  
        webpage+= " Trash can is Full";  
        }  
        else{  
          webpage+= " Trash can is Empty";  
          }  
       webpage += "</h2></p></body>";    
     String cipSend = "AT+CIPSEND=";  
     cipSend += connectionId;  
     cipSend += ",";  
     cipSend +=webpage.length();  
     cipSend +="\r\n";

     sendData(cipSend,1000,DEBUG);  
     sendData(webpage,1000,DEBUG);      
     String closeCommand = "AT+CIPCLOSE=";   
     closeCommand+=connectionId;   
     closeCommand+="\r\n";  
     sendData(closeCommand,3000,DEBUG);  
    }  
  }  
}  
   
String sendData(String command, const int timeout, boolean debug)  
{  
    String response = "";     
    esp8266.print(command);   
    long int time = millis();  
    while( (time+timeout) > millis())  
    {  
      while(esp8266.available())  
      {  
        char c = esp8266.read();   
        response+=c;  
      }    
    }  
    if(debug)  
    {  
      Serial.print(response);  
    }  
    return response;  
}