CHAPTER:16 ANDRIOD APPLICATION

PRACTICAL: 16A

AIM: Andriod based LED ON OFF using ESP01.

ARDUINO CODE :

```
/**********************
 * Author: Shreejicharan
 * Title:  Andriod based LED ON OFF using ESP01.
 * Date: 27/05/2017
 * Time: 6:00
 * Email: shreejicharanelectronics@gmail.com
 **********************/

/*Develop offline Webserver to control GPIO: Demonstrate offline webserver using HTML
webpage which can be accessed from web browser and through which LED can be toggled
 *
 *    The server will set a GPIO pin depending on the request
 *    http://server_ip/gpio/0 will set the GPIO2 low,
 *    http://server_ip/gpio/1 will set the GPIO2 high
 *    server_ip is the IP address of the ESP8266 module, will be
 *    printed to Serial when the module is connected.
 *    access point AP = router
 */

#include <ESP8266WiFi.h>

#define LED 2

const char* ssid = "keyur234";
const char* password = "12345678";

// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);
  pinMode(LED, OUTPUT);
```

```
  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.println(WiFi.localIP());
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }

  // Read the first line of the request
  String req = client.readStringUntil('\r');
  Serial.println(req);
  client.flush();

  // Match the request
  int val;
```

```
  if (req.indexOf("?pin=ON") != -1)
    val = 0;
  else if (req.indexOf("?pin=OFF") != -1)
    val = 1;
  else {
    Serial.println("invalid request");
    client.stop();
    return;
  }

  // Set GPIO2 according to the request
  digitalWrite(LED, val);

  client.flush();

  // Prepare the response
  String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE
HTML>\r\n<html>\r\nGPIO is now ";
  s += (val)?"high":"low";
  s += "</html>\n";

  // Send the response to the client
  client.print(s);
  delay(1);
  Serial.println("Client disonnected");

  // The client will actually be disconnected
  // when the function returns and 'client' object is detroyed
}
```

SIMULATION:

ANDRIOD APPLICATION BASED LED ON/OFF USING ESP-01