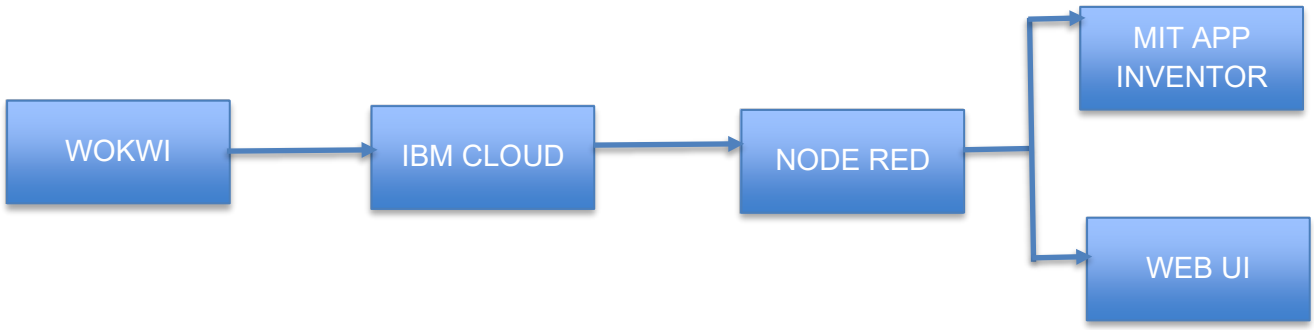
Project Development Phase
Sprint - 1

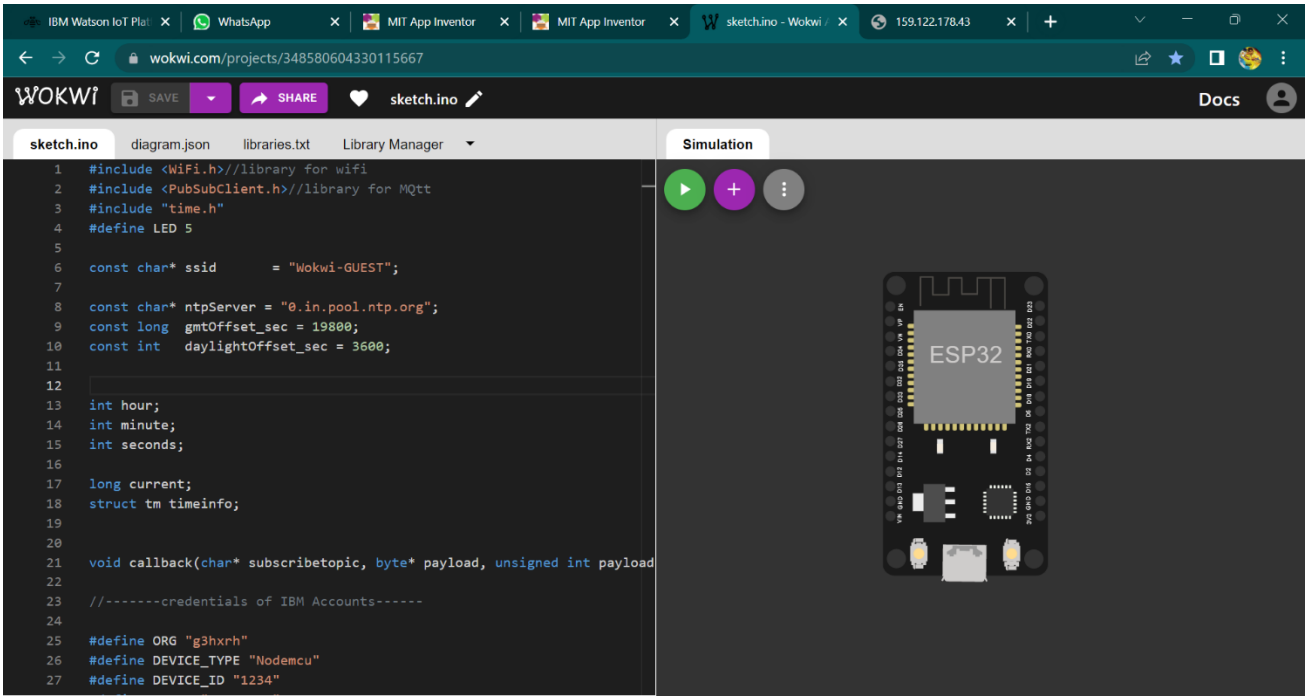
Date	29 October 2022
Team ID	PNT2022TMID35939
Project Name	Medicine Reminder
Maximum Marks	4 Marks

WORKFLOW:



Sprint 1 – We have included the wokwi code and wokwi simulation

Wokwi simulation:



Project code

```
 #include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "time.h"
#define LED 5

const char* ssid       = "Wokwi-GUEST";

const char* ntpServer = "0.in.pool.ntp.org";
const long  gmtOffset_sec = 19800;
```

```
const int   daylightOffset_sec = 3600;
int hour;
int minute;
int seconds;
long current;
struct tm timeinfo;

void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);

//-------credentials of IBM Accounts------
#define ORG "g3hxrh"
#define DEVICE_TYPE "Nodemcu"
#define DEVICE_ID "1234"
#define TOKEN "87654321"
String data3;
float h, t;

//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server
Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of
event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd
REPRESENT command type AND COMMAND IS TEST OF FORMAT
STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
```

```
//---------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefined client id by passing parameter like server id,portand
wificredential

void printLocalTime()
{

  if(!getLocalTime(&timeinfo)){
    Serial.println("Failed to obtain time");
    return;
  }

}

void setup()// configureing the ESP32
{ Serial.begin(115200);

  //connect to WiFi
  Serial.printf("Connecting to %s ", ssid);
  WiFi.begin(ssid);
  while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
  }
  Serial.println(" CONNECTED");

  //init and get the time
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
  printLocalTime();
```

```
 //disconnect WiFi as it's no longer needed
  WiFi.disconnect(true);
  WiFi.mode(WIFI_OFF);

  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{
 delay(1000);

  printLocalTime();
 hour = timeinfo.tm_hour;
  minute = timeinfo.tm_min;
  seconds = timeinfo.tm_sec;
  Serial.print(hour);
  Serial.print(":");
  Serial.print(minute);
  Serial.print(":");
  Serial.print(seconds);
  PublishData(hour,minute,seconds);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}
void PublishData(int hour,int minute,int seconds) {
  mqttconnect();//function call for connecting to ibm
String payload = "{\"hour\":";
  payload += hour;
```

```
  payload += "," "\"minute\":";
  payload += minute;
  payload += "," "\"seconds\":";
  payload += seconds;
  payload += "}";

  Serial.print("Sending payload: ");
  Serial.println(payload);


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud
then it will print publish ok in Serial monitor or else it will print publish
failed
  } else {
    Serial.println("Publish failed");
  }

}
void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
```

```
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to
establish the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength)
{
```

```
  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="lighton")
  {
Serial.println(data3);
digitalWrite(LED,HIGH);
  }
  else
  {
Serial.println(data3);
digitalWrite(LED,LOW);
  }
data3="";
}
```

Output:

```
Publish ok
22:16:12Sending payload: {"hour":22,"minute":16,"seconds":12}
Publish ok
22:16:14Sending payload: {"hour":22,"minute":16,"seconds":14}
Publish ok
22:16:16Sending payload: {"hour":22,"minute":16,"seconds":16}
Publish ok
```

Wokwi simulation: https://wokwi.com/projects/348580604330115667