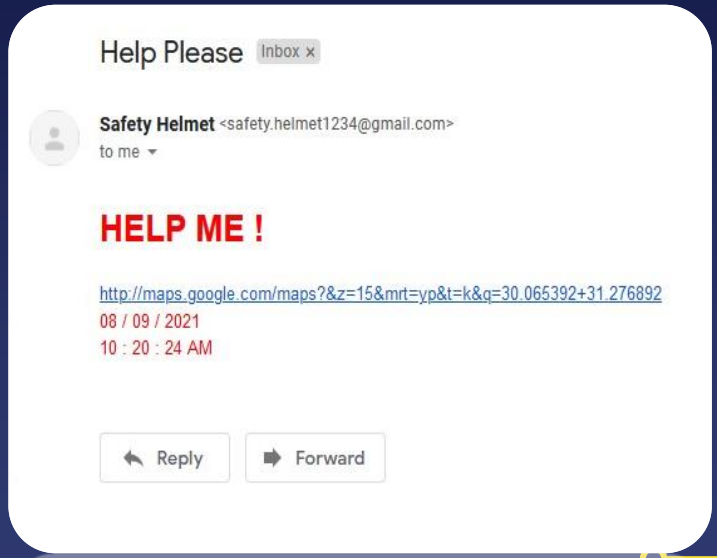
Saftey Helmet


SUSTAINABLE
DEVELOPMENT
GOALS
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

Project Aim :

Project Description :

Any bike rider can use it to safe himself from accidents When the helmet hit any thing, the mobile application start count to 30 second and if the bike rider don't stop it The helmet sends E-mail to specific person with the location to get help



Our Component :

01

NodeMCU
(ESP8266 Wi-Fi)

02

Ublox NEO-6m
GPS Module

03

Vibration Sensor
Module (801S)

Some of Code :

```
int vibration = D7; led_wifi = D4; counter = D5;
void smtpCallback(SMTP_Status status)
{
  Serial.println(status.info());
  if (status.success()){
    Serial.println("----------------");
    ESP_MAIL_PRINTF("Message sent success: %d\n", status.completedCount());
    ESP_MAIL_PRINTF("Message sent failled: %d\n", status.failedCount());
    Serial.println("----------------\n");
    struct tm dt;
    for (size_t i = 0; i < smtp.sendingResult.size(); i++)
    {
      SMTP_Result result = smtp.sendingResult.getItem(i);
      time_t ts = (time_t)result.timestamp;
      localtime_r(&ts, &dt);
      ESP_MAIL_PRINTF("Message No: %d\n", i + 1);
      ESP_MAIL_PRINTF("Status: %s\n", result.completed ? "success" : "failed");
      ESP_MAIL_PRINTF("Date/Time: %d/%d/%d %d:%d:%d\n", dt.tm_year + 1900, dt.tm_mon + 1, dt.tm_mday, dt.tm_hour, dt.tm_min, dt.tm_sec);
      ESP_MAIL_PRINTF("Recipient: %s\n", result.recipients);
      ESP_MAIL_PRINTF("Subject: %s\n", result.subject);
    }
    Serial.println("----------------\n");
  }
}
```

```
}
void sendemail()
{
  smtp.debug(1);
  smtp.callback(smtpCallback);
  ESP_Mail_Session session;
  session.server.host_name = SMTP_HOST;
  session.server.port = SMTP_PORT;
  session.login.email = AUTHOR_EMAIL;
  session.login.password = AUTHOR_PASSWORD;
  session.login.user_domain = "";
  SMTP_Message message;

  message.sender.name = "Safety Helmet";
  message.sender.email = AUTHOR_EMAIL;
  message.subject = "Help Please";
  message.addRecipient("team-12", RECIPIENT_EMAIL);

  String htmlMsg = "<div style=\"color:#FF0000;\"><h1>HELP ME !</h1><p>" + sms + "</p></div>";
  message.html.content = htmlMsg.c_str();
  message.text.charSet = "us-ascii";
  message.html.transfer_encoding = Content_Transfer_Encoding::enc_7bit;

  if (!smtp.connect(&session))return;

  if (!MailClient.sendMail(&smtp, &message))
    Serial.println("Error sending Email, " + smtp.errorReason());
}

```

```
void setup()
{
  Serial.begin(115200);
  ss.begin(9600);
  Blynk.begin(auth, WIFI_SSID, WIFI_PASSWORD);
  Serial.println();
  Serial.print("Connecting to AP");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(200);
  }
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  pinMode(vebration, INPUT);
  pinMode(led_wifi, OUTPUT);
  pinMode(counter, OUTPUT);
  digitalWrite(led_wifi, HIGH);
  time1=millis();
}
```

```
void loop()
{
  gps_fn();
  Blynk.run();
  int num = digitalRead(vebration);
  if (num == 1 && run2)
  {
    Serial.println("count is started for 30 second");
    run1 = true;
    run2 = false;
    time1 = millis();
    digitalWrite(counter, HIGH);
  }
  if (run1)
  {
    if (pin==0)
    {
      if (millis()- time1 > 30000)
      {
        sendemail();
        time1 = millis();
        run1 = false;
        run2 = true;
        digitalWrite(counter, LOW);
      }
    }
    else
    {
      run1 = false;
      run2 = true;
      digitalWrite(counter, LOW);
      Serial.println("count is stoped :( ");
    }
```

Thanks