**PART1 NFC SCAN:**The first step is to change the bike, i.e from LOCK to UNLOCK and vice versa, msg sent from nrf device to the esp LoRa device

Open, build and flash the **NFC\_SCAN.c** into the nrf board

**PART 2 LoRa Payload Upload:**

This code **LoRa.cpp** was supposed to take payload (STATE: LOCKED, Location: 42.089654, -75.970346) from the NFC nrf device to LoRa esp device and upload it to the TTN, but we were unable to establish successful connection between the nrf and esp device, and so for testing purpose we developed another code **LoRa\_test.cpp** which sent the payload (STATE: LOCKED, Location: 42.089654, -75.970346) to the ttn.

Open, build and flash the **LoRa\_test.cpp** into the esp board, make sure the antenna is connected to esp board

Use the **update.js** to update the formatting in Payload Formatters

**PART 3 Website**

Right-click on the folder, select "Send to" > "Compressed (zipped) folder," and rename the resulting zip file (e.g., smart\_bike\_webserver.zip)

To load and execute the project , start by ensuring the folder structure is correctly organized. The smart\_bike\_webserver folder should contain subfolders static (which includes files like script.js, style.css, and image assets such as Lock.gif, 22.jpg, and 2.gif) and templates (which contains the main index.html file). Alongside these, the root folder must include the Python file app.py, the live\_data.json file for storing data, and the requirements.txt file for installing dependencies.

Once the structure is confirmed, open a terminal and navigate to the smart\_bike\_webserver directory. Use the command pip install -r requirements.txt to install all the necessary Python libraries listed in the requirements.txt file. After the dependencies are installed, you can start the web server by running the command python app.py. This will initialize the server locally, typically accessible at http://127.0.0.1:5000. Open this URL in a web browser to verify that the website is loading correctly and fetching data from the live\_data.json file as expected