1. Stocker les coordonnées sur une carte SD/l’esp8266 pour les utiliser sur un ordinateur  
   <https://www.gpsvisualizer.com/>

https://www.instructables.com/DIY-GPS-Tracker-Python-Application/

1. Communiquer en utilisant le Wifi avec un ordinateur (voir <https://www.instructables.com/Audio-Controlled-Home-Using-Esp8266/>)
2. Afficher les données sur un écran LCD ou OLED + Contrôle avec boutons et/ou potentiomètre
3. Montrer les coordonnées sur Google Earth :

<https://earth.google.com/>  
<https://developers.google.com/kml/documentation/kml_tut?hl=fr>  
<https://www.gpsbabel.org/>

1. Communiquer avec un service Cloud à travers une connexion cellulaire/4G/Lora
2. Boite ne s’ouvrant que à un endroit précis
3. Jeu de chasse au trésor(geocaching) avec un système de chaud/froid + directions + localisation des participants
4. Donner des directions pour suivre un chemin précis
5. Envoyer un signal de détresse(mail, SMS,…) incluant la position.
6. Enregistrer et afficher le parcours d’un promeneur/cycliste (déjà fait/peu original)

Idées de nouvelles étapes

1. ESP8266 NodeMCU Web Server using Server-Sent Events (Update Sensor Readings Automatically)

<https://randomnerdtutorials.com/solved-reconnect-esp8266-nodemcu-to-wifi/>

1. ESP8266 NodeMCU Async Web Server – Control Outputs with Arduino IDE (ESPAsyncWebServer library)

<https://randomnerdtutorials.com/esp8266-nodemcu-async-web-server-espasyncwebserver-library/>

1. ESP8266 Web Server using SPIFFS (SPI Flash File System) – NodeMCU

<https://randomnerdtutorials.com/esp8266-web-server-spiffs-nodemcu/>

1. Reconnect ESP8266 NodeMCU to Wi-Fi Network After Lost Connection

<https://randomnerdtutorials.com/solved-reconnect-esp8266-nodemcu-to-wifi/>