Guide Projet IoT : "Système Surveillance de Santé à Distance"

Présenter par :

- Mr YAYA Mohamedhen – Code Apogée : 23031093
- Mr LEFORT Nomenjanahary Nuno – Code Apogée : 23031093

Encadrer par :

- Mme Hafssa BENABOUD – Docteur HDR

Introduction

Ce document fournit un guide étape par étape pour reproduire le projet "Système Surveillance de santé à distance". L'objectif est d'assurer que toute personne ayant accès à ces instructions puisse recréer le projet avec succès sous Windows 10.

Description Projet

Le projet Système de Surveillance de Santé à Distance vise à développer une solution intégrée permettant de surveiller, analyser et afficher les données vitales des patients en temps réel. En s'appuyant sur une architecture technologique complète, ce système garantit un suivi précis et une interaction fluide entre les différents composants techniques.

Pré-requis système

Avant de commencer, assurez-vous d'avoir :

- Un ordinateur sous Windows 10.
- Une connexion Internet stable pour télécharger les outils et bibliothèques nécessaires.
- Des privilèges administrateur pour l'installation des logiciels.

Architecture Projet

Le projet se construira comme suite :

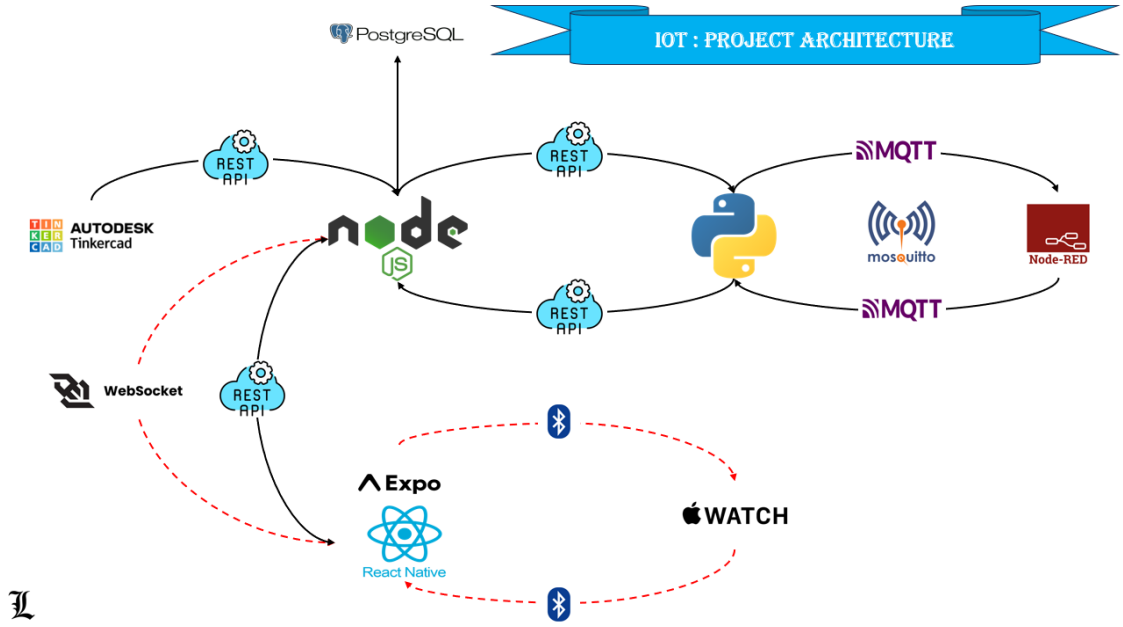

Figure 1: Architecture du projet

Étapes détaillées

Configuration de ThinkerCAD

Utilité : Simuler un circuit Arduino Uno pour collecter des données (e.g., fréquence cardiaque, température, taux d'oxygène, …).

1. Accédez à ThinkerCAD https://www.tinkercad.com/.
2. Créez un compte ou connectez-vous.
3. Lancez un nouveau projet en choisissant 'Circuits'.
4. Ajoutez une carte Arduino Uno et configurez les capteurs nécessaires en suivant le schéma électrique.
5. Chargez un programme Arduino (code) pour transmettre les données via API REST.

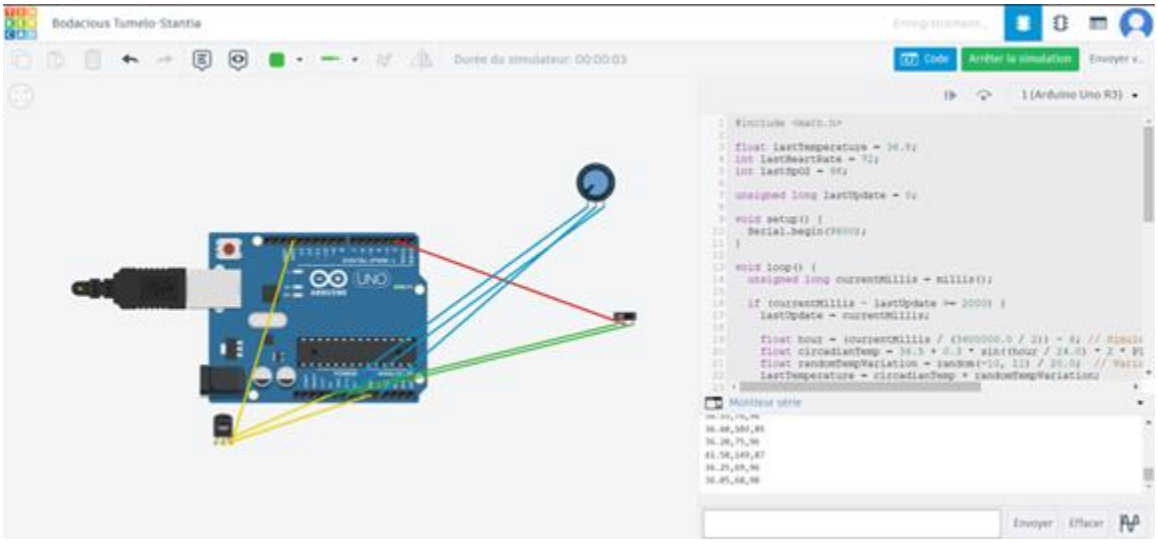

Figure 2: Simulation avec Arduino-Uno

Installation de Node.js et PostgreSQL

Utilité : Node.js sert à créer une API pour stocker et gérer les données simulées, et PostgreSQL est la base de données pour persister ces données.

Node.js

1. Téléchargez et installez Node.js depuis https://nodejs.org/.
2. Vérifiez l'installation avec :
   node -v
   npm -v
1. Lancez :
   a. Project\IoT-3S\api_service\> npm install
   b. Project\IoT-3S\api_service\> npm start
2. Résultat :

   Lancé sur ADRESS_IP_MACHINE:PORT ....

   Connexion à la base de données '[nom_base_de_donnée]' réussie.

PostgreSQL

1. Téléchargez PostgreSQL depuis : https://www.postgresql.org/download/.

2. Suivez l’assistant d’installation pour configurer une base de données locale avec pgAdmin.
3. Créez une base de données nommée par exemple 'surveillance_sante'.
4. Configurez les tables suivant le fichier :

    Project\IoT-3S\api_service\config\sql arduino-uno-init.sql

Configuration de Python, MQTT et Mosquitto

Utilité : Python pour traiter les données en temps réel, MQTT pour la communication, et Mosquitto comme broker MQTT.

Python:

1. Téléchargez Python depuis https://www.python.org/.
2. Ajoutez Python à votre PATH lors de l’installation.
3. Installez les bibliothèques nécessaires : pip install paho-mqtt psycopg2
4. Lancez :

    Project\IoT-3S\python> py main.py

Mosquitto:

1. Téléchargez Mosquitto depuis https://mosquitto.org/download/.
2. Installez Mosquitto et démarrez le service en exécutant :
    C:\Program Files\mosquitto> mosquitto
3. Configurez un fichier mosquitto.conf pour ajuster le port ou ajouter un mot de passe si nécessaire.

Installation et Configuration de Node-RED

Utilité : Afficher les données en temps réel dans un tableau de bord.

1. Installez Node-RED globalement : npm install -g node-red
2. Lancez Node-RED depuis CMD avec : node-red
3. Accédez à l’interface via http://localhost:1880 dans un navigateur.
4. Ajoutez des nœuds MQTT et connectez-les au broker Mosquitto pour recevoir les données.

Capture d’écran NodeRED

Développement de l'Application Mobile avec React Native

Utilité : Fournir une interface utilisateur pour les données des patients et effectuer des actions CRUD.

1. 1.Telecharger Expo Go dans App Store(iOs) | Play Store(Android)
2. Lancez :

   Project\IoT-3S\mobile> npm install

   Project\IoT-3S\mobile> npm start
3. 3.Testez l'application sur un appareil physique en scannant le QR code avec Expo Go.

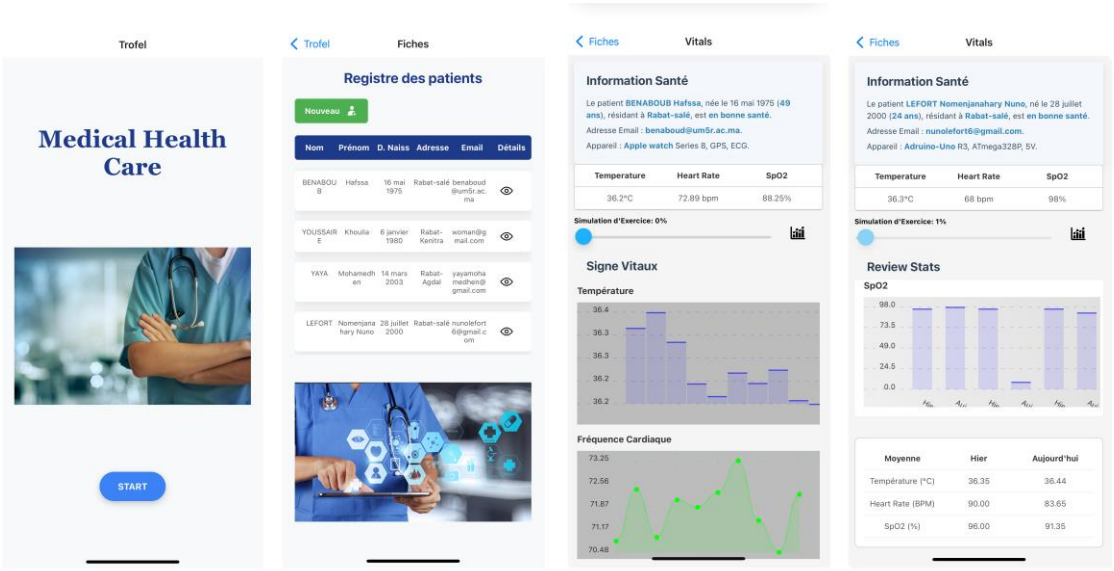

Figure 3: Interface Mobile

Implémentation de l'Envoi d'E-mails

Utilité : Notifier les utilisateurs pour des alertes ou des événements critiques.

1. Installez Nodemailer : npm install nodemailer
2. Configurez un transporteur SMTP avec Gmail :

```
const nodemailer = require('nodemailer');

const transporter = nodemailer.createTransport({

  service: 'gmail',
```

```
        auth: { user: 'votre_email@gmail.com', pass: 'votre_mot_de_passe' }
    });
```

3. Ajoutez une fonction pour envoyer des e-mails avec Nodemailer.

Test et Validation

1. Simulez des données avec ThinkerCAD.
2. Vérifiez leur transmission via MQTT ; HTTPS et leur stockage dans PostgreSQL.
3. Assurez-vous que les données apparaissent dans Node-RED et l'application Mobile.
4. Testez l'envoi des e-mails avec des alertes simulées.

Conclusion

En suivant ce guide, il est possible de reproduire intégralement le projet "Surveillance de santé à distance" avec les outils et technologies spécifiés. Cette solution permet une surveillance efficace et une gestion intuitive des données de santé.