

Rapport de Projet - SmartPlant

Vue d'Ensemble du Projet

SmartPlant est un système intelligent d'arrosage automatique de plantes qui combine hardware IoT (Arduino/ESP32), capteurs environnementaux, et une interface web moderne pour moniter et contrôler l'arrosage en temps réel.

Objectifs du Projet

1. **Automatiser l'arrosage** des plantes en fonction de l'humidité du sol
2. **Surveiller en temps réel** les conditions environnementales (température, humidité air/sol, pluie)
3. **Déetecter les maladies** des plantes via intelligence artificielle
4. **Offrir 3 modes d'arrosage** : Automatique, Manuel, Programmé
5. **Visualiser les données** avec des graphiques et statistiques
6. **Gérer un inventaire** de plants de tomates avec suivi détaillé

Architecture Technique

Hardware (Arduino + Capteurs)

- **Microcontrôleur** : Arduino UNO avec ESP8266 WiFi
- **Capteurs**:
 - DHT22 (température et humidité air)
 - Capteur d'humidité du sol (analogique)
 - Capteur de pluie (analogique)
- **Actuateurs** : Relais pour pompe à eau
- **Indicateur** : LED de statut

Backend (Firebase)

- **Firebase Realtime Database** pour stockage des données
- Synchronisation bidirectionnelle temps réel
- Stockage de :
 - Données capteurs
 - Configuration système
 - Historique d'arrosage
 - Inventaire des plantes
 - Résultats d'analyse IA

Frontend (Dashboard Web)

- **Technologies** : HTML5, CSS3, JavaScript

- **Librairies:**
 - Chart.js (visualisation données)
 - Firebase SDK (connexion backend)
 - TensorFlow.js & Teachable Machine (IA)
 - Roboflow API (détectio
- **Design :** Interface moderne dark mode, responsive

Fonctionnalités Principales

1. Page Accueil (Dashboard)

- Vue d'ensemble en temps réel
- Cartes capteurs avec valeurs actuelles et statut
- Widget météo intégré (API Open-Meteo)
- Système d'alertes intelligent
- Indicateurs de santé globale

2. Page Analyse

- Graphiques en temps réel (humidité sol, température, humidité air)
- Filtres temporels (temps réel, 7 jours, 30 jours)
- Statistiques détaillées
- Score de santé global
- Évolution historique

3. Page Arrosage (3 Modes)

Mode Automatique

- Déclenchement basé sur seuil d'humidité
- Configuration personnalisable :
 - Seuil minimum (%)
 - Durée d'arrosage (secondes)
 - Délai minimum entre arrosages (minutes)
- Prise en compte de la pluie

Mode Manuel

- Contrôle direct ON/OFF
- Durée configurable
- Feedback visuel instantané
- Arrêt d'urgence possible

Mode Programmé

- 2 programmes horaires indépendants
- Configuration :
 - Heure et minute d'exécution

- Durée personnalisée
- Activation/désactivation par programme
- Exécution automatique quotidienne

4. Page Mes Plantes

- **Gestion d'inventaire** de plants de tomates
- **Informations par plant:**
 - Nom et variété
 - Date de plantation et âge
 - Emplacement
 - Notes personnalisées
 - Dernier arrosage
 - Nombre d'arrosages et scans IA
- **Statistiques globales:**
 - Total plants actifs
 - Plants en bonne santé vs nécessitant attention
 - Âge moyen
- **Conditions environnementales actuelles**
- **Actions rapides** : Arroser, Scanner IA
- **Filtres et tri** : par santé, âge, nom

5. Page Santé & IA

- **Upload d'images** (drag & drop ou clic)
- **Analyse par IA:**
 - Teachable Machine (modèle local)
 - Roboflow API (détectio cloud)
- **Résultats détaillés:**
 - Classe détectée
 - Niveau de confiance
 - Recommandations
- **Historique des analyses** avec images
- **Intégration** avec l'inventaire des plantes

6. Page Configuration

- **Paramètres d'arrosage:**
 - Durée par défaut
 - Mode au démarrage
 - Heure programmée
 - Délai minimum
- **Seuils d'alerte:**
 - Humidité minimum/maximum
 - Température critique
 - Visualisation des zones
- **Configurations prédéfinies:**
 - Méditerranéen

- Plants en pots
- Économie d'eau
- Plein terre
- Arrosage soir
- Standard

Flux de Données

Arduino → Firebase Realtime DB ← Dashboard Web



Cycle :

1. Arduino lit les capteurs (toutes les 2s)
2. Données envoyées à Firebase
3. Dashboard affiche en temps réel
4. Utilisateur modifie configuration
5. Arduino lit commandes Firebase
6. Arduino exécute actions (arrosage)
7. Historique sauvegardé

Points Forts du Design

- **Interface moderne** avec dégradés et effets glassmorphism
- **Dark mode** optimisé pour usage longue durée
- **Animations fluides** et transitions professionnelles
- **Responsive design** : desktop, tablette, mobile
- **Icônes emoji** pour lisibilité et aspect ludique
- **Status badges** colorés pour feedback instantané
- **Graphiques interactifs** avec Chart.js
- **Modal dialogs** pour actions importantes

Technologies Utilisées

Hardware

- Arduino UNO
- ESP8266 WiFi Module
- DHT22, Capteurs analogiques
- Relais 5V

Backend

- Firebase Realtime Database
- Firebase Authentication

Frontend

- HTML5, CSS3 (Variables CSS, Grid, Flexbox)
- JavaScript ES6+ (Promises, Async/Await)
- Firebase SDK 9.22.0
- Chart.js
- TensorFlow.js
- Teachable Machine
- Roboflow API

APIs Externes

- Open-Meteo (météo)
- Roboflow (détection maladies)

Indicateurs de Performance

- **Rafraîchissement données** : 2 secondes
- **Synchronisation Firebase** : Temps réel
- **Historique** : Jusqu'à 720 points (30 jours)
- **Analyse IA** : < 5 secondes
- **Responsive** : Desktop, tablette, mobile

Évolutions Futures Possibles

1. **Application mobile native** (React Native, Flutter)
2. **Multi-zones d'arrosage** (plusieurs pompes)
3. **Capteurs pH et luminosité**
4. **Prévisions météo avancées**
5. **Intégration caméra** pour surveillance
6. **Machine Learning** pour optimisation arrosage
7. **Système de recommandations** personnalisées
8. **Export de rapports PDF**
9. **Intégration assistant vocal** (Alexa, Google Home)