

OnBoarding

▼ Domaine	Management
☰ pole(s)	General

Checklist d'Onboarding – Projet Harvest Hub (Équipe Web & Mobile)

1. Compréhension du projet

- [] Lire la fiche de présentation du projet (description, objectifs, livrables)
- [] Comprendre le concept de jardin connecté (capteurs, IA, recettes, jardin \
- [] Identifier les grandes étapes du projet : MVP → Finalisation → Soutenance

2. Rôles et responsabilités

- [] Identifier son équipe (Mobile/Web ou IoT)
- [] Connaître les responsables de chaque pôle :
 - Web & Mobile : Ewan (chef), Pierre, Ahmed
 - IoT : Baptiste (chef), Dorian, Evan, Maxime
- [] Comprendre les interactions entre les équipes

3. Présentation de l'équipe Web & Mobile

- [] Connaître la répartition entre le **site web** et **l'application mobile**
- [] Comprendre les attentes côté UX/UI (maquettes, responsive, accessibilité
- [] Objectif MVP : interface fonctionnelle affichant les données des capteurs

4. Stack technique Web & Mobile

- [] Langages : TypeScript, HTML/CSS
- [] Frameworks :
 - Web : React.js
 - Mobile : React Native
- [] API Backend : Go (Golang)
- [] Outils : Shadcn/UI
- [] Base de données : TimescaleDB

5. Installation de l'environnement de dev

- [] Cloner le repo GitHub de l'équipe Web & Mobile

- [] Installer :
 - [] Node.js (v18+ recommandé)
 - [] React.js ou React Native (selon la plateforme)
 - [] Dépendances du projet (`npm install`, `pnpm`, ou `yarn`)
 - [] Axios (requêtes HTTP vers le backend en Go)
 - [] Shadcn/UI et Tailwind (UI components)
- [] Configurer l'environnement :
 - [] Fichier `.env` pour les variables (URL API, tokens...)
 - [] Débogueur (VSCode, React DevTools, etc.)
- [] Suivre les branches Git définies : `web/dev`, `mobile/dev`, `web/ui`, etc.

📁 6. Organisation et gestion du projet

- [] Rejoindre les outils :
 - [] GitHub (repos)
 - [] Discord (communication)
 - [] Notion (docs, suivi)
- [] Utiliser le tableau **Kanban**
- [] Participer aux **réunions mensuelles**
- [] Suivre les **sprints mensuels**

🔒 7. Sécurité & Fiabilité

- [] Gérer l'authentification (JWT, stockage sécurisé des tokens)
- [] Vérifier les appels vers l'API Go (gestion d'erreurs, statut 401/500, etc.)
- [] Implémenter les protections UI (form validation, fallback, etc.)
- [] Tests frontend :
 - [] Unitaires
 - [] End-to-end
- [] Vérifier la cohérence des données issues de TimescaleDB (via l'API)

📄 8. Documentation à lire ou compléter

- [] API contract Go (endpoints, schéma JSON, erreurs)
- [] TimescaleDB : structure des données (séries temporelles)
- [] Spécifications fonctionnelles (interfaces, parcours utilisateurs)
- [] API contract (routes, paramètres, formats JSON)
- [] Conventions de nommage / architecture front
- [] Bonnes pratiques de développement web & mobile : [cf. Plan de qualité, 1.A & 1.B]
- [] Normes de développement web & mobile : [cf. Plan de qualité, 1.A & 1.B]

- [] Maquettes UI (Figma, Notion)
- [] Wiki interne sur Notion
- [] Roadmap produit

🎯 9. Objectifs MVP (à atteindre avant été 2026)

- [] Affichage en temps réel des données capteurs (température, humidité, et
- [] Visualisation graphique de l'évolution des données (charts, courbes)
- [] Création d'un compte utilisateur + authentification
- [] Dashboard jardin virtuel connecté
- [] Interfaçage API avec le backend IoT

✅ Checklist d'Onboarding – Projet Harvest Hub (Équipe IoT)

🧠 1. Compréhension du projet

- [] Lire la fiche de présentation du projet (description, objectifs, livrables)
- [] Comprendre le concept de jardin connecté (capteurs, IA, recettes, jardin v
- [] Identifier les grandes étapes du projet : MVP → Finalisation → Soutenance

👥 2. Rôles et responsabilités

- [] Identifier son équipe (Mobile/Web ou IoT)
- [] Connaître les responsables de chaque pôle :
 - Web & Mobile : Ewan (chef), Pierre, Ahmed
 - IoT : Baptiste (chef), Dorian, Evan, Maxime
- [] Comprendre les interactions entre les équipes

🌿 3. Présentation de l'équipe IoT

- [] Comprendre la finalité des capteurs (température, humidité, luminosité, et
- [] Connaître le rôle du hub de terrasse (matériel, communication avec les so
- [] Objectif MVP : 2 sondes + 1 hub connecté + collecte de données fiables

🛠️ 4. Stack technique IoT

- [] Langages : Go (backend), Rust (embarqués/capteurs)
- [] Protocole réseau sondes : ESP-NOW et BLE
- [] Protocole réseau hub : Wi-Fi et cellulaire

- [] Matériel envisagé : STM32WB55CCU6 (Sonde), ESP32 (Hub)
- [] Base de données : TimescaleDB

🧱 5. Installation de l'environnement de dev

- [] Cloner le repo GitHub de l'équipe IoT
- [] Installer :
 - [] Go (backend)
 - [] Rust (pour la partie firmware sonde)
 - [] STM32CubeIDE ou STM32CubeProgrammer (pour flasher les STM32WB55CCU6)
 - [] Espressif IDF ou ESP-Flasher (pour l'ESP32)
 - [] Drivers USB vers UART (CP210x ou CH340 selon matériel)
- [] Suivre les branches Git définies : `iot/dev`, `iot/firmware`, etc.

📁 6. Organisation et gestion du projet

- [] Rejoindre les outils :
 - [] GitHub (repos)
 - [] Discord (communication)
 - [] Notion (docs, suivi)
- [] Utiliser le tableau **Kanban**
- [] Participer aux **réunions mensuelles**
- [] Suivre les **sprints mensuels**
- [] rejoindre le dossier partagé pour les firmwares / binaires des sondes et du hub

🛡️ 7. Sécurité & Fiabilité

- [] Implémenter :
 - [] Pairage BLE sécurisé entre STM32 et ESP32 (whitelist, authentification)
 - [] Cryptage des données transmises (ESP-NOW ou BLE)
- [] Gérer les tests :
 - [] Simulation de capteurs (avec payloads fixes)
 - [] Tests en conditions réelles (transfert BLE/ESP-NOW)
- [] Mettre en place :
 - [] Un système de logs côté ESP32 (UART ou stockage local)
 - [] Gestion des pertes de communication / timeout

📄 8. Documentation à lire ou compléter

- [] Fiches techniques des capteurs
- [] Architecture système (schémas, dataflow)
- [] Bonnes pratiques de développement IOT : [cf. Plan de qualité, 1.A & 1.B](h

- [] Normes de développement IOT : [cf. Plan de qualité, 1.A & 1.B](https://gith
- [] Wiki interne sur Notion
- [] Spécifications fonctionnelles
- [] Roadmap produit
- [] Documentation STM32WB55 : BLE stack, flashage, consommation
- [] Protocole ESP-NOW (format, portée, limites)
- [] Communication BLE → Wi-Fi via ESP32
- [] Schéma physique des composants (sonde ↔ hub ↔ backend)

🎯 9. Objectifs MVP (à atteindre avant été 2026)

- [] Lecture fiable depuis 2 capteurs physiques
- [] Transmission sécurisée au hub de terrasse
- [] Centralisation des données sur le backend
- [] Affichage dans l'application web/mobile via l'API