SPRINT 1 — STABILISATION

Projet : HabitQuest
Période : 17 — 21 février 2026
Objectif : Le parcours utilisateur principal fonctionne à 100%, zéro bug visible.

USER STORIES — Sprint 1

Classées par ordre de priorité. **On ne commence pas une US de priorité inférieure tant que celles au-dessus ne sont pas terminées.**

US-1.1 — Audit du parcours complet

Assigné à : Shiro + Kuro (ensemble)
Estimation : 4h (lundi matin)
Priorité : CRITIQUE

En tant que lead de projet,
je veux un audit complet du parcours utilisateur sur l'environnement de production,
afin de lister tous les bugs à corriger cette semaine.

Parcours à tester :

1. Register (email + mot de passe)
2. Onboarding (création personnage : nom, classe, genre, apparence)
3. Dashboard (affichage correct : XP, streak, niveau)
4. Créer une habitude (quotidienne, avec icône et couleur)
5. Créer une habitude avec objectif comptable (ex: 3/8 verres d'eau)
6. Compléter une habitude → vérifier XP et Coins gagnés
7. Vérifier le streak après complétion
8. Créer une tâche → vérifier évaluation IA (si active)
9. Aller en boutique → acheter un item
10. Aller dans l'inventaire → équiper l'item
11. Vérifier que le personnage affiche l'équipement
12. Page stats/calendar → vérifier affichage
13. Page badges → vérifier affichage
14. Logout → Re-login → vérifier persistance des données

Tester sur :

- Desktop (Chrome)
- Mobile (iPhone Safari)
- Mobile (Android Chrome)

Critères d'acceptation :

- ☐ Chaque étape est testée et documentée (OK ou BUG)

- ☐ Chaque bug est consigné avec : page, action, résultat attendu, résultat obtenu, screenshot
- ☐ Les bugs sont classés : 🟥 Bloquant / 🟧 Gênant / 🟨 Cosmétique
- ☐ Liste de bugs partagée avant midi

🟥 US-1.2 — Fix bugs core loop (Backend)

Assigné à : Kuro
Estimation : 12h (budget, dépend de l'audit)
Priorité : CRITIQUE
Dépend de : US-1.1

En tant que utilisateur,
je veux que toutes les API du parcours principal fonctionnent correctement,
afin de pouvoir utiliser l'app sans erreur.

Périmètre : Tous les bugs 🟥 Bloquants identifiés par l'audit côté backend :

- Endpoints qui retournent des erreurs 500
- Calculs XP/Coins incorrects
- Streak qui ne s'incrémente pas
- Onboarding qui crash
- Shop/achat qui ne fonctionne pas
- Équipement qui ne s'applique pas

Critères d'acceptation :

- ☐ Tous les bugs 🟥 backend identifiés dans l'audit sont fixés
- ☐ Chaque fix est accompagné d'un test unitaire qui couvre le cas
- ☐ Les endpoints concernés retournent les bonnes réponses (tester via /api/docs)
- ☐ Aucune régression sur les 83 tests existants (pytest passe toujours)

🟥 US-1.3 — Fix bugs core loop (Frontend)

Assigné à : Shiro
Estimation : 12h (budget, dépend de l'audit)
Priorité : CRITIQUE
Dépend de : US-1.1

En tant que utilisateur,
je veux que toutes les pages du parcours principal s'affichent et fonctionnent correctement,
afin de pouvoir naviguer sans erreur ni page blanche.

Périmètre : Tous les bugs 🟥 Bloquants identifiés par l'audit côté frontend :

- Pages qui crash (React error boundary)
- Formulaires qui ne soumettent pas
- Animations XP/LevelUp qui ne se déclenchent pas
- Affichage personnage/sprites cassé
- Navigation cassée (liens morts, redirections en boucle)

Critères d'acceptation :

- [ ] Tous les bugs 🟢 frontend identifiés dans l'audit sont fixés
- [ ] Aucun crash React visible (pas d'écran ErrorBoundary sur le parcours principal)
- [ ] Le parcours complet est fluide sur Desktop Chrome
- [ ] Les formulaires (register, create habit, create task) fonctionnent sans erreur

---

🟢 US-1.4 — Configurer OpenAI + fiabilité Celery/LLM

Assigné à : Kuro
Estimation : 8h
Priorité : HAUTE

En tant que utilisateur,
je veux que quand je crée une tâche, l'IA évalue automatiquement sa difficulté et propose des XP,
afin de recevoir des récompenses adaptées.

Sous-tâches :

1. Configurer les env vars en prod : `OPENAI_API_KEY`, `LLM_PROVIDER=openai`, `LLM_MODEL=gpt-4o-mini`
2. Tester la création d'une tâche avec évaluation IA end-to-end (via Celery)
3. Vérifier les cas d'erreur :
   - API OpenAI timeout (simuler avec un timeout court)
   - API OpenAI retourne une erreur 429 (rate limit)
   - API OpenAI down (clé invalide)
4. Implémenter un fallback : si le LLM échoue, assigner automatiquement `difficulty=medium`, `xp_reward=30`, `coins_reward=6`
5. Logger les erreurs LLM proprement (structlog)
6. Vérifier que le rate limit (20 évaluations/jour/user) fonctionne

Critères d'acceptation :

- [ ] Créer une tâche → l'évaluation IA retourne un résultat en < 10 secondes
- [ ] Si l'API OpenAI est down → la tâche est créée quand même avec des valeurs par défaut
- [ ] Le fallback est loggé comme WARNING (pas ERROR)
- [ ] Le rate limit bloque au-delà de 20 évaluations/jour et retourne un message clair
- [ ] Les logs Celery montrent le résultat de l'évaluation (difficulté + XP)

🟢 US-1.5 — Corriger les 12 tests skipped

Assigné à : Kuro
Estimation : 6h
Priorité : HAUTE

En tant que développeur,
je veux comprendre et résoudre les 12 tests skipped du backend,
afin de garantir que notre suite de tests couvre le code réellement en production.

Sous-tâches :

1. Lancer `pytest -v` et lister les 12 tests skipped avec leur raison
2. Pour chaque test skipped, une des 3 actions :
    - **FIXER** si le test est pertinent et que le code existe → corriger le test ou le code
    - **SUPPRIMER** si le test couvre une feature qu'on a volontairement désactivée (ex: combat) → supprimer avec un commentaire expliquant pourquoi
    - **DOCUMENTER** si le test ne peut pas être fixé maintenant → créer une issue GitHub avec le contexte

Critères d'acceptation :

- [ ] `pytest -v` affiche 0 tests skipped sans raison valable
- [ ] Les tests supprimés sont documentés (commentaire ou issue GitHub)
- [ ] Le total de tests qui passent est >= 83 (pas de régression)
- [ ] Un récap est posté dans le standup : X fixés, Y supprimés, Z documentés

---

◍ US-1.6 — Test responsive mobile

Assigné à : Shiro
Estimation : 4h
Priorité : HAUTE

En tant que utilisateur sur mobile,
je veux que l'app soit utilisable sur mon smartphone,
afin de tracker mes habitudes au quotidien.

Appareils/navigateurs à tester :

- iPhone Safari (taille iPhone 14 ou équivalent)
- Android Chrome (taille standard ~390px)
- Si pas de device physique : Chrome DevTools responsive mode

Points critiques à vérifier :

1. Menu hamburger fonctionne (ouverture/fermeture sidebar)
2. Formulaire de register lisible et utilisable
3. Onboarding (sélection classe/genre) : boutons assez gros pour le tactile
4. Liste d'habitudes : cards lisibles, bouton check-in assez gros
5. Dashboard : pas de débordement horizontal
6. Boutique : grille d'items lisible
7. Page personnage : sprite visible et centré

Critères d'acceptation :

- [ ] Aucun débordement horizontal (pas de scroll latéral involontaire)
- [ ] Tous les boutons interactifs font minimum 44x44px (standard Apple)
- [ ] Le texte est lisible sans zoomer
- [ ] Le menu hamburger ouvre/ferme la sidebar sans bug
- [ ] Les formulaires sont utilisables avec le clavier mobile (pas de champ caché par le clavier)
- [ ] Les bugs trouvés sont ajoutés à la liste de bugs de l'audit

---

🟢 US-1.7 — Masquer les features désactivées

Assigné à : Shiro
Estimation : 3h
Priorité : MOYENNE

En tant que utilisateur beta,
je veux que l'app ne montre pas de pages vides ou de fonctionnalités cassées,
afin de ne pas avoir l'impression que l'app est buggée.

Features à traiter :

1. **Combat PvP** → Si la page est accessible : afficher un écran "Coming Soon — Disponible bientôt !" avec une illustration ou un emoji. Sinon, masquer le lien dans la sidebar.
2. **OAuth Google/Apple** → Masquer les boutons sur la page login/register
3. **Toute autre page vide** découverte pendant l'audit → Même traitement (Coming Soon ou masquer)

Design "Coming Soon" :

- Fond avec le thème actuel (dark)
- Icône ou emoji thématique (⚔️ pour le combat, etc.)
- Texte : "Cette fonctionnalité arrive bientôt !"
- Pas de date promise
- Bouton "Retour au dashboard"

Critères d'acceptation :

- [ ] Aucune page de l'app n'affiche un écran blanc ou une erreur
- [ ] Les features désactivées ont soit un écran "Coming Soon", soit leur lien est masqué
- [ ] Les boutons OAuth sont cachés sur login/register
- [ ] La sidebar ne montre que des liens vers des pages fonctionnelles (ou Coming Soon)

🟢 US-1.8 — Vérifier les seeds en production

Assigné à : Kuro
Estimation : 2h
Priorité : MOYENNE

En tant que utilisateur,
je veux que la boutique contienne des items à acheter et que les badges soient configurés,
afin de profiter de la gamification dès le premier jour.

Sous-tâches :

1. Se connecter à la base de prod : `docker exec -it habit-postgres psql -U habit_user -d habit_tracker`
2. Vérifier les items : `SELECT COUNT(*) FROM items WHERE is_available = true;` → doit être >= 38
3. Vérifier les badges : `SELECT COUNT(*) FROM badges;` → doit être >= 75

4. Vérifier la répartition des items par catégorie : `SELECT category, COUNT(*) FROM items GROUP BY category;`
5. Vérifier la répartition des items par rareté : `SELECT rarity, COUNT(*) FROM items GROUP BY rarity;`
6. Si des données manquent → relancer les scripts de seed :
    - `docker compose exec backend python scripts/seed_items.py`
    - `docker compose exec backend python scripts/seed_badges.py` (si existant)

Critères d'acceptation :

- ☐ >= 38 items disponibles en boutique, avec toutes les catégories représentées
- ☐ >= 75 badges en base, avec toutes les raretés représentées
- ☐ La page boutique affiche les items correctement côté frontend
- ☐ La page badges affiche la collection correctement côté frontend

---

📊 RÉCAP CHARGE PAR DEV

Dev	US assignées	Heures estimées	Marge sur 35h
Kuro	1.1 (2h) + 1.2 (12h) + 1.4 (8h) + 1.5 (6h) + 1.8 (2h)	**30h**	5h
Shiro	1.1 (2h) + 1.3 (12h) + 1.6 (4h) + 1.7 (3h)	**21h**	14h

> ⚠ La marge de Shiro est volontairement plus grande car les bugs frontend découverts par l'audit pourraient être plus nombreux que prévu. Si Shiro finit en avance, elle peut commencer à préparer le terrain pour le Sprint 2 (social/friends UI).

☑ GATE DE FIN DE SPRINT

Vendredi 21 février, 17h — Test collectif :

On crée un compte neuf sur prod et on déroule :

1. ☑ Register fonctionne
2. ☑ Onboarding complet sans erreur
3. ☑ Créer 3 habitudes (1 quotidienne, 1 jours spécifiques, 1 comptable)
4. ☑ Compléter les habitudes → XP et Coins corrects
5. ☑ Dashboard affiche les bonnes stats
6. ☑ Boutique affiche les items → Achat possible
7. ☑ Inventaire → Équiper un item → Personnage mis à jour
8. ☑ Créer une tâche → Évaluation IA fonctionne (ou fallback)
9. ☑ Tout fonctionne sur mobile
10. ☑ Aucune page blanche ou erreur visible

Si un seul point est en échec → on ne passe pas au Sprint 2.