

# Grille de correction détaillée – EC02

## Informations générales

- **Épreuve** : EC02 – Conception d’interfaces avec Framework
- **Durée** : 4h – Individuelle
- **Projet fil rouge** : SkillHub – Tableau de bord utilisateur (apprenant / formateur)
- **Technologies imposées** : React, Vue ou Angular (au choix selon consignes)

## Critères d’évaluation et attentes

Critère	Intitulé officiel	Attentes côté correcteur	Points de vigilance
C1.3	Responsive Web Design avec framework	Composants et styles adaptatifs (Grid/Flexbox, utilitaires CSS du framework). Vérifier que les vues s’adaptent sur desktop, tablette, mobile.	Tester sur plusieurs tailles d’écran. Éviter les débordements et scroll horizontal.
C1.4	Accessibilité (framework)	Vérifier l’usage d’attributs ARIA dans les composants. Navigation clavier possible dans les principales interactions (listes, formulaires, modales). Rapport Wave/Lighthouse exporté.	Accessibilité souvent négligée dans les SPAs : tester focus, contraste, aria-expanded, aria-label.
C2.2	Structuration du front-end	Projet organisé en composants réutilisables, système de routing clair ( Dashboard , Mes	Vérifier arborescence cohérente ( src/components , src/views ,

Critère	Intitulé officiel	Attentes côté correcteur	Points de vigilance
		Ateliers , Créer un Atelier ). Gestion d'état maîtrisée (local ou global). Rapport Lighthouse fourni.	src/router ). Code lisible et documenté.

## Livrables à corriger

- **Code source** : projet front complet ( src/ )
- **README.md** : choix du framework, justification de l'architecture, organisation des composants
- **Rapports d'analyse** :
  - Accessibilité (Wave ou Lighthouse)
  - Structuration front (Lighthouse)

## Barème indicatif (sur 20 points)

Axe évalué	Points
Responsive design (C1.3)	/6
Accessibilité (C1.4)	/6
Structuration front (C2.2)	/8

**Tolérance** :  $\pm 1$  point par critère selon la propreté globale du code et l'effort manifeste.

## Points positifs attendus

- Composants réutilisables et bien nommés
- Routing simple et lisible (pas d'arborescence inutilement complexe)

- Bonne gestion de l'état (prop drilling limité, hooks / stores bien utilisés)
- Accessibilité respectée : labels, aria, navigation clavier fluide
- Code commenté et structuré, README utile

## Erreurs fréquentes à surveiller

- Projet structuré en un seul fichier/composant massif
- Responsivité défailante : tableau figé, cartes illisibles en mobile
- Accessibilité négligée (absence de labels, navigation clavier bloquée)
- Routing confus ou inexistant (navigation simulée sans vrai router)
- README trop minimal ou absent
- Rapport Lighthouse manquant ou non exploitable

## Rappel pédagogique

Cette épreuve marque le passage d'une **intégration statique** (EC01) à une **application dynamique et modulaire**.

Le rôle du correcteur est de vérifier que l'apprenant sait :

- Construire une **SPA (Single Page Application)** simple mais bien organisée
- Réutiliser des composants et gérer l'état de manière maintenable
- Produire une interface responsive **et accessible**
- Documenter son projet de façon claire et exploitable

Elle prépare directement les EC suivantes (API, back-end), où ce front sera relié à des données réelles et sécurisées.