

---

## 1 INFORMATIONS GENERALES

Apprenti :	Nom:	Prénom :
Lieu de travail :	ETML / VENNES / 1010 LAUSANNE	
Chef de projet :	Nom:	Prénom:
Dates de réalisation :	Du mercredi 29.01.2025 au mercredi 12.03.2025	
Horaire de travail :	5 périodes par semaine le mercredi, 6 semaines de travail, 1 de vacances	
Temps total :	30 périodes	

---

## 2 PROCÉDURE

- Tous les apprentis réalisent le projet sur la base d'un cahier des charges.
- Le cahier des charges est présenté, commenté et discuté en classe.
- Les apprentis sont entièrement responsables de la sécurité et sauvegarde de leurs données.
- En cas de problèmes graves, les apprentis avertissent le client au plus vite.
- Les apprentis ont la possibilité d'obtenir de l'aide externe, mais ils doivent le mentionner.
- Les informations utiles à l'évaluation de ce projet sont disponibles au chapitre 8.

---

## 3 TITRE

### **Flashcards**

---

## 4 SUJET

Application Web Flashcards en JS avec le framework AdonisJS.

---

## 5 MATÉRIEL ET LOGICIEL À DISPOSITION

- Un PC ETML
- Accès à Internet
- Visual Studio Code
- Node.js
- Git et github

---

## 6 PRÉREQUIS

Solides bases en javascript après le P\_BULLE du trimestre précédent

## 7 CAHIER DES CHARGES

### 7.1 Objectif pédagogique

L'objectif pédagogique de cette pratique est d'atteindre les niveaux de connaissance et d'application des principaux composants d'AdonisJS.

Le recours à l'intelligence artificielle dans le cadre de cette pratique est autorisé mais vous devez être en capacité d'expliquer chaque ligne de code que vous avez écrits (ou copiées/collées).

### 7.2 Objectif produit

Le but est de réaliser Flashcards, une application Web destinée à des étudiants qui veulent apprendre une matière d'une manière ludique : vocabulaire, mathématique, histoire...

Une flashcard est une carte servant à acquérir des connaissances. Elle a une question d'un côté et une réponse de l'autre. Pour s'exercer avec, l'étudiant passe les cartes d'un deck en revue: il consulte le côté question, énonce une réponse avant de la vérifier en retournant la carte.

Les flashcards sont regroupées par thème en paquets (« decks »).

Chaque utilisateur qui s'authentifie sur Flashcards peut créer des decks (et donc des cartes).

### 7.3 Déroulement

L'analyse fonctionnelle de l'application FlashCards a déjà été entièrement réalisé pour vous, elle est fournie séparément.

Durant votre travail sur ce projet vous devez :

- Laisser des **traces d'activité**. La trace la plus simple c'est un commit Git. L'autre moyen, c'est le journal de travail. Faites attention au niveau de détail : une tâche qui a duré plus de 40 minutes doit impérativement être détaillée dans sa description.
- Effectuer une **revue de code**. A la fin de la réalisation d'une US sur une branche Git, vous devrez créer une `pull request`, laquelle sera traitée durant une revue de code suivant le modèle fourni. Il est de votre responsabilité d'organiser la revue. Il est possible de refaire une deuxième (éventuellement une troisième) revue en cas de besoin.
- Démontrer votre maîtrise technique à travers la **réalisation d'une US** de manière autonome. Par autonome, on entend : individuellement, avec Internet, mais sans IA. La US à réaliser dans ce cadre sera révélée le jour même ; elle sera la même pour tout le monde. Une US d'exercice vous sera proposée à l'approche de la fin de module.
- Trois **Jokers** vous sont accordés au début de projet. Un joker est retiré suite à un comportement non professionnel (utilisation abusive de l'IA, du téléphone, incapable d'expliquer le code commité, non-observation des consignes, etc)
- Faire le **suivi de vos activités** dans un outil de gestion de projet. L'outil doit permettre à l'enseignant de savoir quel est le travail en cours, ce qui a déjà été fait, ce qui reste à faire. Le choix de l'outil est libre, notre recommandation est un KanBan sur Github Project.
- Tenir un **journal de travail** qui respecte les critères suivants :
  - o La structure et la présentation sont claires et soignées.

- Les sources, les fichiers, les répertoires, les commits, et autres sources d'informations concernées par le journal sont référencés.
- L'état significatif et les durées des tâches mentionnées sont précisés.
- Toutes les activités planifiées, les aides extérieures, ainsi que les imprévus (maladie) et les heures supplémentaires y sont mentionnés.
- Les succès et les échecs sont mentionnés.
- Le travail journalier et son appréciation critique, ainsi que les réflexions y figurent.

Vous n'avez pas le droit de choisir l'outil que vous voulez pour votre journal de travail. Nous vous imposons **GitJournal** qui vous évite de devoir saisir des informations redondantes.

- Travailler de manière efficace et intelligente avec (ou sans) l'IA. La stratégie que vous aurez utilisée devra être documenté.
- Produire une documentation minimale mais de qualité. Concrètement, le **ReadMe** de votre repo doit présenter
  - Les objectifs pédagogiques et produits, reformulés dans vos mots à partir de ce cahier des charges. Minimum 80 mots par type
  - La marche à suivre pour qu'un développeur lambda, ne connaissant pas particulièrement AdonisJS sois capable d'installer et de démarrer votre application.
  - Une section bilan dans laquelle vous décrivez les résultats obtenus dans ce projet par rapport aux objectifs initiaux.
  - Une section y a qui décrit et analyse la stratégie IA que vous avez utilisé. Minimum 150 mots.

## 7.4 Technologies Utilisées

**AdonisJS** : AdonisJS est un framework backend Node.js moderne, opinionated et basé sur TypeScript. Il s'inspire de Laravel (PHP) en adoptant une approche structurée et pragmatique pour le développement web, avec un accent sur la productivité des développeurs.

**npm** est inclus avec l'installation de Node.js. Cela signifie que lorsque vous installez **Node.js** sur votre machine, **npm** est automatiquement installé en même temps.

### ECMAScript Modules (ESM)

Le projet est structuré en utilisant la syntaxe ESM, permettant d'importer et d'exporter des fonctions et des variables entre les différents fichiers JavaScript. Cela permet de séparer les différentes responsabilités du code et de maintenir une structure propre et modulaire.

## 7.5 Démarrage

Node.js : Assurez-vous que Node.js est installé sur votre machine grâce à l'option '-v'.

```
~: node -v
v23.5.0
~: npm -v
10.9.2
```

Référez-vous aux documents reçus durant les premières semaines du module (application des surnoms des profs) pour créer une nouvelle application Adonis.

Dans le dossier contenant vos repos Git locaux (habituellement :  
C:\utilisateurs\p...\Documents\Github), créez le dossier flashcards

A la ligne de commande dans ce dossier, lancez la commande :

```
npm init adonisjs@latest src -- --db=mysql
```

Sélectionnez « Web Starter Kit », puis « Session » pour l'authentification.

Validez l'installation en démarrant le serveur de développement, exécutez :

```
cd src
```

```
npm run dev
```

Le projet sera servi localement à l'adresse <http://localhost:3333/>.

### Contrôle de version :

Quand votre serveur fonctionne, créez un repo Git dans le dossier flashcards avec :

```
~/Documents/Github: cd flashcards/  
~/Documents/Github/flashcards: git init -b main  
Dépôt Git vide initialisé dans /Users/Xavier/Documents/Github/flashcards/.git/
```

Ajoutez ce repo dans Github Desktop. Publiez-le sur votre compte, en privé, puis invitez votre enseignant (en lui donnant le droit « write »)

---

## 8 Évaluation

La grille d'évaluation du projet est fournie en pièce jointe au job sur Marketplace.

L'onglet « Contextualisation » fait référence aux éléments de ce cahier des charges.