

## 1 INFORMATIONS GENERALES

Elève :	Nom :	Prénom :
Lieu de travail :	ETML / Avenue de Valmont 28b, 1010 Lausanne	
Client DevOps	Nom :	Prénom :
Dates de réalisation :	2ème trimestre	
Temps total :	~24 périodes / ~32 périodes (FPA/CFC)	

## 2 PROCÉDURE

- Tous les apprentis réalisent le projet sur la base d'un cahier des charges.
- Les apprentis travaillent par groupe de deux élèves.
- 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

**Todo App**

## 4 SUJET

Mise en place du déploiement CI/CD avec GitHub Action.

## 5 MATÉRIEL ET LOGICIEL À DISPOSITION

- PC ETML
- Accès à Internet
- Journal de travail
- Accès SSH sur une instance Ubuntu
- Compte GitHub
- Visual Studio Code

## 6 PRÉREQUIS

- Modules de programmation de base
- Modules de bases de données de base
- ICT-324 en cours

## 7 CAHIER DES CHARGES

### 7.1 Gestion de projet

1. La planification est à faire selon les instructions spécifiques de votre chef de projet.
2. Un journal de travail devra être rendu, format selon modalité de votre enseignant :
  - o La structure et la présentation sont claires et soignées ;
  - o Les sources, les fichiers, les répertoires, les commits, et autres sources d'informations concernées par le journal sont référencés ;
  - o L'état et les durées des tâches mentionnées sont précisés ;
  - o Toutes les activités planifiées, les aides extérieures, ainsi que les imprévus et les heures supplémentaires y sont mentionnés ;
  - o Les succès et les échecs sont mentionnés ;
  - o Le travail journalier et son appréciation critique, ainsi que les réflexions y figurent.

### 7.2 Qualité

1. Prouver que vous êtes digne de confiance lorsqu'on vous confie un projet
  - o Journal de travail à jour
  - o Pro-activité
    - **Poser des questions** au client
    - Faire des démonstrations
2. Repo **privé** sur github avec le nom : **cicd-todo-app** (partagé avec vos enseignants) utilisant un **.gitignore** concluant. Utilisant la structure suivante :

```
cicd-todo-app/
    ├── backend/  # Code serveur (API)
    ├── frontend/ # Interface utilisateur (Vue.js)
    ├── docs/     # Documentation technique et fonctionnelle (Markdown)
    ├── .gitignore
    ├── docker-compose.yml
    └── README.md # Présentation générale du projet
```

### 7.3 Fonctionnalités requises (du point de vue client)

L'application « Todo » fonctionne avec un frontend **Vue.js / TailwindCSS** responsive et un backend **Node.js / Express** qui fournit une API utilisée par le frontend.

La persistance des données de l'API est assurée par une base de données MySQL.

### 7.3.1 Spécificités CI/CD

Le workflow CI/CD sera mis en place avec « GitHub Action »

- a. Le déploiement sur l'infrastructure de « test » doit se déclencher lors de chaque « push » sur la branche principale « main ».
  - b. Un déploiement sur l'infrastructure de « production » doit être déclenché uniquement manuellement.
  - c. Les actions sur le frontend et sur le backend doivent inclure :
    - i. l'**audit** des package pour éviter toutes vulnérabilités des packages de dépendances
    - ii. le **linting** pour éviter des erreurs potentielles dans le code.
    - iii. Les **tests** (frontend avec les 3 explorateurs différent)
- Si vous modifiez les packages pour corriger des vulnérabilités ou le code pour corriger des erreurs linting ou bug (et donc faire passer les tests correspondants), ceci doit être clairement indiqué dans les commit de votre GIT ainsi que dans le rapport (corrections apportées). Les bugs ayant une priorité « haute » devront être corrigés en premier.*
- d. Le workflow sera optimisé pour exécuter certaines tâches en parallèle et utilisera des techniques de mise en cache pour optimiser le temps de déploiement.
  - e. Les déploiements de l'application de « test » et de « production » seront effectués sur un serveur dédier.
- f. Optionnel :
- i. Le code « coverage » des tests E2E du frontend – avec rapport sur codecov.io
  - ii. Le code « coverage » des tests unitaires du backend – avec rapport sur codecov.io
  - iii. Ajout de badges (statut GitHub Action et % code coverage avec codecov.io) au README.md
  - iv. Ajouter un workflow spécifique pour anticiper les conflits de fusion dans la branche principale « main », qui s'exécute lorsqu'une Pull Request (PR) est ouverte ou mise à jour, afin de vérifier si la PR peut être fusionnée sans conflit avec « main »
  - v. Ajouter un workflow spécifique pour un assignement initial des issues à un utilisateur pour le triage

## 7.4 Livrables

1. Une **release** GitHub comprenant le « code source » de l'application, le workflow « GitHubAction », fichier YAML correctement commenté. Les commentaires doivent apporter des informations pertinentes à sa compréhension.
2. Planification et suivi de projet type « KanBan » faite sur « GitHub Project ».
3. Un rapport au « format PDF » concernant les spécificités CI/CD contenant :
  - a. Introduction ;
  - b. Description de l'application (buts, techno concernées, fonctionnement, architecture) ;
  - c. Description des actions nécessaires (linting, audit, test, build, run) pour faire fonctionner le backend et le frontend incluant les spécificités selon les environnements de développement et de production ;
  - d. Description de toutes les étapes du workflow CI/CD (backend/frontend) ;
  - e. Le schéma (diagramme de flux – draw.io) initial du workflow complet (backend/frontend) exprimant aussi les conditions et dépendances ;
  - f. Rapport de déploiement (statut, conclusion) ;
  - g. Rapport de code « coverage » CI backend et frontend (capture d'écran) incluant des explications, statuts, et couverture ;
  - h. Le schémas CI/CD de « GitHub Action » mis en place (capture d'écran)
  - i. Chapitre explicatif de l'usage fait de l'IA dans ce projet ;
  - j. Conclusion.
4. Journal de travail individuel (selon modalité de votre enseignant)

---

## 8 Évaluation

1. Auto-évaluation et évaluation challengée par le client basé sur des éléments observables à 80%.
2. Pour la partie technique du workflow de déploiement : Deux questions techniques posées oralement par l'enseignant lors de la dernière séquence du projet.
3. Le recours à des outils en ligne d'intelligence artificielle (ex. : Chat GPT) est autorisé mais ne peut servir que **d'inspiration, support à la réalisation**. Chaque développeur doit être à tout moment en mesure d'expliquer le code de manière précise et convaincante. **En cas d'abus, l'évaluation du projet en tiendra.**