

324-TS2 – Portfolio CI/CD

Élément	Description
Objectifs	Le candidat réalise un portfolio sur les étapes de planification (Plan), développement (Dev), génération (Build), de test (Tests) et de déploiement (Deploy).
Durée estimée	~18 périodes (15 p. en classe + 2 p. à domicile)
A produire	Un portfolio (rapport), deux applications fonctionnelles.
Moyens d'aide	Tout
Pondération	50%: 69 points
Évaluation	<p><u>Calcul de la note :</u></p> <p>L'évaluation conduit à une note intermédiaire (au 1/10^e) calculée selon le barème fédéral.</p> <p>Cette note est ensuite moyennée selon la pondération pour former une note module arrondie aux 5/10^e.</p>
Présentation / défense	<p>Lors d'une présentation / défense de votre portfolio à votre enseignant, une évaluation sera faite :</p> <ul style="list-style-type: none">- durée 15 min (questions/réponses incluse)- pondération : 20% <p>L'évaluation conduit à une note intermédiaire (au 1/10^e) calculée selon le barème fédéral.</p> <p>Cette note est ensuite moyennée selon la pondération pour former une note module arrondie aux 5/10^e.</p>



1. Les attentes

Un portfolio est une collection de documents et de travaux montrant l'étendue des compétences d'une personne.

Ce portfolio doit contenir les livrables suivants :

- Un document PDF contenant :
 - o Introduction
 - o Analyse et planification initial
 - o Configuration de l'environnement de développement
 - o Journal des commits (datés)
 - o Développement et tests
 - o Planification / jobs / étapes CI/CD
 - o Résultats finaux et conclusion
- Deux applications (chacune d'un domaine différent)
 - o Ces deux applications sont indépendantes l'une de l'autre
 - o Le domaine ou le but d'une application est laissé libre, mais doit être validé par l'enseignant au plus tard 45mn après le début du projet. L'enseignant donnera son feu vert pour le début de l'examen. En cas de rejet (ou modification) du projet cette étape peut être prolongée, mais le temps total reste comptabilisé dans l'évaluation.
 - o Application imposée : Add-on Firefox – Todo App
 - o Exemple d'application à choix :
 - Node/VueJS
 - Python/Flask
 - .net
- Deux dépôts Github privé (un par application) partagé avec votre enseignant dès le début de l'épreuve. Les noms des dépôts doivent être :
 - o **324_portfolio_app1_<nom-prénom>**
 - o **324_portfolio_app2_<nom-prénom>**

Critères d'évaluation

Aspect	Critère	Description	Points
Plan			6
	Identification des exigences fonctionnelles	Préparation de User Stories - en utilisant « Issues » de GitHub	3
	Documentation continue	Le journal de travail (xls) est tenu à jour, le statut des tâches est actualisé.	3
Dev			21
	Configuration complète des outils de développement	Quiconque lirait le portfolio pourrait reproduire l'environnement de développement.	6
	Utilisation des outils intégrés	run, linter, audit, etc.	3
	Fonctionnalités principales	Deux MVP ont été préparés et sont fonctionnels (l'enseignant doit pouvoir exécuter vos applications)	6
	Gestion du code source	Utilisation efficace de Git avec commit fréquents et bien décrits.	3
	Organisation et lisibilité du code	Code propre, bien structuré et bien documenté	3
	Respect des normes de codage	Les normes ETML sont en vigueur	3
Build			12
	Utilisation des outils intégrés	build, ci, Docker, etc.	3
	Automatisation	Mise en place d'un workflow GitHub Actions	6
	Optimisation du workflow	Utilisation de mécanisme de cache / parallélisme	3

Aspect	Critère	Description	Points
Tests			18
	Cas de tests	Les tests couvrent les principales fonctionnalités, ils sont décrits et traçables.	6
	Résultat des tests	Les tests sont correctement exécutés et les résultats sont documentés.	3
	Automatisation	Mise en place d'un workflow GitHub Actions	6
	Optimisation du workflow	Utilisation de mécanisme de cache / parallélisme	3
Deploy			9
	Utilisation des outils propre à chaque application	Le déploiement est fait soit sur une plateforme comme <i>Developer Hub</i> de Firefox, soit sur un environnement serveur de préproduction	3
	Automatisation	Mise en place d'un workflow GitHub Actions	6
Rédaction			3
	Qualité du portfolio	Le document fourni doit répondre à une structure et à une mise en page soigné.	3
Total			69



2. Les applications

Il est possible (et souhaitable) de partir d'une base existante. Il faudra alors la rendre compatible avec les exigences DevOps. Ajouter les tests nécessaires pour couvrir les fonctionnalités principales ; adapter le code pour une utilisation multi-environnement (développement, production).

Toujours dans ce cas, votre documentation doit clairement établir la situation de départ (que fait déjà votre application par exemple) et quelle ajout / modification vous avez apporté.

Les applications doivent apporter un certain défi - par exemple une authentification et un stockage permanent des données « base de données, stockage JSON ».