

# TRAVAUX D'APPLICATION INDUSTRIELLE

## **FISE** Promo 2026

### Calendrier

- Appel à sujets auprès des enseignants - semaine du 29 septembre 2025
- Diffusion des sujets aux élèves - lundi 13 octobre 2025
- Retour des élèves attendu - jeudi 16 octobre 2025 avant 18h
- Démarrage des travaux - lundi 20 octobre 2025
- Date limite pour le rendu du travail - 27 janvier 2026 avant minuit
- Soutenances (30 min par groupe) - à fixer 28, 29 ou 30 janvier 2026

### Consignes

- Les élèves travaillent en binôme.
- Rendu : vous devrez rendre les éléments suivants
  - o Un rapport au format PDF (~ 8 pages). Le rapport doit être rédigé en utilisant le langage LateX selon le modèle proposé à cette adresse : <https://amubox.univ-amu.fr/s/92YK3XY2GWwzpes>
  - o Une archive au format zip contenant codes sources, fichier « readme », scripts d'installation, documentation ...
  - o Une courte vidéo (~ 3 min) de présentation de votre projet et de démonstration de votre application
  - o Tous les documents devront être déposés sur ce dépôt : <https://amubox.univ-amu.fr/s/zC5BRWJYgoExiEf> selon la nomenclature
    - TAI\_[NUMERO DU SUJET]\_rapport\_NOM1\_NOM2.pdf
    - TAI\_[NUMERO DU SUJET]\_code\_NOM1\_NOM2.zip
    - TAI\_[NUMERO DU SUJET]\_video\_NOM1\_NOM2.mp4
- Soutenance : la soutenance se fera sous la forme « présentation d'un poster »

Ces consignes seront complétées par des réunions et rendus intermédiaires fixés par l'encadrant du projet choisi.

**Émission des voeux pour l'attribution des sujets** : au plus tard le **16 octobre 2025 avant 18h**, les élèves forment des binômes et remplissent le tableau disponible ici :

<https://docs.google.com/spreadsheets/d/1h4oyWGJHISQ5p02U8bBPqU017gwGnI99yRc6jv79Cfg/edit?usp=sharing>

**Pour toutes les questions relatives aux sujets, vos contacts sont les tuteurs mentionnés.**

# 1 - PLATEFORME WEB D'APPRENTISSAGE DE LA PROGRAMMATION

**Encadrant(s)** Nicolas BAUDRU ([nicolas.baudru@univ-amu.fr](mailto:nicolas.baudru@univ-amu.fr))

Nicolas DURAND ([nicolas.durand@univ-amu.fr](mailto:nicolas.durand@univ-amu.fr))

## Mots clés

Application web,  
Base de données, ...



Exercice 2.1 : Minimum

Entrer une fonction `minimum` qui retourne le minimum d'une liste de nombres passé en paramètre.

• Sur l'entrée: [12, 1, 0, 7] votre solution doit renvoyer 1 .  
• Sur l'entrée: [-3, 7, 0, -12] votre solution doit renvoyer -12 .  
• Sur l'entrée: [-12, -1, -25, -1] votre solution doit renvoyer -25 .  
• Sur l'entrée: [0] votre solution doit renvoyer 0 .

```
def minimum(liste):
    for i in range(1, len(liste)):
        if liste[i] < liste[0]:
            liste[0] = liste[i]
    return liste
```

Votre fonction ne fait pas ce qui est attendu

• Sur l'entrée: [112, 1, 25, 71] votre programme a renvoyé 0 alors qu'il devrait 1 .  
• Sur l'entrée: [-1, 0, 1, 2] votre programme a renvoyé 0 alors qu'il devrait -1 .  
• Votre fonction ne renvoie pas ce qu'il est attendu pour ce moins une erreur invisible

Envoyer

## Description

L'objectif du projet est de développer une plateforme web permettant à des étudiants d'apprendre la programmation dans divers langages, via des séries d'exercices préparées au préalable par des enseignants. La priorité sera donnée aux langages C et Java, puis si possible C++ et Python.

La plateforme proposera à chaque étudiant un espace personnel sécurisé (compte utilisateur) dans lequel il trouvera :

- Des séries d'exercices thématiques à réaliser, préparés au préalable par les enseignants.
- Une zone d'édition de code afin d'écrire la solution proposée à un exercice.
- Un mécanisme d'aide à la résolution des exercices (indices, jeux de tests réussis/échoués, ...).
- Un tableau de bord avec des indicateurs de suivi et de progression.
- Un module d'interaction avec l'enseignant (Chat, ...).

La plateforme proposera aussi à chaque enseignant un espace personnel sécurisé (compte utilisateur) dans lequel il trouvera :

- Un tableau de bord avec des indicateurs de suivi et de performances de la promo et des étudiants.
- Un mode « édition » permettant la rédaction de nouveaux exercices à partir de modèles prédéfinis. Pour chaque exercice, l'enseignant devra fournir une solution, des jeux de tests publics (pour le mécanisme d'aide de l'étudiant), des tests privés (pour évaluation et/ou indicateur de performance).
- Un module d'interaction avec les étudiants (Chat, ...).

## Contraintes sur l'architecture logicielle

La plateforme web reposera évidemment sur une architecture combinant des technologies frontend (interface utilisateur) et backend (traitement des données et logique métier). La base de données permettra de stocker les données nécessaires au fonctionnement de l'application (utilisateurs, enseignants, exercices, ...) et les données utiles au suivi et à l'analyse du travail des étudiants.

L'architecture devra être modulaire, maximisant l'indépendance des divers services (tableau de bord, pages d'exercices, Chat, ...), et pensée pour permettre l'intégration rapide de nouveaux langages de programmation. La sécurité des données utilisateurs et des communications devra être garantie (authentification, confidentialité, parade aux attaques par injection SQL, ...).

La plateforme devra être la plus légère possible et son déploiement aisément. Elle devra dépendre le moins possible des contraintes matériels ou logiciels (dont le systèmes d'exploitation). Elle devra supporter la connexion simultanée de nombreux étudiants. Des tests en situation réelle seront à réaliser. Pour cela, nous vous demandons d'utiliser pour le frontend, le backend et la base de données, des langages, bibliothèques ou technologies classiques, efficaces, légers et simples. Prévoir une version « dockerisée » peut être une bonne idée.