**Plan de test – Fonction getUAbsence**

**1 - Identification du test**

**Nom** : Test de récupération des absences utilisateur  
**Numéro** : T4.1

**2 - Référence du module testé**

routes/psw.py – Fonction getUAbsence

**3 - Objectif du test**

Vérifier que la route retourne correctement toutes les absences justifiées/non justifiées **liées à des cours passés** d’un utilisateur donné.

Le test doit aussi vérifier :

* Que les données sont bien formatées
* Que les cas d’absence de données sont correctement gérés

**4 - Procédure du test**

* **Initialisation** : Remplir la BDD avec :
  + Un utilisateur
  + Des cours dans le passé (via EDTUtilisateur)
  + Des absences associées (via Absence)
* **Lancement** : Requête GET sur /psw/absence/{id\_utilisateur}
* **Observation** : Comparaison du code de retour et du corps de la réponse JSON

**5 - Résultats attendus**

| **N° Test** | **Condition** | **BDD Préparée** | **Résultat attendu** | **Statut attendu** |
| --- | --- | --- | --- | --- |
| 1 | Aucun cours passé | EDTUtilisateur vide pour cet utilisateur | 404 Absences non trouvées | Erreur |
| 2 | Cours passés mais aucune absence | EDTUtilisateur rempli, Absence vide | 404 Absences non trouvées | Erreur |
| 3 | Absences présentes | Cours passés + absences valides associées | 200 Liste des absences formatées (cours, horaire, motif...) | Succès |

**6 - Moyens à mettre en œuvre**

* **Logiciels** : FastAPI, SQLite/PostgreSQL, Pytest
* **Matériel** : Poste de développement avec environnement Python
* **Préconditions** :
  + Données cohérentes dans EDTUtilisateur (cours terminés uniquement)
  + Présence d’absences valides (valide = True)
  + Lien correct entre Absence.id\_edtutilisateur et les ID des cours