

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS	SESSION 2026
Administration des systèmes et des réseaux	

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 3
Nom, prénom : Belloum Nedjmeddine		N° candidat :
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>	Date : 20 / 06 / 2025
Organisation support de la réalisation professionnelle Mediaschool IRIS - Nice		
Intitulé de la réalisation professionnelle : Déploiement d'un serveur de messagerie multi-clients sécurisé — ClassCord		
Période de réalisation : 16/06/2025 au 20/06/2025 Lieu : Mediaschool – IRIS Nice		
Modalité : <input checked="" type="checkbox"/> Seul(e) <input type="checkbox"/> En équipe		
Compétences travaillées <input checked="" type="checkbox"/> Concevoir une solution d'infrastructure réseau <input checked="" type="checkbox"/> Installer, tester et déployer une solution d'infrastructure réseau <input checked="" type="checkbox"/> Exploiter, dépanner et superviser une solution d'infrastructure réseau		
Conditions de réalisation¹ (ressources fournies, résultats attendus) Ressources : Utilisation du repositories GitHub contenant la fiche mission et le script python de base classcord-server . Résultats attendus : Un serveur multi-clients - Base de données SQLite - Gestion de canaux de discussion – Interface d'administration côté serveur - Export des données - Logs d'audit - Support invité.		
Description des ressources documentaires, matérielles et logicielles utilisées² Ressources documentaires : README_server / README_client - ChatGPT / Copilot, - Documentation Python Matérielles et logicielles utilisées : VS Code, Oracle VirtualBox, Debian (OS Linux), SQLite3, UFW, Réseau local (NAT), Python3 et Docker.		
Modalités d'accès aux productions³ et à leur documentation⁴ Lien du repositories contenant le README et le projet : Documentation et réalisation du projet .		

¹ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Administration des systèmes et des réseaux » prévues dans le référentiel de certification du BTS SIO.

² Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

³ Conformément au référentiel du BTS SIO « *Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve.* ». Les éléments nécessaires peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁴ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation, par exemples schéma complet de réseau mis en place et configurations des services.

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

Le projet Classcord visait à concevoir et déployer un serveur de messagerie multi-clients écrit en Python, permettant la communication en temps réel entre plusieurs utilisateurs sur différents canaux (#général, #dev), avec une authentification sécurisée, une interface d'administration et une sauvegarde des données via SQLite.

server_classcord.py : Script principal du serveur multi-clients, gestion des utilisateurs, canaux, messages, statuts, et commandes admin.

test_client.py : Client de test interactif (enregistrement, connexion, mode invité, envoi de message).

admin_console : Interface de contrôle côté serveur (menu texte) : visualisation des connexions, export CSV, consultation des logs, envoi de messages admin.

SQLite : Base embarquée avec 2 tables : users et messages, gérée automatiquement au démarrage.

audit.log et classcord.log : Fichiers de logs pour suivre les connexions, actions, erreurs.

start_server.sh : Script de lancement rapide en tâche de fond avec redirection des logs.

Dockerfile & docker-compose.yml : Conteneurisation complète du serveur pour un déploiement portable.
admin_view.py Interface alternative pour visualiser les utilisateurs et messages (format tableau).

Authentification sécurisée (hash SHA-256) et mode invité (sans création de compte).

Stockage des échanges en base SQLite (utilisateurs & messages).

Diffusion des messages par canal : chaque client ne reçoit que les messages du canal courant.

Export CSV automatique des messages et utilisateurs avec date.

Journalisation avancée : audit des connexions, actions administrateur, erreurs.

Interface admin permettant :

- Visualiser les clients connectés,
- Gérer les canaux,
- Exporter les données,
- Envoyer des messages globaux,
- Arrêter le serveur proprement.