



RD Congo
Ministère de l'Enseignement Supérieur et Universitaire
Université Internationale de Nouvelles Technologies
Phone : +243992888777 ;
Email : contact@unitechcv.org, webssite.wwwunitechcv.org

TRAVAIL PRATIQUE DE GENIE LOGICIEL

OMMERLLAND DC/ Task Manager REPORT 2026

Par : NECHI ABBAMWINJA Omer-Serge

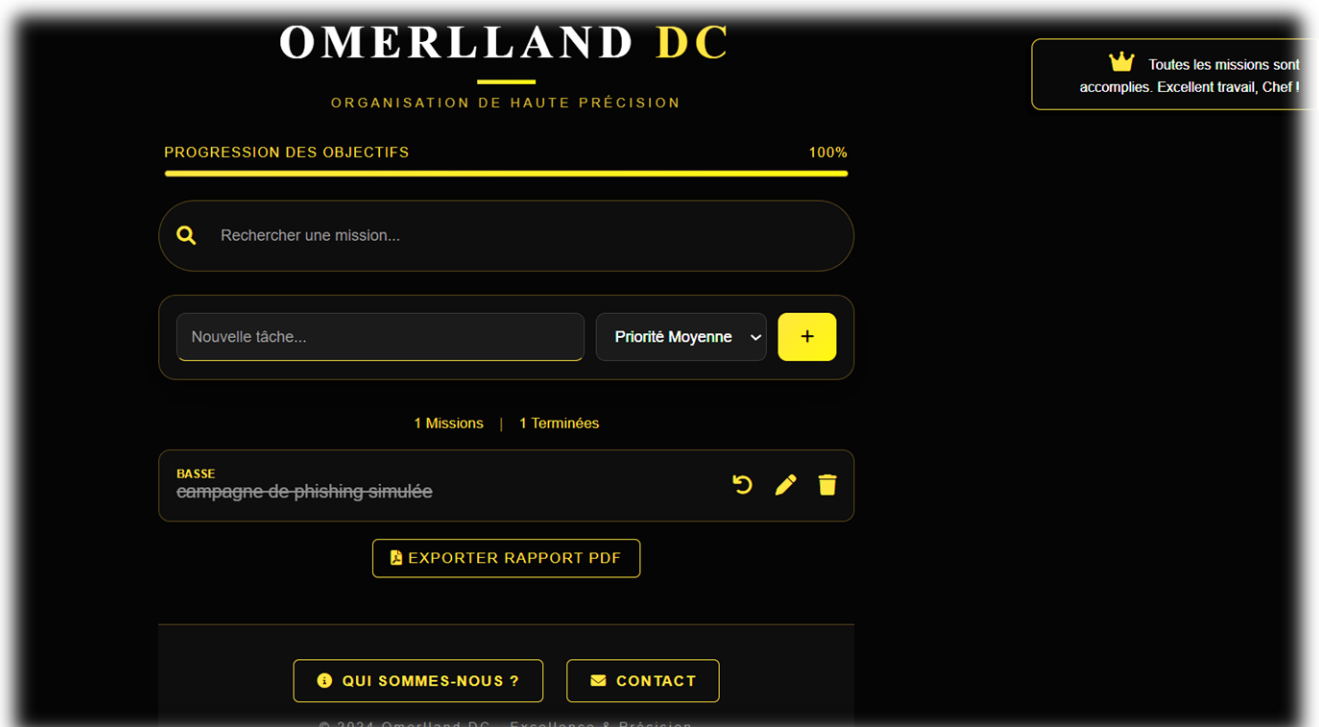
Lic3 Réseaux Informatiques et Télécommunications

Lien GitHub:

Lien Youtub :

Lien render :

Interface Omerlland DC / Task Manager



Rapport de Conception : Omerlland DC (Task Manager)

Avec une méthodologie rigoureuse de génie logiciel, **Omerlland DC** est un logiciel de gestion de tâche (To-Do List / Task Manager).

Ce document sert de feuille de route pour le développement d'une application web robuste, ergonomique et évolutive.

I. ANALYSE DES BESOINS

A. Besoins Fonctionnels (Ce que fait le logiciel)

- **Création** (Ajouter une tâche).
- **Modification** (Éditer le texte d'une tâche existante).
- **Suppression** (Supprimer une tâche de la liste).
- **Affichage** (Liste des tâches actuelles).
- **Marquer comme terminé** (Changer l'état d'une tâche).
- **Gestion des Utilisateurs** : Inscription, connexion et profil utilisateur.
- **Gestion des Tâches (CRUD)** : Créer, Lire, Mettre à jour et Supprimer des tâches.
- **Organisation** : Attribution de priorités (Haute, Moyenne, Basse), catégories (Tags) et dates d'échéance.
- **États de Progression** : Colonnes "À faire", "En cours", "Terminé".
- **Recherche et Filtres** : Moteur de recherche interne et filtrage par statut ou priorité.
- **Notifications** : Alertes pour les tâches arrivant à échéance.

B. Besoins Non Fonctionnels (Qualité du logiciel)

- **Ergonomie/Interface** : Thème clair Noir et Or, boutons d'action visibles.
- **Portabilité/Compatibilité** : Application web fonctionnant dans un navigateur (HTML/CSS/JS).
- **Performance** : Utilisation du stockage local (localStorage) pour persister les tâches même après rechargement de la page.
- **Performance** : Chargement rapide des données (optimisation JavaScript).
- **Responsive Design** : Interface parfaitement adaptée aux mobiles, tablettes et ordinateurs.
- **Sécurité** : Hachage des mots de passe et protection contre les injections SQL (via la base de données).
- **Expérience Utilisateur (UX)** : Interface intuitive et mode sombre/clair.

II. DIAGRAMME DE CAS D'UTILISATION

Ce diagramme définit les interactions entre l'utilisateur et le système.

- **Acteur** : Utilisateur (User)
- **Actions principales** :
 - S'authentifier (Connexion/Inscription).
 - Créer une tâche (Saisir titre, description, priorité).
 - Modifier/Mettre à jour le statut d'une tâche.
 - Supprimer une tâche.
 - Filtrer/Rechercher des tâches.
 - Paramétrer le profil (Changer le thème, nom).
 - Marqué une tâche comme terminée

1. Diagramme de Classes (Class Diagram)

Il représente la structure logique et les données du logiciel.

Extrait de code

```
classDiagram
class Utilisateur {
    +int id
    +string nom
    +string email
    +string motDePasse
    +login()
    +logout()
}

class Tache {
    +int id
    +string titre
    +string description
    +DateTime dateEcheance
    +string priorite
    +bool estTerminee
    +int categorield
    +creer()
    +modifier()
    +supprimer()
    +changerStatut()
}

class Categorie {
    +int id
    +string libelle
    +string couleur
}
```

Utilisateur et Tache : possède

Tache et Categorie : appartient à

2. Diagramme de Séquence

Voici le flux logique pour la **création d'une tâche** :

- 1) **L'utilisateur** remplit le formulaire sur l'Interface (UI).
- 2) **L'Interface** envoie les données au Contrôleur (JavaScript).
- 3) **Le Contrôleur** valide les données (ex : titre non vide).
- 4) **Le Contrôleur** demande à la Base de Données (DB) d'enregistrer.
- 5) **La DB** confirme l'enregistrement.
- 6) **L'Interface** met à jour la liste affichée à l'écran sans recharger la page.

3. Modèle Conceptuel de Données (Base de Données)

Pour le fichier SQL ou la structure JSON, voici les tables nécessaires :

Tableau	Colonnes principales
Utilisateurs	id, nom d'utilisateur, email, hachage_du_mot_de_passe, date_de_création
Tâches	id, user_id, titre, description, priorité, date_d'échéance, statut, category_id
Catégories	id, étiquette, code_couleur

Pourquoi cette architecture ?

Cette approche garantit qu'Omerlland **DC** ne soit pas juste une simple liste, mais un outil capable de gérer des milliers de tâches sans perdre en performance, grâce à une séparation claire entre les données (Model), l'affichage (View) et la logique (Controller).

III. TECHNIQUE D'ARCHITECTURE ET PILE

Pour un logiciel web moderne, nous utiliserons la stack suivante :

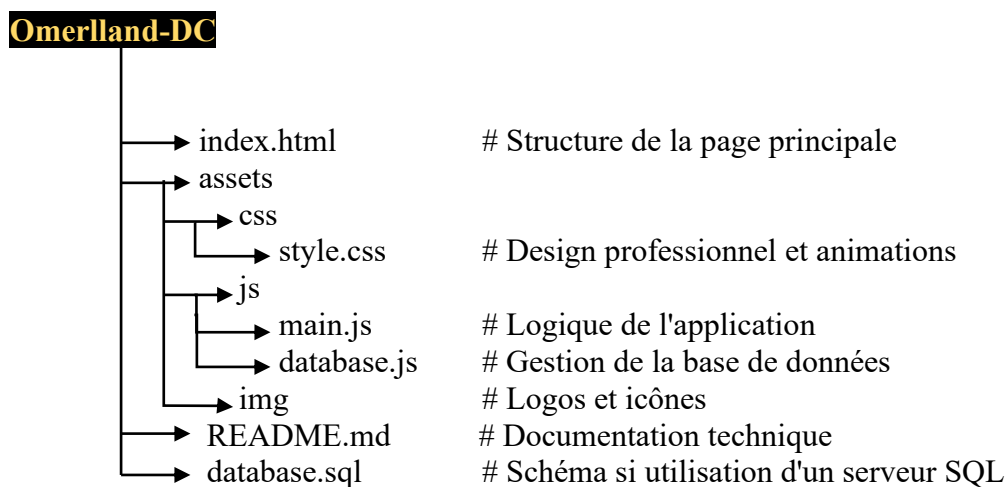
- **Frontend** : HTML5 (Structure), CSS3 (Stylisation avec Flexbox/Grid) et JavaScript (Logique dynamique).
- **Base de Données** : Pour un projet web standard, nous pouvons utiliser **IndexedDB** (local au navigateur) ou **Firebase/Supabase** pour une persistance réelle dans le cloud.
- **Documentation** : Un fichier README.md complet pour l'installation et l'utilisation.

IV. ÉTAPES DU CYCLE DE VIE (Génie Logiciel)

1. **Planification** : Définition du périmètre (le présent rapport).
2. **Conception (UI/UX)** : Création du maquettage (Wireframes).
3. **Développement (Coding)** : Écriture du code modulaire.
4. **Tests** : Vérification des bugs et de la réactivité.
5. **Déploiement** : Mise en ligne (GitHub Pages).

V. Structure du projet

Voici comment sera organisé notre répertoire de travail :



Architecture finale du flux de données

Voici un rappel visuel de la manière dont les données circulent maintenant dans notre logiciel professionnel :

1. **Saisie Intelligente** : Ajout avec niveau de priorité.
2. **Tri Algorithmique** : Les urgences (Haute priorité) remontent seules en haut de la pile.
3. **Recherche Instantanée** : Filtrage dynamique des missions.
4. **Persistance d'État** : Vos données survivent à la fermeture du navigateur.
5. **Reporting Professionnel** : Exportation PDF pour un usage administratif ou personnel.

Bref, Le logiciel est désormais complet et répond aux standards du génie logiciel moderne.