



DOCUMENT DE LIVRAISON – UniKinHub



Informations Projet

Champ	Valeur
Nom du projet	Projet Numéro 3 Fondation CCC
Auteur	Anthony Kamoto
Email	aanthonykamoto1@gmail.com
Organisation	Fondation Children Coding Club
Type de projet	Mini-Projet N°3
Date de livraison	6 Novembre 2025
Version	1.0.0
Licence	MIT



Accès au Code Source

Repository GitHub

URL complète : <https://github.com/AnthonyKamoto/UniKinHub>

```
# Cloner le repository
git clone https://github.com/AnthonyKamoto/UniKinHub.git
cd UniKinHub
```

Branches

- `main` - Branche principale (stable)
 - Tous les commits sont synchronisés et à jour
-



Description du Projet






UniKinHub est une **plateforme centralisée de diffusion d'actualités universitaires** conçue pour les étudiants des universités de Kinshasa. Le système permet la création, la modération et la diffusion d'informations académiques importantes avec un système de notifications en temps réel.

Objectifs





1. Centraliser les communications universitaires
2. Garantir la qualité des informations via modération
3. Cibler les actualités par programme et université
4. Notifier les étudiants en temps réel
5. Fournir des interfaces web et mobile intuitives

Fonctionnalités Principales





Pour les Étudiants

-  Fil d'actualités personnalisé selon le programme
-  Notifications push pour informations importantes
-  Interactions : likes sur les actualités
-  Recherche et filtres avancés
-  Application mobile native



Pour les Enseignants/Publiants



-  Création d'actualités avec éditeur
-  Upload d'images et fichiers joints
-  Statistiques de lecture
-  Ciblage par programme/université

Pour les Modérateurs

-  Système de modération complet
-  File d'attente des actualités
-  Commentaires de modération
-  Statistiques de modération

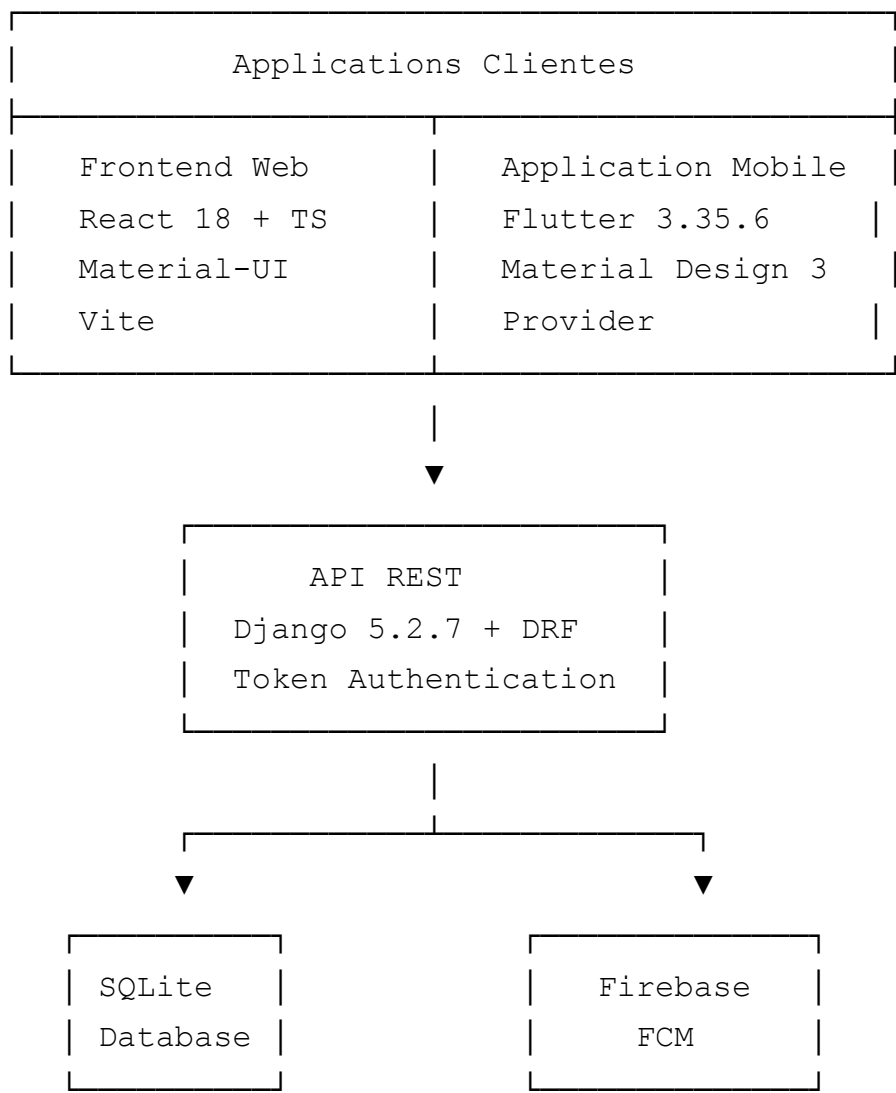
Pour les Administrateurs

-  Gestion complète des utilisateurs
-  Invalidation de contenu publié

-  Tableau de bord avec analytics
-  Configuration système

Architecture Technique

Stack Technologique



Technologies Utilisées

Composant	Technologie	Version
Backend	Django	5.2.7
API	Django REST Framework	3.15.2
Frontend	React	18.x
Langage Frontend	TypeScript	5.x

Composant	Technologie	Version
Mobile	Flutter	3.35.6
Base de données	SQLite	3.x
Authentification	Django Token Auth	-
Notifications	Firebase Cloud Messaging	-
UI Web	Material-UI	6.x
UI Mobile	Material Design 3	-
Build Tool	Vite	6.x
Package Manager	pnpm	9.x

Installation et Démarrage

Prérequis Système

- **OS:** Windows 10/11
- **Python:** 3.12 ou supérieur
- **Node.js:** 18 ou supérieur
- **pnpm:** 9.x
- **Flutter:** 3.35.6 (stable)
- **Git:** 2.x ou supérieur




Installation Rapide






Méthode 1 : Setup Automatique (Recommandé)

```
# 1. Ouvrir PowerShell dans le dossier du projet
cd UniKinHub

# 2. Exécuter le script de setup
.\setup.ps1
```

Le script `setup.ps1` effectue automatiquement :

-  Vérification des prérequis
-  Création environnement virtuel Python
-  Installation dépendances backend

-  Installation dépendances frontend
-  Installation dépendances mobile
-  Migrations de base de données
-  Création de données de test
-  Configuration Firebase (optionnel)

Méthode 2 : Installation Manuelle

Backend Django:

```
cd backend
python -m venv venv
.\venv\Scripts\Activate.ps1
pip install -r requirements.txt
python manage.py migrate
python manage.py loaddata initial_data
```

Frontend React:

```
cd frontend
pnpm install
```

Mobile Flutter:

```
cd mobile
flutter pub get
```

Démarrage des Serveurs

```
# Démarrer tous les serveurs
.\start-all.ps1
```

Serveurs lancés:

- Backend Django : <http://localhost:8000>
- Frontend React : <http://localhost:3001>
- Mobile Flutter : Émulateur Android

Arrêter tous les serveurs:

Comptes de Test

Comptes Disponibles

Rôle	Username	Mot de passe	Permissions
Administrateur	admin	admin123	Accès complet au système
Modérateur 1	moderateur1	password123	Modération d'actualités
Modérateur 2	moderateur2	password123	Modération d'actualités
Enseignant	enseignant1	password123	Création d'actualités
Étudiant UNIKIN	etudiant1	password123	Lecture + Interactions
Étudiant UPN	etudiant2	password123	Lecture + Interactions

 **Liste complète** : Voir fichier `COMPTES_UTILISATEURS.txt` pour tous les comptes (30+ utilisateurs)

Accès aux Interfaces

Interface Web:

- URL: <http://localhost:3001>
- Login avec un des comptes ci-dessus

Application Mobile:

- Lancer depuis VS Code ou Android Studio
- Login avec un des comptes ci-dessus

Admin Django:

- URL: <http://localhost:8000/admin>
- Login: admin / admin123

Test du Système

Workflow de Modération Complet

Étape 1 : Création d'Actualité (Publiant)

1. Se connecter avec : `enseignant1` / `password123`
2. Cliquer sur "Créer une actualité"
3. Remplir :
 - Titre : "Examen de fin de session"
 - Contenu : "Les examens auront lieu..."
 - Catégorie : Académique
 - Programme : Informatique
 - Importance : Importante
4. Soumettre
5. **Status** → `pending` (en attente de modération)

Étape 2 : Modération (Modérateur)

1. Se déconnecter
2. Se connecter avec : `modérateur1` / `password123`
3. Aller dans "Modération"
4. Voir la liste des actualités en attente
5. Cliquer sur l'actualité créée
6. Options :
 - **Approuver** : Publier directement
 - **Modifier et approuver** : Corriger puis publier
 - **Rejeter** : Refuser avec raison
7. Choisir "Approuver"
8. **Status** → `published`

Étape 3 : Consultation (Étudiant)

1. Se déconnecter
2. Se connecter avec : `etudiant1` / `password123`
3. Voir l'actualité dans le fil
4. Possibilités :
 - Liker l'actualité
 - Voir les détails
 - Filtrer par catégorie

Test des API

Login:

```
curl -X POST http://localhost:8000/api/auth/login/ \
-H "Content-Type: application/json" \
-d '{"username":"admin","password":"admin123"}'
```

Créer une actualité:

```
curl -X POST http://localhost:8000/api/news-api/ \
-H "Authorization: Token {votre-token}" \
-H "Content-Type: application/json" \
-d '{
  "draft_title": "Test API",
  "draft_content": "Contenu test",
  "category": 1,
  "importance": "medium"
}'
```

Lister actualités en attente:

```
curl http://localhost:8000/api/news-api/pending/ \
-H "Authorization: Token {token-moderateur}"
```

Approuver une actualité:

```
curl -X POST http://localhost:8000/api/news-api/42/approve/ \
-H "Authorization: Token {token-moderateur}" \
-H "Content-Type: application/json" \
-d '{"comment":"Approuvé"}'
```



Fonctionnalités Implémentées

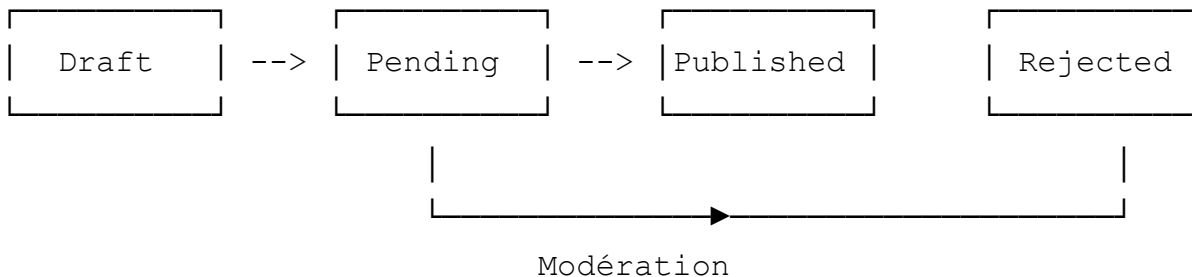


Fonctionnalités Complètes

- ☒ Système d'authentification (Token)
- ☒ RBAC avec 4 rôles (Admin, Modérateur, Publiant, Étudiant)
- ☒ Création d'actualités avec upload d'images
- ☒ Workflow de modération (approve/reject)
- ☒ File d'attente de modération

- ☒ Notifications email (template HTML)
- ☒ Notifications push Firebase (FCM)
- ☒ Filtrage par catégorie, programme, importance
- ☒ Recherche d'actualités
- ☒ Pagination des résultats
- ☒ Système de likes
- ☒ Dashboard administrateur (web + mobile)
- ☒ Écran paramètres mobile
- ☒ Responsive design (web)
- ☒ Material Design 3 (mobile)
- ☒ API REST complète
- ☒ Scripts PowerShell d'automatisation
- ☒ Documentation complète

Workflow Implémenté










Structure du Projet

UniKinHub/

```

|
├──  backend/ # Backend Django
|   ├──  news/ # App principale
|   |   ├── models.py # 8 modèles (News, Category, etc.)
|   |   ├── serializers.py # Serializers DRF
|   |   ├── views.py # Vues API générales
|   |   ├── views_moderation.py # ViewSet de modération
|   |   ├── permissions.py # 6 classes de permissions
|   |   ├── tasks.py # Tâches Celery
|   |   └── urls.py # 50+ endpoints
|   └──  news_system/ # Configuration Django
|       ├── settings.py # Configuration complète
|       ├── urls.py # URLs principales
|       └── celery.py # Config Celery
  
```

- | └─ db.sqlite3 # Base de données
- | └─ manage.py # CLI Django
- | └─ requirements.txt # Dépendances Python
- |
- | └─  frontend/ # Frontend React
 - | └─  src/
 - | └─  components/ # 20+ composants
 - | └─  pages/ # 15+ pages
 - | └─  services/ # Services API
 - | └─  contexts/ # Contextes React
 - | └─  theme/ # Configuration MUI
 - | └─ App.tsx # Composant principal
 - | └─ main.tsx # Point d'entrée
 - | └─ index.html
 - | └─ package.json
 - | └─ vite.config.ts
- |
- | └─  mobile/ # Application Flutter
 - | └─  lib/
 - | └─  models/ # 10+ modèles
 - | └─  screens/ # 25+ écrans
 - | └─  services/ # Services API
 - | └─  providers/ # State management
 - | └─  widgets/ # Widgets réutilisables
 - | └─  config/ # Configuration
 - | └─ main.dart # Point d'entrée
 - | └─ firebase_options.dart # Config Firebase
 - | └─ pubspec.yaml # Dépendances Flutter
 - | └─ android/ # Config Android
- |
- | └─  docs/ # Documentation
 - | └─ INSTALLATION.md # Guide installation
 - | └─ SMTP_CONFIGURATION.md # Config emails
- |
- | └─  screenshots/ # Captures d'écran
 - | └─  web/ # Screenshots web
 - | └─  mobile/ # Screenshots mobile
- |
- | └─  README.md # Documentation principale
- | └─  DEMARRAGE_RAPIDE.md # Guide démarrage rapide
- | └─  CHANGELOG.md # Historique modifications

└─		COMPTES_UTILISATEURS.txt	# Liste comptes test
└─		LICENSE	# Licence MIT
└─		setup.ps1	# Script installation
└─		start-all.ps1	# Démarrage serveurs
└─		stop-all.ps1	# Arrêt serveurs
└─		check-firebase-config.ps1	# Vérification Firebase



Captures d'Écran

Interface Web

Les captures d'écran sont disponibles dans le dossier [screenshots/web/](#) :

- Login / Inscription
- Fil d'actualités
- Création d'actualité
- Dashboard modération
- Dashboard administrateur
- Profil utilisateur

Application Mobile

Les captures d'écran sont disponibles dans le dossier [screenshots/mobile/](#) :

- Écran de connexion
- Fil d'actualités mobile
- Création d'actualité mobile
- Modération mobile
- Dashboard admin mobile
- Paramètres



Support et Contact

Informations de Contact





Auteur : Anthony Kamoto

Email : aaanthonykamoto1@gmail.com

GitHub : [@AnthonyKamoto](#)

Organisation : Fondation Children Coding Club

Resources Disponibles

-  **Documentation** : Voir dossier `docs/` et fichiers `.md`
 -  **Issues** : [GitHub Issues](#)
 -  **Repository** : github.com/AnthonyKamoto/UniKinHub
 -  **Email Support** : aanthonykamoto1@gmail.com
-

Licence

Ce projet est sous licence MIT. Voir fichier [LICENSE](#) pour plus de détails.

MIT License

Copyright (c) 2025 Anthony Kamoto


Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files...

Remerciements

- **Fondation Children Coding Club** pour la formation
 - **Communautés Django, React et Flutter** pour les ressources
 - **Universités de Kinshasa** (UNIKIN, UPN) pour le contexte du projet
-

 **PROJET LIVRÉ ET FONCTIONNEL**

Version 1.0.0 - Novembre 2025

Fait avec  pour les étudiants de Kinshasa

© 2025 Anthony Kamoto - Tous droits réservés