



RAPPORT DESCRIPTIF : PROJET PROGRAMMATION ORIENTÉE OBJET

ENSEIGNANT Dr SAWADOGO

Etudiantes

KAFANDO Aminata

KOUGOUINDIGA

Balguissa

PLAN

- I. Architecture et conception
 - II. Fonctionnalités
- III. Technologies utilises
- IV. Déploiement et installation
- V. Quelques interfaces du projet web

INTRODUCTION

L'essor du numérique et la généralisation des API ouvrent de nouvelles perspectives pour l'accès et l'exploitation de données en temps réel, y compris dans le domaine sportif. Dans le basket-ball professionnel, la disponibilité de statistiques détaillées permet d'enrichir l'expérience des passionnés, tout en offrant aux analystes des outils pour explorer la performance des équipes et des joueurs.

C'est dans ce contexte qu'a été développé le projet **NBA Analytics**, une application web exploitant l'API balldontlie.io pour offrir une visualisation claire et interactive des données de la NBA. Le projet combine une architecture logicielle fondée sur la programmation orientée objet avec une interface utilisateur moderne, afin de proposer un outil à la fois fiable, ergonomique et évolutif.

I. Architecture et Conception

1. Architecture Orientée Objet

Le projet respecte parfaitement les principes de la POO avec une architecture modulaire :

Classes principales (8 classes)

- ✓ User : Gestion des utilisateurs (hérite de `UserMixin`)
- ✓ Team : Gestion des équipes NBA
- ✓ Player : Gestion des joueurs
- ✓ Game : Gestion des matchs
- ✓ PlayerPerformance : Gestion des performances individuelles
- ✓ Notification : Système de notifications
- ✓ Statistics : Classe utilitaire pour les calculs statistiques
- ✓ BalldontlieService : Service d'intégration API

Relations orientées objet :

- ✓ Héritage : `User` hérite de `UserMixin` (Flask-Login)
- ✓ Association: `User` ↔ `Team` (relation many-to-many via `user_team_association`)
- ✓ Agrégation: 'Team' → 'Player' (one-to-many), 'Player' → 'PlayerPerformance'
- ✓ Composition : `Statistics` est composé dans `Player` et `PlayerPerformance`

II. Fonctionnalités

1. Fonctionnalités Techniques

- Système d'Authentification
 - ✓ Méthodes : Connexion par email/mot de passe
 - ✓ Sécurité : Hachage des mots de passe avec `werkzeug. Security`
 - ✓ Sessions : Gestion via Flask-Login
 - ✓ Rôles : user, admin, manager
- Gestion des Données NBA
 - ✓ Source : API balldontlie.io (gratuite et fiable)

- ✓ Synchronisation : Données réelles des équipes, joueurs et matchs NBA
- ✓ Mise à jour : Système de synchronisation automatique

❖ Interface Utilisateur

- ✓ Framework CSS: Bootstrap 5 (responsive design)
- ✓ Icônes : Font Awesome
- ✓ JavaScript : Vanilla JS pour l'interactivité
- ✓ Templates : 13 templates HTML modulaires

Base de Données

- ✓ ORM : SQLAlchemy
- ✓ SGBD: SQLite (development)
- ✓ Relations: One-to-many, Many-to-many
- ✓ Migrations : Gestion automatique des schémas

2. Fonctionnalités Utilisateur

Dashboard Principal

- ✓ Vue d'ensemble des statistiques NBA
- ✓ Accès rapide aux équipes et joueurs
- ✓ Notifications en temps réel
- ✓ Interface responsive et moderne

Gestion des Équipes

- ✓ Liste complète des équipes NBA
- ✓ Détails par équipe (ville, conférence, division)
- ✓ Joueurs par équipe
- ✓ Historique des matchs

Gestion des Joueurs

- ✓ Profils détaillés (taille, poids, position)
- ✓ Statistiques moyennes (points, passes, rebonds)
- ✓ Filtres par équipe et position
- ✓ Ajout de nouveaux joueurs

Suivi des Matchs

- ✓ Calendrier des matchs NBA
- ✓ Scores en temps réel
- ✓ Performances des joueurs
- ✓ Navigation par date

Système de Notifications

- ✓ Notifications internes
- ✓ Types: info, success, warning, error
- ✓ Marquer comme lu
- ✓ Historique des notifications

Intégration API

✓ Service balldontlie.io

III. Technologies Utilisées

- 1. Backend
 - ❖ Python 3.8+ : Langage principal
 - ❖ Flask 2.3.3: Framework web
 - ❖ SQLAlchemy 3.0.5: ORM
 - ❖ Flask-Login 0.6.3: Authentication
 - requests 2.31.0: Appels API HTTP
- 2. Frontend
 - Bootstrap 5: Framework CSS
 - Font Awesome: Icônes
 - Vanilla JavaScript : Interactivité
 - ❖ HTML5/CSS3 : Structure et style
- 3. Base de Données
 - ❖ SQLite : SGBD léger (développement)
 - ❖ SQLAlchemy : ORM Python
- 4. Déploiement
 - ❖ Gunicorn 21.2.0 : Serveur WSGI
 - python-dotenv 1.0.0 : Variables d'environnement

IV. Déploiement et Utilisation

- 1. Installation
 - Bash
 - ✓ Cloner le projet

Git clone https://github.com/KAFANDO-AMINATA/Projet-POO Cd Projet-POO/code

- ✓ Installer les dépendances pip install -r requirements.txt
- Lancer l'application python main.py
- 2. Accès
 - URL: http://localhost:5000
 - ❖ Admin: admin@nba-analytics.com / admin123
 - Port : 5000 (configurable)
- V. Quelques interfaces

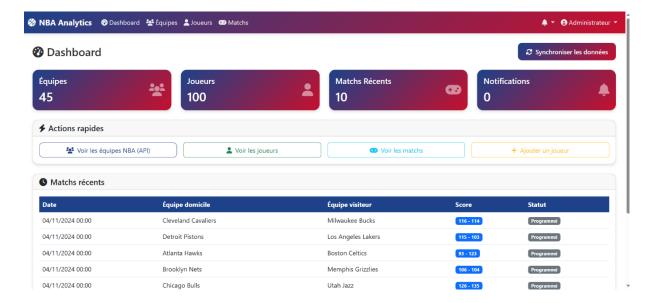
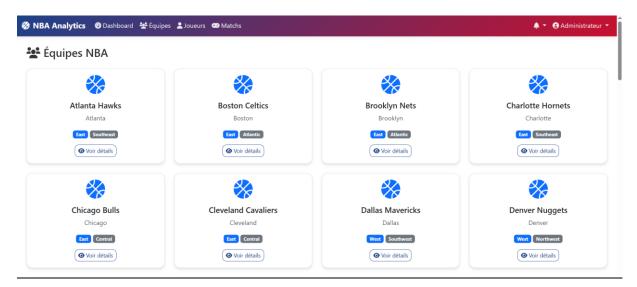
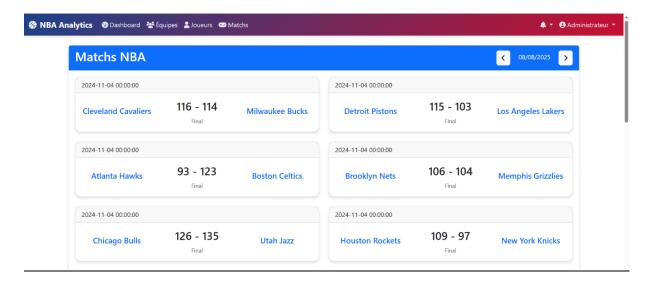


Tableau de bord



Interface de gestion des equipes



Interface de gestion des matchs

CONCLUSION

La réalisation de l'application **NBA Analytics** m'a permis de consolider mes compétences en programmation orientée objet, en concevant et structurant huit classes interconnectées par héritage, agrégation et composition. J'ai appris à concevoir une architecture modulaire et maintenable, à intégrer une API de manière fiable, et à appliquer des bonnes pratiques de sécurité. Ce projet m'a également permis de développer une interface utilisateur moderne et responsive, renforçant ainsi ma maîtrise du développement web complet et professionnel.