

markdown

🎓 i-Learn - Plateforme d'Apprentissage en Ligne

```
![Next.js](https://img.shields.io/badge/Next.js-15-black)
![Node.js](https://img.shields.io/badge/Node.js-18+-green)
![TypeScript](https://img.shields.io/badge/TypeScript-5-blue)
![MySQL](https://img.shields.io/badge/MySQL-8.0-blue)
![License](https://img.shields.io/badge/License-MIT-yellow)
```

i-Learn est une plateforme éducative intelligente permettant aux établissements scolaires de gérer leurs ressources pédagogiques et d'offrir un environnement d'apprentissage virtuel adapté à chaque niveau (primaire, secondaire, supérieur).

📸 Captures d'écran

> ⚠ **Note :** Ajoute tes captures d'écran dans le dossier `docs/screenshots/`

Page d'accueil	Dashboard Admin	Dashboard Étudiant
----- ----- -----		
![Homepage](./docs/screenshots/homepage.png)	![Admin](./docs/screenshots/admin-dashboard.png)	![Student](./docs/screenshots/student-dashboard.png)

✨ Fonctionnalités

🔒 **Authentification & Gestion des utilisateurs**

- Inscription avec validation des données
- Connexion sécurisée (JWT + bcrypt)
- 3 rôles : **Étudiant**, **Enseignant**, **Administrateur**
- 3 niveaux d'études : **Primaire**, **Secondaire**, **Supérieur**
- Gestion des permissions par rôle

📄 **Gestion des ressources pédagogiques**

- Upload de fichiers (PDF, DOC, PPT, Vidéos, Archives)
- 6 types de ressources : `DOCUMENT`, `VIDEO`, `ARTICLE`, `ARCHIVE`, `SIMULATION`, `LINK`
- Filtrage par niveau, matière et type
- Recherche par titre, description ou sujet
- Compteur de vues automatique

- Visibilité public/privé

⚑ **Dashboards personnalisés**

- **Étudiant** : Dashboard adapté à son niveau (Primaire, Secondaire, Supérieur)
- **Enseignant** : Upload de ressources, gestion de contenu
- **Administrateur** : Gestion complète des utilisateurs et ressources

🌐 **Pages informatives**

- Page d'accueil avec présentation
- À propos
- Contact
- Établissements partenaires
- Pages par niveau d'études

🛡 Stack Technique

Frontend

```
bash
Next.js 15 (App Router) # Framework React
React 19                 # Bibliothèque UI
TypeScript 5              # Typage statique
Tailwind CSS              # Framework CSS
Lucide React              # Icônes
```

Backend

bash

```
Node.js 18+               # Runtime
Express                   # Framework web
TypeScript 5              # Typage statique
Sequelize 6               # ORM MySQL
MySQL 8.0                 # Base de données
JWT                       # Authentification
Bcrypt                    # Hash des mots de passe
Multer                   # Upload de fichiers
```

Prérequis

Logiciel	Version	Lien
Node.js	18+	https://nodejs.org/
MySQL	8.0+	https://www.mysql.com/
Git	2.30+	https://git-scm.com/
VS Code	-	https://code.visualstudio.com/

Installation (5 minutes)

1 Cloner le projet

```
bash
git clone https://github.com/Mjunior237/ilearn.git
cd ilearn
```

2 Configurer la base de données MySQL

Installation de MySQL

```
<details> <summary><b>Windows</b></summary>
bash
# Télécharger MySQL Installer :
# https://dev.mysql.com/downloads/installer/

# Ou via Chocolatey :
choco install mysql

</details><details> <summary><b>macOS</b></summary>
bash
```

```
brew install mysql
brew services start mysql

</details><details> <summary><b>Linux (Ubuntu/Debian)</b></summary>
bash
sudo apt update
sudo apt install mysql-server
sudo systemctl start mysql
sudo mysql_secure_installation

</details>
```

🔧 Créer la base de données

```
bash
# Se connecter à MySQL
mysql -u root -p

# Exécuter ces commandes SQL :
CREATE DATABASE ilearn CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE USER 'ilearn_user'@'localhost' IDENTIFIED BY 'ilearn123';
GRANT ALL PRIVILEGES ON ilearn.* TO 'ilearn_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

3 Backend

```
bash
# Aller dans le dossier backend
cd ilearn-backend

# Installer les dépendances
npm install

# Créer le fichier de configuration
cp .env.example .env

# Éditer le fichier .env (adapter vos identifiants)
notepad .env    # Windows
nano .env       # Linux/Mac
```

.env

env

```
# Base de données MySQL  
DB_HOST=localhost  
DB_PORT=3306  
DB_NAME=ilearn  
DB_USER=ilearn_user  
DB_PASSWORD=ilearn123  
  
# JWT  
JWT_SECRET=changer_cette_clé_en_production_123456789
```

Serveur

```
PORT=5000  
NODE_ENV=development
```

bash

```
# Créer le dossier pour les uploads  
mkdir uploads
```

Démarrer le serveur

```
npm run dev
```

Backend lancé sur <http://localhost:5000>

4. Frontend

bash

```
# Nouveau terminal  
cd ilearn-frontend  
  
# Installer les dépendances  
npm install  
  
# Démarrer l'application  
npm run dev
```

Frontend lancé sur <http://localhost:3000>

📝 Guide de test rapide

1. Créer un administrateur

```
bash
# 1. Incription
# http://localhost:3000/register
# Email: admin@ilearn.com
# Mot de passe: admin123
# Rôle: Enseignant (modifié ensuite)

# 2. Se connecter à MySQL
mysql -u iLearn_user -p iLearn

# 3. Passer le rôle en ADMIN
UPDATE users SET role = 'ADMIN' WHERE email = 'admin@ilearn.com';

# 4. Vérifier
SELECT id, email, role FROM users;
```

2. Uploader une ressource (Admin)

1. Connectez-vous avec `admin@ilearn.com`
2. Accédez à `/admin/dashboard`
3. Cliquez sur "**Upload Resource**"
4. Remplissez le formulaire et sélectionnez un fichier

3. Tester avec un étudiant

1. Déconnectez-vous
2. Créez un compte avec rôle "Étudiant"
3. Connectez-vous
4. Accédez au dashboard étudiant adapté à votre niveau

Structure complète du projet

text

```
📦 ilearn
  └── ilearn-backend
      ├── src
      │   ├── config
      │   │   └── database.ts          # Configuration MySQL/Sequelize
      │   ├── models
      │   │   ├── User.model.ts       # Modèle utilisateur
      │   │   └── Resource.model.ts  # Modèle ressource
      │   ├── services
      │   │   ├── auth.service.ts     # Auth JWT
      │   │   └── resource.services.ts # CRUD ressources
      │   ├── middleware
      │   │   └── auth.middleware.ts  # Protection JWT
      │   ├── routes
      │   │   ├── auth.route.ts
      │   │   └── resource.route.ts
      │   ├── app.ts
      │   └── server.ts
      ├── uploads                      # Fichiers uploadés
      └── .env.example
      └── package.json

  └── ilearn-frontend
      └── app
          ├── (auth)
          │   ├── login
          │   └── register
          ├── dashboard
          │   ├── admin
          │   ├── student-primary
          │   ├── student-secondary
          │   └── student-superior
          ├── components
          │   ├── Header.tsx
          │   ├── DashboardHeader.tsx
          │   └── Footer.tsx
```

```

    |   |   └── page.tsx
    |   └── package.json
    |
    └── docs          # Documentation
        ├── ARCHITECTURE.md
        ├── SPECIFICATIONS.md
        └── diagrams
            ├── cas-utilisation.drawio
            ├── classes.drawio
            ├── sequence.drawio
            └── deploiement.drawio
    |
    ├── README.md      # Vous êtes ici
    └── .gitignore

```

🔑 API Reference

Authentification

Méthode	Endpoint	Description	Body
POST	/api/auth/register	Inscription	{ firstName, lastName, email, password, educationLevel, role }
POST	/api/auth/login	Connexion	{ email, password }

Ressources

Méthode	Endpoint	Description	Auth
GET	/api/resources	Liste toutes les ressources publiques	✗
GET	/api/resources/:id	Détail d'une ressource + incrémentale vues	✗

Méthode	Endpoint	Description	Auth
POST	/api/resources/upload	Upload un fichier	<input checked="" type="checkbox"/> ADMIN
DELETE	/api/resources/:id	Supprime une ressource	<input checked="" type="checkbox"/> ADMIN
PUT	/api/resources/:id	Modifie une ressource	<input checked="" type="checkbox"/> ADMIN

?

Dépannage

Problème	Solution
Error: connect ECONNREFUSED 127.0.0.1:3306	MySQL n'est pas démarré. Lance <code>mysql.server start</code> (Mac) ou <code>net start mysql</code> (Windows)
SequelizeDatabaseError: Table 'resources' doesn't exist	Les tables se créent automatiquement au démarrage. Redémarre le backend.
403 Accès refusé. Admin uniquement.	Vérifie que ton rôle en base est bien <code>ADMIN</code> (majuscules).
Upload failed: Fichier trop volumineux	Limite = 100MB. Réduis la taille du fichier.
CORS Error	Vérifie que le frontend tourne sur <code>http://localhost:3000</code> et le backend sur <code>http://localhost:5000</code> .

👥 Contributeurs

- **Mjunior237** - Développeur Full-Stack - [GitHub](#)

Contact

- **Email** : contact@ilearn.com
 - **GitHub Issues** : <https://github.com/Mjunior237/ilearn/issues>
-