Projet -Creation API(Node js/Express / MongoDB/Postman)

Énoncé du projet : Développement d'une API RESTful avec Node.js, Express et MongoDB

Date limite de rendu : avant le 30 novembre 2025 à 23h59

Contexte :

Vous êtes développeur backend au sein d'une jeune startup. Votre mission est de concevoir et développer une API RESTful permettant la gestion d'un système de données (ex : gestion d'une bibliothèque, d'un magasin, d'un site de réservation, etc.).
L'API doit suivre l'architecture MVC (Modèle – Vue – Contrôleur) et être testée à l'aide de Postman.

Objectifs pédagogiques:

- Maîtriser le développement backend avec Node.js et Express.js.
- Savoir interagir avec une base de données MongoDB via Mongoose.
- Comprendre et appliquer le modèle MVC.
- Concevoir et documenter une API RESTful.
- Tester les endpoints avec Postman.

Spécifications techniques :

1. Architecture du projet

Le projet doit suivre la structure suivante :

project/

|

├— app.js

├— package.json

├— /models

| └— (fichiers des schémas Mongoose)

|

├— /routes

| └— (fichiers des routes Express) eT controlleurs API

2. Fonctionnalités minimales

L'API doit gérer au moins 5 objets différents (collections MongoDB).
Exemples :

- Pour une bibliothèque : Livres, Auteurs, Utilisateurs, Emprunts, Catégories.
- Pour une école : Étudiants, Professeurs, Cours, Notes, Groupes.
- Pour un magasin : Produits, Clients, Commandes, Fournisseurs, Paiements.

Chaque objet doit avoir :

- Un modèle Mongoose.
- Un contrôleur avec les 4 opérations CRUD :
  - POST : ajouter un élément
  - GET : lister / obtenir un élément
  - PUT : modifier un élément
  - DELETE : supprimer un élément
- Une route Express associée.

3. Tests avec Postman

Les étudiants doivent :

- Créer une collection Postman contenant tous les endpoints testés.
- Fournir des captures d'écran montrant :
  - Les requêtes (GET, POST, PUT, DELETE)
  - Les réponses du serveur
  - Le code d'état HTTP (200, 201, 404, etc.)

4. Base de données

- Utiliser MongoDB Atlas ou une instance locale.
- Créer des collections correspondant aux objets définis.
- Ajouter des données d'exemple pour les tests.

Étapes de réalisation :

Étape 1 : Initialisation du projet

- Créer le projet Node.js avec npm init.
- Installer les dépendances nécessaires :
- npm install express mongoose nodemon cors body-parser
- Configurer app.js avec les middlewares de base.

Étape 2 : Connexion à MongoDB

- Créer un fichier /config/database.js pour la connexion.
- Tester la connexion avec Mongoose.

Étape 3 : Création du modèle MVC

- Définir les modèles (models/) avec les schémas Mongoose.
- Créer les contrôleurs (Routes/) avec les fonctions CRUD.
- Définir les routes (routes/) associées.

Étape 4 : Tests Postman

- Tester chaque route CRUD.
- Sauvegarder les requêtes dans une collection Postman.
- Faire des captures d'écran.

Étape 5 : Documentation et rapport

- Rédiger un rapport PDF contenant :
    - Introduction du projet
    - Description de l'architecture
    - Explication de chaque dossier/fichier
    - Liste et description des routes (méthodes, URL, exemple JSON)
    - Captures d'écran Postman

- Conclusion sur le fonctionnement du projet

Livrables attendus :

1. Code source

- Un dossier compressé .zip ou .rar contenant :
  - Le projet complet Node.js (sans le dossier node_modules).

2. Rapport PDF

- Minimum 5 pages.
- Inclure :
  - Schéma d'architecture MVC
  - Explication du code
  - Tests Postman avec captures d'écran
  - Résumé et conclusion

Critères d'évaluation (suggestion) :

Critère	**Points**
Structure MVC respectée	1
Connexion et configuration MongoDB	1
5 objets correctement implémentés (CRUD complet)	1
Tests Postman (complets et fonctionnels)	2
Qualité du code (lisibilité, organisation)	1
Rapport PDF (clarté, structure, captures d'écran)	1
Total	**7 points**