Backend

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\config\database.js

require('dotenv').config();

const { Sequelize } = require('sequelize');

const sequelize = new Sequelize(

  process.env.DB\_NAME,

  process.env.DB\_USER,

  process.env.DB\_PASSWORD,

  {

    host: process.env.DB\_HOST,

    dialect: 'mysql',

    logging: false, // Désactiver les logs SQL pour éviter le spam dans la console

  }

);

sequelize.authenticate()

  .then(() => console.log('✅ Connexion à MySQL réussie avec Sequelize !'))

  .catch(err => console.error('❌ Erreur de connexion à MySQL :', err));

module.exports = sequelize;

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\models

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\models\associations.js

const Produit = require('./Produit');

const Categorie = require('./Categorie');

// Déclaration des relations

Categorie.hasMany(Produit, { foreignKey: 'categorie\_id' });

Produit.belongsTo(Categorie, { foreignKey: 'categorie\_id' });

module.exports = { Produit, Categorie };

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\models\Categorie.js

const { DataTypes } = require('sequelize');

const sequelize = require('../config/database');

const Categorie = sequelize.define('Categorie', {

    id: {

        type: DataTypes.INTEGER,

        primaryKey: true,

        autoIncrement: true

    },

    nom: {

        type: DataTypes.STRING,

        allowNull: false,

        unique: true

    },

    image: {

        type: DataTypes.STRING,

        allowNull: true,

      }

}, {

    tableName: 'categories',

    timestamps: false // Désactive createdAt / updatedAt si non utilisés

});

module.exports = Categorie;

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\models\commandes.js

const { DataTypes } = require('sequelize');

const sequelize = require('../config/database');

const Commande = sequelize.define('Commande', {

    id: {

        type: DataTypes.INTEGER,

        primaryKey: true,

        autoIncrement: true

    },

    utilisateur\_id: {

        type: DataTypes.INTEGER,

        references: {

            model: 'utilisateurs',

            key: 'id'

        }

    },

    montant\_total: {

        type: DataTypes.FLOAT,

        allowNull: false

    },

    statut: {

        type: DataTypes.ENUM('en attente', 'expédiée', 'livrée'),

        defaultValue: 'en attente'

    },

    date\_commande: {

        type: DataTypes.TIMESTAMP,

        defaultValue: DataTypes.NOW

    }

}, {

    tableName: 'commandes',

    timestamps: false

});

module.exports = Commande;

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\models\Produit.js

const { DataTypes } = require('sequelize');

const sequelize = require('../config/database');

const Produit = sequelize.define('Produit', {

    id: {

        type: DataTypes.INTEGER,

        primaryKey: true,

        autoIncrement: true

    },

    nom: {

        type: DataTypes.STRING,

        allowNull: false

    },

    description: {

        type: DataTypes.TEXT,

        allowNull: false

    },

    prix: {

        type: DataTypes.FLOAT,

        allowNull: false

    },

    stock: {

        type: DataTypes.INTEGER,

        allowNull: false

    },

    categorie\_id: {

        type: DataTypes.INTEGER,

        references: {

            model: 'categories',

            key: 'id'

        }

    }

}, {

    tableName: 'produits',

    timestamps: false

});

module.exports = Produit;

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\models\utilisateurs.js

const { DataTypes } = require('sequelize');

const sequelize = require('../config/database');

const Utilisateur = sequelize.define('Utilisateur', {

    id: {

        type: DataTypes.INTEGER,

        primaryKey: true,

        autoIncrement: true

    },

    prenom: {

        type: DataTypes.STRING,

        allowNull: false

    },

    nom: {

        type: DataTypes.STRING,

        allowNull: false

    },

    email: {

        type: DataTypes.STRING,

        allowNull: false,

        unique: true

    },

    mot\_de\_passe: {

        type: DataTypes.STRING,

        allowNull: false

    },

    role: {

        type: DataTypes.ENUM('client', 'admin'),

        defaultValue: 'client'

    },

    date\_creation: {

        type: DataTypes.DATE,

        defaultValue: DataTypes.NOW

    }

}, {

    tableName: 'utilisateurs',

    timestamps: false // Si tu ne veux pas de createdAt / updatedAt

});

module.exports = Utilisateur;

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\node\_modules

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\public\images

Avec 1 fichier hero.jpg 2 sous dossier produits et catégories

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\routes

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\routes\auth.js

const express = require('express');

const router = express.Router();

const bcrypt = require('bcrypt');

const Utilisateur = require('../models/utilisateurs');

// POST /api/connexion

router.post('/connexion', async (req, res) => {

  const { email, mot\_de\_passe } = req.body;

  try {

    const utilisateur = await Utilisateur.findOne({ where: { email } });

    if (!utilisateur) {

      console.log('❌ Email non trouvé :', email);

      return res.status(401).json({ message: 'Email ou mot de passe incorrect' });

    }

    const estValide = await bcrypt.compare(mot\_de\_passe, utilisateur.mot\_de\_passe);

    console.log('✅ Saisi :', mot\_de\_passe);

    console.log('📦 En BDD :', utilisateur.mot\_de\_passe);

    console.log('Résultat comparaison :', estValide);

    if (!estValide) {

      return res.status(401).json({ message: 'Email ou mot de passe incorrect' });

    }

    const { mot\_de\_passe: \_, ...utilisateurSansMotDePasse } = utilisateur.toJSON();

    res.status(200).json(utilisateurSansMotDePasse);

  } catch (err) {

    console.error('Erreur connexion :', err);

    res.status(500).json({ message: 'Erreur serveur' });

  }

});

// ✅ POST /api/inscription

router.post('/inscription', async (req, res) => {

  const { nom, prenom, email, mot\_de\_passe, telephone } = req.body;

  try {

    const existant = await Utilisateur.findOne({ where: { email } });

    if (existant) {

      return res.status(400).json({ message: "Cet email est déjà utilisé." });

    }

    if (!nom || nom.length < 2 || !prenom || prenom.length < 2 || !email || !mot\_de\_passe || mot\_de\_passe.length < 8) {

      return res.status(400).json({ message: "Champs invalides ou incomplets." });

    }

    const hashed = await bcrypt.hash(mot\_de\_passe, 10);

    const nouvelUtilisateur = await Utilisateur.create({

      nom,

      prenom,

      email,

      mot\_de\_passe: hashed,

      telephone

    });

    console.log('✅ Utilisateur créé :', nouvelUtilisateur.email);

    res.status(201).json({ message: "Compte créé avec succès." });

  } catch (err) {

    console.error('Erreur inscription :', err);

    res.status(500).json({ message: "Erreur serveur. Veuillez réessayer." });

  }

});

module.exports = router;

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\routes\categories.js

const express = require('express');

const router = express.Router();

const Categorie = require('../models/Categorie');

// GET /categories

router.get('/categories', async (req, res) => {

  try {

    const categories = await Categorie.findAll();

    res.status(200).json(categories);

  } catch (err) {

    console.error('❌ Erreur récupération des catégories :', err);

    res.status(500).json({ message: 'Erreur serveur' });

  }

});

// POST /categorie

router.post('/categorie', async (req, res) => {

  try {

    const { nom } = req.body;

    const categorie = await Categorie.create({ nom });

    res.status(201).json(categorie);

  } catch (err) {

    console.error('❌ Erreur création catégorie :', err);

    res.status(500).json({ message: 'Erreur serveur' });

  }

});

module.exports = router;

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\routes\panier.js

const express = require('express');

const router = express.Router();

const { Commande, DetailCommande } = require('../models');

const { Op } = require('sequelize');

// POST /api/panier

router.post('/', async (req, res) => {

  const { utilisateur\_id, produit\_id, taille, quantite, prix\_unitaire } = req.body;

  try {

    // 1. Vérifie s’il a déjà une commande "panier"

    let commande = await Commande.findOne({

      where: {

        utilisateur\_id,

        statut: 'panier'

      }

    });

    // 2. Sinon, créer une commande vide "panier"

    if (!commande) {

      commande = await Commande.create({

        utilisateur\_id,

        statut: 'panier',

        date\_creation: new Date()

      });

    }

    // 3. Cherche si le produit/taille existe déjà dans la commande

    const existant = await DetailCommande.findOne({

      where: {

        commande\_id: commande.id,

        produit\_id,

        taille

      }

    });

    if (existant) {

      // 4. Si déjà présent, on incrémente la quantité

      existant.quantite += quantite;

      await existant.save();

    } else {

      // 5. Sinon on ajoute une nouvelle ligne

      await DetailCommande.create({

        commande\_id: commande.id,

        produit\_id,

        taille,

        quantite,

        prix\_unitaire

      });

    }

    res.status(200).json({ message: 'Produit ajouté au panier.' });

  } catch (err) {

    console.error('Erreur ajout panier :', err);

    res.status(500).json({ message: "Erreur serveur." });

  }

});

module.exports = router;

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\routes\produits.js

const express = require('express');

const router = express.Router();

const { Op } = require('sequelize');

const Produit = require('../models/Produit');

const Categorie = require('../models/Categorie');

// ✅ GET /produits?categorie=nom

router.get('/', async (req, res) => {

  const nomCategorie = req.query.categorie;

  try {

    const categorie = await Categorie.findOne({

      where: { nom: { [Op.like]: `%${nomCategorie}%` } }

    });

    if (!categorie) {

      return res.status(404).json({ message: 'Catégorie introuvable' });

    }

    const produits = await Produit.findAll({

      where: { categorie\_id: categorie.id },

      attributes: ['id', 'nom', 'description', 'prix', 'stock', 'image']

    });

    res.json(produits);

  } catch (err) {

    console.error('❌ Erreur récupération produits :', err);

    res.status(500).json({ message: 'Erreur serveur' });

  }

});

// ✅ GET /produits/:id

router.get('/:id', async (req, res) => {

  try {

    const produit = await Produit.findByPk(req.params.id, {

  attributes: ['id', 'nom', 'description', 'prix', 'stock', 'image', 'categorie\_id']

});

    if (!produit) {

      return res.status(404).json({ message: 'Produit non trouvé' });

    }

    res.json(produit);

  } catch (err) {

    console.error('❌ Erreur récupération produit par ID :', err);

    res.status(500).json({ message: 'Erreur serveur' });

  }

});

module.exports = router;

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\.env

DB\_NAME=ecommerce

DB\_USER=root

DB\_PASSWORD=

DB\_HOST=localhost

DB\_DIALECT=mysql

PORT=5000

JWT\_SECRET=supersecretkey

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\hash.js

const bcrypt = require('bcrypt');

const motDePasseClair = 'premiervraicomptetest';

bcrypt.hash(motDePasseClair, 10)

  .then(hash => {

    console.log('🔐 Mot de passe hashé :', hash);

  })

  .catch(err => console.error(err));

C:\Users\cohen\Desktop\Ethan\HT PARIS\ecommerce\backend\index.js

require('dotenv').config();

const express = require('express');

const cors = require('cors');

const path = require('path');

const authRoutes = require('./routes/auth');

const panierRoutes = require('./routes/panier');

const sequelize = require('./config/database');

const produitsRoutes = require('./routes/produits');

const categoriesRoutes = require('./routes/categories');

const app = express();

// Middlewares

app.use(express.json());

app.use(cors());

// Images accessibles depuis le frontend

app.use('/images', express.static(path.join(\_\_dirname, 'public/images')));

// Routes

app.use('/produits', produitsRoutes);

app.use('/', categoriesRoutes);

app.use('/api', authRoutes);

app.use('/api/panier', panierRoutes);

// Connexion à la base de données

sequelize.sync()

  .then(() => console.log('✅ Base de données synchronisée avec Sequelize !'))

  .catch(err => console.error('❌ Erreur de synchronisation :', err));

const { Produit, Categorie } = require('./models/associations');

// Lancement du serveur

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => {

  console.log(`🚀 Serveur lancé sur http://localhost:${PORT}`);

});

Puis 2 packages Json à la racine