ENSA Khouribga

Compte Rendu TP 2

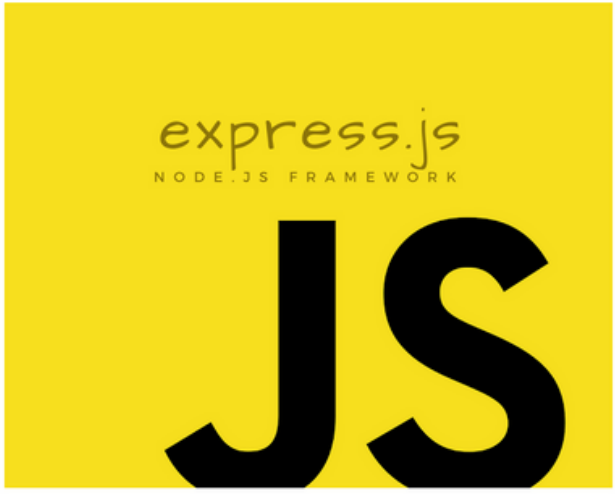
Express.js

Made By

EL OUAKYL EL MEHDI

Suprevised By

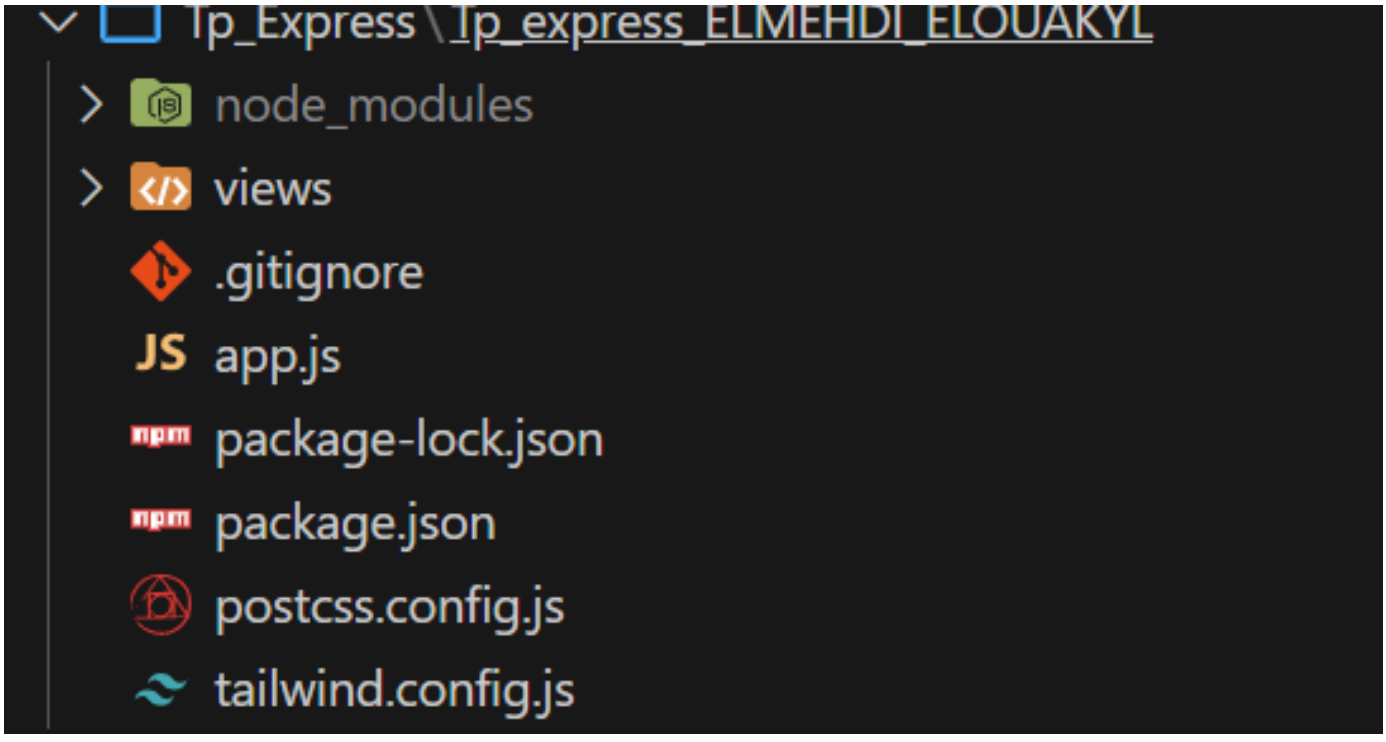
Pr. Amal OURDOU



CONTEXT GENERALE

1. L'objectif de ce travail pratique est d'intégrer l'ensemble des concepts abordés dans les précédents TPs afin de réaliser une application web complète qui inclut :
2. une page d'inscription (register) et une page de connexion (login) créées avec Pug,
3. une base de données MongoDB destinée à enregistrer les informations des utilisateurs,
4. l'utilisation de Passport.js pour gérer le processus d'authentification,
5. une redirection vers la page books affichant une liste locale de livres après une connexion réussie,
6. et l'emploi de Tailwind CSS pour la mise en forme des interfaces.

PARTIE I : ARCHITECTURE DU PROJET



PARTIE 2: IMPLÉMENTATION ET EXPLICATIONS

a) Connexion à MongoDB

Le serveur se connecte à la base locale express_auth_tp :

```
mongoose.connect("mongodb://127.0.0.1:27017/express_tp")
  .then(() => console.log("Connecté à MongoDB"))
  .catch(err => console.error(err));
```

b) Modèle utilisateur

Chaque utilisateur possède un nom d'utilisateur et un mot de passe :

```
const userSchema = new mongoose.Schema({
    username: { type: String, required: true, unique: true },
    password: { type: String, required: true }
});

userSchema.methods.comparePassword = async function (password) {
    return password === this.password;
};

// creation du modele a partir du schema
const User = mongoose.model("User", userSchema);

```

c) Configuration de Passport.js

Passport est configuré avec une stratégie locale (LocalStrategy) :

```
passport.use(new LocalStrategy(async (username, password, done) => {
  try {
    const user = await User.findOne({ username });
    if (!user) return done(null, false, { message: "Utilisateur non trouvé" });

    const isMatch = await user.comparePassword(password);
    if (!isMatch) return done(null, false, { message: "Mot de passe incorrect" });

    return done(null, user);
  } catch (err) {
    return done(err);
  }
}));

```

d) Routes principales

Page d'accueil, login, register

```
app.get("/", (req, res) => res.redirect("/login"));

app.get("/login", (req, res) => res.render("login"));
app.get("/register", (req, res) => res.render("register"));
```

Inscription (/register) et login (/login):

```
app.post("/register", async (req, res) => {
  const { username, password } = req.body;
  try {
    const user = new User({ username, password });
    await user.save();
    res.redirect("/login");
  } catch (err) {
    res.send("Erreur : utilisateur déjà existant");
  }
});
```

Page protégée /books:

```
function isAuthenticated(req, res, next) {
  if (req.isAuthenticated()) return next();
  res.redirect("/login");
}

// Liste des livres (visible uniquement si connecté)
const books = [
  { title: "Les Misérables", author: "Victor Hugo" },
  { title: "L'Étranger", author: "Albert Camus" },
  { title: "Le Comte de Monte-Cristo", author: "Alexandre Dumas" },
  { title: "Notre-Dame de Paris", author: "Victor Hugo" }
];

app.get("/books", isAuthenticated, (req, res) => {
  res.render("books", { user: req.user, books });
});

app.get("/logout", (req, res) => {
  req.logout(() => res.redirect("/login"));
});
```

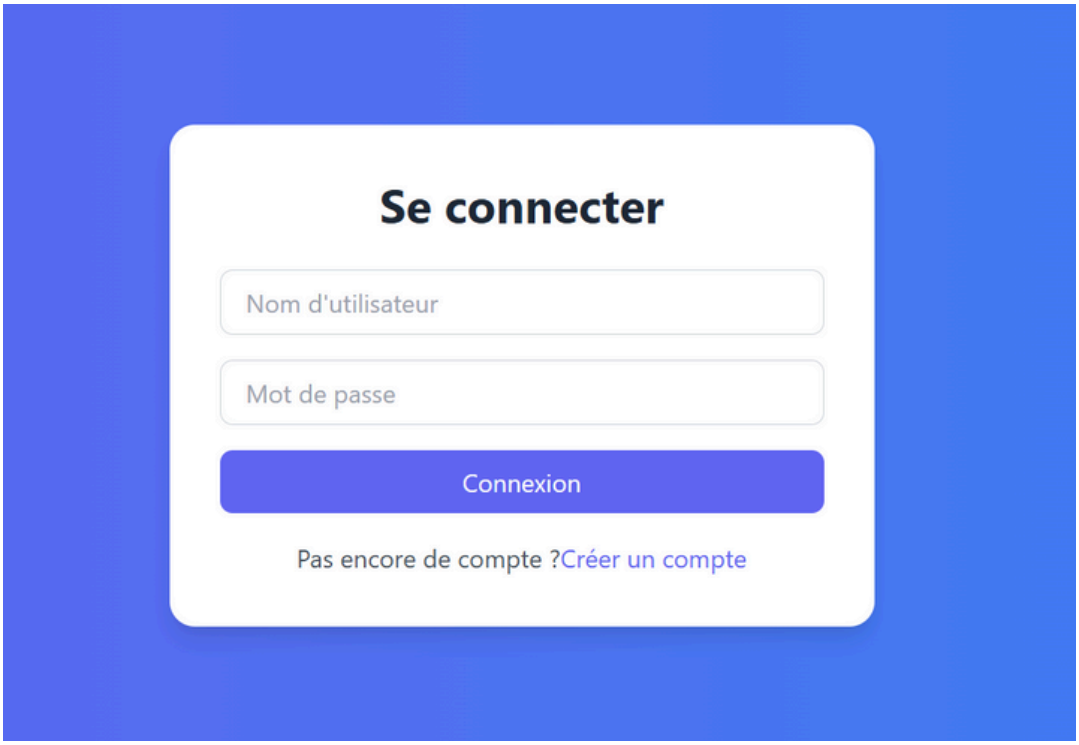
e) Interface utilisateur

Page login.pug

```
doctype html
html
  head
    title Connexion
    link(rel="stylesheet", href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css")
  body(class="bg-gradient-to-r from-indigo-500 to-blue-500 min-h-screen flex items-center justify-center")
    div(class="bg-white rounded-2xl shadow-lg p-8 w-full max-w-md")
      h1(class="text-3xl font-bold text-center text-gray-800 mb-6") Se connecter
      form(action="/login" method="POST" class="space-y-4")
        input(type="text" name="username" placeholder="Nom d'utilisateur" required
          class="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-400")
        input(type="password" name="password" placeholder="Mot de passe" required
          class="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-indigo-400")
        button(type="submit"
          class="w-full bg-indigo-500 text-white py-2 rounded-lg hover:bg-indigo-600 transition") Connexion
      p(class="text-center text-gray-600 mt-4")
        | Pas encore de compte ?
        a(href="/register" class="text-indigo-500 hover:underline") Créer un compte

```

Browser



Page registry.pug

```
doctype html
html
  head
    title Inscription
    link(rel="stylesheet", href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css")
  body(class="bg-gradient-to-r from-green-400 to-teal-500 min-h-screen flex items-center justify-center")
    div(class="bg-white rounded-2xl shadow-lg p-8 w-full max-w-md")
      h1(class="text-3xl font-bold text-center text-gray-800 mb-6") Créer un compte
      form(action="/register" method="POST" class="space-y-4")
        input(type="text" name="username" placeholder="Nom d'utilisateur" required
          class="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-green-400")
        input(type="password" name="password" placeholder="Mot de passe" required
          class="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-green-400")
        button(type="submit"
          class="w-full bg-green-500 text-white py-2 rounded-lg hover:bg-green-600 transition") S'inscrire
      p(class="text-center text-gray-600 mt-4")
        | Déjà un compte ?
        a(href="/login" class="text-green-500 hover:underline") Se connecter

```

Browser

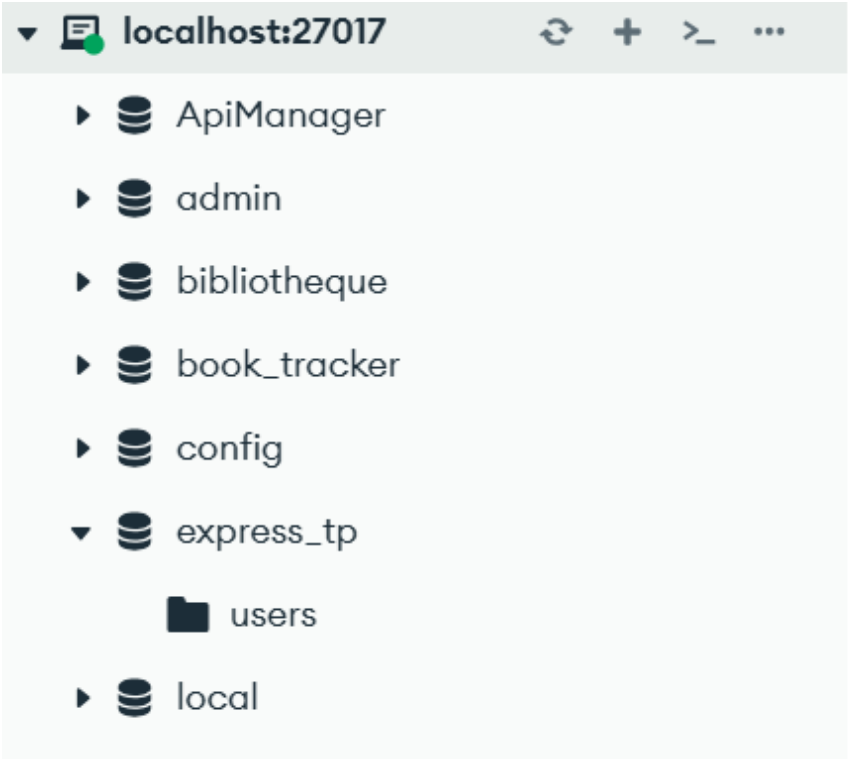
Page books.pug

```
doctype html
html
  head
    title Liste des livres
    link(rel="stylesheet", href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css")
  body(class="bg-gray-100 min-h-screen flex flex-col items-center justify-center")
    div(class="bg-white shadow-lg rounded-2xl p-8 w-full max-w-md text-center")
      h2(class="text-2xl font-bold text-gray-800 mb-4") Bienvenue #{user.username} !
      h3(class="text-xl text-gray-700 mb-4") Liste des livres disponibles :
      ul(class="text-left mb-6 space-y-2")
        each book in books
          li(class="p-3 bg-gray-50 rounded-lg shadow-sm hover:bg-gray-100 transition")
            strong(class="text-indigo-600") #{book.title}
            | - #{book.author}
      a(href="/logout" class="inline-block bg-red-500 text-white px-4 py-2 rounded-lg hover:bg-red-600 transition") Se déconnecter

```

Browser

f) Database (mongo Compass):




```
_id: ObjectId('690e08e6fbe4961b63886264')
username : "med"
password : "Mehdi1234@"
__v : 0
```

CONCLUSION :

Ce travail pratique a permis d'appliquer concrètement :

- la mise en place d'un serveur Express utilisant des vues Pug,
- la connexion et la manipulation d'une base de données MongoDB à l'aide de Mongoose,
- la mise en œuvre d'un système d'authentification complet et sécurisé grâce à Passport.js,
- ainsi que l'utilisation de Tailwind CSS pour concevoir une interface moderne, claire et responsive.

Cette réalisation "full-stack" met en évidence la manière dont les différentes couches d'une application web serveur, base de données, middleware et interface utilisateur interagissent pour former un ensemble cohérent et fonctionnel.