

Bài 8. Tạo API Đăng Xuất 🎉

1. File package.json 🍌

```
npm i cookie-parser
```

```
{
  "name": "training",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js",
    "dev": "nodemon server.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "bcrypt": "^5.0.0",
    "body-parser": "^1.19.0",
    "cookie-parser": "^1.4.5",
    "express": "^4.17.1",
    "jsonwebtoken": "^8.5.1",
    "mongoose": "^5.9.28",
    "nodemon": "^2.0.4",
    "passport": "^0.4.1",
    "passport-jwt": "^4.0.0",
    "passport-local": "^1.0.0"
  }
}
```

2. File server.js 🍌 🍌

```
const express = require("express");
const mongoose = require("mongoose");
const bodyParser = require("body-parser");
const app = express();

app.use(bodyParser.json());

const db = require("./config/key").mongoURI;

mongoose
  .connect(db, { useNewUrlParser: true, useUnifiedTopology: true })
  .then(() => console.log("MongoDB Connected..."))
  .catch((err) => console.log(err));

app.use("/", require("./router/User"));
const PORT = process.env.PORT || 8080;
app.listen(PORT, console.log(`Server Run With Port ${PORT}`));
```

3. File User.js 🍌🍌🍌

```
const express = require("express");
const userRouter = express.Router();
const JWT = require("jsonwebtoken");
const passport = require("passport");
const passportConfig = require("../config/passport");
const User = require("../model/User");

// register
userRouter.post("/register", (req, res) => {
  const { username, password } = req.body;

  const role = "user";

  //kiem tra username co ton tai khong
  User.findOne({ username }, (err, user) => {
    if (err)
      res.status(500).json({
        message: { msgBody: "Error", msgError: true },
      });
    if (user)
      res.status(201).json({
        message: { msgBody: "Ten Dang Nhap Da Ton Tai", msgError: true },
      });
    else {
      const newUser = new User({ username, password, role });
      newUser.save((err) => {
        if (err)
          res.status(500).json({
            message: { msgBody: "Error", msgError: true },
          });
        else
          res.status(200).json({
            message: { msgBody: "Tao Tai Khoan Thanh Cong", msgError: false },
          });
      });
    }
  });
});

// login
const signToken = (userID) => {
  return JWT.sign(
    {
      iss: "QuocLiem",
      sub: userID,
    },
    "QuocLiem",
  );
};
```

```

    { expiresIn: "1d" }
  );
};

userRouter.post(
  "/login",
  passport.authenticate("local", { session: false }),
  (req, res) => {
    if (req.isAuthenticated()) {
      const { _id, username, role } = req.user;
      const token = signToken(_id);
      res.cookie("access_token", token, { httpOnly: true, sameSite: true });
      res.status(200).json({
        isAuthenticated: true,
        user: { _id, username, role },
      });
    }
  }
);

// logout
userRouter.get(
  "/logout",
  passport.authenticate("jwt", { session: false }),
  (req, res) => {
    res.clearCookie("access_token");
    res.json({ user: { username: "", role: "" }, success: true });
  }
);

// kiểm tra đã đăng nhập hay chưa
userRouter.get(
  "/authenticated",
  passport.authenticate("jwt", { session: false }),
  (req, res) => {
    const { _id, username, role } = req.user;
    res.status(200).json({
      isAuthenticated: true,
      user: {
        _id,
        username,
        role,
      },
    });
  }
);

module.exports = userRouter;

```

The End

