Lecture 12. Authentication in Nodejs using Express and MongoDB

13 April 2025 18:48

```
// Folder structure assumed:
// - auth-app/
//   - public/
//       - login.html
//       - signup.html
//       - home.html
//   - models/
//       - User.js
//   - routes/
//       - auth.js
//   - server.js
//   - .env

// ------------------- .env -------------------
PORT=5000
MONGO_URI=mongodb://localhost:27017/authdb
JWT_SECRET=your_jwt_secret_key

// ------------------- models/User.js -------------------
const mongoose = require('mongoose');
const bcrypt = require('bcryptjs');

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

userSchema.pre('save', async function (next) {
  if (!this.isModified('password')) return next();
  const salt = await bcrypt.genSalt(10);
  this.password = await bcrypt.hash(this.password, salt);
  next();
});

userSchema.methods.comparePassword = async function (enteredPassword) {
  return await bcrypt.compare(enteredPassword, this.password);
};

module.exports = mongoose.model('User', userSchema);

// ------------------- routes/auth.js -------------------
const express = require('express');
const jwt = require('jsonwebtoken');
const User = require('../models/User');
require('dotenv').config();

const router = express.Router();

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

```
  const { name, email, password } = req.body;
  try {
    let user = await User.findOne({ email });
    if (user) return res.status(400).json({ msg: 'User already exists' });

    user = new User({ name, email, password });
    await user.save();

    res.status(201).json({ msg: 'User registered successfully' });
  } catch (err) {
    res.status(500).send('Server error');
  }
});

router.post('/login', async (req, res) => {
  const { email, password } = req.body;
  try {
    const user = await User.findOne({ email });
    if (!user) return res.status(400).json({ msg: 'Invalid credentials' });

    const isMatch = await user.comparePassword(password);
    if (!isMatch) return res.status(400).json({ msg: 'Invalid credentials' });

    const token = jwt.sign({ id: user._id }, process.env.JWT_SECRET, { expiresIn: '1d' });
    res.json({ token });
  } catch (err) {
    res.status(500).send('Server error');
  }
});

module.exports = router;

// ------------------- server.js -------------------
const express = require('express');
const mongoose = require('mongoose');
const dotenv = require('dotenv');
const path = require('path');
const authRoutes = require('./routes/auth');

dotenv.config();
const app = express();

app.use(express.json());
app.use(express.static(path.join(__dirname, 'public')));

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

app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'public', 'login.html'));
});

app.get('/home', (req, res) => {
  res.sendFile(path.join(__dirname, 'public', 'home.html'));
});

mongoose.connect(process.env.MONGO_URI, {
  useNewUrlParser: true,
```

```
  useUnifiedTopology: true,
}).then(() => {
  console.log('MongoDB connected');
  app.listen(process.env.PORT, () => console.log(`Server running on port ${process.env.PORT}`));
}).catch(err => console.error('Mongo Error:', err));

// ------------------- public/login.html -------------------
<!DOCTYPE html>
<html>
<head><title>Login</title></head>
<body>
  <h2>Login</h2>
  <form id="loginForm">
    <input type="email" placeholder="Email" id="email" required><br>
    <input type="password" placeholder="Password" id="password" required><br>
    <button type="submit">Login</button>
  </form>
  <p id="msg"></p>
  <p>Don't have an account? <a href="signup.html">Sign up</a></p>

  <script>
    document.getElementById('loginForm').addEventListener('submit', async (e) => {
      e.preventDefault();
      const res = await fetch('/api/auth/login', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({
          email: document.getElementById('email').value,
          password: document.getElementById('password').value
        })
      });
      const data = await res.json();
      if (res.ok) {
        window.location.href = '/home';
      } else {
        document.getElementById('msg').innerText = data.msg || 'Login failed';
      }
    });
  </script>
</body>
</html>

// ------------------- public/signup.html -------------------
<!DOCTYPE html>
<html>
<head><title>Sign Up</title></head>
<body>
  <h2>Sign Up</h2>
  <form id="signupForm">
    <input type="text" placeholder="Name" id="name" required><br>
    <input type="email" placeholder="Email" id="email" required><br>
    <input type="password" placeholder="Password" id="password" required><br>
    <button type="submit">Sign Up</button>
  </form>
  <p id="msg"></p>
  <p>Already have an account? <a href="login.html">Login</a></p>
```

```
 <script>
  document.getElementById('signupForm').addEventListener('submit', async (e) => {
   e.preventDefault();
   const res = await fetch('/api/auth/signup', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({
     name: document.getElementById('name').value,
     email: document.getElementById('email').value,
     password: document.getElementById('password').value
    })
   });
   const data = await res.json();
   document.getElementById('msg').innerText = data.msg;
  });
 </script>
</body>
</html>

// ---------- public/home.html ----------
<!DOCTYPE html>
<html>
<head><title>Home</title></head>
<body>
 <h2>Welcome to the Home Page!</h2>
 <p>You have successfully logged in.</p>
</body>
</html>
```