## server.js (Main backend file)

const express = require('express');

const mongoose = require('mongoose');

const bcrypt = require('bcryptjs');

const session = require('express-session');

const bodyParser = require('body-parser');

const path = require('path');

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

const app = express();

const PORT = 3000;

// MongoDB connection

mongoose.connect('mongodb://127.0.0.1:27017/userAuth', {

useNewUrlParser: true,

useUnifiedTopology: true

}).then(() => console.log('✅ MongoDB connected'))

.catch(err => console.log(err));

// Middleware

app.use(bodyParser.urlencoded({ extended: true }));

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

app.use(session({

secret: 'secretKey',

resave: false,

saveUninitialized: true

}));

// Routes

app.get('/', (req, res) => res.sendFile(path.join(\_\_dirname, 'public/index.html')));

app.get('/register', (req, res) => res.sendFile(path.join(\_\_dirname, 'public/register.html')));

app.get('/home', (req, res) => {

if (req.session.userId) {

res.sendFile(path.join(\_\_dirname, 'public/home.html'));

} else {

res.redirect('/');

}

});

// Register route

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

const { username, email, password } = req.body;

const hashed = await bcrypt.hash(password, 10);

const user = new User({ username, email, password: hashed });

await user.save();

res.redirect('/');

});

// Login route

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

const { email, password } = req.body;

const user = await User.findOne({ email });

if (user && await bcrypt.compare(password, user.password)) {

req.session.userId = user.\_id;

res.redirect('/home');

} else {

res.send('<h3>Invalid login! <a href="/">Try again</a></h3>');

}

});

// Logout

app.get('/logout', (req, res) => {

req.session.destroy(() => res.redirect('/'));

});

app.listen(PORT, () => console.log(`🚀 Server running at [http://localhost:${PORT}`));](http://localhost:$%7bPORT%7d%60));)

## models/User.js

const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({

username: { type: String, required: true },

email: { type: String, required: true, unique: true },

password: { type: String, required: true }

});

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

## public/index.html (Login Page)

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

<style>

body { font-family: Arial; text-align: center; background: #eef2f3; }

form { margin-top: 100px; background: white; padding: 30px; display: inline-block; border-radius: 10px; }

input { margin: 10px; padding: 10px; width: 200px; }

button { padding: 10px 20px; }

</style>

</head>

<body>

<h2>Login</h2>

<form action="/login" method="POST">

<input type="email" name="email" placeholder="Email" required /><br>

<input type="password" name="password" placeholder="Password" required /><br>

<button type="submit">Login</button>

</form>

<p>Don’t have an account? <a href="/register">Register</a></p>

</body>

</html>

## public/register.html (Signup Page)

<!DOCTYPE html>

<html>

<head>

<title>Register</title>

<style>

body { font-family: Arial; text-align: center; background: #eef2f3; }

form { margin-top: 100px; background: white; padding: 30px; display: inline-block; border-radius: 10px; }

input { margin: 10px; padding: 10px; width: 200px; }

button { padding: 10px 20px; }

</style>

</head>

<body>

<h2>Create Account</h2>

<form action="/register" method="POST">

<input type="text" name="username" placeholder="Username" required /><br>

<input type="email" name="email" placeholder="Email" required /><br>

<input type="password" name="password" placeholder="Password" required /><br>

<button type="submit">Register</button>

</form>

<p>Already have an account? <a href="/">Login</a></p>

</body>

</html>

## public/home.html (After Login)

<!DOCTYPE html>

<html>

<head>

<title>Home</title>

<style>

body { font-family: Arial; text-align: center; background: #d7ffd9; }

h2 { margin-top: 100px; }

a { text-decoration: none; color: blue; }

</style>

</head>

<body>

<h2>Welcome! You are logged in.</h2>

<a href="/logout">Logout</a>

</body>

</html>