# User Profile Microservice – Code Documentation

## server.js

const express = require('express');  
const app = express();  
const PORT = process.env.PORT || 3000;  
  
const userRoutes = require('./src/routes/userRoutes');  
const authRoutes = require('./src/routes/authRoutes');  
  
app.use(express.json());  
app.use('/api/users', userRoutes);  
app.use('/api/auth', authRoutes);  
  
app.get('/health', (req, res) => {  
 res.json({ message: 'User Profile Service is running' });  
});  
  
app.listen(PORT, () => {  
 console.log(`Server running on port ${PORT}`);  
});

## src/models/userModel.js

let users = [];  
  
module.exports = {  
 users  
};

## src/utils/passwordUtils.js

const bcrypt = require('bcrypt');  
  
const hashPassword = async (password) => {  
 const salt = await bcrypt.genSalt(10);  
 return await bcrypt.hash(password, salt);  
};  
  
const comparePassword = async (plain, hashed) => {  
 return await bcrypt.compare(plain, hashed);  
};  
  
module.exports = { hashPassword, comparePassword };

## src/utils/validation.js

const isValidEmail = (email) => {  
 return /\S+@\S+\.\S+/.test(email);  
};  
  
const isValidPassword = (password) => {  
 return password.length >= 6;  
};  
  
module.exports = { isValidEmail, isValidPassword };

## src/services/userService.js

const { users } = require('../models/userModel');  
  
const createUser = (user) => {  
 users.push(user);  
 return user;  
};  
  
const findUserByEmail = (email) => {  
 return users.find((u) => u.email === email);  
};  
  
module.exports = { createUser, findUserByEmail };

## src/middleware/authMiddleware.js

const jwt = require('jsonwebtoken');  
  
const authMiddleware = (req, res, next) => {  
 const token = req.headers.authorization?.split(' ')[1];  
 if (!token) return res.status(401).json({ message: 'No token provided' });  
  
 try {  
 const decoded = jwt.verify(token, 'sekret');  
 req.user = decoded;  
 next();  
 } catch (err) {  
 res.status(401).json({ message: 'Invalid token' });  
 }  
};  
  
module.exports = authMiddleware;

## src/controllers/authController.js

const jwt = require('jsonwebtoken');  
const { findUserByEmail } = require('../services/userService');  
const { comparePassword } = require('../utils/passwordUtils');  
  
const login = async (req, res) => {  
 const { email, password } = req.body;  
 const user = findUserByEmail(email);  
  
 if (!user || !(await comparePassword(password, user.password))) {  
 return res.status(401).json({ message: 'Invalid credentials' });  
 }  
  
 const token = jwt.sign({ email: user.email }, 'sekret', { expiresIn: '1h' });  
 res.json({ token });  
};  
  
module.exports = { login };

## src/controllers/userController.js

const { hashPassword } = require('../utils/passwordUtils');  
const { isValidEmail, isValidPassword } = require('../utils/validation');  
const { createUser, findUserByEmail } = require('../services/userService');  
  
const register = async (req, res) => {  
 const { email, password } = req.body;  
  
 if (!isValidEmail(email) || !isValidPassword(password)) {  
 return res.status(400).json({ message: 'Invalid input' });  
 }  
  
 if (findUserByEmail(email)) {  
 return res.status(409).json({ message: 'User already exists' });  
 }  
  
 const hashedPassword = await hashPassword(password);  
 const newUser = createUser({ email, password: hashedPassword });  
  
 res.status(201).json({ message: 'User registered', user: { email: newUser.email } });  
};  
  
const getProfile = (req, res) => {  
 res.json({ email: req.user.email });  
};  
  
module.exports = { register, getProfile };

## src/routes/userRoutes.js

const express = require('express');  
const router = express.Router();  
const { register, getProfile } = require('../controllers/userController');  
const authMiddleware = require('../middleware/authMiddleware');  
  
router.post('/register', register);  
router.get('/profile', authMiddleware, getProfile);  
  
module.exports = router;

## src/routes/authRoutes.js

const express = require('express');  
const router = express.Router();  
const { login } = require('../controllers/authController');  
  
router.post('/login', login);  
  
module.exports = router;