# User Profile Microservice – Detailed Developer Documentation

## Step 1: Project Setup

🛠️ Çfarë kam bërë në Visual Studio Code:

* • Kam krijuar strukturën e projektit me komandat `mkdir` në terminal.
* • Kam inicializuar projektin me `npm init -y`.
* • Kam instaluar libraritë: `express`, `bcrypt`, `jsonwebtoken`, `nodemon`.
* • Kam krijuar `server.js` me konfigurim bazë të Express dhe endpoint-in `/health`.
* • Kam verifikuar funksionimin e serverit me `curl http://localhost:3000/health`.

🗣️ Si ta prezantoj te profesori:

* • Tregoj që kam përdorur strukturë profesionale me dosje për controllers, routes, services, etj.
* • Shpjegoj që `server.js` është pika hyrëse e aplikacionit dhe se endpoint-i `/health` vërteton që serveri është aktiv.
* • Tregoj që kam përdorur `nodemon` për zhvillim më të shpejtë.

## Step 2: User Model & Storage

🛠️ Çfarë kam bërë në Visual Studio Code:

* • Kam krijuar `userModel.js` dhe një array bosh `users[]` për të ruajtur përdoruesit.
* • Kam krijuar `userService.js` me dy funksione: `createUser()` dhe `findUserByEmail()`.
* • Të gjitha të dhënat ruhen në memorien RAM dhe jo në databazë (si testim).

🗣️ Si ta prezantoj te profesori:

* • Shpjegoj që përdor të dhëna në memorie për shkak se është një mikrosherbim testues.
* • Tregoj që `createUser` dhe `findUserByEmail` janë funksione logjike të domosdoshme për regjistrim dhe login.

## Step 3: Authentication

🛠️ Çfarë kam bërë në Visual Studio Code:

* • Kam krijuar `passwordUtils.js` me funksione për `hashPassword` dhe `comparePassword` duke përdorur bcrypt.
* • Kam ndërtuar `authController.js` për të gjeneruar JWT token gjatë login-it.
* • Kam krijuar `authMiddleware.js` që verifikon tokenin JWT për endpoint-et e mbrojtura.
* • Kam mbrojtur `/api/users/profile` me middleware për autentifikim.

🗣️ Si ta prezantoj te profesori:

* • Shpjegoj që fjalëkalimet nuk ruhen në tekst të thjeshtë por ruhen të enkriptuara me bcrypt.
* • JWT përdoret për të krijuar një sesion të sigurt për përdoruesin.
* • Middleware siguron që vetëm përdoruesit e kyçur mund të aksesojnë endpoint-et sensitive.

## Step 4: API Routes

🛠️ Çfarë kam bërë në Visual Studio Code:

* • Kam krijuar `authRoutes.js` me endpoint-in `/login`.
* • Kam krijuar `userRoutes.js` me `/register` dhe `/profile`.
* • Kam përdorur `authMiddleware` në `/profile` për të kufizuar qasjen.
* • Kam krijuar `validation.js` për të verifikuar email dhe password-in përpara ruajtjes.

🗣️ Si ta prezantoj te profesori:

* • Shpjegoj që kam ndarë ruterat sipas përgjegjësive për mirëmbajtje më të lehtë.
* • Shpjegoj që `validation.js` ndihmon që të parandalohet input-i i gabuar ose i rrezikshëm.
* • Tregoj që përdor metoda HTTP të duhura (`POST`, `GET`) sipas REST.

## Step 5: Testing

🛠️ Çfarë kam bërë në Visual Studio Code:

* • Kam testuar me `curl` në Git Bash të gjitha endpoint-et: `/health`, `/register`, `/login`, `/profile`.
* • Kam testuar raste gabimi (negative testing):
* • - Regjistrim me email të njëjtë → 409 Conflict
* • - Login me password të gabuar → 401 Unauthorized
* • - Qasje në `/profile` pa token → 401 Unauthorized
* • - Qasje në `/profile` me token të pavlefshëm → 401 Unauthorized

🗣️ Si ta prezantoj te profesori:

* • Shpjegoj që kam përdorur Git Bash për të testuar API-në si një klient real.
* • Demostroj komandat `curl` dhe tregoj përgjigjet që marr nga serveri.
* • Theksoj që kam bërë testime pozitive dhe negative për të provuar që sistemi është i sigurt dhe i saktë.

## Step 6: Documentation

🛠️ Çfarë kam bërë në Visual Studio Code:

* • Kam krijuar një `README.md` të strukturuar me: përshkrim, teknologji, udhëzime për nisje dhe testim.
* • Kam dokumentuar API endpoint-et me shembuj konkret `curl`.
* • Kam shtuar komente në kod për të shpjeguar logjikën në kontrollues dhe middleware.

🗣️ Si ta prezantoj te profesori:

* • Tregoj që dokumentimi është i domosdoshëm për çdo projekt të mirë.
* • Shpjegoj që `README.md` ndihmon zhvillues të tjerë ta përdorin dhe kuptojnë projektin pa e lexuar të gjithë kodin.
* • Theksoj që kam komentuar kodin për qartësi maksimale.