Detyre Kursi

# Prezantimi i Projektit

Në këtë projekt, kemi zhvilluar një aplikacion Spring Boot që përdor një bazë të dhënash MySQL. Projekti përfshin menaxhimin e përdoruesve, autentikimin, regjistrimin, dhe modifikimin e profileve nga një përdorues administrator. Kemi ndarë funksionalitetet në disa klasa dhe controllera.

# Klasa User

Kemi një klasë të quajtur `User`, e cila përdoret për të ruajtur entitetet e përdoruesve. Kjo klasë përmban fusha të tilla si emri, mbiemri, numri i telefonit, data e lindjes, email-i, fjalëkalimi dhe fotoja e profilit.

# Controllerat

Kemi ndërtuar disa controllera për menaxhimin e operacioneve CRUD dhe funksionaliteteve të tjera:

* `RegistrationController`: Për regjistrimin e përdoruesve.
  + POST /register – për të regjistruar (krijuar) një përdorues të ri, jo-admin.
* `ProfileController`: Për shfaqjen dhe modifikimin e të dhënave të profilit të përdoruesit (si dhe komandat e adminit).
  + GET /login – Shfaq faqen e log in për përdoruesit
  + GET /register – Shfaq faqen e regjistrimit për përdoruesit
  + GET /profile – Shfaq faqen e profilit për përdoruesit
  + POST /profile – ruan të dhënat e reja të përdoruesit (përfshire dhe fotografinë nëse ka)
  + GET /admin/users – Shfaq një list të të gjithë përdoruesve në platformë. (Vetëm për admin)
  + GET /admin/users/edit – Shfaq një ndërfaqe ku admini mund të ndryshojë të dhënat e një përdoruesi. (Vetëm për admin)
  + POST /admin/users/edit – Ruan të dhënat e rejat të një përdoruesi. (Vetëm për admin)
  + GET /admin/users/delete – Fshin një përdorues. (Vetëm për admin)
  + POST /profile/remove-photo – Fshin foton e profilit të një përdoruesi.
* `AuthController`: Ka vetëm një action, POST për login për të kyçur përdoruesit në sistem.

# Pamjet (Views)

Kemi krijuar disa pamje për të mundësuar ndërveprimin e përdoruesve me aplikacionin:  
- `register.html`: Pamja për regjistrimin e përdoruesve të rinj.  
- `login.html`: Pamja për autentikimin e përdoruesve ekzistues.  
- `profile.html`: Pamja për shfaqjen dhe modifikimin e profilit të përdoruesit.   
- `user\_list.html`: Pamja për administratorin për të menaxhuar përdoruesit e regjistruar.   
- `edit\_user.html`: Pamja për shfaqjen dhe modifikimin e profilit të një përdoruesi të zgjedhur nga administratori.

# Config dhe Services

Kemi dy files në një folder config për aplikacionin.   
- `DataInitializer.java`: Përdoret për të inicalizuar një përdorues fillestar administrator në databazë.   
- `SecurityConfig.java`: Përdoret për të konfiguruar login, register, dhe logout routes. Gjithashtu të gjithë routet /admin/\*\* janë të konfiguruara që të aksesohen vetëm nga përdorues admin.

Gjithashtu kemi dhe services të ndarë në AuthService dhe UserService. Në këto service ndodhet logjika e cila është përdorur në controllerat më sipër.

# Konfigurimi i Docker

Kemi krijuar një file `docker-compose.yml` për të konfiguruar aplikacionin dhe bazën e të dhënave MySQL në Docker. Kjo file përfshin shërbimet për aplikacionin dhe bazën e të dhënave, dhe konfiguron variablat e ambientit të nevojshme për lidhjen me DB.

Gjithashtu kemi krijuar një Dockerfile për të krijuar një “paketë” të aplikacionit e cila përdoret në docker-compose.

# Screenshot të aplikacionit

## Login

A screenshot of a login page

Description automatically generated

## Profile

A screenshot of a car

Description automatically generated

## Admin – lista përdoruesve

A screenshot of a contact us

Description automatically generated