**Adventures park**

- Teknologjite: Spring Boot, Angular, Mongo DB, WebSocket

- Permban: CRUD per lojerat, Login/Register per admin dhe user, Auth guard per aksesimin e

Ruteve, PayPal

***Back end***

-Shto dependencies per Security tek pom.xml

-Shto jwtSecret dhe jwtExpirationMs tek application.properties

COFIGURATION

WebSocketConfig 🡪 konfigurimet per websocket, krijimi i endpoint dhe percaktimi i topic

CONTROLLER

GameController 🡪 kerkesat me websocket per funksionet e listimit, shtimit dhe fshirjes se lojerave

UserController 🡪 kerkesa http per te shtuar nje rol

TestController 🡪 kerkesat http per aksesin ne varesi te roleve ‘USER’ dhe ‘ADMIN’

AuthController 🡪 kerkesat per signIn dhe signUp. Per SignIn behet autentikimi i userit dhe merren kredencialet username, password per te bere login ne faqe. Per SingUp kontrollohet ne fillim nese useri ekziston dhe nxjerr mesazh gabimi. Me pas ben krijimin e accountit te ri te userit, duke marre emailin, username dhe password. Behet kontrolli per rolin dhe nese nuk eshte percaktuar, vendoset ‘USER’ si default. (ne front nuk percaktohet roli, pra te gjithe do te jene usera; admini krijohet nga postman) Behet save useri i ri dhe ruhet ne database.

MODEL

Game 🡪 modeli i lojes

User 🡪 modeli i userit, i cili permban dhe nje liste me role

ERole 🡪 enumeracion qe permban dy rolet ‘USER’ dhe ‘ADMIN’

Role 🡪 mban nje id dhe nje atribut ERole. Keto do te ruhen ne tabelen e roleve ne database

REPOSITORY

GameRepository 🡪 krijon repository per te bere kerkesat nga database per lojerat

UserRepository 🡪 krijon repository per te bere kerkesat nga database per userat. Shton dhe tre metoda te reja: findByUsername, existsByUsername dhe existsByEmail

RoleRepository 🡪 krijon repository per te bere kerkesat nga database per rolet. Shton dhe metoden fidByName

SERVICE

GameService 🡪 metodat e CRUD per lojerat

REQUEST

LoginRequest 🡪 kerkesa qe do te beje klienti ne momentin qe do te logohet (username + password

SignUpRequest 🡪 kerkesa qe ben klienti kur behet register. Klasa permban vetem se cfare duhet nga kerkesa

RESPONSE

JwtResponse 🡪 Pergjigjia qe server do te ktheje pasi behet kerkesa. Ka nje metode getAccesToken e cila kthen tokenin

MessageResponse 🡪 Mesazhi qe server do te ktheje pasi behet kerkesa

SECURITY

**JWT**

AuthEntryPointJwt 🡪 implementon nderfaqen qe e nis skemen e autentikimit

AuthTokenFilter 🡪 permban dy metodat doFilterInternal pasi merr requestin dhe response.

Metoda parseJwt e cila ben parse kerkesen

JwtUtils 🡪 lidhet me jwtSecret dhe jwtExpirationMs qe jane percaktuar tek

application.properties. Permban metodat generateJWtToken per gjenerimin e tokenit,

getUsernameFromJwtToken dhe validateJwtToken.

**SERVICES**

UserDetailsImpl 🡪 implementon nderfaqen UserDetails, merr te gjitha te dhenat e userit

dhe ben kontroll nese jane expired ose jo

UserDetailsServiceImpl 🡪 ben loadUserByUsername dhe i ben build nje useri

WebSecurityConig 🡪 ben konfigurime per token filter, ben enkriptimin e password dhe metoda configure ne te cilen jane te gjitha permissions per rutet e controllerave, pra kush mund ti aksesoje.

***Frontend***

-Perdorimi i AuthGuard dhe PayPal

\_SERVICES

Auth.service 🡪 ben lidhjen me spring dhe endpoint te controllerit ‘/auth’ . Metodat login, register qe marrin te dhenat nga format, metoda isAuthenticated() qe kontrollon nese useri eshte i autentikuar apo jo pasi kontrollon tokenin, metoda canActivate() qe tregon nese navigimi ne nje route eshte i lejuar apo jo

Token-storage.service 🡪 ka dy konstante token per auth-token dhe auth-user, metoda signOut qe ben clear data qe ishte ruajtur vetem per nje session, metodat saveToken() dhe getToken(), metodat saveUser() dhe getUser() qe ruajne dhe marrin tokenin e user.

User.service 🡪 ben lidhjen me spring dhe endpoint te controller ‘/test’. Permban metodat qe merr user role, admin dhe all. (une nuk e kam perdorur ne front)

BOARD-ADMIN

Komponenti i admin i cili pasi logohet mud te aksesoje faqen e tij, te krijoje dhe te fshije lojerat.

GAME-HOME

Komponentja qe ben shfaqjen e te gjitha lojerave qe jane te ruajtura ne database. Butoni ‘delete’ per fshirjen e lojerave do te shfaqet vetem per rolin admin. Kontrolli I rolit behet tek onInit() dhe me pas tek butoni delete ne html. Butoni ‘pay’ te con direct tek faqja e PayPal.

PAY-PAL

Ben instalimin ‘npm install ngx-paypal –save’ dhe me pas krijon formen dhe funksionet per te naviguar te faqja e pay pal.

MODELS

Modelet e lojerave dhe userave

REGISTER

Komponentja per formen e regjistrimit te userit dhe krijimit te accountit te ri.

LOGIN

Komponentja per loginin e userit dhe adminit. Ne momentin qe behet login, lar ten navbar shfaqet LogOut. Merr dhe rolin. Ne html behen validimet nese kredencialet jane te future te sakta apo jo

HOME

Faqja kryesore qe permban butonat LogIn dhe SignUp.

APP-ROUTING.MODULE.TS

Permban te gjitha rutet. ‘/allGames’ mund te aksesohet vetem nese je i loguar, s’ka rendesi roli, nersa ‘/admin’ mund te aksesohet vetem nga roli admin. Ndaj eshte perdorut canActivate: [AuthGuard] dhe data:{expectedRole: ‘ADMIN’}

ASSETS

Permban fotot dhe muzika qe degjohet nese navigon tek faqja Home

WEBSOCKETAPI.TS

Permban te gjitha konfigurimet e WebSocket, te cilat do te perdoren tek CRUD i lojerave.