

SKILLSWAP



BETA

Sistem për Menaxhimin e Eventeve Peer-to-Peer

KONNEKSI

Ky projekt është një aplikacion web i ndërtuar me Spring Boot që mundëson krijimin dhe menaxhimin e eventeve ku pjesëmarrësit mund të mësojnë aftësi të ndryshme nga njëritjetri (p.sh. programim, fotografi, gjuhë, fitness, etj.). Platforma synon të krijojë një mjedis të organizuar ku një përdorues mund të jetë edhe host (organizator), edhe pjesëmarrës, duke promovuar mësimin bashkëpunues dhe eventet komunitare.

Shtrimii problemit

Organizimi i eveneteve zakonisht bëhet në mënyrë të çrregullt, përmes mesazheve, rrjeteve sociale ose komunikimit informal, ku shpesh humben informacione të rëndësishme si: data, ora, vendndodhja, numri i pjesëmarrësve apo listat e regjistimeve.

Synimetë projektit

- ★ Eventet publikohen me të gjitha detajet përkatëse
- ★ Përdoruesit regjistrohen në mënyrë të kontrolluar
- ★ Host-i mund të monitorojë pjesëmarrësit
- ★ Krijohet mundësia për feedback dhe vlerësim (rating)

QFARË SYNON PROJEKTI?

SkillSwap Events Platform synon të krijojë një mjedis të centralizuar dhe të strukturuar për organizimin e eventeve peer-to-peer, ku përdoruesit ndajnë aftësi dhe njohuri me njëri-tjetrin.

Qëllimi i projektit

Qëllimi kryesor është krijimi i një aplikacioni funksional që:

- ★ Lehtëson krijimin dhe menaxhimin e eveneve
- ★ Siguron regjistrimin e pjesëmarrësve me kufizim (limit)
- ★ Përmirëson cilësinë e eveneve përmes vlerësimeve dhe komenteve
- ★ Krijon një strukturë të qartë të backend-it me API të gatshme për t'u lidhur me frontend-in

ANALIZA E KERKE SAVE

Funksionale dhe Jo-funksionale

Kërkesat jo-funksionale

Qëllimi kryesor është krijimi i një aplikacioni funksional që:

- ★ Aplikacioni duhet të jetë i strukturuar dhe i kuptueshëm (**Controllers / Services / Repositories**)
- ★ Response duhet të kthehen në format **JSON**
- ★ Sistemi duhet të ketë validime bazë (p.sh. email unik, limit pjesëmarrësish, yje 1-5)
- ★ Testimi kryhet përmes **Postman** (prova ekzekutimi)

KËRKESAT FUNKSIONALE

Sistemi duhet të realizojë këto funksione kryesore:

Menaxhimi i përdoruesve (Auth)

- ★ Përdoruesi mund të regjistrohet në sistem
- ★ Përdoruesi mund të bëjë login
- ★ Sistemi ruan rolin e përdoruesit (PERDORUES / ADMIN)

Menaxhimi i kategorive

- ★ Kategoritë shfaqen për filtrime dhe organizim eventesh
- ★ Sistemi mban kategori të aftësive (p.sh. Programim, Gatim, Fotografi...)

Menaxhimi i eveneve

- ★ Host-i krijon evene të reja
- ★ Eventi mund të përditësohet nga host-i
- ★ Eventi mund të anulohet nga host-i
- ★ Eventet listohen dhe filtrueshme sipas kategorisë / kërkimit

Regjistrimi në evenete (Bookings)

- ★ Përdoruesi regjistrohet në event
- ★ Sistemi nuk lejon regjistrim dy herë në të njëjtin event
- ★ Sistemi kontrollon limitin e pjesëmarrësve
- ★ Host-i mund të shohë listën e pjesëmarrësve

Vlerësimet (Feedback & Rating)

- ★ Përdoruesi mund të vlerësojë eventin/host-in me yje (1-5)
- ★ Përdoruesi mund të shtojë koment
- ★ Sistemi llogarit mesataren e rating-ut për host-in
- ★ Sistemi nuk lejon vlerësim pa qenë pjesëmarrës i eventit

Teknologjite e perdorura

Java

Spring Boot

Spring Data JPA

H2 Database

Spring Web
(REST API)

Postman

draw.io

për testim të shpejtë gjatë zhvillimit

për testim të endpoint-eve

për UML: Class Diagram + Sequence Diagram

Arkitektura e sistemit

Sistemi është organizuar sipas një strukture tipike Spring Boot:

- ★ Controller → pranon kërkesat HTTP (API)
- ★ Service → përmban logjikën e biznesit (validime, kontolle)
- ★ Repository → komunikon me databazën (JPA)
- ★ Entities → përfaqësojnë tabelat në databazë
- ★ DTO → përdoren për input/output të kontrolluar për API

Dizajnimi UML

Në këtë projekt është realizuar UML për:

- Class Diagram (struktura e klasave dhe marrëdhëniet)
- Sequence Diagram (komunikimi Controller → Service → Repository)

STRUKTURA E KLASAVE KRYESORE

Në këtë pjesë paraqitet përbledhja e klasave më të rëndësishme:

Entitetet (Entities)

Variablat kryesore:

Perdoruesi

`id: Long`

`email: String`

`emri: String`

`fjalekalim: String`

`mbiemri: String`

`roli: RoliPerdoruesit`

Qëllimi: ruan informacionin e përdoruesit dhe rolin në sistem.

Eventi

`id, titulli, pershkrimi`

`kufiri Pjesemarresve`

`data, ora`

`statusi: StatusEventi`

`vendndodhja`

`host: Perdoruesi`

`kategoria: Kategori Aftesie`

Qëllimi: ruan të gjithë informacionin e një eventi dhe lidhet për hostin/kategorinë.

Entitetet (Entities)

Variablat kryesore:

KategoriAftesie

`id: Long`

`emri: String`

Qëllimi: ruan kategoritë e aftësive që përdoren për klasifikimin e eventeve.

Regjistrimi

`id`

`statusi: StatusRegjistrimi`

`dataRegjistrimit`

`eventi: Eventi`

`perdoruesi: Perdoruesi`

Qëllimi: lidh përdoruesit me eventet dhe kontrollon pjesëmarrjen.

Entitetet (Entities)

Variablat kryesore:

Vleresimi

id

uje (1-5)

koment

dataKrijimit

eventi: Eventi

vleresuesi: Perdoruesi

hosti: Perdoruesi

Qëllimi: ruan feedback-un dhe rating-un e eventit/host-it.

**CONTROLLERS &
SERVICES
(PERMITTED)**

AutentifikimKontrollor

- ★ POST /api/auth/regjistro
- ★ POST /api/auth/hyr

EventiKontrollor

- ★ GET /api/vente
- ★ GET /api/vente/{id}
- ★ POST /api/vente
- ★ PUT /api/vente/{id}
- ★ DELETE /api/vente/{id}

VleresimiKontrollor

- ★ POST /api/vente/{id}/vleresime
- ★ GET /api/host/{id}/vleresime

Regjistrimi në evente (Bookings)

- ★ POST /api/vente/{id}/regjistrime
- ★ GET /api/permores/{id}/regjistrime
- ★ GET /api/vente/{id}/pjesemarres

Kodi i aplikimit

Në rapport përfshihet kodi kryesor i aplikimit, i organizuar sipas strukturës:

★ Entities

★ Controllers

★ Services

★ Repositories

★ DTO

Përfundime

Projekti “SkillSwap Events Platform” realizon funksionalitetet kryesore për menaxhimin e eventeve peer-to-peer

Sistemi ofron:

- ★ strukturë të qartë backend-i
- ★ API të gatshme për integrim me frontend-in
- ★ validime dhe kontolle logjike (limit, duplikime, pjesëmarrje, rating)

Sistemi ofron:

- ★ autentikim me token (JWT)
- ★ role-based authorization reale (ADMIN/USER)
- ★ pagination/filtering më e avancuar
- ★ upload foto për eventet
- ★ njoftime me email/notifications

RRJEDHAE FUNKSIONALITETEVE KRYESORE

Use Case / Workflow

USE CASE: REGJISTRIMI I PËRDORUESIT (REGISTER)

Aktorë: Përdoruesi

Qëllimi: Të krijohet një llogari e re në sistem.

Parakushte: Përdoruesi nuk duhet të ketë një llogari ekzistuese me të njëjtin email.

Rrjedha kryesore:

1. Përdoruesi plotëson të dhënat: emër, mbiemër, email, fjalëkalim.
2. Aplikacioni kontrollon nëse emaili ekziston në databazë.
3. Nëse emaili nuk ekziston, sistemi krijon përdoruesin dhe e ruan në databazë.
4. Sistemi kthen përgjigje me të dhënat e përdoruesit (ID, emër, email, rol).

Rrjedha alternative: Nëse emaili ekziston: sistemi kthen gabim “Ekziston një përdorues me këtë email”.

Rezultati: Përdoruesi regjistrohet me sukses dhe mund të vazhdojë me hyrjen në sistem.

USE CASE: HYRJA NË SISTEM (LOGIN)

Aktorë: Përdoruesi

Qëllimi:Të autentikohet përdoruesi dhe të lejohet përdorimi i sistemit.

Parakushte:Përdoruesi duhet të jetë i regjistruar.

Rrjedha kryesore:

1. Përdoruesi vendos email dhe fjalëkalim.
2. Sistemi kërkon përdoruesin në databazë sipas email-it.
3. Sistemi verifikon fjalëkalimin.
4. Nëse është i saktë, sistemi e lejon hyrjen dhe kthen të dhënat e user-it

Rrjedha alternative:Nëse email ose password janë të pasakta: kthehet gabim “Email ose fjalëkalim i pasaktë”.

Rezultati: Përdoruesi hyn me sukses dhe mund të përdorë funksionet e sistemit.

USE CASE: LISTIMI I KATEGORIVE TË AFTËSIVE

Aktorë: Përdoruesi / Host

Qëllimi:Të shfaqen kategoritë në mënyrë që eventet të klasifikohen dhe filtrimi të jetë i lehtë.

Parakushte:Kategoritë duhet të ekzistojnë në databazë (seed data ose të shtuara manualisht).

Rrjedha kryesore:

1. Përdoruesi kërkon listën e kategorive.
2. Sistemi i kthen kategoritë në format JSON.
3. Frontend i shfaq si listë ose dropdown.

Rezultati:Kategoritë janë të disponueshme për t'u përdorur gjatë krijimit/filtrimit të eveneve.

USE CASE: KRIJIMI I NJË EVENTI (CREATE EVENT)

Aktorë: Host (përdorues organizator)

Qëllimi: Të krijohet një event i ri me detaje të plota.

Parakushte:

1. Host duhet të ekzistojë si përdorues në databazë.
2. Duhet të zqidhet një kategori e vlefshme.

Rrjedha kryesore:

1. Host plotëson të dhënat e eventit: titull, përshkrim, datë, orë, vendndodhje, kategori dhe kufi pjesëmarrësish.
2. Sistemi verifikon që të dhënat janë të plota.
3. Sistemi kontrollon që data të mos jetë në të kaluarën.
4. Sistemi krijon eventin me status AKTIV.
5. Eventi ruhet në databazë dhe kthehet si përgjigje.

Rezultati: Eventi krijohet me sukses dhe shfaqet në listën e eventeve.

USE CASE: LISTIMI DHE FILTRIMI I EVENTEVE

Aktorë: Përdoruesi

Qëllimi:Përdoruesit të shohin eventet aktive dhe t'i filtrojnë sipas nevojës.

Parakushte:Duhet të ekzistojnë evente në databazë.

Rrjedha kryesore:

1. Përdoruesi kërkon listën e eventeve.
2. Sistemi kthen vetëm eventet me status AKTIV.
3. Opsionalisht, përdoruesi mund të filtrojë:
 - sipas kategoriId
 - sipas kërkimit me tekst (kerkimi)

Rezultati: Përdoruesi gjen eventin që i intereson më shpejt.

USE CASE: REGJISTRIMI NË EVENT (JOIN EVENT) + KONTROLLI I LIMITIT

Aktorë: Përdoruesi (pjesëmarrës)

Qëllimi: Përdoruesi të marrë pjesë në një event, duke respektuar kufirin e pjesëmarrësve.

Parakushte: Eventi duhet të jetë AKTIV. Përdoruesi nuk duhet të jetë i regjistruar më parë në të njëjtin event

Rrjedha kryesore:

1. Përdoruesi zgjedh eventin dhe klikon “Registrohu”.
2. Sistemi verifikon statusin e eventit (duhet të jetë AKTIV).
3. Sistemi kontrollon që përdoruesi nuk është regjistruar më parë në atë event.
4. Sistemi kontrollon kufirin e pjesëmarrësve: Nëse kufiri është $0 \rightarrow$ pa limit. Nëse kufiri është $> 0 \rightarrow$ numërohen regjistrimet aktive
5. Nëse ka vende të lira, sistemi e ruan regjistrimin.
6. Sistemi kthen përgjigje suksesi.

Rezultati: Regjistrimi ruhet dhe përdoruesi bëhet pjesëmarrës i eventit.

USE CASE: SHFAQJA E EVENTEVE KU USER ËSHTË REGJISTRUAR

Aktorë: Përdoruesi

Qëllimi:Përdoruesi të shohë listën e eventeve ku ka pjesëmarrje aktive.

Parakushte:Përdoruesi duhet të ketë të paktën 1 regjistrim

Rrjedha kryesore:

1. Përdoruesi hap “Eventet e mia”.
2. Sistemi kërkon regjistrimet aktive të user-it.
3. Sistemi kthen listën e eventeve përkatëse.

Rezultati: Përdoruesi sheh eventet ku është pjesëmarrës.

USE CASE: SHFAQJA E PJESËMARRËSVE TË EVENTIT (VETËM HOST)

Aktorë: Host

Qëllimi: Host-i të monitorojë listën e pjesëmarrësve në eventin e tij.

Parakushte: Host-i duhet të jetë pronari i eventit.

Rrjedha kryesore:

1. Host kërkon listën e pjesëmarrësve për një event.
2. Sistemi kontrollon nëse hostId përputhet me host-in e eventit.
3. Sistemi kthen listën e pjesëmarrësve.
4. Rrjedha alternative: Nëse hostId nuk përputhet → gabim “Nuk keni të drejtë”

Rezultati: Host-i sheh pjesëmarrësit e eventit.

USE CASE: VLERËSIMI I EVENTIT / HOST-IT (RATING & FEEDBACK)

Aktorë: Përdoruesi (pjesëmarrës)

Qëllimi: Të sigurohet feedback për eventin dhe host-in.

Parakushte:

- Nuk lejohet vlerësim dy herë për të njëjtin event nga i njëjti përdorues
- Përdoruesi duhet të jetë pjesëmarrës i eventit

Rrjedha kryesore:

1. Përdoruesi vendos vlerësim me yje (1-5) dhe koment.
2. Sistemi kontrollon:
 - user ka regjistrim aktiv në event
 - vlerësimi nuk ekziston më parë
3. Sistemi ruan vlerësimin në databazë.
4. Sistemi mund të llogarisë mesataren e vlerësimeve për host-in.

Rezultati: Vlerësimi ruhet dhe kontribuon në reputacionin e host-it.

Përbledhje teknike e endpoint-eve (API Summary)

Në këtë projekt janë implementuar endpoint-et kryesore:

Auth

★ POST /api/auth/registro

★ POST /api/auth/hyr

Evente

★ GET /api/evente

★ GET /api/evente/{id}

★ POST /api/evente

★ PUT /api/evente/{id}

★ DELETE /api/evente/{id}

Kategori

★ GET /api/kategori

★ POST /api/kategori

Regjistrime

★ POST /api/evente/{id}/regjistrime

★ GET /api/perdorues/{id}/regjistrime

★ GET /api/evente/{id}/pjesemarres

THE END

Tabela e Testimit (Test Case Table)

Këto testime janë realizuar me Postman, duke verifikuar request/response dhe funksionet kryesore të sistemit.

Testet për Autentifikim (Auth)