

# Tehniline Dokumentatsioon: Külaliste Registreerimissüsteem

## 1.Sissejuhatus

### 1.1 Eesmärk

Külaliste registreerimissüsteem on loodud ürituste haldamiseks, võimaldades osavõtjate registreerimist ja andmete haldamist.

### 1.2 Tehnoloogiad

- **Frontend:** Angular versioon 17 koos RxJS BehaviourSubjects olekuhaldusega.
- **Backend:** Java versioon 17, Spring Boot 3, Maven versioon 4 (Sõltuvuste haldus ja rakenduse baas struktuur) järgides Domain-Driven Design (DDD) põhimõtteid.
- **Andmebaas:** MySQL.

## 2. Rakenduse Arhitektuur

### 2.1 Kihtide Struktuur

#### 2.1.1. User Interface kiht / Controller:

- Vastutab HTTP päringute vastuvõtmise ja tagasi saatmise eest.
- Protsessib kasutaja sisendit ning edastab päringud Application kihile.
- Käsitleb päringuid seoses ürituste loendi kuvamise, ürituste lisamise, osavõtjate loendi kuvamise, osavõtja lisamise, kustutamise ja detailandmete vaatamise/muutmisega.
- Valideerib päringuid, näiteks Eesti isikukoode.

### **2.1.2. Application kiht / Services:**

- Vastutab päringute koordineerimise eest.
- Delegeerib päringud edasi Domain ja Infrastructure kihtidele.
- Rakendab funktsionaalsed nõuded, sealhulgas ürituste loendi haldamine, ürituste lisamine, kustutamine, osavõtjate loendi haldamine, osavõtja lisamine, kustutamine ja detailandmete vaatamine/muutmine.

### **2.1.3. Domain kiht / Service:**

- Haldab äriloogikat, sealhulgas üritusi ja osavõtjaid.
- Implementeerib ärireegleid, haldab osavõtjate lisamist üritustele ja säilitab ürituste ja osavõtjate suhteid.

### **2.1.4. Infrastructure kiht / Repository:**

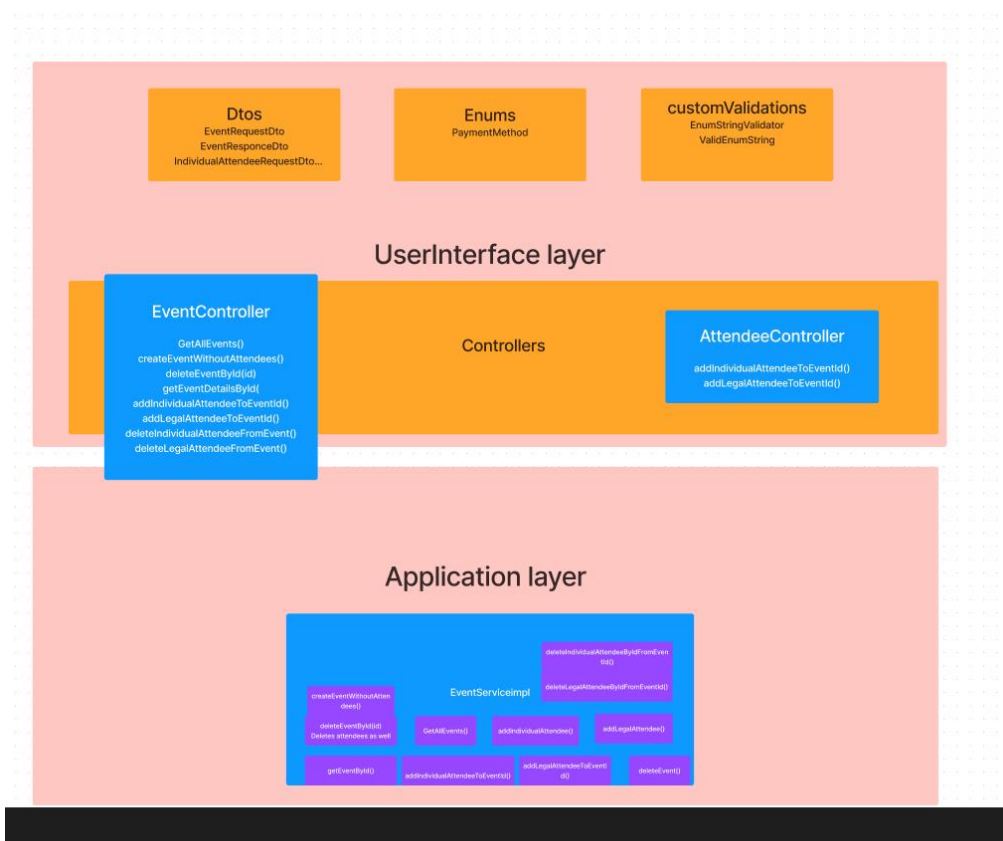
- Suhtleb andmebaasiga, salvestades ja taastades ürituste ja osavõtjate andmeid.
- Hallab makseviiside andmeid ilma lähtekoodi ja andmebaasi kirjeid käsitsi muutmata.
- Tagab andmete püsivuse ja tervikluse.

## **2.2. Andmevahetuse Võimalused**

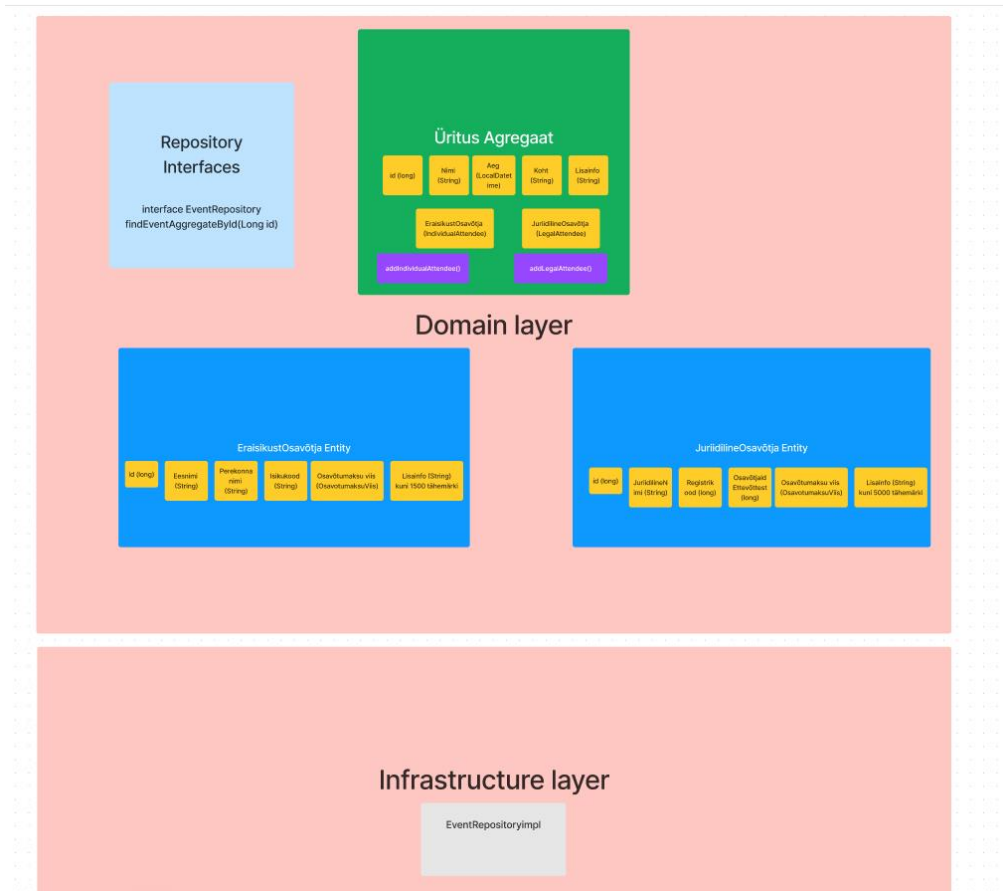
- User Interface kiht saab päringud ja saab vastused JSON-formaadis.
- Application ja Ui kiht koordineerib andmete teisendamist DTO-deks ja vastupidi.
- Domain kiht töötleb andmeid oma sisemises mudelis.
- Infrastructure kiht suhtleb MySql andmebaasiga Spring Data Jpa kaudu

## **2.2 Domeenimudel**

Domeenimudel sisaldab komponente nagu üritus, osavõtja, füüsiline isik ja juriidiline isik vastavalt äriülesande nõuetele.



Team project / Minu ürituste projekt [Free](#)



### 3. Andmebaas

#### 3.1 Andmebaasi Struktuur

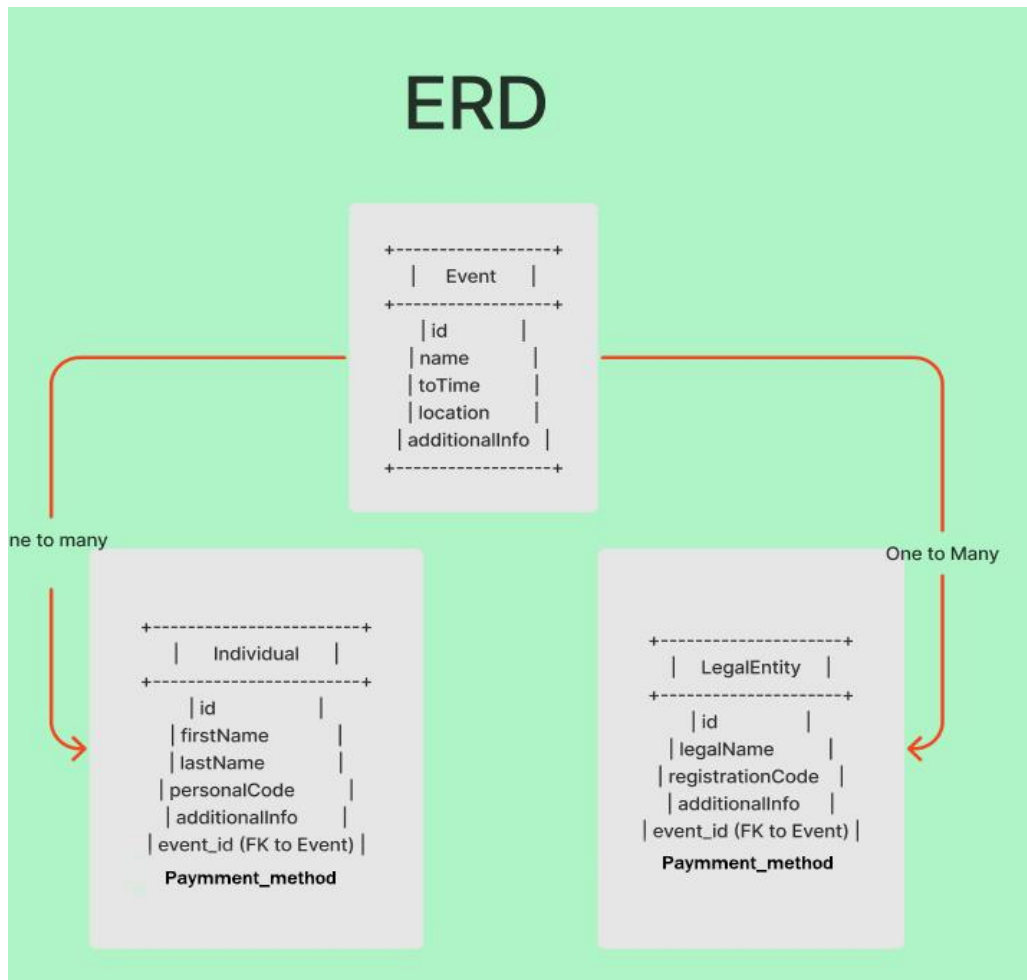
Andmebaasi struktuur koosneb järgmistest tabelitest:

- **event\_table**
- **individual\_attendee\_table**
- **legal\_attendee\_table**

#### 3.2 Andmebaasi Seosed

Andmebaasi seosed määratletakse:

- Üritusel on mitu osavõtjat.
- Osavõtjad jaguneva Eraisikuteks ja juriidilisteks isikuteks



## 4. Kasutajaliides

### 4.1 Avaleht:

\*Kuvab toimunud ja tuleviku ürituste loetelu. (Kuvab ürituse nime, aega, kohta ja osavõtjate arvu.) Ürituse nimele vajutades liigutakse **üritusest osavõtivate isikute vaatesse**.

\*Iga tulevikus toimuva ürituse taga on osavõtja lisamise nupp (Viib **osavõtja lisamise vaatele**), samuti antud ürituse kustutamise nupp.

\*Nupp liikumiseks ürituse lisamise vormile.

### 4.2 Ürituse Lisamise Vaade:

\*Võimaldab sisestada ürituse nime, toimumise aja (ainult tulevikus), toimumise koha ja lisainfo(kuni 1000 tähemärki).

\*Lisamise nupp (Lisa üritus)

\*Tagasi nupp avalehele navigeerimiseks .

\*Pärast ürituse lisamist toimub avalehele tagasi suunamine automaatselt.

### 4.3 Üritusest Osavõtivate Isikute Vaade:

\*Kuvab kõikide antud üritusest osavõtivate isikute loetelu.

Iga isiku kohta ees- ja perekonnanimi (ettevõtte puhul juriidiline nimi), isikukood (ettevõtte puhul registrikood). Osavõtjate Nimele vajutades viib **Osavõtja detailandmete vaatamise/muutmise vaatesse**

\*Iga Osavõtja järel selle ürituselt kustutamise nupp (Kustuta osavõtja ürituselt) ja

\*Tagasi nupp avalehele.

#### 4.4 Osavõtja Lisamise Vaade:

\*Võimaldab valida, kas lisada eraisikut või ettevõtet. (Nupud lisa eraisik, lisa ettevõtte)

Mõlema valiku jaoks vastavate andmete sisestamise võimalus.

Tabel:

Eraisik:

Eesnimi

Perekonnanimi

Isikukood

Osavõtumaksu maksmise viisi valik (pangaülekanne või sularaha).

Lisainfo väli maksimaalselt 1500 tähemärki (eraisik) või 5000 tähemärki (ettevõtte).

Ettevõtte:

Ettevõtte juriidiline nimi

Ettevõtte registrikood

Ettevõttest tulevate osavõtjate arv

Osavõtumaksu maksmise viisi valik (pangaülekanne või sularaha).

Lisainfo väli maksimaalselt 1500 tähemärki (eraisik) või 5000 tähemärki (ettevõtte).

\*Salvesta osavõtja nupp (Toimub automaatne avalehele suunamine)

\*Tagasi nupp avalehele.

## 4.5 Osavõtja Detailandmete vaatamise/muutmise Vaade:

\*Võimaldab vaadata ja muuta osavõtja lisamise vormilt salvestatud andmeid. Avaneb uuesti vaade mis sarnaneb osavõtja lisamise vaatele (Olenevalt kas vajutati ettevõtte või eraisiku omale on vaade vastav)

\*Väljad on eeltäidetud vastava osavõtja andmetega ja vormil on **muuda nupp** uute andmete salvestamiseks.

## 5. Automaattestid

### 5.1 Testide Struktuur

- Automaatsed integratsiooni testid backendi API endpointide kontrollimiseks
- Kõigi üritusega seotud API endpointide kohta on loodud vastavad testid.

### 5.2 Testikava – EventController

**Testi klass: EventControllerTest**

**Märkus:** Testikava eeldab, et andmebaasis on olemas vähemalt üks üritus (ID-ga 1) ja sellele on lisatud vähemalt üks osaleja.

#### **Test 1: testCreateEventWithoutAttendees**

**Eesmärk:** Kontrollib kas saab luua uue ürituse ilma osalejateta.

1. Loob uue ürituse ja saab päringule vastuse.
2. Kontrollib kas ürituse lisamine oli edukas.
3. Pärib andmebaasist lisatud ürituse.
4. Kontrollib kas andmebaasis olev üritus vastab lisatud üritusele.

## Test 2: testGetAllEvents

**Eesmärk:** Kontrollib kas tagastatakse kõikide ürituste andmed.

1. Saada kõikide ürituste andmed.
2. Kontrollib kas päringu vastus on edukas.
3. Deserialiseeri vastus EventResponseDto listiks.
4. Pärib andmebaasist kõikide ürituste arv.
5. Kontrollib kas päringu vastusena saadud ürituste arv kattub andmebaasis olevate ürituste arvuga.

## Test 3: testIndividualAttendeeToEventId

**Eesmärk:** Kontrollib kas saab lisada üritusele individuaalse osaleja.

1. Lisab üritusele individuaalse osaleja.
2. Kontrollib kas osaleja lisati edukalt.
3. Pärib andmebaasist uuesti ürituse ja tagastab selle osalejate arvu.
4. Kontrollib kas osalejate arv on suurenenud.

## Test 4: testLegalAttendeeToEventId

**Eesmärk:** Kontrollib kas saab lisada üritusele ettevõttest osaleja.

1. Lisada üritusele ettevõttest osaleja.
2. Kontrollib kas osaleja lisati edukalt.
3. Pärib andmebaasist uuesti ürituse ja tagastab selle osalejate arvu.
4. Kontrollib kas osalejate arv on suurenenud.

## Test 5: testGetEventDetails

**Eesmärk:** Kontrollib kas saab üksiku ürituse koos andmetega.

1. Lisada uus üritus ja saada selle ID.
2. Saada ürituse andmed kasutades ID-d.
3. Kontrollib kas päringu vastus on edukas.
4. Deserialiseeri vastus EventResponseDto-ks.
5. Pärib andmebaasist lisatud ürituse.
6. Kontrollib kas päringuvastusena saadud üritus vastab andmebaasis olevale üritusele.



## **Test 6: testgetEventDetailsWithAttendeesById**

**Eesmärk:** Kontrollib kas saab üksiku ürituse andmed koos osalejatega.

1. Lisab uue ürituse ja tagastatakse selle ID.
2. Saada ürituse andmed koos osalejatega kasutades ID-d.
3. Kontrollib kas päringu vastus on edukas.
4. Deserialiseeri vastus EventResponseDto-ks.
5. Pärib andmebaasist lisatud ürituse koos osalejatega.
6. Kontrollib kas päringu vastusena saadud üritus vastab andmebaasis olevale üritusele koos osalejatega.

## **Test 7: testDeleteEventById**

**Eesmärk:** Kontrollida, kas saab kustutada ürituse.

1. Lisab uue ürituse ja saada selle ID.
2. Saadab kustutamise päring kasutades ürituse ID-d.
3. Kontrollib kas üritus kustutati edukalt.

## **Test 8: testDeleteIndividualAttendeeFromEvent**

**Eesmärk:** Kontrollib kas saab kustutada individuaalse osaleja ürituselt.

1. Lisada üritusele individuaalne osaleja ja saada tema ID.
2. Saadab kustutamise päringu kasutades ürituse ja osaleja ID-d.
3. Kontrollib kas osaleja kustutati edukalt.

## **Test 9: testDeleteLegalAttendeeFromEvent**

**Eesmärk:** Kontrollib kas saab kustutada ettevõttest osaleja ürituselt.

1. Lisada üritusele ettevõttest osaleja ja saada tema ID.
2. Saadab kustutamise päringu kasutades ürituse ja osaleja ID-d.
3. Kontrollib kas osaleja kustutati edukalt.

## Test 10: testUpdateIndividualAttendeeFromEvent

**Eesmärk:** Kontrollib kas saab uuendada individuaalse osaleja andmeid üritusel.

1. Pärib esimese ürituse andmed koos osalejatega.
2. Saadab individuaalse osaleja ID ja ürituse ID.
3. Saadab uuendamise päringu uuendatud andmetega.
4. Kontrollib kas uuendamine õnnestus.

## Test 11: testUpdateLegalAttendeeFromEvent

**Eesmärk:** Kontrollib kas saab uuendada ettevõttest osaleja andmeid

1. Pärib esimese ürituse andmed koos osalejatega.
2. Saadab individuaalse osaleja ID ja ürituse ID.
3. Saadab uuendamise päringu uuendatud andmetega.
4. Kontrollib kas uuendamine õnnestus.

## 6. Paigaldusjuhised

### 6.1 Nõuded

**Veenduge, et teie süsteem vastab järgmistele nõuetele:**

- Angular: Installitud ja konfigureeritud Angular'i versioon vastavalt rakenduse nõuetele.

### Angular versioon 16

Lisatud sõltuvused:

```
"dependencies": {  
  "@angular/animations": "^16.2.0",  
  "@angular/common": "^16.2.0",  
  "@angular/compiler": "^16.2.0",  
  "@angular/core": "^16.2.0",  
  "@angular/forms": "^16.2.0",  
  "@angular/platform-browser": "^16.2.0",  
  "@angular/platform-browser-dynamic": "^16.2.0",  
  "@angular/router": "^16.2.0",
```

```

    "rxjs": "~7.8.0",
    "tslib": "^2.3.0",
    "zone.js": "~0.13.0"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "^16.2.3",
    "@angular/cli": "~16.2.3",
    "@angular/compiler-cli": "^16.2.0",
    "@types/jasmine": "~4.3.0",
    "jasmine-core": "~4.6.0",
    "karma": "~6.4.0",
    "karma-chrome-launcher": "~3.2.0",
    "karma-coverage": "~2.2.0",
    "karma-jasmine": "~5.1.0",
    "karma-jasmine-html-reporter": "~2.1.0",
    "typescript": "~5.1.3"
  }

```

- **Spring Boot: Installitud ja konfigureeritud Spring Boot vastavalt rakenduse nõuetele.**

#### **Lisatud sõltuvused:**

```

<java.version>17</java.version>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>

<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>

<groupId>com.mysql</groupId>
<artifactId>mysql-connector-j</artifactId>
<scope>runtime</scope>

<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
<optional>true</optional>

```

```

<groupId>io.rest-assured</groupId>
  <artifactId>rest-assured</artifactId>
  <version>5.3.2</version>
  <scope>test</scope>

  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>

  <groupId>com.fasterxml.jackson.datatype</groupId>
  <artifactId>jackson-datatype-jsr310</artifactId>
  <version>2.16.0</version>

<groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-json</artifactId>

  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-validator</artifactId>
  <version>8.0.1.Final</version>

  <groupId>org.modelmapper</groupId>
  <artifactId>modelmapper</artifactId>
  <version>2.4.2</version>

<groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-maven-plugin</artifactId>

```

### **Lisatud application.properties faili:**

```

spring.devtools.restart.enabled=true
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url='Sinu andmebaasi url' ja andmebaasinimi
spring.datasource.username='Sinu andmebaasi kasutaja'
spring.datasource.password='Sinu andmebaasi parool'
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
hibernate.dialect=org.hibernate.dialect.MySQLDialect

```

logging.level.root=INFO //Soovi korral on võimalik logging leveli taset muuta

- **MySQL: Paigaldatud ja seadistatud MySQL vastavalt rakenduse nõuetele.**

\*Loo enda andmebaasi editoris sama nimega andmebaas mille lisasid Spring boot properties faili.

\*Tabelid ja seosed lisatakse Spring Data JPA poolt

## 6.2 Paigaldusprotsess

Järgige allpool toodud juhiseid, et paigaldada ja konfigureerida rakendus. Veenduge, et järgite samme täpselt.

Hoidla Lingid:

<https://github.com/Tanel12345/EventProjectBackend.git>

<https://github.com/Tanel12345/EventProjectFrontend.git>

### 1. Angular Rakenduse Paigaldamine:

- a. Avage käsurida ja navigeerige Angular rakenduse kausta.
- b. Käivitage järgmised käsud: npm install  
See installib vajalikud sõltuvused.
- c. Käivitage rakendus kohalikus arendusrežiimis: ng serve

Angular rakendus käivitatakse vaikimisi aadressil <http://localhost:4200/>.

### 2. Spring Boot Rakenduse Paigaldamine:

- a. Avage oma lemmik IDE ja avage Spring Boot rakenduse projekt.
- b. Veenduge, et rakendusel on korralikud seaded application.properties või application.yml failis, sealhulgas andmebaasiühenduse konfiguratsioon.
- c. Käivitage Spring Boot rakendus.

### **3. MySQL Andmebaasi Seadistamine:**

- a. Installige MySQL vastavalt oma opsüsteemi juhenditele.
- b. Loo andmebaas vastavalt rakenduse seadetele.
- c. Veenduge, et andmebaasiühenduse konfiguratsioon Spring Boot rakenduses on õige.

### **4. Rakenduse Konfigureerimine:**

- a. Avage rakendus veebibrauseris.
- b. Lähtestage andmebaas ja looge vajalikud tabelid käivitades rakenduse esmakordselt.

### **5. Rakenduse Kasutamine:**

- a. Avage veebibrauser ja minge rakenduse aadressile (Angular vaikimisi <http://localhost:4200/>).
  - b. Järgige avalehel olevaid juhiseid ürituste lisamiseks, osavõtjate registreerimiseks ja muudeks tegevusteks.
- Nüüd on teie rakendus edukalt paigaldatud ja konfigureeritud. Veenduge, et kõik töötab korralikult ja nautige külaliste registreerimissüsteemi!

## 7. Mittefunktsionaalsed Nõuded

- Veebirakendus peab olema realiseeritud C# programmeerimiskeeles uusimat .NET raamistikku kasutades.
- Arendusvahendina on soovitatav kasutada uusimat Visual Studio versiooni (viimane Visual Studio Community versioon on alla laaditav siit: <https://visualstudio.microsoft.com/downloads/>)
- Rakenduse kood peab olema vigadeta kompileeruv ning eelneva seadistusega Visual Studiost avatav ning käivitav.
- Kasutajaliides peab vastama HTML5 standardile ning kasutajaliidese loomisel on soovitatav võtta aluseks mõni raamistik (näiteks Bootstrap).
- Kasuta töös avalikult kättesaadavat Git repositooriumi (näiteks üks kolmest kõige levinumast: GitHub, GitLab või Bitbucket). Töö käik peab olema hindaja poolt versioonihaldusest selgelt jälgitav.

## 8. Lõpetuseks

Probleemide korral võtta ühendust Tanel Sepp'aga.  
Tel:56681555