Installationsanleitung – Social Media Backend

1️⃣ Voraussetzungen

- **Docker & Docker Compose installiert**
- **Java 21 (für Spring Boot, optional wenn du Docker für Backend nutzt)**
- **Maven (optional, wenn du nicht Docker für Backend nutzt)**
- **Internetzugang für GitHub Clone und Docker Images**

2️⃣ Repository klonen

1. **Öffne ein Terminal.**
2. **Repository klonen mit:**

```
git clone https://github.com/AlenArandjelovic/Social-Media-Projekt_M295.git
```

3. **In das Projektverzeichnis wechseln:**

```
cd Social-Media-Projekt_M295/socialmedia-backend
```

3️⃣ MariaDB / MySQL starten im Docker-Container

Führe folgenden Befehl aus, um die Datenbank zu starten:

```
docker run -d \
 --name socialmedia-db \
 -e MYSQL_ROOT_PASSWORD=deinPasswort \
 -e MYSQL_DATABASE=socialmedia \
 -e MYSQL_USER=alen \
 -e MYSQL_PASSWORD=deinPasswort \
 -p 3306:3306 \
 mariadb:11
```

- **MYSQL_ROOT_PASSWORD → Passwort für Root**
- **MYSQL_DATABASE → Name der Datenbank (socialmedia)**

- **MYSQL_USER → DB-Benutzer (alen)**
- **MYSQL_PASSWORD → Passwort für Benutzer alen**
- **-p 3306:3306 → Port auf Host, damit Spring Boot darauf zugreifen kann**

✅ **Hinweis: Nach dem Start läuft die Datenbank auf localhost:3306.**

Optional für persistente Daten:

```
docker run -d \
 --name socialmedia-db \
 -e MYSQL_ROOT_PASSWORD=deinPasswort \
 -e MYSQL_DATABASE=socialmedia \
 -e MYSQL_USER=alen \
 -e MYSQL_PASSWORD=deinPasswort \
 -p 3306:3306 \
 -v socialmedia-data:/var/lib/mysql \
 mariadb:11
```

🚀 **Spring Boot Backend starten**

Variante 1: Lokal mit Maven

1. **Terminal öffnen und ins Backend-Verzeichnis wechseln:**

```
cd Social-Media-Projekt_M295/socialmedia-backend
```

2. **Dependencies herunterladen & Projekt bauen:**

```
mvn clean install
```

3. **Spring Boot starten:**

```
mvn spring-boot:run
```

✅ **Das Backend läuft standardmäßig auf http://localhost:8080**

Variante 2: Docker-Container für Backend (optional)

1. **Dockerfile im Backend-Verzeichnis erstellen (falls noch nicht vorhanden)**
2. **Image bauen:**

```
docker build -t socialmedia-backend .
```

3. **Container starten:**

```
docker run -d \
 --name socialmedia-backend \
 -p 8080:8080 \
 -e SPRING_DATASOURCE_URL=jdbc:mysql://host.docker.internal:3306/socialmedia \
 -e SPRING_DATASOURCE_USERNAME=alen \
 -e SPRING_DATASOURCE_PASSWORD=deinPasswort \
 socialmedia-backend
```

- **Hinweis: Wenn DB in separatem Container läuft, host.docker.internal (Windows/Mac) oder IP des DB-Containers (Linux) verwenden**

5️⃣ API testen

Du kannst die Endpoints mit Insomnia, Postman oder Browser testen:

Methode	URL	Beschreibung
GET	http://localhost:8080/users	Alle User abrufen
POST	http://localhost:8080/users	User erstellen
GET	http://localhost:8080/posts	Alle Posts abrufen
POST	http://localhost:8080/posts	Post erstellen
POST	http://localhost:8080/posts/{postId}/comments	Kommentar erstellen

Beispiel JSON für POST /users:

```
{
 "username": "foo",
 "email": "foo@example.com"
```

}

Beispiel JSON für POST /posts:

```
{
 "title": "Mein erster Post",
 "content": "Hallo Welt!",
 "userId": 1
}
```

Beispiel JSON für POST /posts/{postId}/comments:

```
{
 "content": "Toller Post!",
 "userId": 1
}
```

6️⃣ Häufige Probleme

Problem	Lösung
Port 8080 belegt	Ändere server.port in application.properties
Keine Tabellen gefunden	spring.jpa.hibernate.ddl-auto=update setzen