

# WISS Forum – Projektdokumentation

**Modul:** 223, Multiuser-Systeme mit Rollen

**Datum:** 15.05.2025

## Teammitglieder & Beiträge:

- Yanis Sebastian Zürcher - Projektleitung, vollständige Umsetzung von Backend & Frontend, Rollen- & Authentifizierungssystem, Dokumentation
- Jason Bichsel - Unterstützung durch Recherche und Feedback (Gruppenwechsel zu uns)
- Dominik Könitzer - Unterstützung durch Recherche und Feedback

## Inhaltsübersicht

- [1. Projektprospekt](#)
- [2. Entwicklungstimeline](#)
- [3. Installationsanleitung](#)
- [4. Benutzeranleitung](#)
- [5. Technologie-Stack](#)
- [6. Architektur & Hauptkomponenten](#)
- [7. Feature-Übersicht](#)
- [8. Sicherheit & Datenschutz](#)
- [9. Fazit / Reflexion](#)
- [10. Anhang](#)

## 1. Projektprospekt

Warum das WISS Forum die perfekte Lösung für moderne schulische Kommunikation und Motivation ist.

### Problem -> Lösung

Herausforderung an Schulen	Wie das WISS Forum hilft
Verstreute Kommunikation über E-Mails & Chat-Gruppen	Zentrale, durchsuchbare Diskussionsplattform -- alles an einem Ort, transparent & nachvollziehbar
Schwierige Praktikumssuche	Sichtbarkeit durch aktives Profil & Platzierung im Leaderboard -- Engagement kann überzeugen
Geringe Beteiligung im Unterricht	Gamifiziertes Punktesystem & Ranking motivieren zur aktiven Teilnahme

## Was macht das WISS Forum einzigartig? (USP)

- 👤 Rollenbasiertes System -- klare Funktionen für Studenten, Lehrpersonen und Admins
- 🎨 Anpassbares UI -- Light Mode, Dark Mode & sogar ein Steam Theme für Gamer\*innen
- 🔧 Technisch erweiterbar -- moderne Architektur (React + REST API) ermöglicht neue Features, z. B.:
  - Automatische Praktikumsempfehlungen auf Basis von Beiträgen
  - Interne Umfragen & Feedback-Tools

## Der Mehrwert für die Schule

Das WISS Forum löst nicht nur das Kommunikationsproblem – es bindet Schüler aktiv ein, fördert digitale Selbstverantwortung und bietet eine Plattform, auf der sich Engagement zeigt und belohnt wird. Arbeitgeber könnten künftig direkt Leaderboard-Profile mitdiskutierter Beiträge einsehen – eine neue Art von Lebenslauf.

## 2. Entwicklungstimeline

Die komplette Commit-Historie und Entwicklung des Projekts können Sie im [GitHub Repository](#) nachverfolgen.

### Wochenübersicht

KW	Zeitraum	Wöchentliche Highlights
09 (24. 02 – 02. 03)	Projektstart: Initial Commit & Lizenzdatei.	
10 (03. 03 – 09. 03)	Grundgerüst aufgesetzt – React + TypeScript-Frontend, Express/Node-Backend & MongoDB verbunden; erste Auth-Routen.	
11 (10. 03 – 16. 03)	Feature-Sprint: Routing, Nutzerverzeichnis & -profile, Latest-Topics-Feed, Statistik-API, Benachrichtigungssystem, erste Admin-Bootstrap-Funktionen.	
12 (17. 03 – 23. 03)	Auth-Härtung (Refresh-Token, AuthContext-Refactor), Steam-Theme, zahlreiche UX- & Performance-Optimierungen.	
13 – 14 (24. 03 – 06. 04)	– keine Commits – Vorbereitung der Backend-Neuausrichtung.	
15 (07. 04 – 13. 04)	Grosse UI-Überarbeitung: Navbar/SideNav/Footer refactored, Toaster & Skeleton-Loader implementiert, Topic-View-/Reply-Zähler eingeführt.	

KW	Zeitraum	Wöchentliche Highlights
16 (14. 04 – 20. 04)	Weitere UX-Politur: Versionsanzeige, Login/Register-Rework, Help-Page, Theme-Switch-Optimierungen.	
17 (21. 04 – 27. 04)	Legacy-Bootstrap-Code entfernt, Category/Topic/User-Role-Handling gestrafft, allgemeine Code-Aufräumarbeiten.	
18 (28. 04 – 04. 05)	UI-Housekeeping: Benachrichtigungen entfernt, Toast-Styles aufgeräumt, Port/ENV-Cleanup, Topic-Creation-Flow verfeinert.	
<b>19 (05. 05 – 11. 05)</b>	<b>Grosser Meilenstein: Migration von Express/Node -&gt; Spring Boot</b> (spring init, neue Controller-Struktur, Jackson-Config); Admin-Tools aktiviert, Rollen normalisiert, Pagination-Fixes.	Zu Spring Boot gewechselt
20 (12. 05 – 18. 05)	End-Polish: Docker-Setup lauffähig, Category-Management-UI verbessert, View-Counter & Leaderboard-Fixes, Markdown-/Dok-Bereinigung – bereit für M223-Abgabe.	

## 3. Installationsanleitung

### 3.1 Voraussetzungen

Falls Sie es mit Docker machen wollen:

- **Windows** mit **WSL 2**
- **Docker Desktop (gestartet und ready)**  
Sie können es hier herunterladen falls Sie es nicht schon haben:  
<https://www.docker.com/products/docker-desktop/>

### 3.2 Services starten

DOCKER, DOCKER ENGINE MUSS GESTARTET SEIN

Begeben sie sich in das Verzeichnis des Projekts `wiss-forum` und führen folgenden Befehl aus:

```
docker compose up -d
```

Einfach warten bis die Images geladen sind und die Container gestartet sind.

Wenn es fertig ist, sollten sie ungefähr folgende Ausgabe sehen:

```
[+] Running 5/5
✓ Network wiss-forum_wiss-network    Created
                                0.0s
✓ volume "wiss-forum_mongodb_data"  Created
                                0.0s
✓ Container wiss-forum-mongodb-1     Started
                                4.4s
✓ Container wiss-forum-backend-1     Started
                                1.8s
✓ Container wiss-forum-frontend-1    Started
```

*Wenn Sie es ohne docker machen wollen oder probleme haben (können sie skippen wenn docker funktioniert):*

Voraussetzungen:

- [Node.js](#)
- [npm](#)
- [Apache Maven](#)
- [Java 21 \(JDK 21 LTS\)](#)
- [MongoDB](#)

alles sollte bereit sein.

1. navigieren sie in das Verzeichnis `wiss-forum/backend` und führen folgende Befehle aus:

```
mvn clean install
mvn spring-boot:run
```

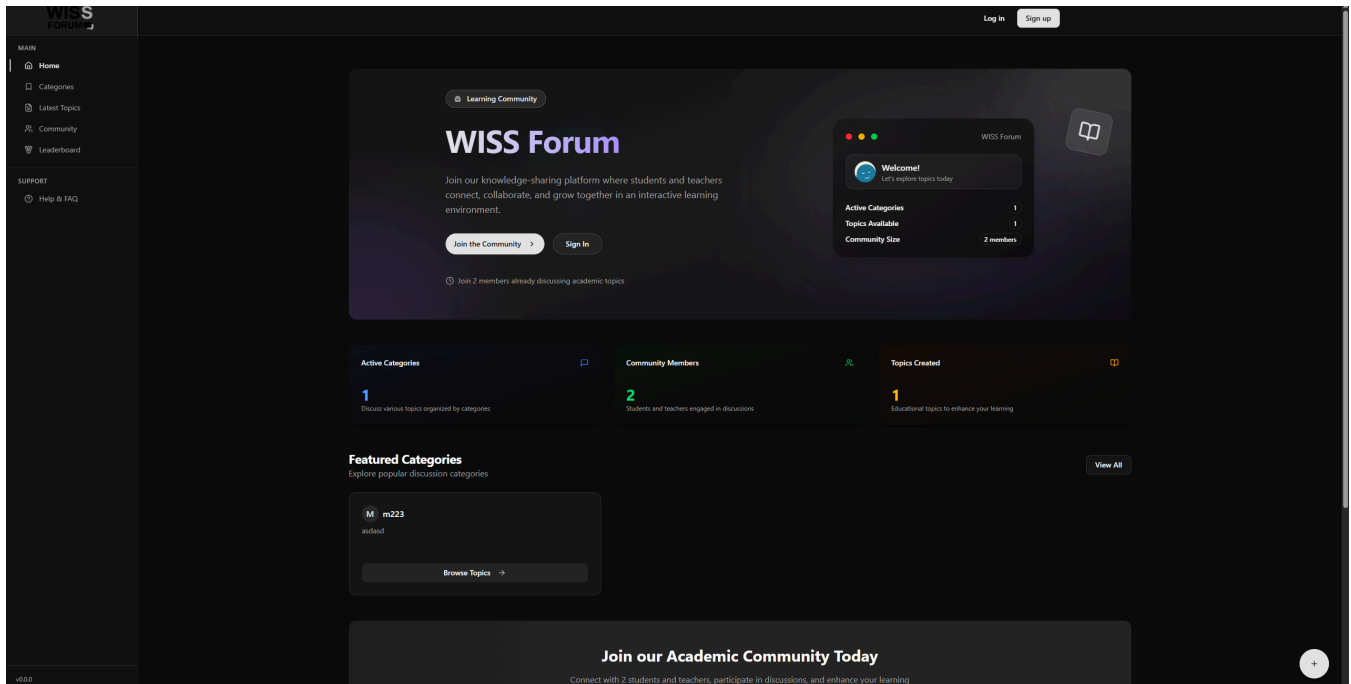
2. navigieren sie in das Verzeichnis `wiss-forum/frontend` und führen folgende Befehle aus:

```
npm install
npm run dev
```

## 3.4 Applikation aufrufen

Jetzt können sie die Applikation unter <http://localhost:3000> aufrufen.

Jetzt sollten sie folgende Seite sehen:



Startseite ohne Login.

## 3.5 Troubleshooting

Problem	Lösung
"Cannot connect to Docker daemon"	Docker Desktop öffnen & Engine starten
Port 3000 / 8080 bereits belegt	killen Sie die Prozesse die den Port 3000 / 8080 benutzen (netstat -ano
Weisse Seite im Browser	console logs im browser anschauen (meistens missing dependencies (wenn man den Namen findet kann man es installieren zb npm install <name> ))
Allgemeine MongoDB Fehler	MongoDB starten und docker container neu starten oder halt manuell starten

## 4. Benutzeranleitung

### 4.1 Erste Schritte – Registrierung

NOTE: Sie sollten wissen das `forgot password` nicht funktioniert in der `Login` Seite, da ich meine Mail konfigurationen nicht teile. Sie können es einfach ignorieren.

1. Rufen Sie <http://localhost:3000> auf.
2. Klicken Sie **Sign Up** -> Formular ausfüllen -> **Register**.

**Create an account**  
Enter your information to create a new account

Display Name  
Yanis Sebastian Zürcher

Username  
lyle01

Email  
yanis.zuercher@wiss-educh

Password  
\*\*\*\*\*

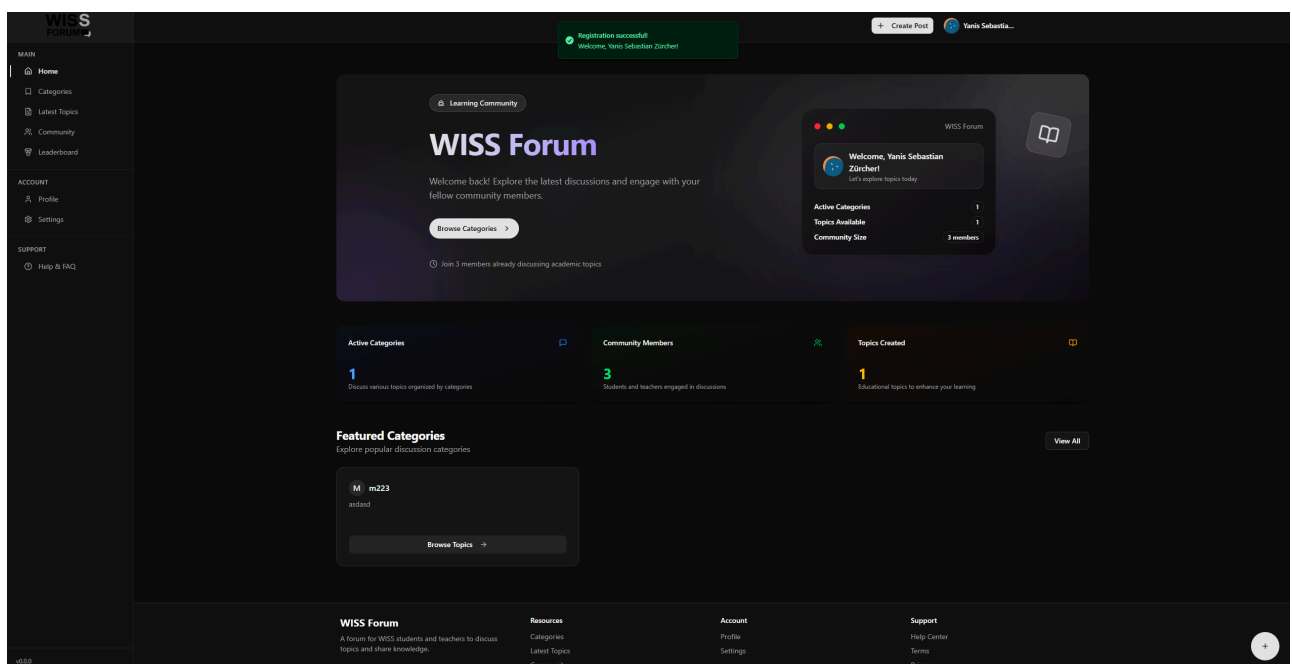
Confirm Password  
\*\*\*\*\*

Sign up

Already have an account? Sign in

## 4.2 Sie sind eingeloggt

- Nach dem Login sehen Sie die authentifizierte sicht.

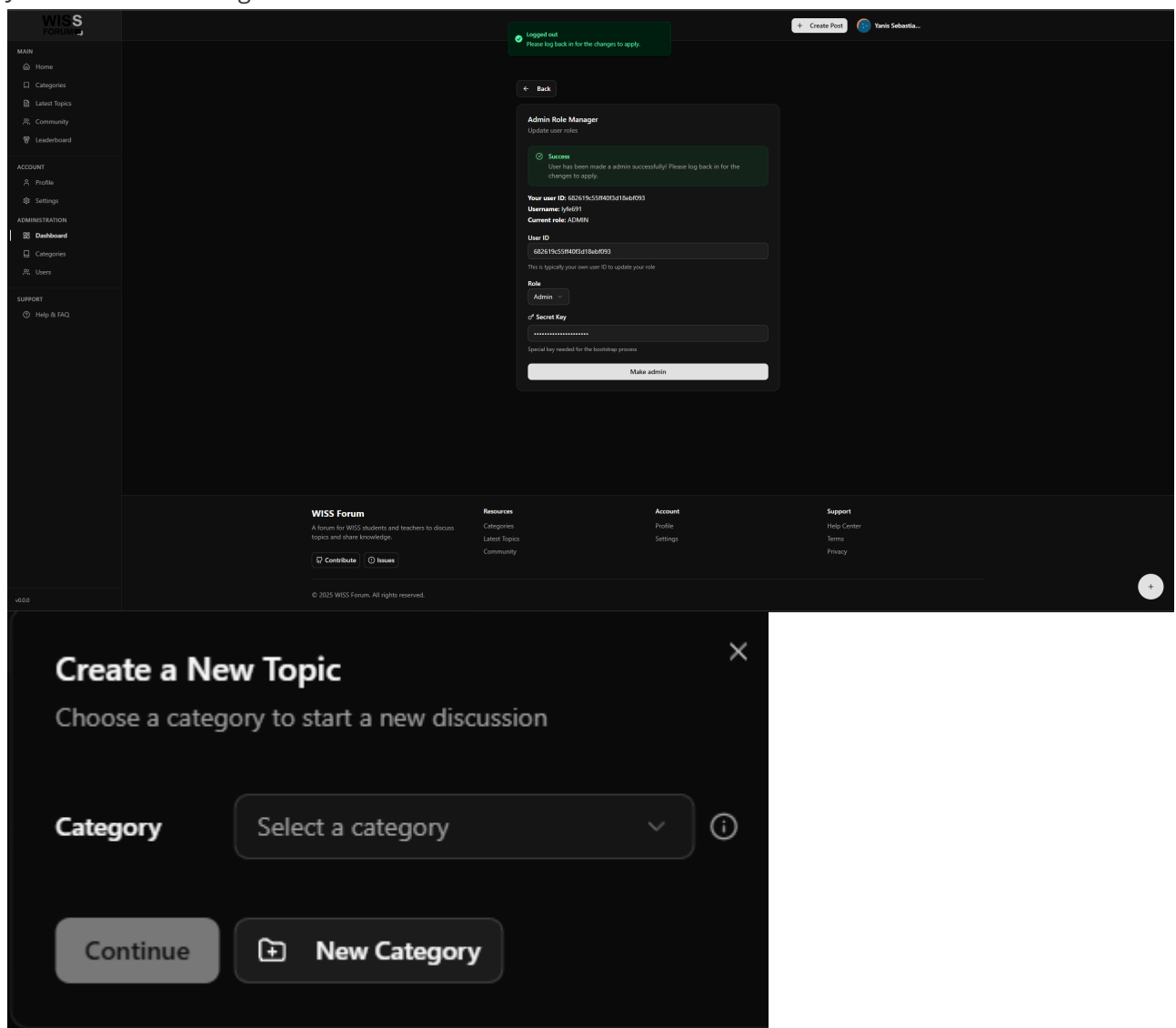


- Klicken Sie auch das **+** unten rechts um einen neuen Beitrag zu erstellen, dann fällt Ihnen auf das es noch gar keine Kategorien gibt.
- Sie können noch keine Beiträge erstellen da Sie die Rolle **student** haben (nur Admins und Teachers können Kategorien erstellen).

## 4.3 Rollen wechseln

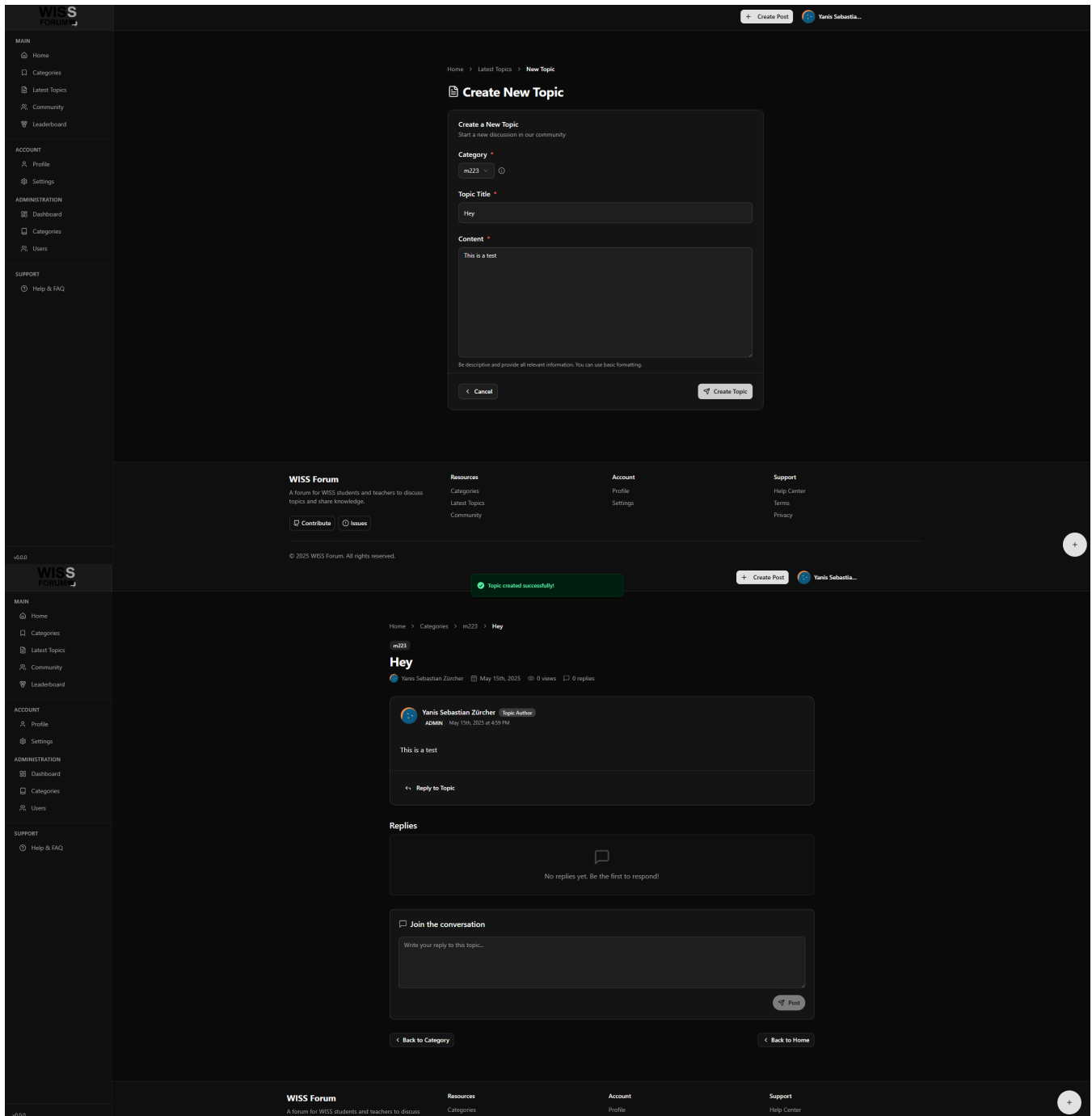
- Navigieren Sie zum Bootstrap Tool <http://localhost:3000/admin-tool>
- Jetzt können Sie beliebig Ihre Rolle wechseln. Der Secret Key ist aus development gründen schon eingetragen für Sie.

3. Nachdem Sie Ihre Rolle zu `Admin` oder `Teacher` geändert haben werden Sie ausgeloggt und müssen sich erneut einloggen.
4. Jetzt können Sie Kategorien erstellen.



## 4.4 Kategorie & Topic erstellen

1. Kategorie erstellen, entweder per `+` unten rechts oder per `localhost:3000/admin/categories`
2. Klicken Sie auf das `+` unten rechts um einen neuen Beitrag zu erstellen.
3. Wählen Sie die Kategorie aus die Sie erstellt haben und füllen Sie das Formular aus.
4. Klicken Sie auf `Create` um den Beitrag zu erstellen.



Sie können beliebig die funktionen testen, wie z.B. `reply`, `like` .





Sie können auch benutzer wechseln mit zb einer anderer rolle und einfach los diskutieren.

Ich empfehle Ihnen mal einfach ein paar Benutzer zu erstellen, zu liken und zu antworten um die Funktionen zu testen.



m223


## Hey

 Yanis Sebastian Zürcher  May 15th, 2025  4 views  1 reply




**Yanis Sebastian Zürcher** Topic Author  
**ADMIN** May 15th, 2025 at 4:59 PM

This is a test

 Reply to Topic

### Replies

 Collapse All




**2nduser** Author  
**STUDENT** May 15th, 2025 at 5:03 PM

Test worked!



1


 Reply

 Hide replies

1



**Yanis Sebastian Zürcher** Author  
**ADMIN** May 15th, 2025 at 5:03 PM


 Replying to @2nduser

Sure did!




0


 Reply

 Join the conversation

Write your reply to this topic...

 Post


 Back to Category

 Back to Home

## 4.5 Profil & Einstellungen

- Display Name, Username, Email, Bio & Passwort unter **Profile** oder **Settings** anpassen.
- Theme-Umschaltung (Light / Dark / Steam) in **Settings**.

### Profil



**Yanis Sebastian Zürcher**

@lyfe691

Admin

✉ yanis.zuercher@wiss-edu.ch

📅 Joined May 15th, 2025

🕒 Last active May 15th, 2025

🔒 Security

🔑 Change Password

👤 Profile🕒 Activity

**Profile Information**

Update your profile information

Username (3-20 characters)

@ lyfe691

Email (@wiss-edu.ch)

✉ yanis.zuercher@wiss-edu.ch

Display Name (3-50 characters)

👤 Yanis Sebastian Zürcher

Bio

I changed my bio!

17/500 characters

💾 Save Changes

Settings (theme)

🟢 Appearance settings saved

+ Create Post

## Settings

Manage your account settings and preferences

- 👤 Account
- 🔒 Password
- 🔗 Appearance

**Appearance**

Customize how the forum looks for you.

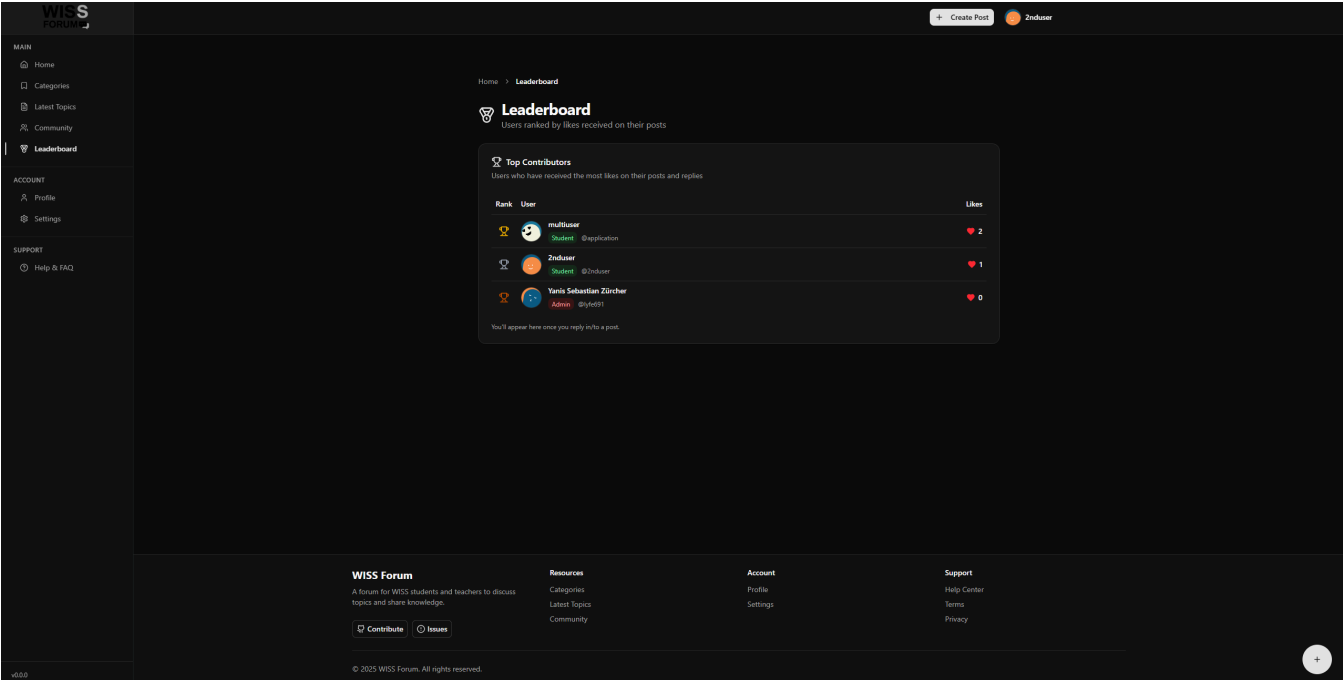
**Theme**

Choose your preferred theme for the forum.

Light

## 4.6 Leaderboard

Sie können den Leaderboard unter `localhost:3000/leaderboard` sehen.

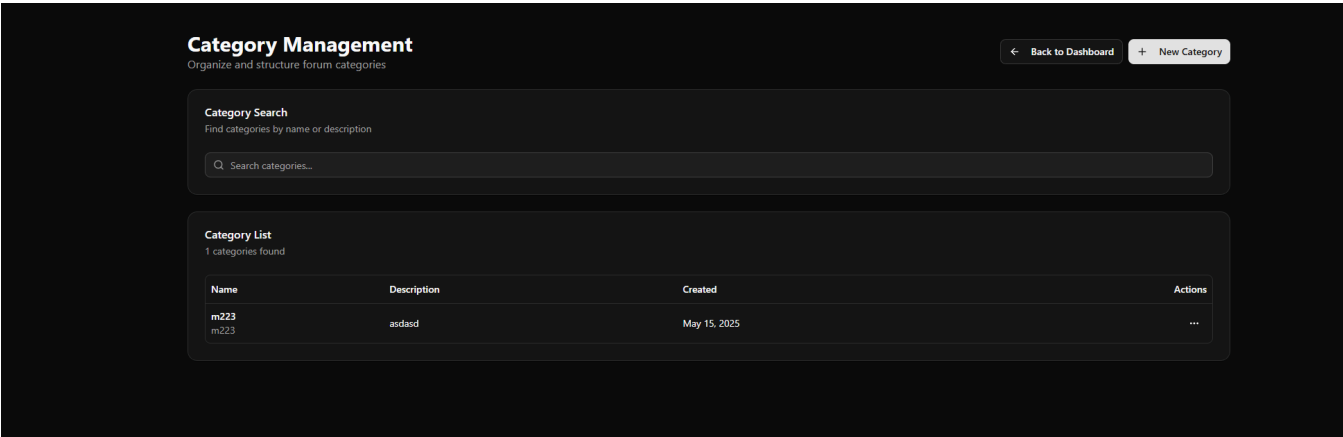


Wie Sie sehen können wird ein gamification angewendet.

## 4.7 Admin-Funktionen (Rolle Teacher)

Ein Teacher hat nur zugriff auf das category management (CRUD).

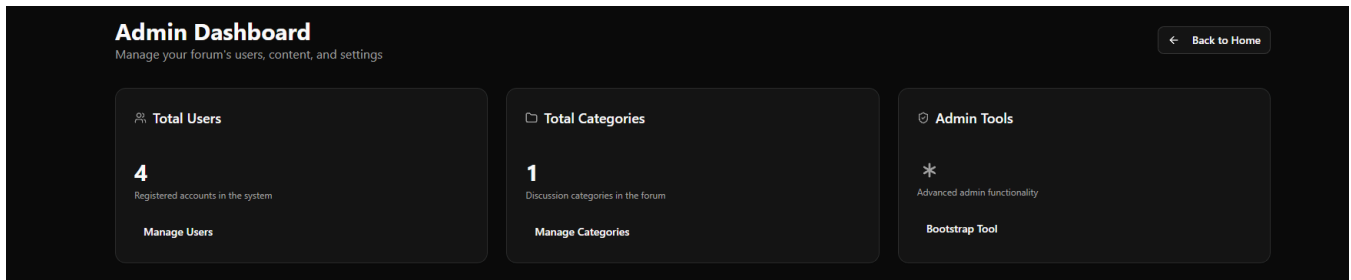
### Category Management



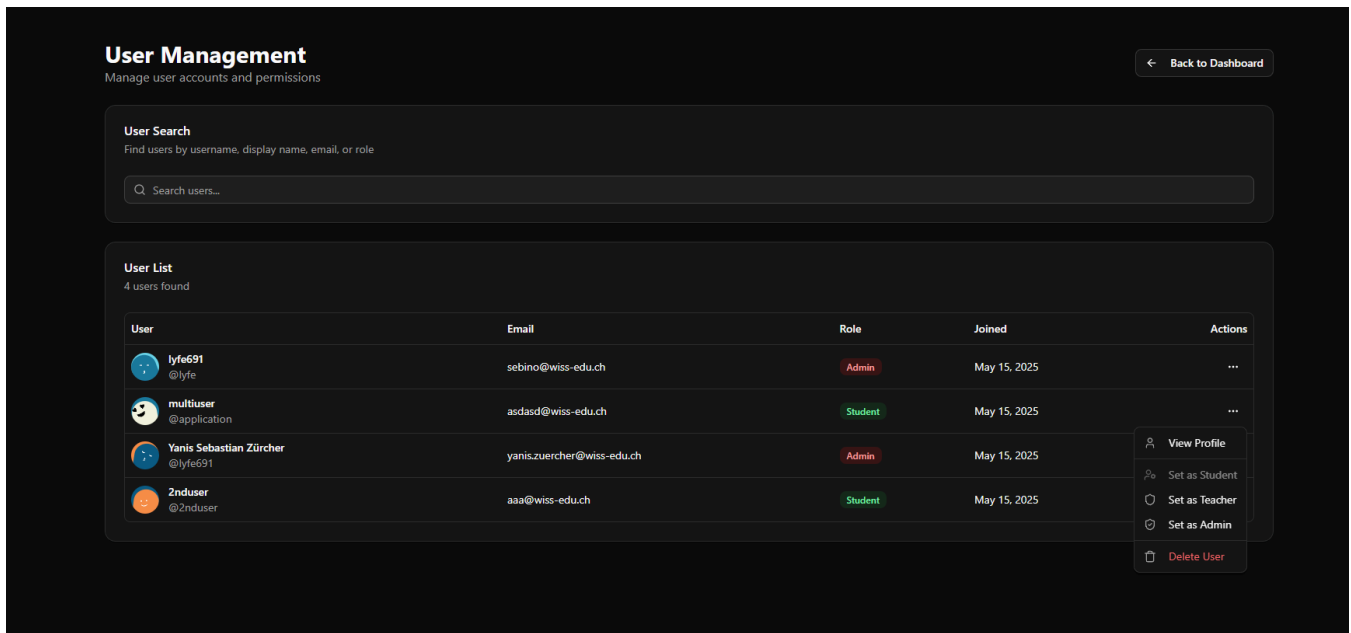
## 4.8 Admin-Funktionen (Rolle Admin)

- **Admin Dashboard** - Benutzer- & Kategorienverwaltung.
- Rollen upgraden / downgraden per Dropdown neben dem Benutzernamen in `http://localhost:3000/admin/users`. Man kann auch Benutzer (Student, Teacher) löschen als Admin.
- Kategorien CRUD unter **Admin -> Categories** in `http://localhost:3000/admin/categories`.

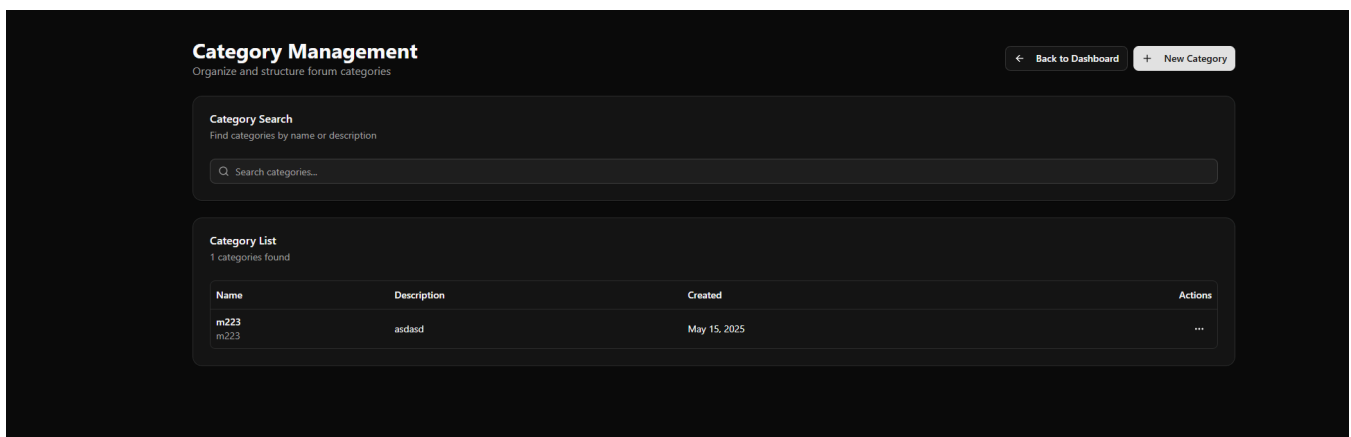
### Admin Dashboard



## User Management



## Category Management



Das waren die Kernfunktionen der app, Sie können sich gerne weitere Funktionen anschauen und ausprobieren wie zb:

- Community: `localhost:3000/community`
- View Count, Reply Count: Sehen sie zb. in `http://localhost:3000/topics/latest`
- etc.

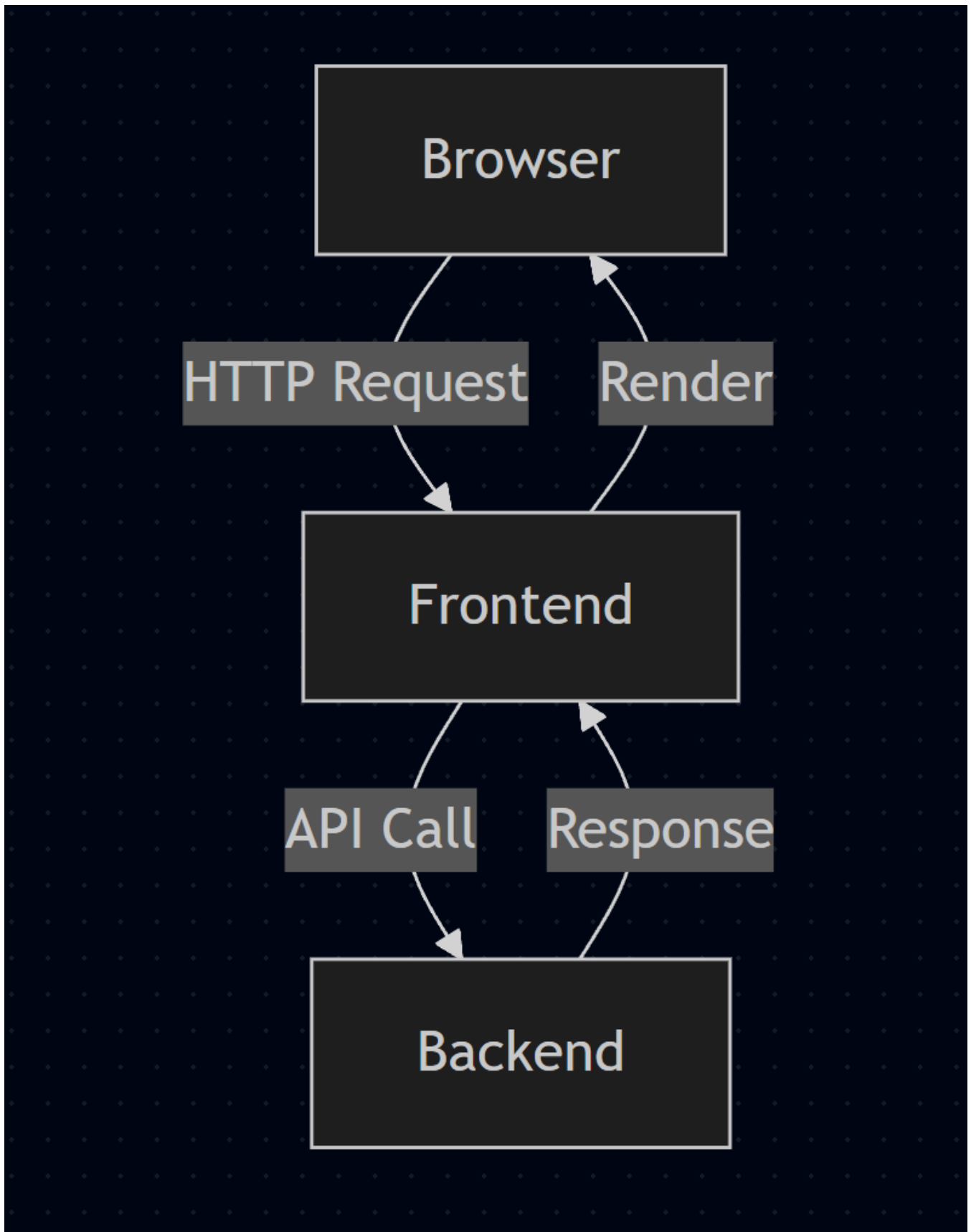
## 5. Technologie-Stack

---

Ebene	Technologie	Kommentar
Frontend	React 18 + TypeScript + Vite	UI - Tailwind CSS + shadcn/ui
Backend	Spring Boot, Maven	REST API, JWT-Auth
DB	MongoDB	Persistenz
Deployment	Docker Compose	Mehrere Container -> One-Click

---

## 6. Architektur & Hauptkomponenten



[ Browser ] ⇄ [ Frontend ] ⇄ REST ⇄ [ Backend ] ⇄ [ MongoDB ]

- **AuthProvider.tsx** hält JWT-Session im React Context.

- **ProtectedRoute.tsx** schützt Seiten & prüft Rollen.
- **AuthContext.tsx** handelt die Authentifizierung und kann einfach verwendet werden mit `useAuth()` hook.
- **roleUtils.ts** ist ein utility um die rollen zu prüfen.
- **api.ts** handelt fast all api requests.

## 7. Feature-Übersicht

Kategorie	Highlights
<b>Benutzerverwaltung</b>	Selbstregistrierung, Rollenwechsel & Deletion über Admin-Tool, Avatare via DiceBear API
<b>Diskussionen</b>	Kategorien, Topics, Replies, Pagination & Accepted Answers
<b>Gamification</b>	Punktesystem basierend auf Aktivität, <b>Leaderboard</b> für Motivation & Sichtbarkeit
<b>Sicherheit</b>	Passwörter mit <b>BCrypt</b> gehasht, JWT in <b>HttpOnly-Cookies</b> , rollenbasierte Zugriffskontrolle
<b>Admin-Funktionen</b>	Benutzerverwaltung, Rollen-Upgrade/Downgrade, Kategorie-Management, Statistiken
<b>Design &amp; Usability</b>	Drei Themes (Light, Dark, Steam) – lokal gespeichert, responsive & performant

## 8. Sicherheit & Datenschutz

- BCrypt-Hashing für Passwörter
- JWT in **HttpOnly-Cookies**
- Rollen-Checks server- & client-seitig ( `roleUtils.tsx` )

## 9. Fazit / Reflexion

Das **WISS Forum** zeigt, wie moderne Webtechnologien, eine modulare Architektur und gezielte UX-Entscheidungen zu einer **skalierbaren und motivierenden Lernplattform** für Schulen kombiniert werden können. Besonders der Gamification-Ansatz hebt die Beteiligung deutlich hervor und bringt einen echten Mehrwert in der schulischen Kommunikation.

Am anfang war es relativ schwer das ganze rollen zeug zu verstehen und zu implementieren. Z.B. die verschiedenen ansichten der Rollen, jedoch nachdem ich eine solidere rollen Implementierung hatte, war es einfach alles zu verstehen und zu implementieren.

Ebenfalls mühsam war die Konvertierung von Node.js/Express zu Spring Boot.

## Learnings

- **Shift-Left Testing** spart langfristig Zeit – die späte Testphase verzögerte den MVP.
  - **Rollenbasierte Features** sollten frühzeitig definiert und eingeplant werden – nachträgliche Änderungen sind aufwändig.
  - **Dokumentation & Ownership** sind entscheidend – eine klare Projektverantwortung verbessert Qualität und Tempo.
- 

## 10. Anhang

---

### 10.1 Projektquellen

- [GitHub Repository](#).
- Verwendete Bibliotheken und Frameworks sind in den jeweiligen package.json und pom.xml Dateien aufgeführt

### 10.2 Nutzermanagement

Standardrolle	Berechtigungen
Student	Lesen, Antworten, liken, Profil bearbeiten
Teacher	Kategorie-Management
Admin	Benutzer-Management, Kategorie-Management

---