

# LOGARG – Logical Argumentation Platform

*Frontend–Backend Integrated System for Structured Debates*

## 1. Overview

LOGARG is a **structured debate and argumentation system** that allows users to explore and build logical argument trees for various topics.

It connects a **Spring Boot backend** (with MySQL database) and an **Angular frontend**, supporting both:

- Real API mode (with live backend)
- Mock API mode (for demo without backend)

## 2. Features

- Turn-based debate between **Proponent** and **Opponent**
- Visualization of argument trees (claims, justifications, rebuttals)
- Move tracking (Challenge, Rebut, Accept, Skip)
- Mock API layer for offline demonstrations
- Backend connected to MySQL with structured schema

## 3. Project Structure

```
LOGARG/
  backendapi/                # Spring Boot Backend
    src/main/java/com/argumentation/backendapi/
      controller/            # REST Controllers
      model/                  # JPA Entities
      repository/            # Repositories
      service/                # Business Logic Services
    pom.xml

  argument-ui/                # Angular Frontend
    src/app/
      components/            # Angular Components
      core/                   # Services
      assets/mock/           # Mock JSON Files
    angular.json
```

## 4. Backend (Spring Boot)

### 4.1. Tech Stack

- Spring Boot 3+

- Spring Data JPA
- MySQL
- Lombok
- Jakarta Persistence

## 4.2. Key Packages

Package	Purpose
controller	Contains REST endpoints for frontend calls
model	Entity classes mapping to MySQL tables
repository	Data access layer using JPA
service	Business logic implementation layer

## 4.3. Database Structure (Main Tables)

Table	Description
topics	Stores discussion topics
statements	Stores claims, justifications, and rebuttals
arguments	Links statements into logical argument structures
premises	Connects justifications to their parent arguments
sources	References or citations used in arguments

## 4.4. How to Run the Backend

1. Open the project in IntelliJ or VS Code (Java).
2. Set MySQL credentials in `application.properties`:

```
spring.datasource.url=jdbc:mysql://localhost:3306/logarg_db
spring.datasource.username=root
spring.datasource.password=your_password
```

3. Start MySQL server.
4. Run the Spring Boot project:

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

5. The backend runs at `http://localhost:8080`

## 5. Frontend (Angular)

### 5.1. Tech Stack

- Angular 17+
- TypeScript
- Bootstrap 5
- RxJS
- HTML / CSS

### 5.2. Key Components

Component	Description
home	Displays list of available topics
argument-view	Main debate area showing argument trees
argument-node	Recursive display of argument nodes
multi-select-dropdown	Justification selection dropdown
move-tracker-box	Displays allowed moves and timer
argument-info-box	Shows argument details and sources

### 5.3. How to Run the Frontend

1. Navigate to the frontend folder:

```
cd argument-ui
npm install
ng serve
```

2. Open the browser and go to: `http://localhost:4200`

## 6. Mock API Mode (for Offline Demo)

The mock API allows the app to run without a backend.

### 6.1. Mock Data Location

```
src/assets/mock/
topics.json
top-arguments.json
arguments-by-topic/
  Television.json
  Social Media.json
```

## 6.2. Switch Between Mock and Backend

In `app.config.ts`:

```
// Use Mock Service (default)
{ provide: ArgumentService, useClass: MockArgumentService }

// To use Backend:
{ provide: ArgumentService, useClass: ArgumentService }
```

**API Reference:** See the attached `LOGARG_Mock_API_Documentation.pdf`.

## 7. Data Flow Summary

**With Backend**

Angular → Spring Boot Controller → Repository → MySQL → Response to Angular

**With Mock API**

Angular → MockArgumentService → `/assets/mock JSON` → Angular UI

## 8. Entity Relationship Overview

Entity	Key Links
TopicEntity	One-to-many with ArgumentEntity
ArgumentEntity	Connected to StatementEntity; has multiple Premises
StatementEntity	Stores textual content (claims, justifications, rebuttals)
PremiseEntity	Composite key mapping between parent and child statements
SourceEntity	Optional references or citations for arguments

## 9. Example Backend Endpoints

Endpoint	Method	Description
<code>/api/topics</code>	GET	Fetch all topics
<code>/api/arguments/{topic}</code>	GET	Get arguments for a given topic
<code>/api/justifications/{id}</code>	GET	Get justifications for an argument
<code>/api/rebuttals/{id}</code>	GET	Get rebuttals for an argument
<code>/api/rebuttals/create</code>	POST	Create new rebuttal