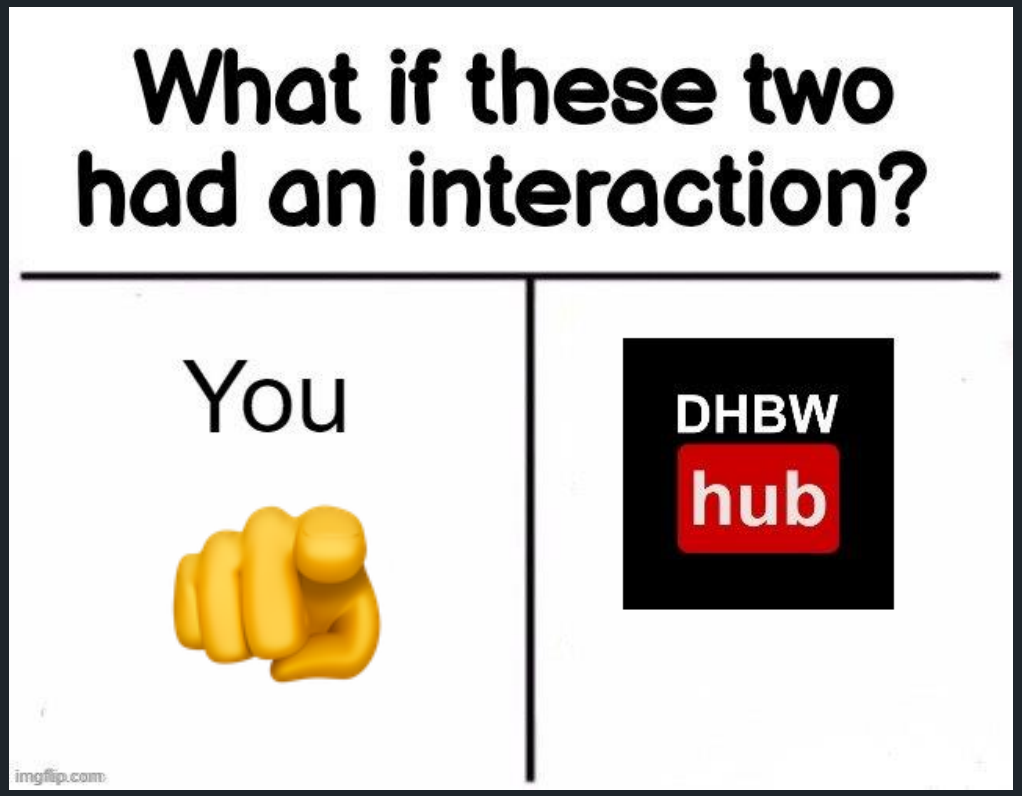
DHBW hub

DENNIS, FURKAN, JONAS, JOSHA, MAX

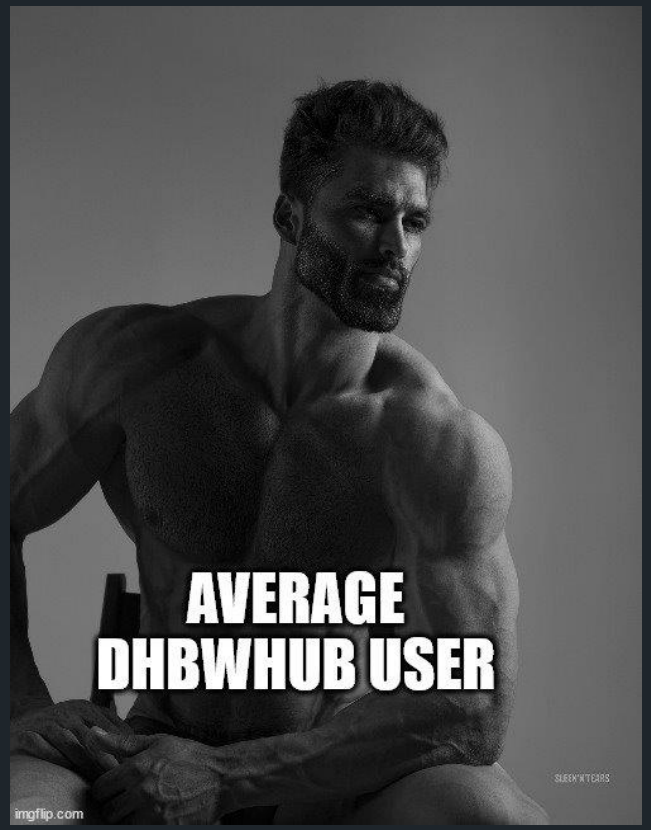
Why use DHBWhub?

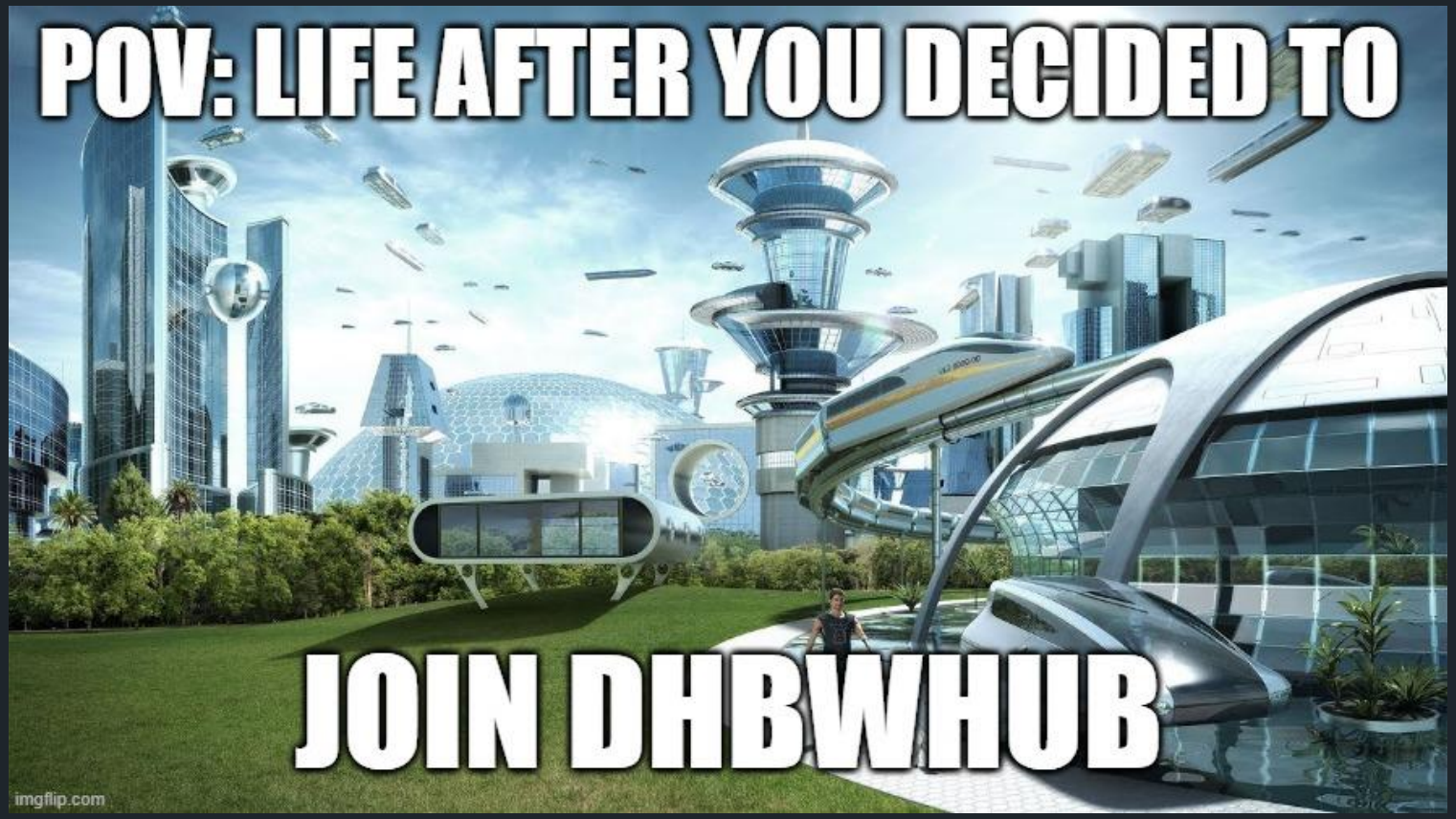
Log in securely

Interact with content



Why use DHBWhub?





Why use DHBWhub?

Keeping track of events

User-friendly design

Architectural design - Goals

Usability

- Intuitive design.
- Visual feedback.

Performance

- Reducing loading time as much as possible.
- Reduce amount of transferred data and data queries.

Modifiability

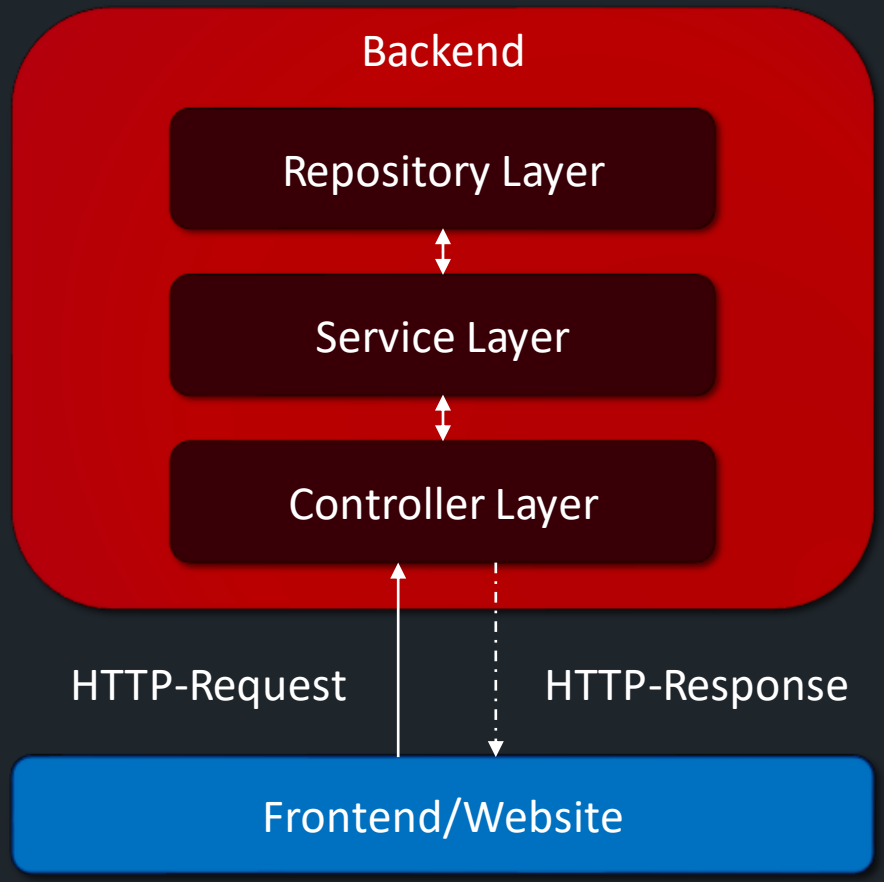
- Creating new features or extending the application shouldn’t be complicated.
- Modularity.

Architectural design - Decisions

REST-architecture

- Website
  - Communicates with the backend through predefined interfaces e.g.

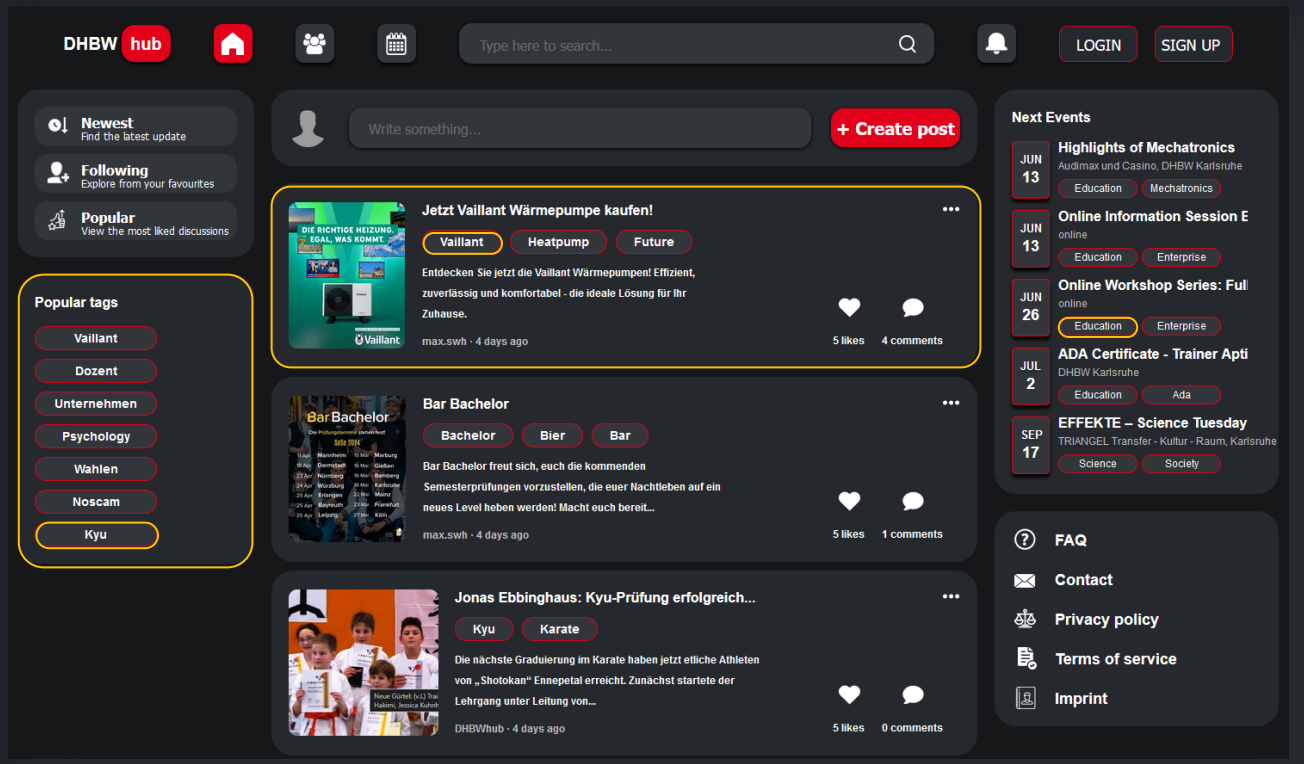
    GET /user/user-information/{id}.
- Backend
  - Controller layer – Data interface for the website.
  - Service layer – Processing of data.
  - Repository layer – Retrieving data from the database.
  - Utility layer - Data mapping and other utilities.



Architectural design - Decisions

Component-based

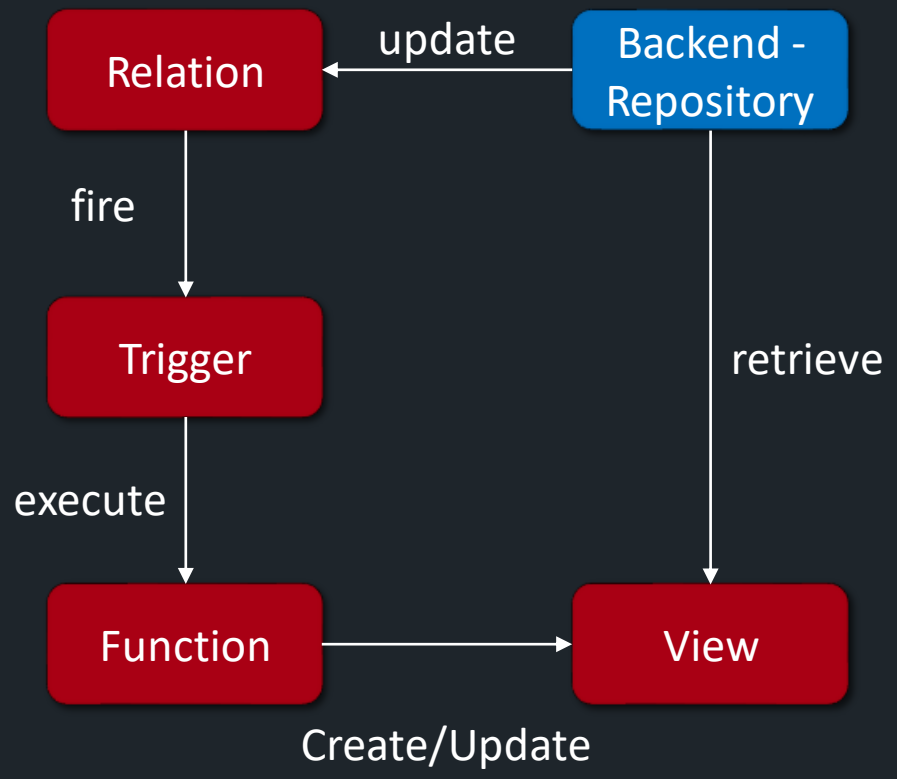
- Splitting the website into smaller components consisting of other smaller components.
- Atoms, molecules, organisms and services.
- Modular built.



Architectural design - Decisions

Database

- External Views
  - Frequent queries or calculations stored in a relation.
- Trigger
  - Executes a trigger function when a table is updated.
- Trigger functions
  - Contains the logic.
  - In our case, the creation of certain views.



Design patterns

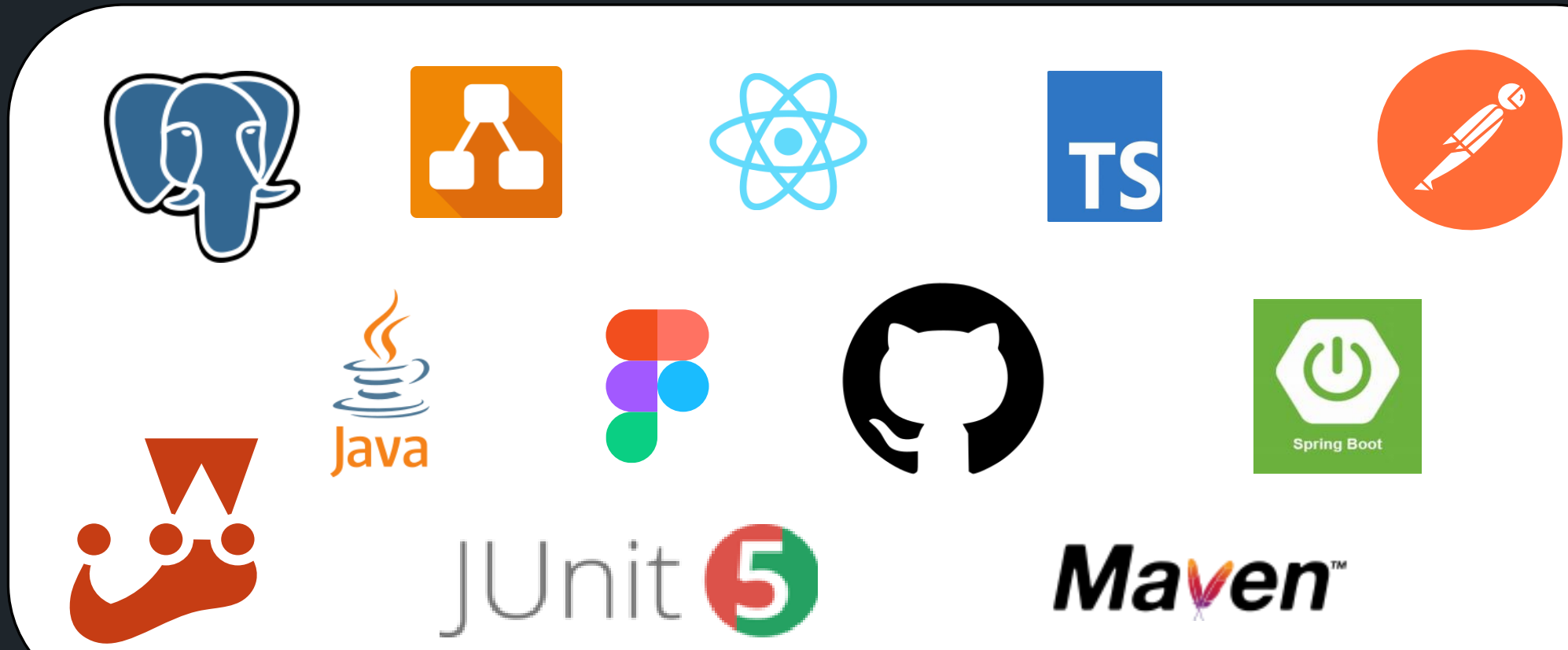
MVC-design

- Separation of the application into model, view and controller.
- Facilitates maintenance and expansion of the application.

Single responsibility principle

- Each method has a single task.
- Increases the clarity and maintainability of the code.

Tech Stack



Quality Assurance - Testing

Backend tests

Frameworks:

- JUnit5
- Mockito
- Maven

Focus:

- Correct transformation of data

```
[INFO] Results:
[INFO]
[INFO] Tests run: 290, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  25.123 s
[INFO] Finished at: 2024-06-10T22:14:11+02:00
[INFO] ------------------------------------------------------------------------
```

Quality Assurance - Testing

Frontend tests

Frameworks:

- Jest
- React

Focus:

- Accurate rendering of objects
- Correct function-call logic

```
▶ Run npm test

> dhbw-hub@0.1.0 test
> jest

PASS src/services/tests/ProfilePictureService.test.tsx (8.202 s)
PASS src/services/tests/AuthService.test.tsx (8.21 s)
PASS src/services/tests/TagService.test.tsx (8.331 s)
PASS src/services/tests/DescriptionService.test.tsx
PASS src/services/tests/TimeService.test.tsx

Test Suites: 5 passed, 5 total
Tests:       41 passed, 41 total
Snapshots:   0 total
Time:        9.45 s
Ran all test suites.
```

Quality Assurance - Testing

Database tests

Quality Assurance - Clean Code

Principle of Least Astonishement

- Methods should be **self-explanatory**
- Comment code logic when needed

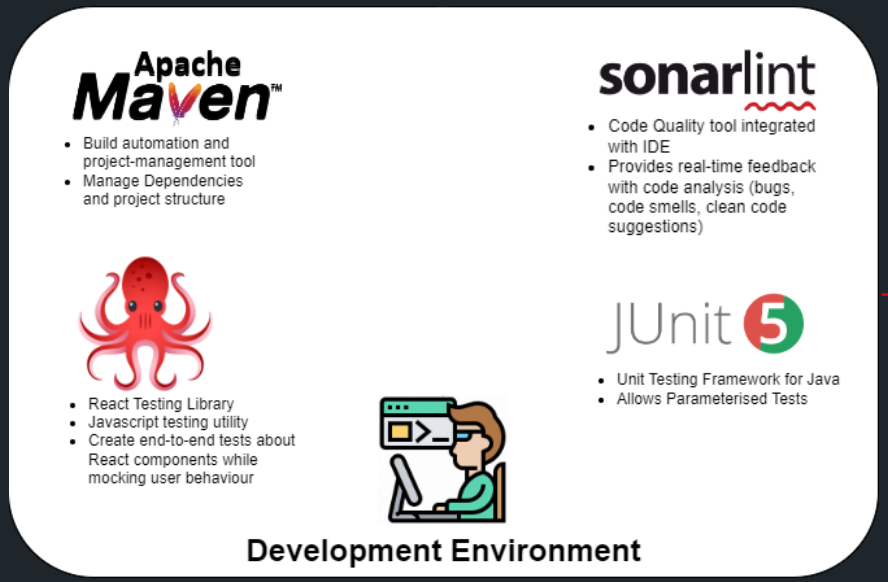
Single Responsibility Principle

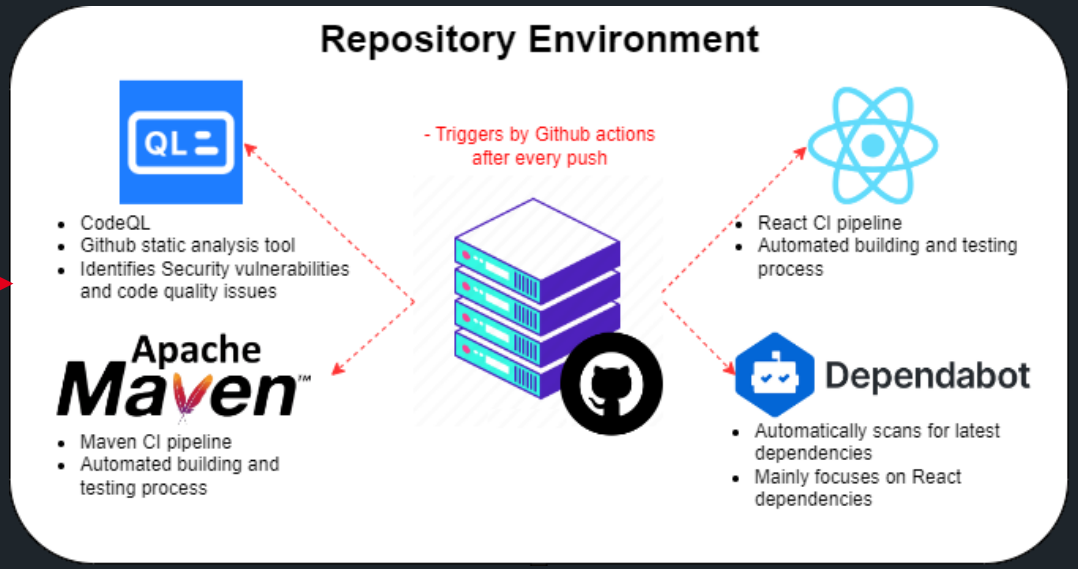
- Each method is responsible for one particular action

Conventions

- CamelCase
- Links divided by hyphens

Project management – CI/CD





Live demonstration

Live demonstration

Search function

Follow users

Calendar and Eventposts

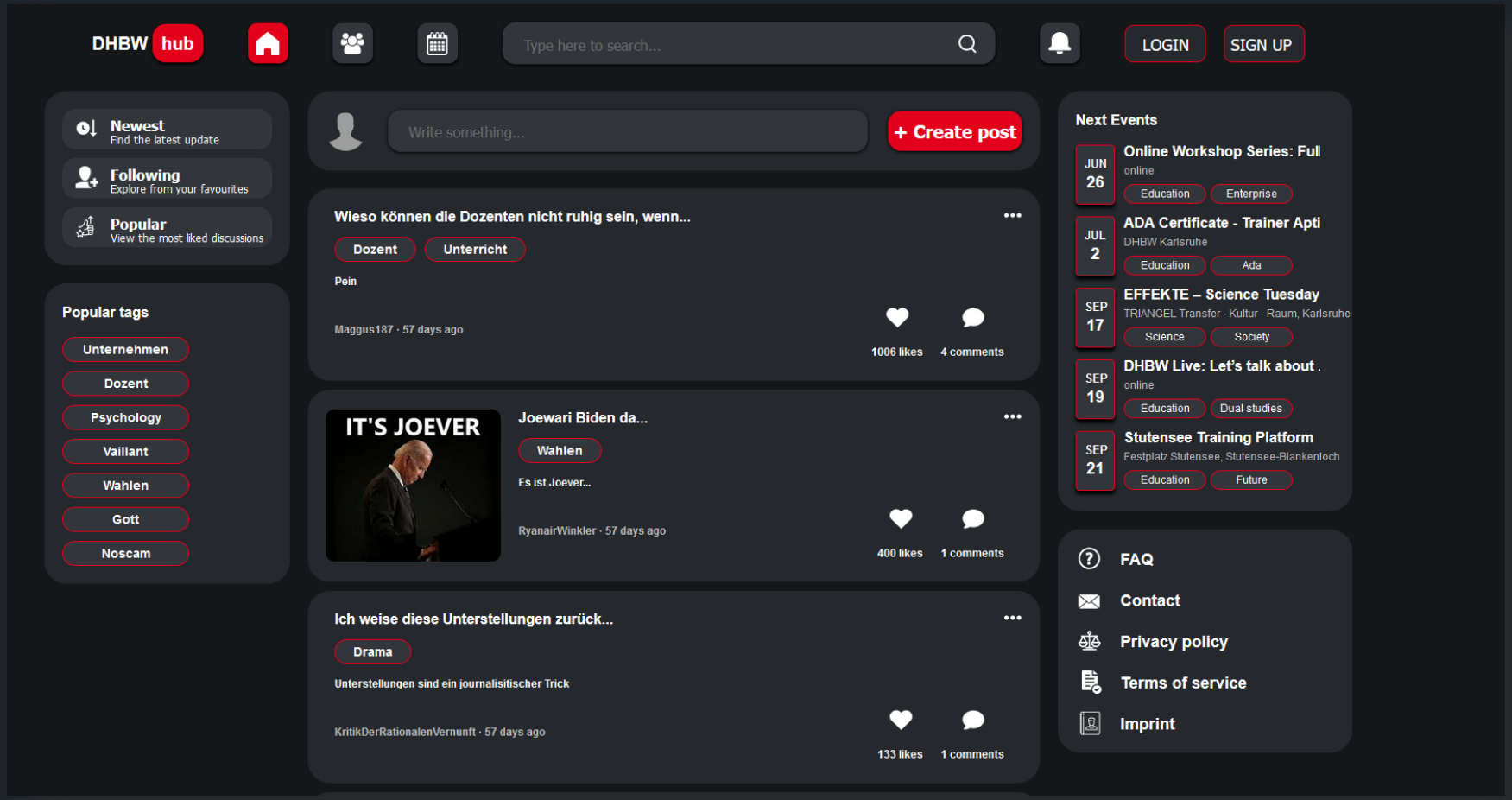
Highlights of DHBWhub (Demo)

Notifications

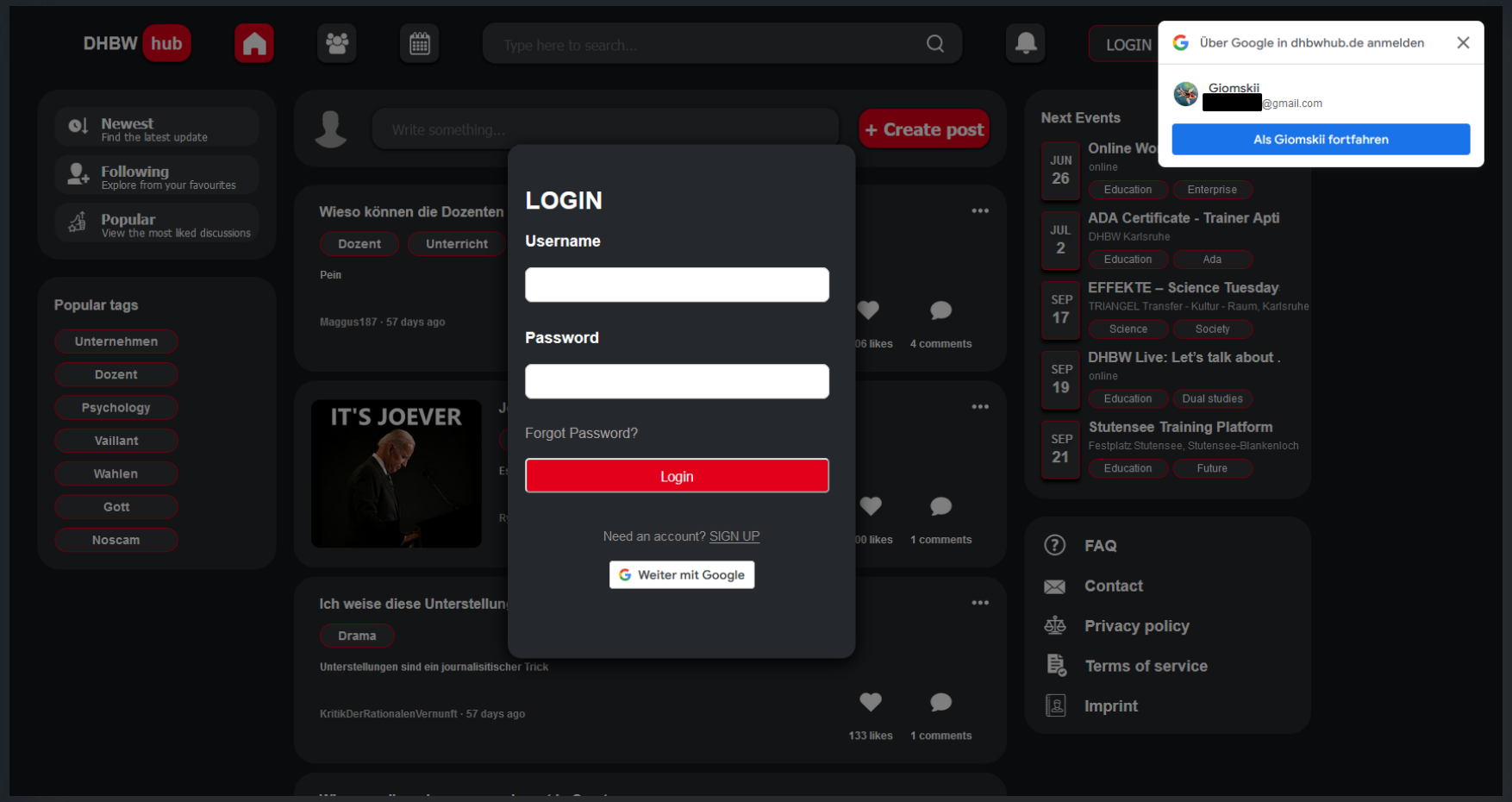
Create posts and comments

Profile personalization

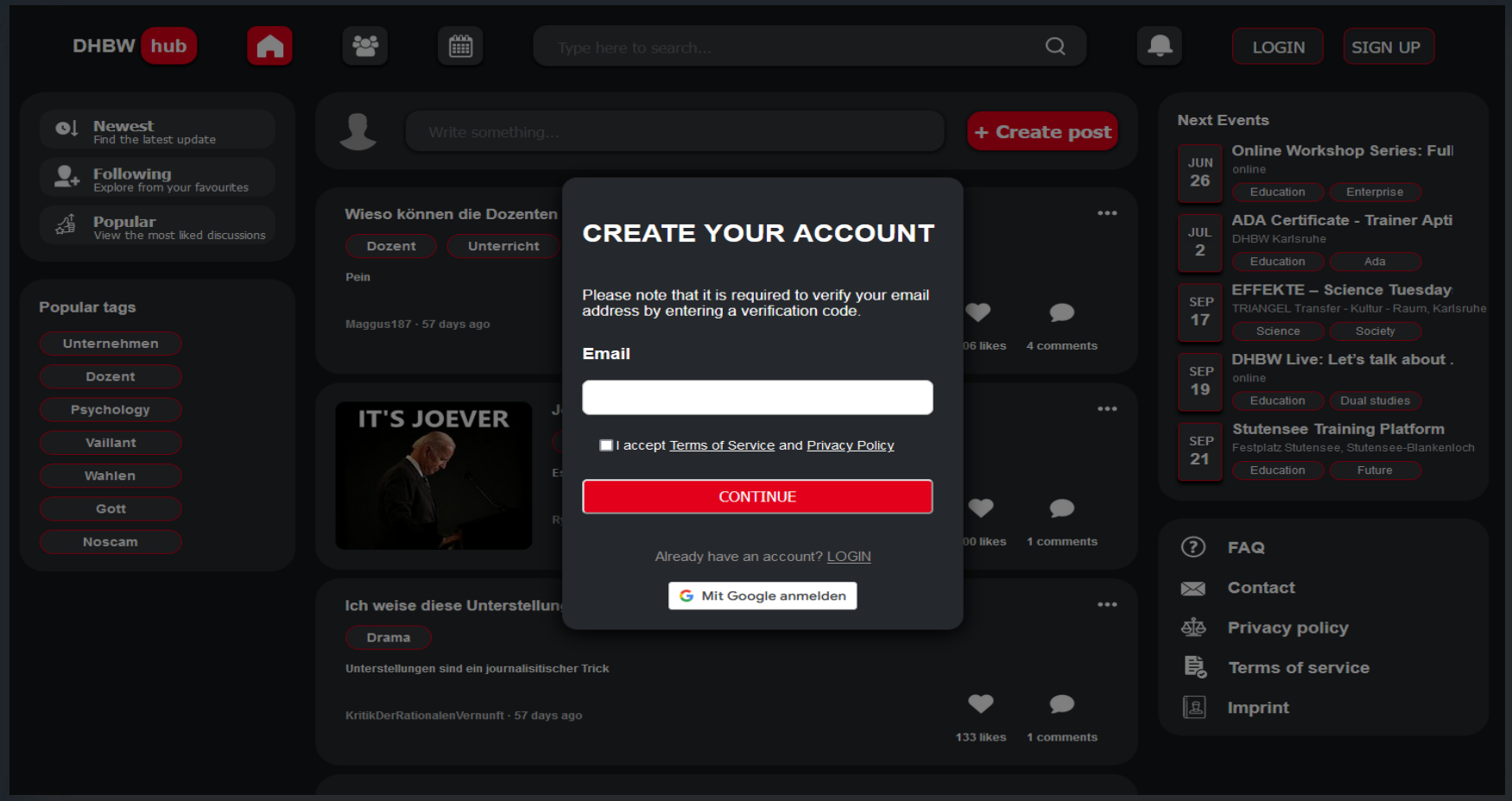
Live demonstration - Homepage



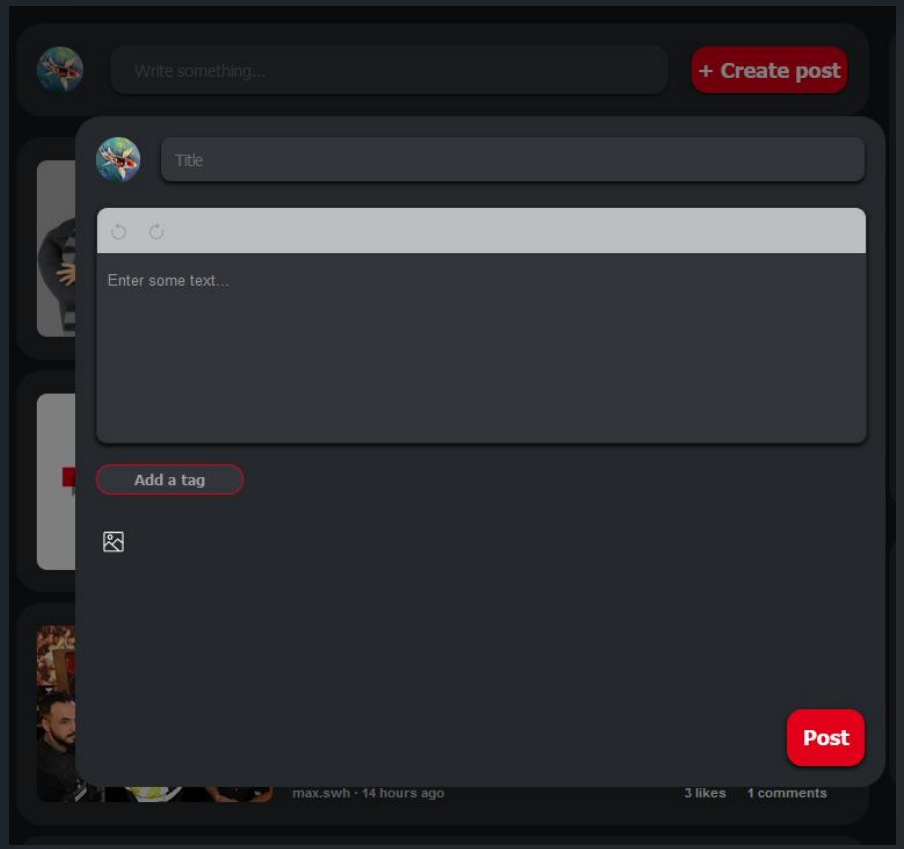
Live demonstration – Sign in

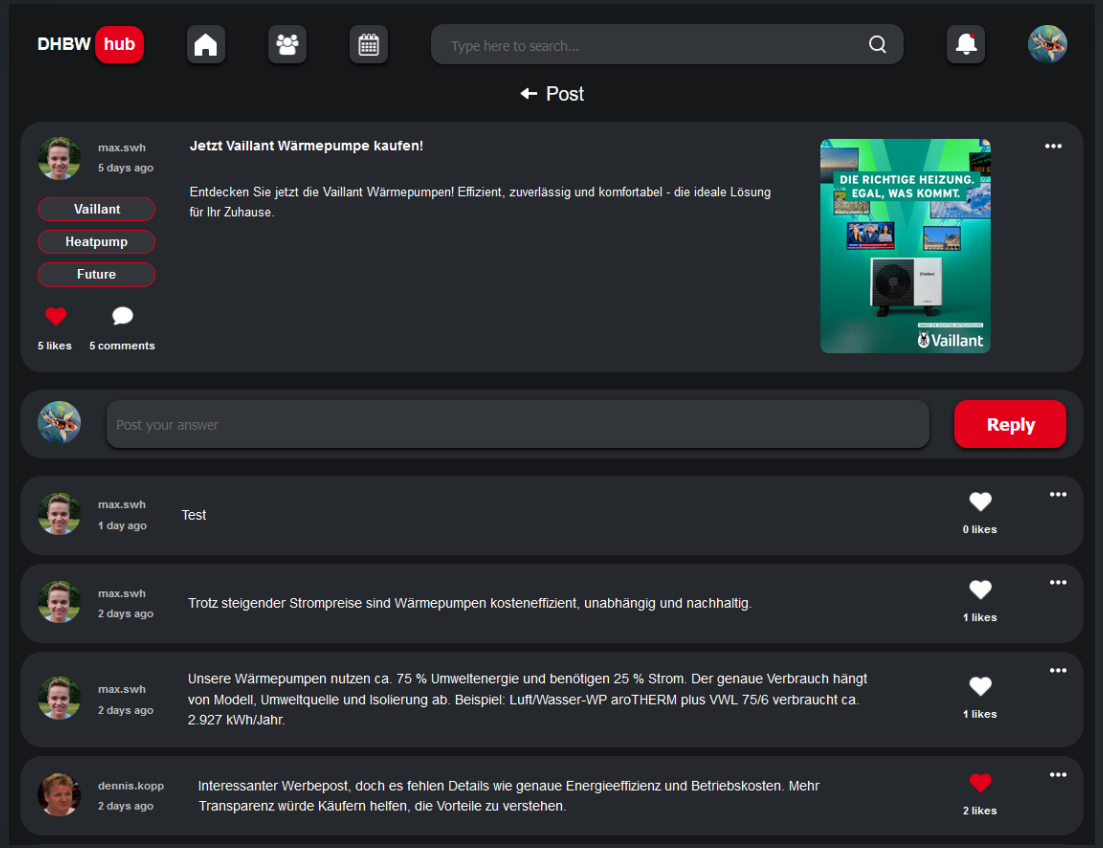


Live demonstration – Sign up

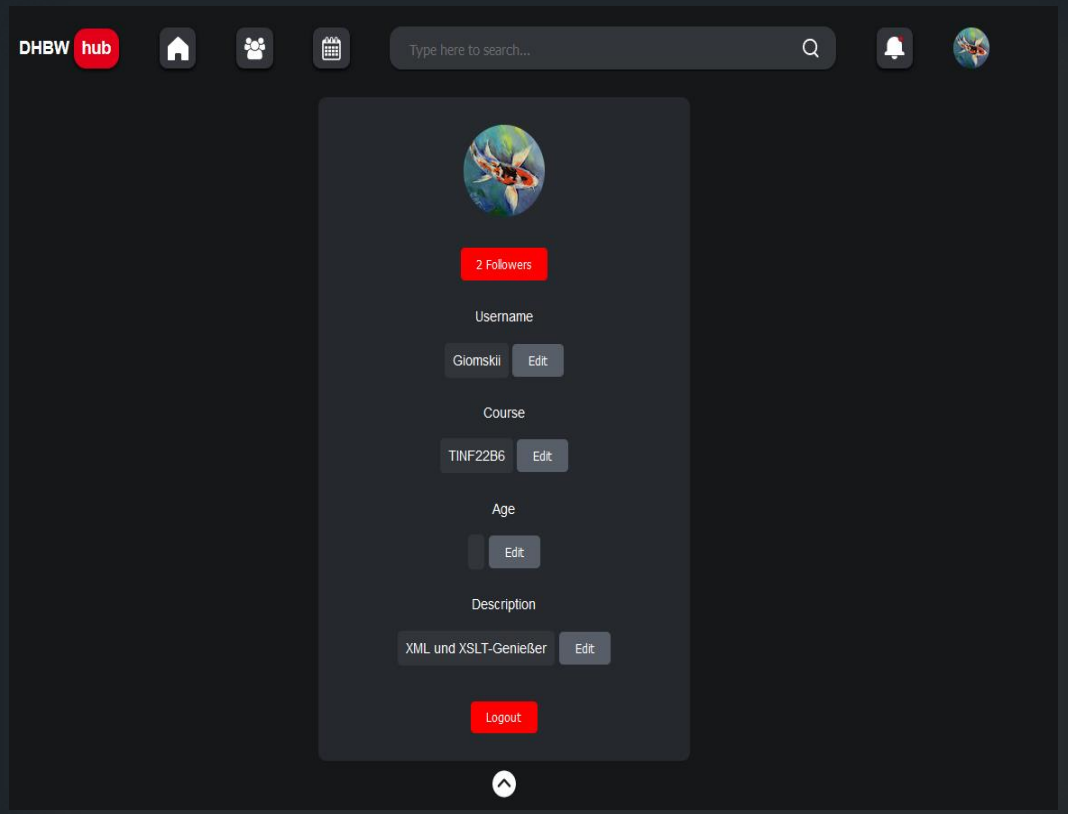


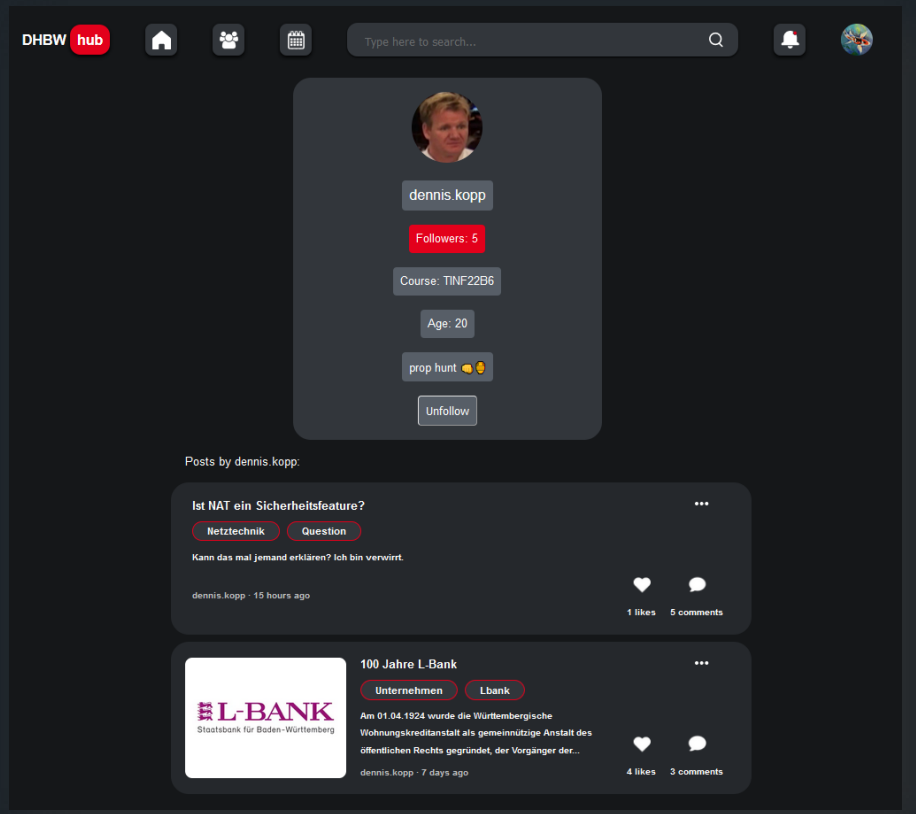
Live demonstration – Posts + Comments



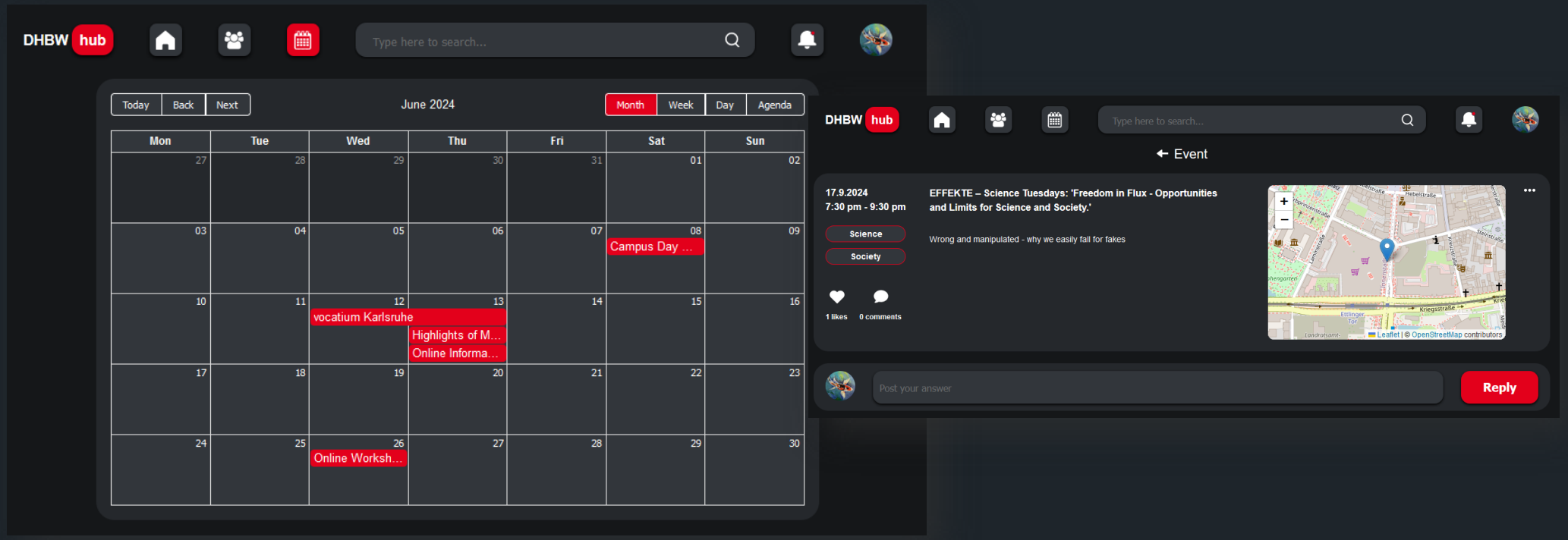


Live demonstration – Profile + User

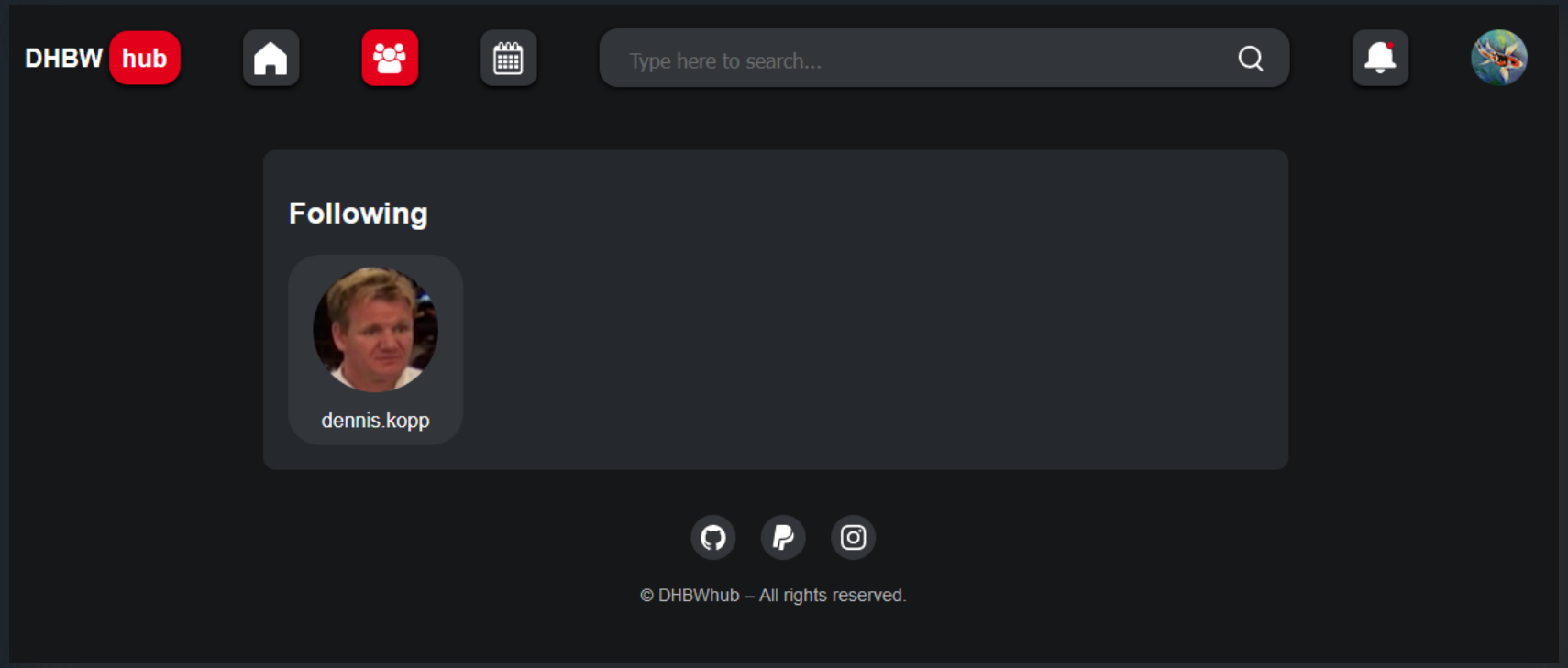




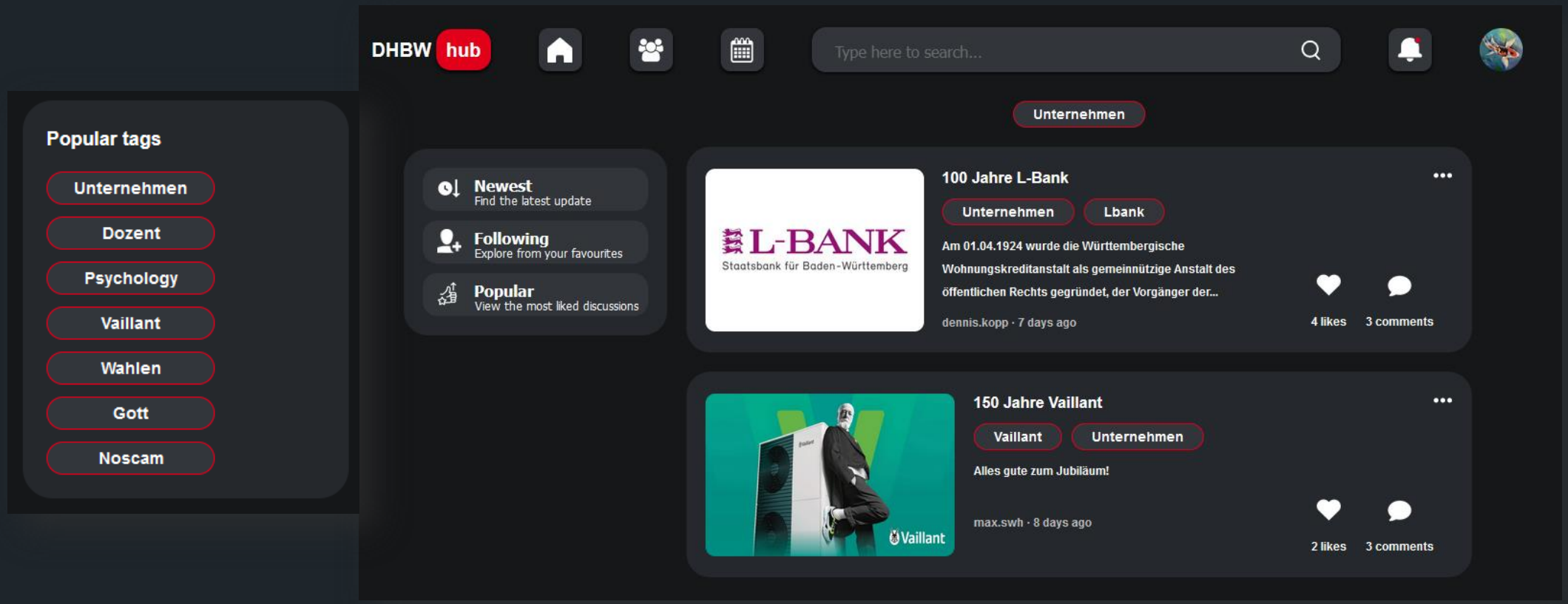
Live demonstration – Calendar + Events



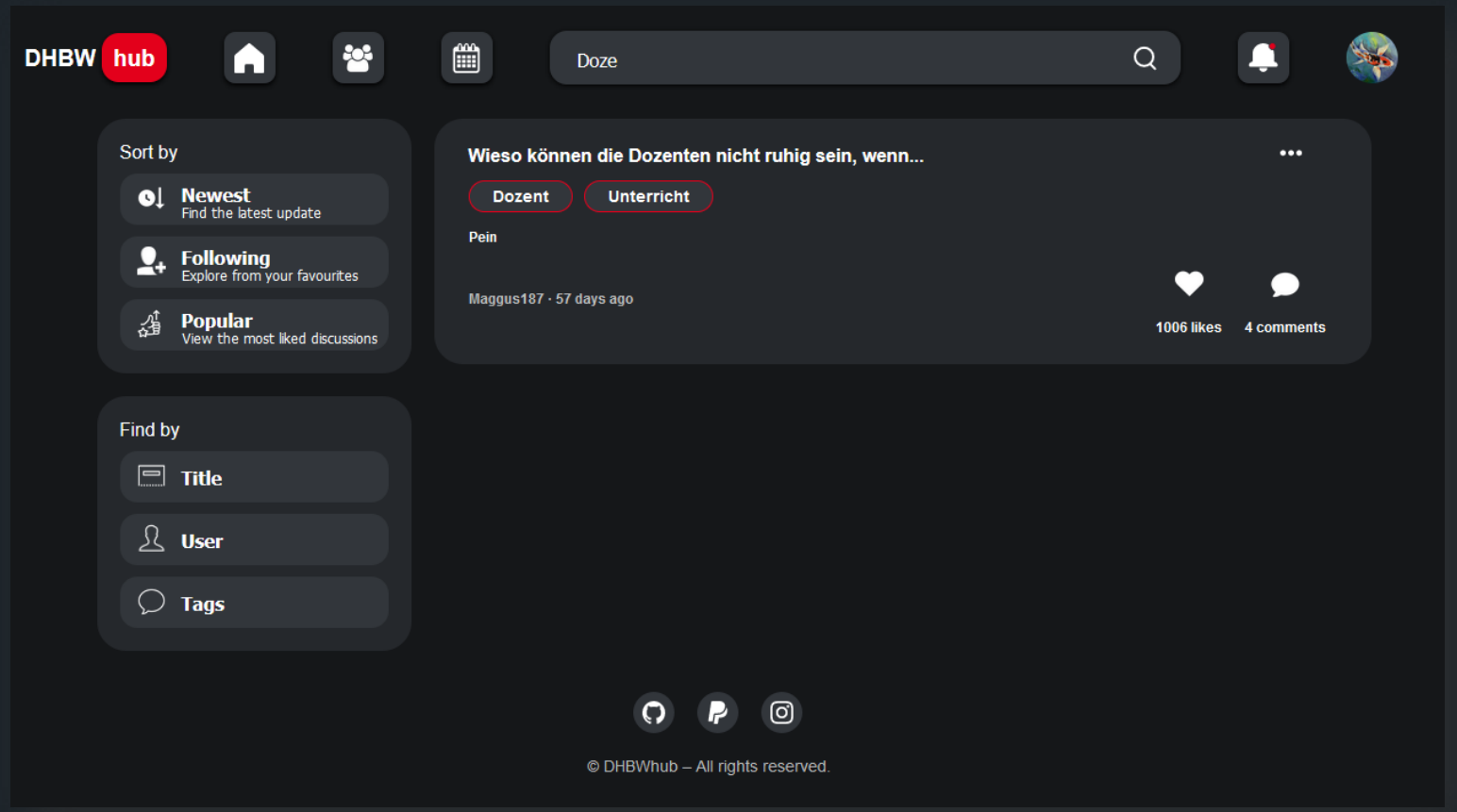
Live demonstration – Following page



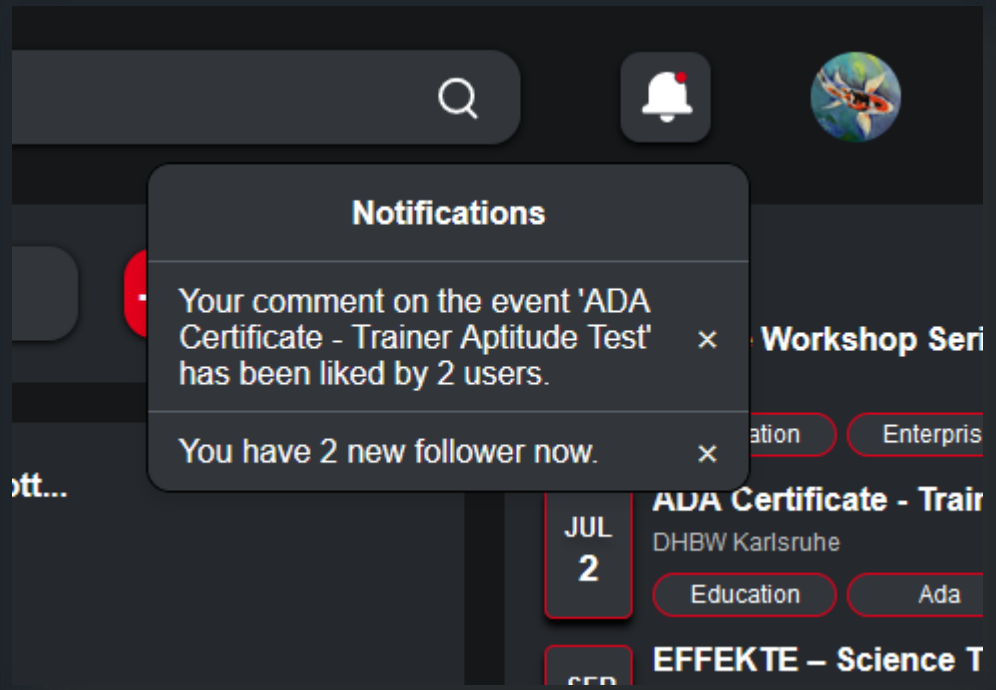
Live demonstration – Popular Tags



Live demonstration – Searchbar



Live demonstration – Notifications



Project management – Lessons learned

- CI/CD is the main branch‘s best friend.
- Diagrams help break down complex structures.
- Communication is underrated.

Any questions?

SCAN ME

Please support us!