Social media web application.

---- Frontend -----

**HTML** 5,

**CSS** (and **CSS** precompiler **SASS**). **CSS** would be divided into **modules**, for each different component.

**ES** 13

**React**. **React** framework will be the core of frontend part. The application would be made in a form of single-page app. **React router** will be used for managing switching pages.

The whole front-end part would be bundled using **Vite**.

To ensure that code is clean, that there are no variables that are declared but not used, **ESLint** will be used.

----- Backend -----

Backend will be built using **Java** and **Spring** framework as its core. Essentially, **SpringBoot** will be used, as it is more suitable for small and medium-sized applications where there is no need for extensive configuration and set up. Also, **Springboot** offers several tools needed specifically for web development.

The security will be made using **Spring Security** framework, the data will be transferred as **JWT’s**.

The whole backend part would be built using **Apache Maven**.

----- Data -----

The database will be a **relational** one. It will have the following tables: Users, Posts and Comments. Between those tables, relations will be organized. Each entity in Posts and Comments will have a foreign key corresponding to (author) user primary key.

----- Organization -----

The **Git** version control system would be used during the development. The whole project would be submitted to **Github** as a single repository. The development process would be split up into tasks and features, that would be managed using **Jira.** Each feature would have its own branch on git, which, after the development is finished, would be merged into “develop” branch. The “develop” branch then would be merged into the main branch.

The backend and frontend parts of the project would be bundled together and started up using **Docker** software. The application would be deployed using **AWS** systems.

The software will be tested during the development using **JUnit** for backend and **React Testing Library** for frontend.

----- Other -----

Docker

* Consistence of the running environment
* Persistence of data

----- Structure -----