**Kanban Board Project Work**

Lappeenrannan–Lahden teknillinen yliopisto LUT

Software Engineering

2025

Aleksi Natunen

# Technology choices

- Package manager
  - NPM
    - I used NPM as the package manager. I used it because it was already known to me as we had used it during the course. Also because it has many packages that are up to date
- Frontend
  - On the frontend I used React.
    - I used react for the frontend. I used it for the purpose that it was easy to use, it has components and other built in functions that makes coding easier.
  - MUI
    - I used MUI as the user interface package. I used it as it gave a better look for the frontend in addition to the fact that it made it more accessible.
- Backend
  - Express
    - I used Express as my web application framework. I used it because it was used during the course making it easier for me to use
  - Database: MongoDB
    - I used MongoDB as my database. I used it because it is easy to use. In addition to this the fact that we used it during the course, making me familiar with it already
- Middleware
  - Mulder
    - I used mulder to get files from frontend to backend. This I used because it was used during the course already making it easy to use. Also it was easy to use.
  - JWT/JWTPayload
    - I used these for the purpose of validating users and making tokens. These I used as they were used on the course already and I was used to them.
  - Bcrypt
    - I used Bcrypt to hash my passwords. I used it purely for the reason it is easy and has been used on the course.
- Testing
  - Cypress
    - I used cypress as the testing framework. I used it as it is a known framework for web application testing and fairly easy to use/learn.

# Installation

1. Install NPM https://docs.npmjs.com/cli/v8/commands/npm-install
2. Install MongoDB https://www.mongodb.com/docs/manual/installation/
3. Install your wanted IDE (I suggest Visual Studio Code)
4. Download/pull project directory to wanted place
5. Open it up on your IDE
6. Open terminal
7. Change directory to /(name of the project directory) in this case /prokkis

   ```
   cd .\prokkis\
   ```
8. Run commands npm install, npm run install:client and npm run install:server

   ```
   npm install
   ```

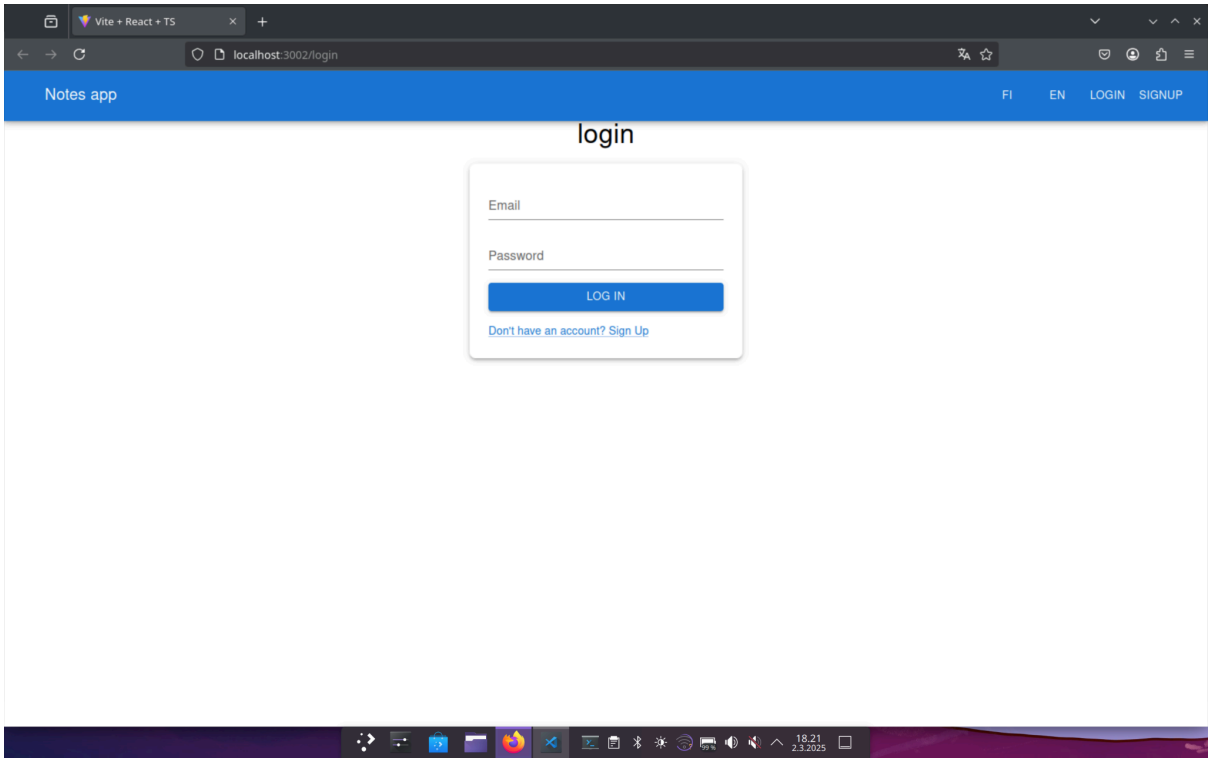
   ```
   npm run install:client
   ```

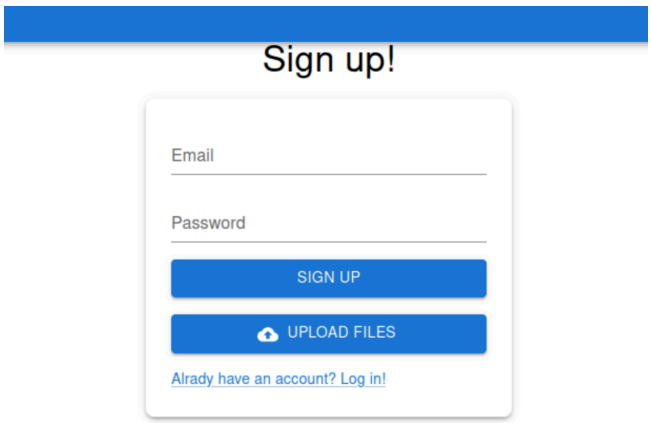
   ```
   npm run install:server
   ```
9. It is now installed

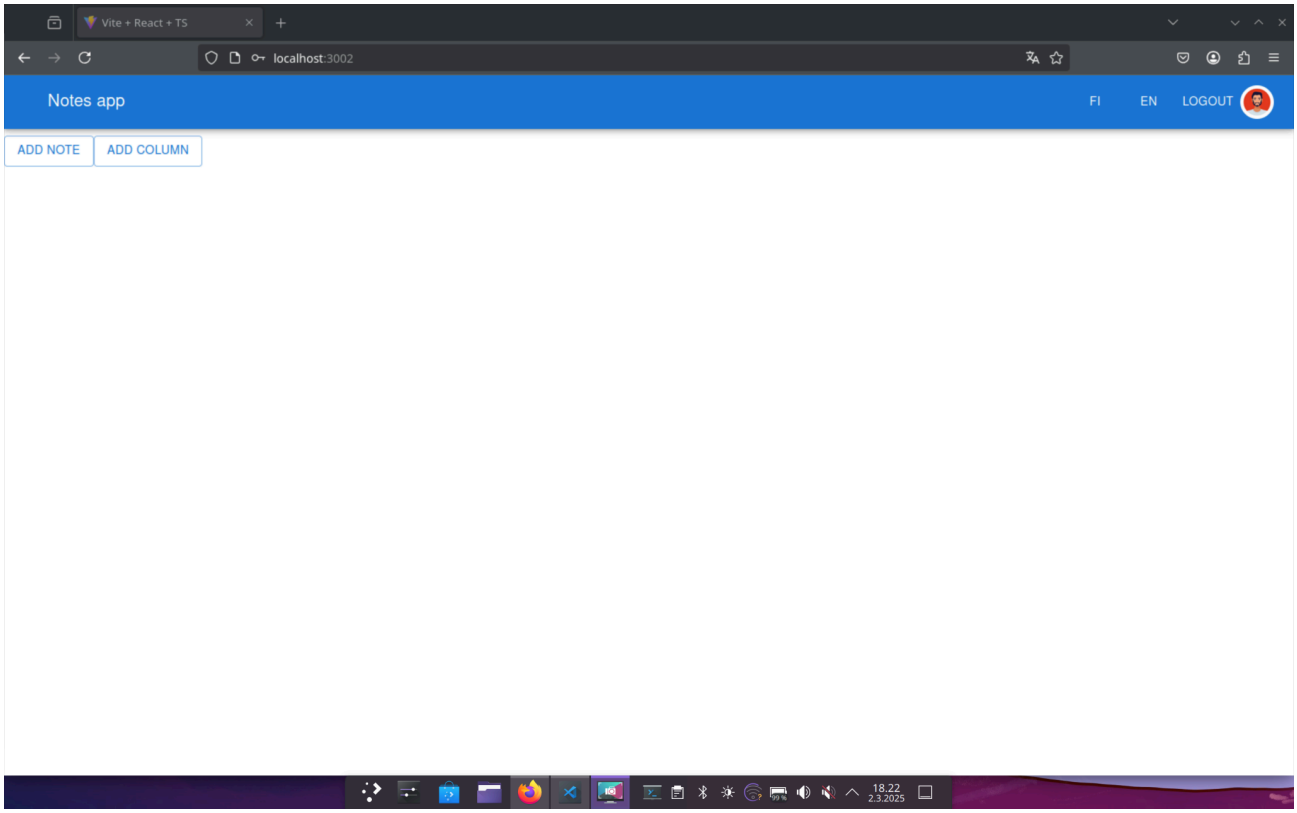
# User manual

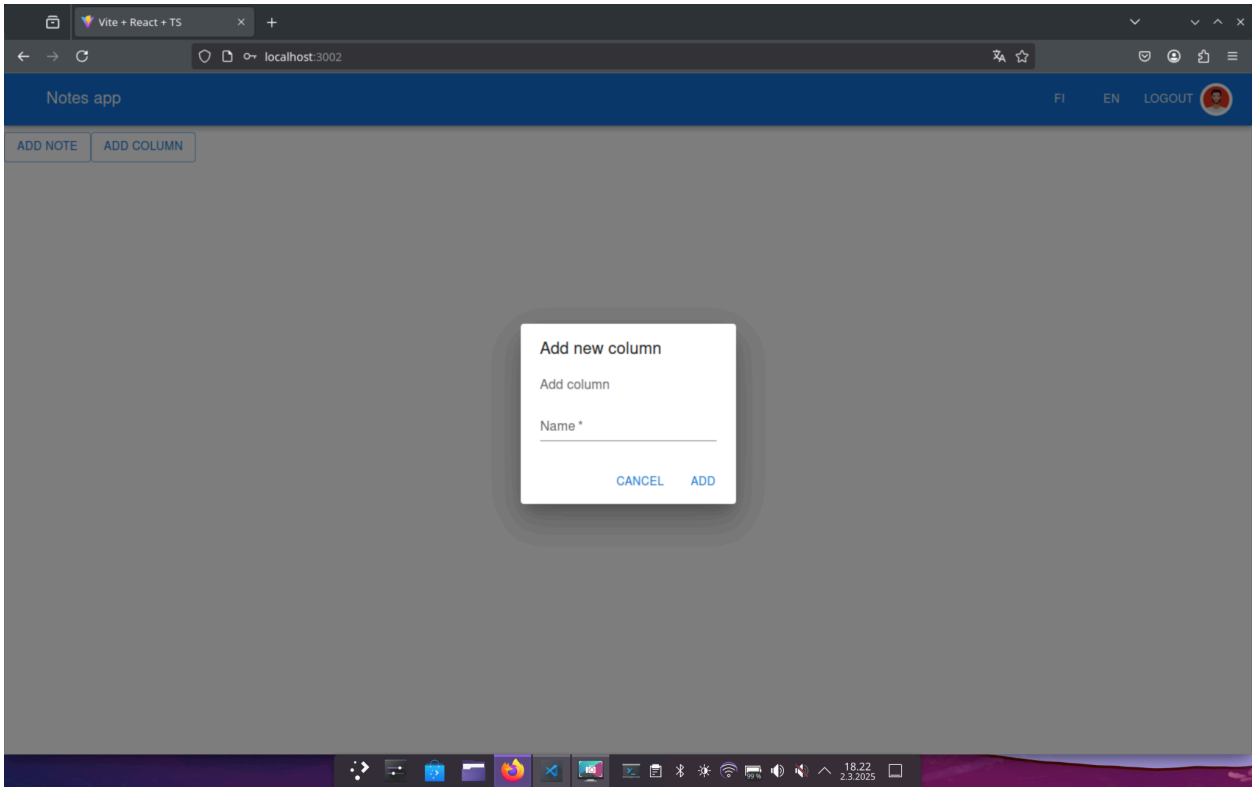
1. Run MongoDB on your device
2. Then run command npm run dev on the IDE terminal in the ./prokkis/ directory
   ```
   npm run dev
   ```
3. Now it should be running (NOTE: Frontend is running on port 3002 and backend on 3001. Also MongoDB is running on `mongodb://127.0.0.1:27017/ProkkisDB`)
4. After this the user can see the frontpage of the project
5. There users can login or signup (when signing up remember to give yourself a cool photo!)
6. After logging in user can see the home page
7. There user can see add column and add note and logout
8. User can add column with add column
9. Then user can add note with add note
10. Then user can see icon buttons on the column and note
11. With gear icon user can change the name of the column
12. With trash icon user can delete column or note (when removing column removes all notes under it)
13. Notes can be moved up and down with buttons or side to side with drag and drop
14. All of made notes and columns will be saved onto database and only your account can see them

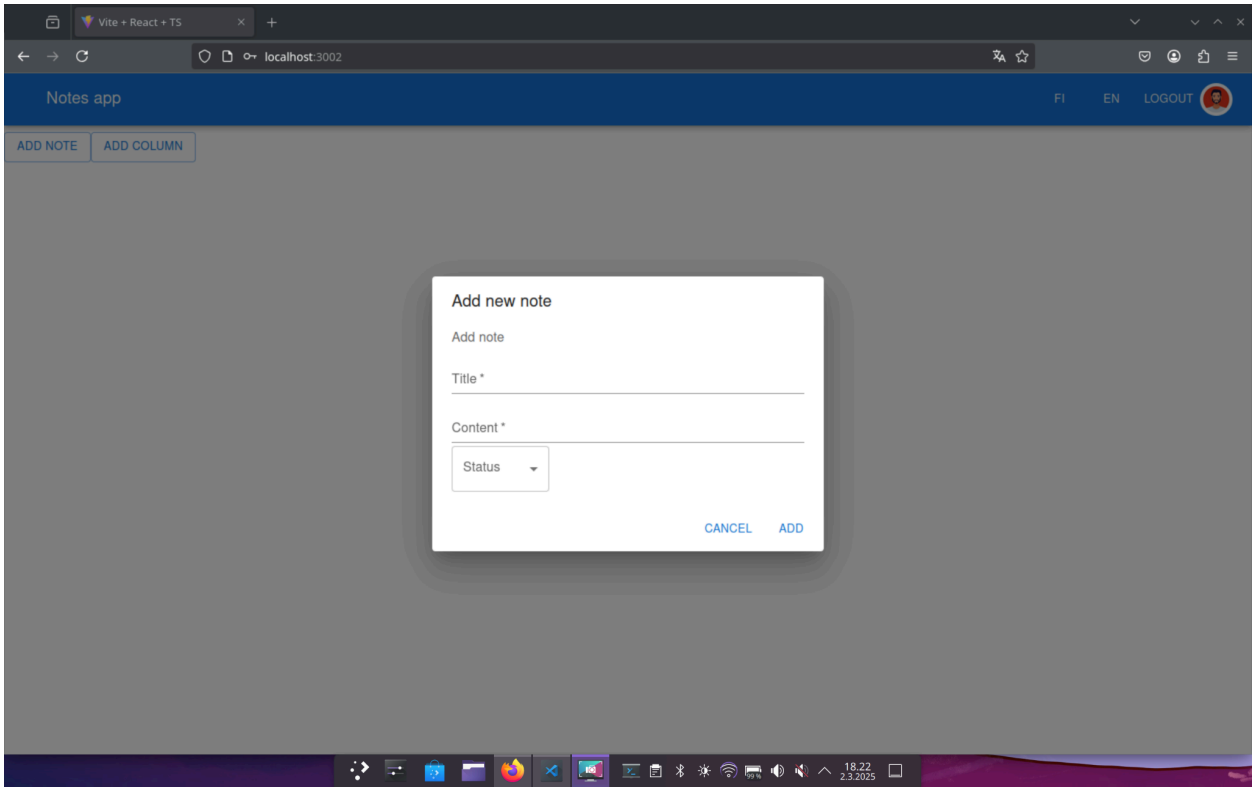
NOTE: User can change language with FI and EN buttons


Vite + React + TS
localhost:3002/login
Notes app
FI
EN
LOGIN
SIGNUP
login
Email
Password
LOG IN
Don't have an account? Sign Up


Sign up!
Email
Password
SIGN UP
UPLOAD FILES
Alrady have an account? Log in!


Vite + React + TS
localhost:3002
Notes app
FI
EN
LOGOUT
ADD NOTE
ADD COLUMN


Vite + React + TS
localhost:3002
Notes app
FI
EN
LOGOUT
ADD NOTE
ADD COLUMN
Add new column
Add column
Name *
CANCEL
ADD


Vite + React + TS
localhost:3002
Notes app
FI
EN
LOGOUT
ADD NOTE
ADD COLUMN
Add new note
Add note
Title *
Content *
Status
CANCEL
ADD
18.22
2.3.2025

# Testing

To run Cypress tests

1. Change dir to ./client
2. Run “npx cypress open” in your terminal
3. Do all the necessary steps
4. Open E2E testing
5. Go to specs
6. And there will be tests then run them

NOTE: SERVER AND DATABASE MUST BE RUNNING BEFORE RUNNING THE CYPRESS APP

If having problems follow given steps here:
https://docs.cypress.io/app/get-started/open-the-app

# Implementations/points

| **Feature** | **Max points** |
|---|---|
| Basic features (as stated in the previous chapter) with well written documentation | 25 |
| Utilization of a frontside framework, such as React, but you can also use Angular, Vue or some other | 3 |
| User profiles can have images which are shown on the main page and in the chat | 3 |
| Cards can have comments in them, one or many | 3 |
| Cards and comments have visible timestamps when they have been created and updated | 4 |
| Translation of the whole UI in two or more languages | 2 |
| Multiple users can work with the same board and the block (card/column/comment) is blocked from another user so that they are not messing with each other's work | 5 |
| Create (unit) tests and automate some testing for example with https://www.cypress.io/ (at least 10 cases have to be implemented) | 5 |
| ***POINTS I DESERVE*** | **50** |

# AI Declaration

1) ChatGPT, Microsoft Copilot

2) For couple bug fixes and proofreading/text correction