

Advanced Web Development Project

Antti Keronen

001855050

Declaration of Ai use:

Ai was used to solve problems in the code by posting the error message to ChatGPT and watching where's the fault and how could you solve it.

1. Introduction

The program is a cloud drive web application. It allows users to register, login, and manage their own documents. Users can create, edit, rename, delete and share documents with other users.

2. Technology Choices

Backend

Backend includes Node.js with Express, typescript, mongodb, JWT and Multer. The absolute main reason for why I chose them is because they are familiar to me, but if more reasons are needed then Node.js and Express are very lightweight, typescript is maintainable although I would have chosen to use only Js, but that would have given me minus points. For databases I chose mongo since I don't really know any other approaches. For authentication I think JWT is the easiest one as well as multer for pictures.

Frontend

Frontend includes React, typescript, react router, Axios and Bootstrap. These are also the most familiar to me, but React gives me component based UI, typescript gives maintainability and little bit clarity to the code, Router helps navigating, Axios simplifies the programs API communication, Bootstrap makes it cooler.

3. Installation Guidelines

Requirements for the program:

Node.js v18 →, MongoDB, npm

Backend setup

```
cd backend  
npm install  
  
create . env file  
  
PORT=5000  
MONGO_URI=mongodb://127.0.0.1:27017/cloud_drive  
JWT_SECRET=yoursecret  
  
npm run dev
```

Frontend Setup

```
cd frontend  
npm install  
npm run dev
```

Program will be running at:

Frontend <http://localhost:5173>
Backend <http://localhost:5000>

4. User Manual

Authentication

Users can register with email and password. Registered users can login. Only authenticated users can access the drive and documents. Users can log out.

Drive

Users can create text documents and spreadsheets. Documents can be renamed, edited and deleted. Documents display creation and last updated timestamps. Users see all documents they own or have edit access to.

Document Editing

Text documents can be edited using simple editor. Comments can be added to overlined text. Only one user can edit a document at a time. If another user is editing, a message is shown. If the editor closes the tab, the lock will expire automatically.

Sharing

Document owners can grant edit access to existing users by email. Owners can revoke editor access. Documents can be shared publicly via link. Public links allow viewing without authentication. By opening a document and opening public view, users can download the file as pdf.

Spreadsheets

Users can create simple spreadsheets. Cells support raw values and sum formulas.

Profile

Users have a profile page. Users can upload a profile picture.

5. Points I'm Begging for

| Feature | Point Amount |
|-------------------------------|--------------|
| Basic features | 25 |
| React frontend | 3 |
| Includes EditorJs | 2 |
| Document timestamps | 1 |
| Profile Picture | 2 |
| Commenting | 5 |
| Spreadsheet with sum | 6 |
| Sorting the documents | 1 |
| Search Functionality | 2 |
| Pagination | 2 |
| Pdf | 3 |
| Returned before 22.2.2026 | 1 |
| Bonus point (It looks nice 😊) | 1 |
| | =54 |