Online Collaborative Text Editor

Table of Contents

Introduction

We have designed and implemented an online real-time collaborative text editor. This type of software enables multiple users on different machines to edit the same document simultaneously, similar to popular tools like Google Docs.

Features

User Management

- **User Registration** and **Authentication**: Register and login to user accounts.

Document Management

- **File Management**: Create, open, rename, and delete files.
- **Access Control**: Share documents with permissions (viewer or editor); ensure security with owner-only file deletion.
- **List Documents**: View owned and shared documents.

Real-time Collaborative Editing

- **Support File Editing**: Edit document text with bold and italic formatting.
- **Support Concurrent Edits**: Enable multiple users to edit simultaneously; manage conflicts effectively.
- **Real-time Updates**: Track edits in real-time; see other users' cursors and active sessions.

UI

- **Simple UI**: Includes login, sign up, intuitive file management (create, list, delete, rename, share, open), and text editing capabilities.

Tech Stack

- **Backend:**
  - Java
  - Spring Boot
  - Spring Security
  - STOMP Web Sockets
  - SQL Database
- **Frontend:**
  - React.js
  - Quilljs

How to Run

1. Redirect to the backend folder:

   ```
   cd Online-Collaborative-Text-Editor/backend
   ```

2. Build the `gradle.build` file:

   ```
   ./gradlew build
   ```

3. Run the Spring Boot app on localhost:

   ```
   ./gradlew bootRun
   ```

4. Redirect to the frontend folder:

```
cd ../frontend
```

5. Install the dependencies:

```
npm i
```

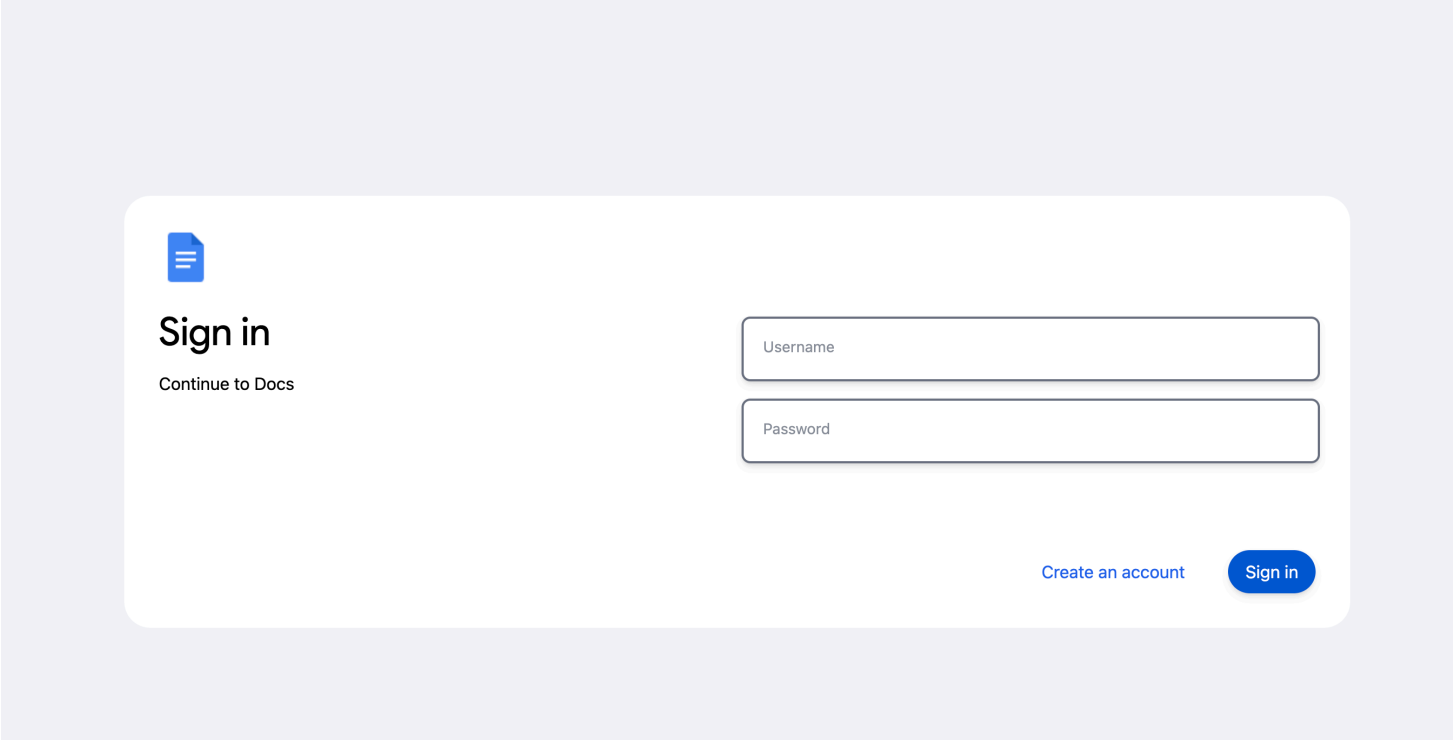
6. Run the development server:

```
npm run dev
```

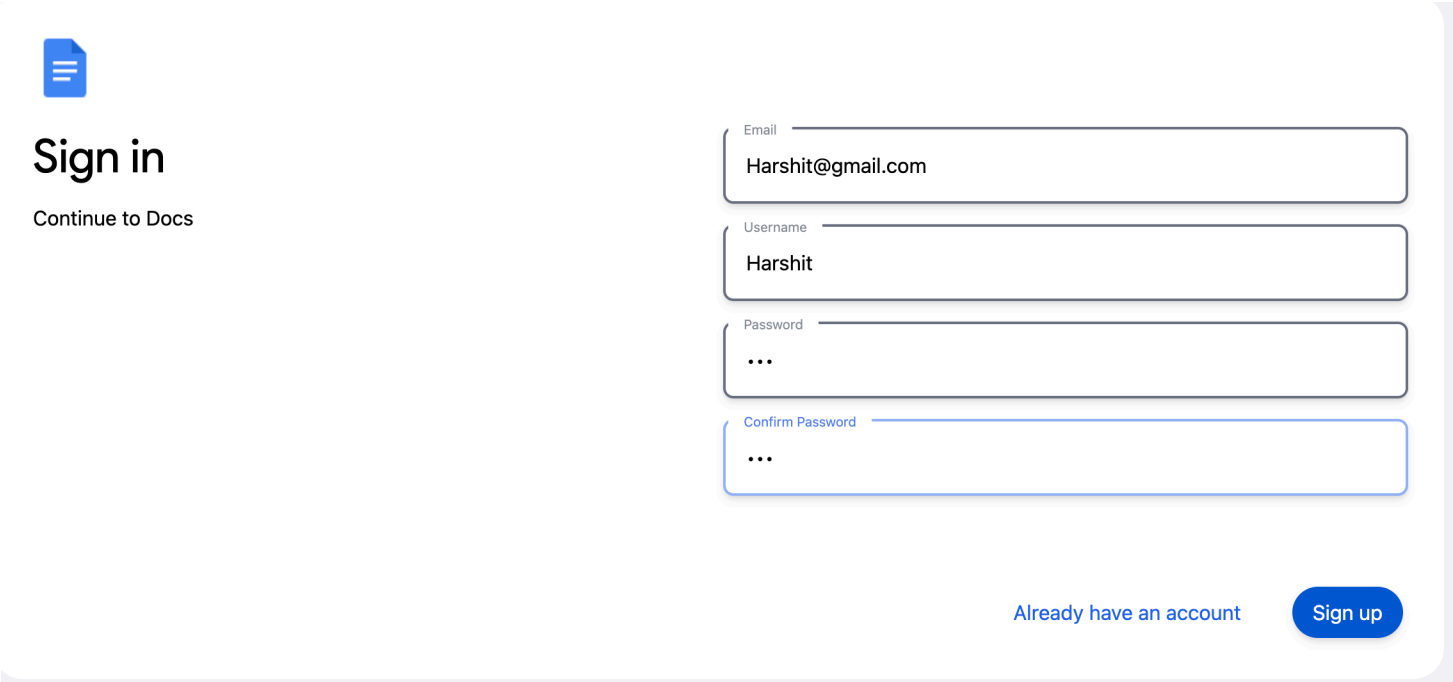
7. Open a web browser and go to the host link.

Screenshots

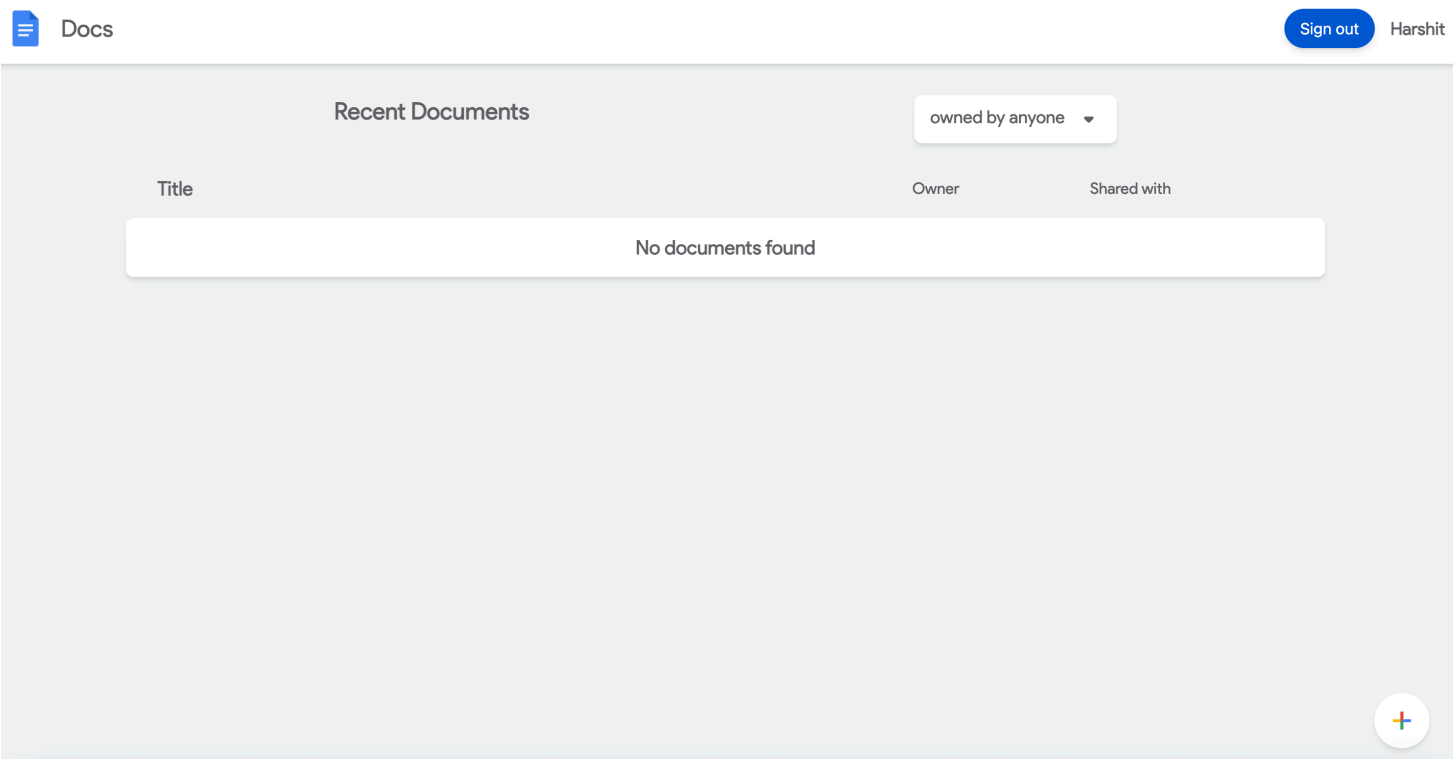
When you sign in



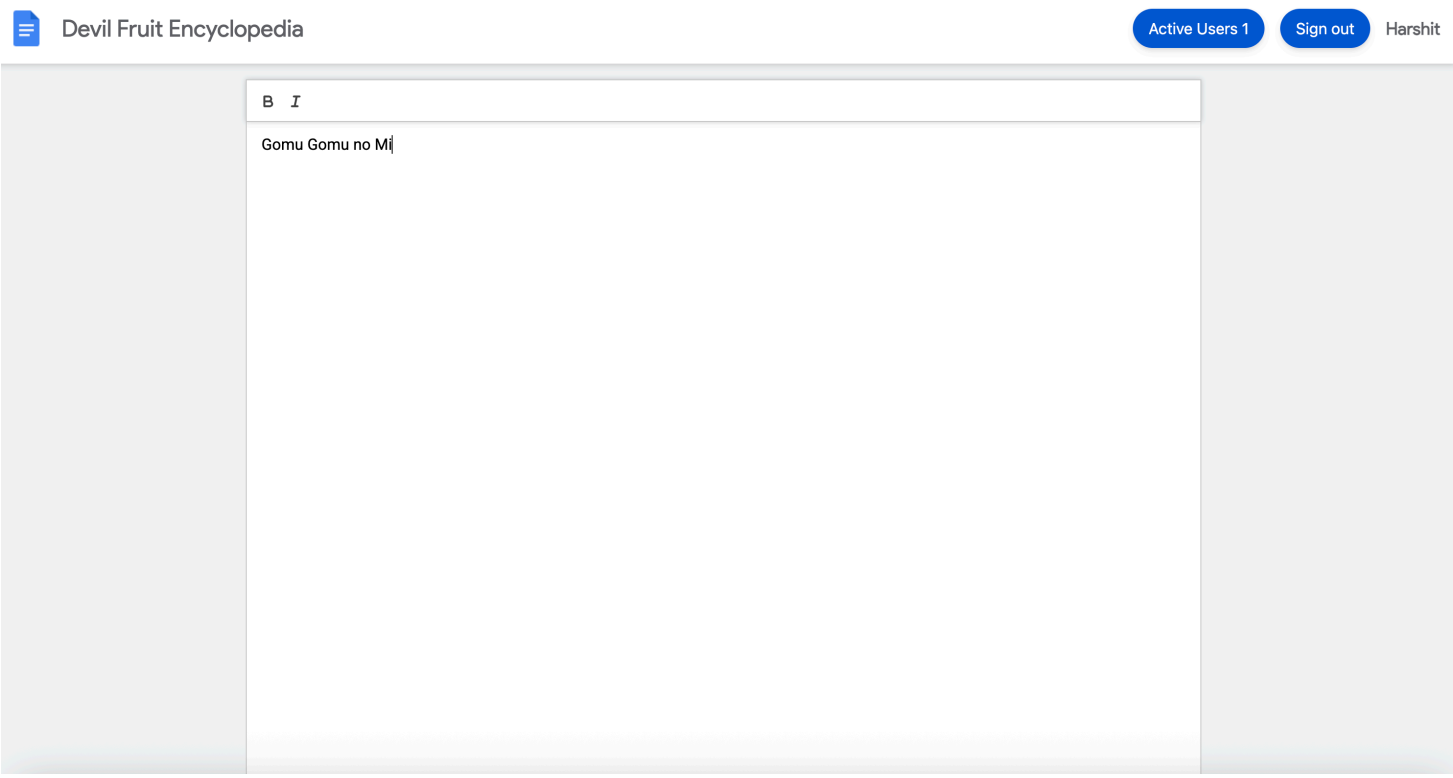
Sign up window



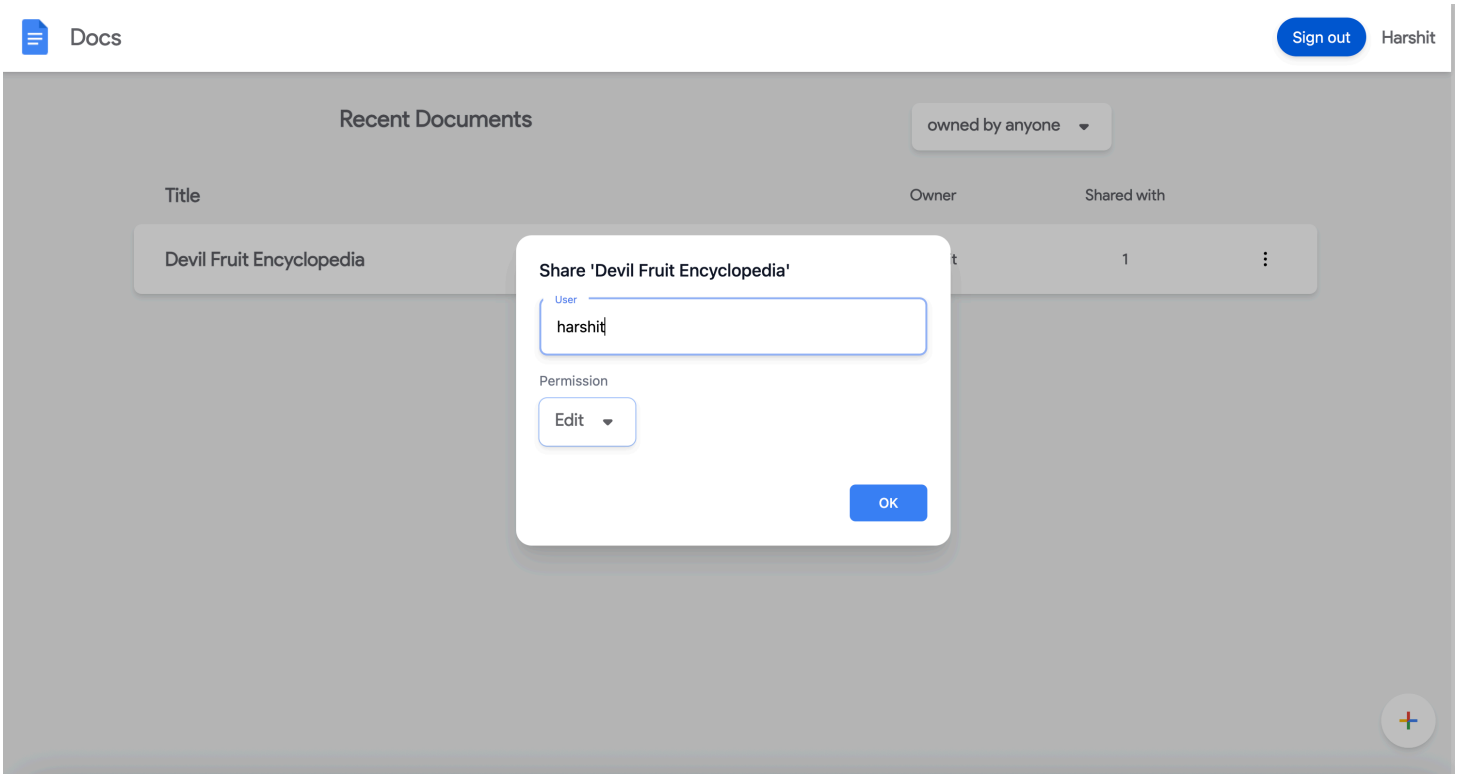
Homepage

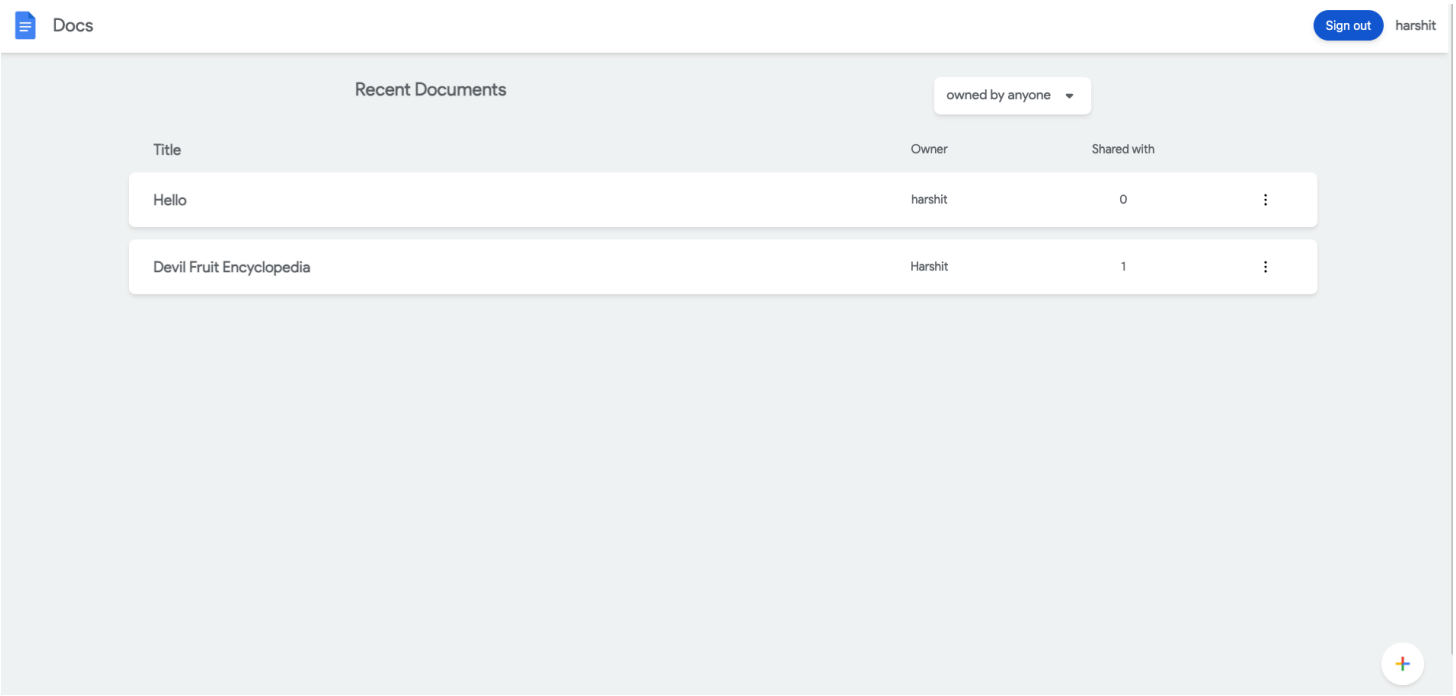


Doc editor

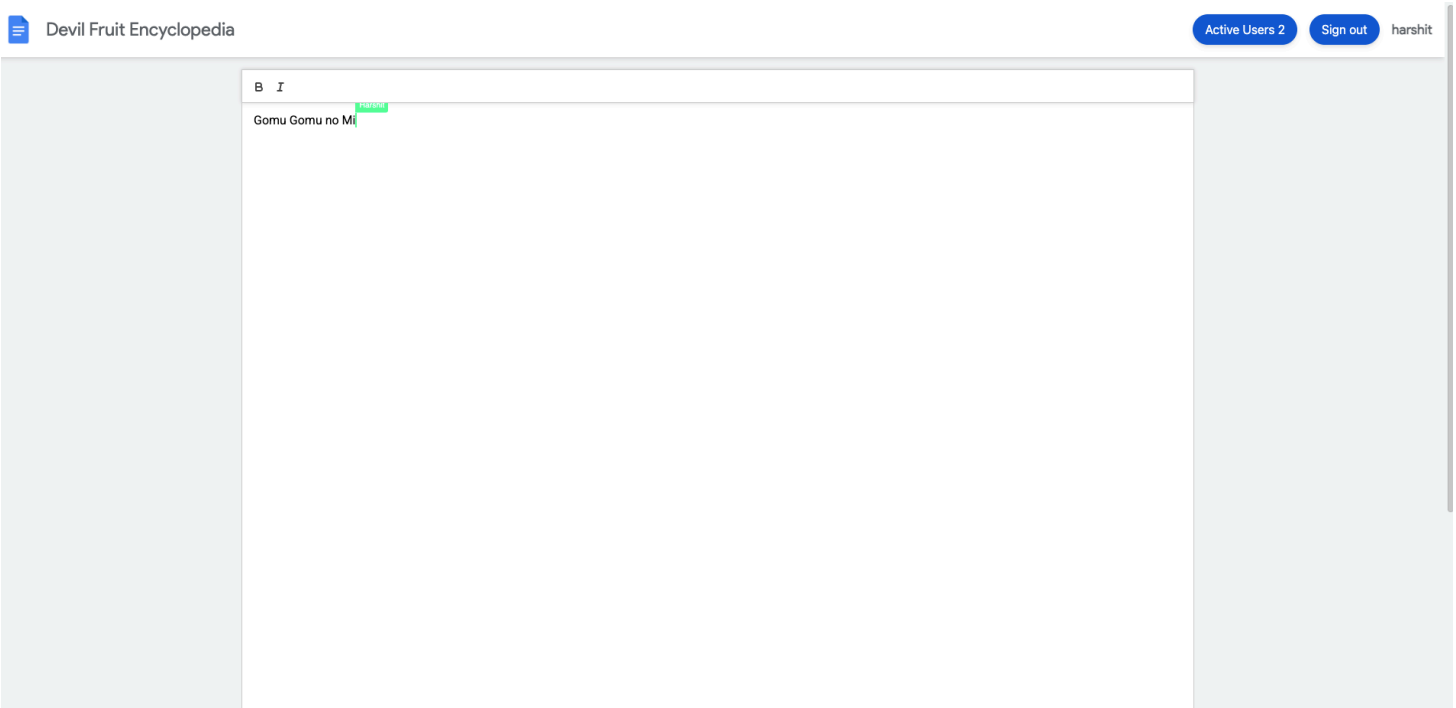


Share the doc

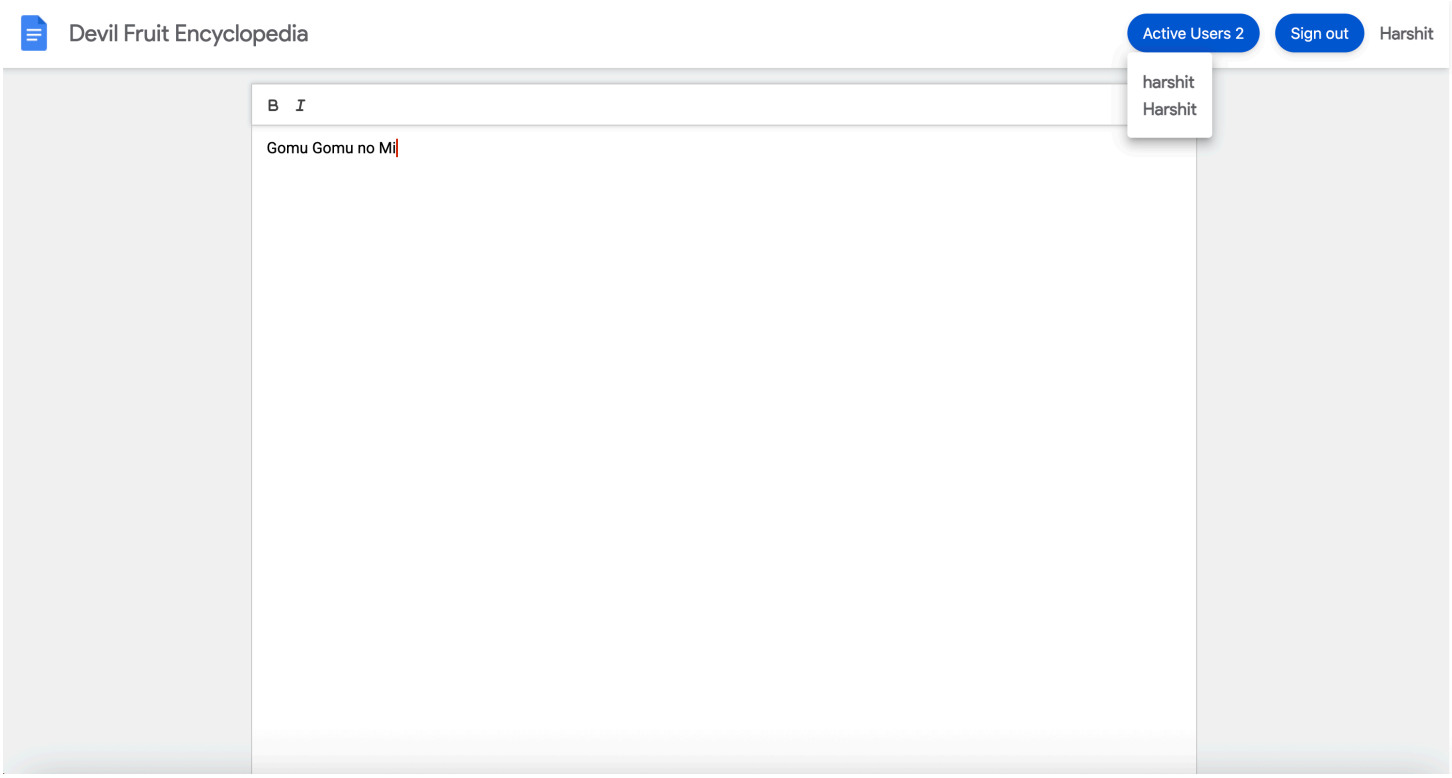




Edit Collaboratively



Real time editing with shared users



Real-Time Conflict Resolution (CRDT Algorithm)

To handle real-time collaboration, we use a CRDT (Conflict-free Replicated Data Type) approach, similar to a doubly linked list.

Each character (node) stores:

- A unique ID (e.g. `h@user`)
- Left and right node IDs (for ordering)
- The character itself
- Bold/Italic flags
- A deleted flag

How It Works

- **Insert**: New characters are placed between left and right node IDs. If multiple users insert at the same place, the username decides the order (alphabetically).
- **Delete**: Characters are marked as deleted (not removed).
- **Bold/Italic**: Formatting flags are stored on each character node.
- **Conflict Resolution**: CRDT ensures edits from multiple users merge cleanly, without overwriting.

All nodes are stored in a map for fast lookup by ID, allowing real-time updates and consistent state for all users.

How to run

- cd Online-Collaborative-Text-Editor/frontend
- npm install # only needed the first time
- npm run dev
- cd Online-Collaborative-Text-Editor/backend
- ./gradlew clean build
- ./gradlew bootRun