Recurring / autofill feature

Topic suggestions (auto - templates)

Create animations (dragging animation, etc.)

Links and images feature

Journal Exports

Custom bullet points

miro

User access to their journal

cookies? (Local storage)

Pros
- No need for a server
- Accessible locally

Cons
-Not a good permanent solution
-Limited data that can be stored

Having a server

Pros
Is cool, can do Logins/Password, etc.

Cons
- Having a server xD

miro

Easy accessibility to where user wants to go

- Hyperlinks?
- Bookmarks?

Index Requirements

info

- Index
- Future Log (year)
- Monthly Planner
- Weekly Log
- Daily Log

Requirements:

miro

Accessible without the internet?

Accessibility

Download the journal?

problems?

Convenience

Keep track of daily class schedule

Keep track of upcoming exams

Put in links, images, files, etc.

Typeset (other languages, jargons, ability to draw, math equations)

Keeping track of busy school schedule

A separate Custom log

needs?

Target Customer: University Students

Could be used to vent out your feelings (mental health)

Topic Suggestions (meals, homework, etc.)

Different date formats (mm/dd/yy vs dd/mm/yyyy ,etc.)

reminder

Taskbar?

Basic functionality for things not school related. Planner? Calender?

Autofill feature, recurring things

Dark mode

Preferred themes?

allow user to have multiple choices

minimalist design?

miro

```mermaid
flowchart TD
    A[Steps: Front-end and back-end] --> B[Display tasks, events, notes]
    A --> C[Create login/home page]
    B --> D[Design UI to create tasks, events, notes]
    C --> E[Create back-end mechanism for login (with Google account?)]
    D --> F[Code and implement that UI]
    F --> G[Design navigation/log UI]
    F --> H[Create JSON format for entries]
    E --> H
    G --> I[Implement navigation/log UI]
    H --> J[Client creates JSON from user entry]
    J --> K[JSON uploaded to server]
    K --> L[Server stores JSON with user account]
    L --> M[Server sends JSON to client]
    M --> N[Client parses JSON and displays it]
```

Steps: Front-end and back-end

- Display tasks, events, notes
  - Design UI to create tasks, events, notes
    - Code and implement that UI
      - Design navigation/log UI
        - Implement navigation/log UI
- Create login/home page
  - Create back-end mechanism for login (with Google account?)

Create JSON format for entries

Client creates JSON from user entry

JSON uploaded to server

Server stores JSON with user account

Server sends JSON to client

Client parses JSON and displays it