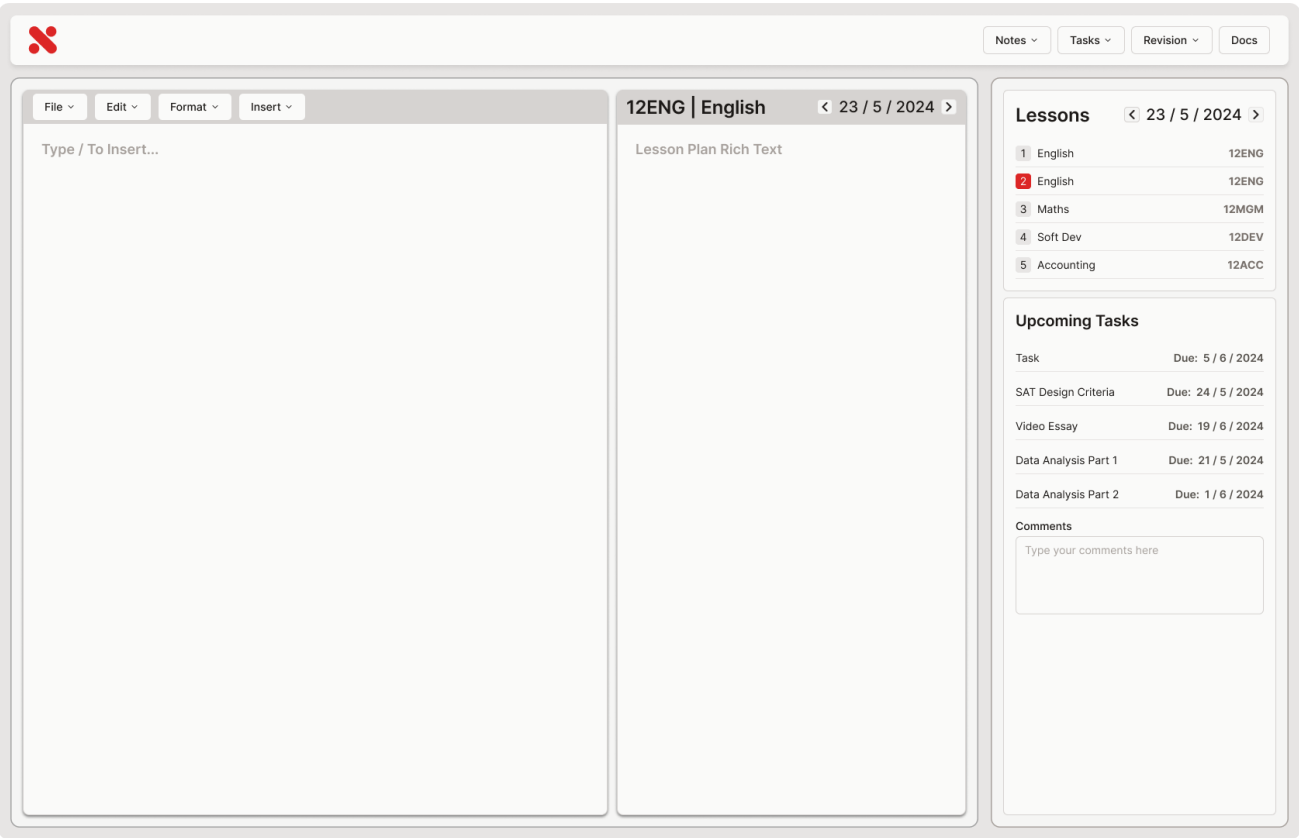


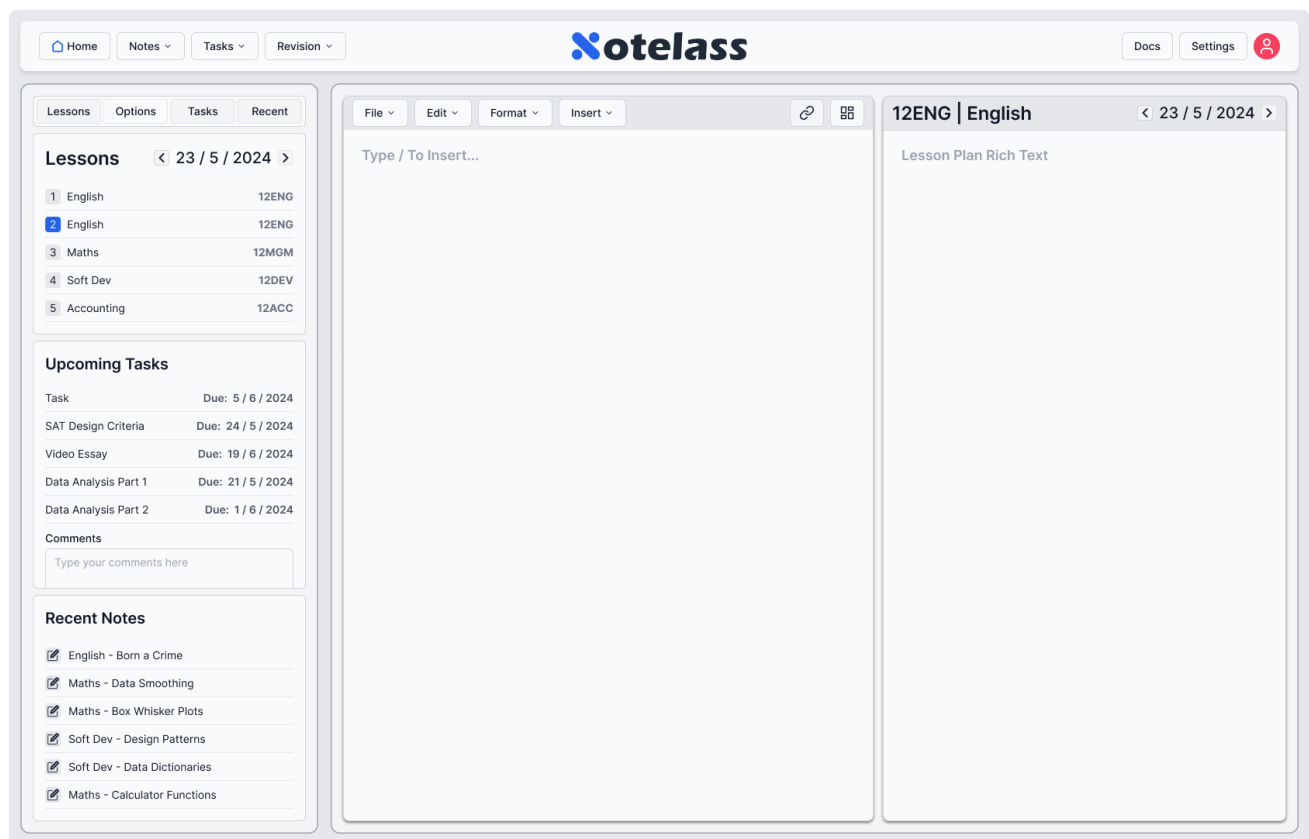
Design

Client Design Evaluation

Design 1



Design 2



Design Criteria Evaluation

 Created from client meeting: [Meeting Notes](#)

Description	Design 1	Design 2
Login page design is minimal and intuitive with simple compass login form	X	X
Documentation Page is accessible from the home page without login	X	X
Each page shares same navbar with intuitive navigation		X
All authenticated pages share sidebar with important information		X
Home page allows user to create, view and manage their notes	X	X
Editor allows users to write markdown or use familiar rich text formatting	X	X
Editor page shows side by side editor and lesson plan	X	X
App shares similar theme and branding to compass interface		X
App uses icons to represent functionality where applicable	X	X
App interface shows all relevant data with minimal navigation		X
App is not excessively verbose or complex for users	X	X

Justification

Design 2 best fits the criteria discussed with the client. It provides a simplistic and intuitive interface for users and maintains a theme stylistically similar to the Compass website it will integrate with. Compared to design 1, design 2 better provides the user with all useful information and navigation on the websites sidebar minimising context switching and allowing for efficient note taking with the application. The navigation bar of design 2 additionally provides separation between common and useful navigation buttons / dropdowns and less used navigation providing an intuitive experience.

Final Design

Mockup

HomeNotesTasksRevision

Notelass

DocsLogin

Login to Notelass

Email

Password

.....

Use Compass Login

Login with Compass

HomeNotesTasksRevision

Notelass

DocsSettings

LessonsOptionsTasksRecent

Lessons

< 23 / 5 / 2024 >

1 English12ENG

2 English12ENG

3 Maths12MGM

4 Soft Dev12DEV

5 Accounting12ACC

Upcoming Tasks

TaskDue: 5 / 6 / 2024

SAT Design CriteriaDue: 24 / 5 / 2024

Video EssayDue: 19 / 6 / 2024

Data Analysis Part 1Due: 21 / 5 / 2024

Data Analysis Part 2Due: 1 / 6 / 2024

Comments

Type your comments here

Recent Notes

English - Born a Crime

Maths - Data Smoothing

Maths - Box Whisker Plots

Soft Dev - Design Patterns

Soft Dev - Data Dictionaries

Maths - Calculator Functions

Search Notes...

+ New Note

Title

Date

Activity

... Born A Crime Topics13 / 3 / 24English

... PSM - Problem Solving Method...13 / 3 / 24Soft Dev

... Born A Crime Topics13 / 3 / 24English

... PSM - Problem Solving Method...13 / 3 / 24Soft Dev

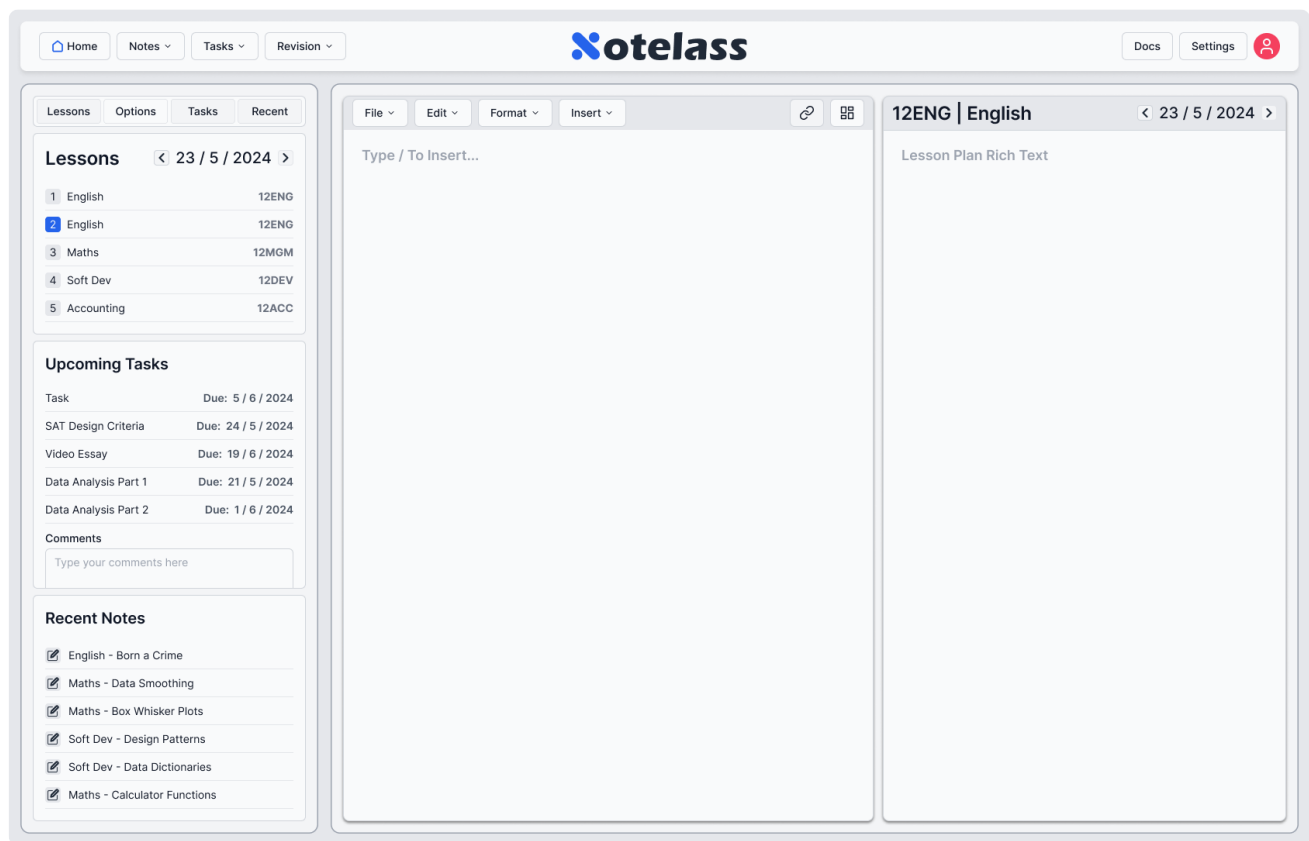
... Born A Crime Topics13 / 3 / 24English

... PSM - Problem Solving Method...13 / 3 / 24Soft Dev

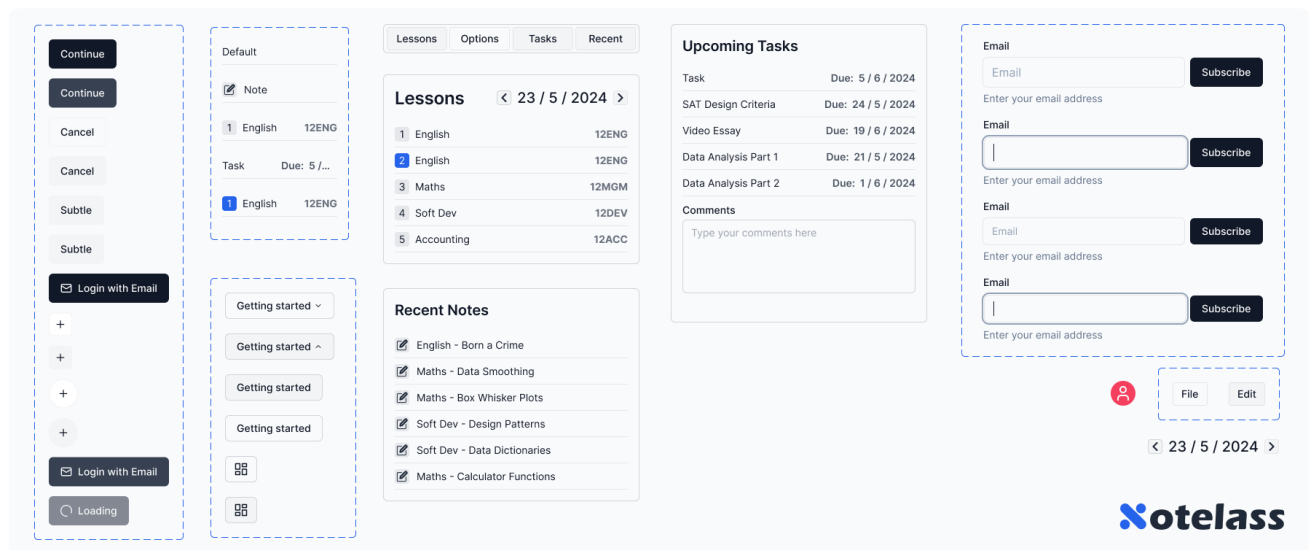
... Born A Crime Topics13 / 3 / 24English

... PSM - Problem Solving Method...13 / 3 / 24Soft Dev

No more notes were found...



Components



Data Dictionary



Compass API Documentation: <https://idkfelix.github.io/compass.js>

Pseudocode

BEGIN

Save note in compass database with metadata

FUNCTION saveNote({content, title, lessonId, activityId})

Encode content to storable string

encoded ← base64Encode(content)

Make request to compass api with data to store

```

response ← POST(compassAPI → saveData, {
  title,
  encoded,
  lessonId,
  activityId,
})
  
```

}}

Return response note id

RETURN response → noteId

END FUNCTION

Read note from compass to local client

FUNCTION readNote(noteId)

Make request to compass api with note id

response ← POST(compassAPI → readData, noteId)

Extract data from response

data ← response

Convert encoded data to readable text

content ← base64Decode(data → encoded)

Return note content and metadata

```

RETURN {
  data → title
}
  
```

```

        content,
        data → lessonId,
        data → activityId,
    }
END FUNCTION

*User inputs note title and content*
localTitle ← INPUT note title
localNote ← INPUT note markdown content
*User selects lesson to link note to*
lesson ← SELECT lesson from schedule

*Save note to compass*
noteId ← saveNote({
    localTitle,
    localNote,
    lesson → lessonId,
    lesson → activityId,
})

*Retrive note from compass for viewing and editing*
noteData ← readNote(noteId)

END

```

Evaluation Criteria

Functional

ID	Requirement	Notes
FR1	User can login with Compass	Login page makes request to Compass to authenticate details and returns user details in a JWT for further requests
FR2	Only authenticated users can access tools and data	JWT verification on all routes to only be used by authenticated users
FR3	Unauthorised users can access documentation	Open route for viewing documentation that allows unauthenticated users
FR4	Requests to compass API are authorised	Compass API requests are properly authenticated with data from the JWT without errors
FR5	Functional note editor with markdown and LaTeX	WYSIWYG editor works seamlessly live rendering of markdown syntax / LaTeX formulas and loading / exporting to plain text

ID	Requirement	Notes
FR6	Notes are stored in and retrieved from Compass	Exported notes are saved in Compass with appropriate metadata (subject, class and date) on the users account and all notes can be retrieved efficiently for viewing or further editing by the user
FR7	Class resources can be viewed	Class lesson plan and resources are fetched from the compass API for viewing by the user when editing notes
FR8	Notes can be filtered by subject, date and class	Notes overview page has filters for metadata that allows the user to refine displayed notes and search for subject, date and class
FR9	User can view upcoming tasks from Compass	Learning tasks for the user are fetched from Compass and are displayed in a timeline in addition to user defined tasks.
FR10	Documentation is dynamically rendered	Documentation statically stored in markdown files on the server is rendered and displayed with the notes editor in the documentation route
FR11	Admin can optionally enable AI revision for users	As AI use requires a paid API key it should be be optionally enabled with environment variables, if disabled the routes and pages for revision tools will not be shown to users

Non-Functional

ID	Requirement	Notes
NFR1	Intuitive Design	90% of users will be able to create and retrieve a note without any prior instruction
NFR2	Accessible Themes	Users should have at least 3 different interface themes to choose from when using the website.
NFR3	No downtime	The website should implement a CI/CD system with serverless hosting to allow for seamless version upgrades
NFR4	Backwards compatability	New versions of Notelass should not break the functionality of existing notes and should allow for continuous use
NFR5	Quick navigation	A user should not require more than 5 clicks to find any of their notes throughout the app