

Revology – The Ideal Revision Website

Table of Contents

Planning.....	2
Introduction.....	2
Project Timeline.....	2
Initial Brainstorm.....	2
Market Research.....	4
Layout and Design.....	4
Vision Board.....	8
Database Layout.....	11
Alternative Approaches.....	13
Coding.....	13
Python.....	13
HTML.....	13
CSS.....	14
JavaScript.....	14
Hosting the site.....	16
Testing and refinement.....	17
Iterative Development Approach.....	17
Ethical Considerations.....	18
Feedback.....	19
Year 12 Student Survey.....	19
Conclusion and next steps.....	19
Try Revology.....	19
Ideas for Version 2.....	20
Final Words.....	20
Bibliography.....	20
Notes about the bibliography.....	20
References.....	20

Planning

Introduction

During my GCSEs, I found that although there were many great resources available, it was often hard to find them in one place. One of the main

problems as a student was figuring out which platforms to use, what to use them for and how to keep all of my revision together. This sparked the idea to create a revision website that acted as a library of resources, where a student could revise, track their progress and access other resources in one hub.

Project Timeline

As shown by the Gantt chart in Figure 1, the project spanned a period of 18 months, from January 2025 to June 2026. A Gantt chart has been used to plan this project as opposed to a regular table because it allows tasks to be carried out simultaneously and provides a visual representation of how time will be divided. This overlapping structure reflects the iterative development approach used throughout the project, where design, coding and testing were not continuous rather than strictly sequential.

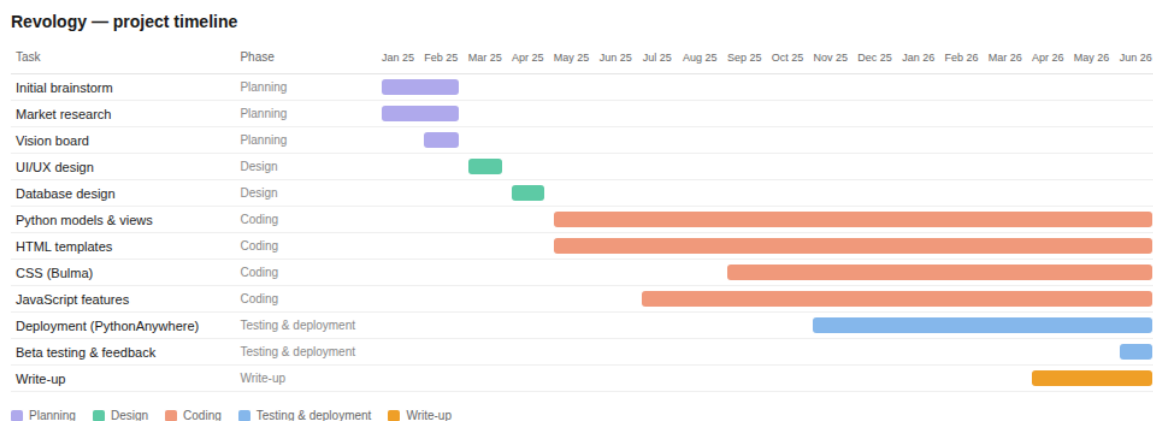


Figure 1: Revology Gantt Chart

Initial Brainstorm

The initial idea consisted of two main components, flashcards and notes. Users could create a set of notes which would automatically upload to a library for others to access, and therefore they could also access other people's notes. There could also be methods such as verification badges to add a layer of confidence when accessing notes. Likewise for flashcards, they would exist in sets, and these sets could be accessed by other users. The website would grow with its users to become self-sustaining, creating a hive of revision resources. Thus, the idea for Revology was born.

During the initial brainstorm, many potential features were considered including a notes library, student forum, online flashcards, study planner and leaderboard. While this initial plan was ambitious, it helped abstract the most important features and provides a baseline for future updates. Figure 2 shows a screenshot of a section of the initial plan, which outlines

the vision for a notes library. Being a fundamental concept for the website, this plan helped solidify a clear outline as well as extra features (such as the incentive to upload notes) that may appear in following versions.

```
Final Project > plan.txt
1  REVISION WEBSITE PLAN
2
3  potential features:
4
5  - notes library
6    - must be an incentive (e.g. coins) for users to upload resources
7    - way for users to create their own notes on the website
8      - typed notes in a handwriting font with the option to add diagrams and labels (laptop friendly)
9      - automatically added to the notes library unless the user changes the setting manually
10   - menu with all the subjects displayed for the user to choose which one to view
11     - division between GCSE / A-Level
12   - the option of website notes and user notes (display user under their authored notes which leads to user profile)
13     - users can request for their notes to be verified (fact checked) which will earn a badge
14
```

Figure 2: Notes Library Initial Brainstorm

Similarly, the other features were researched (Figure 3) and this helped rule out a student forum, because it would open a new niche of networking rather than revising. There are already platforms for this, such as The Student Room which is a forum which focuses specifically on education and student lifestyle. A study planner and leaderboard were also temporarily ruled out because their functions are to assist with studying, rather than providing the revision itself.

```
15  - student forum
16    - works in a similar way to yapper except has different threads for each question
17      - before starting a new thread there should be an option to view similar existing threads
18      - delete threads after two years of inactivity if there are more than x number of threads
19      - menu showing the titles of each thread and then each thread has its own page of replies
20    - can use tags (like hashtags) to make it easier to search / filter through threads
21      - separate tab under user profile for forum posts so they don't mix with notes and flashcards
22
23  - online flashcards
24    - users would make their own flashcards or can access other users' sets
25      - like the notes library these are automatically available publically unless the user changes the setting manually
26    - flashcard rating system (green, amber, red) to revise specific topics
27      - will get a user to sort through all the red flashcards (and moves them to amber)
28      - then all the amber flashcards (and moves them to green)
29    - some sort of quiz feature? e.g. some words are blanked out and the user has to fill in the gaps
30      - the user can pick which words / phrases to blank out to use this feature to prevent unhelpful recall
31
32  - study planner
33    - drag and drop blocks to create a timetable for each day of the week
34      - these prescheduled blocks can be marked complete once the user as done them
35    - subject tracking to see how many hours etc per subject via...
36      - stopwatch
37      - countdown
38      - or a prescheduled block which has been marked complete
39    - (daily) todo list
40      - tick box checklist of homework etc
41      - boxes that are ticked move to a 'completed tasks' list (unless the user deletes them)
42      - this is not visible to other users
43
44  - leaderboard with friends
45    - based on time revising or coins etc (can include coins from uploading notes)
46      - includes all the users which the user is 'friends' with
47    - add friends using their username or directly on their profile page
48      - user profiles can also display their notes / flashcard sets (if they are public)
49
50  - a 'useful links' page (static page)
51    - revision playlist
52    - useful resources sorted by subjects (displayed like a blog)
53      - pmt and other resources
54      - revision yt channels / study accounts
55
```

Figure 3: Remaining Features Initial Brainstorm

Market Research

To understand which resources students are using to revise and how readily available these are, a survey was carried out by 78 students from Year 7-12. Firstly, 83% of students do not feel completely sure how to revise, so providing them with more guidance in this area would be very helpful. 65% of students surveyed make their own notes, with a further 18% using pre-made notes to revise. Similarly, 66% of participants use flashcards to revise.

3. Do you use notes to revise?

[More details](#)

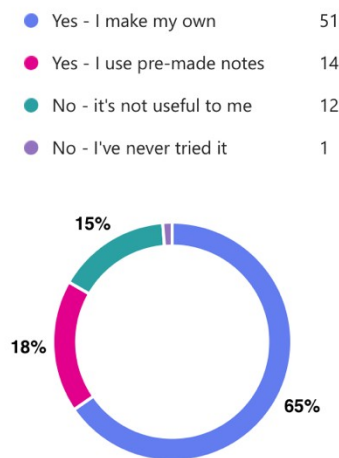


Figure 4: Survey Data

Regarding existing access to resources, 45% of students said the availability of free revision resources was between “very limited” and “somewhat alright”. 58% of students said finding pre-made notes for their subjects “requires a bit of digging” as did 48% for online flashcards. It’s clear that a common platform where all resources can be accessed from one place is currently lacking.

To analyse the trends more closely, it’s important to note that this sample has been conducted solely from one school. The school’s portfolio is independent, highly competitive and academic. Therefore, the responses will tend to differ slightly from a poll of the general population. It is likely that students surveyed have an easier time finding resources and revising due to their circumstances. This means the statistics are actually underplaying the national situation.

Layout and Design

Once the features were outlined, the next step was to draft out a visual plan of the webpages and their layout. This helped form a more concrete idea of how the website would work. Each page was designed on Canva, using the colour blue as it is associated with trust and serenity. Most of the initial features from the initial brainstorm have been included, regardless of whether they would be implemented into Version 1.

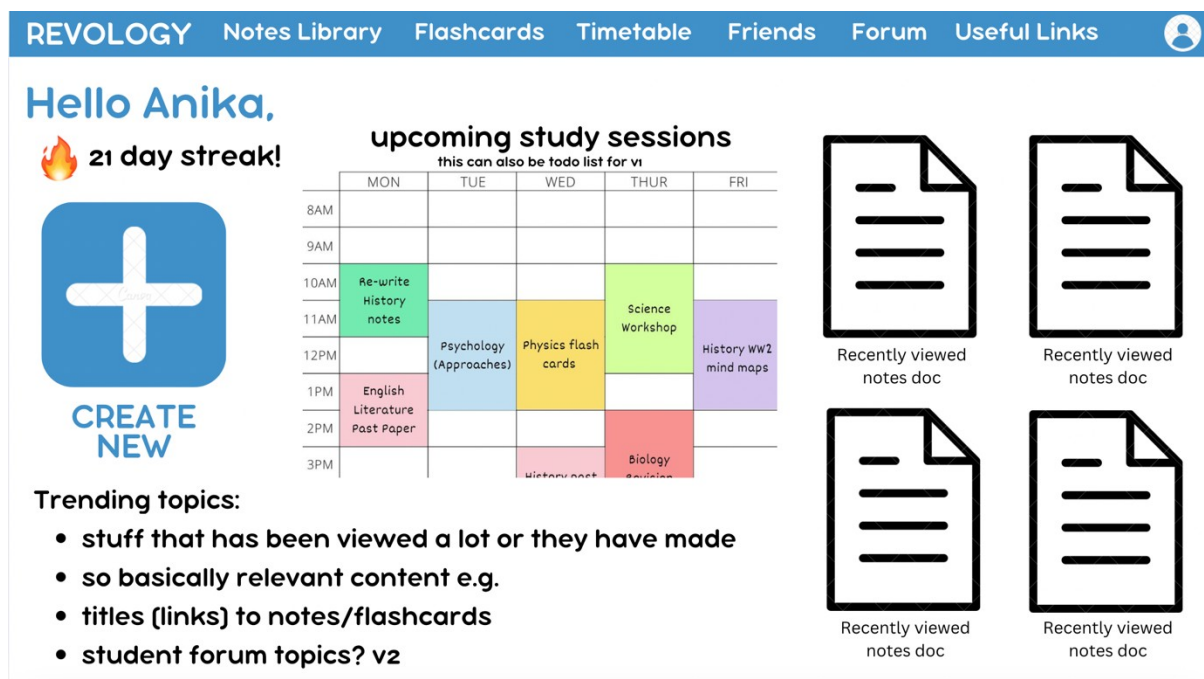


Figure 5: Homepage UI

Figure 5 is a plan for the User Interface of the homepage, which contains a large button encouraging the user to create a new resource. It also clearly displays their timetable, recently viewed documents and general trending topics to draw the user into the site. The navigation bar has also been carefully crafted so that each button is exactly where it should be intuitively, improving user experience. UI/UX is particularly important for the target user age of 14-21, as research suggests that colour and interface quality significantly influence trust and engagement in digital environments (Elliot, 2025).

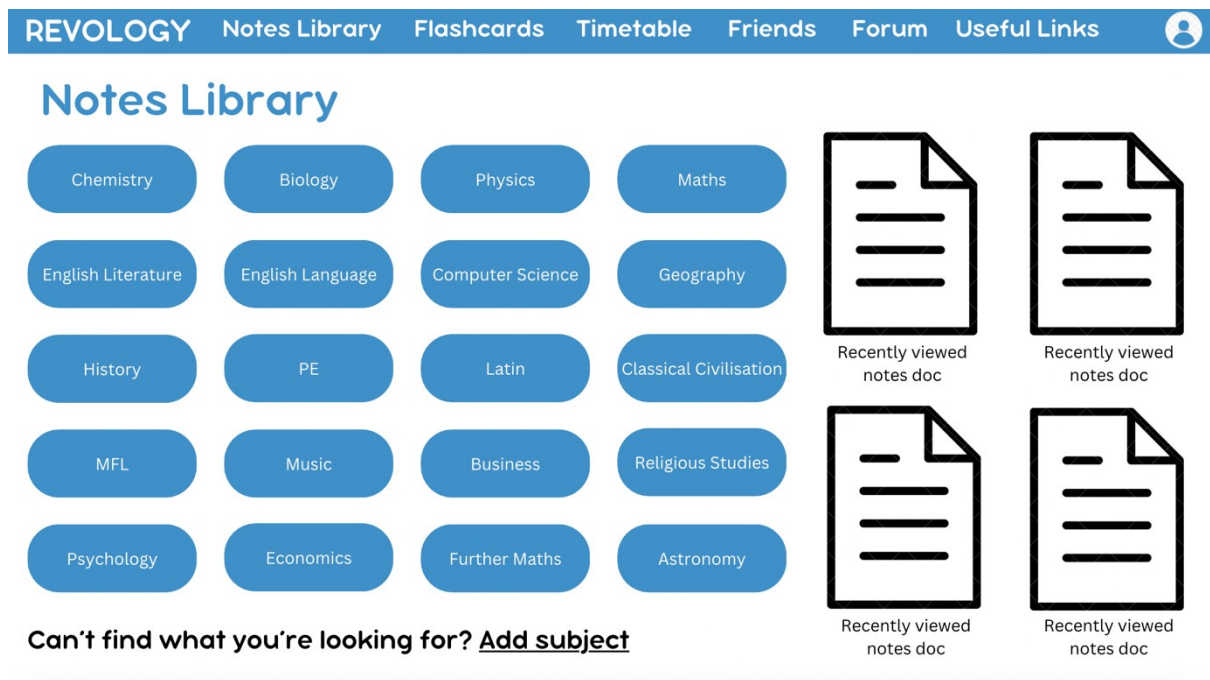


Figure 6: Notes Library UI

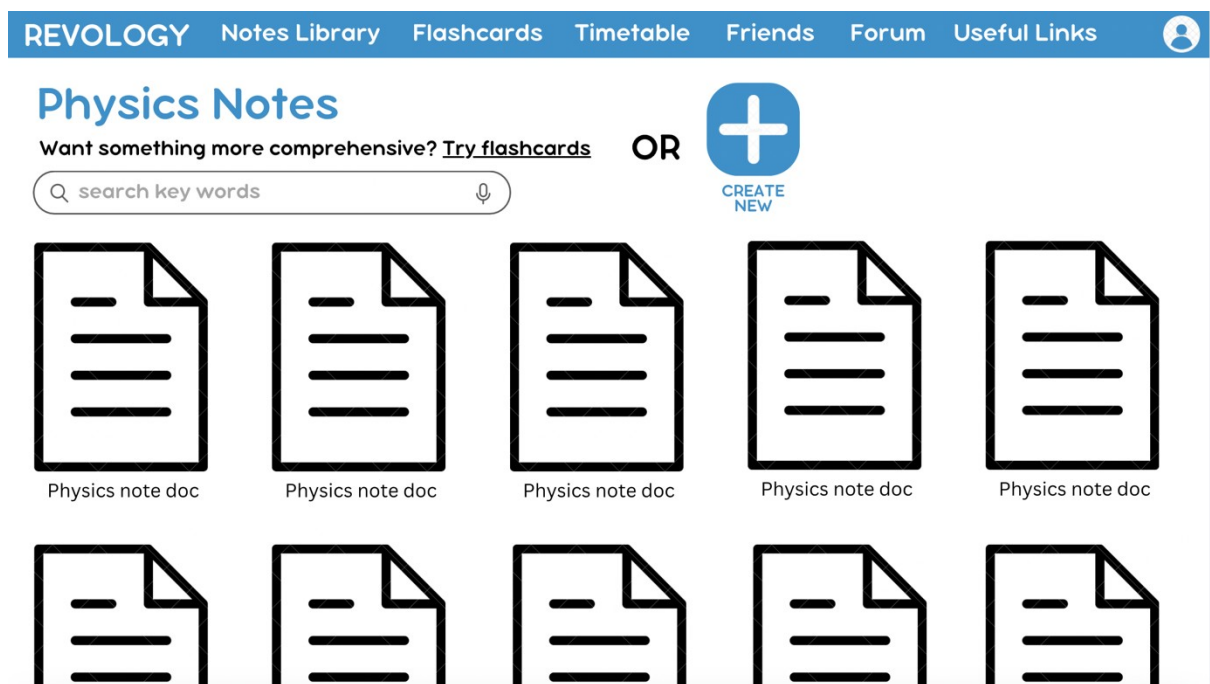


Figure 7: Subject Notes UI

Figures 6 and 7 show the initial design for the notes library interface. Figure 6 shows main page of the notes library, and once a subject is selected, it will lead the user to the page shown on Figure 7. Users also have the option to create their own notes if they wish.

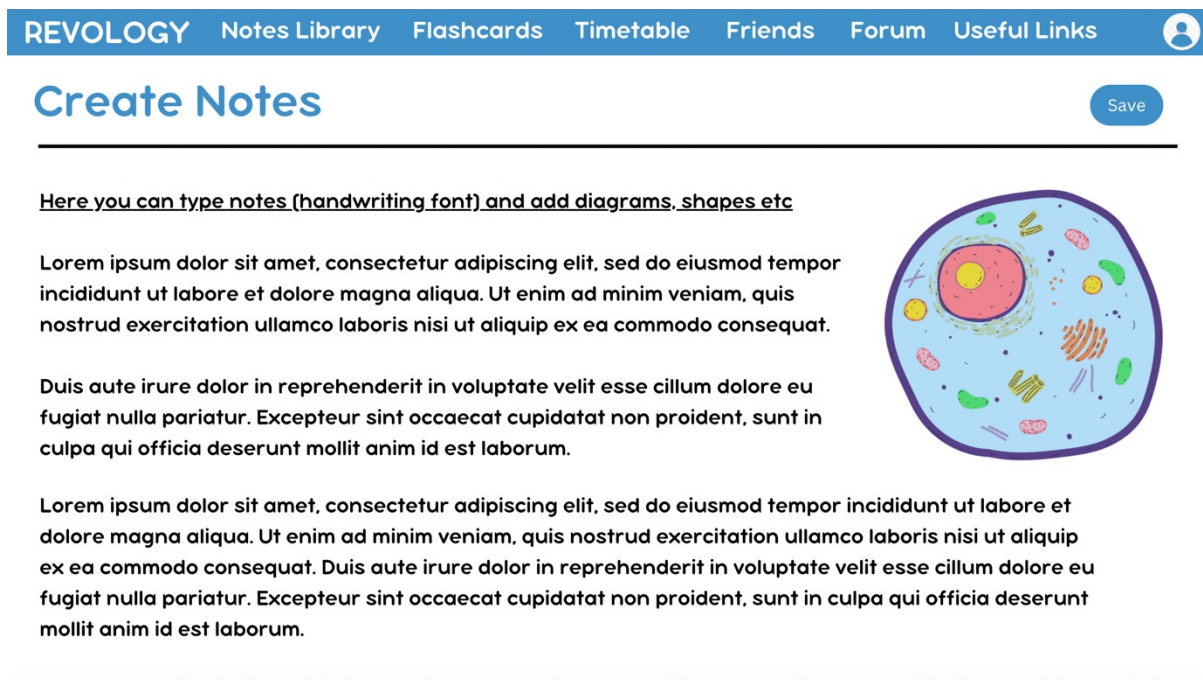


Figure 8: Create Notes UI

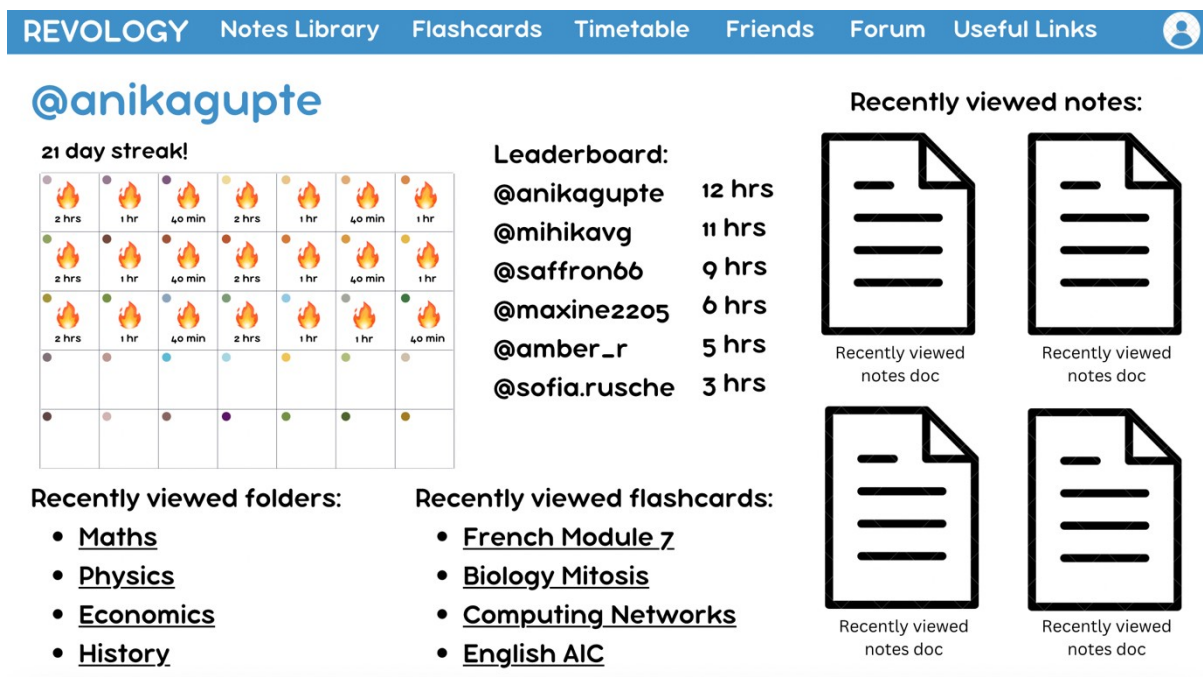


Figure 9: User Profile UI

Another feature of the website is the “social” aspect, where users can view either their own or another user’s profile. Elements of this page could include a leaderboard, activity streak and recently viewed resources. This adds to the UX by creating a personal touch to the platform.

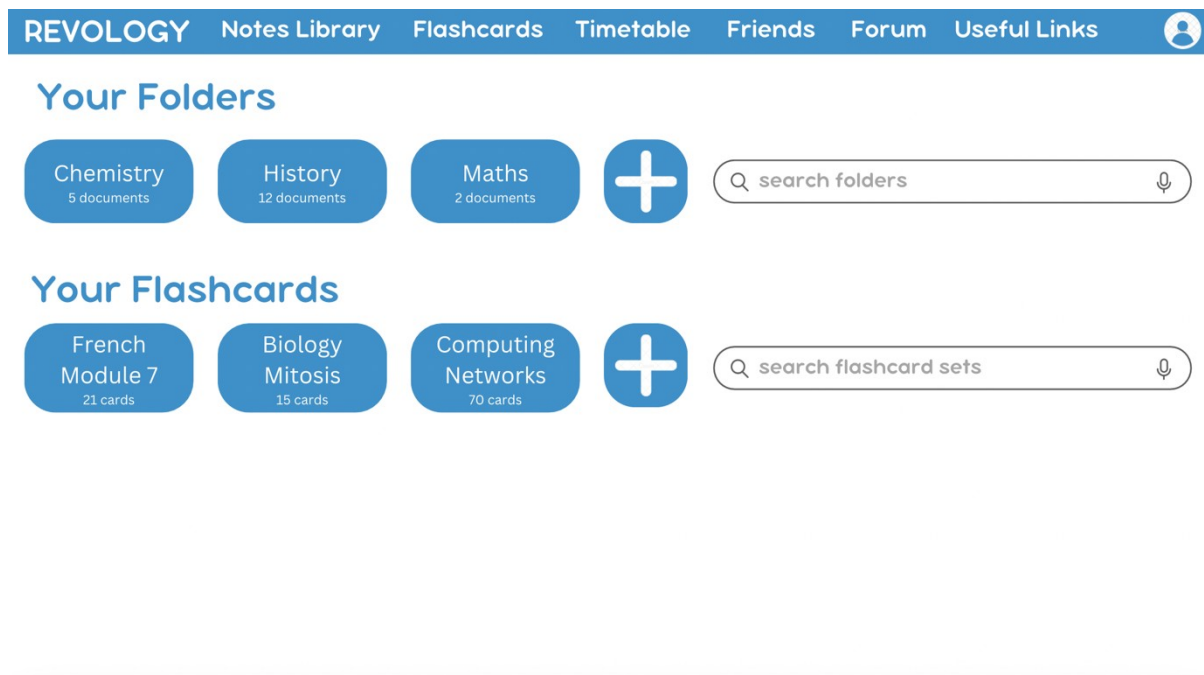


Figure 10: User Resources UI

Here, a user can view their own resources. There is also a search feature for easy access, and this will work as a keyword search, so the input text does not need to exactly match a title. This is the same for all search functions on the website.

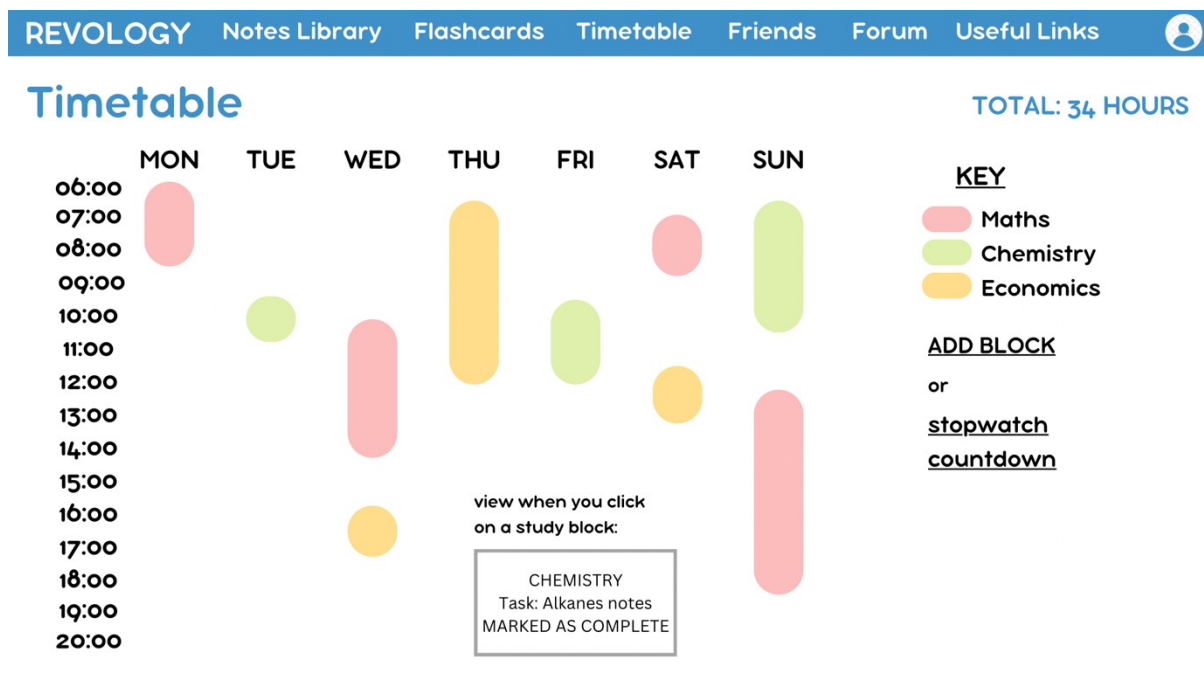


Figure 11: Timetable UI

As an additional feature, a timetable (Figure 11) could be implemented for the user to track their studying as well as plan for future sessions. When

each of the coloured blocks is clicked, it opens a pop-up such as the CHEMISTRY block above and shows some extra details of the study block.

Vision Board

The next stage of planning was creating a vision board for the website. This was created intentionally over-ambitious, because thinking bigger automatically helps to create a better product. The vision board, originally created in Excel, has been pasted below.

Vision: create a useful revision website with multiple features			
Audience / Roles	Needs	Product	Business Goals
Users / Students	access notes / flashcards conveniently	homepage prompting user signup and displaying sample notes + student recommendations of the website maybe a video on how to get started?	500 users in 1 month of launch
Website Notes Creator + Verifier	test understanding of topics	navbar sections:	eventual expansion to become a Quizlet type software
	create notes / flashcards	notes library (sorted subject-wise) displays boxes of each of the main subjects and then a box for any other subject the user may have made notes for (the subject is selected compulsorily when creating notes) which once clicked on lead to a page for that subject containing all of the notes for that subject (and any website notes at the top) OR displays all the notes and a search bar on top so	

		users can search for notes which have tags of their subject/s website notes are automatically prioritised at the top of search results users can request for their notes to be verified (fact-checked) to earn a badge for more trust	
	learn from other people's resources	timetable + study tracking drag and drop boxes to fill up a one day/week/month schedule to plan when to revise each subject which can be marked as complete once done daily checklist of tasks they intend to do each day for revision: - boxes that are ticked move to a 'completed tasks' list (unless the user deletes them) - this is not visible to other users stopwatch or countdown to time their work which automatically gets added to their total hours + timetable (marked as complete)	
	plan timetable	user's profile (top right corner) able to view their own notes/flashcards from their profile, as well as their progress, such as total study hours (that are logged) or number of notes created	
	track study hours	friends tab able to add "friends" (from links under their resources	

		or by search) who will appear on their leaderboard which is based on hours spent revising (all time + this month) - other user profiles would display their notes / flashcards to all users "friend" exclusive features e.g. some progress is also visible to users who you are "friends" with such as total study hours	
	motivation	other resources tab just a blog-like page with 'useful links' to other resources (such as past paper questions, playlists, study accounts)	
	track own progress	create notes / flashcard set when creating they have to select a subject (or add new subject) so it can be sorted in the notes library - typed notes in a handwriting (or normal) font with the option to add diagrams and labels (laptop friendly yet still aesthetic) - automatically added to the notes / flashcards library unless the user changes the setting manually can have tags to easily find their own notes the users can sort their notes into folders of their choice they can then go through flashcards / read their notes to learn content	

		there should also be an option to complete a quiz from a flashcard set which can be AI generated (this is dependant on good quality flashcards)	
	ask questions about revision	student forum has different threads for each question - before starting a new thread there should be an option to view similar existing threads - delete threads after two years of inactivity if there are more than x number of threads - menu showing the titles of each thread and then each thread has its own page of replies can use tags (like hashtags) to make it easier to search / filter through threads unrelated to notes / flashcards but can reference a particular document e.g. to ask a question about something mentioned in some notes	
		functionality for user logins / accounts	

While a few of these features were eliminated after the initial brainstorm, like the UI/UX design, they were still planned for future versions.

Database Layout

To plan out the database, all the required tables were planned (on Excel) and their primary and foreign keys defined. These are shown in Figures 12 to 17.

	A	B	C	D	E	F	G	H
1	username (pk)	email_address	first_name	last_name	streak (in days)	authored_notes (fk)	authored_flashcards	todo_list
2	anikagupte	anika@email	Anika	Gupte	21	1	2,6,11	text items

Figure 12: Database Design - User Table

The first four columns in Figure 12 have been shaded in because they are already provided by the Django database administrator, one of the advantages of choosing this framework.

	A	B	C	D	E	F
1	id (pk)	content	author (fk)	subject (fk)	date_uploaded	title
2	1	some text and images formatted nicely	anikagupte	Physics	24/03/2025	"Moments"

Figure 13: Database Design - Notes Table

	A	B	C	D	E
1	id (pk)	author (fk)	subject (fk)	date_created	title
2	1	anikagupte	Physics	24/03/2025	"Nuclear Fission"

Figure 14: Database Design - Flashcard Set Table

	A	B	C	D
1	id (pk)	front_of_card	back_of_card	flashcard_set (fk)
2	1	question text	answer text	1

Figure 15: Database Design - Flashcard Table

	A	B
1	name (pk)	action
2	open_notes	clicking a url for a notes document
3	open_flashcards	clicking a url for a flashcard set
4	create_notes	saving a new notes document to the library
5	create_flashcards	saving a flashcard set to the collection
6	edit_todolist	editing the user's todo list

Figure 16: Database Design - Activity Table

	A	B	C
1	user (fk)	activity_type (fk)	date
2	1	edit_todolist	30/04/2025
3	1	open_notes	01/05/2025
4	2	create_flashcards	01/05/2025
5			
6	calculate streak using activity dates		

Figure 17: Database Design - User Activity Table

There are three types of relationships between entities (or tables). These are one-to-one, such as a book having one barcode and a barcode mapping to exactly one book. The next relationship many-to-one; many books can be written by an author, but a book can only have one author. The final type of relationship is many-to-many. For example, in a library, one person can read many books but each book can also be read by many people.

As shown above, notes and flashcards both have a many-to-one relationship with a user, known as the author or creator. Likewise, they also have a many-to-one relationship to subjects. Flashcards exist as individual cards and have a many-to-one relationship with a flashcard set, which is simply a group of related cards.

The activity table was created to track user activity on the site in order to provide the user recommendations based on their recent activity. This feature unfortunately did not make it to Version 1, but in the future, it will also be used to track a user's streak to provide incentive for them to use the platform daily or weekly.

Alternative Approaches

It's important to note alternative approaches that were considered when making decisions on the tools used in this project. For the web framework, the two main Python-based options were Django and Flask. Django was chosen because it provides user authentication, database management and many other administrative tasks, allowing more time to develop the unique features of Revology. Django also provides an admin interface which is particularly useful when dealing with a relational database such as the one designed above.

Regarding the CSS framework, both Bootstrap and Bulma were considered. While Bootstrap is more well-known, it ships with JavaScript dependencies meaning for this Python-based project, Bulma is lighter and easier to integrate. Its documentation is also clear and easy to follow, making Bulma the right choice for Revology.

Coding

Python

The first step of writing the code was creating the database models in python. The Django framework was chosen for this because it handles database admin and allows simplification when creating relationships between entities. The models were created using the database structure explained in the previous section, so it was fairly straightforward to program because it was simply a matter of translating the plan.

The other major Python file is `views.py` which provides the functionality for every single webpage on the site. The views have been written to render pages, provide them with appropriate data and context, and even carry out calculations where appropriate. All of the views have been clearly commented within the code so they are easy to read and follow.

HTML

Once the Python models had been written, the next step was to create the HTML templates for each webpage. This was written simultaneously with the views (in Python) for step-by-step testing and debugging. Each template provides the content of the webpage, as well as any links to the other pages.

CSS

The HTML itself was fairly straightforward until CSS was incorporated into it to provide its design. Using a CSS framework greatly helps to simplify the process, and as previously explained, Bulma was selected over its competitor Bootstrap.

text	
link	
primary	
info	
success	
warning	
danger	

The Bulma framework provides 7 primary colours as shown in Figure 18. Success has been chosen as the main colour for the website because it is a calm and neutral colour. This has been used for the Navbar as shown in Figure 19, as well as buttons across the site. Some features also make use of a lighter shade of success, such as the background of the homepage which incorporates the use of a gradient.

Figure 18:
Bulma Primary

JavaScript

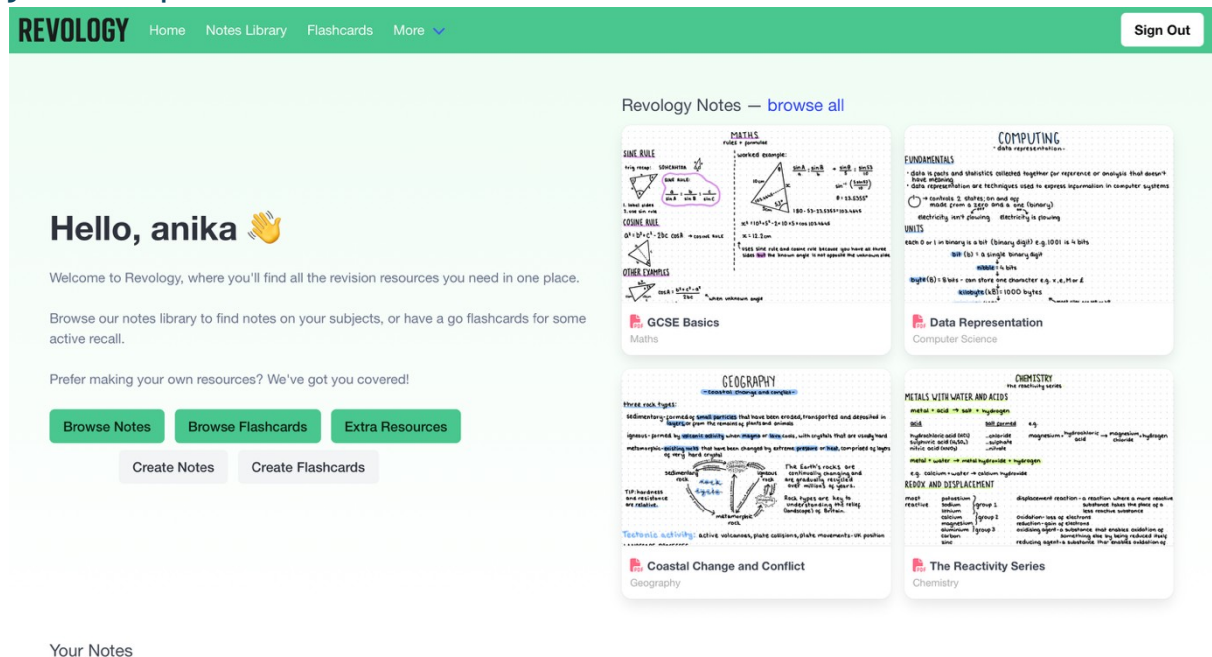


Figure 19: Revology Homepage

While majority of the code for this project was written in Python, two features needed the use of JavaScript embedded into their HTML templates.

The first JavaScript feature is dynamic flashcards; each card needs to be flipped when clicked on. This feature was particularly important for the site to make intuitive sense, because it is replicating physical flashcards that have a term on one side and a definition on the other. The JavaScript code listens for a click event on each card and toggles the CSS class when it finds one. Each card has a front and back inside a container which makes the 3D effect work, as shown in the code snippets below.

```
<div id="flashcard" onclick="flipCard()" style="
  background: white;
  border-radius: 12px;
  box-shadow: 0 8px 32px rgba(0,0,0,0.12);
  min-height: 260px;
  display: flex;
  align-items: center;
  justify-content: center;
```

```

        cursor: pointer;
        padding: 48px;
        text-align: center;
        transition: background 0.2s;
        position: relative;">
        <p id="card-text" style="font-size: 1.4rem; font-weight: 500;"></p>
        <span style="position: absolute; bottom: 16px; right: 20px; font-size: 0.8rem; color:
#aaa;" id="flip-hint">Click to flip</span>
    </div>

function flipCard() {
    if (cards.length === 0) return;
    flipped = !flipped;
    const card = cards[current];
    document.getElementById('card-text').textContent = flipped ? card.back : card.front;
    document.getElementById('flip-hint').textContent = flipped ? 'Click to flip back' : 'Click to
flip';
    document.getElementById('flashcard').style.background = flipped ? '#f0fdf4' : 'white';
    document.getElementById('rag-section').style.display = flipped ? 'block' : 'none';
}

```

The second feature that required JavaScript is displaying PDF previews of Revology notes on the site. The concept of Revology notes is that there is a set of verified GCSE notes for each subject, which have more credibility as they have been created by the Revology team rather than other users. However, when implementing these as buttons or boxes like the rest of the notes on the site, their uniqueness was not visible. Hence, previews were created so that users can see an example of the notes and are therefore more likely to click onto them to view the full PDF. These PDF previews appear on both the homepage and within the notes library (after selecting a particular subject). The code from PDF.js (Mozilla's library) has been used to render the first page in a <canvas> element because PDFs cannot be used natively in a <div> element without a library. The previews elevated the design of the page as well as prompting users to click on the documents, improving the overall site.

```

<script>
    pdfjsLib.GlobalWorkerOptions.workerSrc =
'https://cdnjs.cloudflare.com/ajax/libs/pdf.js/3.11.174/pdf.worker.min.js';
    document.querySelectorAll('.pdf-thumb').forEach(canvas => {
        const url = canvas.dataset.url;
        pdfjsLib.getDocument(url).promise.then(pdf => {
            return pdf.getPage(1);
        }).then(page => {
            const viewport = page.getViewport({ scale: 1.5 });

```

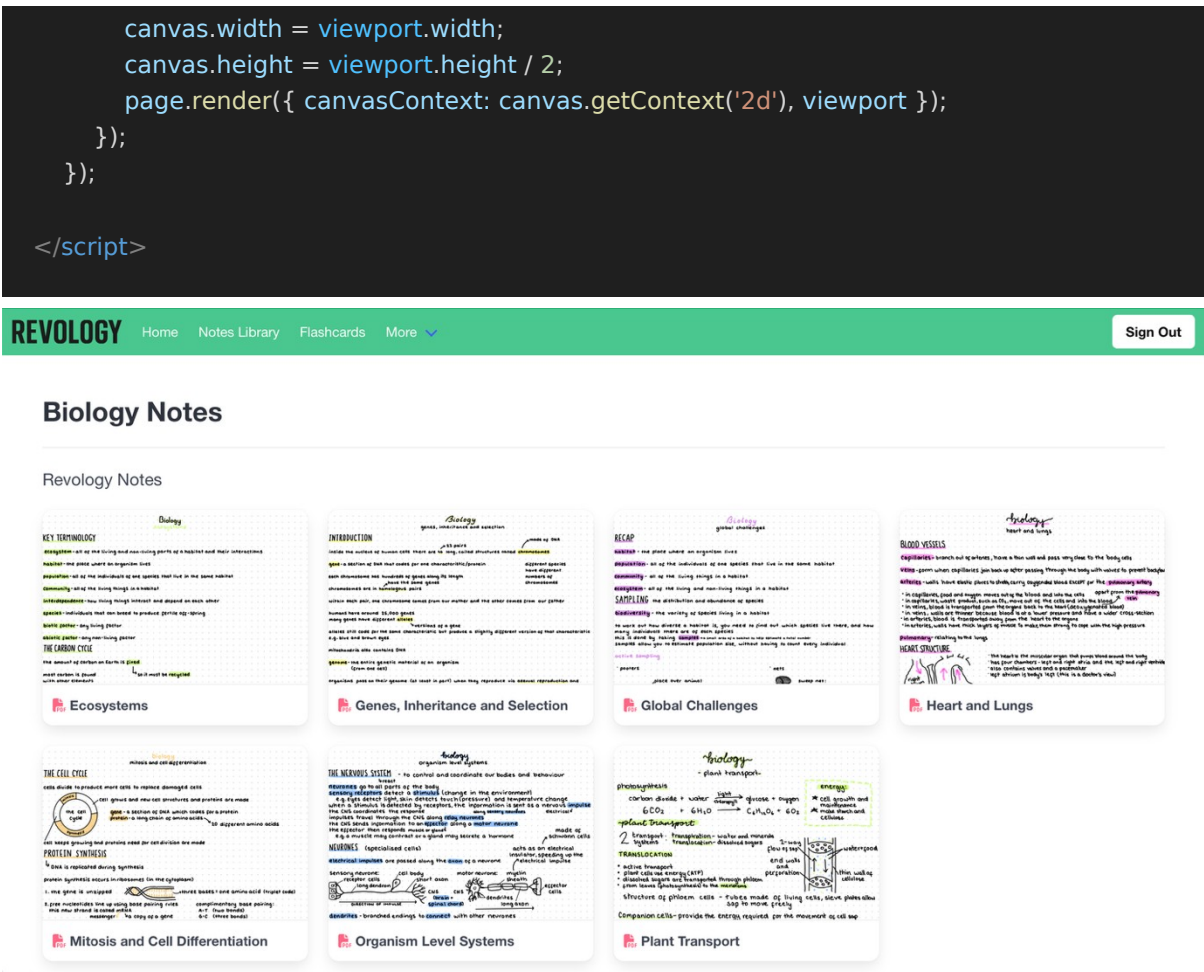


Figure 20: Within the Notes Library - Biology

Hosting the site

This website will be hosted on PythonAnywhere, a platform that allows websites to be hosted for free within a certain storage and CPU usage allowance. A Bash console was used to pull the code from the GitHub repository, with utilised my previous familiarity with Linux commands. The models had to be re-migrated onto this console and every time a change was made to the code from this point, the updated code needed to be pulled from GitHub again.

Testing and refinement

Iterative Development Approach

For this project, the iterative development approach was used. As shown in Figure 21, this program development life cycle involves creating each feature by individually



Figure 21: Iterative Development

designing, developing and testing it. These small cycles are known as iterations. This is more effective than a traditional waterfall method because it incorporates lots of feedback and refinement. This underpins the Agile methodology widely used in industry (Larman and Basili, 2003), which allows each feature to be designed, built and tested in small cycles.

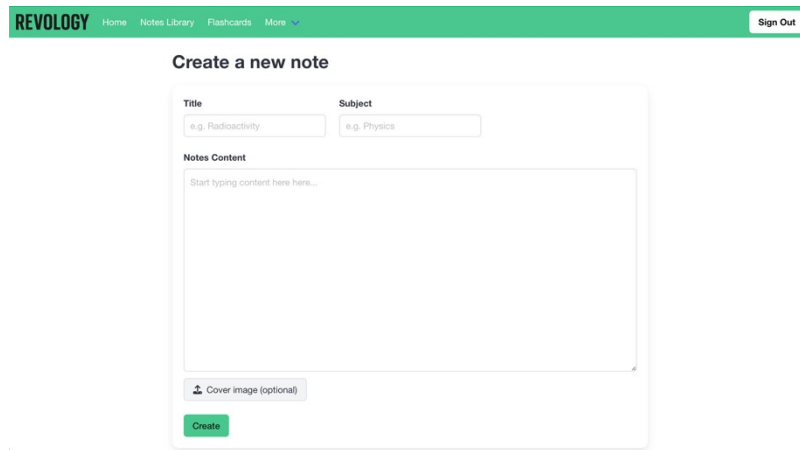
The image shows a web application interface for creating a new note. At the top, there is a green navigation bar with the text 'REVOLGY' on the left and 'Sign Out' on the right. Below the navigation bar, the main heading is 'Create a new note'. The form consists of several input fields: a 'Title' field with the placeholder text 'e.g. Radioactivity', a 'Subject' field with the placeholder text 'e.g. Physics', and a large 'Notes Content' text area with the placeholder text 'Start typing content here...'. Below the text area, there is a small button labeled 'Cover image (optional)' with an upload icon. At the bottom of the form is a green 'Create' button.

Figure 22: Create a new note

Once the frameworks of each page were written in HTML, they were designed and refined using this approach. The design of the website took the greatest proportion of refinement time. Each small change, such as fixing the spacing of a title or changing the colour of the

navigation bar was altered and tested numerous times before being finalised. This allowed me to learn more about the importance of divs, boxes, classes, sections and more in CSS.

For example, the flashcards filtering feature was particularly tricky. This involves a rating system, where users can assign a rating to each

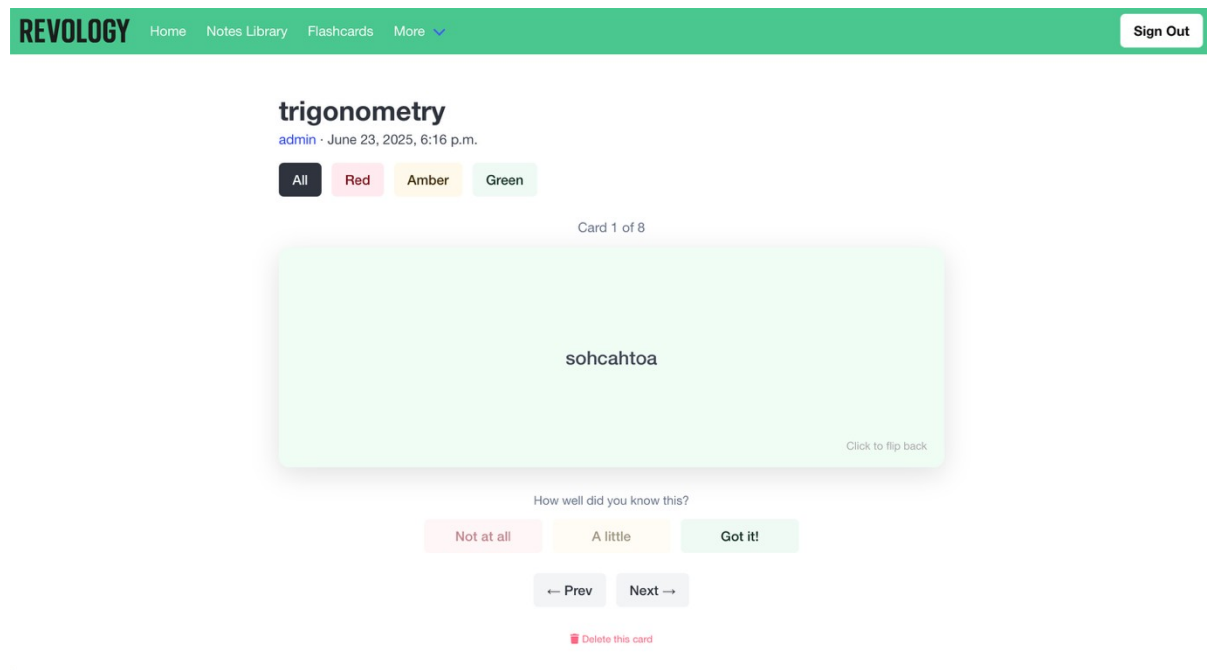


Figure 23: Flashcard Filtering

flashcard depending on their confidence. Then, they can choose to only practise the cards within a certain confidence bracket. The ratings follow the three simple traffic light colours, because these are universal visual indicators. When filtering the cards by colour, there were many bugs from the ratings not updating on the site to the cards disappearing entirely. After finally reaching a stage where the feature could be tested, it became evident that there needed to be a shortcut to change all of the ratings for a set of flashcards back to red, hence the reset flashcards button was born. This method of design → development → testing is the perfect illustration of a single iteration within the development life cycle.

Other issues that required debugging included not having a login requirement on certain pages which caused errors when a user tried to carry out any actions, forgetting to migrate the database after making changes to models and running out of free storage space on Python Anywhere!

Ethical Considerations

As the site collects data from users, an important risk to consider is that of data protection and its surrounding legislation, such as GDPR and the Data Protection Act. Upon registration, the site asks for a first and last name, email address, username and password. All authentication and credential handling is handled by Django's built-in administration system, which encrypts passwords and follows security best practises. User-created resources are also stored in the database, but this falls within the scope of

UKGDPR which requires that data is collected lawfully, stored securely and only used for its stated purpose (ICO, 2021).

For the student surveys and beta testing, all participants engaged with the material willingly and with the knowledge that their data would be used in this write-up. The survey was conducted anonymously and no personal information (other than year group) was collected as part of the process.

Feedback

Year 12 Student Survey

The site was shared with a sample of Year 12 beta users, who commented on the usability, appearance and structure of the website. Overall, they agreed that Revology successfully identified and filled the gap for a shared library of resources. One student wrote “it’s really useful to divide Revology notes and community notes because I know I can trust what I’m reading”. This shows an appreciation for credibility of resources, and a way to improve this further could be a verification system for user-created notes. When asked where the website could be improved, another student suggested “you could add past paper questions organised by topic”. This is the next step in the revision process, and implementing questions in a future version of the software is a strong idea.

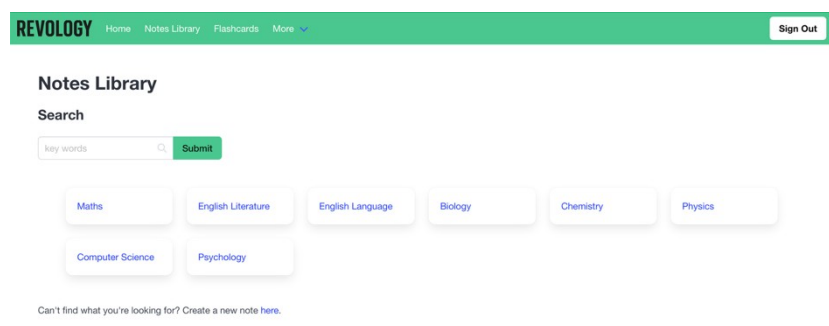


Figure 24: Notes Library

Conclusion and next steps

Try Revology

The full code for Revology can be viewed at the GitHub repository available at this link: <https://github.com/anikagupte/revology>

The site will require initial data (such as notes, flashcards and some users) before it can officially be launched to the public, but the beta platform can be viewed here: <https://revology.pythonanywhere.com/supernova/signup>

Revology sign up here

Firstname

Firstname

Lastname

Lastname

Email

email@example.com

Username

Username

Password

password

Sign up

Already have an account? [Login](#)

Figure 25: Revology Sign Up Page

Ideas for Version 2

In the initial brainstorm, there were many additional features that were not implemented in Revology due to time constraints. However, if there was to be a Version 2 of the platform, the first thing to implement would be the activity tracking – streaks, leaderboards and interactive timetables. This would add a competitive element to the website and “gamify” revision, a technique that has been used by many large platforms such as Duolingo to keep their users engaged over long periods of time (Shortt et al., 2023).

Final Words

Overall, I am very proud of how Revology turned out. It was a very large project, that has been in development for around 18 months due to balancing it alongside school, A-levels and extracurricular, but it was worth the wait. Nearly 1 in 2 students surveyed at the beginning couldn't find resources easily. This site successfully addressed this issue by collating them all in one shared library, in a way to benefit as many students across the country as possible.

Over this project, I've learnt how to navigate the depths of Django, from user authentication to database administration. I developed using four programming languages, the most I've ever used in one project, and deployed the site on PythonAnywhere, which required Linux knowledge as well. Database relationships and CSS frameworks aside, my biggest takeaway from this project is beyond technical knowledge. It's the understanding of the development process that I've come away with, having spent hours on end writing code, debugging, testing and repeating.

For now, I will be working on building up Revology's database of resources to attract initial users to the site. Thank you for reading my journey and I'd love any feedback you have!

Bibliography

Notes about the bibliography

Throughout this project, two pieces of documentation required constant referral: the Django framework documentation (Django Software Foundation, 2025) and the Bulma CSS component library documentation (Dansart, 2025). These were the pillars for its development.

References

Dansart, J. (2025) *Bulma CSS documentation*. Available at: <https://bulma.io/documentation/> (Last accessed: 14 June 2026).

Django Software Foundation (2025) *Django documentation*. Available at: <https://docs.djangoproject.com/en/5.2/>. (Last accessed: 14 June 2026).

Elliot, A.J. (2015) *Color and psychological functioning: a review of theoretical and empirical work*. *Frontiers in Psychology*, 6, p. 368. Available at: <https://pmc.ncbi.nlm.nih.gov/articles/PMC4383146/>. (Last accessed: 15 June 2026).

Information Commissioner's Office (2021) *A guide to the data protection principles*. Available at: <https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/data-protection-principles/a-guide-to-the-data-protection-principles/>.

Larman, C. and Basili, V.R. (2003) *Iterative and incremental developments: a brief history*. *Computer*, 36(6) pp. 47-56. Doi: 10.1109/MC.2003.1204375.

Shortt, M., Tilak, S., Kuznetcova, I., Martens, B. and Bhatt, B. (2023) *Gamification in mobile-assisted language learning: a systematic review of Duolingo literature from public release of 2012 to early 2020*, *Computer Assisted Language Learning*, 36(6), pp. 517-544. Doi: 10.1080/09588221.2021.1933540.