

UNIVERSITY OF TWENTE

Think Green, Study Green!

Design Project

Group 2

Technical Computer Science

Project Team

Andreea Şerban - s3174069
Tudor Sandu - s3148157
Joel Poerba - s3193527
Fabiana Žukaitė - s3242404
Amir Martijn - s2384892

Supervisor
Gerwin Hoogsteen

October 2026

Contents

1 Introduction **3**
1.1 Background and Motivation . . . 3
1.2 Project Goal and Proposed Solution . . . 3
1.3 Report Structure . . . 4

2 Requirements **4**
2.1 Functional Requirements . . . 4
2.2 Non-Functional Requirements . . . 6

3 Architecture **6**
3.1 UML Diagrams . . . 6
3.2 Backend . . . 6

4 Design **11**
4.1 Framework . . . 11
4.2 Initial Prototype . . . 12
4.3 Key Design Choices & Motivation . . . 12
4.4 Final Product Design and Description . . . 14
4.4.1 Log-In system . . . 14
4.4.2 Main page . . . 15
4.4.3 Course page . . . 15
4.4.4 Courses page . . . 16
4.4.5 Teacher editing page . . . 16
4.4.6 Help page . . . 17
4.4.7 SDG and GreenComp pages . . . 17

5 Integration **17**
5.1 API . . . 17

6 Testing **18**
6.1 Moderated Usability Testing . . . 18
6.1.1 Environment Setup . . . 18
6.1.2 Session Structure . . . 18
6.1.3 Results . . . 18
6.2 Unit Testing . . . 18
6.2.1 Results . . . 18
6.3 API Testing . . . 18
6.3.1 Results . . . 19

7 Team Work **19**
7.1 Meetings and Communication . . . 19
7.2 Task Division and Planning . . . 20

8 Discussion **23**
8.1 Institutional Single Sign-On . . . 23
8.2 Balancing LLM Generated and human made information . . . 23
8.3 Potential for misuse of the review system . . . 23
8.4 Sustainability implications of using LLMs . . . 23

9 Conclusion **24**

10 Acknowledgments **25**

A Figma Prototype **25**

B User testing interview questions & tasks **28**
B.1 Semi-structured Interview Questions . . . 28
B.2 Follow-up Questions . . . 28
B.3 Testing Instructions and Tasks . . . 29
B.4 Student Tasks . . . 29
B.5 Teacher Tasks . . . 29

C Usability Testing interview answers **30**
C.1 Student Interviewee 1 . . . 30
C.2 Student Interviewee 2 . . . 30
C.3 Student Interviewee 3 . . . 31
C.4 Student Interviewee 4 . . . 31
C.5 Student Interviewee 5 . . . 32
C.6 Student Interviewee 6 . . . 32
C.7 Student Interviewee 7 . . . 33
C.8 Student Interviewee 8 . . . 34
C.9 Teacher Interviewee 1 . . . 34
C.10 Teacher Interviewee 2 . . . 35

D Unit tests **35**
D.1 Course Page Functionality . . . 35
D.2 Teacher Editing Functionality . . . 37
D.3 Information Pages and Navigation Bar . . . 38
D.4 Authentication and Access Control . . . 40
D.5 Course and Module Navigation . . . 41

E UML Diagrams **43**
E.1 State Machine Diagram . . . 43
E.2 Class Diagram . . . 43
E.3 Sequence Diagrams . . . 44
E.4 Activity Diagram . . . 46
E.5 Topology & Architecture . . . 48

1 Introduction

1.1 Background and Motivation

Sustainability has become an increasingly important consideration for higher education institutions. Universities play an important role in preparing students to address societal and environmental challenges, and integrating sustainability into education is one way to contribute to this responsibility. The University of Twente acknowledges this role through its commitment to the Think Green, Act Green manifesto, which calls on knowledge institutions to embed sustainability in education, research, operations and governance [5].

An important framework for understanding sustainability is the United Nations Sustainable Development Goals (SDGs). Adopted by UN member states in 2015, the 17 goals address global challenges ranging from poverty and inequality to climate change and environmental protection [4]. However, understanding how these goals relate to individual university courses is not always straightforward. Course descriptions primarily communicate academic content and learning objectives, making it difficult for students to identify which sustainability topics a course addresses. Alongside goal-oriented frameworks like the SDGs, assessing education for sustainability requires examining the concrete skills and mindsets cultivated in students. To this end, the European Commission developed GreenComp: The European Sustainability Competence Framework [2]. GreenComp establishes a consensual reference framework that defines sustainability as a holistic competence composed of twelve specific competences across four interrelated areas: embodying sustainability values, embracing complexity in sustainability, envisioning sustainable futures, and acting for sustainability. By moving beyond declarative knowledge to encompass critical thinking, system perspectives, and active agency, GreenComp provides a standardized benchmark for curriculum developers to evaluate how educational programmes foster practical sustainability literacy, which is an objective explicitly endorsed by the Think Green, Act Green manifesto.

This project builds on a large language model (LLM) framework that maps University of Twente course descriptions from the Osiris database to relevant SDGs and sustainability competencies. While the system scales automated curriculum analysis across hundreds of courses, it faces two primary challenges: verifying that the model's determinations reflect actual teaching and student experience, and presenting the sustainability overview generated by the LLM in a concise and intuitive way. This work addresses these needs by developing an intuitive platform to visualize course-level sustainability data and evaluating how effectively it serves both students selecting courses and instructors reviewing course coverage.

1.2 Project Goal and Proposed Solution

The goal of Think Green, Study Green is to make the relationship between university education and sustainability more transparent and accessible. We address this goal by developing a web application that allows students and teaching staff to explore, evaluate and provide feedback on the sustainability-related content of university courses.

The application combines automated assessments with opportunities for human input. It analyses course descriptions using an LLM and presents the resulting information in relation to the 17 United Nations Sustainable Development Goals (SDGs) and the European GreenComp sustainability competence framework. Rather than presenting the assessments as definitive judgments, the application provides an overview of the sustainability topics identified in a course, together with explanations of the generated results.

The application provides several ways to interact with this information:

- **Course exploration:** Students can browse and search for university courses, using filters such as study programme, academic year and SDG.

- **Sustainability overview:** Individual course pages display the SDGs and GreenComp competencies identified in the course description, alongside an AI-generated explanation of the results.
- **Student feedback:** Students can submit reviews describing their experiences and indicating the extent to which particular SDGs are covered in a course.
- **Teacher contributions:** Teaching staff can review and edit the AI-generated course overview with their own assessment when necessary.
- **Sustainability information:** Dedicated pages explain the SDGs and GreenComp framework, providing context for users who are unfamiliar with these concepts.

By combining these features, the application provides students with an additional perspective when exploring courses and considering their educational choices. It also gives teaching staff an opportunity to contribute to the accuracy of the information presented. The application therefore functions as an information and feedback platform rather than a ranking system that determines whether a course is sustainable.

1.3 Report Structure

The remainder of this report is organised as follows. Section 2 presents the functional and non-functional requirements of the application, including the prioritisation of features using the MoSCoW method. The following sections describe the system architecture, design decisions and integration of the application. Section 6 presents the usability, unit, and API tests conducted to evaluate the application, followed by an overview of project management and team work. Finally, Section 8 reflects on the limitations of the developed solution and possible improvements, while Section 9 summarises the project outcomes and the extent to which the project goal was achieved.

2 Requirements

In this section, the functional and non-functional requirements of the system are identified based on discussions with our client. To establish their relative importance, the requirements were prioritised using the MoSCoW method, categorising them according to their level of priority.

2.1 Functional Requirements

Table 1 presents the identified functional requirements and their respective priorities. No requirements were assigned to the “Will not have” category, as all identified requirements were considered relevant to the current project scope.

Table 1: MoSCoW prioritisation of functional requirements

Priority	**Functional Requirements**
Must Have	1. The system must display a sustainability score for all CIS courses. 2. The system must provide a web interface. 3. The system must score courses based on the UN Sustainable Development Goals and GreenComp. 4. The system shall use a database to store all data. 5. The system must allow logged-in professors to edit generated material/course descriptions. 6. The system must store changes made by logged-in professors in the database. 7. The system must allow users to submit their own experiences and feedback for the LLM to take into account. 8. The system must only collect text used for research purposes, teacher feedback, and anonymised LLM summaries.
Should Have	1. The system should display a user score separate from the LLM score. 2. The system should use multi-user average user scores to mitigate malicious feedback.
Could Have	1. The system could require evidence (e.g., pictures of slides or details) to discourage griefing. 2. The system could require evidence (e.g., pictures of slides or details) for negative feedback to discourage griefing. 3. The system could require a unique login before feedback can be submitted. 4. The system could include a chatbot that allows students to talk about courses (e.g., for recommendations). 5. The system could increase LLM accuracy by combining outputs from multiple models. 6. The system could generate a sustainability certificate for each course. 7. The system could have a personalized user dashboard that shows their overall sustainability score.

2.2 Non-Functional Requirements

The non-functional requirements are presented in Table 2. Note that no requirements were assigned to the "Could Have" category, as the identified non-functional requirements were considered sufficiently covered by the Must Have and Should Have priorities. Similarly, no requirements were assigned to the "Will not have" category, as no identified non-functional requirements were considered outside the scope of the current project.

Table 2: MoSCoW prioritisation of non-functional requirements

Priority	Non-Functional Requirements
Must Have	1. The system must create awareness among students. 2. The system shall clearly state that the UN does not endorse the project nor raise funds if UN icons or assets are used. 3. The system must be protected against prompt injection. 4. The system must store and encrypt user data in a protected database.
Should Have	1. The system should fetch the course scores and display them within 2 seconds. 2. The system should be modular and easily expandable. 3. The system should be intuitive and easy to use for both students and teachers.

3 Architecture

3.1 UML Diagrams

In complex projects developed by multiple team members, visualizing the internal workings of the system pipelines is essential for clear communication. Therefore, a series of UML diagrams was constructed. To better illustrate the backend architecture discussed in the following subsection, the most relevant UML diagrams are interleaved throughout the text, while supplementary architectural and behavioral diagrams are provided in Appendix E.

3.2 Backend

The backend of our system was designed around a small, pre-made proof-of-concept script received from our client and supervisor. Originally, it only supported scraping Osiris for courses, a rudimentary 500-character filter for courses without sufficient information or depth to be rated by the LLM, and a CSV output of sustainability ratings provided by a UT server running a local Qwen model. Throughout the project, we expanded this starting point into a feature-rich, modular, multi-tier data and evaluation platform. The Osiris scraper (`get_osiris.py`) has remained largely unchanged, with several command-line quality-of-life options added.

```
# Scrape UTwente courses for the 2026-2027 academic year with 15 concurrent
    workers
python get_osiris.py --uni utwente --year 2026 --workers 15
```

```
# Scrape Utrecht University (UU) across a specific ID range
python get_osiris.py --uni uu --year 2025 --start 240000 --end 250000 --workers
    15
```

After the Osiris scraper finishes, it outputs a large volume of text files containing the raw JSON of every course registered in Osiris within the specified parameters, which is difficult to work with directly in the other parts of our system. Additionally, many courses are registered in Osiris solely for administrative purposes—such as entries with no descriptions, dummy module courses with no real curriculum, internships, and bachelor theses—which would neither make sense to list on the website nor be feasible for the LLM to evaluate for sustainability competencies.

To increase the quality of the dataset and ensure minimal wasted compute server time, we introduced a filtering script (`filter_utils.py`) that catches as many of these edge cases as possible and labels them accordingly (Figure 1).

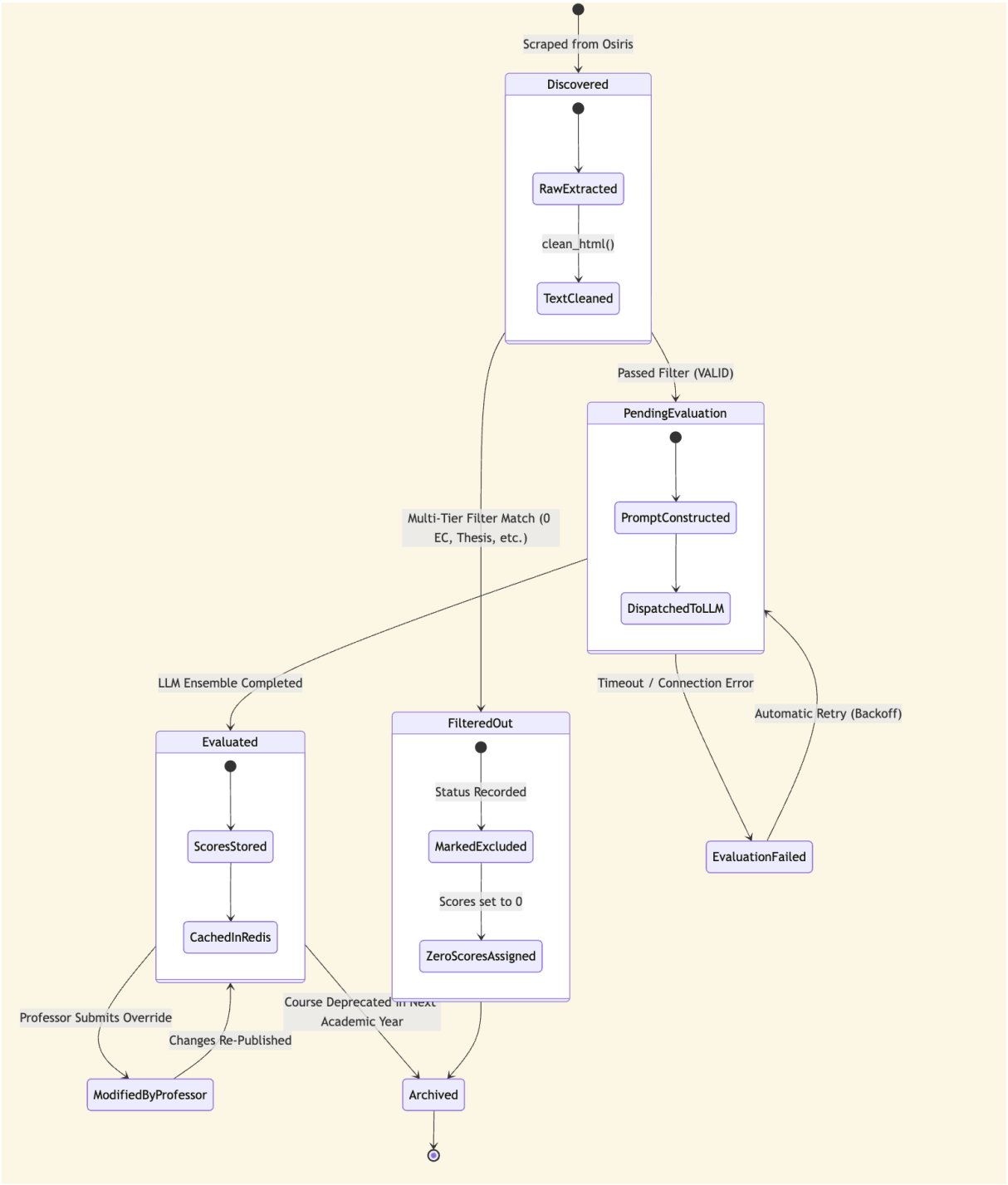


Figure 1: Course Evaluation Lifecycle

After the filtering stage, `hierarchy_parser.py` is used to infer structural information not directly provided by Osiris, such as grouping courses by their module, linking recurring year-over-year courses under the same `course_series`, and identifying module containers for generating our module pages. This script iterates over all scraped files deterministically using a three-step classification process. First, if a course retains its course code across years, the two course identifiers are linked under the same course series. Because it is customary at the University of Twente to change course codes across different editions of the same course, the remaining way to identify relationships between courses is by name. We normalize the course names after

stripping common formatting variations (e.g., `Part 1`, `(1)`, `[1]` → `1`). If the normalized name already exists in the database, the course is linked to that series; otherwise, it becomes a new "slug" for subsequent courses to link with.

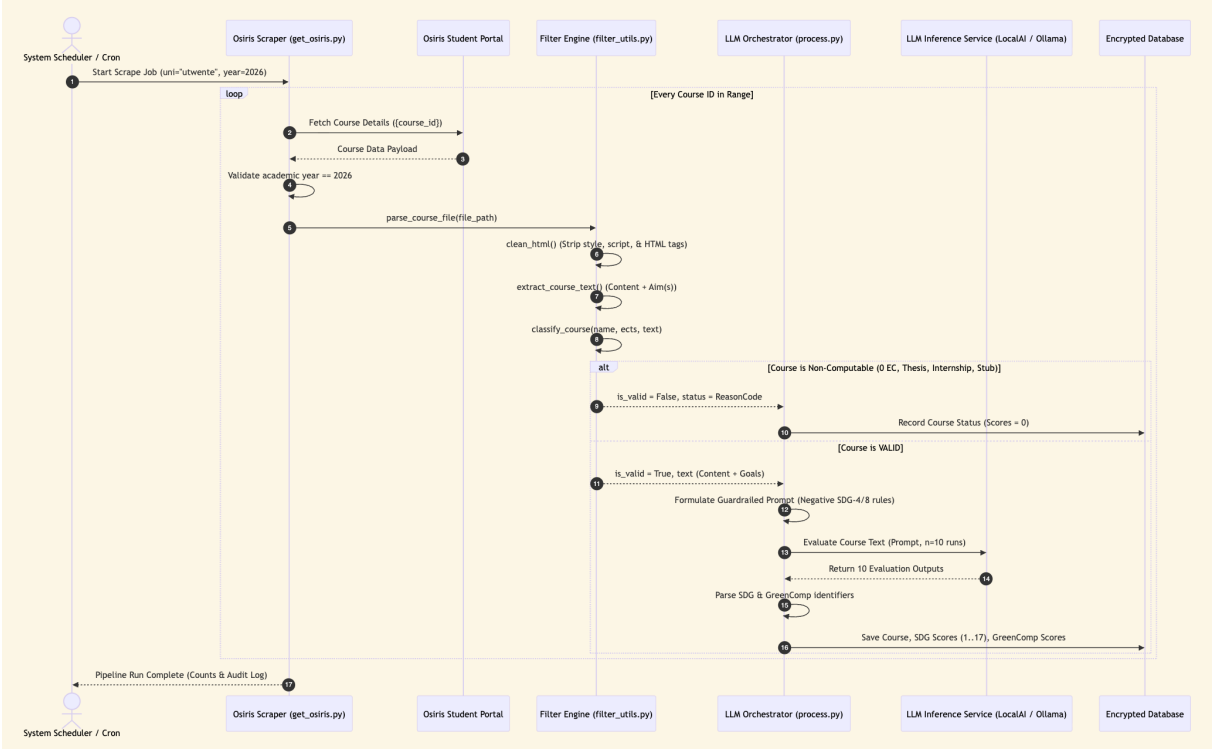


Figure 2: Pipeline: Scraping, Multi-Tier Filter & LLM Grading

Within the same script, we inspect the courses that were flagged by `filter_utils.py` as module containers or 0 EC courses to search for mentions of other course codes. If a module page references specific courses, or if courses reference their parent module, they are linked under that module. Finally, `evaluate_and_sync.py` uploads the categorized and filtered courses to our Supabase database and uses an LLM to infer from the Osiris text which SDGs and GreenComp competencies are covered (Figure 2). The script skips previously evaluated courses and pushes results to the database incrementally as it runs—ensuring the process can be safely interrupted and resumed at any time—before generating a brief summary of the overall sustainability coverage for each course, module, or programme.

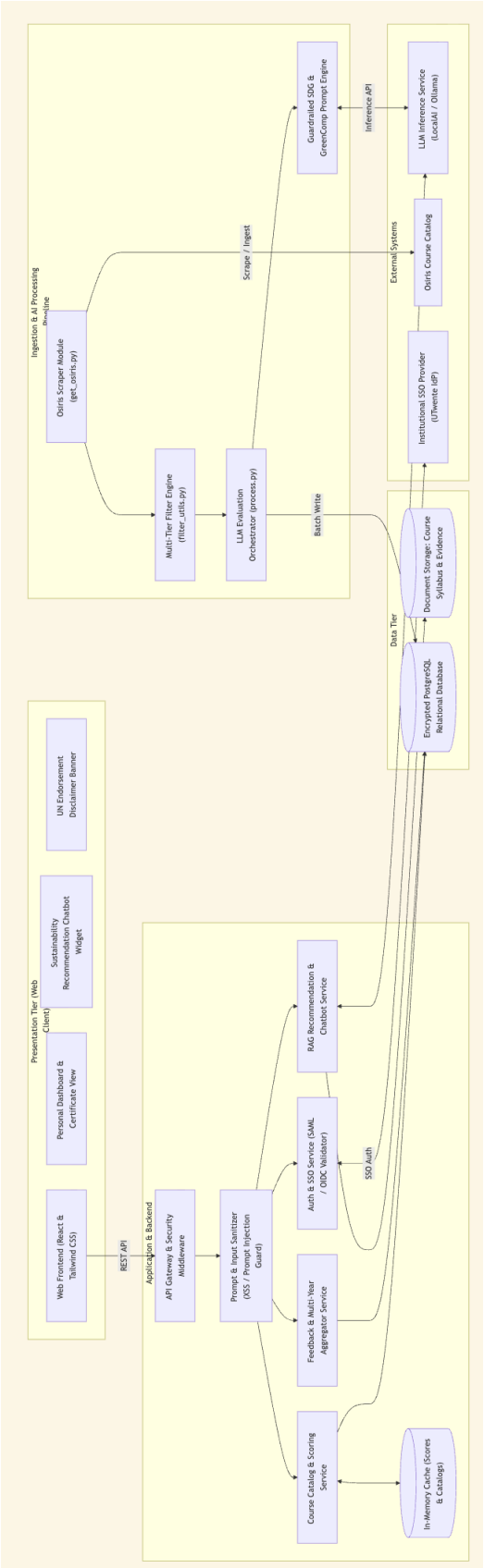


Figure 3: Component Architecture

On the application side (Figure 3), our web platform uses `fetch.ts` to query data from the Supabase database (Figure 4) and return strongly typed TypeScript objects that mirror

the database tables. These objects are then passed to the various front-end components to be displayed on the website. Supplementary architectural views—including the initial domain model, target user-flow sequence diagrams, filtering activity logic, and deployment topology—are documented with detailed descriptions in Appendix E.

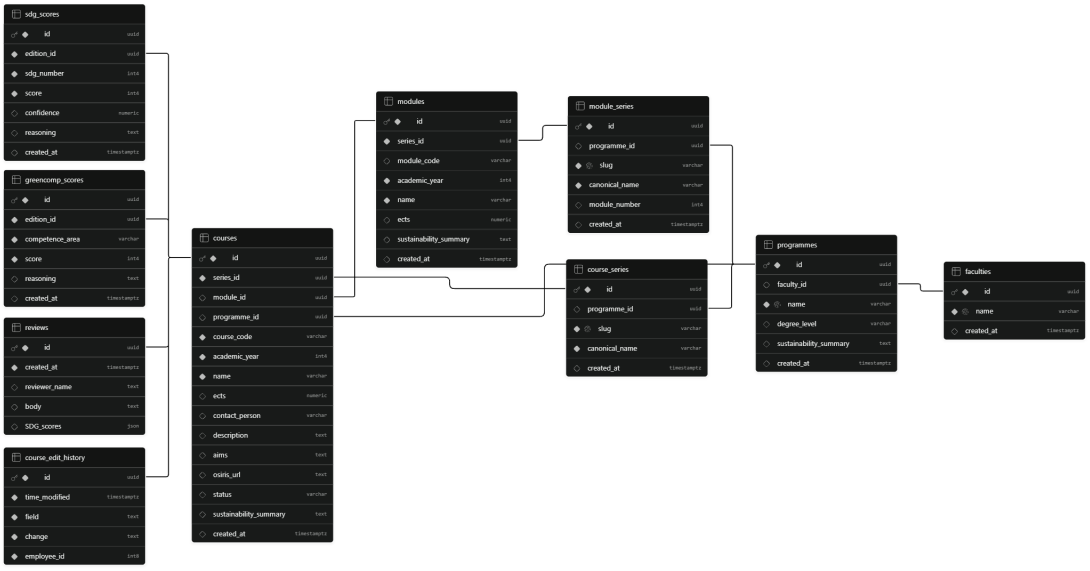


Figure 4: Database schema

4 Design

4.1 Framework

The core functionality of the web platform centers on four primary capabilities: displaying course-level sustainability scores, enabling student reviews, allowing instructors to manage course pages, and providing role-based authentication to restrict permissions to authorized users. To implement these features while acquiring experience with modern web technologies, we selected Next.js as the primary full-stack framework. This choice was also collectively favoured since the majority of our team members did not have prior development experience with this framework. A secondary goal of this project, as a team, was to take on a new technical challenge and broaden our skill set as Computer Science students. Several technical advantages also motivated this choice:

- **Server-Side Rendering:** Many of our pages (course page, module page) are relatively content-heavy and do not require much client-side interactivity; server rendering allows the server to prepare the page's HTML before sending it to the browser, helping users see relevant content sooner.
- **File-based routing:** The file and folder structure directly determines the page URLs.
- **Streamlined developer experience:** Features such as hot reloading allow code changes to update pages in real time, improving the development workflow.
- **Scalability and modularity:** Component-based architecture simplifies long-term maintenance and feature expansion.

- **UI component ecosystem:** Compatibility with a wide range of React UI component libraries.

4.2 Initial Prototype

The design process for the product started with a dummy prototype in Figma. All of the final pages on our website, such as the main page and the courses page, are derived from an initial prototype. All prototype pages and initial designs can be found in Appendix A. This prototype was showcased to the client before moving forward to the implementation process. After discussion with the client, the following pages were agreed upon:

- **Main Page:** The main landing page of the website which gives a bit of information on the goal of the website; it will be the first thing the users see.
- **Courses Page:** A page where you can find all courses with the ability to filter on academic year, faculty, programme and SDGs if necessary, along with a search bar.
- **Course Page:** Where users can find more in-depth information on a course. Users are able to see a breakdown of the SDGs, general course information such as course code and academic year, and if it belongs to a module or not. It also displays an AI summary of the course in regards to the SDGs.
- **Module Page:** Similar to the Course page, but for modules and with a few adjustments.
- **SDG and GreenComp explainer page:** A page that gives a description of the SDGs and GreenComp.
- **Teacher Editing Page:** A page that allows the teacher to edit the summary produced by the AI if they feel the need to.
- **Student Review Page:** A page where students can leave a review on the course.

4.3 Key Design Choices & Motivation

Presenting rich sustainability data such as SDG classifications, GreenComp competences, course descriptions, and student evaluations, without overwhelming the user, presents a core front-end design challenge. Rather than exposing all raw outputs simultaneously, the interface underwent several iterative refinements in how data is structured, scored, and visualized.

The representation of SDG coverage evolved directly alongside our understanding of the LLM's behavior. In the initial prototype, we implemented a 0–10 scoring scale generated by the LLM that was visualized via a horizontal bar chart to illustrate varying intensities of coverage across individual SDGs. However, evaluating the model's outputs revealed that continuous scores lacked consistency and explainability because the LLM tended to exaggerate results into extreme values, clustering around 0 or 10, and struggled to reliably justify incremental differences based on the Osiris data. Consequently, we refined the scoring logic into a binary classification that simply identifies whether an SDG is explicitly covered or not covered. This shift rendered the extra bar chart redundant and introduced unnecessary visual clutter. This observation is further validated through moderated usability testing that was conducted after the first initial change. We therefore transitioned to SDG emphasis only through a pie chart (Fig. 6), which displays the proportional distribution of covered goals at a glance while retaining official UN SDG color encoding for quick recognition.

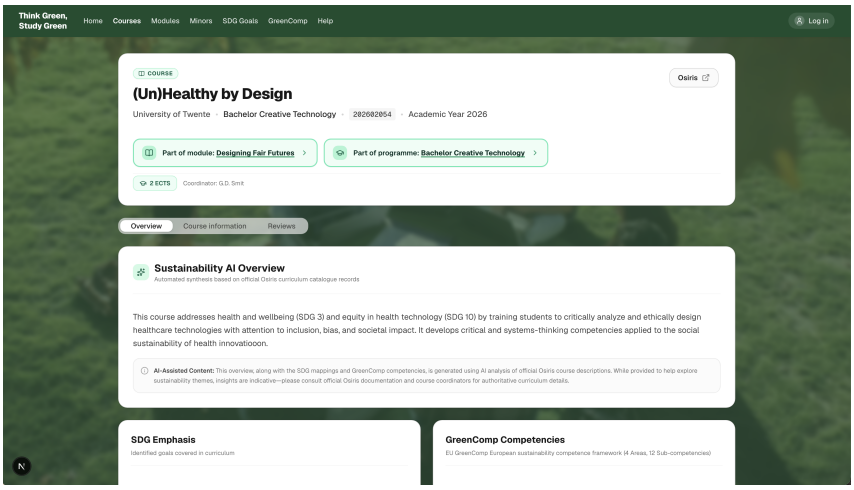


Figure 5: Main header of a course page

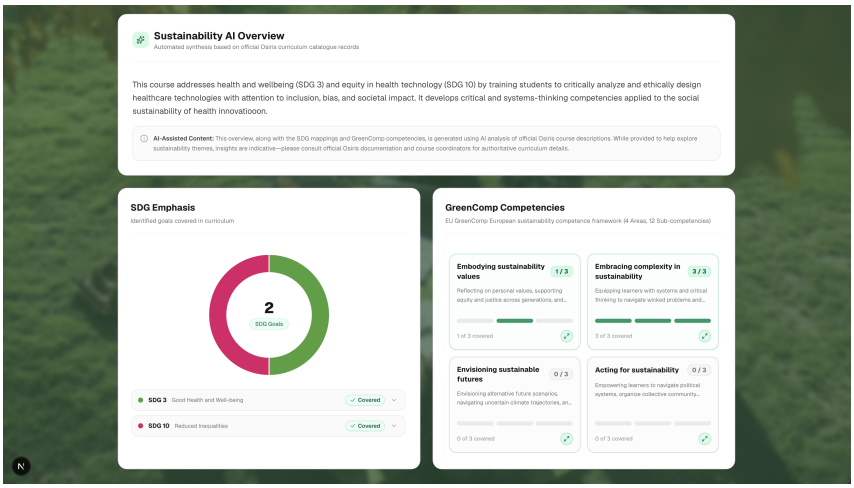


Figure 6: Sustainability graphs on a course page

To manage the remaining textual clutter without inducing cognitive overload, the interface attempts to minimize the information displayed throughout the course page layout. Course metadata, LLM reasoning summaries and GreenComp competency breakdowns are organized into dedicated modular views accessible via a carousel component (Fig. 5), allowing users to focus on one block of information at a time. Within this structure, GreenComp criteria are grouped into four high-level domain panels that display summarized coverage indicators, expanding only when the user clicks on one of them. Supplementary details across all views are similarly bound to hover and click interactions, preserving a clean visual hierarchy while maintaining access to data. This interaction model extends to the review section as well, where student feedback is organized into individual dedicated cards for each evaluated SDG (Fig. 7). Rather than displaying all textual feedback and ratings in a single line of text, users can select an individual SDG card and click on it to reveal the reviewer's 1–5 Likert scale rating.

The design of this review format was partly inspired by related work evaluating university course alignment with sustainability goals [1]. Although that study demonstrated notable methodological and practical limitations, it provided an initial precedent for capturing goal-specific feedback within an academic setting. Moderated usability testing of our prototype confirmed that students found the 1–5 Likert scale intuitive, justifying the design for the current implementation. Nevertheless, further empirical evaluation and targeted user testing remain

necessary to thoroughly validate this design choice and determine whether a more nuanced feedback structure is needed.



Figure 7: Review section of a course



Figure 8: SDG page

Lastly, a significant but not so obvious design choice that is present throughout our entire web application is the SDG color coordination. As mentioned in an earlier paragraph, we kept the pre-assigned colors corresponding to each SDG, as shown in Fig. 8, so that users would be able to recognize a goal at first glance upon interacting with a feature that involves SDGs.

4.4 Final Product Design and Description

The following section lists all components of the final implementation of our product.

4.4.1 Log-In system

To ensure a functioning and secure authentication despite this hurdle, we simply used the OAuth authentication using Google Sign-In managed through Supabase. Since this solution does not inherently distinguish between university staff and students, we implemented a role-selection step after logging in. This would be replaced by the SSO log-in in future development.

4.4.2 Main page

The main page is the first that users will see when visiting the website. Thus, its primary goals are familiarizing the user with the website's functionality and spreading awareness about sustainability. The page contains brief explanations for: where the idea for the website originated from, and the purpose of the website. The page links to other relevant pages depending on the user's needs.

4.4.3 Course page

A course page enables users to view the AI-generated overview containing SDG scoring and GreenComp evaluation. Student users must be able to leave a review. In parallel, teachers who are responsible for a course must be able to edit the information on the corresponding course page.

Course Overview and Information sections:

- **LLM Reasoning:** To enhance the usability of the website and satisfy the requirement to justify the identified sustainability coverage, we provide an LLM-generated justification for each covered SDG or GreenComp sub-goal. Additionally, an LLM-generated overview provides a brief summary of a course's sustainability coverage. This naturally comes with an AI-generated content warning to ensure that users are informed how this information is obtained and that there is the possibility that it could be incomplete, un-nuanced or inaccurate information.
- **SDG:** Our final system presents SDG coverage with a color-coded pie chart and a number representing how many SDGs are covered by the course. An LLM justification is available for each SDG which explains how or why the course covers the topic.
- **GreenComp competencies:** GreenComp competencies are represented via four clickable panels for each GreenComp competency. To avoid information overload, each panel contains only a brief description of the competency and which sub-goals are covered. More detailed sub-goal descriptions and an AI justification for coverage are revealed when clicked.
- **Course Information:** To emphasise the usability of the course page, we decided it would be inconvenient for users to navigate to the course on Osiris for its description and learning goals. However, displaying the information directly in the course overview would overwhelm the user with too much information. Thus, a carousel component is used to display this information in its own section on the course page (the user can switch to another section via a menu to change the content of the page).

Review Section: A major requirement for the system is allowing student users to leave their own feedback and experiences for the LLM to take into account in the form of reviews. The idea was further iterated on via peer reviews, group discussions and supervisor meetings where it was ultimately decided that the reviews are most valuable if presented separately instead of being used as further input for the LLM-generated overviews and scores. This allows the user to browse the opinions and experiences of students regarding the sustainability of the course and give them a broader overview of which topics the course may cover and thus form a more holistic opinion.

Furthermore, the aim of the LLM-generated overviews is to only identify which sustainability goals are covered and summarize this information based on the course description; however, there is always risk of hallucination or inaccurate information. Adding input from reviews, which also have no guarantee of being accurate or helpful, may not necessarily aid in gaining more correct LLM-generated information. Additionally, the system allows teachers to edit the LLM overview

if they deem it inaccurate or want to add more information. This gives teachers the opportunity to adjust the generated information based on their knowledge of the course.

- **Review system:** Logged-in student users can leave a textual review and rating for each SDG they believe is covered in a course. These reviews are displayed in the review section for others to read. The user can also suggest new SDGs that they believe are also covered but are not currently displayed on the course overview.
- **Likert scale:** It was decided that a 1–5 Likert scale is best to rate how much an SDG topic is covered within a course: 1 – Not Covered (The course does not address this area), 2 – Partially Covered (The course addresses this area to a limited extent), 3 – Moderately covered (The course addresses this area to a noticeable extent), 4 – Considerably covered (The course addresses this area in a substantial way), 5 – Extensively covered (The course addresses this area in depth).
- **Average review score:** To give users a breakdown of the opinions shared in the review section, the average score for each SDG is presented via a bar chart.

4.4.4 Courses page

Considering that there are a large amount of courses in the UT and the potential for expanding the system to include courses from other universities in the future, in order for users to be able to see the sustainability scores, leave reviews or edit the information for a specific course, an intuitive method for finding and accessing the courses is required. This is the primary function of the courses page.

- **Search Bar and Filters:** To accommodate for the large amount of courses in the database, a search bar and filters were deemed useful features. The filter options were chosen based on what was deemed most useful to students, prospective students and professors. The Study programme filter allows users to find courses within a programme or multiple programmes that they are a part of or are interested in. The SDG filter allows users to browse courses that cover SDGs they are specifically interested in. The Academic year filter allows users to look for courses for the current year or previous years if they would like to see how a course has changed. The search bar searches courses by name and the search results are queried as the user types. To avoid a database query occurring on each keystroke, debouncing is used to limit the number of times a search query can occur within a short interval.
- **Course Cards:** The course card element is what makes up the list of courses. The element needs to contain the following information: Course Name, University, Programme, Course code and Academic year. To give a brief overview of which SDGs are covered by a specific course, a small pie chart on the right is present, the color of each section on the chart corresponding to the color of an SDG.
- **Course List and Pagination:** Displaying all courses on a single page requires the user to scroll for an unreasonable amount of time; to remedy this, the course list is split up into pages of 20 courses.

4.4.5 Teacher editing page

The teacher editing page is accessible via the course page only by logged-in teachers. Since the LLM summary may not always be in line with what a teacher thinks is accurate, a logged-in teacher is able to edit the LLM summary for a course and replace it with their own summary. Initially it was decided that teachers should also be able to edit course descriptions and goals, but since this information is scraped from Osiris, which teachers also have access to and can

edit the information in, such a feature was deemed redundant. The final system allows those with the teacher role to edit the AI summary and replace it with their own explanation. These teacher-made overviews are stored in the database and displayed on the course overview page.

4.4.6 Help page

Throughout development and testing, it was found that many aspects of our website that at first appeared obvious were unclear to new users. Thus, it was deemed necessary for the system to have a help page which answered frequently asked questions to improve usability.

- **FAQ:** The FAQ section contains important information about the LLM Overviews including: how they are used to generate a sustainability overview, why they are used and how students are able to give feedback.
- **Green Energy Guard Widget:** A common concern is the energy cost of the web application due to its use of an LLM. To address these concerns, LLM queries are scheduled to occur only during periods of lower carbon intensity for the electricity grid. This widget provides further information on how this works. Further discussion of this topic is contained in Section 8.

4.4.7 SDG and GreenComp pages

To ensure that users are able to understand the frameworks against which courses are reviewed, the system has an SDG Goal and GreenComp page which provides information on what they are and a link to courses which provide further information. The 17 SDG Goals are color coded and users can hover over each of them to view more information on that specific goal. The GreenComp competencies are separated into 4 components, with each having 3 sub-goals that the user can hover over to read a more detailed explanation on the sub-goal.

5 Integration

5.1 API

We integrated the Supabase database into our application using the `supabase.js` library, which allows us to communicate with the database through a TypeScript interface. To organize database interactions, we created a `fetch.ts` file that defines TypeScript interfaces for the relevant data structures. These interfaces describe the expected structure of the retrieved data, allowing us to use consistent data types throughout the application. Additionally, we implemented dedicated functions in `fetch.ts` to retrieve specific information from the database. Each function constructs a query using the Supabase client and returns the requested data. By centralizing these queries, we separate database interactions from the components that display the data. This approach also makes queries easier to maintain and reuse across the application.

We have also created API routes to support data retrieval in client-side components. Next.js distinguishes between client-side and server-side components. Server components execute on the server and client-side components execute in the browser of the user so are used to provide interactive functionality in the user's browser. Although client-side components are capable of retrieving data directly from the server, API routes allow us to separate database interactions from client-side functionality so that we can manage data retrieval centrally. Furthermore, API routes can be extended with additional functionality such as authentication and authorization checks without big changes to client-side components.

6 Testing

6.1 Moderated Usability Testing

6.1.1 Environment Setup

Each usability testing session was conducted in a private room with one participant and two project supervisors. The participant was provided with a laptop. Each participant started on the main page of the web application. One project supervisor guided the session by providing instructions, while the second project supervisor took notes on the participant's actions, observations, and feedback.

6.1.2 Session Structure

After setting up, participants were asked to perform a series of tasks based on their role as a student or a teacher. The list of tasks can be found in Appendix B. The participants received minimal instructions and were not provided with assistance on how to complete the tasks, allowing us to evaluate the usability and intuitiveness of the application. Participants were instructed to think out loud while interacting with the application and performing each task. This allowed us to gain insight into their thought processes, expectations, and potential difficulties during navigation. Each session concluded with a semi-structured interview in which participants answered open-ended questions, found in Appendix B.

6.1.3 Results

We attempted to compile a pool of participants that were representative of different faculties and target groups within our university. Our resulting testing pool is rather skewed, as teacher participants were difficult to acquire due to their time constraints and busy schedules. Regardless of this drawback, we managed to acquire a total of 10 participants consisting of 8 students and 2 teachers. Each testing session proved to be particularly eye-opening, as features and interactions which at first seemed intuitive proved rather the opposite, which created a very interesting phenomenon. The critical changes that were derived from this feedback are discussed in earlier chapters.

6.2 Unit Testing

The unit testing section covers all features implemented in the web application and aims at demonstrating the successful implementation of baseline requirements. Through unit testing we were able to verify that the application meets the criteria for a functional Minimum Viable Product (MVP). A complete list can be found in Appendix D.

6.2.1 Results

All unit test cases covering course overview rendering, review submission, teacher editing, information pages, authentication, and course/module navigation passed successfully, as documented in Appendix D.

6.3 API Testing

To verify our backend integration, API testing was conducted on the core Supabase schema with multiple SQL queries to confirm that data is accurately fetched, inserted, and updated. An additional focus throughout API testing was also verifying the integrity and correctness of our database, such as whether our schema follows a logical structure. Finally, these tests ensure data

cleanliness, which is a critical factor when managing large datasets. Table 3 contains an overview of the SQL test files and query results.

6.3.1 Results

While most of the queries initially passed, some significant details revealed a few inconsistencies in our database schema. For example, some tables had overlapping primary and foreign keys, coupled with inconsistent naming conventions which generated confusion between table variables. Additionally, given that the original requirement asked for a dummy implementation of a user log in, API testing highlighted the importance of access policies when using API POST and UPDATE. Thus, as a result, it was discussed and agreed with the client to ultimately implement a simple admin log in and create a proper log in which is actually connected to the database. We can conclude that this decision will further benefit our application in future development and embraces the system in a modular way. Since then we have implemented a fully working alternative log in system using Supabase while allowing the possibility of university-level SSO support.

Test File Name	**Purpose**	**Status**
`variable_types_test`	Verify columns strictly accept data matching their defined types (e.g., integer, varchar, boolean).	Pass
`policies_test`	Validate Row Level Security (RLS) and access policies for leaving a review or modifying the AI course summary.	Pass
`default_test`	Check that columns correctly populate with default values when no input is provided.	Pass
`not_null_test`	Ensure `NOT NULL` constraints correctly reject empty/null data in required fields.	Pass
`FK_test`	Verify Foreign Key (FK) constraints maintain referential integrity between tables.	Pass
`unique_test`	Confirm `UNIQUE` constraints prevent duplicate data entries in specific columns.	Pass
`boundaries_test`	Test that there is no invalid data outside of set value boundaries, specifically for SDG and GreenComp scoring.	Pass
`PK_test`	Validate Primary Key (PK) constraints enforce uniqueness and non-nullability.	Pass

Table 3: Summary of API and Database Tests

7 Team Work

7.1 Meetings and Communication

Regular communication helped our team coordinate tasks and maintain a shared understanding of the project's progress and scope. We held meetings almost every working day, usually in person, and arranged online meetings when necessary. During these meetings, we discuss ongoing tasks, share progress, identified problems and coordinated work between team members. We also held weekly meetings with our client and supervisor. These meetings allowed us to present our

progress and discuss questions regarding the application requirements whilst receiving feedback on design and implementation decisions. We used this feedback to guide subsequent development. Several tools were used to support collaboration throughout the project. GitLab provided version control and allowed us to manage changes to the codebase while working on different parts of the application.

7.2 Task Division and Planning

We divide the work according to the main components of the application and the tasks required for their development. This approach allowed team members to focus on specific areas while maintaining communication about components that depend on each other. Our division of responsibilities remained flexible throughout the project. When a task required collaboration between multiple team members, we coordinated our efforts to complete it. We also redistributed tasks when necessary to account for the remaining workload and technical requirements. Regular meetings helped us identify tasks that require additional attention and monitor their progress. We also use the Git repository to track development activity and monitor contributions. This activity was recorded throughout the project using GitLab issues, as well as the commit history of the project's repository and can be visualized through the repository GitLab link. Table 4 provides an overview of the main responsibilities of each team member.

Team Member	**Main Responsibilities**
Andreea Şerban	Supabase schema, Front-end implementation of teacher page, review page and log-in page, Back-end implementation of UPDATE AI summary, POST review and user authentication, Moderate Usability Testing, Unit Testing, API Testing.
Tudor Sandu	Non-webapp and webapp back-end, data handling (Osiris scraper, data filtering, LLM prompting, database implementation, population and integration, abstracted page for courses, NED and emissions tracking widget, modules and programmes (CurriculumDetailPage), security auditing (sql injection, prompt injection etc), miscellaneous help with login and course search, bugs.
Joel Poerba	Front-end implementation of courses page (search bar, pagination), GreenComp page, Course page (course information section, average review score bar chart) Back end changes for courses page, Moderated Usability testing, report.
Fabiana Žukaitė	Main page, SDG page, help page, Figma prototype, usability testing.
Amir Martijn	Front-end implementation of courses page, filters and some back-end adjustments for the courses page, report , User Testing.

Table 4: Overview of team responsibilities

A visual timeline of our project is illustrated using a Gantt Chart (Fig. 9). The chart reflects an iterative development life cycle structured across the project duration. Key milestones, including the initial project proposal approval, architecture definition, prototype validation, and final deliverables, were established early to set measurable checkpoints. Development tasks were scheduled with parallel work streams, such as the foundational infrastructure, meaning database modeling and API integration, ran concurrently with early front-end prototyping to prevent

bottlenecks. Subsequent phases dedicated focused windows to user testing, interface refinement, and debugging prior to deployment. Throughout the 10-week life cycle, our team adhered strictly to the planned timeline, meeting each milestone. This consistent pacing ensured enough time for documentation, report writing, and final presentation preparation in the last week of the project.

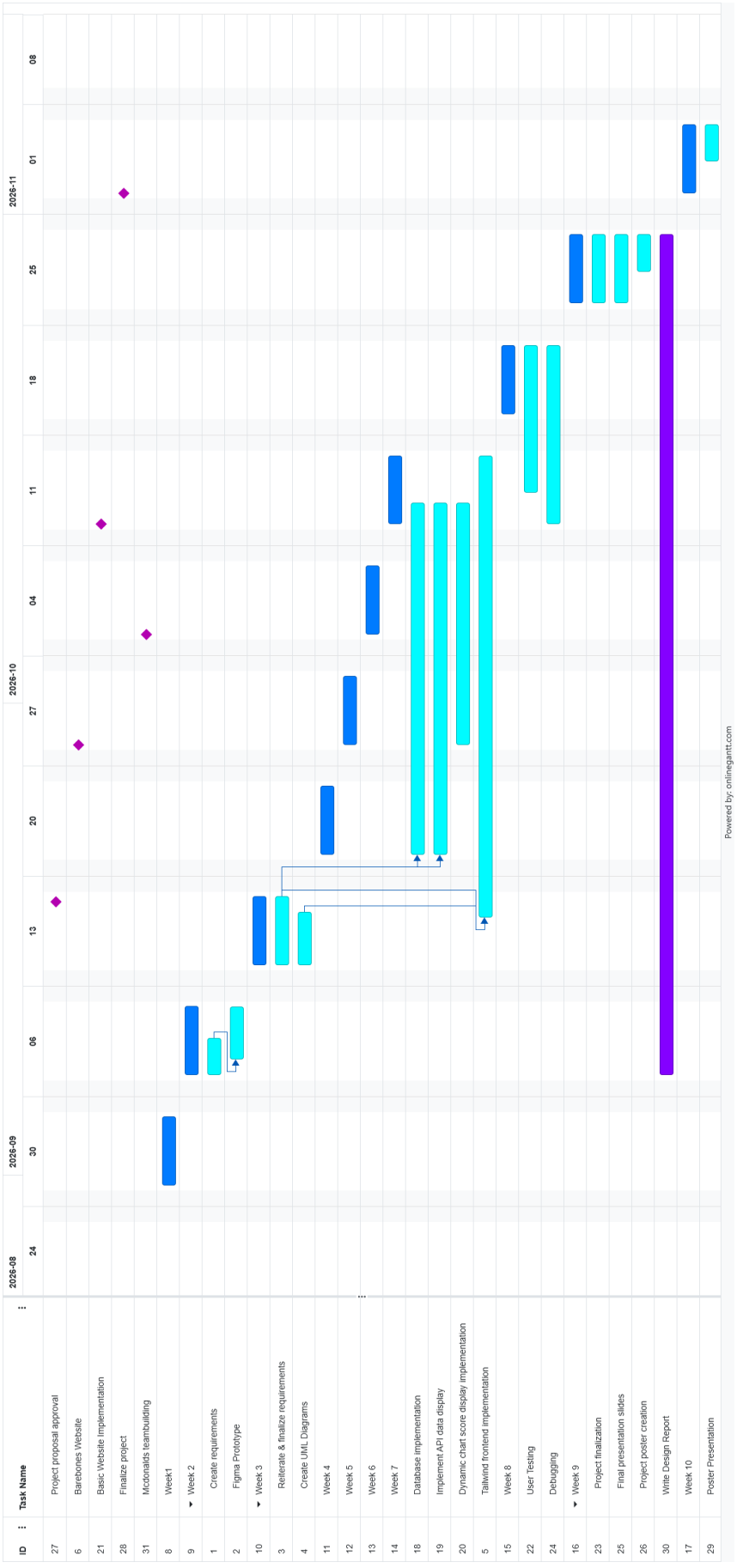


Figure 9: Timeline of Think Green, Study Green development

8 Discussion

8.1 Institutional Single Sign-On

Our initial intention was to implement institutional Single Sign-On (SSO) via the university's Microsoft log-in. This would have made it easier to verify role assignment based on email addresses. However, due to unresolved communication with the department of IT regarding authorization and API permissions within the project time frame, this integration could not be completed. This was also discussed with the client/supervisor and was ultimately deemed as out of scope.

8.2 Balancing LLM Generated and human made information

A critical design challenge in content governance concerns the balance between AI assessments and human edits. Because the platform allows teachers to modify their course information, questions arise regarding how to maintain transparency when human input alters or overrides AI generated data. Introducing a verification badge (e.g., distinguishing between an unverified sustainability overview and a teacher-validated sustainability overview) provides an initial safeguard for users browsing course pages. However, determining how to handle this post-editing presents a trade-off. Retaining the original LLM reasoning alongside instructor amendments may clutter the interface or cause confusion if they diverge. A more nuanced solution to explore in future iterations is a small visual flag or indicator, where the AI overview is flagged as "partially verified" and allowing the teacher to fully deprecate it if they choose to replace it with their own verified sustainability criteria.

8.3 Potential for misuse of the review system

During our presentation, a critical concern was raised regarding the misuse of the review system, specifically the risk of students submitting personal or defamatory remarks directed at teachers. Because our platform currently permits anonymous submissions without a proper moderation system, it leaves teachers vulnerable to ill intent. While the initial goal of student anonymity was to encourage unbiased assessments of courses, the absence of moderation introduces potentially legal and ethical risks. In future iterations, this vulnerability should be mitigated through a moderation strategy. Potential solutions include implementing automated content moderation APIs (e.g., using natural language processing to flag personal pronouns, profanity, or toxic language prior to posting), introducing a community reporting mechanism, or adopting "pseudonymous" submissions where identity is verified through the university email of the user. While the last choice might seem more validating, we have to take into account user privacy. This approach directly conflicts with a primary design incentive of prioritizing minimal to no data collection and user privacy. Storing traceable identity information introduces significant privacy invasion and increases regulatory obligations under GDPR. As a team we collectively agree that user data should only be collected at utmost necessity, and even in these situations we believe it is of ethical integrity to consider alternative options that would require the least amount of data collection. With this principle in mind, we recommend that future developers that implement our platform strongly take this aspect into account.

8.4 Sustainability implications of using LLMs

Lastly, we have to address the elephant in the room. Although our system aims to assess the sustainability of university courses, we acknowledge that using an LLM to perform these assessments also has an environmental impact. The main motivator for using an LLM is because it reduces the initial amount of manual effort required to evaluate each course. This process can become unfeasible given a large dataset containing thousands of courses. Nevertheless, our

team considered ways to minimize the environmental impact of this approach. One possible improvement is to schedule LLM requests during periods when the electricity grid has a lower carbon intensity. An example of this implementation could be a Green Energy Guard widget that provides information on the carbon intensity of electricity, which could help identify more suitable times to run computational tasks (Fig. 10). It synchronizes with the ned.nl (Nationaal Energie Dashboard), an official Dutch energy data platform [3]. A script calculates a threshold based on which it automatically pauses LLM background jobs when the electricity mix relies heavily on fossil fuels, and resumes when renewable share is high. This widget was only developed at the end of the project during spare time as it was suggested earlier by the client but could be considered as part of the main sustainability overview in the future. Additionally, course information does not need to be retrieved and reassessed daily. Instead, we could schedule updates every ten weeks, reducing the frequency of LLM requests and avoiding unnecessary computation. These measures could help reduce the environmental impact of maintaining the system.

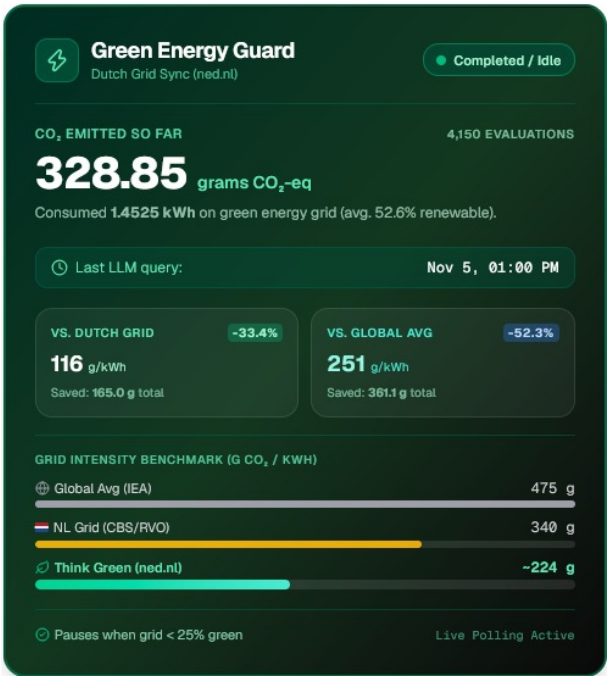


Figure 10: Green Energy Guard widget

9 Conclusion

Think Green, Study Green addresses the challenge of making sustainability-related information about university courses more accessible and understandable to students and teaching staff. The project translates AI-generated assessments of course descriptions into an interactive web application that allows users to explore how courses relate to the United Nations Sustainable Development Goals and the GreenComp framework.

The developed application brings together several elements that support this goal. Users can search and filter courses, explore their sustainability assessments, and access explanations of the SDGs and GreenComp competencies. The application also gives students the opportunity to share their perspectives through course reviews and allows teaching staff to review and edit AI-generated course overviews. By combining these features, the application presents sustainability-related information in a way that users can explore and interpret, rather than treating AI-generated assessments as definitive judgments about a course.

Overall, the project delivers a functional web application that supports the exploration of sustainability in university education. It connects course information with established sustain-

ability frameworks and provides users with ways to engage with and reflect on the presented assessments. In this way, Think Green, Study Green! contributes to making the relationship between education and sustainability more visible and gives students and teaching staff a shared starting point for considering this relationship.

10 Acknowledgments

We would like to thank our client and supervisor, Gerwin Hoogsteen, for his guidance, feedback and support throughout the project. His input helped us refine our ideas and make informed decisions during the design and development process. We would also like to thank the Green Young Academy (GrYA), for providing us with the opportunity to work on this project and for sharing their knowledge and perspective on sustainability in higher education. Finally, we would like to thank all students and teaching staff who participated in our usability testing. Their time, feedback and suggestions contributed to evaluating and improving the final application.

Use of Generative AI: Generative AI tools were used solely to proofread the document and assist with minor grammatical and formatting refinements. All system architecture, software implementation, testing, analysis, and core written content are the original work of the project team.

A Figma Prototype

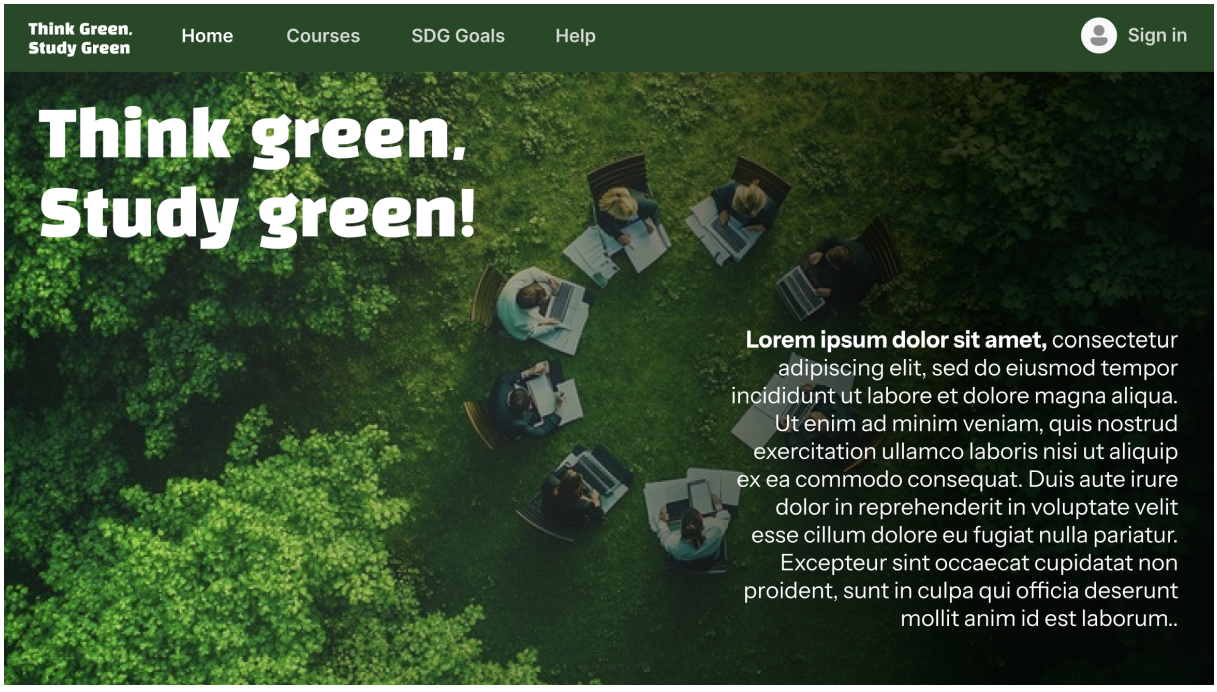


Figure 11: Main page

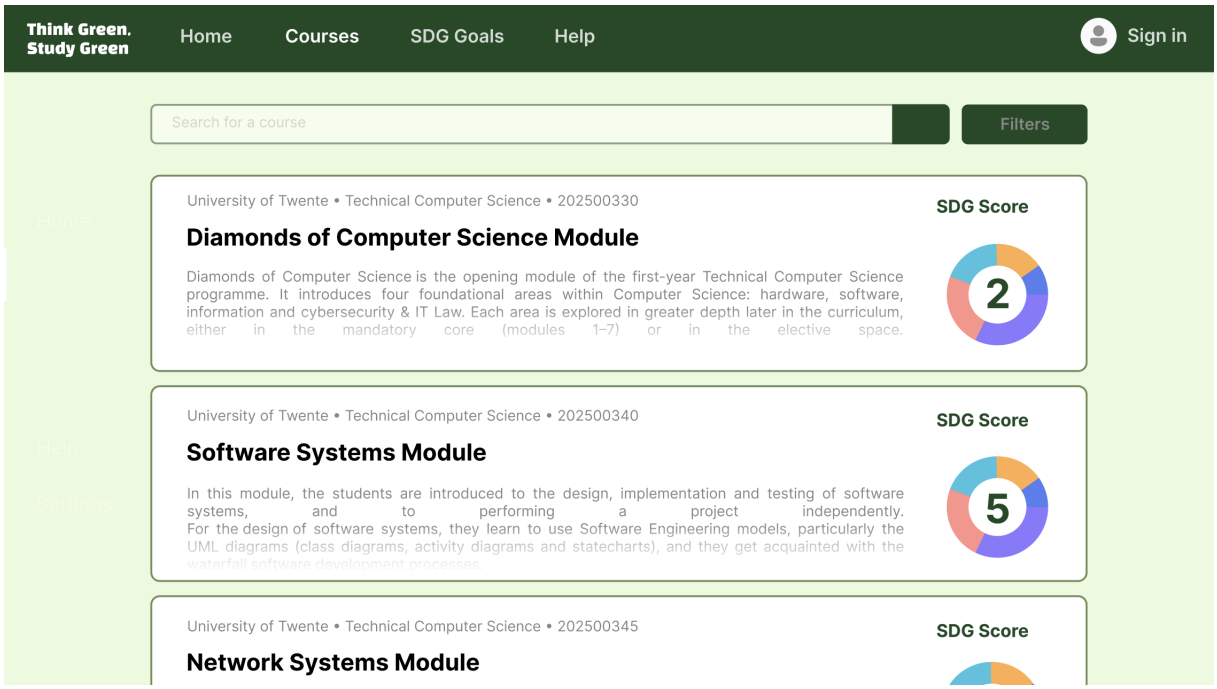


Figure 12: Courses page

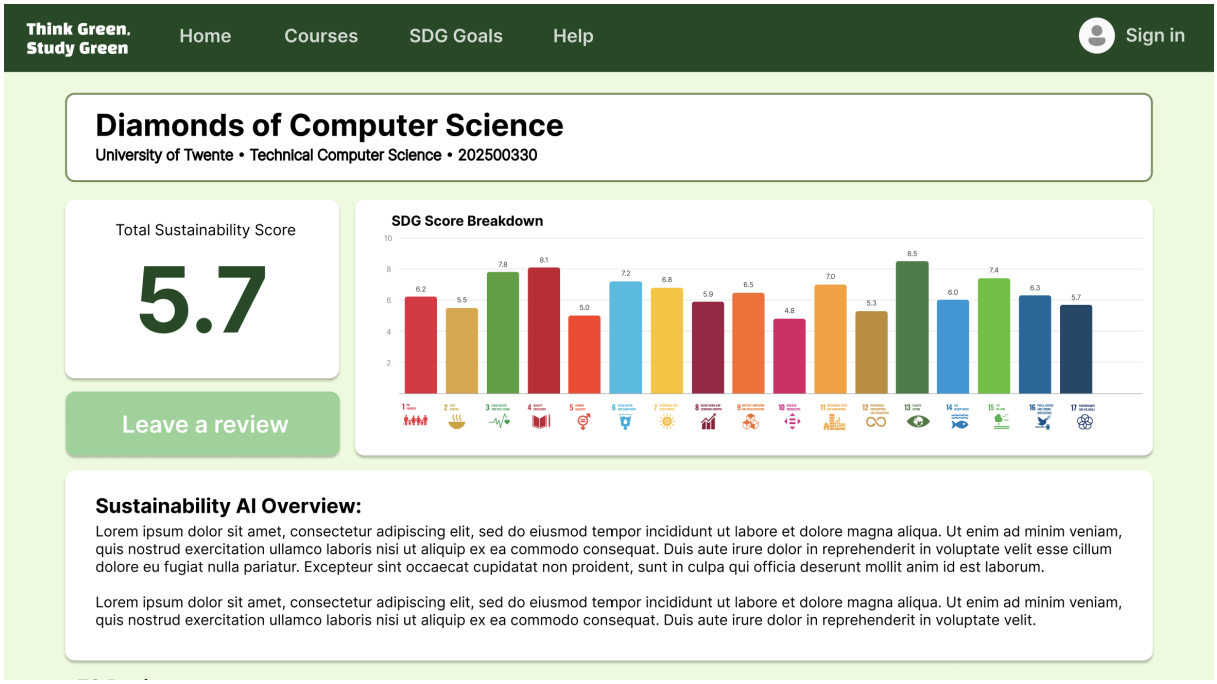


Figure 13: Detailed course page

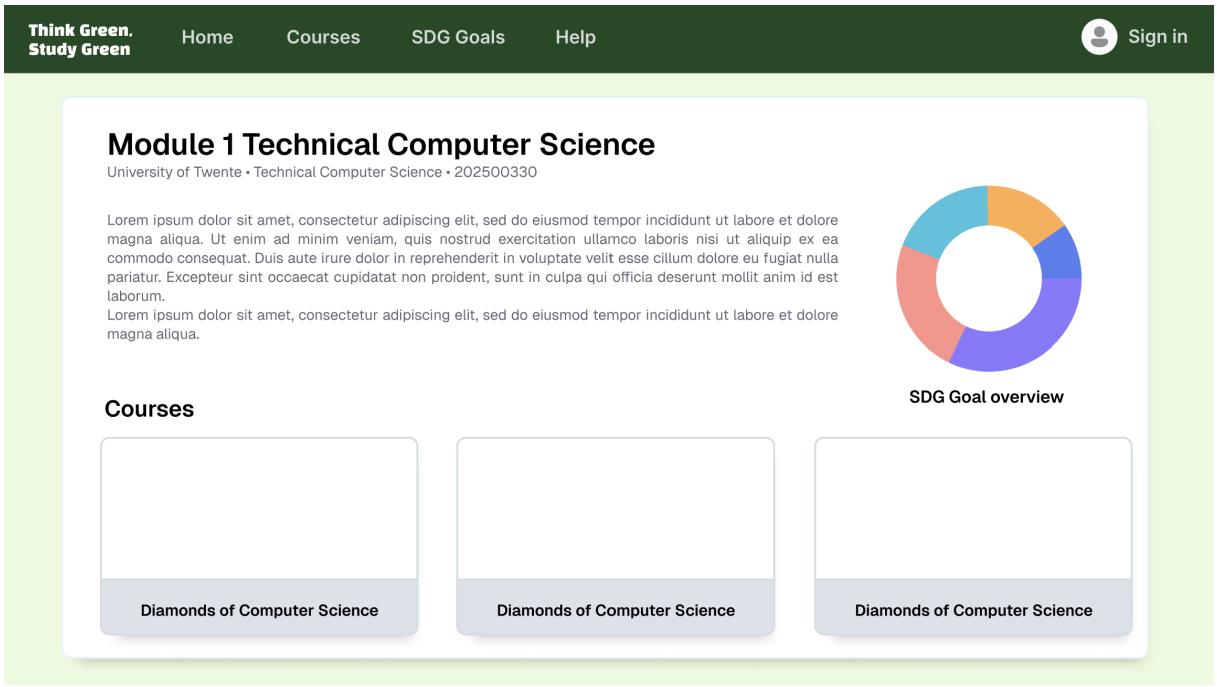


Figure 14: Detailed module page

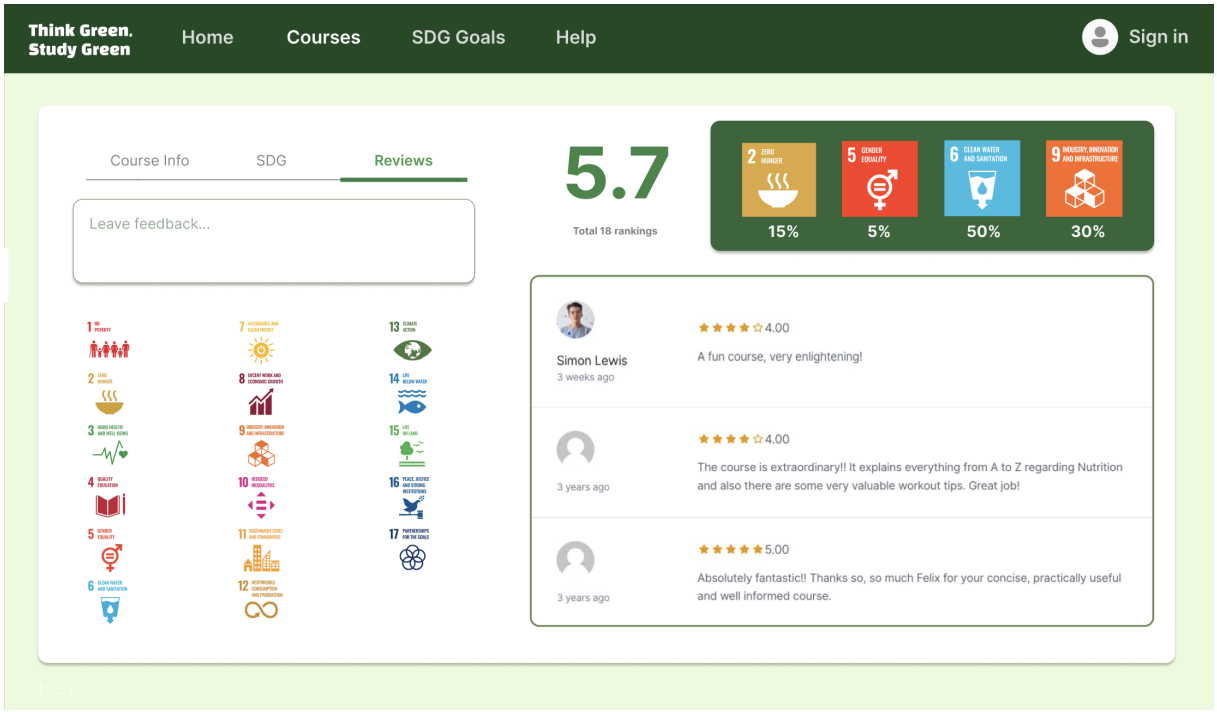


Figure 15: Review page



Figure 16: Sustainable Development Goals page

B User testing interview questions & tasks

B.1 Semi-structured Interview Questions

1. How would you describe your experience with the web application?
2. How would you evaluate the navigability of the web application?

B.2 Follow-up Questions

1. Was there anything that was difficult to find?
2. What did you understand about the Sustainable Development Goals (SDGs) and GreenComp?
3. Did you encounter anything frustrating or annoying that you would like to see changed?
4. Was the amount of information on the website appropriate, excessive, or insufficient?
5. What did you think of the overall visual design? Were there any design choices you found questionable?
6. What changes would you suggest?
7. Were the instructions sufficient and clear?
8. Did the Likert scale make it easy or difficult to rate a course?
9. Were the available filtering options sufficient, or would you suggest adding any others?

B.3 Testing Instructions and Tasks

Participants were asked to complete the tasks in the order presented below. The tasks were adapted to their respective roles as students or teachers. During the testing, participants were encouraged to think aloud, while the supervisors observed their interactions and documented relevant findings.

B.4 Student Tasks

Students were asked to complete the following tasks:

1. Log in as a student.
2. Navigate to the SDG page.
3. Read the descriptions of the SDGs.
4. Navigate to the GreenComp page.
5. Read the information presented on the GreenComp page.
6. Navigate to the list of courses.
7. Navigate to the next page of courses.
8. Return to the first page of courses.
9. Filter the courses by study programme and a selected SDG.
10. Remove the applied filters.
11. Search for “Sustainable Technology”.
12. View the GreenComp information.
13. Submit a course review.
14. Navigate to a module page, where available.
15. Visit another course within the same module.
16. Log out.

B.5 Teacher Tasks

Teachers were asked to complete the following tasks:

1. Log in as a teacher.
2. Navigate to the SDG page.
3. Read the descriptions of the SDGs.
4. Navigate to the list of courses.
5. Search for a specific course.
6. Open the page of the selected course.
7. Edit the AI-generated summary.

8. Navigate to a module page, where available.
9. Visit another course within the same module.
10. Log out.

C Usability Testing interview answers

C.1 Student Interviewee 1

- Q1: "Mostly intuitive. I would like better explanation for the acronyms."
- Q2: "It was fine."
- Q3: "Not really."
- Q4: "Yea I was able to understand what the SDGs were and the GreenComp thing."
- Q5: "SDG chart did not immediately click, especially what the colors mean."
- Q6: "There were 3 acronyms I didn't understand so I'd like more information about those maybe."
- Q7: "It's cute. I like it."
- Q8: "Change the sorting and the filtering."
- Q9: "There are enough instructions."
- Q10: "Yeah the Likert scale was intuitive."
- Q11: "I would like to be able to filter by the amount of SDGs per course, maybe? Yeah."

C.2 Student Interviewee 2

- Q1: "The web application was interactive and encouraged me to read everything despite the fact that I don't care about sustainability. The website is very intuitive."
- Q2: "Very linear and structure makes sense. Even when for a moment I got confused where to find a certain thing, I found it after a second."
- Q3: "On the course page, I did not realise that SDG emphasis dialogue was there, it didn't catch my eye. Only after the review I realised it's there."
- Q4: "I was already familiar with SDGs and GreenComp."
- Q5: "No, it was actually nice."
- Q6: "A bit too much, especially for students, too many pop ups."
- Q7: "I really liked it, but I didn't like that if I slightly hover over it, it jumps up (GreenComp). Maybe put a drop down that says there's info if you click."
- Q8: "It was quite straightforward, I liked the AI disclaimer."
- Q9: "I would usually go from 1 to 10, to have better resolution."
- Q10: "Scroll-able tab would be much better for the filter, now it looks like too much."

C.3 Student Interviewee 3

- Q1: "There's a lot of pages and we go through only a little, everything is self-explanatory. Also why there's Courses page but no Modules page?"
- Q2: "Ctrl F is a beautiful, beautiful thing. In the course page (the button for) the module is tiny."
- Q3: "The module part was the hardest part to find, also GreenComp. There is a lot of free space in SDG Goal Breakdown. The overview should go first, before the AI content. I don't understand why is it so high up."
- Q4: "I already knew about the SDGs, GreenComp was self-explanatory. I like the hover over."
- Q5: "Having to control F (to find a button)."
- Q6: "It was alright. If only one is covered (GreenComp) why is it not a main focus (aesthetically wise)?"
- Q7: "It's a website. Looks like a website. Navigates like a website."
- Q8: "Can we remove Amir from the project? I didn't notice the errors when leaving the review."
- Q9: "Instructions are fine."
- Q10: "I approve."
- Q11: "Wow. It looks like too much."
- Additional comments: "I would like this website to be a main thing, not an additional thing. So I would like to see that information in Osiris. I would use the module as I'm actually interested in sustainability. Most useful for minors, but there are required courses, which you can't just not do because it doesn't match your sustainability goals. Then for required courses it just becomes like additional information, that I don't mind having."

C.4 Student Interviewee 4

- Q1: "Most of it is quite intuitive, it was clear where to press. It's not too busy but clear where things are."
- Q2: "I think it's fine."
- Q3: "No."
- Q4: "SDG goals were more specific, GreenComp had goals with subgoals, and I feel like SDGs have a broader spectrum."
- Q5: "The review card thing felt too much in my face."
- Q6: "Sufficient. Maybe for SDG goals and GreenComp, too few, I would like to know more details."
- Q7: "I liked it."
- Q8: "I think it's very soft, and not too much happening."
- Q9: "Yeah."

- Q10: ”Pretty intuitive.”
- Q11: ”It was fine.”

C.5 Student Interviewee 5

- Q1: ”Pretty straightforward.”
- Q2: ”Nothing except modules button and clear filters.”
- Q3: ”I see SDG as “no poverty”, “equality” and GreenComp as “the way to get there” (as in to the SDGs).”
- Q4: ”Design is nice, doesn’t scream.”
- Q5: ”Yes.”
- Q6: ”Just right.”
- Q7: ”I wouldn’t know what 1-5 means but if you say “not covered” “partially covered” I like it.”
- Q8: ”Maybe filtering by GreenComp.”
- Q9: ”I would go out of my way to use the site as opposed to Osiris when choosing a minor for example. I find it better for reading about potential courses.”
- Q10: ”In the feedback form, the fact that you can flip through the SDG goals was not clear.”
- Q11: Answered in Q8.

C.6 Student Interviewee 6

- Q1: ”I think its great. I think it leaves a lot for the users, for the students, to analyse and make their own choices on their own impact on each of the respective courses and see whether taking that course impacts the future or the sustainability there of. So I think it’s good, I believe its beneficial.”
- Q2: ”It was easy to use and intuitive and tells about every single quota or category of sustainability and how much it impacts society as a whole. Or [what is] your role in society.”
- Q3: ”Not particularly. You had search bars that help you navigate when the lists are long and every other tab was very easy to find and pretty self explanatory.”
- Q4: ”So for the SDGs page, I understood that when approaching a course and seeing whether its sustainable we have to look into multiple categories on different subjects on whether they think about the future of the planet and its safety and also about the humans themselves that are taking the course. For GreenComp, I believe that it is a very good application upon what SDGs set up to do, basically explaining to you in more detail whether or not the courses are taking sustainability and future proofing and whether or not it is fair in the eyes of humans.”
- Q5: ”No, not necessarily. I think it’s well implemented and every button had its own role and place. The graphs were nicely put and absolutely where they should be. Didn’t need a separate tab. Being able to see each goal encapsulated in a small overview was a nice touch.”

- Q6: "It seems just fine. If you want to go and overview every single point it doesn't seem too complicated or too long. The explanations are just the right size, not too many phrases, not too complex, sending just the right amount of info right where u need it."
- Q7: "I think it's great. I think its a very simple approach in design and minimalistic which is appealing on the eyes. Doesn't put a strain on where to look or too obnoxious with colors and icons everywhere. Easy to look at, it makes a difference."
- Q8: "Honestly, maybe multiple additions of a general overview of every course you're taking. A summary of your impact and a summary of the sustainability of what you're taking, but it wouldn't be too necessary."
- Q9: "Yes, it is very easy to navigate and use. Every single tab and button is placed pretty well to the point where you don't have to think much when you're using it. Everything is where u think it should be."
- Q10: "Yes, I think it does. Over the text to door description review, it is a pretty much standard now to use a 1 out of 5 rating and it helps students categorize or rank each courses as they see past students that review each course."
- Q11: "I think its just right, there are a lot of categories to choose from at first, but i believe in the end its necessary."

C.7 Student Interviewee 7

- Q1: "Interface wise, it's pretty good, it's pretty like easy to use in my opinion."
- Q2: "The navigability is very good. I found what I was looking for."
- Q3: "I guess the module thing was difficult because it was a small thing there. It does say module but like its kind of hidden a bit."
- Q4: "The SDG page was like the one that had the different goals listed out and the GreenComp one was basically listing the different like traits or like stuff that gets covered by courses, because like if I go to GreenComp for a course I can see stuff that's checked off from that list that I looked at earlier in the GreenComp page. So basically and overview for every thing that can be covered."
- Q5: "No."
- Q6: "I mean, it feels enough. I did skim through it quickly so I couldn't actually sit and read every part of it but yea I feel like its just enough not too much or too little."
- Q7: "It's pretty cool. I like the pie chart for like the SDG emphasis or whatever and the presentation especially in the courses. Maybe there could be two columns instead of one?"
- Q8: "Other than the like course organization and the search bar, maybe just two columns for those or the ability to change how many are shown on one page. So a grid style instead. That's about it I guess."
- Q9: "Yea."
- Q10: "Yea, I feel like it does make sense."
- Q11: "It's enough it's literally exactly what you need."

C.8 Student Interviewee 8

- Q1: "Website itself looks nice and works well but it looks a bit generic and I find it hard to know if the ratings are trustworthy"
- Q2: "Website was very navigate-able, going from course page to module was a bit hard to find. It looks like it would link to outside the website due to the icon."
- Q3: "No"
- Q4: "I understood that the SDGs are goals made by the UN aimed at improving sustainability in the world but I'm somewhat skeptical of rating courses on these. Greencomp is a similar framework but kind of more focused on education."
- Q5: "Hovering over stuff is annoying and it should be an on click", "GreenComp overview UI on the course page is hellish", "List of filters is overwhelming"
- Q6: "It's a good amount of information"
- Q7: "Website looks decent but also a bit generic and corporate which doesn't invoke greenness", "Inconsistent color design between pages"
- Q8: "Logging out doesn't kick you out to the home page and it should probably do that", "Filter options are fine but need quality of life features", "I was able to leave multiple reviews on the same course which might cause people to spam courses so that should change.", "Change hover elements to clickable elements with similar icons as the questions on the help page."
- Q9: "Yes."
- Q10: "Likert scale is fine but there should also be a text box for each individual sdg so you can say specific things about how each sdg is covered more formally"
- Q11: "Search is weird and sometimes doesn't immediately give what you would expect based on what you type.", "Filters are mostly fine as I said but there should be a remove filters button and a way to segment the list of filters to make it more readable."

C.9 Teacher Interviewee 1

- Q1: "Overall quite smooth, was able to fins things quickly. Looks slick in general not too cramped."
- Q2: "Satisfied."
- Q3: "No, there wasn't."
- Q4: "I am already familiar with both."
- Q5: "Some unnecessary animations."
- Q6: "It was a good amount, nothing too much. SDG box had quite a small font which might be annoying for older teachers."
- Q7: "Looks nice, liked the images images, I suggest to add that to the courses page also."
- Q8: "I have mentioned before, focus now on filter and font."
- Q9: "Not all instructions were clear. Check whether people grasp well what SDGs are and the purpose of the website."

- Q10: ”Hated hovering over. Overall rating for a course, how many inputs were there. The 1-5 scale is good, I prefer it over a binary option.”
- Q11: ”Filters make sense, but if we can filter by SDGs why can’t we filter by GreenComp?”

C.10 Teacher Interviewee 2

- Q1: ”I really liked it. As long as people know about it it would be very useful in humanitarian engineering”
- Q2: ”Easy to navigate”, ”use friendly” ”I liked that there isn’t too much text and the graphics are nice”
- Q3: ”No”
- Q4: (familiar with both) ”The link to the official SDG and GreenComp pages should be more clear”.
- Q5: ”GreenComp feels like an afterthought, it’s not very well integrated”
- Q6: ”There was not too much information”
- Q7: ”Beautiful website visually”, ”Adding some of the house style of the UT would be nice and UT graphics and colors if it’s part of the UT system”
- Q8: ”You should also be able to sort by quartile”, ”Reviwers are unable to say which GreenComp subgoals are covered”.
- Q9: ”Yes”
- Q10: ”The likert scale is generally good but a little uninterpretable, in education 1 is the highest grade and 5 is lowest. Generally good but maybe change to stars or explain it better?”
- Q11: ”You should also be able to filter by bachelors and masters courses, I was a little confused when you told me to navigate to the module page since masters don’t have modules”

D Unit tests

D.1 Course Page Functionality

Test Case 5.1: Course Overview

ID	5.1		
Name	Course Overview		
Description	Users can view the sustainability information associated with a course.		
Prerequisites	A course with generated sustainability scores and assessment information exists.		
Step	**Test step**	**Expected result**	**Actual result**
1	Open a course or module page.	The course title and accompanying information are displayed. The Overview section is open by default and displays SDG Emphasis, SDG Goal Breakdown, GreenComp competencies, and AI Overview.	Passed.
2	View the SDG Emphasis section.	The system displays the number of SDGs covered, an SDG pie chart, and dropdowns containing the assessment rationale.	Passed.
3	View the SDG Goal Breakdown section.	A bar chart summarising the SDGs covered by the course is displayed.	Passed.
4	Hover over a bar in the SDG Goal Breakdown chart.	A tooltip displays the AI-generated rationale for the corresponding SDG.	Passed.
5	Switch to the GreenComp competencies section.	The system displays the GreenComp competency sub-goals covered by the course.	Passed.
6	Hover over a GreenComp competency box.	A tooltip provides the AI-generated rationale explaining why the corresponding sub-goal is considered covered or not covered.	Passed.
7	View the AI Overview section.	The system displays a general AI-generated overview of the SDGs covered by the course.	Passed.

Test Case 5.2: Course Review

ID	5.2		
Name	Course Review		
Description	Users can provide feedback on the sustainability assessment of a course.		
Prerequisites	The user is logged in and the course is available for review.		
Step	**Test step**	**Expected result**	**Actual result**
1	Open the Review section on the course page.	Existing reviews are displayed. If no reviews exist, the message *No reviews yet...* is displayed.	Passed.
2	Submit a review.	The system saves the review and associates it with the current user.	Passed.
3	View the Review section again.	The newly submitted review is displayed under the current user's name.	Passed.

Test Case 5.3: Course Information

ID	5.3		
Name	Course Information		
Description	Users can view the course description, learning goals, and aims obtained from Osiris.		
Prerequisites	The course page contains a course description, learning goals, and aims.		
Step	**Test step**	**Expected result**	**Actual result**
1	Open the Course Information section on the course page.	The system displays the course description, learning goals, and aims under their respective headings.	Passed.

D.2 Teacher Editing Functionality

Test Case 6: Teacher Editing Page

ID	6		
Name	Teacher Editing Page		
Description	Teachers can review and edit LLM-generated course sustainability information.		
Prerequisites	The user is logged in as a teacher, and a course with generated sustainability information exists.		
Step	**Test step**	**Expected result**	**Actual result**
1	Navigate to the editing page of a course.	The system displays the existing course information and generated sustainability assessment.	Passed.
2	Edit the course information or sustainability assessment.	The system allows the teacher to modify the available fields.	Passed.
3	Save the changes.	The system saves the changes and updates the information displayed for the course.	Passed.
4	Reopen the course after saving the changes.	The previously saved changes are still displayed.	Passed.

D.3 Information Pages and Navigation Bar

Test Case 7: SDG Page

ID	7		
Name	SDG Page		
Description	Users can view information about the Sustainable Development Goals (SDGs).		
Prerequisites	None.		
Step	**Test step**	**Expected result**	**Actual result**
1	Navigate to the SDG page.	The system displays all SDGs.	Passed.
2	Hover over an individual SDG.	The system displays additional information about the selected SDG.	Passed.

Test Case 11: GreenComp Page

ID	11		
Name	GreenComp Page		
Description	Users can view information about the GreenComp sustainability competencies.		
Prerequisites	None.		
Step	**Test step**	**Expected result**	**Actual result**
1	Navigate to the GreenComp page.	The system displays all four GreenComp competency areas and their corresponding sub-goals.	Passed.
2	Hover over a sub-goal.	The sub-goal box expands and displays a more detailed description.	Passed.

Test Case 12: Navigation Bar

ID	12		
Name	Navigation Bar		
Description	Users can navigate to relevant pages and manage their login session using the navigation bar.		
Prerequisites	None.		
Step	**Test step**	**Expected result**	**Actual result**
1	Click the Home page link.	The system navigates to the Home page.	Passed.
2	Click the Courses page link.	The system navigates to the Courses page.	Passed.
3	Click the SDG page link.	The system navigates to the SDG page.	Passed.
4	Click the GreenComp page link.	The system navigates to the GreenComp page.	Passed.
5	Click the Help page link.	The system navigates to the Help page.	Passed.
6	Log in through the navigation bar.	The login session is maintained across pages, and the user's name and a logout option are displayed in the navigation bar.	Passed.

D.4 Authentication and Access Control

Test Case 1: Student Login

ID	1		
Name	Student Login		
Description	Students can log in and access pages and functionality corresponding to their role.		
Prerequisites	A valid student account with a username and password exists.		
Step	**Test step**	**Expected result**	**Actual result**
1	Enter a username and password and select the student role.	The system accepts the entered credentials for submission.	Passed.
2	Click the *Log in* button.	The system authenticates the user and grants access to the student interface.	Passed.
3	Navigate through the available pages and review courses.	The user can access pages and functionality permitted for students.	Passed.

Test Case 2: Teacher Login

ID	2		
Name	Teacher Login		
Description	Teachers can log in and access pages and functionality corresponding to their role.		
Prerequisites	A valid teacher account with a username and password exists.		
Step	**Test step**	**Expected result**	**Actual result**
1	Enter a username and password and select the teacher role.	The system accepts the entered credentials for submission.	Passed.
2	Click the *Log in* button.	The system authenticates the user and grants access to the teacher interface.	Passed.
3	Navigate through the available pages and edit course pages.	The user can access pages and functionality permitted for teachers.	Passed.

Test Case 8: Unauthenticated Review Prevention

ID	8		
Name	Unauthenticated Review Prevention		
Description	Users who are not logged in cannot submit course reviews.		
Prerequisites	The user is not logged in.		
Step	**Test step**	**Expected result**	**Actual result**
1	Navigate to the review section of a course.	The *Leave a review* button is not displayed.	Passed.

Test Case 9: Student Access Restrictions

ID	9		
Name	Student Access Restrictions		
Description	Students cannot edit course information or access teacher-only pages.		
Prerequisites	The user is logged in as a student.		
Step	**Test step**	**Expected result**	**Actual result**
1	Navigate to a course page.	The option to edit course information is not displayed.	Passed.
2	Attempt to access the teacher editing page through its URL.	Access is denied and the student cannot view or edit the page.	Passed.

D.5 Course and Module Navigation

Test Case 3: Filtering Courses and Modules

ID	3		
Name	Filtering Courses and Modules		
Description	Users can filter courses and modules using the available filter options.		
Prerequisites	Courses and modules are available in the database.		
Step	**Test step**	**Expected result**	**Actual result**
1	Select one or more filter options.	Only courses and modules matching the selected filters are displayed.	Passed.

Step	Test step	Expected result	Actual result
2	Change or remove the selected filters.	The displayed courses and modules are updated accordingly.	Passed.

Test Case 4: Searching Courses and Modules

ID	4		
Name	Searching Courses and Modules		
Description	Users can search for courses and modules by name or code and navigate through the results.		
Prerequisites	Courses and modules are available in the database.		
Step	**Test step**	**Expected result**	**Actual result**
1	Enter a course or module name or code in the search bar.	The system displays courses or modules matching the query.	Passed.
2	Select a course or module from the search results.	The system opens the corresponding course or module overview page.	Passed.
3	Enter a query with no matching results.	The system indicates that no matching courses or modules were found.	Passed.
4	Click the next-page button.	The system displays the next page of results, containing up to 20 items while preserving the selected filters and search query.	Passed.
5	Select a course from the displayed results.	The system navigates to the corresponding course page.	Passed.

Test Case 10: Module Page Navigation

ID	10		
Name	Module Page Navigation		
Description	Users can navigate between a module page and the courses belonging to that module.		
Prerequisites	The user is viewing a course that belongs to a module.		
Step	**Test step**	**Expected result**	**Actual result**
1	Click the module button.	The system opens the module page and displays information about the entire module.	Passed.
2	Select one of the courses within the module.	The system opens the corresponding course page.	Passed.
3	Click the *Leave a review* button.	The user can access the review form if logged in.	Passed.

E UML Diagrams

This appendix presents the supplementary architectural and behavioral UML diagrams referenced in Section 3. Note that while the ingestion and evaluation diagrams reflect the implemented data pipeline, several interaction diagrams illustrate the complete target system architecture, incorporating planned future extensions (such as institutional Single Sign-On, automated review summarization, and the recommendation chatbot).

E.1 State Machine Diagram

The Course Evaluation Lifecycle state machine diagram (presented earlier as Figure 1 in Section 3.2) models the complete state transitions of a scraped course record, from initial discovery in Osiris and HTML cleaning through multi-tier filtering, LLM evaluation, professor overrides, and eventual archival.

E.2 Class Diagram

Figure 17 illustrates the initial conceptual domain model designed during the early architectural phase to map the core entities, enumerations, and relationships required by the platform. During implementation, this conceptual model evolved into the normalized relational Supabase schema shown in Figure 4.

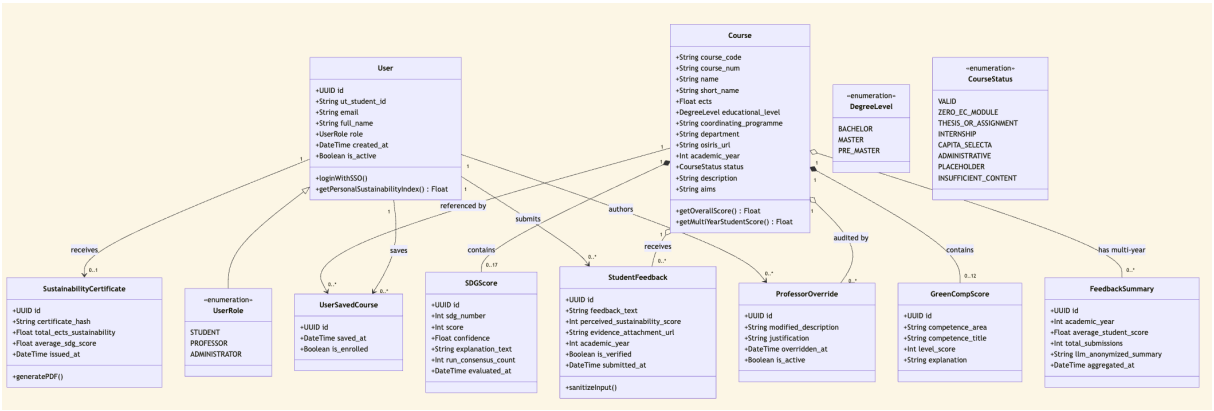


Figure 17: Initial Conceptual Domain Model

E.3 Sequence Diagrams

While the backend ingestion and LLM grading sequence is detailed in Section 3.2 (Figure 2), the following sequence diagrams model the primary user-facing workflows and planned extensions of the platform:

- **Institutional Authentication (Figure 18):** Illustrates the target authentication flow for students and teaching staff once University of Twente Single Sign-On (SSO) and Osiris profile synchronization are integrated.
- **Student Feedback Processing (Figure 19):** Details the end-to-end processing of student course reviews, including input sanitization, and a planned asynchronous worker for generating anonymized LLM review summaries.
- **Instructor Course Override (Figure 20):** Demonstrates how authenticated teaching staff can review, modify, and override the AI-generated sustainability overview and SDG alignment with their own verified assessment.
- **Course Chatbot and Sustainability Certificate (Figure 21):** Models two exploratory features—an interactive RAG-based course recommendation chatbot and a downloadable student sustainability certificate—that were initially scoped as “Should Have” requirements before being de-prioritized to focus on core platform functionality.

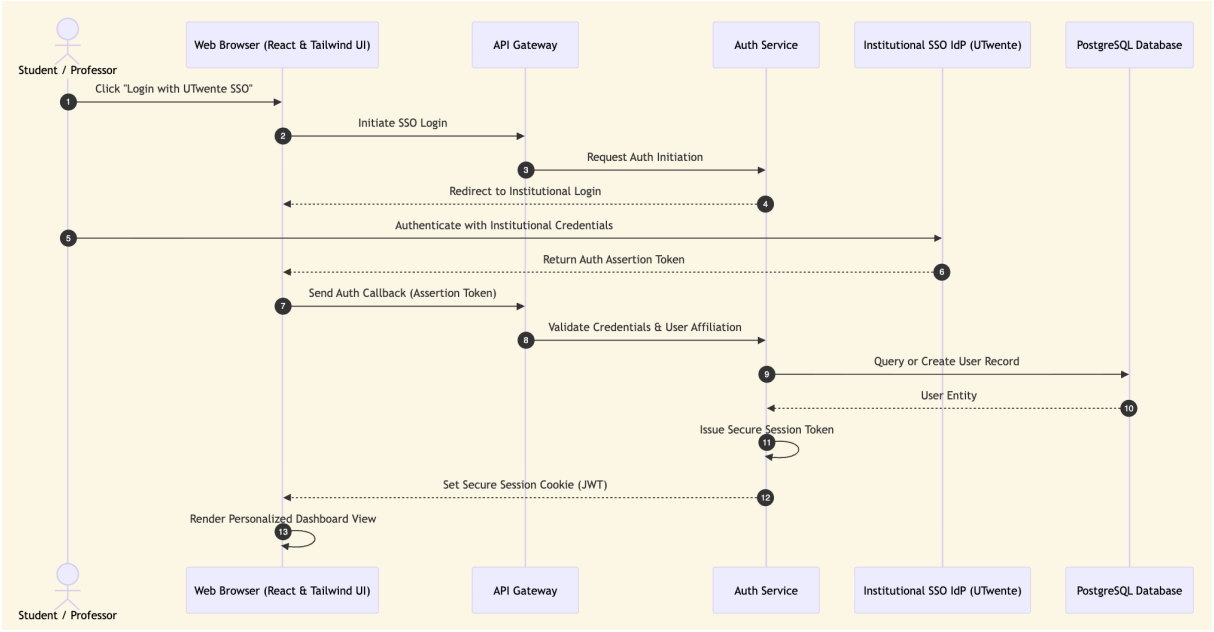


Figure 18: Authentication: Institutional SSO & Profile Sync (Target Architecture)

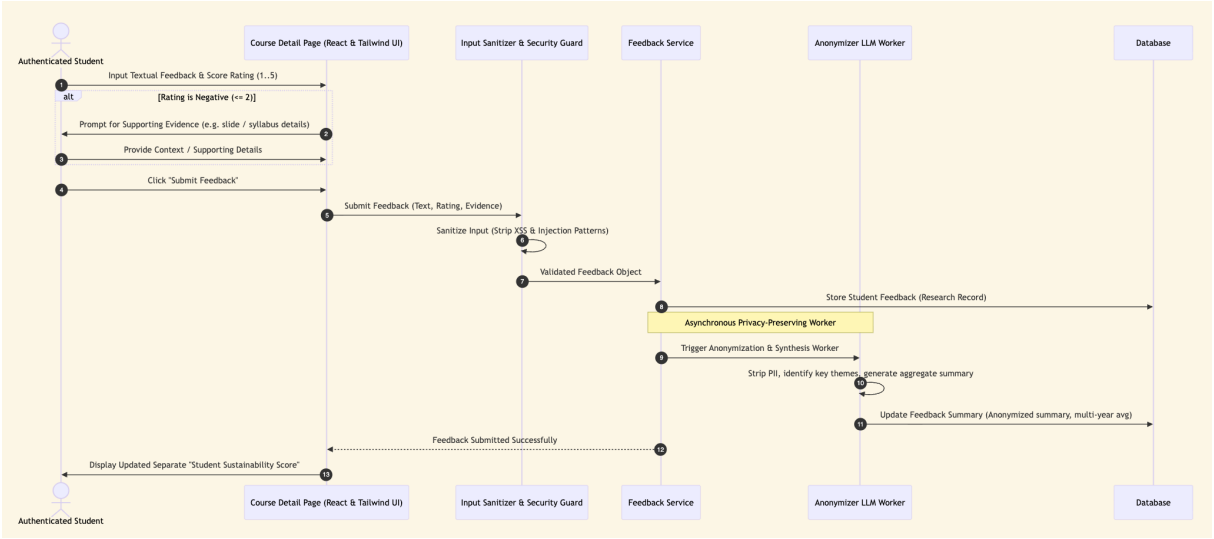


Figure 19: Student Feedback Submission with Griefing Prevention

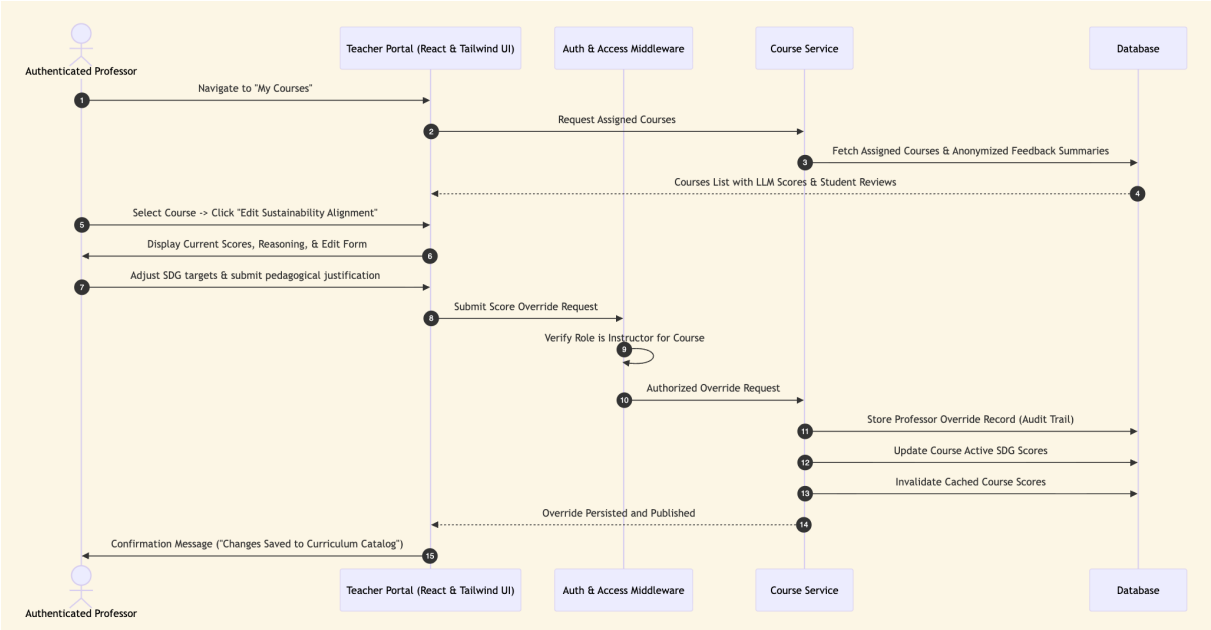


Figure 20: Professor Review and Course Material Override

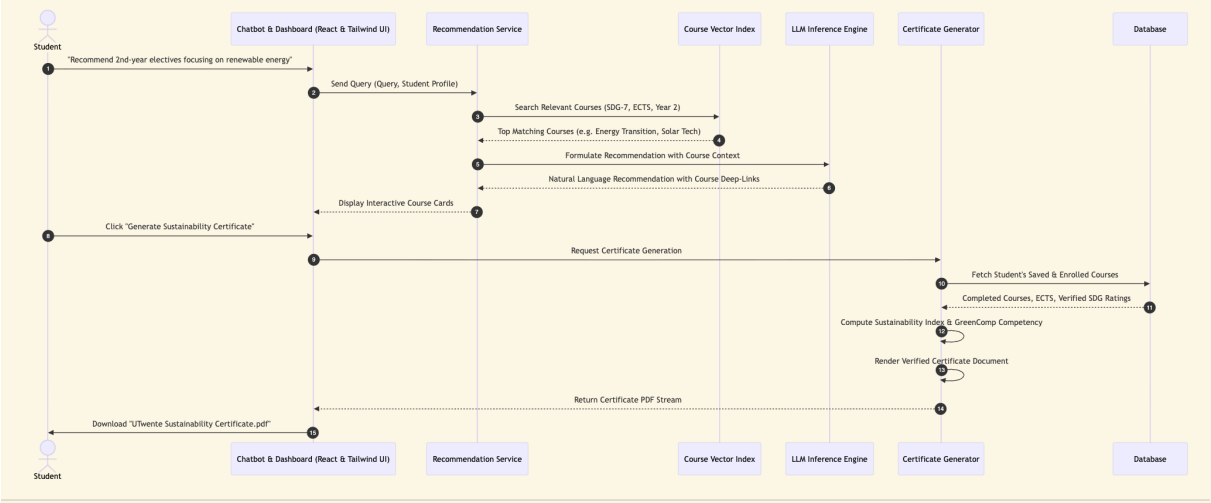


Figure 21: Recommendation Chatbot & Certificate Generation (Exploratory Features)

E.4 Activity Diagram

Figure 22 details the internal decision logic of the multi-tier filtering script (`filter_utils.py`). By deterministically excluding administrative placeholders, zero-credit modules, internships, theses, and courses with insufficient text prior to inference, this pipeline prevents wasted LLM compute time and reduces unnecessary carbon emissions.

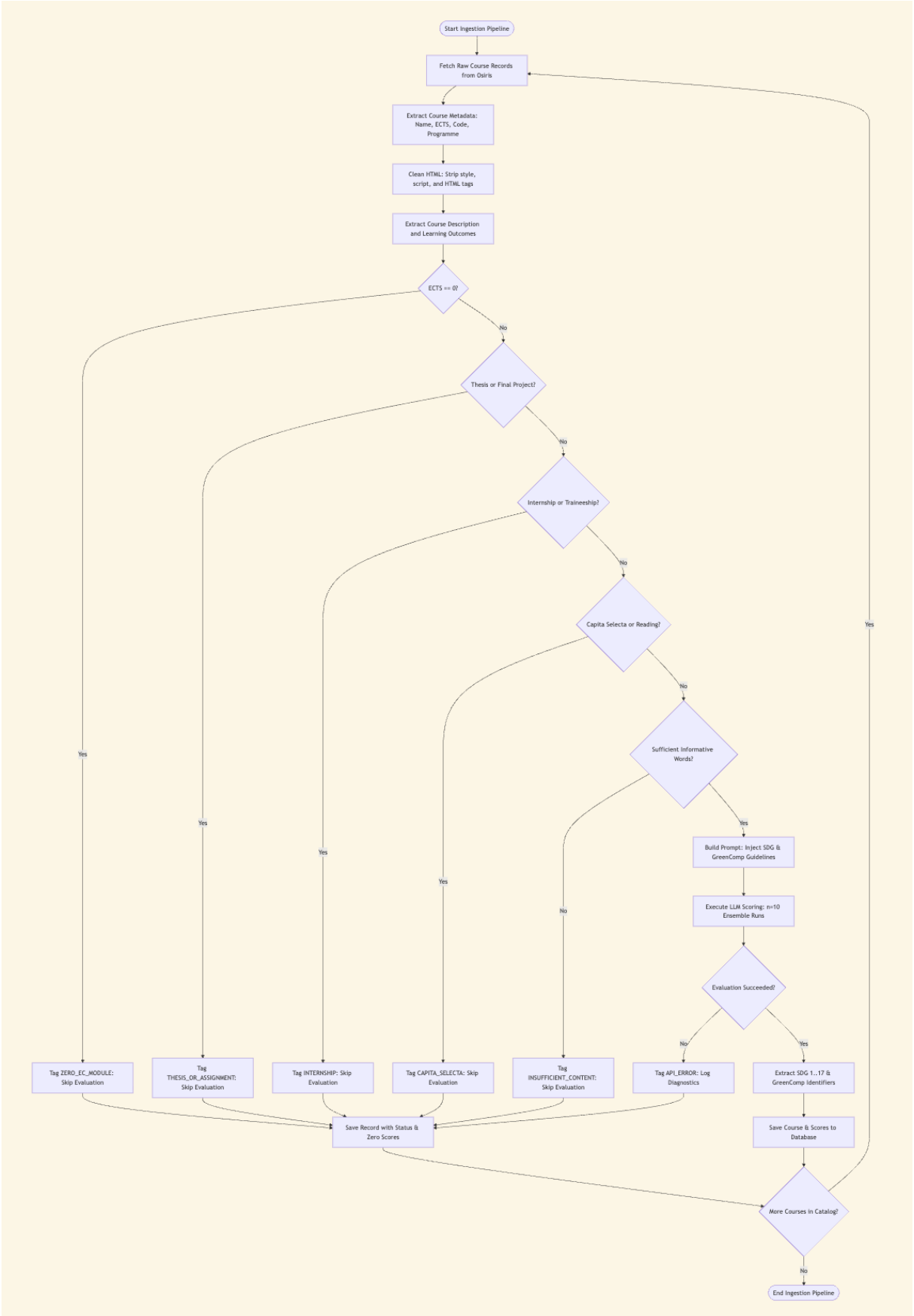


Figure 22: Course Ingestion & Multi-Tier Filtering Logic

E.5 Topology & Architecture

Figure 23 provides a high-level deployment topology of the system, illustrating how the client web browser, Next.js frontend and backend application services, background ingestion worker, local LLM inference service, and encrypted PostgreSQL database interact across network boundaries.

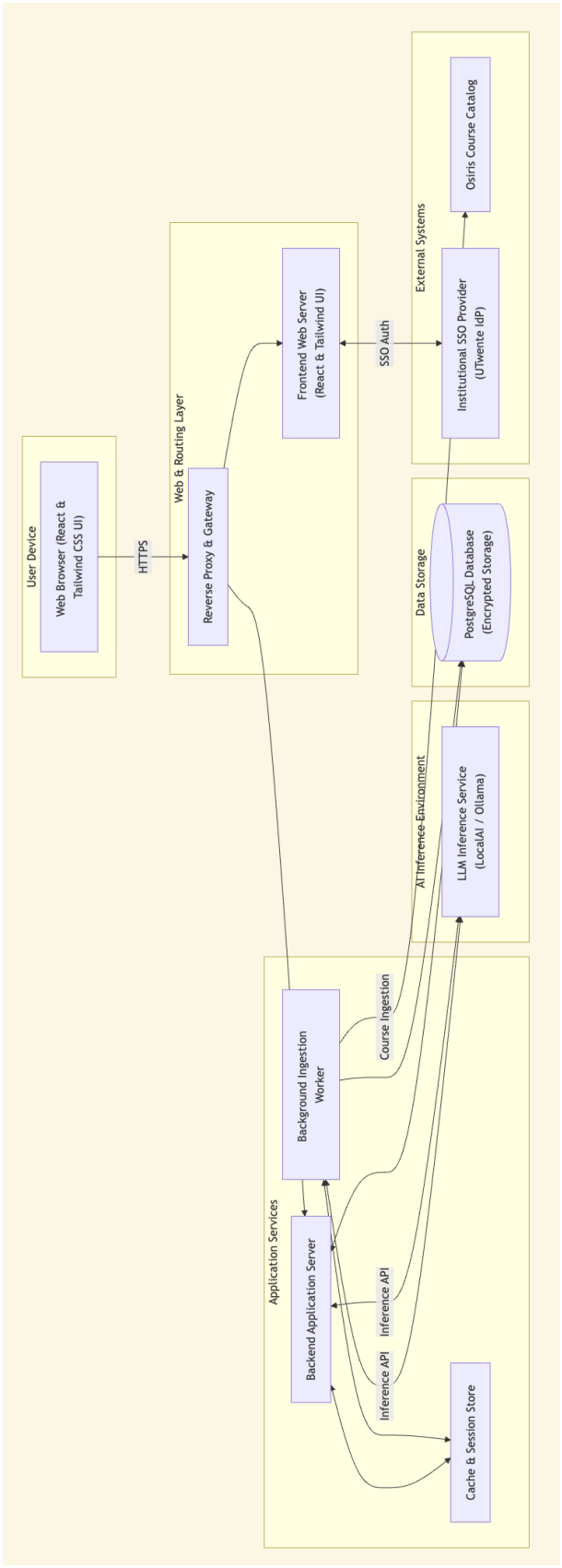


Figure 23: System Deployment Topology

References

[1] Ali Khalfan Al-Naqbi and Qasim Alshannag. The status of education for sustainable development and sustainability knowledge, attitudes, and behaviors of UAE university students. *International Journal of Sustainability in Higher Education*, 19(3), 2018.

[2] Guia Bianchi, Ulrike Pisiotis, and Giraldez Marcelino Cabrera. Greencomp: The european sustainability competence framework. *JRC Publications Repository*, 2022.

[3] Nationaal Energie Dashboard. Samen slimmer met energie — nationaal energie dashboard.

[4] United Nations. The 17 goals — sustainable development. https://sdgs.un.org/goals, 2015. Accessed: 2026.

[5] Anne Urai, Sascha Jansz, Sanli Faez, Cees Haringa, and Rogier Kievit. Think green, act green: Manifesto for knowledge institutions. July 2025.