

# MSc in Computer Science - Team Project

**Interim Report**

**Team Name: AI Study Buddy (Group 3)**

Anika Mayesha – D24125187

Lorenzo Palleschi – D24126922

Rumaysa Babulkhair – D24125711

Yurii Sykal – C23512523

Contents

[MSc in Computer Science - Team Project 1](#_Toc212126563)

[Table of Figures 3](#_Toc212126564)

[List of Tables 4](#_Toc212126565)

[1. Introduction 5](#_Toc212126566)

[2. User Scenario: The Characters 6](#_Toc212126567)

[2.1 Identifying target users 6](#_Toc212126568)

[2.1.1 Personas 6](#_Toc212126569)

[2.2 Importance of target users 8](#_Toc212126570)

[2.2.1 Survey 8](#_Toc212126571)

[2.3 User Problems 10](#_Toc212126572)

[3. Technical Problem: The Setting 11](#_Toc212126573)

[3.1 Reasons for Building This Application 11](#_Toc212126574)

[3.2 Core Technical Problem 11](#_Toc212126575)

[3.3 Existing Systems Review 12](#_Toc212126576)

[3.4 Technical and Educational Context 14](#_Toc212126577)

[3.5 Summary of the Technical Problem 14](#_Toc212126578)

[4. Technical Solution: The Plot 15](#_Toc212126579)

[4.1 System Overview 15](#_Toc212126580)

[4.2 System Architecture Design 16](#_Toc212126581)

[4.2.1 Front-End: React Client 16](#_Toc212126582)

[4.3. AI Chat Tutor and Data Flow 18](#_Toc212126583)

[4.3.1 Retrieval and Response Generation 18](#_Toc212126584)

[4.3.2 Future Work: Decentralised AI Strategy 19](#_Toc212126585)

[4.3.2 Pomodoro Timer 19](#_Toc212126586)

[4.3.3 Flashcards, Quizzes, and Mock Exams 19](#_Toc212126587)

[4.3.4 Achievements, Analytics, and User Notes 20](#_Toc212126588)

[4.3.5 Node.js Back-end 20](#_Toc212126589)

[4.4 Architectural Style 20](#_Toc212126590)

[4.5 Summary 20](#_Toc212126591)

[5. Evaluation: The Reviews 21](#_Toc212126592)

[5.1 Defining Success 21](#_Toc212126593)

[5.2 System Evaluation 21](#_Toc212126594)

[5.2.1 Heuristic Evaluation 21](#_Toc212126595)

[5.2.2 Cognitive Walkthroughs and Think-Aloud 21](#_Toc212126596)

[5.2.3 Quantitative Testing 22](#_Toc212126597)

[5.3 Success Criteria 22](#_Toc212126598)

[5.4 Ethical Considerations 23](#_Toc212126599)

[6. Conclusion: The Plan 24](#_Toc212126600)

[6.1 Project Management Strategy 24](#_Toc212126601)

[6.1.1 Team Meetings 24](#_Toc212126602)

[6.2 Ongoing Challenges 25](#_Toc212126603)

[6.3 Time Management and Future Priorities 25](#_Toc212126604)

[6.4 Expected Outcome 25](#_Toc212126605)

[7. References and Key Resources 26](#_Toc212126606)

[8. Appendix 28](#_Toc212126607)

[8.1 Appendix A – Front-end Prototypes 28](#_Toc212126608)

[8.1.1 Initial Dashboard Prototype 28](#_Toc212126609)

[8.1.2 Initial Module Selection Prototype 29](#_Toc212126610)

[8.1.3 Dashboard Update 30](#_Toc212126611)

[8.1.4 Module Selection Update 30](#_Toc212126612)

# Table of Figures

[Figure 1: Application Logo 5](#_Toc212126545)

[Figure 2: Persona 1 - (The consistent studier) 7](#_Toc212126546)

[Figure 3: Persona 2 - (The coaster) 7](#_Toc212126547)

[Figure 4: Persona 3 - (The mature student) 8](#_Toc212126548)

[Figure 5: Survey Response - Age Group 9](#_Toc212126549)

[Figure 6: Survey Response - AI Majors 9](#_Toc212126550)

[Figure 7: Survey Response - AI Usage 10](#_Toc212126551)

[Figure 8: Quizlet Paywall 12](#_Toc212126552)

[Figure 9: Gizmo.AI Homepage 13](#_Toc212126553)

[Figure 10: Gizmo.AI Upload Section 13](#_Toc212126554)

[Figure 11: Learning Cycle 15](#_Toc212126555)

[Figure 12: System Architecture 16](#_Toc212126556)

[Figure 13: Dashboard Version 1.0 (Logged in) [Figma] 17](#_Toc212126557)

[Figure 14: Sketch Update 17](#_Toc212126558)

[Figure 15: Potential Updated Dashboard (Logged in) [Figma] 18](#_Toc212126559)

[Figure 16: Back-end Libraries Utilised 20](#_Toc212126560)

[Figure 17: Common Application Methods & Example 20](#_Toc212126561)

[Figure 18: Notion Task Board 24](#_Toc212126562)

# List of Tables

[Table 1: Existing System Comparison 12](#_Toc212125468)

[Table 2: Technology Justification 16](#_Toc212125469)

[Table 3: Sample Testing Data 22](#_Toc212125470)

[Table 4: Sample Quantitative Questionnaire 22](#_Toc212125471)

[Table 5: Success Criteria 23](#_Toc212125472)

# 1. Introduction

The goal of this project was to build a mobile first web application named **AI Study Buddy**, which provides an essential service for the modern-day university student. As artificial intelligence (AI) is in a state of rapid advancement, the services it can provide for students are ever expanding; allowing for a highly adaptive and personalised user experience. According to the *OECD (2023),* AI is becoming far more prominent within higher education, particularly in areas such as automated tutoring, content summarisation, and intelligent feedback systems. Despite the aforementioned advancements, many learners are still reliant on a fragmented ecosystem of various applications, rather than one application which accommodates all of their needs. This issue is what AI Study Buddy aims to resolve, by centralising all of a user’s needs into one cohesive space.

This report will run through various details, ranging from who the application is designed for, the core issues faced, along with their resolutions, the technical architecture, user evaluations, and will conclude with how the application evolves from this data.

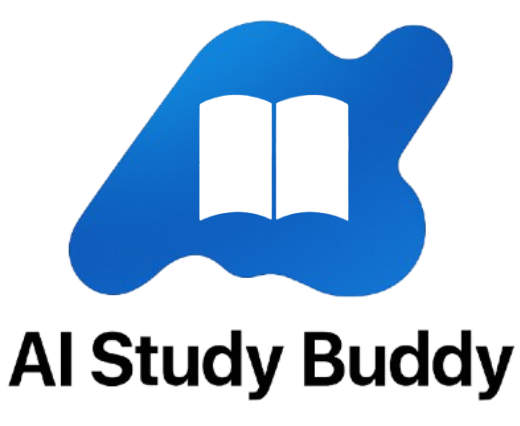


Figure 1: Application Logo

# 2. User Scenario: The Characters

## Identifying target users

AI Study buddy is primarily aimed at third-level students who aim to improve their studying habits with the use of artificial intelligence. This groups in both undergraduate and postgraduate learners across various disciplines and age groups. We can also account for some of the potential struggles that students face, varying from balancing their social, academic, financial, and work commitments, often times leading to time pressuring, and information overloads. Whilst the app is aimed at university students, it may also be utilised by independent learners or professionals who want to further their knowledge within a field and have the need for structured adaptive feedback.

Many students are typically interacting with a multitude of digital tools, ranging from AI chatbots for explanations, note-taking apps for documentation, and even productivity apps to manage their time. However, the use of these applications in isolation creates an inefficient workflow, and cognitive fragmentation. As reported by the Higher Education Authority (*HEA, 2025*), there are over 275,000 students which are currently enrolled in Irish higher education, many of which are reliant on hybrid and digital study modes. The OECD’s *Digital Education Outlook (2023)* notes that whilst AI-tools are becoming increasingly prevalent, their lack of unification remains as a key barrier to real adoption within an academic context *(OECD, 2023).*

As such, AI Study Buddy targets this gap by merging AI oriented note summaries, adaptive quiz generation, and productivity tracking into one accessible platform. This is notably relevant for any students who are reliant on a personalised learning experience, yet do not have the time or resources to build the structured system themselves. The userbase is highly targeted to retain motivation, provide personalised feedback, and support both independent and collaborative learning.

### Personas

Personas are a way for us to further identify our target users; by creating semi-fictional characters, we can represent the various student types, along with their needs, experiences, behaviours, frustrations, and goals. Viewing the product through the eyes of these personas allows us to take somewhat of an outside perspective, and provides a way for us to better understand what is desired from the diverse userbase.

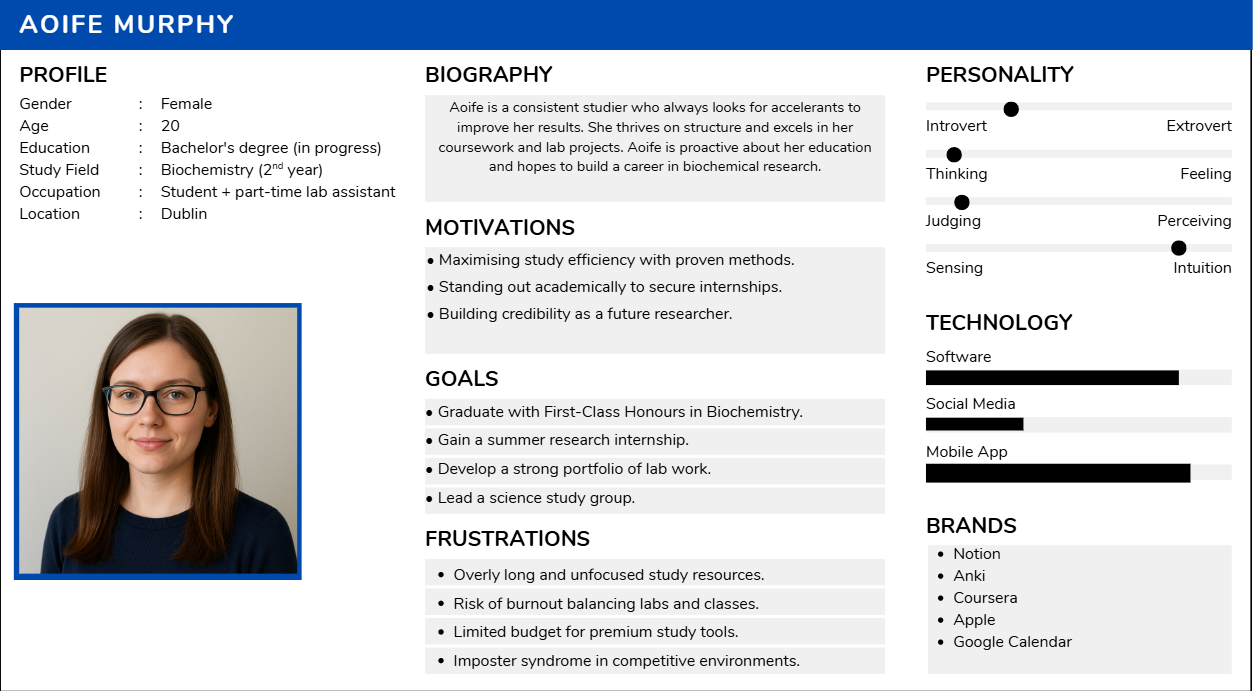


Figure 2: Persona 1 - (The consistent studier)

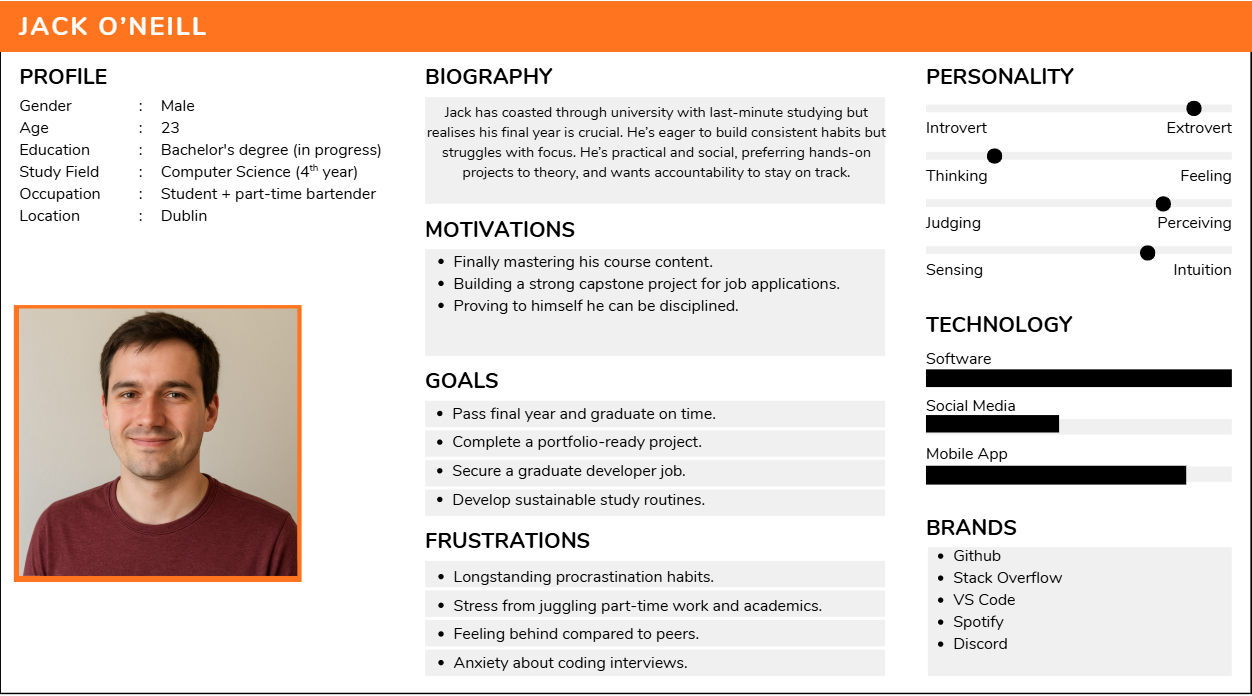


Figure 3: Persona 2 - (The coaster)

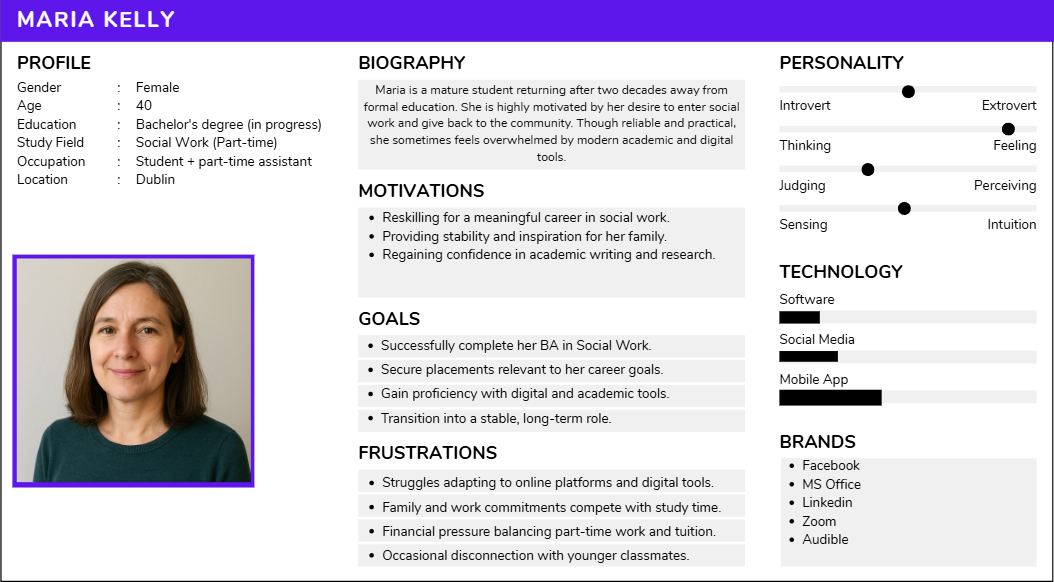


Figure 4: Persona 3 - (The mature student)

These 3 personas provide a diverse userbase for university level studies, and help us understand deeper, what some of the struggles may be; allowing us to accommodate various situations in an adaptable manner.

## Importance of target users

Ultimately, ensuring that the target users are identified correctly, will ensure that the projects development remains on track with relevant features being prioritised and issues being resolved.

### Survey

To verify this data for ourselves, we conducted a survey with various students around the local campus, and student accommodations.

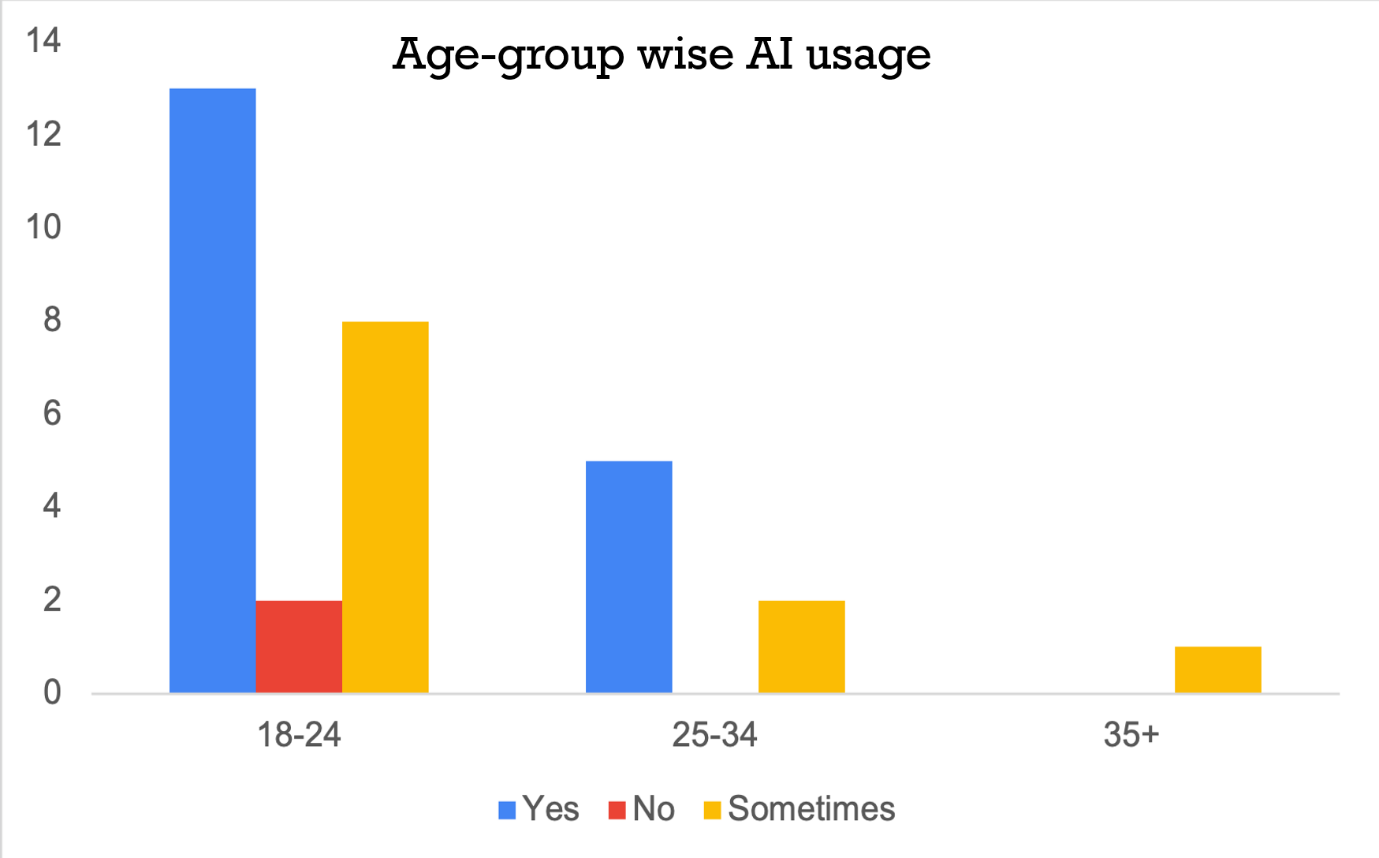


Figure 5: Survey Response - Age Group

Whilst limited to a total of 33 responses, we can get a rough idea of how various age groups are interacting with AI usage.

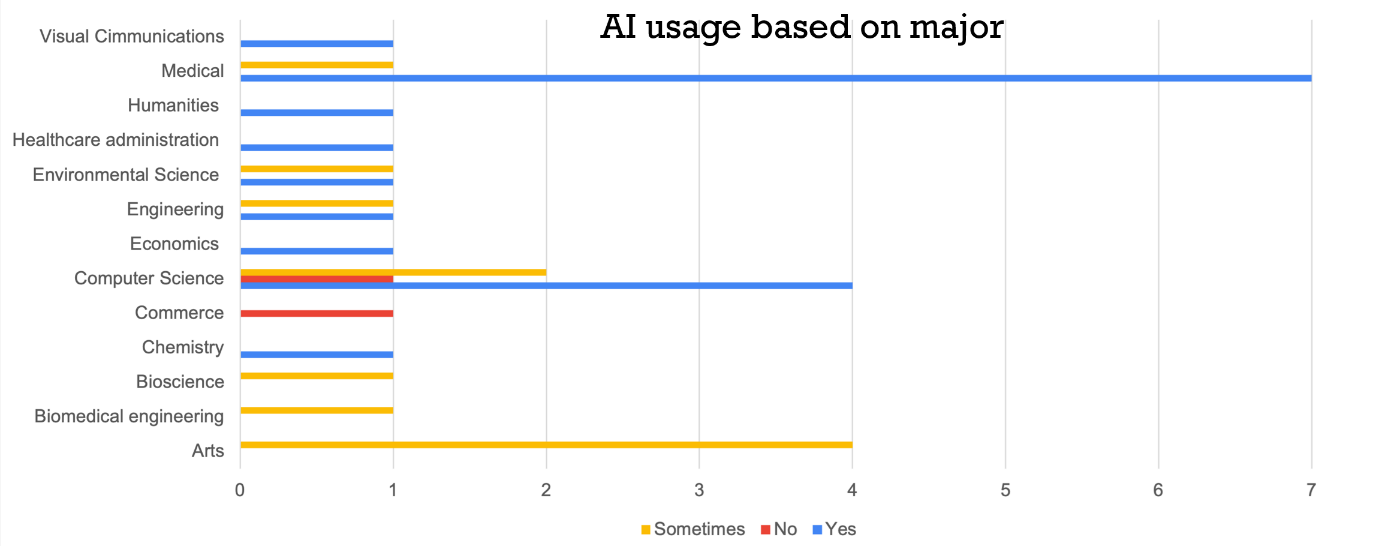


Figure 6: Survey Response - AI Majors

This graph also gives a rough idea of the most common study uses for AI

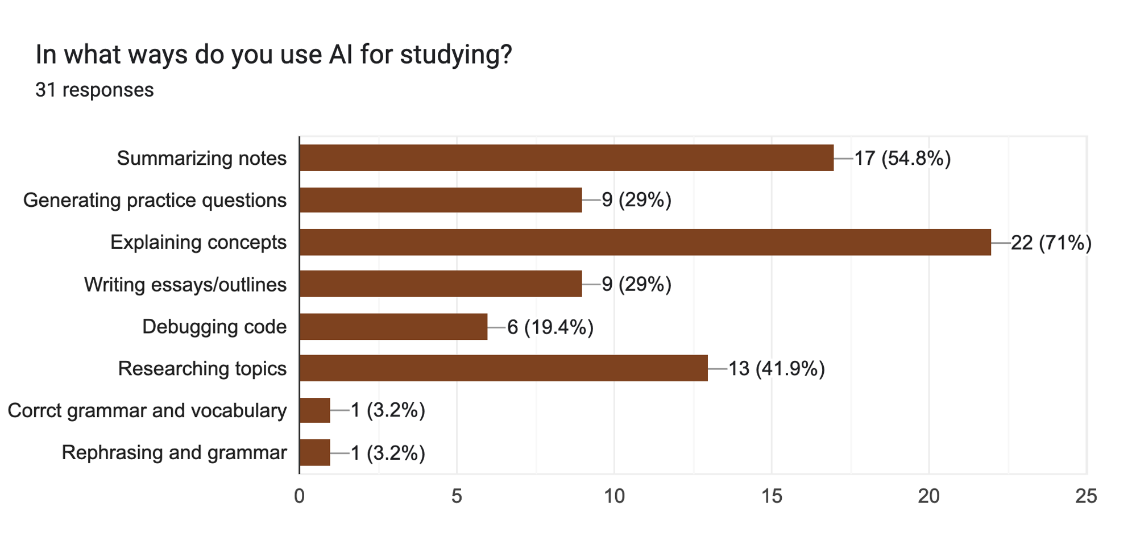


Figure 7: Survey Response - AI Usage

Lastly, the ways in which AI are utilised.

## 2.3 User Problems

From the research conducted, we can determine that the primary issues for university students regarding AI study are as follows:

* Fragmented ecosystems – The switch between platforms to accomplish various tasks leads to a lack of focus.
* Limited feedback – Users not understanding why they are incorrect on feedback leads to poor retention and results.
* Low motivation or discipline – The absence of a gamified system greatly reduces user motivation
* Accessibility limitations – The inability to upload handwritten notes or spoken content is a limiting factor, particularly for disabled users.

# 3. Technical Problem: The Setting

## 3.1 Reasons for Building This Application

This project was created with the core aim of improving modern day studying at a third-level. The idea was discussed between the group which lead to research of existing services, and ultimately lead to the discussion of what they do well, and what could be done better by these services.

As aforementioned, many students use these digital learning tools in a fragmented manner, leaving them without any personalised, adaptive feedback and creating a disjointed workflow. Research based on multitasking and cognitive switching as mentioned by *Rubinstein (2001)* showcased that frequent task switching could reduce productivity by up to 40%, this highlights that the workflow should be put into one coherent interface.

With the lack of centralisation already playing a factor in student studying retention (*particularly affecting students which may have attention disorders*), and the fact that many of these services also lack any sort of gamification; leads to a vast drop in user engagement and studying habits *(Hamari, 2014)*.

Our surveying *(*[*See figure 7*](#_Figure_7:_Survey)*)* showcased that there are a wide range of services which users utilise; however, the majority are based on services such as concept explanation that are typically represented by the likes of ChatGPT. This acts as more of a generic assistant than targeted learning companion, as there is a lack of contextual awareness of the learner’s progress; a study conducted by *Luckin (2016)* brought further light to this matter. AI Study Buddy aims to further resolve these issues with the use of knowledge-gap detection and progress analytics, which make the system far more responsive to a student’s individual learning performance rather than typical generated content. Lastly, as the service is targeted at students, one of the main pros is that it will remain free rather than behind any sort of paywall.

## 3.2 Core Technical Problem

At its core, the project faces a multidimensional technical challenge, with the integration of Natural Language Processing (NLP), Computer Vision (OCR), Speech Recognition, and the use of behavioural analytics all combining into a seamless, low latency environment. Every unique component serves its own purpose; however, they must be able to operate synchronously in order to provide a responsive user experience. These issues can be listed as the following:

* Multimodal Data Handling – Being able to process varied input formats e.g., typed text, handwriting, and voice accurately.
* Context Retention – Retaining an understanding of a user’s notes and quiz data to be able to provide personalised responses.
* Latency and Scalability – Ensuring that the responses by the AI are swift, and that they do not cause great strain to server resources whilst more users are active.
* Data management – The data must ensure privacy, consent, and compliance with GDPR standards are met.

## 3.3 Existing Systems Review

Whilst there are quite a few academic tools aimed at enhancing student productivity, a lot of them are lacking when it comes to full integration of multimodal AI processing, personalised learning, and gamification all within one a single platform.

|  |  |  |
| --- | --- | --- |
| **Existing System** | **Primary Features** | **Limitation vs Study Buddy** |
| Quizlet | Flashcards, Study guides | Likely the greatest competitor, but withhold a lot of their content behind a paywall. |
| Gizmo.ai | Flashcards, gamified system | Lacks notetaking, & quiz generation. |
| Coral.ai | Summaries, quizzes | Paywall, no mock assessments or gamification. |
| Pomofocus | Productivity and focus tracking | Is an isolated service which contributes to the typical fragmented services. |

Table 1: Existing System Comparison



Figure 8: Quizlet Paywall

It is understandable that as a service grows it’s userbase, it will have a lot of processing to do which incurs higher costs; however, there are alternatives to forced paywalls, such as the use of ads with the option to upgrade to remove said ads. With that being said, they are worth observing for interfacing and their flashcard formatting; although they lack notetaking.

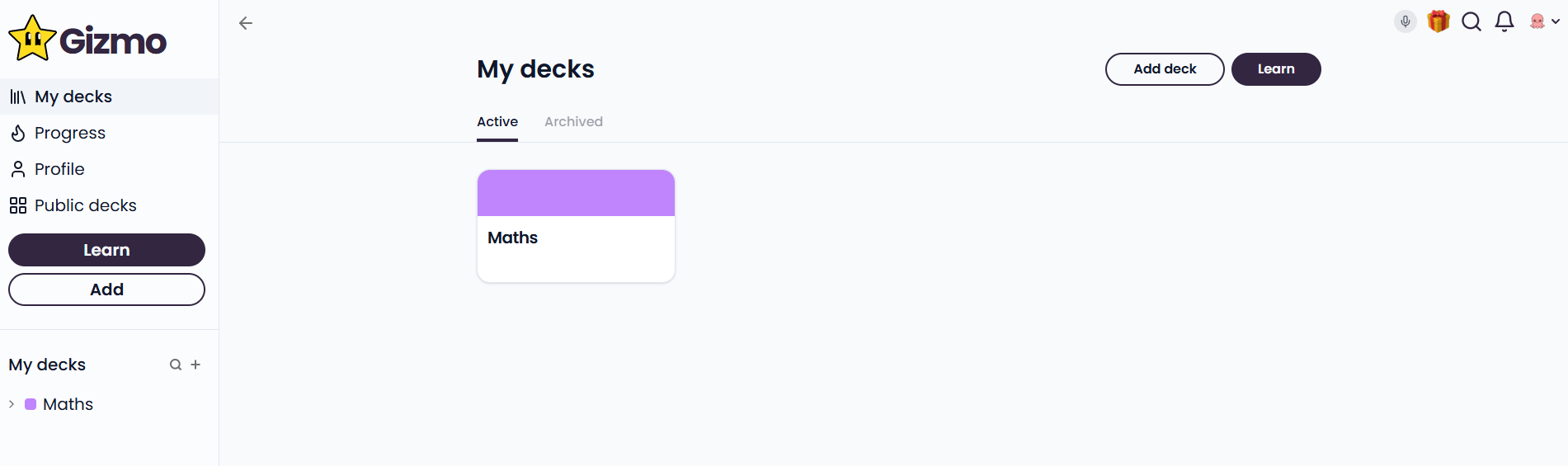


Figure 9: Gizmo.AI Homepage

Gizmo is one of the top results at the time of searching for competitors; as such, it will be the main service to criticise from both a functional and UI perspective. The page is filled with a lot of blank areas, particularly within the main content section. It also doesn’t present an immediate expansion for topics here, leaving the page feeling lackluster and unfinished. Applying *Nielsens Heuristics #8* *(1994)*, mentions that minimalist design is good, but the design here leans more to sparsity as there is a lack of visual balance, making the interface feel barren. The lack of an action such as “Create a new deck” in a more fitting place along with sidebar data overlapping main content, also go against *Jakobs Law (2000).*

Alongside these UI criticisms, the application is lacking in services such as notetaking, quizzing, and focus aspects such as a pomodoro timer. This requires users to utilise additional services, and further breaks focusing.

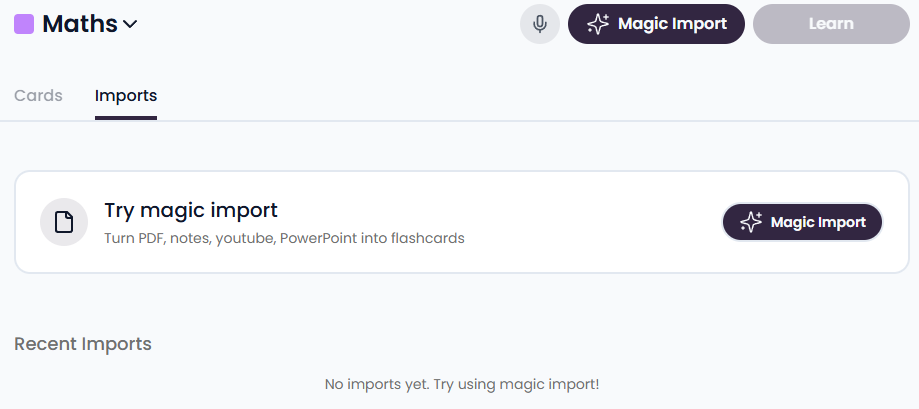


Figure 10: Gizmo.AI Upload Section

Duplicate buttons on the interface are also a poor design choice and are also against *Jakobs Law* *(2000).* Although, it is worth noting that their import does have a good variation of choices, and the page is fast to load, some things to consider within AI Study Buddy’s development.

## 3.4 Technical and Educational Context

Whilst AI summarisation and tutoring are available in quite a vast manner, few of the systems provide multimodal learning pathways which are adaptive to the student’s behaviours. Following the *Microsoft Research Guidelines (Amershi et al., 2019)*, an effective AI system must follow three primary guidelines:

* Have their capabilities be clear
* Support for efficient corrections
* Be able to adapt to any changes in a users’ goals.

AI Study Buddy integrates these principles by displaying the AI’s processing status, allowing the user to edit their summaries, and the ability to adapt their quizzes based on previous performances.

## 3.5 Summary of the Technical Problem

The technical problem can ultimately be summarised by listing the platform and its requirements. The ability to create a unified system with multiple AI capabilities (NLP, OCR, and speech) into one architecture. Retaining usability and cognitive efficiency whilst adhering to UX laws and heuristics. Providing adaptive and transparent feedback, which scales as the user’s interaction increases. The solutions for actually implementing these challenges lies within the modular backend structure and design principles; this is elaborated on within the next section.

# 4. Technical Solution: The Plot

## 4.1 System Overview

AI Study Buddy is a web-based application which is designed to optimise a student’s learning patterns, behaviours, and studying habits. It has the ability to accept various input types whether they are digital files, or handwritten scans and can process this data into a personalised, interactive studying tool.

It also has intelligent features such as an AI-driven chat tutor, automatic summarisation, flashcard and quiz generation, mock assessments, and an adaptive pomodoro timer. With the gamification system planned, the user will also retain motivation through badges, achievements, streaks, and performance analytics; all whilst the knowledge gap detector identifies any wear areas within the users studying to utilise targeted improvement.

With the integration of all these features into one cohesive platform, the aforementioned fragmentation of services is nullified, resolving the issues that students face when switching between various apps for their studying habits.

A diagram of a learning cycle

AI-generated content may be incorrect.

Figure 11: Learning Cycle

## 4.2 System Architecture Design

The system architecture is based on a modular cloud architecture hosted on Amazon Web Services (AWS): Each service is then containerised and can be independently scaled to improve reliability and maintainability.

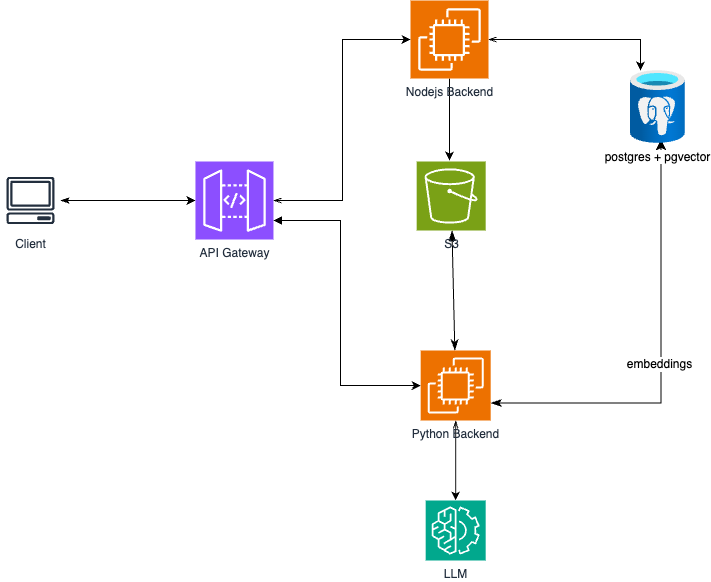
****

Figure 12: System Architecture

Each technology was selected to optimise our performance, and scalability.

|  |  |
| --- | --- |
| React.js | Fast, component driven front-end which allows dynamic and responsive UI development. |
| Node.js | Enables asynchronous operations to handle user auth, concurrent uploads, and session requests. |
| Python (Flask) | Supports lightweight, reliable AI service endpoints, highly compatible which allows it to integrate AI pipelines e.g., RAG workflows. |
| PostgreSQL & pgvector | Combines the relational storage with semantic search for embeddings, which allows the retrieval of contextually relevant content. |
| AWS (Elastic Beanstalk, S3, RDS) | Provides a flexible managed infrastructure with automatic scaling, and security. |

Table 2: Technology Justification

### 4.2.1 Front-End: React Client

With the front end being developed using React, the mobile-first interface is highly responsive; allowing students to utilise features without extensive page loading. It’s designed to be visually clean, consistent, and minimalistic; prioritising usability over complexity.

Presently, the interface is undergoing many changes as testing evolves and feedback is given, the base version is being utilised to perform development with UI changes to be applied later. To ensure accessibility and usability in the future, the design will follow Nielsen’s 10 Usability Heuristics *(Nielsen, 1994)* and where possible, apply some of the fundamental UX laws such as Jakob’s Law, Hick’s Law, Fitt’s Law, and Gestalt Principles.

The base design has undergone criticism, and additional versions have been created with the use of sketches, and Figma. Evolution of the interface is always occurring as feedback increases.

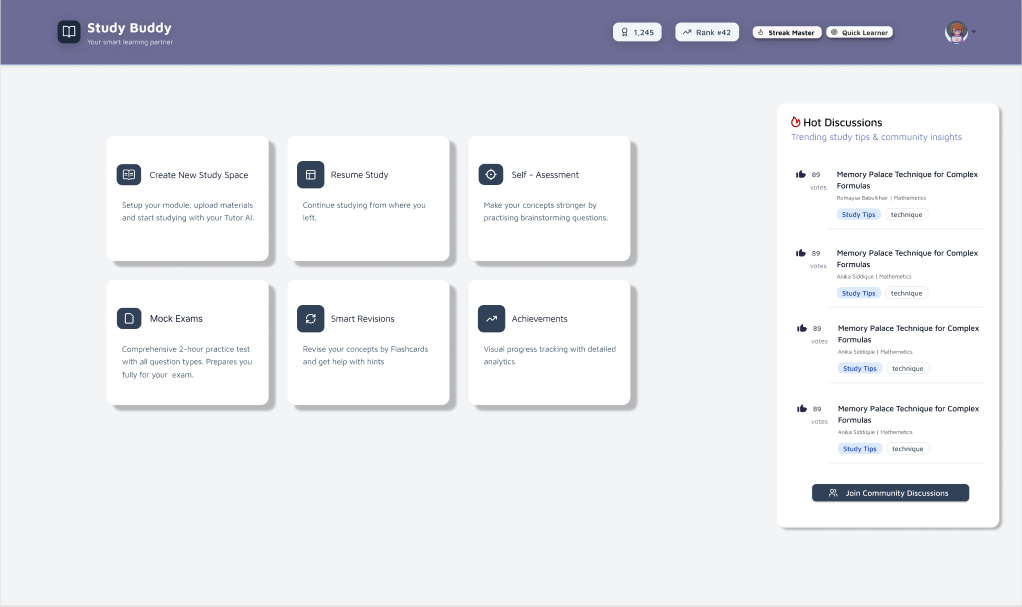


Figure 13: Dashboard Version 1.0 (Logged in) [Figma]

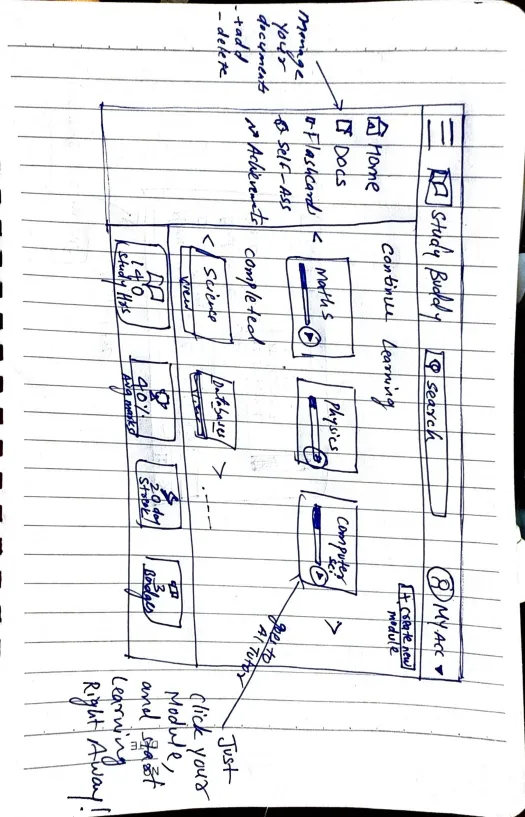


Figure 14: Sketch Update

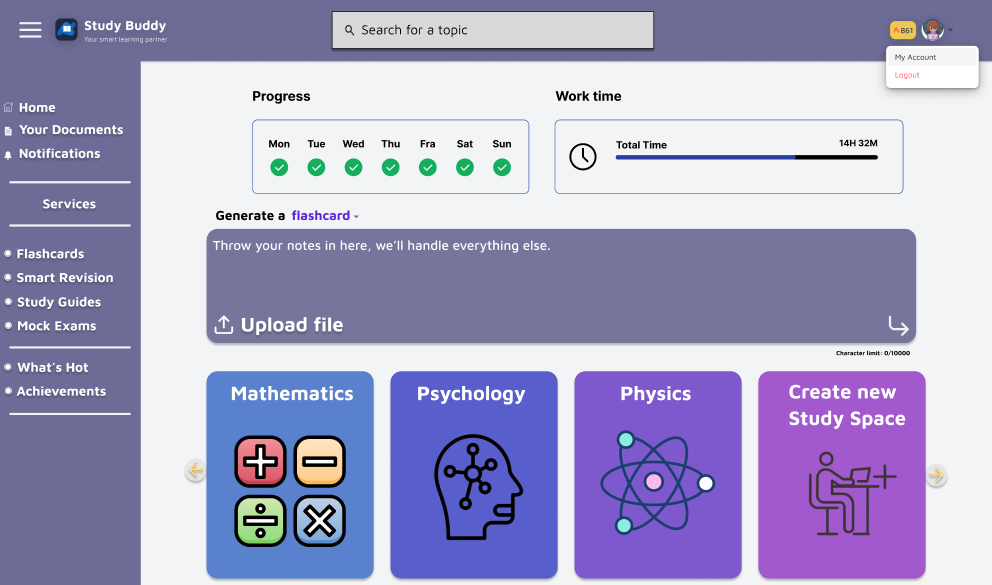


Figure 15: Potential Updated Dashboard (Logged in) [Figma]

### 4.3. AI Chat Tutor and Data Flow

We implemented **pgvector** as the dedicated vector database solution. This choice was made for efficient system management, as we were already utilising PostgreSQL, allowing us to avoid managing separate, specialised databases.

With the AI chat tutor being the central feature of the system, it is best to showcase how it utilises a Retrieval-Augmented Generation (RAG) pipeline to ensure that all of the responses remain grounded in a student’s uploaded material.

The sequence for uploaded content is as follows:

* **Text Extraction:** Retrieving the raw text content from the file.
* **Chunking:** Dividing the text into smaller, manageable sections or "chunks."
* **Embedding Generation:** Using a sentence transformer model, each chunk is converted into a numerical vector (embedding) and stored in pgvector along with its unique file ID. This indexed data forms the knowledge base for the AI tutor.

### 4.3.1 Retrieval and Response Generation

When the user asks a question, the front-end (React) calls a Flask API endpoint. The API

encodes the user's question and performs an index search against the pgvector database to

retrieve the top k most relevant data chunks (context) from the notes.

Finally, the original user question and the retrieved context are passed to a powerful Large Language Model (LLM)—such as Gemini —with a specific instruction prompt: "Answer this question using the provided notes, and supplement with basic general knowledge if needed to connect ideas." This workflow ensures responses are accurate, relevant to the user's material, and cohesive.

### 4.3.2 Future Work: Decentralised AI Strategy

Our long-term strategy focuses on migrating the core inference capabilities to self-hosted,

open-source Large Language Models (LLMs) to ensure cost efficiency, and absolute data

privacy.

**Initial API Integration Challenges**

During initial development, we integrated and extensively tested several major commercial

LLM APIs, including OpenAI and Google Gemini. While these platforms consistently

provided high-quality, intelligent responses necessary for the tutor application, a significant,

recurring challenge was encountered:

* Cost and Sustainability: The token-based consumption model across all commercial providers led to high, unpredictable operational costs. Even with efficient RAG implementation, the rapid consumption of API credits made the platform financially unsustainable for broad scaling and high user volume.
* Data Privacy Concerns: Utilising external APIs inherently means sensitive user data (queries and proprietary notes used as context), (*Cartwright.O, 2024)* is transmitted off-premise, introducing data governance and privacy risks we aim to mitigate completely.

### 4.3.2 Pomodoro Timer

The pomodoro timer will utilise the analytics from a user’s previous sessions to optimise their study durations. The study sessions can adjust dynamically based on how complex the content is, and how engaged the user remains. After any intense topics are covered, a longer break would be applied to reduce any cognitive fatigue. The historical learning patterns can also be used to predict focus windows.

This mechanism is supported by *Cognitive Load Theory (Sweller, 1988) and Miller’s Law (1956),* with the effort to balance mental fatigue and retain concentration.

### 4.3.3 Flashcards, Quizzes, and Mock Exams

The flashcards utilise key terms, definitions, and examples for active recall prompts. A user can flag each card as mastered or unmastered which will then alter future quizzes.

Quizzes are short, topic-based (currently MCQs) which are designed for assessment, whereas the mock exams are more focused on full length tests with variations of questions. The analytics of this data is crucial, as weaknesses are identified, study efficiency is monitored, the user can be assigned topics for revision. With this loop of practice, analysis, and adaptation, long-term memory should improve over time.

### 4.3.4 Achievements, Analytics, and User Notes

The Achievements and analytics system is used to enhance engagement, with visuals of tracked learning hours, quiz accuracy, and streaks. Any user notes are also analysed to identify missing information or misconceptions of the original study material.

### 4.3.5 Node.js Back-end

Hosted on AWS Elastic Beanstalk, the Node.js service manages the data flow between the client and AI layers. There are many responsibilities handled such as the management of authentication, user sessions, upload validation, data transfer to S3, co-ordination of database operations through PostgreSQL, and making secure API calls to the Flask AI service.

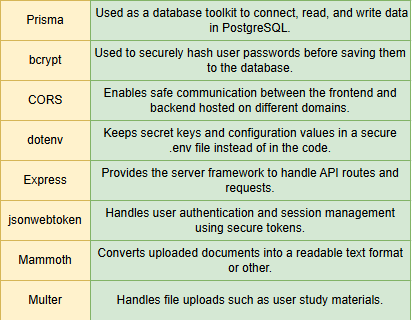


Figure 16: Back-end Libraries Utilised

## 4.4 Architectural Style

The architecture follows RESTful design principles, sending and receiving HTTP requests within the JSON format. Some of the common methods used are as follows.

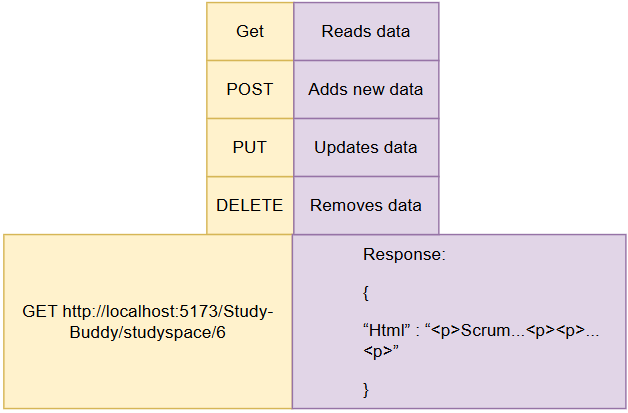


Figure 17: Common Application Methods & Example

## 4.5 Summary

AI Study Buddy combines a modernised cloud-based architecture with intelligent RAG-driven learning. With the integration of React, Node.js, Flask, and PostgreSQL + pgvector, the functionality is highly robust and scalable.

With future iterations also following the best UX practices, the system will deliver upon its goal as a user centred, privacy-conscious, and future proofed solution which helps students globally.

# 5. Evaluation: The Reviews

## 5.1 Defining Success

Success for AI Study Buddy would be determined through various metrics being met, these consist of usability, performance, and learning-impact. Alongside these metrics, there are requirements which should be met for the best results which affect students learning. As such, the application should do as follows:

* Let users upload their notes and have them be processed swiftly.
* Generate accurate, reader friendly summaries which retain ≥80% accuracy of the original document.
* Create adaptive quizzes which measure the user’s success continually.
* Keep the system responsive with less than two seconds of processing for most tasks.
* Have an overall System Usability Scale (SUS) score above 80. (This would be listed as “excellent” by *Bangor et al., (2009)*
* Showcase that user’s motivation is retained, and that study consistency is increased.

This would demonstrate that the application is supporting students learning, with the data providing both qualitative and quantitative perspectives.

## 5.2 System Evaluation

Various methods will be used to ensure the system is evaluated effectively. Both objective usage analytics, and a user-centred approach and will be used to accomplish this task.

### 5.2.1 Heuristic Evaluation

With the use of Nielsen’s 10 Usability Heuristics *(Nielsen 1994),* we can have a small group of expert reviewers who will inspect the prototypes and make note of any of the principal violations. This ensures that we follow the standards such as error prevention, and visibility of system status to name a few. Each issue noted will then be given a severity rating from 0-4 as standardised by Nielsen’s scale. This evaluation will be performed before the final larger scale user testing is done to ensure the UI is refined before being more widely tested.

### 5.2.2 Cognitive Walkthroughs and Think-Aloud

A small group of university students will then be put onto the application, and assigned realistic tasks which would typically be performed, the tasks will be monitored and the participants will speak as they go through the processes. The tasks will likely be as follows:

* Upload document(s) and have them summarised.
* Generate a quiz and complete it.
* Utilise the Pomodoro focus session.

As the participants verbalise their thought processes (think-aloud), we can document any instance where they become confused. This method is used in a lot of research, as it highlights the gaps between system logic and the users’ expectation *(Wharton et al., 1994).*

### 5.2.3 Quantitative Testing

Lastly, a broader evaluation can be performed which tracks task ratings. These task ratings will be applied to completion times, and error ratings for an objective measurement of efficiency. In the table below, we can see how the data can be measured; ‘**S**’ being success, and ‘**F**’ being failure, e.g., if the task is completed (*1*) and also finished within the preset time (*2*), the result is *S/S*.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Task 1 | Task 2 | |
|  | Task Completion / Time Taken | |  |
| User 1 | S/S | F/F | |
| User 2 | F/S | F/S | |
| User 3 | S/S | S/S | |
|  |  |  | |
| Task Score | 66% | 33% | |
| Time Score | 100% | 66% | |

Table 3: Sample Testing Data

After these tasks are run through by the testers, a questionnaire will be presented based on readability, data relevance, accessibility, consistency, and visual weight. For each of these questions, the *Likert* scale is applied with a 5-point agreement system *(Joshi, A., 2015)*. In this example, there are 9 questions which are mapped with a specific metric; whilst the score is four and above, it can be determined as a success, and anything below as a failure.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **Question Index** | Measuring Criteria (**Success**) | Measuring Criteria (**Failure**) | |
|  |  | *Calculated based on average score of related questions* | |  |
| Readability | Q1 | ≥4 | <4 | |
| Data Relevance | Q3 and Q4 | ≥4 | <4 | |
| Accessibility | Q6 | ≥4 | <4 | |
| Consistency | Q2, Q7, and Q5 | ≥4 | <4 | |
| Visual Weight | Q8 and Q9 | ≥4 | <4 | |

Table 4: Sample Quantitative Questionnaire

## 5.3 Success Criteria

|  |  |  |
| --- | --- | --- |
| **Criterion** | **Desired Metric** | **Evaluation Method** |
| Summary Accuracy | ≥ 80% relevant content | Semantic comparator |
| System Usability (SUS) | ≥ 80 | User survey |
| Task Completion Rate | ≥ 90% | Cognitive walkthrough |
| Average Task Time | < 2 minutes | System logs |
| Motivation factor | +20% increase reported | Questionnaire |
| Learning Improvement | ≥ 15% quiz score | Controlled study |

Table 5: Success Criteria

With these criteria and desired metrics, the system can be evaluated as an effective approach to improving productivity and learning outcomes. This framework also follows the ISO standards for human-centred design *(ISO 9241-210, 2019).*

## 5.4 Ethical Considerations

Lastly, the participants will consent to GDPR-compliant procedures, whereby the data is agreed to be anonymised. Accessibility will also be tested where possible and will be following the WCAG 2.1 AA guidelines for the likes of high-contrast mode.

The evaluation measures functionality, retains trust, and is inclusive; all of which are staples when deploying an application for education. *(Amerishi et al., 2019).*

# 6. Conclusion: The Plan

## 6.1 Project Management Strategy

The development of AI Study Buddy is based on a hybrid agile methodology, which combines the flexibility of Scrum with a more traditional structure of milestones. The work is typically divided into weekly sprints, which end with a deliverable prototype and a group overview of the implemented feature.

To manage this, the technologies used are as follows:

* GitHub to handle version control, issue tracking, and general feedback.
* Notion for the feature listings, and status of tasks.
* Discord for meetings and general group conversations.
* Teams for more extensive meetings.

With regular meetings (both in-person and remotely), it is assured that progress is maintained throughout the lifespan of the project.

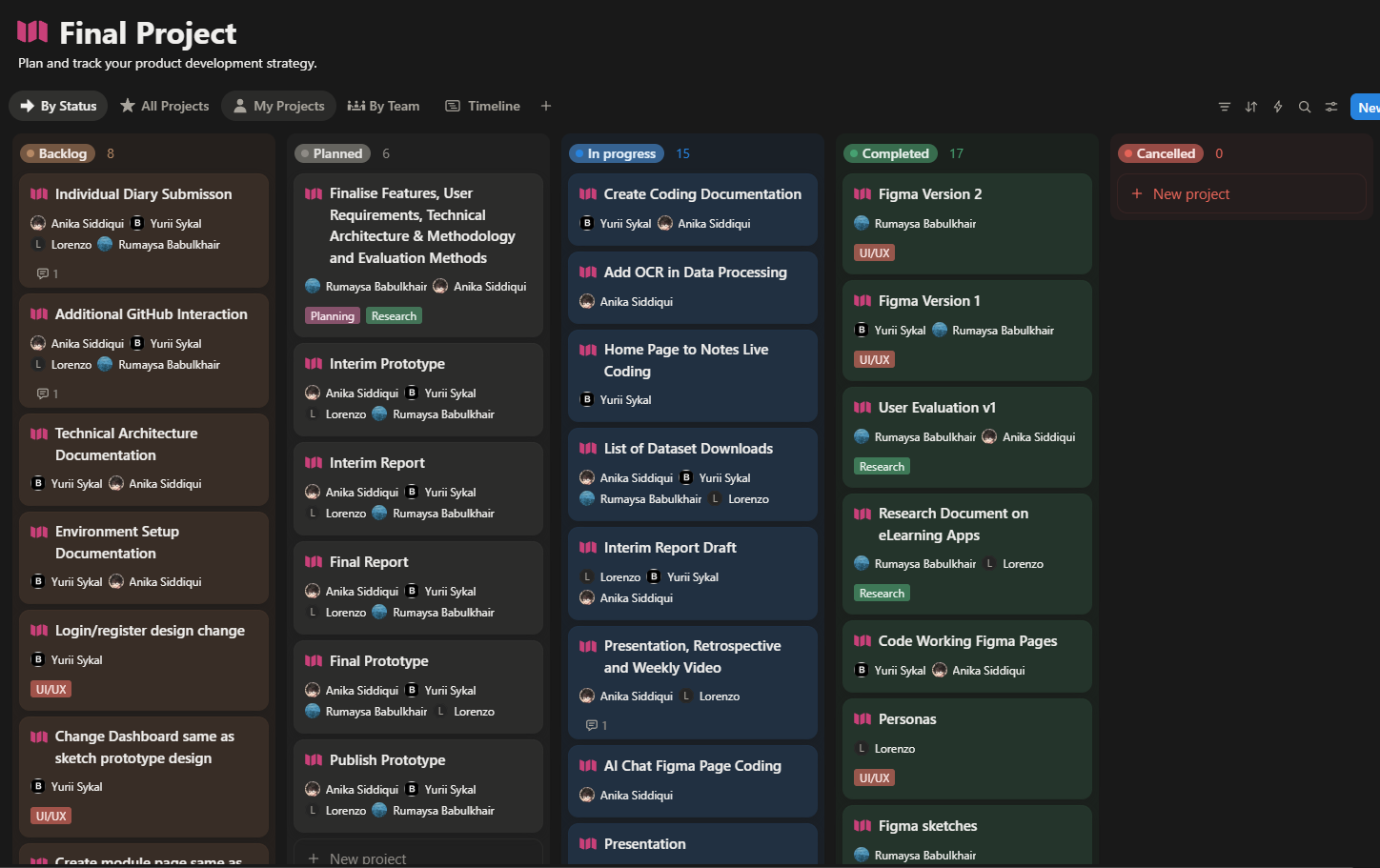


Figure 18: Notion Task Board

### 6.1.1 Team Meetings

Due to working conditions of various group members, an adaptive meeting schedule was created. The staple days of Tuesday and Friday were selected with an adaptable two to three additional days being selected, depending on the workload, and tasks assigned. These flexible days were also done to prevent redundancy of conversations. Each meeting was initiated with each group member answering the following three questions:

* What have you done since the last meeting?
* What are you working on now?
* Are there any issues with your current tasks?

Each member also has the ability to elevate any issues or ideas which needed to be discussed within the team. The Friday meeting is designated as in-person to ensure the group remains in good standing, and to plan out the next sprint.

## 6.2 Ongoing Challenges

* Multimodal Integration: Ensuring that the text, audio, and images features can be integrated to our service successfully and can be processed within a reasonable time.
* Model performance and Cost: The use of AI API’s means that quite a large cost can arise after processing large documents for summaries.
* Data Privacy: Maintaining GDPR compliance with user data.
* User Engagement: Having a future gamification system which is motivating without being overly distracting is in the works.
* Scalability: Ensuring we know how the system handles currently, and how it would be upgraded, should the concurrent sessions increase.

These challenges also have a planned solution, e.g., Model cost is handled by using a lesser model for testing purposes, which can then be upgraded for user testing or demonstrations.

## 6.3 Time Management and Future Priorities

The remaining development time will likely be dedicated to the following:

* Enhancing technical robustness with the use of refactoring and general performance optimisation.
* Additional features, which are not yet implemented such as the gamified system are yet to be developed.
* Run the LLM on our own manage infrastructure to eliminate token costs.
* Refining UI which will be done from additional feedback from experts and general user testing.
* Cornell method integration to create a structure for uploaded content and note taking.
* Full evaluation cycle, where the results are analysed, and the final report can be prepared.

## 6.4 Expected Outcome

Upon completion, AI Study Buddy should serve as an excellent service which centralises all of a student’s studying needs into one place. It should increase productivity, and motivation which in turn increase the students’ learning capabilities. The success will be determined by results in various user tests, ideally showing an improvement in users results in a quiz or flashcards.

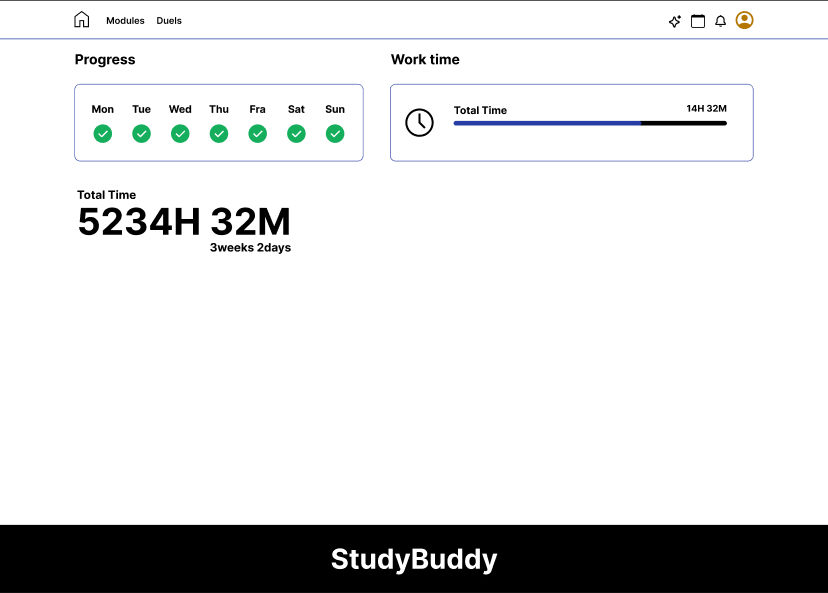
# 7. References and Key Resources

* List of resources (software, papers, tutorials, books, stats, business indicators)
* *Amazon Web Services S3 (2025): Retrieved from:* [*https://docs.aws.amazon.com/s3/*](https://docs.aws.amazon.com/s3/)
* *Node JS (2025): Retrieved from:* [*https://nodejs.org/docs/latest/api/*](https://nodejs.org/docs/latest/api/)
* *Python (2025): Retrieved from:* [*https://docs.python.org/3/*](https://docs.python.org/3/)
* *PostgreSQL (2025): Retrieved from:* [*https://www.postgresql.org/docs/*](https://www.postgresql.org/docs/)
* *OECD. Digital Education Outlook (2023): Towards an Effective Digital Education Ecosystem. Retrieved from:* [*https://www.oecd.org/content/dam/oecd/en/publications/reports/2023/12/oecd-digital-education-outlook-2023\_c827b81a/c74f03de-en.pdf*](https://www.oecd.org/content/dam/oecd/en/publications/reports/2023/12/oecd-digital-education-outlook-2023_c827b81a/c74f03de-en.pdf)
* *HEA. Higher Education Authority (2025): Student data. Retrieved from:* [*https://hea.ie/statistics/data-for-download-and-visualisations/access-our-data/Access%20our%20Data%20-%20Students/*](https://hea.ie/statistics/data-for-download-and-visualisations/access-our-data/Access%20our%20Data%20-%20Students/)
* *Deci, E.L, & Ryan, R.M. (2000): The “What” and “Why” of Goal Pursuits: Human Needs and the Self-Determination of Behavior. Retrieved from:* [*https://selfdeterminationtheory.org/SDT/documents/2000\_DeciRyan\_PIWhatWhy.pdf*](https://selfdeterminationtheory.org/SDT/documents/2000_DeciRyan_PIWhatWhy.pdf)
* *Rubinstein, J. S, Meyer, D.E., & Evans, J.E. (2001): Executive Control of Cognitive Processes in Task Switching. Retrieved from:* [*https://www.researchgate.net/publication/11827832\_Executive\_Control\_of\_Cognitive\_Processes\_in\_Task\_Switching*](https://www.researchgate.net/publication/11827832_Executive_Control_of_Cognitive_Processes_in_Task_Switching)
* *Hamari, Juho., Kolvisto, J., Sarsa, H. (2014): Does Gamification Work? Retrieved from:* [*https://ieeexplore.ieee.org/document/6758978*](https://ieeexplore.ieee.org/document/6758978)
* *Luckin, Rose., Holmes, W. (2016): An Argument for AI in Education. Retrieved from:* [*https://www.researchgate.net/publication/299561597\_Intelligence\_Unleashed\_An\_argument\_for\_AI\_in\_Education*](https://www.researchgate.net/publication/299561597_Intelligence_Unleashed_An_argument_for_AI_in_Education)
* *Nielsen, Jakob. (1994): Nielsen’s Heuristics. Retrieved from:* [*https://thedecisionlab.com/reference-guide/design/nielsens-heuristics*](https://thedecisionlab.com/reference-guide/design/nielsens-heuristics)
* *Nielsen, Jakob (2000): Jakob’s Law. Retrieved from:* [*https://lawsofux.com/jakobs-law/*](https://lawsofux.com/jakobs-law/)
* *Amershi, S et al. (2019): Guidelines for Human-AI Interaction. Retrieved from:* [*https://www.microsoft.com/en-us/research/publication/guidelines-for-human-ai-interaction/*](https://www.microsoft.com/en-us/research/publication/guidelines-for-human-ai-interaction/)
* *Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean. Retrieved from:* [*https://dl.acm.org/doi/10.5555/2835587.2835589*](https://dl.acm.org/doi/10.5555/2835587.2835589)
* *Wharton, C., Rieman, J., Lewis, C. (1994). The Cognitive Walkthrough Method. Retrieved from:* [*https://www.researchgate.net/publication/220302514\_State\_of\_the\_Art\_on\_the\_Cognitive\_Walkthrough\_Method\_Its\_Variants\_and\_Evolutions*](https://www.researchgate.net/publication/220302514_State_of_the_Art_on_the_Cognitive_Walkthrough_Method_Its_Variants_and_Evolutions)
* *Joshi, A. (2015). Likert Scale: Explored and Explained. Retrieved from:* [*https://www.researchgate.net/publication/276394797\_Likert\_Scale\_Explored\_and\_Explained*](https://www.researchgate.net/publication/276394797_Likert_Scale_Explored_and_Explained)
* *ISO 9421-210 (2019) Human-Centred Design. Retrieved from:* [*https://www.iso.org/standard/77520.html*](https://www.iso.org/standard/77520.html)
* *Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning. Cognitive Science. Retrieved from:* [*https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1202\_4*](https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1202_4)
* *Miller, G.A. (1956). The Magical Number Seven, Plus or Minus Two. Psychological Review. Retrieved from:* [*https://psycnet.apa.org/record/1957-02914-001*](https://psycnet.apa.org/record/1957-02914-001)
* *Cartwright, O. (2024). Evaluating Privacy Compliance in Commercial LLM’s. Retrieved from:* <https://www.researchsquare.com/article/rs-4792047/v1>

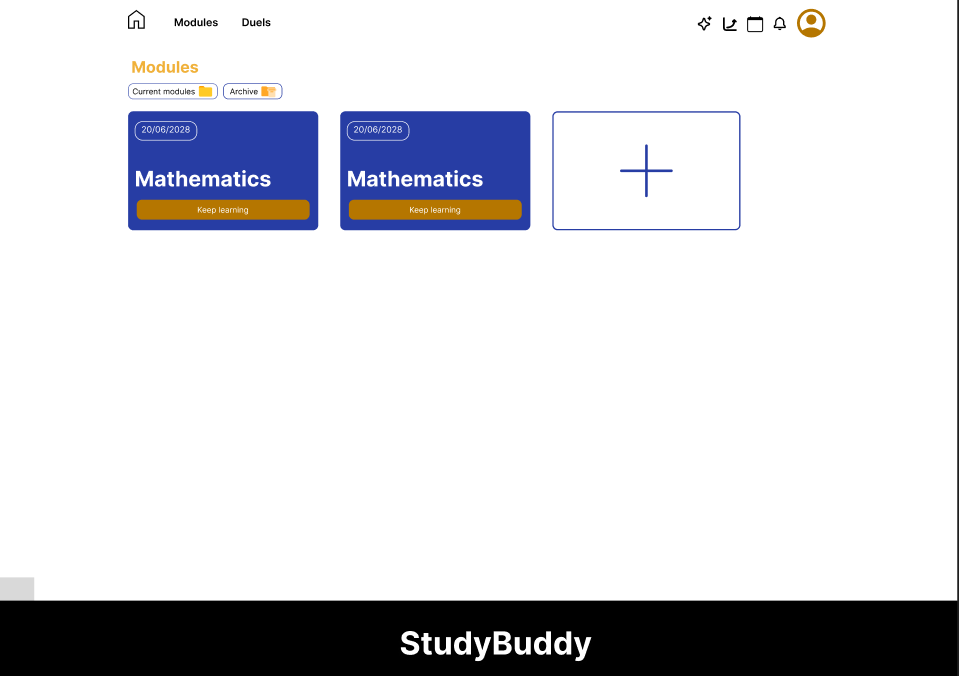
# 8. Appendix

## 8.1 Appendix A – Front-end Prototypes

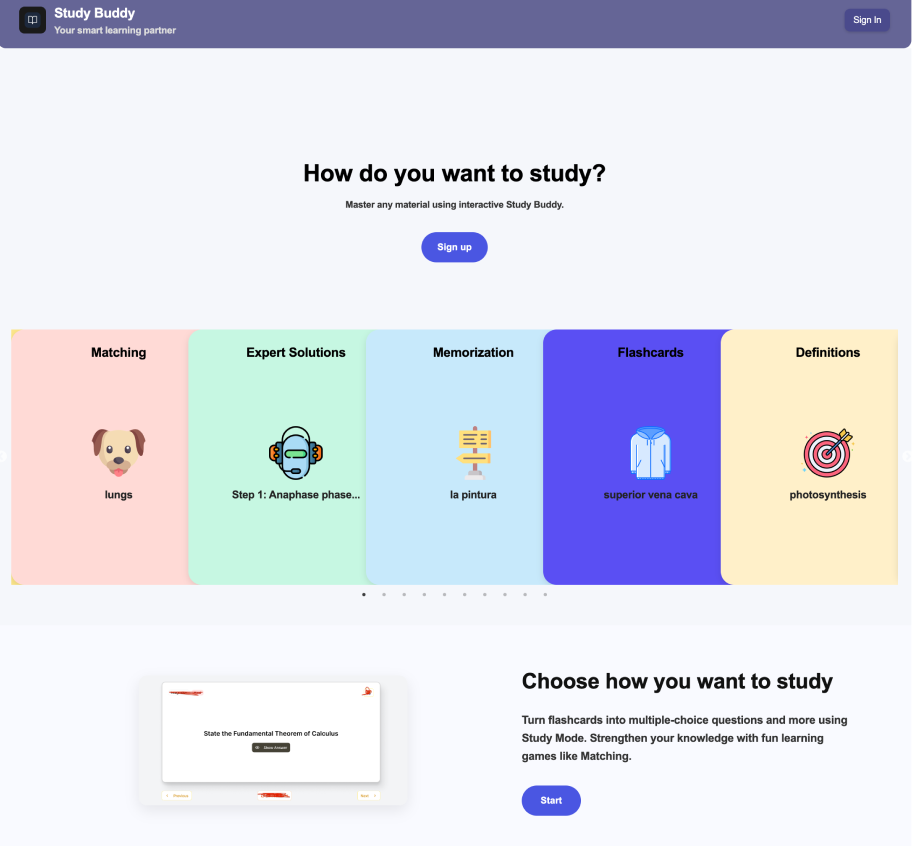
### 8.1.1 Initial Dashboard Prototype



### 8.1.2 Initial Module Selection Prototype



### 8.1.3 Dashboard Update



### 8.1.4 Module Selection Update

