

# 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](#_7pkkejaermmg)

[Table of Figures 2](#_r9b4l4ekotmc)

[List of Tables 2](#_g1ybuv19gnmb)

[1. Introduction 3](#_sgkl1brvjrfr)

[2. User Scenario: The Characters 3](#_kqnmzia841hp)

[2.1](#_kee3fi82ey8f) Identifying target users 3

[2.1.1](#_dm6tpny9d2cf) Personas 4

[2.2](#_fsdki14l0bk3) Importance of target users 6

[2.2.1](#_34z7i6yglj5n) Survey 6

[2.3 User Problems 7](#_hhw6038yd10d)

[3. Technical Problem: The Setting 7](#_r8lx3ny1pih)

[3.1 Reasons for Building This Application 7](#_cjvezqc5w1q2)

[3.2 Core Technical Problem 8](#_nutq9zjyjz7p)

[3.3 Existing Systems Review 8](#_q0o8desby5v0)

[3.4 Technical and Educational Context 11](#_94fwmevbeu13)

[3.5 Summary of the Technical Problem 11](#_lqbw9c125keg)

[4. Technical Solution: The Plot (1,000 words approx.)**.** 11](#_lozq2pg8l1yg)

[5. Evaluation: The Reviews (500 words approx.) 12](#_lqof0gr9c8ve)

[6. Conclusion: The Plan (500 words approx.) 12](#_jraw407vh8ju)

[7. References and Key Resources 13](#_xsi47lzp57b)

# Table of Figures

[Figure 1: Application Logo 4](#_jfg1bmur46iz)

[Figure 2: Persona 1 - (The consistent studier) 5](#_ldgjpfktazne)

[Figure 3: Persona 2 - (The coaster) 6](#_1mbtkid0c1lh)

[Figure 4: Persona 3 - (The mature student) 6](#_lbdamqfkgg8p)

[Figure 5: Survey Response - Age Group 7](#_rq7efn57xbh8)

[Figure 6: Survey Response - AI Majors 7](#_edbvxlv8d886)

[Figure 7: Survey Response - AI Usage 8](#_lklr51j3svkf)

[Figure 9: Quizlet Paywall 10](#_6poime3i5r0c)

[Figure 10: Gizmo.AI Homepage 11](#_jshcq8ltq3up)

[Figure 11: Gizmo.AI Upload Section 11](#_exq7fdcbf745)

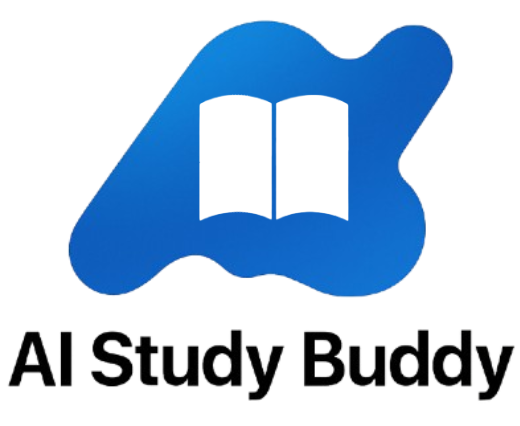
# List of Tables

[*Table 1: Existing System Comparison* 10](#_4108v5zahgvw)

# 1. Introduction

The goal of this project was to build a mobile first web application called 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

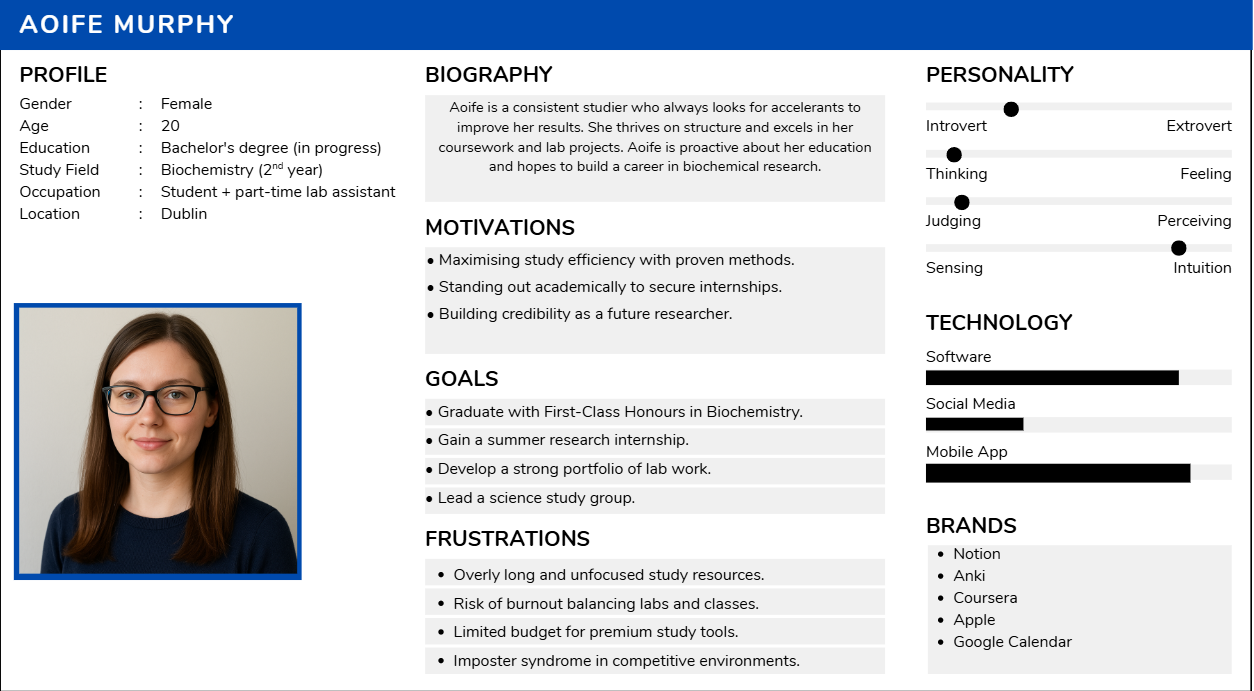
AI Study buddy is primarily aimed at third-level students who aim to improve their studying habits with the use of artificial intelligence. This groups in both undergraduate and postgraduate learners across various disciplines and age groups. We can also account for some of the potential struggles that 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).*

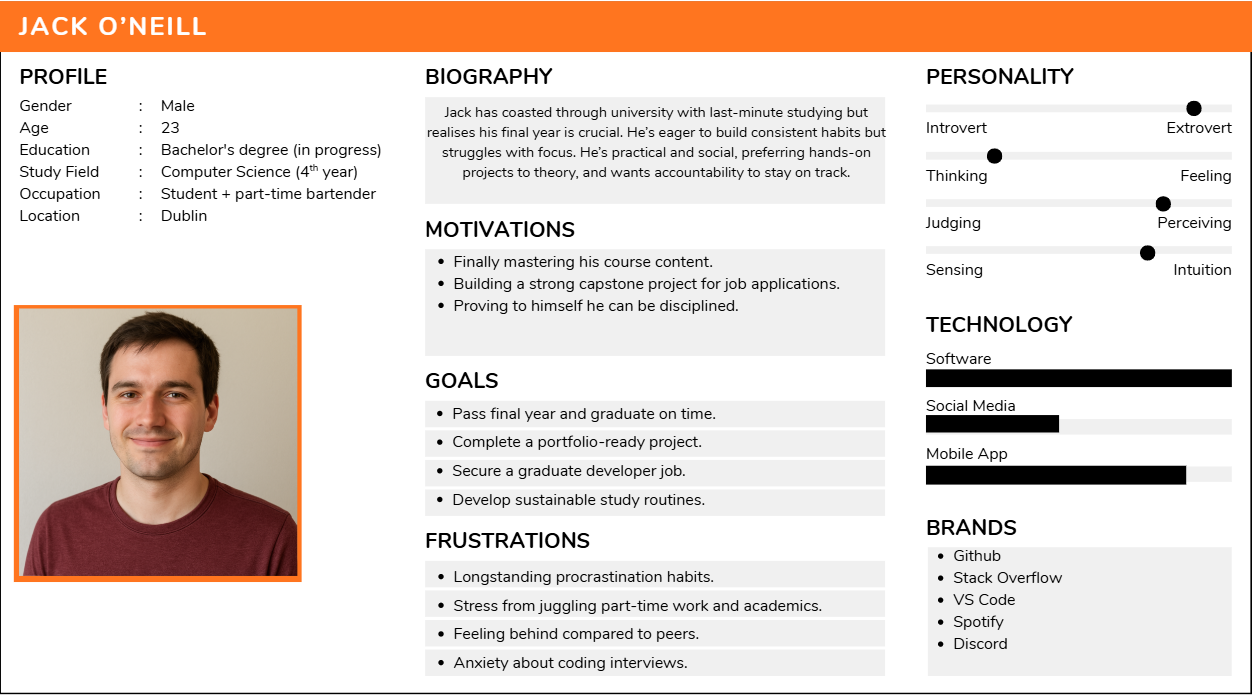
As such, AI Study Buddy targets this gap by merging AI oriented note summaries, adaptive quiz generation, and productivity tracking into one accessible platform. This is notably relevant for any students who are reliant on a personalised learning experience, yet do not have the time or resources to build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

### Personas

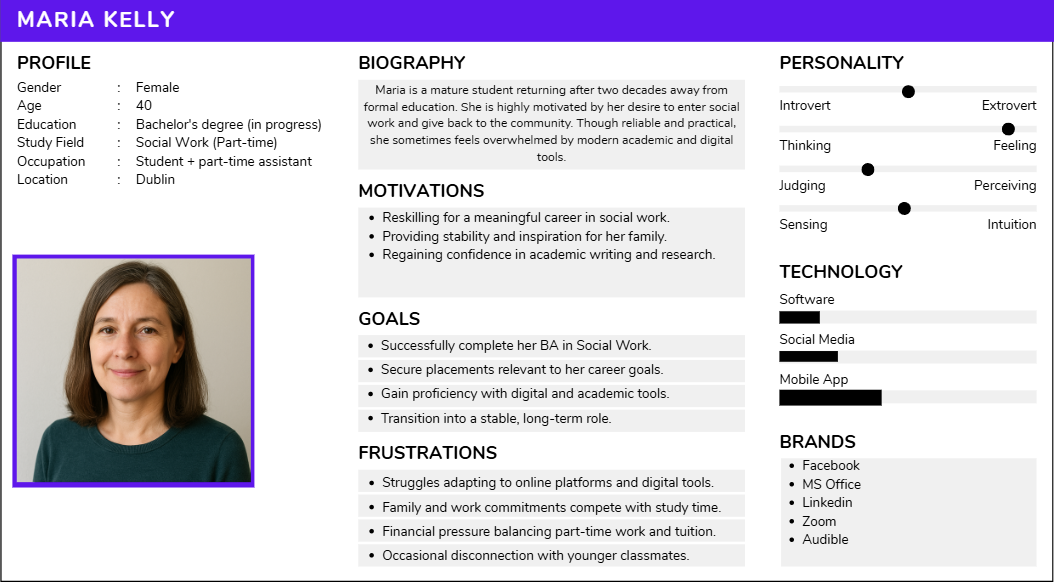
Personas are a way for us to further identify our target users; by creating semi-fictional characters, we can represent the various student types, along with their needs, experiences, behaviours, frustrations, and goals. Viewing the product through the eyes of these personas allows us to take somewhat 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)

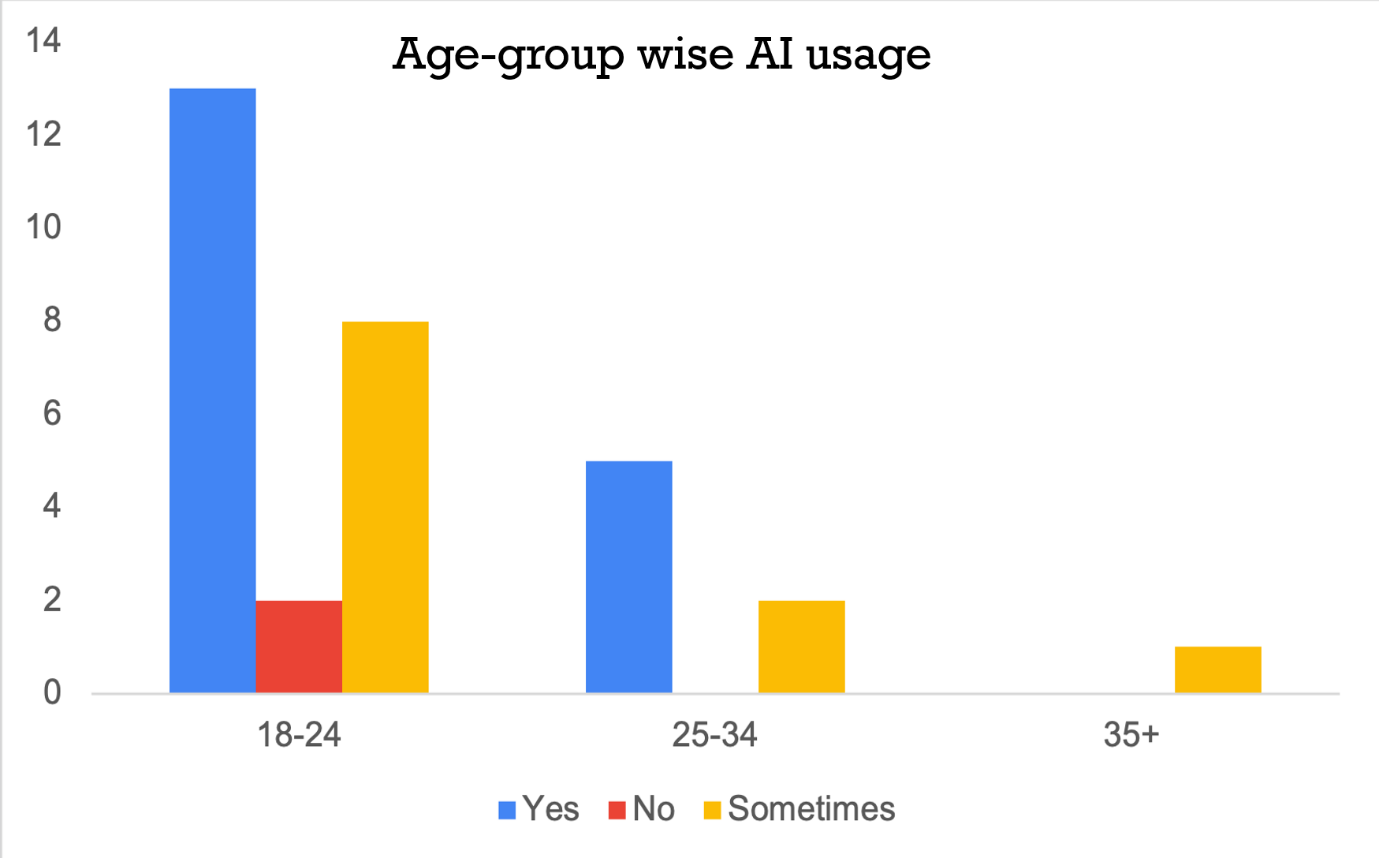
These 3 personas provide a diverse userbase for university level studies, and help us understand deeper, what some of the struggles may be; allowing us to accommodate various situations in an adaptable manner.

## 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.

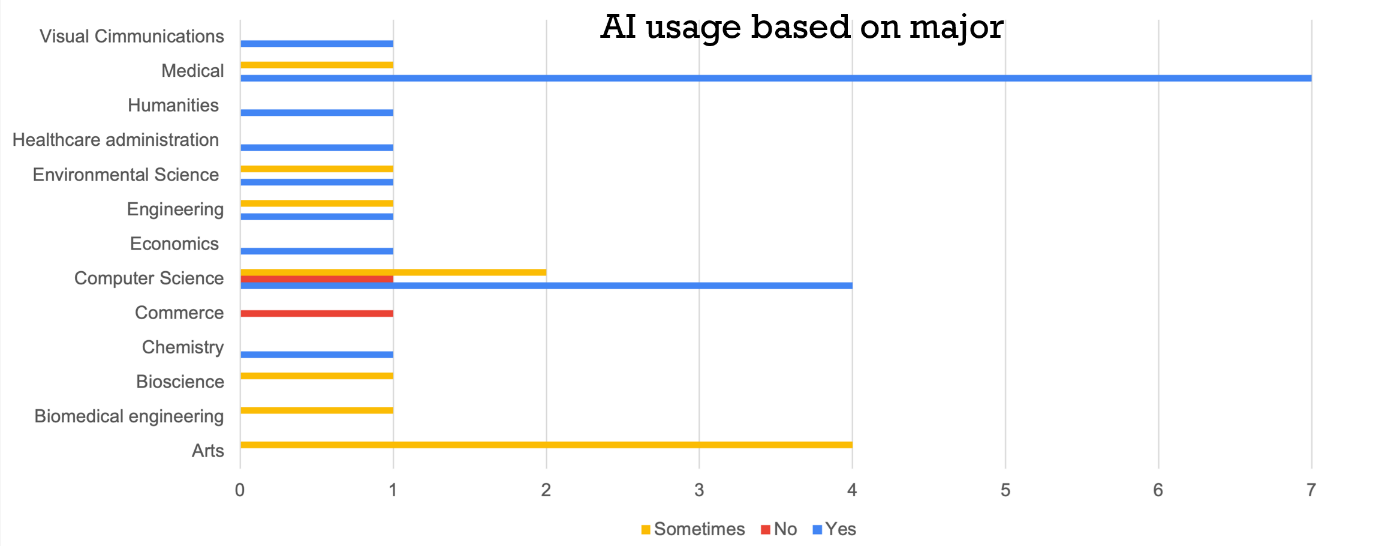
### Survey

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



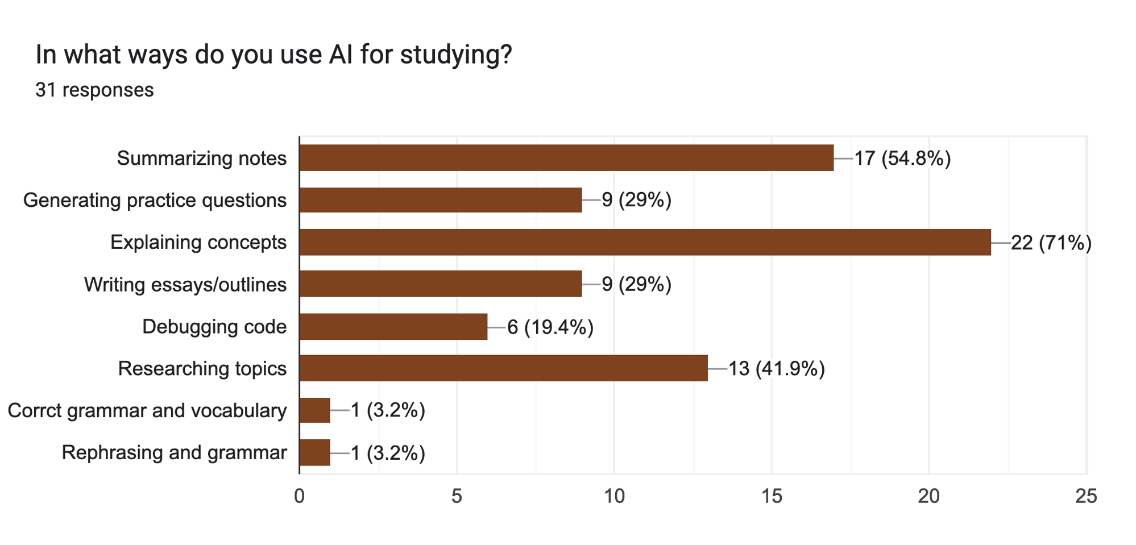
#### Figure 5: Survey Response - Age Group

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, especially affecting students which may have attention disorders, some of these services also lack any sort of gamification; leading to a vast drop in user engagement and studying habits *(Hamari, 2014)*.

Our surveying *(*[*See figure 7*](#_lklr51j3svkf)*)* 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 (2022)* 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.

INSERT DIAGRAM OF TECHNICAL COMPONENTS (FIGURE 8)

## 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.

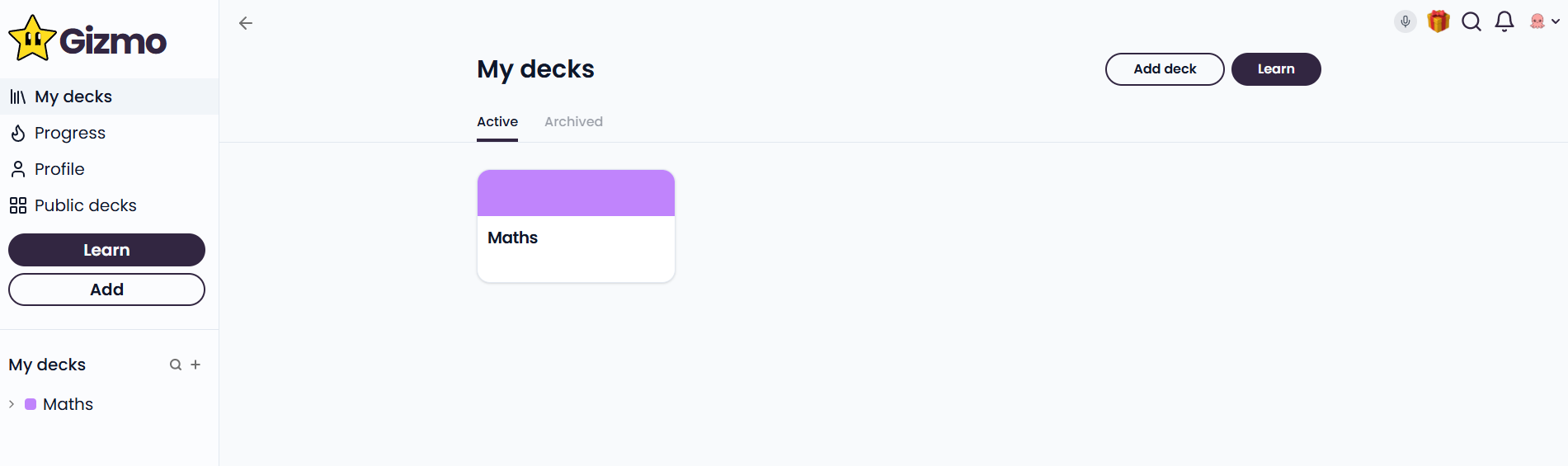
| **Existing System** | **Primary Features** | **Limitation vs Study Buddy** |
| --- | --- | --- |
| Quizlet | Flashcards, Study guides | Likely the greatest competitor, but withhold a lot of their content behind a paywall. |
| Gizmo.ai | Flashcards, gamified system | Lacks notetaking, & quiz generation |
| Pomofocus | Productivity and focus tracking | Is an isolated service which contributes to the typical fragmented services. |
| ChatGPT | Conversational AI | Lacks contextual learning progression |

##### *Table 1: Existing System Comparison*



#### Figure 9: Quizlet Paywall

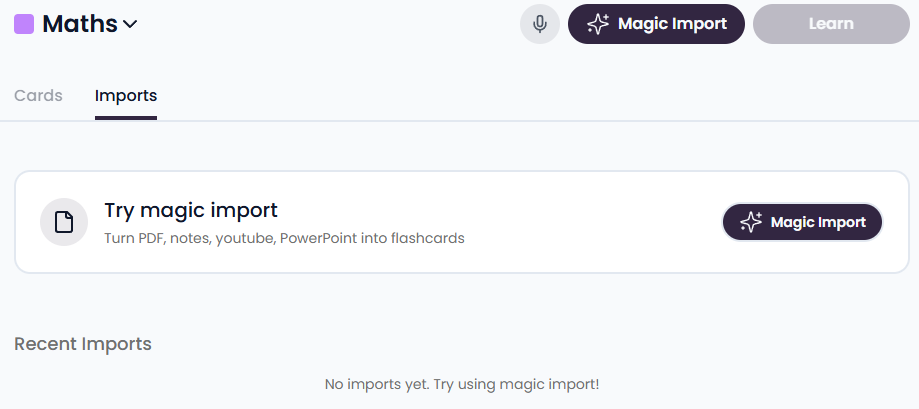
It is understandable that as a service grows it’s userbase, it will have a lot of processing to do which incurs higher costs; however, there are alternatives to forced paywalls, such as the use of ads with the option to upgrade to remove said ads. With that being said, they are worth observing for interfacing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



#### Figure 10: 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 11: 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 Studdy 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 (1,000 words approx.)**.**

## 4.1 System Overview

Study Buddy app is a web-based tool designed to help users learn more effectively from their notes. It can process all types of notes, whether they are digital files or handwritten scans. The application then creates a study chat to tutor the user from the material using AI. To enhance user engagement and motivation, the app incorporates a gamification system alongside its key practice tools. These tools include brainstorming quizzes, flashcard revisions, and mock exams for realistic practice. A built-in Pomodoro timer also helps students maintain focus and structure their study sessions.

### 4.1.1 AI chat tutor

Smart summaries, q&a from notes + external datasource, pre grenerated prompts, analyse image, tables from the material, can analyse audio, video lectures, brainstorming quizez to make concept strong, mock exams. (this is the core feature make it as detailed as u can ^-^)

### 4.1.2 Pomodoro Timer

* Based on how much time student takes to learn a particular topic ,
* complexity or simplicity of the topic, the pomodoo timer adjusts itself,
* helps in focus and time management.
* Breaks in some interval

### 4.1.3 Flashcard revisions

Flashcards with hints

Flag - mastered, unmastered concepts

### 4.1.4 Quiz and mocks

Quiz is mcq type QA generally on single concept or single module, like a brainstorming questions to make the subject strong. After the quiz it analyse your weak and strong areas. Mock test is like 2/3 hour long exam based test with all type of questions of the module-results have analytics, weak points, strong points, improvement suggestions, tips, etc. the analysed data is processed to learn the study pattern of the student(like student spent 1 hour on this comcept but still couldn’t learn the concept due to his weakness on practical learning, user is easy learner on literature based study and weak on practical based)

### 4.1.5 Achievements and analysis

What is achieved till now, analysis of study patterns, hours, badges,etc

### 4.1.6 user notes

User can take their own notes blah blah and ai can detect their gaps from the actual notes to these notes

# 4.2 System Architecture and working

The Study Buddy application has a modular design and an architecture composed of multiple components. Each component is responsible for a specific function, which allows the project to scale and remain maintainable as the number of users grows. The Study Buddy architecture can be easily deployed to any cloud platform or on-premises environment. In our case, we decided to use Amazon Web Services (AWS) because it offers a wide range of managed services that simplify deployment and maintenance.

## 4.2.1 Architecture Overview

The Study Buddy system consists of three main services.

**Python service**

It is the core of the Study Buddy application. This component is responsible for helping users by answering questions based on the documents they provide. It can also generate simple tests, flashcards, and quizzes to support personalized learning.

**NodeJs service**

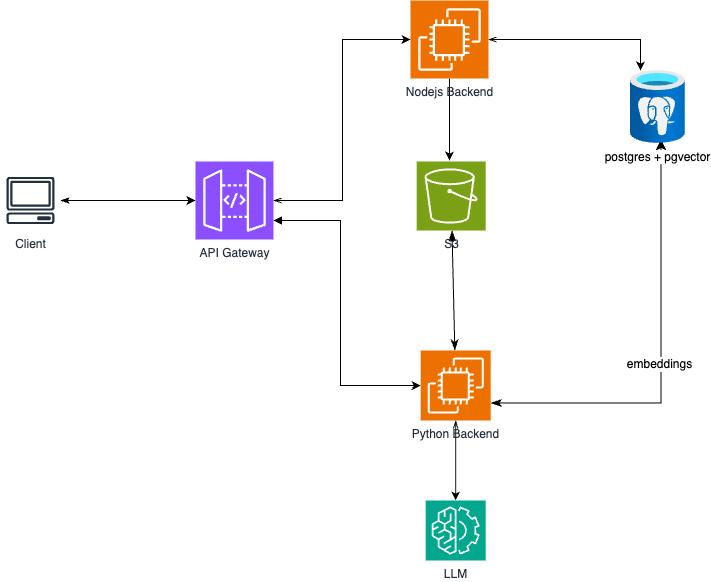
The Node.js service forms the backbone of the Study Buddy application. It is responsible for handling user management, including account creation and authentication, as well as processing and managing uploaded files.

**React client service**

It is a user interface of the Study Buddy application and is responsible for presenting information in a clear, intuitive, and easy-to-navigate way. Through its responsive design, users can seamlessly interact with the system: uploading study materials, generating flashcards and quizzes, and communicating with the AI Study Buddy through the chat for personalized learning support. Study Buddy client focuses on delivering consistent user experience across all devices and screen sizes.

## 4.2.2 System Architecture Design

The AI Study Buddy system is deployed on Amazon Web Services (AWS), and the following diagram illustrates how it operates.



**API Gateway (AWS API Gateway)**  
  
Used as the single entry point for all client interactions. It routes requests from the user web client to Node.js or Python backend, managing request validation. Without an API Gateway, the client would be required to communicate separately with multiple backend services, which would increase security risks. The Api Gateway simplifies this by providing unified access.

**Node.js Backend (AWS Elastic Beanstalk)**  
  
AWS Elastic Beanstalk Hosts Node.js backend with basic application logic, handling user sessions, file management, and communication with the database and S3 storage. Elastic Beanstalk is a powerful AWS service that automates application deployment and includes built-in load balancing to help scale the application as the number of users grows. It also provides integrated monitoring and health checks, which help maintain system stability and performance.

**Python Backend (AWS Elastic Beanstalk)**

For our Python backend service, we also chose AWS Elastic Beanstalk. The reasons for this choice are the same as those for the Node.js backend

**Storage (Amazon S3)**  
  
Amazon S3 saves uploaded user study materials and other related files. S3 is a secure and reliable solution that can handle large amounts of data. Stored files can easily be stored by the node.js backend and accessed by the python backend services whenever needed.

**PostgreSQL with pgvector Extension(Amazon RDS)**

Using Amazon RDS is easy to manage, and it automatically handles backups and maintenance. Most importantly, it supports PostgreSQL with pgvector, which helps store and search AI embeddings efficiently.

**LLM Integration**

This is the Study Buddy’s intelligent reasoning engine used for generating explanations and summaries. It powers the AI Tutor, quizzes, flashcards, and other interactive learning features.  
  
4.2.3 Architectural Style

Study Buddy follows the REST (Representational State Transfer) architecture.

That means it sends and receives data over the internet using simple web HTTP requests in JSON format.

Application uses common methods such as:

* GET – to read data

GET http://localhost:5173/Study-Buddy/studyspace/6  
Response:

{  
 “Html” : “<p>Scrum ...<p><p>...<p>”

}

* POST – to add new data
* PUT – to update data
* DELETE – to remove data

.

Explain the journey, Put the 1st diagram…then 2nd evolution to make it better.

Diagram.

Divide in 4 parts

1. Frontend – explain each component, framework used in detail and reason why and mockups
2. Backend - explain each component, framework used in detail and reason why
3. Database -
4. AI

### 4.2.3 frontend

### 4.2.4 Backend

### 4.2.5 DB

### 4.2.5 AI Integration

The system's core function is powered by a **Retrieval-Augmented Generation (RAG) architecture**, which is critical for grounding the AI tutor's responses strictly within the user's uploaded notes. The overall architecture, detailing the interaction between client, services, and data stores, is illustrated below in Figure 1.

**Data Flow and Storage:**

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

**Note Processing and Indexing:**

When a user creates a study module, the application processes the selected source files retrieved from AWS S3 storage. The pre-processing involves three main steps:

1. **Text Extraction:** Retrieving the raw text content from the file.
2. **Chunking:** Dividing the text into smaller, manageable sections or "chunks."
3. **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.

**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.

### Future Work: Decentralized 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, includingOpenAI 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:** Utilizing external APIs inherently means sensitive user data (queries and proprietary notes used as context) is transmitted off-premise, introducing data governance and privacy risks we aim to mitigate completely. <https://www.researchsquare.com/article/rs-4792047/v1>

These compromises in cost and privacy necessitated a shift in our AI deployment strategy.

#### Target Architecture: Self-Hosted Inference with Llama

The primary focus for future development is the complete decoupling from third-party APIs by implementing a self-hosted inference service based on the open-source **Llama** model or other high-performing models available on **Hugging Face**.

**Key Objectives:**

1. **Cost Neutrality:** By running the LLM directly on our managed infrastructure (or on dedicated user machines, post-optimization), we eliminate token costs, transitioning the expense from consumption to predictable hardware investment.
2. **Multimodal Capabilities:** A key target is to integrate a **multimodal Llama** variant to process not only text but also diagrams, handwritten notes, and images uploaded by the user, enhancing the breadth of the RAG system (reference : <https://medium.com/@rajratangulab.more/building-a-multimodal-rag-chatbot-with-image-text-and-table-understanding-91946bc9c51c>)
3. **Performance Optimization:** The current trade-off for zero cost is slower response latency. Future work will concentrate on solving this challenge through:

* **Implementing Ollama:** We plan to use Ollama as the local model runner and API server, which simplifies model management and integration with the Python backend (<https://python.langchain.com/docs/integrations/llms/ollama/>)
* Model quantization (e.g., using GGUF formats) to reduce memory footprint and improve CPU/GPU performance..
* Exploring techniques like batch processing to maintain low latency even under heavy load.

We are confident that by tackling these performance challenges, we can achieve a highly accurate, cost-free, and privacy-preserving AI experience that is superior to the compromises of proprietary API usage.

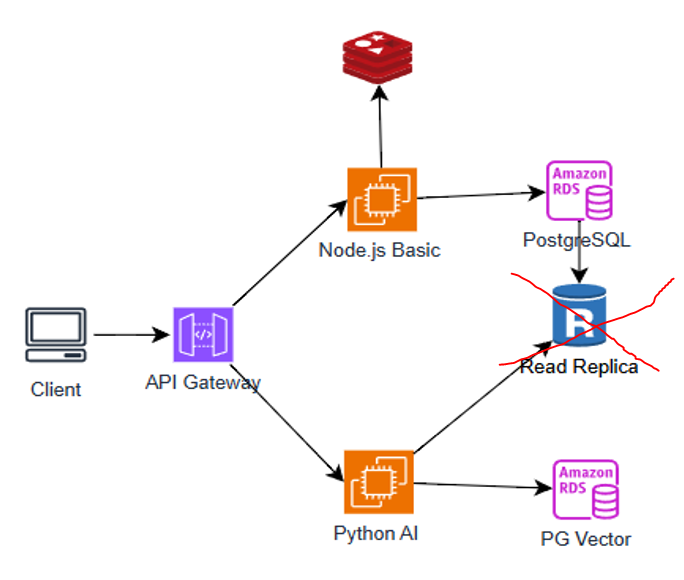
* How does it work? (System diagram)
* Front-end: Technologies, User interface components including interface mock-ups
* Back-end: Technical components
* Data: What data resources are you going to use and how will you access, collect, and store them?

# 5. Evaluation: The Reviews (500 words approx.)

* What does success look like for your system?
* How will you evaluate the system that you built?

# 6. Conclusion: The Plan (500 words approx.)

* What is your project management strategy?
* What are the biggest challenges you are currently facing?
* How will you use the time remaining to achieve a successful outcome?

**Sec 4 stuff**

A diagram of a learning cycle

AI-generated content may be incorrect.

# 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/)