Bilkent University

Department of Computer Engineering

Senior Design Project II – CS 492

T2416
Edux

Detailed Design Report

Cahit Ediz Civan	22003206	ediz.civan@ug.bilkent.edu.tr
Efe Kaan Fidancı	22102589	kaan.fidanci@ug.bilkent.edu.tr
Murat Çağrı Kara	22102505	cagri.kara@ug.bilkent.edu.tr
Bilginer Oral	22103163	bilginer.oral@ug.bilkent.edu.tr
Görkem Kadir Solun	22003214	kadir.solun@ug.bilkent.edu.tr

Supervisor: Doruk Öner

Course Instructors: Atakan Erdem, Mert Bıçakçı

March 10th, 2025

1. Introduction

1.1. Purpose of the System

Edux is designed to be a comprehensive, AI-driven learning platform that empowers students to study more effectively by providing easy access and deeper insights into their course materials. Through integration with modern large language models (LLMs), Edux offers detailed, context-aware explanations of uploaded content—including scholarly articles, textbooks, slides, and syllabi—on both a granular (page-by-page) and holistic level. To reinforce learning outcomes, Edux provides interactive study tools such as flashcards, quizzes, and skill trees, all of which adapt to a student's study materials.

Beyond content explanation, Edux enhances academic performance by generating personalized study schedules that align with user-specific syllabi and performance metrics, enabling learners to organize their study routines efficiently. A planned question-and-answer chatbot extends these capabilities as a responsive assistant for clarifying course-related questions. These features help students retain information more effectively, use their study time better, and remain motivated to achieve their academic goals.

Edux provides a cohesive environment that integrates multiple study materials, interactive learning aids, and LLM-enhanced feedback mechanisms. The result is a flexible platform that supports individual academic progress.

1.2. Design Goals

1.2.1. Usability

Edux aims to provide an intuitive and streamlined user interface that caters to learners of varying technical proficiency. The platform ensures that learners can complete essential tasks in just a few clicks by presenting features like uploading study materials, generating flashcards, and creating personalized schedules in an easy-to-understand manner. Straightforward navigation, readable typography, responsive layout, and accessibility support—such as alternative text for images and screen-reader compatibility—further bolster usability. This emphasis on user-friendliness reduces the learning curve for new users and promotes repeated engagement, as students can more easily integrate Edux into their daily study routines.

1.2.2. Performance

Since Edux processes potentially large volumes of data—from lengthy textbooks to complex slides—performance is a central concern. Quick load times and efficient content parsing are essential to keep user frustration low and maintain platform responsiveness. Leveraging scalable cloud infrastructure, caching frequently accessed data, and optimizing queries can ensure near real-time feedback when learners interact with features like flashcard creation and page-by-page explanations. As usage spikes during peak academic periods such as midterms or finals, Edux's architecture should handle large concurrent loads without significant slowdowns, guaranteeing a seamless user experience regardless of demand.

1.2.3. Reliability

Edux is intended to be an integral study companion; therefore, reliable operation underpins its trustworthiness. Reliability encompasses stable uptime, consistent data integrity, and the ability to recover swiftly from system failures. Automated backups of user-uploaded content, transactional handling of critical data (e.g., quiz results, flashcard generation), and robust error monitoring all help prevent loss of work or progress. By maintaining data consistency and mitigating disruptions—for instance, through failover mechanisms—the platform can reassure learners that the materials and analytics stored in Edux will remain safe and accessible whenever they need them.

1.2.4. Scalability

As Edux aspires to serve a wide range of students—from individual learners to entire classrooms—its design must accommodate growth. This scalability pertains to the number of users and the variety and volume of content types (e.g., PDFs, slides, images, future audio/video recordings). Adopting microservices, containerization, or distributed architectures will enable on-demand resource allocation and streamline performance optimizations. By anticipating future feature expansions—such as advanced retrieval-augmented generation (RAG) or extended analytics tools—Edux's modular infrastructure can evolve incrementally, ensuring improvements do not compromise existing capabilities.

1.2.5. Security

Edux deals with sensitive academic data, such as course materials, user profiles. Security, therefore, is fundamental to safeguarding intellectual property and maintaining user trust. This involves implementing secure channels (e.g., HTTPS) for data transmission, encrypting stored files, and enforcing role-based access control so only authorized individuals can view or modify particular data. Compliance with relevant regulations (GDPR or other data protection policies) is essential, while measures like regular penetration testing, continuous security updates, and robust authentication protocols further solidify the platform's integrity.

1.2.6. Maintainability

Long-term success for Edux depends on code quality and ease of maintenance. As the platform evolves—whether adding question-generation features, refining the user interface, or integrating more sophisticated analytics—clear documentation, modular code structures, and consistent naming conventions help new team members quickly grasp the existing system. Version control and continuous integration practices reduce the risk of introducing regressions and ensure that bug fixes and new features can be rolled out systematically. A well-maintained codebase also facilitates timely responses to user feedback, allowing Edux to adapt smoothly to the shifting demands of modern educational environments.

1.3. Definitions, Acronyms, and Abbreviations

LLM: Large language model

A Large Language Model (LLM) is an advanced artificial intelligence system trained on extensive text datasets to understand and generate human-like language. These models are pivotal in natural language processing tasks, including text generation, translation, and sentiment analysis.

API: Application programming interface

An Application Programming Interface (API) is a set of protocols and tools that allow different software applications to communicate with each other. APIs enable developers to access specific functionalities or data from external services without understanding their internal workings.

RAG: Retrieval augmented generation

Retrieval Augmented Generation (RAG) combines large language models with external data sources to enhance the accuracy and relevance of generated content. RAG systems can produce more informed and contextually appropriate responses by retrieving pertinent information from databases or documents.

GDPR: General data protection regulation

The General Data Protection Regulation (GDPR) is a comprehensive data protection law enacted by the European Union to safeguard individuals' data and privacy. Implemented in 2018, it imposes strict guidelines on data collection, processing, and storage, with significant penalties for non-compliance.

TPS: Transactions per second

Transactions Per Second (TPS) is a metric used to measure the number of transactions a system can process in one second. It's commonly used to assess the performance and scalability of databases, networks, and other transactional systems.

ID: Identifier

An Identifier (ID) is a unique symbol or sequence assigned to an entity to distinguish it from others. IDs are essential in databases, programming, and various systems to efficiently reference and manage specific records or objects.

MySQL

MySQL is an open-source relational database management system (RDBMS) that uses SQL for accessing and managing data. It is widely used for web applications, data storage, and processing large-scale structured data.

Docker

Docker is an open-source platform that enables developers to automate the deployment of applications within lightweight, portable containers. These containers package the application along with its dependencies, ensuring consistency across different environments.

Kubernetes (K8s)

Kubernetes is an open-source system for automating the deployment, scaling, and management of containerized applications. It provides load balancing, service discovery, and self-healing capabilities, making it ideal for managing large-scale containerized workloads.

AKS (Azure Kubernetes Service)

Azure Kubernetes Service (AKS) is a managed Kubernetes service provided by Microsoft Azure. It simplifies the deployment, scaling, and management of Kubernetes clusters, integrating with Azure's security, monitoring, and networking features.

Azure Blob Storage

Azure Blob Storage is Microsoft's cloud-based object storage solution designed for unstructured data such as images, videos, backups, and logs. It provides high availability, scalability, and security for storing large amounts of data.

SHA-256: Secure Hash Algorithm 256-bit

SHA-256 is a cryptographic hash function used widely in various security applications and protocols, including SSL certificates and blockchain technology, providing a fixed-size 256-bit (32-byte) hash that is nearly unique for different inputs and used for ensuring data integrity.

HTTPS: HyperText Transfer Protocol Secure

HTTPS is an internet communication protocol that ensures secure data transfer between a user's web browser and a website. It is an encrypted version of HTTP (HyperText Transfer Protocol) and is widely used to protect sensitive information such as login credentials, payment details, and personal data from cyber threats.

1.4. Overview

In today's fast-paced world, learning and staying ahead requires more than just access to resources. It demands efficiency, adaptability, and focus. While the internet and digital platforms have made educational content more accessible than ever, they've also created an ocean of information that often overwhelms learners. Juggling multiple resources, managing time, and ensuring progress can be intimidating, but many find it hard to keep their knowledge and reach their goals.

This is where Edux steps in as a game-changer. Edux changes the learning process by utilizing the power of large language models (LLMs) and personalized tools to make learning more effective and interactive. With features like detailed explanations powered by LLMs, customizable study aids, and dynamic progress tracking, Edux transforms scattered study efforts into structured, goal-oriented journeys. By tailoring study schedules to individual needs and generating interactive tools such as flashcards, quizzes, and skill trees from user-uploaded content, Edux simplifies complex learning processes while boosting understanding and retention. With plans for future expansion into mobile and tablets, Edux is poised to meet the evolving demands of modern education. However, these developments fall outside the current scope of the senior design project.

This report outlines the purpose, design goals, system architecture, and various engineering considerations that shape the development of Edux. The document provides a structured overview of the platform's functionalities, subsystem services, security measures, and testing strategies. It also discusses aspects of teamwork involved in the project, relevant industry standards, and key references supporting the development process.

2. Current Software Architecture

2.1. Current Situation and Limitations of Existing Learning Solutions

In the evolving educational landscape, digital learning solutions have become increasingly prominent. However, many existing platforms present significant limitations. Learners often struggle with fragmented study materials spread across multiple applications, making maintaining an organized and structured learning process difficult. Additionally, while AI powered tools have emerged to assist with learning, they often focus on isolated features such as automatic note generation or quiz creation rather than providing an integrated and holistic approach. A significant challenge is the lack of personalized study materials. Most learning platforms offer static content that does not adapt to individual user needs. This results in inefficiencies where learners must manually curate their study resources.

Moreover, existing solutions often fail to provide detailed insights into progress tracking. Learners struggle to evaluate their strengths and weaknesses, as many tools lack comprehensive analytics that monitor study habits and performance over time.

2.2. Comprehensive and Personalized Learning Solution

Our proposed system addresses these limitations by integrating multiple learning tools into a single, cohesive platform. Instead of offering fragmented solutions, our system provides an end to end approach to digital learning, combining resource management, interactive study aids, and AI driven insights.

Key improvements over existing systems include:

- **Personalized Study Plans**: Users can upload course materials, and our system generates dynamic study schedules based on syllabi.
- **Advanced Progress Tracking**: Learners receive detailed performance reports that highlight knowledge gaps and suggest areas for improvement.
- **AI-Driven Learning Assistance**: The system uses LLMs to generate contextual explanations, quizzes, and flashcards tailored to uploaded study materials.
- **Holistic Approach**: Unlike platforms focusing solely on content generation or AI tutoring, Edux provides a comprehensive learning environment that addresses all aspects of the educational process.
- **Enhanced Personalization**: Edux's ability to tailor study schedules and generate interactive tools based on individual needs ensures that learners receive support that aligns closely with their goals and learning styles.
- **Future-Ready Features**: Integrating advanced technologies like RAG positions Edux at the forefront of educational innovation, offering capabilities that competitors lack.

By bringing these features together, our platform enhances efficiency, engagement, and adaptability, making learning more structured and goal-oriented.

2.3. Edux vs. Competitors: Addressing the Gaps

Unlike existing solutions that focus on individual learning aids, our platform takes a holistic approach to education. We address several key gaps that current tools fail to resolve:

- **Engagement**: Interactive flashcards, quizzes, and AI-generated explanations make learning more engaging.
- **Customization**: Personalized study schedules adapt to individual progress, unlike static resources found on other platforms.
- **Performance Analysis**: Comprehensive analytics provide learners with actionable insights, unlike basic progress tracking in many competitors.
- **Future Personalization**: Planned features like RAG and personalized schedules will further unify the learning experience.

By integrating these features, our platform transforms disorganized study efforts into structured, efficient learning experiences. This comprehensive approach helps students stay on track, enhance retention, and optimize their study process.

Current Features Implemented

1. **Interactive Learning Tools**: Edux generates flashcards and quizzes from user-uploaded content, transforming static study materials into engaging aids that improve understanding and retention. These tools are available via the platform's dashboard.
2. **New Frontend Adoption:** We have adopted a new frontend to enhance user experience and interface design.
3. **Backend Transformation:** Our monolithic backend has been transformed into microservices, including users, user analytics, courses, chat, and filesystem microservices. This modular approach improves scalability and maintainability.
4. **User Analytics:** We have added user analytics to provide insights into user behavior and learning patterns, enabling more personalized and effective educational strategies.
5. **Google Flash 1.5 Integration:** Users can upload slides to Google Flash 1.5, where the LLM explains the content of the slides and creates dynamic flashcards and quizzes based on these explanations. This feature enhances the learning experience by making slide content more interactive and engaging.

Technologies Used:

- **Frontend**: React.js [1] for the user interface, ensuring a responsive and interactive experience.
- **Backend**: Rest [2] is used for inter-microservice communications, and MySQL [3] is used for database management.

Features in Progress

1. **Skill Trees**: The development of interactive skill trees is underway. This feature will allow learners to visualize their progress and see how individual topics connect to broader learning objectives.
2. **Course Creation Tools**: We build tools to help learners create and manage custom courses, allowing for greater flexibility and personalization.

Planned Features

1. **Retrieval-Augmented Generation (RAG)**: This feature will enhance personalized learning by generating unique practice questions. The goal is to improve academic integrity and provide deeper personalization.

2. **Personalized Study Schedules**: We plan to develop automated study plans tailored to individual syllabi, assessment results, and available study time. This will help learners manage their time effectively and stay on track.

By combining implemented features, ongoing developments, and future innovations, Edux transforms scattered study efforts into structured, goal-oriented journeys. This comprehensive approach empowers learners to navigate the overwhelming sea of information confidently and efficiently, enhancing academic performance and lifelong learning.

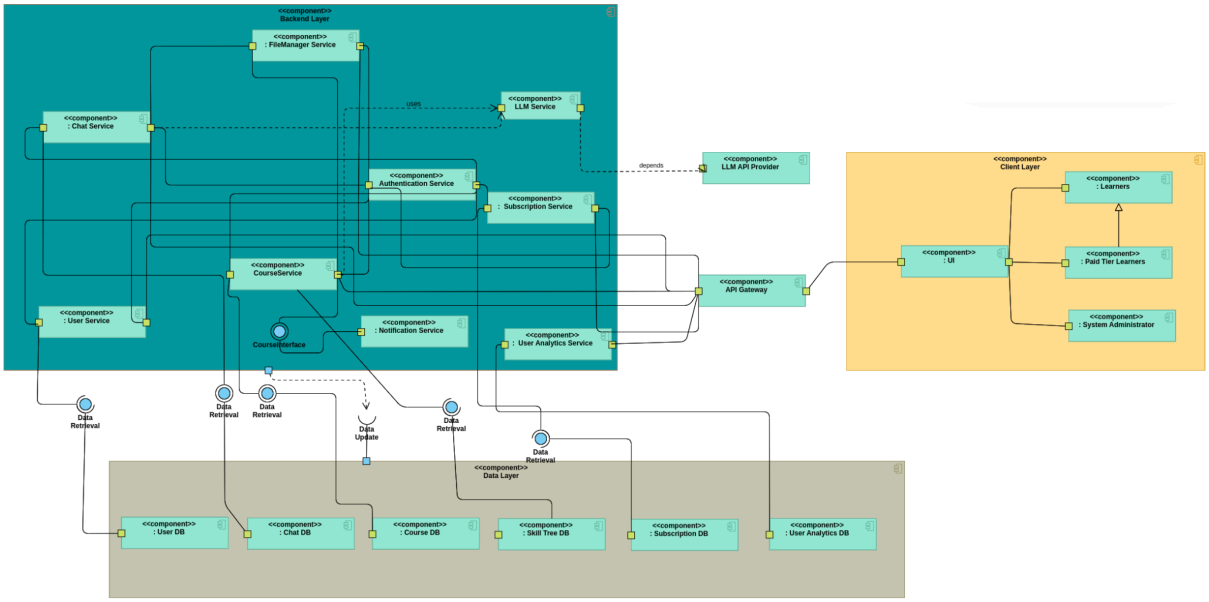
3. Proposed Software Architecture

3.1. Overview

The Edux system follows a multi-layered architecture, consisting of the Client Layer, Backend Layer, and Data Layer. The Client Layer provides the interface for different user roles, while the Backend Layer manages core functionalities through various interconnected services. The Data Layer ensures structured storage and retrieval of information. An API Gateway facilitates communication between components, and an LLM Service integrates AI-powered capabilities. This modular design enhances scalability, maintainability, and seamless user interaction.

3.2. Subsystem Decomposition

Below is Edux's subsystem decomposition diagram:



The Edux system follows a multi-layered architecture, comprising the Client Layer, Backend Layer, and Data Layer, each serving distinct roles to ensure modularity, scalability, and maintainability.

The Client Layer consists of the UI component, which facilitates interaction between users and the system. Different user roles, including Learners, Paid Tier Learners, and System Administrators, access the platform through this layer.

The Backend Layer serves as the core of the system, managing user authentication, course services, subscriptions, notifications, chat functionalities, and user analytics. It is composed of multiple interconnected services, including User Service, Chat Service, Course Service, Authentication Service, Subscription Service, Notification Service, User Analytics Service and FileManager Service. Except FileManager and Course services, all other services provide public routes for UI layer to interact with the backend. Moreover, the services use internal mechanisms to interact with themselves, without exposing such private routes publicly. Additionally, an LLM Service integrates AI capabilities by interacting with external LLM API providers, enhancing learning experiences. The API Gateway acts as the central communication hub between the client and backend components.

The Data Layer provides structured storage and retrieval through various databases, such as User DB, Chat DB, Course DB, Skill Tree DB, Subscription DB, and User Analytics DB. These databases ensure efficient data management, supporting backend services.

3.3. Persistent Data Management

MySQL databases will be used in Docker containers, orchestrated by Kubernetes, and deployed on Azure Kubernetes Services. Some of our microservices have databases of their own, that means we will have multiple database containers, i.e., one per microservice group needing a database. The following will be stored in MySQL databases: user information (email, ID, subscription level, hashed password, etc.), course information (course ID, course name, weekly study plan, syllabus, skill tree reference, etc.), chat information (chat id, chat name, slide references, quiz references, flashcard references, etc.), skill tree information (nodes, pointers, quiz references for each node, etc.). We will use Azure Blob Storage to store files like chat histories, slide explanation histories, images (slide and pdf pages), quizzes, flashcards, quizzes of skill tree nodes, and other user-uploaded files. FileManager service keeps their references in a dedicated MySQL database and provides unique IDs so that the other services can access them easily.

3.4. Access Control and Security

Users can sign up using their email or Google accounts. All of our public endpoints, which the front-end clients contact, are secured with JWT tokens [4]. Requests that don't have valid tokens are not served. The authentication service oversees these operations. Furthermore, some of our functionalities (like advanced LLM selection) can be used only by a set of subscription plans. Some endpoints check for the user's subscription level before serving them. API keys also secure our interservice communication channels. Each service pair has a secret API key and they handshake using those keys. Requests with invalid API keys are deemed to have come from strangers and are not served.

4. Subsystem services

4.1. Authentication Service

The Authentication Service of Edux is responsible for managing user authentication and authorization, ensuring secure access to various resources within the system. It primarily revolves around issuing and validating JWT (JSON Web Tokens) to authenticate users.

At the core of the service is a FastAPI-based application that provides both public and private APIs. The public API allows users to log in using their credentials, which are then verified against the User Service. If authentication is successful, a JWT token is generated and returned to the client. This token contains encoded user information and an expiration timestamp, allowing the client to authenticate subsequent requests without needing to log in again. The private API, on the other hand, is protected and requires either a valid JWT token or an API key for access.

The service supports OAuth2 password-based authentication, where users provide their email and password. The authentication process involves retrieving the user's hashed password from the User Service and verifying it. If the password is correct, a JWT token is issued, signed with a secret key and an expiration time. This token is used in subsequent requests to prove the user's identity.

Security is further enforced through API key verification, which is used to control access to internal endpoints. The authentication service maintains a list of allowed API keys, and any request to private endpoints must include a valid key in the request headers.

The authentication flow in Edux ensures that only legitimate users can access protected resources while maintaining security best practices such as token expiration, password hashing, and API key validation. This service plays a crucial role in safeguarding user accounts and securing interactions within the Edux ecosystem.

4.2. User Service

The User Service of Edux is a backend component built using FastAPI that manages user-related operations such as user creation, authentication, retrieval, and updates. It interacts with a database to store and manage user records. The data access layer in the service abstracts database operations, allowing users to be created, fetched, updated, and potentially deleted. The User service also communicates with other services, including authentication and course services, through REST API.

At the core of the service is its API, which is divided into public and private endpoints. The public API allows users to create an account, authenticate themselves using their authentication tokens, and retrieve their own profile and associated courses. The private API, secured with an API key, provides administrative operations such as retrieving user details based on various parameters and only open for internal communication mechanisms between services.

The User service handles authentication through the external Authentication Service, which verifies JWT tokens and hashes passwords. The authentication client sends requests to this service, ensuring secure user verification. Similarly, a course client fetches a user's enrolled courses from the course service, integrating course data into user profiles.

4.3. Course Service

The Course Service in Edux is responsible for managing courses, allowing users to create, retrieve, update, and delete course-related data. This service operates as a FastAPI-based microservice, integrating with multiple external services like the User Service, GenAI Service, FileManager, and Chat Service to provide a comprehensive course management system.

At its core, the service allows users to create courses by providing details such as course name, course code, and description. During creation, users can also upload a course syllabus and an icon image. If a syllabus is uploaded, the service interacts with the GenAI Service to generate a weekly study plan, which is then stored as a file using the FileManager Service. All course-related files, including the syllabus, study plan, and icon, are managed through this external file storage.

To maintain security, the Course Service requires authentication and authorization. Public endpoints, such as retrieving a course by ID, require a valid JWT token, which is verified against the User Service. Private endpoints, like fetching all courses for a specific user, are protected using API keys to ensure that only authorized services can access them.

The service also provides functionality for updating courses, allowing users to modify details such as the name, code, description, and uploaded files. When a syllabus is updated, a new study plan is generated automatically. If users choose to update the course icon, the system ensures that only valid image formats are accepted, and the existing icon is replaced.

Deletion of a course involves removing its associated files from the FileManager Service and deleting any linked conversations in the Chat Service to ensure data consistency. The database is managed using SQLAlchemy, ensuring efficient storage and retrieval of course-related data.

4.4. Chat Service

The Chat Service in Edux facilitates real-time conversation between users and AI, integrating various functionalities such as text-based chat, file support, slide-based learning, quiz generation, and flashcards. It serves as a key component in providing interactive and intelligent discussions within the Edux ecosystem.

At its core, the service manages chat sessions associated with courses. Users can create a chat within a course and interact with the system either through text-based messages or by uploading files such as PDFs and PowerPoint presentations. The service processes these files and incorporates them into the conversation.

The chat history is stored and structured in a way that allows integration with multiple Generative AI providers such as OpenAI, Google Gemini, and Anthropic [5], [6], [7]. The system ensures compatibility by converting chat history into specific formats required by each provider. When a user sends a message, the service retrieves the corresponding chat history, formats it appropriately, and forwards it to the AI model for response generation.

In addition to text-based interaction, the service supports slide-based learning where users can upload and interact with slides as part of their discussions. The system extracts content from slides, allowing the AI to generate explanations or summaries. Furthermore, users can create quizzes and flashcards based on chat interactions, enabling a more structured learning experience.

Security and access control are enforced through JWT-based authentication and API key validation. Users must be authenticated to access their chat history, while certain private API endpoints require a valid API key for authorization. Additionally, chat-related data is stored securely, and files are managed through an integrated File Manager Service.

The Chat Service plays a crucial role in enabling AI-assisted learning within Edux by offering a seamless, multi-modal interaction experience. Through its structured chat management, AI model integration, and educational tools like quizzes and flashcards, it enhances the way users engage with learning materials and AI-driven discussions.

4.5. GenAI Service

The GenAI Service of Edux is a specialized FastAPI-based microservice designed to facilitate AI-driven educational interactions. It acts as an interface between users and multiple large language models (LLMs), providing chat-based assistance, study plan generation, quiz creation, and flashcard generation to enhance learning experiences.

At its core, the service integrates three major AI providers: OpenAI (GPT models), Anthropic (Claude models), and Google (Gemini models). It includes a flexible client abstraction layer that allows selecting an AI model dynamically based on user preferences or system configurations. Depending on the chosen model, requests are routed to the appropriate API client, where prompts and chat history are processed before generating responses.

The public-facing API allows students to engage in chat-based interactions with the AI assistant. The assistant is guided by a system prompt that enforces educational constraints, ensuring that responses remain focused on course-related topics. Additionally, the assistant can generate weekly study plans by analyzing uploaded syllabi, ensuring structured learning over a given time frame. For active engagement, it can also generate quizzes and flashcards based on previous chat history or provided content, reinforcing student understanding through recall-based learning.

Security and access control are integral to the service. The API enforces authentication via API keys, ensuring that only authorized users or services can access its endpoints. Additionally, integration with the User Service allows retrieval of authenticated user details, verifying access rights before processing requests.

4.6. FileManager Service

The FileManager Service in Edux is designed to handle file storage, retrieval, and management efficiently. It provides a secure way for users to upload, download, and delete files, while integrating with a cloud-based Azure Blob Storage for scalable storage solutions.

At its core, the service is built using FastAPI and provides a private API that allows only authorized services or users with valid API keys to interact with it. When a user uploads a file, the service first records the file's metadata in a MySQL database and then transfers the actual file to Azure Blob Storage. Each file is assigned a unique identifier (file_id), ensuring that it can be referenced and managed later.

For downloading files, the service retrieves the file metadata from the database and then fetches the actual file from the cloud. The file is streamed back to the user, preserving its original content type and filename. The service also supports batch operations, allowing multiple files to be uploaded, downloaded, or deleted in a single request.

Security is enforced through API key validation, ensuring that only authorized clients can perform file operations. Additionally, it uses a database layer to keep track of files associated with users, enabling structured access control.

5. Test Cases

The procedures are written for test software engineers.

5.1. Functionality Test Cases

Test ID	TCF0	Category	Functional	Severity	Medium
Objective	Logout				
Steps	● Login to Edux with any account. ● After logging in, log out using the navigation bar on the right. ● Click the “Logout” button.				
Expected	The user will log out and be directed to the login page.				
Date/Result	13.08.2024 / Success				

Test ID	TCF1	Category	Functional	Severity	Low
Objective	Show Terms of Service and Privacy Policy				
Steps	● Open Edux on the browser. ● Navigate to the “Sign in” page. ● At the bottom, click “Terms of Service” and “Privacy Policy,” and click one of them.				
Expected	Terms of Service or Privacy Policy are shown.				

Date/Result	13.08.2024 / Success

Test ID	TCF2	Category	Functional	Severity	Low
Objective	Show About Page				
Steps	● Login to Edux on the browser. ● Navigate to the “About” page from the navigation bar. ● Click “Team” or “Github Page.”				
Expected	Team or Github Page are shown.				
Date/Result	17.08.2024 / Success				

Test ID	TCF3	Category	Functional	Severity	Medium
Objective	Show Subscription Service				
Steps	● Login to Edux on the browser. ● Navigate to the “Services” dropdown from the navigation bar. ● Select the “Subscription Service.”				
Expected	Subscriptions page will be shown.				
Date/Result	15.02.2025 / Success				

Test ID	TCF4	Category	Functional	Severity	Medium
Objective	Show Notifications				
Steps	● Login to Edux on the browser.				

	● Navigate to the “Services” dropdown from the navigation bar. ● Select the “Notifications.”
Expected	Notifications page will be shown.
Date/Result	Not yet.

Test ID	TCF5	Category	Functional	Severity	Medium
Objective	Search in Edux				
Steps	● Login to Edux on the browser. ● Navigate to the “Search in Edux…” dropdown from the navigation bar. ● Write the feature that you want to access. ● Select the feature shown below.				
Expected	The selected feature will be opened.				
Date/Result	24.08.2024 / Success				

Test ID	TCF6	Category	Functional	Severity	Medium
Objective	Switch between dark and light mode				
Steps	● Open Edux on the browser. ● Navigate to the moon icon on the right of the navigation bar. ● Click the moon icon. ● Select the theme you want: dark, light, or system.				
Expected	The user interface will follow the wanted theme.				
Date/Result	27.08.2024 / Success				

Test ID	TCF7	Category	Functional	Severity	Critical
Objective	Open Skill Tree				
Steps	● Login to Edux on the browser. ● Click the “Skill Tree” card in the main menu.				
Expected	The Skill Tree page will be shown.				
Date/Result	25.10.2024 / Success				

Test ID	TCF8	Category	Functional	Severity	Critical
Objective	Creating a Skill Tree				
Steps	● Login to Edux on the browser. ● Click the “Skill Tree” card in the main menu. ● Open the sidebar on the Skill Tree page. ● Click the Create icon on the top of the sidebar. ● Fill in the required fields in the creation modal.				
Expected	A new Skill Tree will be created.				
Date/Result	17.02.2025 / Success				

Test ID	TCF9	Category	Functional	Severity	Critical
Objective	Creating a Skill Tree: Unsuccessful Creation				
Steps	● Login to Edux on the browser. ● Click the “Skill Tree” card in the main menu. ● Open the sidebar on the Skill Tree page. ● Click the Create icon on the top of the sidebar.				

	● Fill in the required fields in the creation modal. ● Skill Tree creation failed.
Expected	Skill Tree creation fails with an error message: “Unable to create Skill Tree, please try again later.” Error logs indicate server-side validation failure. Issue requires immediate backend service review.
Date/Result	17.02.2025 / Fail

Test ID	TCF10	Category	Functional	Severity	Critical
Objective	Editing a Skill Tree				
Steps	● Login to Edux on the browser. ● Click the “Skill Tree” card in the main menu. ● Open the sidebar on the Skill Tree page. ● Click the edit icon on the Skill Tree that you want to edit. ● Change the fields in the edit modal.				
Expected	The selected Skill Tree will be edited.				
Date/Result	17.02.2025 / Success				

Test ID	TCF11	Category	Functional	Severity	Critical
Objective	Editing a Skill Tree: Unsuccessful Edit				
Steps	● Login to Edux on the browser. ● Click the “Skill Tree” card in the main menu. ● Open the sidebar on the Skill Tree page. ● Click the edit icon on the Skill Tree that you want to edit. ● Change the fields in the edit modal.				
Expected	The selected Skill Tree could not be edited; an error occurred stating “Unable to save changes, please retry.”				

Date/Result	17.02.2025 / Fail

Test ID	TCF12	Category	Functional	Severity	Critical
Objective	Deleting a Skill Tree				
Steps	● Login to Edux on the browser. ● Click the “Skill Tree” card in the main menu. ● Open the sidebar on the Skill Tree page. ● Click the delete icon on the Skill Tree that you want to delete. ● Click confirm on the deletion modal.				
Expected	The selected Skill Tree will be deleted.				
Date/Result	17.02.2025 / Success				

Test ID	TCF13	Category	Functional	Severity	Critical
Objective	Viewing a Skill Tree				
Steps	● Login to Edux on the browser. ● Click the “Skill Tree” card in the main menu. ● Open the sidebar on the Skill Tree page. ● Click the view icon on the Skill Tree that you want to view.				
Expected	The selected Skill Tree will be shown, and the graph will be rendered.				
Date/Result	17.02.2025 / Success				

Test ID	TCF14	Category	Functional	Severity	Critical
Objective	Creating an Individual Study				
Steps	● Login to Edux on the browser.				

	● Click the Create icon on the right of the “Your Studies” in the main menu. ● Fill in the required sections on the modal. ● Click the Save button.
Expected	A new Individual Study will be created.
Date/Result	14.09.2024 / Success

Test ID	TCF15	Category	Functional	Severity	Critical
Objective	Creating an Individual Study: Creation Fail				
Steps	● Login to Edux on the browser. ● Click the Create icon on the right of the “Your Studies” in the main menu. ● Fill in the required sections on the modal. ● Click the Save button.				
Expected	Error message displayed: “Unable to create Individual Study due to incomplete fields.” despite all required fields being filled. Validation or backend issue suspected.				
Date/Result	14.09.2024 / Fail				

Test ID	TCF16	Category	Functional	Severity	Critical
Objective	Editing an Individual Study				
Steps	● Login to Edux on the browser. ● Click the Edit icon on the right of the “View Course” in the Your Studies section on the Individual Study you want to edit. ● Change the sections on the modal. ● Click the Save button.				

Expected	The selected Individual Study will be updated.
Date/Result	15.09.2024 / Success

Test ID	TCF17	Category	Functional	Severity	Critical
Objective	Editing an Individual Study: Edit Fail				
Steps	● Login to Edux on the browser. ● Click the Edit icon on the right of the “View Course” in the Your Studies section on the Individual Study you want to edit. ● Change the sections on the modal. ● Click the Save button.				
Expected	The edit action fails, displaying an error message: “Update failed, please retry.”				
Date/Result	15.09.2024 / Fail				

Test ID	TCF18	Category	Functional	Severity	Critical
Objective	Deleting an Individual Study				
Steps	● Login to Edux on the browser. ● Click the Delete icon on the right of the “View Course” in the Your Studies section on the Individual Study you want to delete. ● Click the Confirm button.				
Expected	The selected Individual Study will be deleted.				
Date/Result	15.09.2024 / Success				

Test ID	TCF19	Category	Functional	Severity	Critical
Objective	Open an Individual Course				
Steps	● Login to Edux on the browser. ● Click the “View Course” button on the Individual Study you want to open in the Your Studies section.				
Expected	The user will be navigated to the selected Individual Study.				
Date/Result	14.09.2024 / Success				

Test ID	TCF20	Category	Functional	Severity	Critical
Objective	Open Flashcards				
Steps	● Login to Edux on the browser. ● Click the “View Course” button on the Individual Study you want to open in the Your Studies section. ● Click the Flashcards card on the Individual Study page.				
Expected	Flashcards page will be opened.				
Date/Result	23.01.2025 / Success				

Test ID	TCF21	Category	Functional	Severity	Critical
Objective	Open Quizzes				
Steps	● Login to Edux on the browser. ● Click the “View Course” button on the Individual Study you want to open in the Your Studies section. ● Click the Quizzes card on the Individual Study page.				

Expected	Quizzes page will be opened.
Date/Result	23.01.2025 / Success

Test ID	TCF22	Category	Functional	Severity	Critical
Objective	Open Instructor/Chat				
Steps	● Login to Edux on the browser. ● Click the “View Course” button on the Individual Study you want to open in the Your Studies section. ● Click the Instructor/Chat card on the Individual Study page.				
Expected	Instructor/Chat page will be opened.				
Date/Result	23.10.2024 / Success				

Test ID	TCF23	Category	Functional	Severity	Critical
Objective	Open Weekly Study Plan				
Steps	● Login to Edux on the browser. ● Click the “View Course” button on the Individual Study you want to open in the Your Studies section. ● Click the Weekly Study Plan card on the Individual Study page.				
Expected	Weekly Study Plan page will be opened.				
Date/Result	19.11.2025 / Success				

Test ID	TCF24	Category	Functional	Severity	Critical

Objective	Upload Syllabus
Steps	● Login to Edux on the browser. ● Click the “View Course” button on the Individual Study you want to open in the Your Studies section. ● Click the Upload Syllabus card on the Individual Study page. ● Upload the Syllabus to the opened modal. ● Click the Upload button.
Expected	The syllabus will be uploaded/updated.
Date/Result	19.11.2025 / Success

Test ID	TCF25	Category	Functional	Severity	Critical
Objective	Create Chat				
Steps	● Login to Edux on the browser. ● Click the “View Course” button on the Individual Study you want to open in the Your Studies section. ● Click the Instructor/Chat card on the Individual Study page. ● Click the Create chat button on the top left of the page. ● Fill in the opened sidebar. ● Click the Create chat button.				
Expected	A new chat will be created.				
Date/Result	23.10.2024 / Success				

Test ID	TCF26	Category	Functional	Severity	Critical
Objective	Create Chat: Creation Fail				
Steps	● Login to Edux on the browser. ● Click the “View Course” button on the Individual Study you want to open in the Your Studies section. ● Click the Instructor/Chat card on the Individual Study page. ● Click the Create chat button on the top left of the page. ● Fill in the opened sidebar. ● Click the Create chat button.				

Expected	Chat creation failed with error: “Chat creation unsuccessful, please check input fields and retry.”
Date/Result	23.10.2024 / Fail

Test ID	TCF27	Category	Functional	Severity	Critical
Objective	Send a Message to a Chat				
Steps	- Login to Edux on the browser. - Click the “View Course” button on the Individual Study you want to open in the Your Studies section. - Click the Instructor/Chat card on the Individual Study page. - Select the chat from which you want to send the message from the sidebar. - Type your message at the bottom of the page. - Click the Send Message button.				
Expected	A new message will be sent to the chat.				
Date/Result	23.10.2024 / Success				

Test ID	TCF28	Category	Functional	Severity	Critical
Objective	Rename Chat				
Steps	- Login to Edux. - Click the “View Course” button on the Individual Study you want to open in the Your Studies section. - Click the Instructor/Chat card on the Individual Study page. - Select the chat from which you want to rename from the sidebar. - Click the three dots on the right of the selected chat. - Write the new name. - Click the Confirm button.				
Expected	The selected chat will be renamed.				
Date/Result	23.10.2024 / Success				

Test ID	TCF29	Category	Functional	Severity	Critical
Objective	Create a Quiz from a Chat				
Steps	● Login to Edux. ● Click the “View Course” button on the Individual Study you want to open in the Your Studies section. ● Click the Instructor/Chat card on the Individual Study page. ● Select the chat from which you want to create a quiz from the sidebar. ● Click the three dots on the right of the selected chat. ● Click the Create Quiz button. ● Click the Confirm button.				
Expected	A new Quiz will be created.				
Date/Result	23.01.2025 / Success				

Test ID	TCF30	Category	Functional	Severity	Critical
Objective	Attaching a file to a Chat				
Steps	● Login to Edux. ● Click the “View Course” button on the Individual Study you want to open in the Your Studies section. ● Click the Instructor/Chat card on the Individual Study page. ● Select the chat from which you want to attach a file from the sidebar. ● Click the file attach button from the bottom of the page. ● Select the files and click the Confirm button.				
Expected	A new file will be attached.				
Date/Result	25.10.2024 / Success				

Test ID	TCF31	Category	Functional	Severity	Critical
Objective	Deleting a Chat				
Steps	● Login to Edux. ● Click the “View Course” button on the Individual Study you want to				

	open in the Your Studies section. ● Click the Instructor/Chat card on the Individual Study page. ● Select the chat that you want to delete from the sidebar. ● Click the three dots on the right of the selected chat. ● Click the Delete button. ● Click the Confirm button.
Expected	Selected chat will be deleted.
Date/Result	25.10.2024 / Success

Test ID	TCF32	Category	Functional	Severity	Critical
Objective	Create Flashcards from a Chat				
Steps	● Login to Edux. ● Click the “View Course” button on the Individual Study you want to open in the Your Studies section. ● Click the Instructor/Chat card on the Individual Study page. ● Select the chat from which you want to rename from the sidebar. ● Chat for some time ● Click the three dots on the right of the selected chat. ● Click Create Flashcard button ● Confirm a Flashcard has appeared in Flashcards				
Expected	A Flashcard must appear in the individual study’s flashcard section.				
Date/Result	23.01.2025 / Success				

Test ID	TCF33	Category	Functional	Severity	Low
Objective	Ensure that users can update their profile information.				
Steps	● Login to Edux. ● Navigate to the “Profile” section. ● Edit profile details (e.g., name, bio, avatar). ● Click “Save.”				

Expected	The updated information should be saved and displayed correctly.
Date/Result	Not yet.

5.1.1. Integration Test Cases

Test ID	TCI0	Category	Integration	Severity	Critical
Objective	Retrieving courses from the course database				
Steps	● Sign in as one of the users ● Call the course service's endpoint for retrieving the courses that belong to our user				
Expected	● The courses are expected to be retrieved from the course service's database successfully.				
Date/Result	19.02.2024 / Success				

Test ID	TCI1	Category	Integration	Severity	Critical
Objective	Retrieving chats from the chats database				
Steps	● Sign in as one of the users ● Call the course service's endpoint for retrieving the chat id’s associated with it. ● Call the chat service’s endpoint for retrieving the chats with the given id’s from the chat database				
Expected	● The chats are expected to be retrieved from the chat service's database successfully.				
Date/Result	17.02.2024 / Success				

Test ID	TCI2	Category	Integration	Severity	Critical
Objective	Getting the current user from the user database				
Steps	● Sign in as one of the users ● Call any public endpoint in any service, that endpoint depends on the current user so it will call the user service to fetch the current user. ● The current user will get the current user using the given Json web token				
Expected	● The current user is expected to be retrieved from user service's database				
Date/Result	16.02.2024 / Success				

Test ID	TCI3	Category	Integration	Severity	Critical
Objective	Uploading files to Azure Blob Storage				
Steps	● Sign in to Edux ● Open up one of the existing course ● Open up one of the existing chats ● Upload a file ● Make a fetch query for the uploaded file to Azure Blob Storage				
Expected	● The uploaded file is expected to be retrieved successfully from Azure Blob Storage				
Date/Result	Not yet.				

Test ID	TCI4	Category	Integration	Severity	Critical
Objective	Uploading slides page by page as images to Azure Blob Storage				
Steps	● Sign in to Edux				

	● Open up one of the course ● Create a chat by providing some slides (pdf or pptx). The course will be created in slides mode. ● The slide pages will be uploaded page by page to Azure Blob Storage and their id's will be returned. ● In a loop, make a fetch query for the page images to Azure Blob Storage.
Expected	● The slide pages are expected to be retrieved from the Azure Blob Storage successfully.
Date/Result	Not yet.

Test ID	TCI5	Category	Integration	Severity	Major
Objective	When a chat is deleted, the files associated with it should also be deleted from Azure Blob Storage				
Steps	● Sign in to Edux ● Open up a course, find a chat that has files uploaded to it. ● Get the id's of the files that are associated with that chat. ● Delete the chat. ● Make a query to Azure Blob Storage for those files with the given id's.				
Expected	● No files are expected to be retrieved.				
Date/Result	Not yet.				

Test ID	TCI6	Category	Integration	Severity	Medium
Objective	When a chat is deleted the slide histories associated with it should also be deleted from Azure Blob Storage				
Steps	● Sign in to Edux ● Open up a course, find a chat that has files uploaded to it. ● Get the id's of the slide histories of the slides that are associated with that chat. ● Delete the chat. ● Make a query to Azure Blob Storage for those slide histories with the				

	given id’s.
Expected	- No files are expected to be retrieved.
Date/Result	Not yet.

Test ID	TCI7	Category	Integration	Severity	Major
Objective	When a chat is deleted, the chat history associated with it should also be deleted from Azure Blob Storage				
Steps	- Sign in to Edux - Open up a course, find a chat with messages. - Get the id of the chat history. - Delete the chat. - Make a query to Azure Blob Storage for the chat history with the given id.				
Expected	- No file is expected to be retrieved.				
Date/Result	Not yet.				

Test ID	TCI8	Category	Integration	Severity	Critical
Objective	When an explanation for the current slide is asked, the response should be inserted into the slide history.				
Steps	- Sign in to Edux - Open up a course. - Create a chat in slides mode (upload slides during creation). - Call the necessary endpoint for explanations on the current slide page. - Get the id of the slide history of the current slide page. - Make a query to Azure Blob Storage for the slide history with the given id. - Search for the last response (explanations) from the LLM service inside the slide history.				

Expected	- The last response from the LLM service is expected to be present in the file.
Date/Result	Not yet.

5.2. Non-Functional Test Cases

5.2.1. Performance

Test ID	TCNF0	Category	Performance	Severity	Low
Objective	Measure the time taken to create a new chat.				
Steps	- Login to Edux and navigate to the "Instructor/Chat" section. - Click the "Create Chat" button. - Enter a chat name and select participants. - Click the "Confirm" button to create the chat. - Measure the time taken for the system to create the chat.				
Expected	- The chat should be created within an acceptable time frame (must not exceed 5 seconds). - The chat should appear in the list of active chats. - A confirmation message should indicate successful chat creation.				
Date/Result	Not yet.				

Test ID	TCNF1	Category	Performance	Severity	Medium
Objective	Measure the time taken to upload files to a chat.				
Steps	- Login to Edux and navigate to the "Instructor/Chat" section. - Select an active chat conversation. - Click the "Attach File" button. - Select a file (e.g., PDF, image, document) and confirm upload. - Measure the time taken for the system to upload the file and make it				

	accessible.
Expected	● The file should be uploaded within an acceptable time frame (must not exceed 8 seconds for files up to 5MB). ● The file should be accessible to all participants in the chat. ● A confirmation message should indicate successful file upload.
Date/Result	Not yet.

Test ID	TCNF2	Category	Performance	Severity	Medium
Objective	Measure the response time of the instructor chat to ensure timely communication.				
Steps	● Login to Edux with an instructor account. ● Navigate to the “Instructor/Chat” section from the main menu. ● Select an ongoing chat conversation or create a new one. ● A student sends a message to the instructor. ● Measure the time taken for the instructor to receive the message and respond.				
Expected	● The instructor's response should be retrieved within an acceptable timeframe (must not exceed 5 seconds).				
Date/Result	Not yet.				

Test ID	TCNF3	Category	Performance	Severity	Medium
Objective	Measure the time taken to create a quiz from chat interactions.				
Steps	● Login to Edux and navigate to the “Instructor/Chat” section. ● Select a chat conversation containing relevant study discussions. ● Click the “Create Quiz” button from the chat options menu. ● Confirm the quiz generation request. ● Measure the time taken for the system to generate the quiz.				

Expected	● The quiz should be generated within an acceptable time frame (must not exceed 10 seconds). ● The quiz should contain relevant questions based on the chat discussion. ● A confirmation message should appear indicating successful quiz creation.
Date/Result	Not yet.

Test ID	TCNF4	Category	Performance	Severity	Medium
Objective	Measure the time taken to create a quiz from chat interactions.				
Steps	● Login to Edux and navigate to the “Instructor/Chat” section. ● Select a chat conversation containing study-related content. ● Click the “Create Flashcards” button from the chat options menu. ● Confirm the flashcard generation request. ● Measure the time taken for the system to generate the flashcards.				
Expected	● The flashcards should be generated within an acceptable time frame (must not exceed 10 seconds). ● The flashcards should be relevant to the content discussed in the chat. ● A confirmation message should appear indicating successful flashcard creation.				
Date/Result	Not yet.				

5.2.2. Security Test Cases

Test ID	TCS0	Category	Security	Severity	Critical
Objective	Sign up for an account with registered mail				
Steps	● Open Edux. ● Direct the page to the Sign Up page using the button Sign Up.				

	● Enter a mail that is already registered in the system. ● Click on the Sign Up button. ● Pop up a toaster that displays the error message “The mail is already registered for another user.”.
Expected	The error toaster should display the message “The mail is already registered for another user.”, and the user shall not be able to sign up for an account until they write a mail that is not in the User database.
Date/Result	22.10.2024 / Success

Test ID	TCS1	Category	Security	Severity	Critical
Objective	Sign up for an account with a registered username				
Steps	● Open Edux. ● Direct the page to the Sign Up page using the button Sign Up. ● Enter a username that is already registered in the system. ● Click on the Sign Up button. ● Pop up a toaster that displays the error message “The username is already registered for another user.”.				
Expected	The error toaster should display the message “The username is already registered for another user.”, and the user shall not be able to sign up for an account until they write a username that is not in the User database.				
Date/Result	22.10.2024 / Success				

Test ID	TCS2	Category	Security	Severity	Critical
Objective	Sign up for an account with an invalid password				
Steps	● Open Edux. ● Direct the page to the Sign Up page using the button Sign Up. ● Enter a password not in the format of a valid password with at least eight characters up to twenty-four characters, and include at least one unique, numeric, and uppercase character in the password or check password fields. ● Click on the Sign Up button. ● Pop up a toaster that displays the error message “This is not a valid password; a valid password should contain at least eight characters up				

	to twenty-four characters and include at least one special, numeric and uppercase character.”.
Expected	The error toaster should display the message “This is not a valid password; a valid password should contain at least eight characters up to twenty-four characters and include at least one special, numeric, and uppercase character.” The user shall not be able to sign up for an account until they write a valid password.
Date/Result	22.10.2024 / Success

Test ID	TCS3	Category	Security	Severity	Critical
Objective	Sign up for an account with Check Password and Password Fields are not equivalent.				
Steps	● Open Edux. ● Direct the page to the Sign Up page using the button Sign Up. ● Enter a valid password in the password field. ● Enter a valid password in the check password field, which differs from the one entered in the password field. ● Click on the Sign Up button. ● Pop up a toaster that displays the error message “Two passwords do not match.”.				
Expected	The error toaster should display the message “Two passwords do not match.”, the user shall not be able to sign up for an account until two password fields have the same input.				
Date/Result	22.10.2024 / Success				

Test ID	TCS4	Category	Security	Severity	Critical
Objective	Sign in with an invalid mail				
Steps	● Open Edux. ● Enter an invalid mail that is not in the format of a mail or not a mail in the User database in the mail field. ● Enter a valid password in the password field. ● Click on the Sign In button. ● Pop up a toaster that displays the error message “Invalid credentials, cannot find a user that has this mail in the Edux system.”				

Expected	The error toaster should display the message “Invalid credentials, cannot find a user that has this mail in the Edux system.”, the user shall not be able to sign in for an account until the user enters a valid mail belonging to a user, provided that they enter a valid password belonging to that user.
Date/Result	22.10.2024 / Success

Test ID	TCS5	Category	Security	Severity	Critical
Objective	Sign in with an invalid password				
Steps	● Open Edux. ● Enter a valid mail in the mail field. ● Enter an invalid password, either not in the User database or the wrong format in the password field. ● Click on the Sign In button. ● Pop up a toaster that displays the error message “Invalid credentials, for the given user, a wrong password has been entered.”				
Expected	The error toaster should display the message “Invalid credentials, for the given user, a wrong password has been entered.”, the user shall not be able to sign in for an account until the user enters a valid mail belonging to a user, provided that they enter a valid password belonging to that user.				
Date/Result	22.10.2024 / Success				

Test ID	TCS6	Category	Security	Severity	Critical
Objective	Free plan users cannot use paid plan endpoints.				
Steps	● Sign in to Edux with a free plan user account. ● Create a course, create a chat. ● Send a message using a paid plan API endpoint (paid LLM APIs).				
Expected	The endpoint should refuse serving the user and return HTTP code “403 Forbidden”.				
Date/Result	Not yet.				

Test ID	TCS7	Category	Security	Severity	Critical
Objective	Unauthorized parties cannot send API requests to private interservice communication endpoints.				
Steps	● Make an API call to one of the private endpoints of any service and do not provide an API key.				
Expected	The endpoint should refuse serving the caller and return HTTP code “403 Forbidden”.				
Date/Result	Not yet.				

Test ID	TCS8	Category	Security	Severity	Critical
Objective	Unauthorized parties cannot send API requests to private interservice communication endpoints.				
Steps	● Make an API call to one of the private endpoints of any service and provide an invalid API key (the API key should not be in the accepted API keys list of the service).				
Expected	The endpoint should refuse serving the caller and return HTTP code “403 Forbidden”.				
Date/Result	Not yet.				

Test ID	TCS9	Category	Security	Severity	Critical
Objective	Private interservice communication endpoints should serve authorized services.				
Steps	● Make an API call to one of the private endpoints of any service and provide a valid API key (the API key should be in the accepted API				

	keys list of the service).
Expected	The endpoint should return HTTP code “200 OK”.
Date/Result	Not yet.

Test ID	TCS10	Category	Security	Severity	Low
Objective	Entering Invalid Email While Changing Password				
Steps	● Open Edux on the browser. ● Navigate to the “Login” page. ● At the bottom of the password section, there is “Forgot Password” click it. ● Fill in an invalid email address and click the “Reset” button.				
Expected	The error toaster should display the message “Invalid mail address.”				
Date/Result	Not yet.				

5.2.3. Usability Test Cases

Test ID	TCU0	Category	Usability	Severity	Low
Objective	Evaluate if a new user can easily sign up for an account on Edux.				
Steps	● Recruit a new user. Provide them only with a brief overview of the platform. ● Instruct the user to locate the sign-up option and create an account using valid input data. ● Ask the user to verbalize their thoughts as they navigate through the Sign In and Sign Up pages (i.e., a "think aloud" protocol). ● Record any difficulties or confusion regarding navigation and responsiveness during the account creation process. ● After the task, request feedback on the overall experience, including ease of navigation and app responsiveness.				
Expected	The user is able to locate the sign-up option and successfully create an account with minimal or no assistance. The overall experience should be reported as intuitive, with the user reporting a positive feedback regarding the app’s				

	responsiveness.
Date/Result	Not yet.

Test ID	TCU1	Category	Usability	Severity	Low
Objective	Evaluate if a new user can easily sign in for an account on Edux.				
Steps	● Recruit a new user. Provide them only with a brief overview of the platform. ● Instruct the user to locate the sign-in option for a created account. ● Ask the user to verbalize their thoughts as they try to enter the credentials(i.e., a "think aloud" protocol). ● Record any difficulties or confusion regarding navigation and responsiveness during the account creation process. ● After the task, request feedback on the overall experience, including ease of use and app responsiveness.				
Expected	The user is able to locate the sign-in option and successfully sign in with minimal or no assistance. The overall experience should be reported as intuitive, with the user reporting a positive feedback regarding the app's responsiveness.				
Date/Result	Not yet.				

Test ID	TCU2	Category	Usability	Severity	Low
Objective	Assess how intuitively a user can interpret the weekly time-spent graph.				
Steps	● Provide a user who has already spent some time in Edux, so they have analytics data. ● Instruct user to sign in to their account ● Note if they understand the bar chart without extra explanation. ● Ask them if the labeling of days/hours is clear. ● After the task, request feedback on the overall experience.				
Expected	The user should be able to read and interpret the data (e.g., quickly see how many hours they studied on a given day). No or minimal confusion about the meaning of chart.				
Date/Result	Not yet.				

Test ID	TCU3	Category	Usability	Severity	Low
Objective	Assess how intuitively a user can resume their course.				
Steps	● Ask the user to find and open one of their existing courses from the dashboard. ● Instruct user to route to the Course Dashboard through Your Individual Study container after user signed in via View Course button. ● Record any difficulties or confusion regarding navigation process. ● After the task, request feedback on the overall experience, including ease of use and app responsiveness.				
Expected	The user should be able to find the container within **5 seconds** without needing help. The user should recognize View Course as a clickable element and not be confused by the terminology. User shall successfully go to the course dashboard without assistance.				
Date/Result	Not yet.				

Test ID	TCU4	Category	Usability	Severity	Low
Objective	Assess how intuitively a user can access the chat feature from the dashboard.				
Steps	● Provide a user who has already spent some time in Edux and have chat data for a course. ● Ask the user to locate the Chat redirection card on the dashboard. ● Instruct the user to click on the Chat card to open the chat interface. ● Record any difficulties or confusion regarding the navigation process. ● After the task, request feedback on the overall experience, including ease of use and app responsiveness.				
Expected	The user should be able to find the Chat card within **5 seconds** without assistance. The user should recognize the Chat card as a clickable element without confusion regarding its purpose. The chat interface should open successfully, allowing the user to start a conversation without additional guidance.				
Date/Result	Not yet.				

Test ID	TCU5	Category	Usability	Severity	Low
Objective	Assess how intuitively a user can navigate to the Skill Tree from the dashboard.				
Steps	● Provide a user who has already spent some time in Edux and have chat data for a course. ● Ask the user to locate the Skill Tree redirection card on the dashboard. ● Instruct the user to click the Skill Tree card to navigate to the Skill Tree section. ● Record any difficulties or confusion regarding the navigation process. ● After the task, request feedback on the overall experience, including ease of use and app responsiveness.				
Expected	The user should be able to find the Skill Tree card within **5 seconds** without needing help. The card should be clearly identifiable as a clickable element that directs the user to view their skill progression. The Skill Tree section should load correctly, enabling the user to view and interact with their learning progress effortlessly.				
Date/Result	Not yet.				

Test ID	TCU6	Category	Usability	Severity	Medium
Objective	Assess how intuitively a user can utilize the Engagement Metrics feature.				
Steps	● Provide a user who has already spent some time in Edux and have chatted in a course and has quiz data provided that they have wrong answers. ● Ask the user to locate the Engagement Metrics container on the dashboard. ● Instruct the user to interact with the container (e.g., click on a recommended quiz or a detail link) to view more insights. ● Record any difficulties or confusion regarding the navigation process. ● After the task, request feedback on the overall experience, including ease of use and app responsiveness.				
Expected	The user should be able to find the Engagement Metrics container within **5 seconds** without assistance. The container should clearly indicate its interactive elements (such as recommended quizzes or detailed metrics) without confusing labels or icons. The user should successfully access and				

	interact with the detailed metrics without needing further instruction.
Date/Result	Not yet.

Test ID	TCU7	Category	Usability	Severity	Medium
Objective	Assess how intuitively a user can start and interact with the Chat feature.				
Steps	● Recruit a new user. Provide them only with a brief overview of the platform. ● Instruct the user to locate to the chat screen for a pre-created account with a pre-created existing Individual Study. ● Open a chat with pre-uploaded material for that Individual Study. ● Ask user to navigate in the slides inside of chat mode. ● Observe if the user understands how to type and send a message without assistance. ● Record any difficulties or confusion regarding message input, chatbot responses, or chat functionality. ● After the task, request feedback on the overall experience, including ease of use, clarity of responses, and app responsiveness.				
Expected	The message input field and send button should be easily recognizable and usable. The chatbot should respond appropriately to the user's question. The user should find the interaction smooth and intuitive, with minimal confusion regarding how to ask a question or interpret the chatbot's responses.				
Date/Result	Not yet.				

5.2.4. Document Test Cases

Test ID	TCD1	Category	Documentation	Severity	Medium
Objective	Verify the availability and correctness of the User Manual.				
Steps	● Navigate to the "Help" or "Documentation" section in Edux. ● Click the link for the User Manual. ● Download or open the User Manual. ● Verify that the document includes accurate descriptions of features, workflows, and images. ● Confirm that contact/support information is included.				
Expected	● The User Manual opens or downloads successfully. ● Content matches the latest system version.				

	- No broken links, grammar, or formatting issues.
Date/Result	Not yet.

Test ID	TCD2	Category	Documentation	Severity	Low
Objective	Ensure the Terms of Service document is accessible from the Sign-Up page.				
Steps	● Navigate to the Edux Sign-Up page. ● Locate the “Terms of Service” link. ● Click the link. ● Verify that the document opens and is readable.				
Expected	● Terms of Service opens without errors in a new window or tab. ● The content is readable and up-to-date.				
Date/Result	Not yet.				

Test ID	TCD3	Category	Documentation	Severity	Low
Objective	Confirm that the Privacy Policy document is available and complies with GDPR standards.				
Steps	● Navigate to the “Privacy Policy” link in the footer of the application. ● Click the link. ● Review if GDPR compliance statements are included. ● Confirm sections on data handling, user rights, and contact information.				
Expected	● Privacy Policy loads successfully. ● GDPR compliance statements are present. ● Content is accurate and clear.				
Date/Result	Not yet.				

Test ID	TCD4	Category	Documentation	Severity	High
Objective	Verify the installation guide covers all deployment methods (Docker, Kubernetes, Azure) [This is for developers only].				
Steps	● Access the Edux API documentation ● Locate and download the installation guide. ● Review sections for Docker setup. ● Review sections for Kubernetes deployment via AKS. ● Verify cloud storage (Azure Blob) instructions are clear.				
Expected	● The guide provides step-by-step instructions. ● Commands are correct and reproducible. ● Screenshots or diagrams are present.				
Date/Result	Not yet.				

Test ID	TCD5	Category	Documentation	Severity	Medium
Objective	Confirm the API documentation includes complete and accurate endpoint details [This is for developers only].				
Steps	● Access the Edux API documentation ● Check authentication endpoints. ● Verify endpoint parameters, request/response formats, and error codes are correct. ● Look for versioning information.				
Expected	● API documentation is complete, accurate, and up-to-date. ● Example requests/responses are available. ● Includes authentication/token instructions.				
Date/Result	Not yet.				

6. Consideration of Various Factors in Engineering Design

6.1. Public Health Considerations

Edux strives to elevate education through personalized learning tools, ultimately fostering better mental and intellectual well-being. While Edux is not directly involved in health practices, it contributes to mental health by reducing cognitive stress through adaptive study schedules and interactive tools. These features help learners manage their time efficiently and focus on their goals.

6.2. Public Safety Considerations

Edux prioritizes public safety by implementing robust security measures to protect sensitive user data. The platform will and should ensure that personal information, such as login credentials and uploaded resources, remains encrypted and secure. Unauthorized access is prevented through multi-layered security protocols, while compliance with KVKK, GDPR, and other data protection standards ensures that the platform respects user privacy. In addition, Edux prevents the misuse of uploaded educational materials by strictly prohibiting unauthorized sharing or publicizing of data safeguarding users and their intellectual property.

6.3. Welfare and Security Considerations

While Edux does not directly impact public welfare, it indirectly contributes by enhancing access to personalized education, fostering intellectual growth, and preparing individuals for professional success. Edux can help society in the long term by creating a better-educated workforce and reducing inequalities in access to quality learning resources.

6.4. Global Considerations

Edux is designed with global accessibility in mind. The platform defaults to English, ensuring broad reach across international audiences. Additionally, it is planned to adhere to international data protection laws such as GDPR, underscoring its commitment to global user safety. Future iterations of the platform aim to include language-specific adaptations and collaborations with global educational organizations to enhance its worldwide relevance.

6.5. Cultural Considerations

Edux acknowledges the significance of cultural diversity in education and ensures that its tools and resources are designed to support learners from diverse backgrounds. The platform strives to eliminate biases in AI-generated content, promoting inclusivity and fairness in its study materials and recommendations. Furthermore, Edux is committed to bridging gaps in access to educational resources and can provide promotional discounts or free versions for underserved communities, thereby ensuring equal learning opportunities.

6.6. Social Considerations

Edux handles sensitive user data, raising social and ethical concerns; therefore, encrypted and secure storage solutions are essential to protect privacy. The stored data in our application will not contain unique user identifiers such as names or identification numbers, ensuring an added layer of anonymity and protection. Furthermore, Edux will not use its existing user data to train machine learning and LLM models and will not employ hidden data collection mechanisms within the platform. Moreover, the medical data needed for model training and constructing a knowledge base will be collected from reputable institutions with proper permissions, emphasizing transparency and adherence to ethical standards.

6.7. Environmental Considerations

Edux strongly emphasizes environmental considerations, recognizing that errors can jeopardize patient safety and lead to wasteful resource consumption. By promptly providing users with comprehensive information, Edux aims to reduce response times and resource usage. In alignment with sustainability goals, Edux will encourage medical centers to adopt more environmentally friendly practices.

6.8. Economic Considerations

Edux offers users a wide range of functionalities; however, access to certain features will depend on the user type. Therefore, the system will adopt a tiered monetization structure based on user categories. For example, users who use the software regularly can get discounted offers. Edux will use a subscription business model, offering different subscription options for the number of patient cases to be analyzed through the platform. Additionally, some advanced features, such as skill trees, might require an extra payment. The specific subscription amounts are not determined at this stage. Furthermore, varying regional needs and the impact of inflation rates complicate establishing a universal subscription model.

Factor	Impact Level (0-10)	Effect on Edux
Public Health	2	Edux reduces cognitive stress through adaptive study schedules and interactive tools.
Public Safety	9	Edux enforces strong security measures, encryption, and compliance with GDPR/KVKK, preventing data breaches and

		ensuring user privacy.
Security & Welfare	8	Edux fosters intellectual growth and professional preparedness by enhancing access to personalized education, contributing to social welfare.
Global	10	Edux is accessible internationally, adheres to global data protection laws, and aims for multilingual support in future iterations.
Cultural	5	Edux promotes inclusivity by eliminating AI biases, ensuring fair study materials, and supporting underserved communities.
Social	10	Edux prioritizes data privacy by avoiding using user data for AI training and collecting data ethically from authorized sources.
Environmental	1	Edux seeks to reduce waste and resource use in medical centers by providing timely and comprehensive information.
Economic	9	Edux employs a tiered subscription model to ensure student affordability while monetizing advanced features for institutions.

7. Teamwork Details

7.1. Contributing and Functioning Effectively on the Team

Our team follows a structured approach to task allocation and collaboration to ensure an efficient and productive workflow. We primarily use Jira for task management and GitHub for version control, enabling us to break down complex tasks into manageable components and track individual contributions effectively. By maintaining a well-defined roadmap, we ensure that each team member has a clear understanding of their responsibilities and deadlines.

We hold weekly meetings, where each session is led by a different team member acting as the Scrum Master. This rotating leadership role allows us to evenly distribute responsibilities while allowing each member to develop leadership and facilitation skills. Meetings are conducted via Google Meet, ensuring all members stay updated on project progress, address challenges, and align on upcoming tasks.

We record meeting notes in Google Docs to maintain transparency and share them with our supervisors and coordinators. These notes are then transferred to Jira for tracking and reference, ensuring that action items are not overlooked. In addition, we actively discuss issues, share insights, and provide constructive feedback to enhance the quality of our work.

7.2. Helping Create a Collaborative and Inclusive Environment

Our team fosters a collaborative and inclusive environment by encouraging open communication and valuing each member's input. Recognizing that effective teamwork hinges on mutual respect, we emphasize equal workload distribution to prevent any member from being overburdened.

To ensure inclusivity, we encourage members to share their concerns, experiences, and development progress, creating a supportive atmosphere where everyone feels comfortable expressing their ideas. When facing technical challenges, we work together to find solutions, often pairing up to debug code, inspect pull requests, optimize algorithms, or brainstorm design choices.

By documenting key processes and creating comprehensive guides in Google Docs, we make onboarding more manageable for new contributors. These guides cover aspects such as backend integration, middleware functionalities, and best practices for our software development lifecycle.

7.3. Taking the Lead Role and Sharing Leadership on the Team

WP 1: RAG Integration	
Start Date: 3 March 2024	**End Date:** 25 Aptril 2025
Leader: Efe Kaan Fidancı	**Members Involved:** Görkem Kadir Solun
Objectives: RAG integration for specialized content generation	
Tasks: **Task 1.1** Set up RAG environment: Prepare and configure the required tools and environment for RAG integration. **Task 1.2** Develop retrieval pipeline: Design and implement the retrieval component for content generation. **Task 1.3** Integrate RAG with backend: Connect the RAG model to the existing backend infrastructure.	
Deliverables: **D1.1** Configured RAG environment ready for integration. **D1.2** Successful retrieval pipeline implementation for content generation.	

WP 2: UI Improvements	
Start Date: 17 March 2025	**End Date:** 20 April 2025
Leader: Murat Çağrı Kara	**Members Involved:** Efe Kaan Fidancı, Bilginer Oral
Objectives: Improve the user interface to enhance user experience and optimize design consistency across the application.	
Tasks: **Task 2.1** Conduct UI analysis: Analyze the current UI and gather feedback for improvements. **Task 2.2** Redesign interface components: Implement updated designs for key interface elements. **Task 2.3** Optimize responsiveness: Ensure the application UI works seamlessly across devices and screen sizes. **Task 2.4** Perform usability testing: Test the updated UI with users to identify any issues and gather feedback.	
Deliverables: **D2.1** Finalized design updates for key UI components. **D2.2** A responsive and user-tested application interface.	

WP 3: Skill Tree	
Start Date: 3 February 2025	**End Date:** 5 April 2025
Leader: Murat Çağrı Kara	**Members Involved:** Cahit Ediz Civan, Görkem Kadir Solun
Objectives: Implement skill tree functionality for dynamic user progression and customization.	
Tasks: **Task 3.1** Design skill tree structure: Develop the layout and logic for the skill tree system. **Task 3.2** Implement backend logic: Integrate the functionality to track and manage user skill progression. **Task 3.3** Develop frontend components: Create visual components to display and interact with the skill tree. **Task 3.4** Test functionality: Perform unit and integration tests to ensure the skill tree works as intended.	
Deliverables: **D3.1** A functional skill tree system integrated with the application. **D3.2** Completed testing and validation of skill tree features.	

WP 4: Admin Dashboards	
Start Date: 17 March 2025	**End Date:** 20 April 2025
Leader: Cahit Ediz Civan	**Members Involved:** Efe Kaan Fidancı
Objectives: Develop dashboards for administrators to monitor, manage, and analyze user activities and system performance.	
Tasks: **Task 5.1** Design dashboard layout: Plan and design user-friendly interfaces for both admin dashboards. **Task 5.2** Implement data visualization: Integrate visual components to display key metrics and performance indicators. **Task 5.3** Add user management features: Develop tools for managing users, roles, and permissions. **Task 5.4** Integrate reporting tools: Provide functionality for generating and exporting system activity reports.	
Deliverables: **D5.1** Fully functional admin dashboards. **D5.2** Visualized metrics and reporting tools integrated into the system. **D5.3** User management interface with role-based access controls.	

WP 5: Profile	
Start Date: 17 March 2025	**End Date:** 6 April 2025
Leader: Görkem Kadir Solun	**Members Involved:** Bilginer Oral
Objectives: Create profile to manage, username, mail, passwords and subscription tier plan.	
Tasks: **Task 5.1** Design profile layout: Plan and design user-friendly interfaces for profile. **Task 5.2** User profile backend: Develop backend for managing users to change their username, mail, password and subscription tier.	
Deliverables: **D5.1** Fully functional profile interface. **D5.2** All profile features are tested and bug free.	

WP 6: Chat Improvements	
Start Date: 17 March 2025	**End Date:** 6 April 2025
Leader: Bilginer Oral	**Members Involved:** Murat Çağrı Kara
Objectives: Develop the chat interface to be more user friendly and add new features.	
Tasks: **Task 5.1** Have chat the functionality of mini preview when user tries to switch between the slides easily. **Task 5.2** Integrate the user dashboard chat card to move to a generic page that you can route to chats within a selected course. **Task 5.3** Update the chat sidebar to be compatible with the skill tree sidebar's design.	
Deliverables: **D5.1** Fully functional chat interface. **D5.2** All chat features are tested and bug free.	

Leadership within our team is shared among all members through the rotating Scrum Master role during our weekly meetings. This approach ensures that each team member gains experience directing discussions, prioritizing tasks, and facilitating decision-making. Moreover, as it can be seen there are work packages under the lead of different members. Thus, this create an environment where everybody works as a lead at some point in our development process.

Aside from formal leadership, members take initiative based on their strengths and expertise. For instance, those with stronger backend development skills may lead discussions on database optimization, while others proficient in frontend development guide UI/UX decisions. By recognizing individual strengths and allowing members to take ownership of specific areas, we enhance productivity and knowledge sharing within the team.

Overall, our team's approach to collaboration, leadership, and structured task management ensures that we work efficiently while fostering an environment of mutual respect and support.

8. Glossary

AKS (Azure Kubernetes Service)

Azure Kubernetes Service (AKS) is a managed Kubernetes service provided by Microsoft Azure. It simplifies the deployment, scaling, and management of Kubernetes clusters, integrating with Azure's security, monitoring, and networking features [8], [9].

API (Application Programming Interface)

An Application Programming Interface (API) is a set of protocols and tools that allow different software applications to communicate with each other. APIs enable developers to access specific functionalities or data from external services without understanding their internal workings.

Azure Blob Storage

Azure Blob Storage is Microsoft’s cloud-based object storage solution designed for unstructured data such as images, videos, backups, and logs. It provides high availability, scalability, and security for storing large amounts of data [10].

D

Docker

Docker is an open-source platform that enables developers to automate the deployment of applications within lightweight, portable containers [11]. These containers package the application along with their dependencies, ensuring consistency across different environments.

G

GDPR (General Data Protection Regulation)

The General Data Protection Regulation (GDPR) is a comprehensive data protection law enacted by the European Union to safeguard individuals' data and privacy. Implemented in 2018, it imposes strict guidelines on data collection, processing, and storage, with significant penalties for non-compliance [12].

H

HTTPS (HyperText Transfer Protocol Secure)

HTTPS is an internet communication protocol that ensures secure data transfer between a user’s web browser and a website. It is an encrypted version of HTTP (HyperText Transfer Protocol) and is widely used to protect sensitive information such as login credentials, payment details, and personal data from cyber threats.

I

ID (Identifier)

An Identifier (ID) is a unique symbol or sequence assigned to an entity to distinguish it from others. IDs are essential in databases, programming, and various systems to efficiently reference and manage specific records or objects.

K

Kubernetes (K8s)

Kubernetes is an open-source system for automating the deployment, scaling, and management of containerized applications. It provides load balancing, service discovery, and self-healing capabilities, making it ideal for managing large-scale containerized workloads.

L

LLM (Large Language Model)
A Large Language Model (LLM) is an advanced artificial intelligence system trained on extensive text datasets to understand and generate human-like language. These models are pivotal in natural language processing tasks, including text generation, translation, and sentiment analysis.

M

MySQL
MySQL is an open-source relational database management system (RDBMS) that uses SQL for accessing and managing data. It is widely used for web applications, data storage, and processing large-scale structured data.

R

RAG (Retrieval Augmented Generation)
Retrieval Augmented Generation (RAG) combines large language models with external data sources to enhance the accuracy and relevance of generated content. RAG systems can produce more informed and contextually appropriate responses by retrieving pertinent information from databases or documents.

S

SHA-256 (Secure Hash Algorithm 256-bit)
SHA-256 is a cryptographic hash function used widely in various security applications and protocols, including SSL certificates and blockchain technology. It provides a fixed-size 256-bit (32-byte) hash that is nearly unique for different inputs and is used for ensuring data integrity.

T

TPS (Transactions Per Second)
Transactions Per Second (TPS) is a metric used to measure the number of transactions a system can process in one second. It's commonly used to assess the performance and scalability of databases, networks, and other transactional systems.

9. References

[1] Facebook, Inc. (n.d.). React - A JavaScript library for building user interfaces. [Software].

Retrieved from https://reactjs.org

[2] IBM. “What is a REST API?” [Software]. Retrieved from https://www.ibm.com/think/topics/rest-apis

[3] MySQL. [Software]. Retrieved from https://www.mysql.com/

[4] JWT.io. JSON Web Tokens. Available at: https://jwt.io/ (Accessed: March 9, 2025).

[5] OpenAI. GPT-4 Technical Report. Available at: https://openai.com/research/gpt-4 (Accessed: March 9, 2025).

[6] Google AI. Gemini 1.5 Technical Overview. Available at: https://deepmind.google/technologies/gemini/ (Accessed: March 9, 2025).

[7] Anthropic. Claude AI Language Model Overview. Available at: https://www.anthropic.com/index/claude (Accessed: March 9, 2025).

[8] Kubernetes. Production-Grade Container Orchestration. Available at: https://kubernetes.io/ (Accessed: March 9, 2025).

[9] Microsoft Azure. Azure Kubernetes Service (AKS). Available at: https://azure.microsoft.com/en-us/products/kubernetes-service/ (Accessed: March 9, 2025).

[10] Microsoft Azure. Azure Blob Storage. Available at: https://azure.microsoft.com/en-us/products/storage/blobs/ (Accessed: March 9, 2025).

[11] Docker. Empowering App Development for Developers. Available at: https://www.docker.com/ (Accessed: March 9, 2025).

[12] European Union. General Data Protection Regulation (GDPR). Regulation (EU) 2016/679. Available at: https://gdpr.eu/ (Accessed: March 9, 2025).