AI4ED

TOWARDS AN AI DRIVEN EDUCATIONAL PROCESS INTEGRATING MODERN CAREERS IN THE EDUCATIONAL SYSTEM

Deliverable

D6.2 – Report on validation trials and recommendation for future work

Deliverable Lead: AlchemyML
Deliverable due date: 13/05/25
Actual submission date: 28/05/25
Dissemination level: PU
Version: COMPLETE

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Document Control Page	
Title	Report on validation trials and recommendations for future work
Creator	AlchemyML
Description	This document describes the history of developing and implementing the chatbot and summarises “lessons learnt” from the different use cases.
Contributors	AlchemyML
Creation date	13/05/2025
Type	Report
Language	English
Audience	☒ public ☐ confidential
Review status	☐Draft ☐ Assigned Reviewer accepted ☒ Coordinator accepted
Action requested	☐ to be revised by the Assigned Reviewer ☐ for approval by the Project Coordinator ☐ for acknowledgement by Partners

Revision history

Version	Author(s)	Changes	Date
V0	AlchemyML	Document creation	13/05/2025
V1	Uni Bremen, ALL	Review	19/05/2025
V2	AlchemyML	Final changes	28/05/2025

Table of contents

1 Introduction

The AI4ED project is a forward-looking initiative that explores the integration of generative artificial intelligence into digital learning environments to support and enhance student experiences. As the educational sector continues to evolve with the adoption of digital tools, AI4ED investigates how AI-powered conversational agents -commonly referred to as chatbots- can deliver targeted, personalized, and scalable academic support. The project particularly focuses on improving student outcomes in Moodle-based learning systems, one of the most widely adopted learning management platforms globally.

Rooted in both pedagogical theory and technical innovation, AI4ED seeks to bridge the gap between instructional content and learner needs through real-time, intelligent assistance. By embedding a generative AI chatbot within Moodle courses, the project aspires to offer learners timely guidance, reinforce learning through personalized testing, and adapt interactions to diverse student contexts – all while maintaining rigorous ethical standards and transparency.

2 Document objectives

This document serves as a comprehensive technical and pedagogical overview of the AI4ED project. Its objectives are to:

1. **Explain the strategic evolution** of the project's AI approach, from predictive analytics to chatbot-based support.
2. **Detail the data handling methodology**, including how instructional content is collected, processed, and structured to build the chatbot's knowledge base using Retrieval-Augmented Generation (RAG).
3. **Describe the logic and behavior** of the AI-powered chatbot, highlighting its dual-mode functionality -Solution Guider and Learning Reinforcer- and its intent-based conversation flow.
4. **Outline the technical architecture**, including the Moodle plugin, backend services, and integration with OpenAI's generative models.
5. **Propose enhancements and future developments**, covering proactive support, contextual awareness, educator analytics, and model scalability.
6. **Offer actionable recommendations** for educators on how to meaningfully integrate and leverage the chatbot to improve teaching and learning outcomes.

Through these goals, the document not only illustrates the current implementation of the AI4ED system but also serves as a roadmap for its continuous development and pedagogical alignment.

3 AI development

The development of AI modules within the AI4Ed project—particularly those focused on personalised tutoring, active learning, and dropout prevention—requires a solid foundation rooted in machine learning (ML) methodologies. Unlike classical rule-based systems, ML-based models rely on data-driven learning processes, which require a series of technical, pedagogical, and organisational prerequisites to be in place.

First and foremost, the development of ML-based AI modules depends on the availability of high-quality, well-structured data. In the educational context, this includes learner interaction data (e.g., time-on-task, quiz results, forum engagement), emotional and motivational indicators (if available), and contextual variables (such as course content or instructional strategies). This data must be gathered ethically and in compliance with GDPR, and its scope must be sufficient to represent the diverse behaviours and learning patterns of different students. Typically, a minimum dataset in the range of several hundred to thousands of examples is required to develop meaningful predictive models. When the dataset is too small—for instance, fewer than 30 examples—ML models risk overfitting, providing unreliable or non-generalisable insights.

Second, data must be cleaned, annotated, and transformed into structured formats suitable for ML processing. This involves feature engineering, where raw input (such as timestamped logs or qualitative feedback) is converted into variables that capture learning-relevant signals. For example, the frequency of quiz attempts or time spent on a module can be turned into numerical features representing engagement or persistence.

Third, the development of AI4Ed's ML modules also requires a clear pedagogical framework to guide model training. This is articulated through the three core models outlined in the project: the pedagogical model (how learning is structured and assessed), the domain model (the knowledge being taught), and the learner model (individual learner profiles and behaviours). These models inform the design of training targets—e.g., predicting risk of dropout or identifying students who may benefit from remedial content—and ensure that the AI operates in service of human-centric, educationally meaningful goals.

Following the work carried out during the first half of the project, initial datasets were collected from the various use cases and pilot sites. These datasets were processed and transformed into structured formats aligned with the project's learning analytics and pedagogical models, making them technically ready for ML processing. However, the actual number of usable examples remained below 20 per use case, which is significantly below the threshold required for meaningful machine learning.

At the same time, in November 2022—shortly after the AI4Ed project began—the release of ChatGPT marked a major leap forward in the performance and accessibility of generative AI. Having experienced firsthand the limitations in gathering sufficient training data for ML and observing the rapid maturity and reliability of generative AI technologies, the consortium made a strategic decision to shift focus. Rather than persisting with traditional ML approaches that were no longer viable within the project's timeframe and data constraints, it was decided to develop the three AI4Ed modules (on personalised tutoring, active learning, and dropout prevention) using generative AI technologies. This new approach has allowed the project to maintain its educational ambition while leveraging AI capabilities to support learners and educators in real time.

To operationalise this shift, each module was developed using a generative AI model (based on the GPT architecture) that was **specifically fine-tuned or configured using curated datasets and context-specific prompts**. For the *personalised tutoring* module, the GenAI system was trained on course-specific instructional materials, assessment rubrics, and examples of teacher feedback, enabling it to generate adaptive explanations, scaffold learning tasks, and answer student questions in line with pedagogical goals. The *active learning* module was built around dynamic dialogue prompts and interactive learning strategies, encouraging learners to reflect, debate, and explore concepts through AI-mediated conversations. For the *dropout prevention* module, the GenAI assistant was configured to detect signals of disengagement (e.g., low

activity or repeated errors), offer motivational feedback, and proactively recommend supportive resources or learning pathways. All modules were designed with safeguards to ensure alignment with ethical and educational principles, including transparency, non-bias, and learner agency.

This generative AI-based approach enables personalised, context-aware, and pedagogically grounded learner support without requiring the scale of training data that traditional ML models demand, making it a sustainable and innovative solution within the AI4Ed ecosystem.

4 Data collection

In compliance with data protection regulations and to ensure full alignment with ethical standards, the **AI4ED project does not collect or process any personal data from students**. The chatbot's development and operation are entirely based on course-related instructional materials, avoiding the need for any sensitive or identifying information.

The primary goal of data collection in the AI4ED project is to **assemble a comprehensive knowledge base** that allows the AI-powered chatbot to function as a domain expert. This knowledge enables the chatbot to support students effectively through **Retrieval-Augmented Generation (RAG)** techniques. In this architecture, a language model generates answers based on real-time retrieval of relevant course content, ensuring factual accuracy and contextual relevance.

To achieve this, the project team gathered **all instructional content** from the Moodle-hosted courses involved in the pilot. The materials include a wide variety of formats, such as:

- Plain text documents (PDFs, Word files)
- Instructional videos and recorded lectures
- Images and diagrams

Since not all this content was originally in a text-readable format, a **multi-tool extraction strategy** was adopted to process and standardize the data.

Various **text extraction tools** were employed to convert multimedia and visual content into usable textual data. This included:

- **Optical Character Recognition (OCR)** for extracting text from images and diagrams
- **Speech-to-text transcription** for converting audio from video lectures

After extraction, all textual content was **cleaned, standardized, and segmented** to optimize its usability in the RAG pipeline. Redundant or irrelevant information (e.g., slide formatting details, filler speech in videos) was excluded during this phase.

To maintain pedagogical coherence and contextual traceability, **every piece of extracted knowledge was linked to its corresponding module in the Moodle course**. This tagging ensures that the chatbot can retrieve module-specific answers, thereby preserving the alignment between the AI's responses and the course's instructional design.

Each content fragment is stored with metadata that includes:

- The original file or source type
- The module identifier
- The topic or learning objective (where applicable)

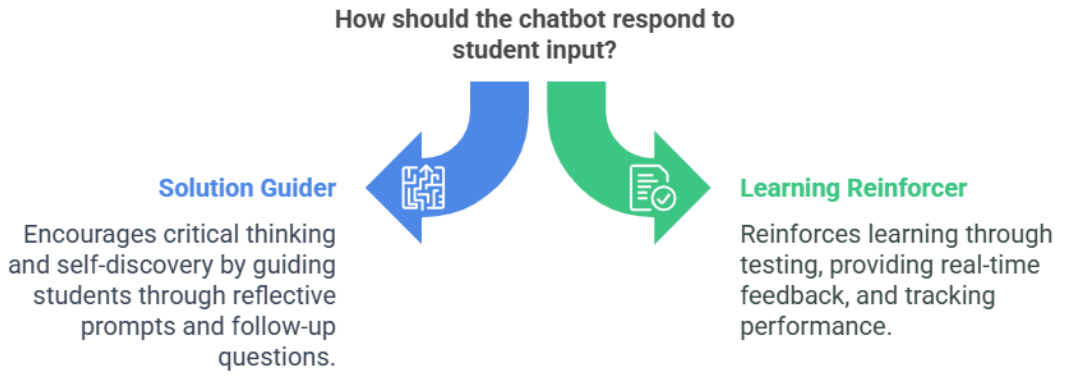
This structure enables granular retrieval of knowledge, enhancing the chatbot's ability to provide **targeted, accurate, and context-sensitive support** to students.

5 Chatbot logic behaviour and conversation flow

The AI-powered chatbot developed in the AI4ED project operates based on a **multi-intent logic system** designed to enhance student engagement and promote deeper learning. Its behavior is governed by a conversational flow that intelligently adapts to the nature of the student's input, prioritizing pedagogical value and personal interaction.

The chatbot integrates **two primary pedagogical functions**, each serving a distinct learning purpose:

1. **Solution Guider**
2. **Learning Reinforcer**



○ Solution Guider

The **Solution Guider** function is triggered whenever a student submits a question related to the course subject matter. Rather than offering a direct answer, the chatbot's response is designed to **encourage critical thinking and self-discovery**. It guides the student through reflective prompts, follow-up questions, or hints, aiming to lead them toward constructing the answer independently.

This approach fosters:

- Active engagement with the learning material
- Development of problem-solving and reasoning skills
- Improved long-term retention of knowledge

○ Learning Reinforcer

The **Learning Reinforcer** functionality is activated when a student explicitly requests to be tested. In this mode, the chatbot:

- **Selects a question** from a curated stack of assessment items
- **Presents it** to the student
- **Evaluates** the student's response in real time
- **Logs the outcome** (correct/incorrect) into its internal performance tracking database
- **Delivers targeted feedback**, explaining why the answer was correct or how it could be improved

This testing loop supports formative assessment by reinforcing learning, highlighting gaps, and encouraging active recall — a key component in cognitive learning theory.

○ Conversation flow and functional routing

Each time a new message is received from a user, the chatbot performs an **intent classification step** to determine which of the following states to activate:

- **Solution Guider**
- **Learning Reinforcer**
- **Fallback (Small Talk / Off-topic / Idle State)**

If the message does not correspond to either of the two educational functions (e.g., simple greetings, off-topic remarks, or inappropriate content), the chatbot transitions into a **fallback mode**. In this case, it issues a **brief, polite response** that:

- Clarifies its intended purpose (e.g., helping with course questions or testing knowledge)
- Prompts the student to choose between asking a question or requesting a test

This fallback ensures that interactions remain focused and meaningful, while still allowing for a friendly and human-like experience.

○ Language handling adaptability

The chatbot was designed to always **respond in the instructional language of the course**, ensuring consistency with the educational material. However, it has the **capability to understand input in any language**, allowing multilingual students to interact naturally. This cross-linguistic understanding enhances accessibility and inclusiveness without compromising the course's linguistic integrity.

6 Question stack for learning reinforcement

A key component of the chatbot's **Learning Reinforcer** functionality is the **question stack**—a curated and dynamic bank of assessment questions designed to test students' understanding and reinforce learning across the course content. This structured repository enables the chatbot to provide meaningful and contextually aligned testing experiences tailored to each student's progression.

○ Question stack generation

The initial version of the question stack is **automatically generated by a generative AI model**. This model uses the same **course knowledge base** that was compiled during the data collection phase, ensuring that all generated questions are grounded in the specific educational content of the course.

Once the draft question stack is generated, it undergoes a **manual review and curation process** carried out by the **course owners or instructors**.

Their responsibilities include:

- **Removing** questions deemed inappropriate, misleading, or redundant
- **Rewording** questions to improve clarity, tone, or difficulty
- **Adding** new questions that might have been overlooked by the AI
- **Reducing** the number of questions if the stack is considered too extensive for the course structure

This human-in-the-loop refinement ensures that the final question set is pedagogically sound, course-specific, and aligned with the instructor's teaching objectives.

Each question in the stack is enriched with **metadata**, including, a unique identifier, the **module or content ID** from which the answer can be derived and the identifier of the specific piece of knowledge the answer lives in.

By linking questions to specific content items in the Moodle course, the chatbot can **efficiently locate reference material** during answer evaluation, improving both accuracy and response time.

Once finalized, the validated question stack is **stored in a structured database**, where it becomes accessible to the chatbot's Learning Reinforcer logic. The stack is **shared across all students enrolled in the same course**, maintaining consistency in assessment coverage while allowing for personalized delivery.

○ Question selection and answer evaluation

The chatbot employs a **smart selection algorithm** to determine which question to ask during each reinforcement interaction. This algorithm considers several factors:

1. **Course Progress**: Only questions from **modules the student has already started** in Moodle are eligible for selection.
2. **Performance History**: Priority is given to questions the student has **previously answered incorrectly**, enabling targeted reinforcement of weak areas.
3. **Recency and Novelty**: The system favors:
    - Questions that have **never been asked** to the student
    - Questions that have **not been asked for a long time**

This selection strategy supports **spaced repetition**, personalized remediation, and comprehensive content coverage over time.

Each time a question is posed:

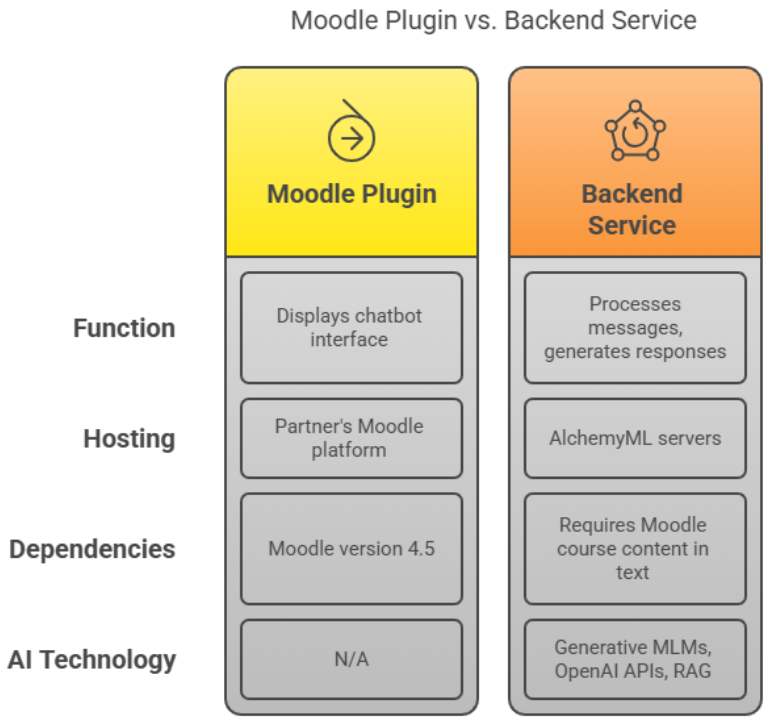
- A **record is created in the database** linking the question, student, and timestamp.
- After the student responds, the chatbot **evaluates the answer** using the reference content ID.
- The system then **updates the record** with the result (e.g., correct, incorrect, partially correct) and provides **constructive feedback** to the student.

This process allows the project to monitor progress, analyse learning patterns, and continuously optimize reinforcement strategies.

7 Technical aspects of the development

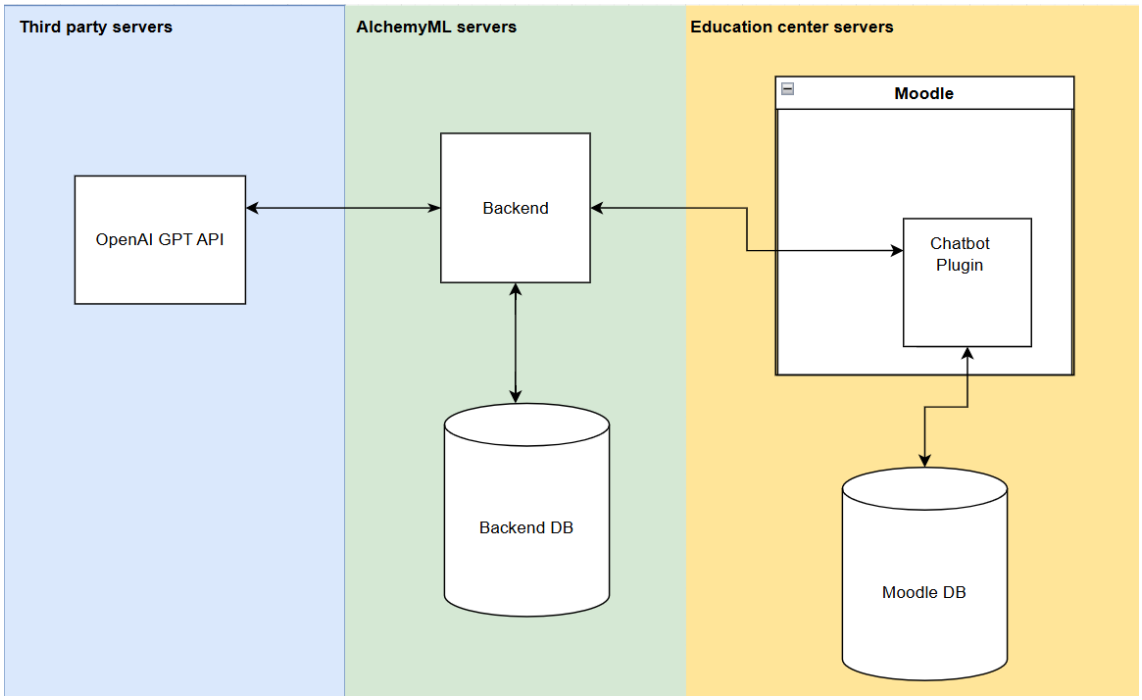
The chatbot system is composed of two main components: the **Moodle plugin** and the **backend service**.

- The **Moodle plugin** is installed on each partner's Moodle platform. Its primary function is to display the chatbot interface to students and manage communication with the backend service.
- The **backend service**, hosted on **AlchemyML servers**, is responsible for processing all incoming student messages and generating appropriate responses.



The Moodle plugin is **compatible with Moodle version 4.5** and must be configured within a course that has been previously enabled on the backend. This setup requires a **Moodle course code**, which is predefined in coordination with AlchemyML.

The AI powering the chatbot uses **large language models (LLMs)**, specifically leveraging **OpenAI APIs**. It also incorporates **Generative AI techniques**, such as **Retrieval-Augmented Generation (RAG)**, to provide context-aware and pedagogically relevant responses.



○ Moodle plugin

Moodle, as a widely adopted open-source learning management system, provides a flexible infrastructure for enhancing its core functionality through the use of **plugins**. These modular extensions allow institutions to customize user experience, integrate external tools, and introduce new pedagogical features. Plugins can be developed in various programming languages supported by Moodle, including **JavaScript**, which enables real-time, interactive components to be embedded in the platform.

○ Plugin development and installation

To integrate the AI chatbot developed within the AI4ED project into the students' learning environment seamlessly, a **custom Moodle plugin** was designed and implemented. The plugin is **fully compatible with Moodle version 4.5** and is intended to ensure both **high usability** and **minimal disruption** to the core learning experience.

Once enabled in a course's configuration settings, the chatbot appears as a **floating chat window** anchored to the **bottom-right corner** of the Moodle interface. This positioning ensures that the chatbot is **easily discoverable and accessible**, while being carefully designed not to interfere with students' engagement with other course materials.

Key design features include:

- A clean and minimalistic **chat-style interface**
- Behavior tuned to be **encouraging but non-intrusive**

The plugin operates through a **request-response mechanism**. Each time a student sends a message via the chatbot interface, the plugin:

1. **Sends a request** to the AI4ED **Backend API**, containing the student's input and session context
2. **Receives the chatbot's response** from the backend
3. **Renders the reply** in the chatbot window in real time

This architecture ensures that all the logic, knowledge processing, and response generation occur on the server side, keeping the client-side plugin lightweight and responsive.

For the chatbot to become active in a Moodle course, the plugin must be installed and **explicitly enabled by the instructor or administrator** in the course that is intended to offer the chatbot.

○ Ethical transparency and disclaimer

The plugin includes a built-in **disclaimer** prominently displayed at the top of the frame of the conversation window. This message informs users that:

- They are interacting with an **artificial intelligence system**
- Their **conversations will be stored** for educational and system improvement purposes

This transparency aligns with ethical guidelines and data responsibility principles adopted throughout the project.

○ Backend API

The backend of the AI4ED chatbot system plays a central role in managing logic, interaction flow, data persistence, and integration with AI services. It is built using robust and scalable technologies to ensure smooth performance, maintainability, and pedagogical accuracy.

○ Architecture and technology stack

The backend is implemented in **Python** using the **Django REST Framework**, a widely adopted toolkit for building web APIs that follow RESTful design principles. This framework enables clear separation of logic, security, and scalability in the service architecture.

The backend uses **PostgreSQL** as its database engine, ensuring reliable storage and retrieval of structured data such as:

- User interactions
- Question stack records
- Evaluation results
- Session and metadata

This relational database system is chosen for its performance, reliability, and compatibility with Django-based applications.

○ Database information structure

The most relevant tables in this database are centered around course information and content, and student interaction. The schema is organized to support the tracking of user engagement, content segmentation, question-based assessments, and course metadata. Key entities include api_coursesinformation for general course data, api_interaction for logging dialogue between users and the system, api_piecesofcontent for storing content fragments, api_questions for question mapping, and api_studentsquestions for student performance tracking in the learning reinforcer feature.

1. Table: api_coursesinformation

Stores general information about the available courses.

Column	Type	Description
id	int4	Primary key. Unique identifier for the course.
moodle_code	int4	Course code used in Moodle.

Column	Type	Description
course_id	int4	Internal course identifier.
disclaimer	text	Legal or instructional disclaimer text.
language	text	Language of the course (e.g., 'en', 'es', 'pt').
description	text	General description of the course.

2. Table: api_interaction

Tracks all interactions between users (students) and the system/chatbot.

Column	Type	Description
id	int4	Primary key. Unique ID for the interaction.
student_id	int4	ID of the student involved.
session_id	text	Identifier for the interaction session.
message	text	Message content sent by either user or system.
date	timestamptz	Timestamp of when the interaction occurred.
is_human	bool	Indicates whether the message was written by a human (true) or bot (false).
course_id	int4	Associated course ID.
moodle_code	int4	Moodle course code.
is_question	int4	Marks if the message contains a question (likely boolean or question ID).

3. Table: api_piecesofcontent

Contains course content broken down into smaller chunks or pieces.

Column	Type	Description
id	int8	Primary key. Unique ID for the content piece.
content_id	int4	ID of the parent content this chunk belongs to.
moodle_code	int4	Moodle course code.
course_id	int4	Associated course ID.
module_id	int4	ID of the module this piece belongs to.
chunk_text	text	Actual text of the content chunk.

Column	Type	Description
url	text	Optional: URL pointing to the full content or resource.
embedding	public.vector	Vector representation (embedding) for NLP/AI tasks.

4. Table: api_questions

Stores questions related to course content.

Column	Type	Description
id	int4	Primary key. Unique ID for the question.
moodle_code	int4	Moodle course code.
course_id	int4	Associated course ID.
module_id	int4	ID of the module this question belongs to.
piece_of_content_id	int4	Reference to the content piece the question is about.
question	text	The actual question text.

5. Table: api_studentsquestions

Stores student performance data for each question if the learning reinforcer.

Column	Type	Description
id	int4	Primary key. Unique ID of the record.
student_id	int4	ID of the student.
times_answered	int4	Number of times the student attempted the question.
score	int4	Score obtained by the student.
last_time	timestamptz	Timestamp of the last attempt.
question_id	int4	Reference to the question being answered.

○ Integration with generative AI and functional logic

The backend integrates with **OpenAI's GPT-4o-mini model**, which is responsible for powering three key cognitive functions of the chatbot:

1. **Function Selection:** Upon receiving a user message, the model interprets the input to determine whether to activate the **Solution Guider**, **Learning Reinforcer**, or **Fallback** functionality.

2. **Answer Evaluation:** In the case of Learning Reinforcer sessions, the model evaluates the student's response against the correct content and provides feedback.

3. **Response Generation:** The model generates engaging, pedagogically aligned chatbot responses based on the selected functionality and user input.

To enhance the contextual accuracy of the responses, the backend implements a **Retrieval-Augmented Generation (RAG)** pipeline. This mechanism allows the model to retrieve relevant course content dynamically, based on the topic of the student's query or the expected answer to a test question. The retrieved knowledge is appended to the prompt before being sent to the language model, ensuring that the chatbot's output is factually grounded and course-specific.

○ Technical improvements for the future

In the ongoing effort to evolve the AI4ED platform and ensure its long-term relevance, several technical enhancements have been identified to enrich the chatbot's capabilities, improve user experience, and empower educators with greater control and insight. These future developments are designed not only to address current limitations but also to align the system with pedagogical best practices and the rapid evolution of generative AI technologies.

1. Structural awareness and course navigation support

A recurring student need observed during pilot deployments is the desire for **clarity about course structure and navigation**. Currently, the chatbot specializes in content support and knowledge reinforcement, but lacks awareness of the course's instructional layout. A planned improvement involves enabling the chatbot to respond to **structural queries**, such as:

- "What do I need to complete before this module?"
- "Where can I find the assessment for this topic?"
- "Is this page part of the final exam content?"

To support this, the system should be enhanced with **structural metadata parsing**, allowing the chatbot to access and communicate the hierarchical organization of each Moodle course.

2. Proactive learning reinforcement

The current Learning Reinforcer functionality relies on the student **explicitly requesting to be tested**. However, educational research suggests that **spaced and unsolicited questioning** can deepen engagement, enhance retention, and encourage metacognitive reflection. To this end, the chatbot could be redesigned to:

- **Proactively offer knowledge-check questions** at natural breaks in conversation or after providing guidance
- Tailor the timing and difficulty of questions based on recent interaction history and prior performance

This shift aims to **transform the chatbot from a reactive assistant into an active learning companion**, capable of dynamically reinforcing knowledge without waiting for user prompts.

3. Context-aware interaction

Understanding the student's context is essential for accurate and helpful responses—particularly when they use deictic expressions like "this page" or "here." A planned enhancement involves enabling the chatbot to **detect the current page or resource** the student is viewing in Moodle. By integrating this context-awareness into the backend request pipeline, the chatbot will be able to:

- Interpret vague or location-based queries more precisely
- Automatically link questions to the content currently being studied
- Provide smoother, more personalized interaction

4. Teacher-facing analytics and dashboard

To maximize the value of chatbot interactions, a new **teacher-focused plugin** is planned. This plugin will provide **insights and visual statistics** based on aggregated chatbot activity. Key features may include:

- Most frequently asked questions
- Topics with the highest incorrect response rates
- Student engagement metrics over time
- Summary of reinforcement attempts and success rates

These insights will enable educators to **fine-tune course content**, identify learning gaps, and better understand how students interact with the material outside of formal assessments.

5. Instructor empowerment in chatbot configuration

To reduce dependency on developers and increase flexibility for educators, a key improvement involves enabling **course owners to manage chatbot knowledge and deployment autonomously**. Future updates will introduce tools for teachers to:

- **Upload, edit, and tag knowledge materials** directly through the Moodle interface
- **Enable or disable the chatbot** for specific courses
- **Trigger question stack regeneration** when course content changes

This will create a **self-service framework** where the pedagogical responsibility of keeping the chatbot updated lies in the hands of those closest to the students.

6. Future-proofing through AI model agility

As generative AI models continue to evolve at an accelerated pace, it is essential to **maintain technological adaptability** to ensure that the chatbot does not become obsolete. The backend architecture has been intentionally designed to be **model-agnostic**, allowing for easy integration of newer, more capable models as they become available.

Future work will involve:

- **Monitoring emerging models** for improved contextual reasoning, multilingual performance, and cost-efficiency
- **Benchmarking replacements** for GPT-4o-mini against project-specific criteria
- **Implementing modular upgrades** with minimal disruption to existing deployments

This proactive strategy ensures that AI4ED remains at the cutting edge of educational technology and continues to deliver high-quality, relevant support to learners across evolving pedagogical contexts.

8 Recommendations for educators

While the integration of AI tools like chatbots into education offers promising potential, maximizing their impact requires more than just technical deployment. It involves thoughtful collaboration between technology and pedagogy. The following reflections highlight key considerations that go beyond implementation—emphasizing the essential role of educators, the importance of understanding student interactions, and the untapped potential of data-informed decision-making.

The responsibility of leveraging AI to add value to education lies not only in its technological implementation, but also in how it is used. In the educational context, **educators play a crucial role in providing the human dimension**—ensuring that AI complements rather than replaces their guidance.

To make the most of the chatbot, it is recommended that teachers stay informed about the conversations their students are having with it. This allows them to **identify common areas of confusion or difficulty** and adjust their teaching approach accordingly.

It is also important to recall why, in this project, the initial idea of implementing **predictive analytics** was discarded: the lack of historical data, which is essential for generating meaningful predictions. However, today there are numerous **no-code tools** available that, with minimal effort, can deliver valuable insights into both institutional performance and student learning behaviours.