

heyprof – Eine KI-gestützte Plattform für die Lehre

Architektur, Funktionen und didaktische Grundlagen

Volker Reichenberger

Dirk Schieborn

23. September 2026, Version 2.0

heyprof ist eine webbasierte Plattform für die Lehre, in deren Mittelpunkt ein KI-Tutor steht, der nicht aus dem Weltwissen eines Sprachmodells antwortet, sondern aus dem Kurs: aus Folien, Skript und Vorlesungsmitschnitt der Lehrkraft, aus der aktuellen Aufgabe samt Musterlösung und privaten Hinweisen und aus dem bisherigen Lösungsweg der Studierenden. Er führt sokratisch, statt Lösungen zu liefern, und bleibt auf dem Niveau des Kurses. Zweiter Baustein ist eine Bibliothek von über 590 interaktiven Visualisierungen (VisuApps), die der Tutor selbst vorführen kann. Dazu kommt der vollständige Lebenszyklus einer Lehrveranstaltung: Materialien, Übungen mit automatischer Rückmeldung, Tests, Probeklausuren mit handschriftlicher Abgabe, Aufnahmen, Live-Quiz und Community. heyprof wurde von den Autoren entwickelt und im Sommersemester 2026 als experimenteller Prototyp in der Lehre eingesetzt. Dieses Dokument beschreibt die Architektur, die Funktionen für Lehrende und Lernende, die VisuApps, die KI-Integration, die didaktischen Prinzipien, den Datenschutz und den Stand des Einsatzes.

1 Einleitung

Große Sprachmodelle sind in der Lehre angekommen, meist in Gestalt allgemeiner Assistenten, die den Kurs nicht kennen. Daraus folgen zwei Probleme. Das erste ist der *cognitive bypass*:¹ Die Lernenden lassen sich die Aufgabe lösen, statt sie zu verstehen. Das zweite ist die fehlende Passung: Die Antwort kennt den Stand der Lehrveranstaltung nicht, ist zu komplex oder verwendet Konzepte, die im Kurs noch nicht eingeführt sind. Ein typisches Beispiel sind Lösungen zu Programmieraufgaben mit Sprachmitteln, die nicht Teil der Veranstaltung waren, etwa Listenkomprehensionen oder Typhinweise.

heyprof setzt genau hier an. Der KI-Tutor erhält zu jeder Frage den Kontext des Kurses: die relevanten Passagen aus Skript und Folien, die aktuelle Aufgabe mit Musterlösung und privaten Tutorhinweisen der Lehrkraft, den bisherigen Code der Studierenden und, auf Wunsch, das Transkript der mitgeschnittenen Vorlesung. Er gibt zunächst *keine* vollständigen Lösungen,

¹Eine technische Hilfe übernimmt die kognitive Arbeit vollständig; der Lerneffekt entfällt.

sondern führt mit Fragen und Hinweisen, die auf das Niveau des Kurses abgestimmt sind, und erklärt mit den Beispielen und Metaphern, die im Hörsaal verwendet wurden.

Seit der ersten Fassung dieses Whitepapers (März 2026) ist die Plattform in drei Richtungen gewachsen. Erstens um eine Bibliothek interaktiver Visualisierungen, die *VisuApps*, mit über 590 Apps in 66 Fachsammlungen, die der Tutor fernsteuern und vorführen kann. Zweitens um den Einsatz in Grundschule und betrieblicher Weiterbildung, mit jeweils eigenem Regelwerk. Drittens um den Betrieb: Mandantenfähigkeit, Datenschutzkonzept, Moodle-Import und ein Testprojekt neben der Produktion. Der Text beschreibt den Stand vom September 2026.

2 Technische Architektur

heyprof basiert auf einem Web-Stack, der für schnelle Iteration und minimalen Betriebsaufwand ausgelegt ist.

Frontend

Next.js 16 mit React 19 und TypeScript (App Router), Tailwind CSS und shadcn/ui. Mathematik wird clientseitig mit KaTeX gesetzt, PDFs mit pdf.js dargestellt, Code in CodeMirror (Python, SQL) bearbeitet.

Backend

Serverlose API-Routen in Next.js für alle KI- und Geschäftslogik. Supabase (PostgreSQL mit Row-Level Security) für Daten, Authentifizierung und Dateiablage; die Vektorsuche läuft in der Erweiterung pgvector.

KI-Dienste

OpenAI-Modelle der Reihen GPT-4.1 und GPT-5 für Tutor-Dialog, Bewertung, Übersetzung und Zusammenfassung, `text-embedding-3-small` für die semantische Suche, Whisper für die Transkription, ein Sprachmodell für die Sprachausgabe und ein Vision-Modell für Handschrift und Fotos. Einzelne KI-Werkstätten (CogniApps) nutzen Claude-Modelle von Anthropic. Alle Aufrufe laufen ausschließlich über die Plattform, nie direkt aus dem Browser zum Anbieter.

VisuApps

Statische, in sich geschlossene HTML-Anwendungen mit einer eigenen Klassenbibliothek (VisuKit, 2D und 3D auf Three.js), die per iframe in Kurs, Demo und Website eingebettet werden und über ein Nachrichtenprotokoll Theme, Farbpalette, Sprache und Tutor-Befehle empfangen.

Die Anwendung wird auf Vercel betrieben und benötigt keine eigene Serverinfrastruktur. Rollen (Lehrende, Studierende, Administration) werden auf Datenbankebene durch Row-Level Security und serverseitige Prüfungen durchgesetzt. Seit August 2026 ist die Plattform mandantenfähig: Organisationen erhalten getrennte Datenräume, und ein eigenes Testprojekt trennt Entwicklung von Produktion. Eine CI-Pipeline prüft bei jeder Änderung die Typen und die Trennung der Mandanten. Schemaänderungen erfolgen ausschließlich als versionierte Migrationen.

Das **Datenmodell** besteht aus den Kernentitäten *Course*, *Session*, *Material* (mit semantisch indextierten *Chunks*), *Exercise* (Freitext, Multiple-Choice, Code) mit *Collections* und *Labels*, *Exam*,

Attempt beziehungsweise *Submission*, *Recording* mit Transkript, *LiveQuiz*, *CommunityPost* und den Lernsignalen der *Lernlandkarte*.

Über diesem Kern liegt ein **Baukasten aus 42 Kursmodulen**. Die Lehrkraft schaltet je Kurs frei, was sie braucht: Übungen, Tests und Probeklausuren, VisuApps und CogniApps, Community und anonyme Fragen, Live-Quiz, Blitzrunden, Aktionswochen und Punktetafel, Vokabeltrainer, Datenlabor, Mikrolektionen, Lernlandkarte, Wissensräume, heyCanvas, Klassenverwaltung und fachspezifische Werkstätten. Ein Kurs zeigt nur die eingeschalteten Module; die Oberfläche bleibt dadurch schmal, obwohl der Baukasten breit ist.

3 Funktionen für Lehrende

3.1 Kurs anlegen

Ein Kurs entsteht auf drei Wegen. Klassisch im Dashboard, mit einem Zugangscode, über den sich Studierende einschreiben. Aus vorhandenen Dokumenten: Aus Skripten, Foliensätzen oder Aufgabenblättern erzeugt die Plattform Sessions, Materialien und Übungsvorschläge, die die Lehrkraft prüft und übernimmt. Oder als Import eines bestehenden Moodle-Kurses im .mbz-Format, bei dem Abschnitte, Dateien und Aufgaben übernommen werden. Jeder Kurs ist in Sessions gegliedert, die Vorlesungsterminen entsprechen; die Reihenfolge aller Elemente lässt sich per Drag-and-Drop ändern.

3.2 Materialien und Annotation

Materialien sind PDF-Folien, Skripte, Videos oder HTML-Foliensätze, die in der Plattform als eigenständige Decks mit Formeln und eingebetteten VisuApps dargestellt werden. PDF-Seiten lassen sich mit einem Freihandwerkzeug annotieren; Studierende sehen die Anmerkungen als Overlay. Nach dem Hochladen werden Materialien *ingested*: Der Text wird seitenweise extrahiert, in Abschnitte von höchstens 1500 Zeichen geteilt, eingebettet und im Vektorspeicher abgelegt, damit der Tutor gezielt aus dem Kursmaterial zitieren kann. Handschriftliche Annotationen der Lehrkraft werden per Handschrifterkennung in Text und $\text{\LaTeX}$ überführt und dem Tutor als Kontext bereitgestellt.

3.3 Folien-Werkstatt

Wer keinen fertigen Foliensatz hat, lässt ihn bauen. In der *Folien-Werkstatt* (SlideLab) lädt die Lehrkraft Quellen hoch – Skript, Notizen, ein altes Deck –, die KI analysiert sie und schlägt eine Gliederung vor. Daraus entsteht ein HTML-Foliensatz, der sich Folie für Folie überarbeiten lässt; passende VisuApps werden als Steckbriefe angeboten und direkt eingebettet. Das Ergebnis ist ein Kursmaterial wie jedes andere, durchsuchbar, annotierbar und für den Tutor indexiert.

3.4 Präsentation im Hörsaal

Die Lehrkraft präsentiert unmittelbar aus der Plattform, ohne Umweg über ein anderes Programm. Auf PDF- und HTML-Folien liegt ein Handschrift-Overlay; ein Laserpointer funktioniert

auch über eingebettete VisuApps hinweg. Eine Abdeckung gibt die Folie schrittweise frei. Daneben steht eine digitale Tafel mit einstellbarer Tafelfarbe und einer zweiten Fläche für Nebenrechnungen. Ein Geräteschalter entscheidet, ob der Finger schreibt oder blättert – am Touch-Panel im Hörsaal eine wichtigere Einstellung, als sie klingt. Was auf der Folie entsteht, bleibt erhalten und steht den Studierenden hinterher als Overlay zur Verfügung.

3.5 Klassen und Konten ohne E-Mail

Nicht jede Lerngruppe hat E-Mail-Adressen. Für Schulklassen legt die Lehrkraft die Konten deshalb selbst an, aus einer Namensliste, und druckt Zugangskarten. Die Lernenden melden sich über einen Klassenlink an, mit Benutzername und Startpasswort. In Grundschulkursen tritt an dessen Stelle ein *Bildpasswort*: Namen antippen, drei Bilder in der richtigen Reihenfolge tippen, fertig – ohne Lesen müssen. Als Avatar dient ein Monster. Passwortrücksetzung, Deaktivierung und Löschung liegen bei der Lehrkraft; einen Selbstbedienungs-Reset gibt es bewusst nicht. Damit lassen sich Lerngruppen führen, für die eine gewöhnliche Registrierung rechtlich oder praktisch ausscheidet.

3.6 Übungen

Der Aufgabeneditor kennt drei Typen: *Freitext*, *Multiple-Choice* und *Code* (Python, SQL). Zu jeder Aufgabe gehören eine Musterlösung (auf Wunsch für Studierende sichtbar), private Tutorhinweise, die nur der KI-Tutor sieht, sowie Labels und Sammlungen zur Organisation, etwa nach Thema und Schwierigkeit. Aufgabensammlungen lassen sich aus Markdown-Dateien importieren, von der KI aus einer Themenbeschreibung erzeugen oder auf Knopfdruck zwischen Deutsch und Englisch übersetzen.

3.7 Tests

Ein *Test* ist eine benotbare Aufgabenfolge, die vollständig in der Plattform abläuft. Fragetypen sind Freitext, Lückentext, Multiple-Choice und numerische Antwort. Zu jeder Frage lassen sich *Varianten* hinterlegen, aus denen die Plattform je Versuch zufällig eine auswählt; zusätzlich können die Fragen gemischt werden. Die Lehrkraft legt Zeitlimit, Versuchszahl, Freigabezeitraum, eine Beschränkung auf bestimmte Netze, einen Abzug für Wiederholungsversuche und den Umfang der Rückmeldung fest – keine, nur Punkte oder vollständiges Feedback. Der Tutor lässt sich für einen Test abschalten oder auf eine feste Zahl von Hinweisen begrenzen. Auf Wunsch erzeugt die KI den Test aus den Materialien des Kurses. Musterlösungen und Versuche liegen hinter serverseitigen Routen, damit der Browser sie nicht auslesen kann.

3.8 Klausuren

Klausuren bestehen aus nummerierten Fragen mit Punktzahl und Zeitlimit. Fragen werden aus gescannten Klausurblättern importiert; eine KI-Pipeline extrahiert Text, Nummerierung und Punkte. Studierende reichen Text, Zeichnungen oder Fotos handschriftlicher Lösungen ein, die das Vision-Modell liest und bewertet. Nach der Abgabe stehen alle Einreichungen mit KI-Vorbewertung und Feedback bereit; die Lehrkraft behält die Entscheidung.

Klausuren dienen derzeit ausschließlich der *Vorbereitung*; das Modul heißt in der Plattform folgerichtig *Probeklausur*. Studierende üben unter Klausurbedingungen und sehen anschließend, wo sie stehen. Eine prüfungsrechtlich verbindliche Abnahme ist damit nicht verbunden und derzeit auch nicht beabsichtigt.

3.9 Aufnahmen, Transkription und Live-Quiz

Ein browserbasierter Rekorder zeichnet die Vorlesung auf, alternativ werden Audiodateien hochgeladen. Whisper transkribiert im Hintergrund, ein Sprachmodell erzeugt eine gegliederte Zusammenfassung in mehreren Schritten. Transkript und Zusammenfassung stehen den Studierenden zur Verfügung, und der Tutor kann aus dem gesprochenen Wort der Vorlesung antworten, mit genau den Beispielen und Metaphern, die im Hörsaal fielen. Für die Präsenzlehre gibt es ein *Live-Quiz*: Die Lehrkraft startet eine Frage, die Studierenden antworten am Handy, das Ergebnis erscheint sofort. Daneben stehen *Blitzrunden* mit QR-Code für kurze Abfragen im Saal, eine gemeinsame *Punktetafel* für Gruppenwettbewerbe, die Rundensummen und Rang selbst rechnet, und *Aktionswochen* für zeitlich begrenzte Kursaktionen samt Auswertung und CSV-Export.

3.10 Lernanalysen

Ein kursweites Dashboard zeigt der Lehrkraft, wie die Kohorte arbeitet: welche Materialien Reichweite haben, welche VisuApps benutzt werden, wie sich die Vokabel-Lernkurve entwickelt, wie viel Zeit investiert wird und welche Themen den Tutor beschäftigen. Eine Ampel markiert Auffälligkeiten, ein Kohortenvergleich ordnet sie ein, Semesterbericht und CSV-Export gehen aus dem Haus. Die Auswertung ist datenschutzseitig eng geführt: Ereignisse werden tagesweise verdichtet, nach 60 Tagen gelöscht, und ob Klarnamen sichtbar sind, entscheidet ein eigener Schalter. Die zugrunde liegenden Tabellen sind für Clients gesperrt und ausschließlich über geprüfte Serverrouten erreichbar.

3.11 Tutor-Konfiguration

Auf Kursebene hinterlegt die Lehrkraft Tutorhinweise für Material- und Aufgabenfragen oder schaltet den Tutor ab. Seine *Persönlichkeit* (im System *Soul*) wird aus Profilen gewählt, die Tonfall, Stil und Schwerpunkte beschreiben. Neben sachlichen und ermutigenden Varianten gibt es volksnahe (*Soul Proll*: „x geteilt durch Null und dann ...? Merkste selber, Digger, oder?“). Ein Tutor-Blick auf Kursebene zeigt, welche Themen die Kohorte gerade beschäftigen.

4 Funktionen für Studierende

4.1 Lernstudio

Das Dashboard zeigt die eingeschriebenen Kurse mit Fortschritt und Hinweisen auf neue Materialien. Im *Lernstudio* liegen je Session Folien mit Annotationen, Aufnahme mit Transkript und Zusammenfassung und die Übungen des Termins auf einer Seite, der KI-Tutor daneben. Im

PDF-Viewer markieren Studierende einen Bereich einer Folie oder eines Tafelbilds; der Tutor erklärt genau diesen Ausschnitt. Eine kursabhängige Begrüßungstour führt beim ersten Besuch durch die Oberfläche; der Tutor dient danach als Wegweiser. Die Oberfläche ist deutsch und englisch, für Rechner und Handy, hell und dunkel.

4.2 Übungen

Für jede Aufgabe stehen mehrere Wege offen:

1. **Text oder Code** mit sofortiger KI-Rückmeldung.
2. **Zeichenfläche** für handschriftliche Lösungen, auch mit Stift auf dem Tablet.
3. **QR-Foto**: Der Rechner zeigt einen QR-Code, das Handy nimmt die handschriftliche Lösung auf, das Foto erscheint am Rechner und wird gelesen und bewertet.
4. **KI-Tutor** im integrierten Dialog, der Aufgabe, Musterlösung, Tutorhinweise und den aktuellen Lösungsstand kennt.
5. **Variante üben**: Die KI erzeugt eine strukturgleiche Aufgabe mit anderen Zahlen oder Namen.
6. **Lösung anzeigen**, nach einem echten Versuch.

Python läuft vollständig im Browser (CPython als WebAssembly, Pyodide) mit Syntaxhervorhebung, Ausführen-Knopf und Ausgabefeld; ein Zeitlimit beendet Endlosschleifen, gefährliche Importe sind gesperrt. Für SQL gibt es einen Playground mit einer Datenbank im Browser. Übungen richten sich nach Stand und Motivation: Bewertete Versuche, Labels und die Lernsignale der VisuApps fließen in eine *Lernlandkarte* je Kompetenz, aus der Vorschläge für die nächsten Aufgaben entstehen.

4.3 Community und Sprache

Jeder Kurs hat eine Community mit anonymen Fragen, Antworten und Abstimmungen; Lehrende sehen, was die Kohorte bewegt, ohne dass sich jemand exponieren muss. Der Tutor spricht auf Wunsch: Antworten und Erklärungen in VisuApps werden über eine Serverstimme vorgelesen, was für Grundschule und Sprachlernende den Unterschied macht.

4.4 Weitere Lernräume

Neben Material und Aufgabe stehen Kursmodule, die anderen Zugängen folgen.

Wissensräume

ein begehbare 3D-Gebäude je Person: Räume, Stockwerke und Treppen werden selbst angebaut, an die Innenwände kommen VisuApps, Materialien, Videos und Notizen. Die Apps laufen dort *live* und behalten ihren Zustand. Die Lehrkraft kann ihr Gebäude als Kurs-Ausstellung veröffentlichen.

heyCanvas

eine unendliche, zoombare Fläche für Vorlesung und Videoproduktion, auf der Handschrift, Text- und Formelinseln und bedienbare VisuApps nebeneinander liegen; eine Übersicht zeigt die Fläche als Baum, und der Blick fliegt gezielt zu einer Insel.

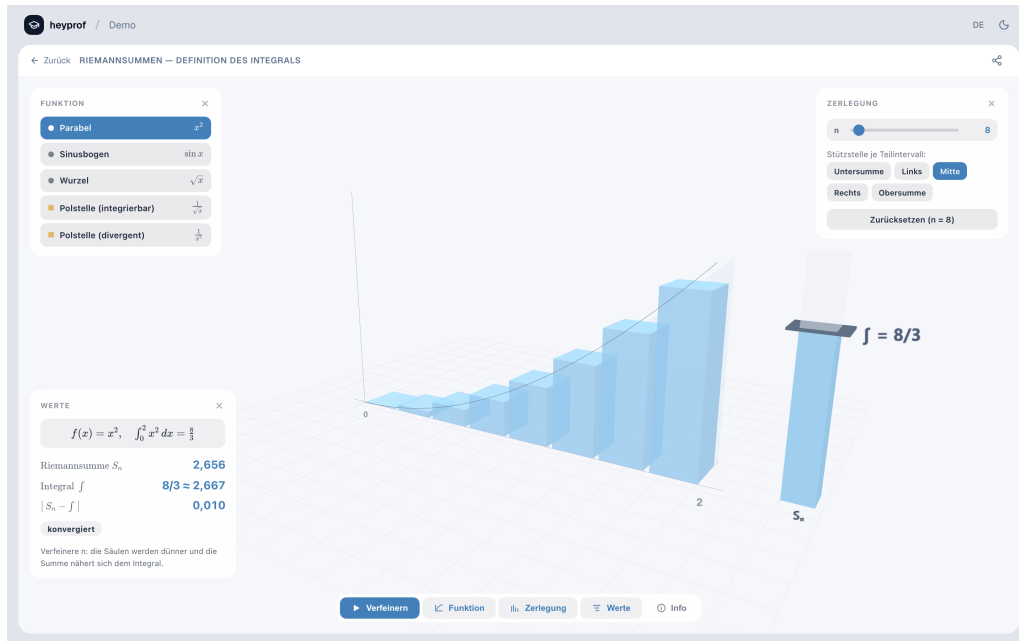


Abbildung 1: Die VisuApp *Riemannsummen* in 3D: Zerlegung, Stützstelle und Funktion sind einstellbar, die Werte-Karte zeigt Summe, Integral und Abstand. Der Tutor kann die App über *Tutor-Show* selbst bedienen und dabei erklären.

Mikrolektionen

kurze, KI-geführte Dialoge mit Zwischenfragen und adaptiver Hilfe, die sich um ausgewählte VisuApps des Kurses bauen lassen – Entwürfe erzeugt die KI aus dem Material.

Datenlabor

visuelle Daten-Workflows nach Art von KNIME: einlesen, aufbereiten, Modelle bauen, Dashboard. Die Knoten rechnen wirklich, und der Tutor hilft bei der Fehlersuche.

Vokabeltrainer

Sätze und Vokabeln per Texterkennung aus einer Vorlage übernehmen und in mehreren Übungsformen trainieren.

Für die Grundschule kommt *Kompass 4* hinzu, ein Trainer für die Vergleichsarbeiten der vierten Klasse: KI-erzeugte Übungstests mit Lernstandsprofil, ein Begleiter durch den Übungslauf, ein adaptiver Schwierigkeitsausgleich, Aufgaben in neun Familiensprachen, ein Papiermodus (drucken, scannen, bewerten, nachbesprechen) und ein Bericht für die Eltern.

5 VisuApps: Wissen zum Anfassen

Viele Zusammenhänge versteht man nicht aus einer Formel, sondern indem man an ihnen dreht. Die VisuApps sind interaktive Visualisierungen, die genau das erlauben: Riemannsummen verfeinern, bis die Säulen zum Integral werden; Anscombes Quartett mit identischen Kennzahlen

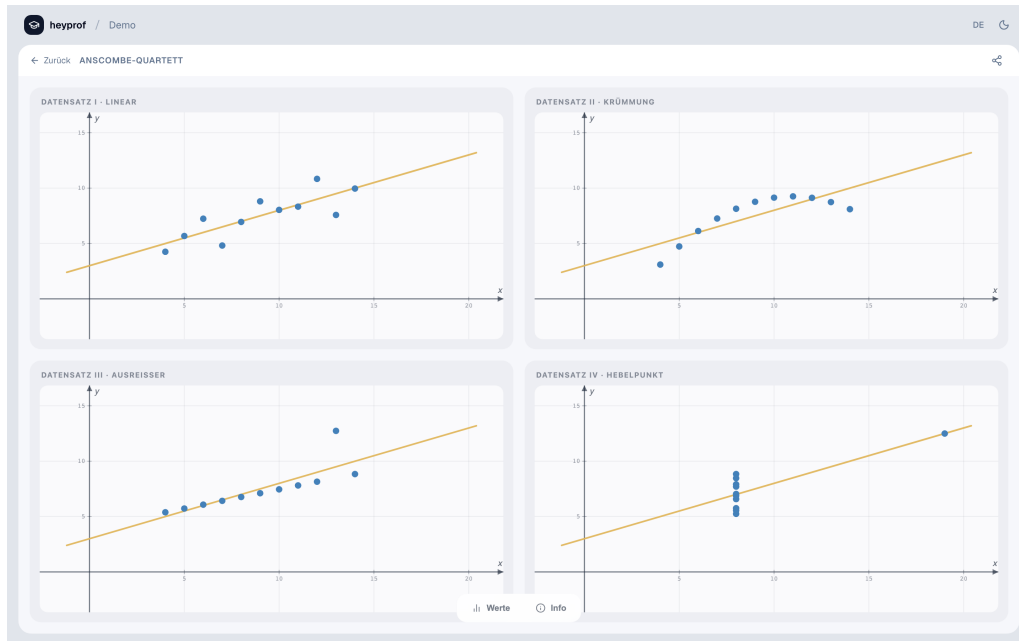


Abbildung 2: Anscombes Quartett: vier Datensätze, gleiche Regressionsgerade, gleiche Kennzahlen, vier Formen. Ein Argument für das Plotten, das man nicht mehr vergisst.

und völlig verschiedener Gestalt; ein Neuron, ein Pendel, eine Kreditverlustverteilung, ein ICE-Netz. Im September 2026 sind 590 Apps in 66 Fachsammlungen registriert, von Mathematik, Physik, Chemie, Biologie und Informatik über Wirtschaft, Statistik und Finanzderivate bis zu Linguistik, Musik, Geographie und Wissenschaftlichem Arbeiten.

Gemeinsame Grundlage. Alle Apps bauen auf *VisuKit* auf, einer eigenen Klassenbibliothek für 2D und 3D. Sie liefert die Laufzeit (Theme, Farbpalette, Sprache, Drosselung während eines Tutor-Streams), Szenen, Knoten, austauschbare Formensprachen und das einheitliche Bedienschema: Panels an den Ecken, eine Pillenleiste unten, ein Info-Panel mit der didaktischen Kurzerklärung, Bottom-Sheets auf dem Handy. Farben kommen aus der Kursfarbe, damit sich eine App in jeden Kurs fügt, hell und dunkel.

Didaktik zuerst. Jede App beginnt mit einem Didaktik-Brief: Was ist die kleinste korrekte Wissenseinheit? Welche Fehlvorstellung soll die Szene sichtbar widerlegen? Welche Objekte müssen unterscheidbar sein, welche Zustände gibt es, welcher Aha-Moment soll entstehen? Was kein Konzept trägt, kommt nicht in die Szene; keine Animation ohne Lernzweck.

Tutor-Show. Über ein maschinenlesbares Manifest deklariert eine App, welche Anker, Verben und Bedienelemente sie dem Tutor anbietet. Der Tutor antwortet dann nicht mit Chat-Text, sondern führt vor: Er setzt Sprechblasen an die Stelle, um die es geht, zeigt mit dem Pointer, stellt Regler, schwenkt die Kamera und hebt Panels hervor; die Lernenden können jederzeit eingreifen. 568 der Apps sind so steuerbar. Eine geführte Erklärung kommt damit immer vom Tutor und richtet sich nach der Frage, nicht nach einem Drehbuch.

CogniApps und VisuLab. 23 *CogniApps* lassen die KI selbst Übungsmaterial erzeugen: unbegrenzte Zitierübungen, ein Sparring zur Forschungsfrage, ein Gegenleser für Gliederungen,

eine Frage-Klinik für Fragebogen-Items. Im *VisuLab* bauen Lehrende und Studierende eigene Visualisierungen mit KI-Unterstützung; 414 solcher Apps sind entstanden. Öffentliche Demo-Apps sind ohne Anmeldung auf der Website erreichbar und in den Kurs-Decks eingebettet.

Einbettung auf fremden Seiten. Eine VisuApp lässt sich als iframe in eine beliebige Website stellen, etwa in den Auftritt einer Hochschule oder eines Verlags. Der Zugang läuft über signierte Schlüssel, die in der Administration erzeugt werden und den zugehörigen Einbettungsschnipsel gleich mitliefern; ein Parameter setzt die Akzentfarbe, damit sich die App in die fremde Gestaltung fügt. Die Nutzung wird je Schlüssel gezählt, und die Auslieferung ist auf die freigegebenen Seiten beschränkt.

Grundschule. Für die Klassen 1 bis 4 gilt ein eigenes Regelwerk: ein sprechender Begleiter statt Text, Bedienung ohne Lesenmüssen mit großen Tippflächen, Familiensprache, Lektionen mit Fachsprache, ruhige Rückmeldung ohne Konfetti. 23 Apps eines 100-Apps-Programms sind fertig, von der Zahlzerlegung bis zum Einmaleins.

6 KI-Funktionen und pädagogische Integration

6.1 Kontext statt Weltwissen

Der zentrale Unterschied zu allgemeinen Assistenten ist der Kontext. Bevor der Tutor antwortet, wird sein System-Prompt angereichert mit

- dem vollständigen Text der aktuellen Aufgabe und der Musterlösung,
- den privaten Tutorhinweisen der Lehrkraft für diese Aufgabe und den Kurs,
- dem aktuellen Code oder Lösungsstand der Studierenden,
- semantisch abgerufenen Passagen aus den Kursmaterialien und dem Vorlesungsstranskript,
- den erkannten handschriftlichen Annotationen der aktuellen Folie und dem markierten Ausschnitt,
- bei einer VisuApp dem Manifest der App und ihrem aktuellen Zustand,
- der gewählten Tutor-Persönlichkeit.

So kann der Tutor präzise und niveaugerecht helfen, *ohne* die Antwort vorwegzunehmen. Eine allgemeine KI kennt weder das Lehrbuch noch die im Kurs eingeführten Konstrukte. heyprof schon.

6.2 Retrieval-Augmented Generation

Retrieval-Augmented Generation versorgt ein Sprachmodell vor der Antwort mit relevantem Kontext aus einer externen Wissensbasis [1]. Das Modell bleibt unverändert; der Prompt wird mit passenden Passagen angereichert. In der *Ingestion-Phase* wird jedes Dokument seitenweise extrahiert, in Abschnitte von höchstens 1500 Zeichen geteilt und mit `text-embedding-3-small` in Vektoren überführt, die zusammen mit dem Text in `pgvector` liegen. In der *Abfrage-Phase* wird die Frage ebenso eingebettet, die `k` ähnlichsten Abschnitte (Kosinus-Ähnlichkeit) werden gesucht und in den Prompt eingefügt. Fragt jemand „Was versteht man unter einer verketteten Liste?“, antwortet der Tutor aus dem Kursskript, mit der Notation der Vorlesung, nicht aus dem Weltwissen des Modells.

6.3 Sokratischer Modus und Erklärmodus

Im **sokratischen Modus** stellt der Tutor Gegenfragen, die einen Schritt weiterführen, ohne die Lösung zu verraten:

Studierende: Ich weiß nicht, wie ich mit der Rekursion anfangen soll.

Tutor: Überlege: Was ist der einfachste Eingabewert, für den du die Antwort sofort weißt, ohne zu rechnen?

Im **Erklärmodus** erklärt der Tutor direkt und anschaulich, etwa wenn jemand eine Folie nicht versteht, mit konkreten Beispielen, Alltagsmetaphern und ausdrücklichem Bezug auf den Folieninhalt. In einer VisuApp wird die Erklärung zur Vorführung.

6.4 Automatische Bewertung

Textantworten bewertet ein strukturierter Modellaufruf mit einem Urteil und kurzem Feedback. Handschriftliche oder fotografierte Lösungen liest das Vision-Modell, prüft sie auf Relevanz und überführt sie in $\text{\LaTeX}$ -Notation. Die Bewertung aktualisiert den Fortschritt sofort; die Lehrkraft kann jede Bewertung einsehen und übersteuern.

6.5 Weitere KI-Dienste

Aufgabenvarianten

strukturgleiche Aufgaben mit anderen Zahlen, Namen oder Funktionen für Wiederholungsübungen.

Aufgabengenerierung

ganze Sammlungen aus einer Themenbeschreibung, mit Freigabe durch die Lehrkraft.

Übersetzung

von Aufgaben zwischen Deutsch und Englisch; Formeln und Code bleiben erhalten.

Klausurhinweise

begrenzte sokratische Hinweise je Frage, auch während der Klausur.

Handschrifterkennung

für Annotationen, Zeichenflächen und Fotos.

Sprachausgabe

über eine Serverstimme, damit Zahlen und Formeln auf jedem Gerät gleich klingen.

7 Pädagogische Prinzipien

Konstruktive Abstimmung [2]. Jede Aufgabe, jeder Tutorhinweis und jede Sammlung ist mit einem Lernziel verknüpft. Die KI nutzt die privaten Hinweise der Lehrkraft, um zur erwarteten Lösung zu führen, ohne sie preiszugeben.

Zone der nächsten Entwicklung [3]. Der sokratische Tutor sucht die Grenze des aktuellen Verständnisses und stellt eine Frage, die genau einen Schritt darüber hinausgeht.

Deliberate Practice [4]. Die Variantenfunktion erlaubt unmittelbares Üben nach dem Verstehen: gleiches Konzept, neue Zahlen, sofortige Rückmeldung.

Multimodaler Input. Tippen, Zeichnen, Fotografieren und Sprechen sind gleichberechtigt; die Diskrepanz zwischen Denken und Einreichungsformat wird klein.

Anschaung mit Absicht. VisuApps zeigen nur, was ein Konzept trägt. Jede Bewegung dient einem Zustandswechsel oder dem Aha-Moment; wissenschaftliche Stimmigkeit geht vor Foto-realismus. Der Tutor führt vor, statt zu dekorieren.

Lernende als Gestalter. Im VisuLab bauen Studierende eigene Visualisierungen, in der Community beantworten sie Fragen der anderen, im Peer-Feedback bewerten sie nach gemeinsamen Prüflisten.

8 Datenschutz und Betrieb

Lernplattformen verarbeiten Daten von Menschen, die sich nicht aussuchen können, wo sie lernen. heyprof behandelt das als Gestaltungsaufgabe, nicht als Fußnote. Mit allen Dienstleistern bestehen Auftragsverarbeitungsverträge; die Anbieterliste mit Zweck, Sitz und Rechtsgrundlage ist auf der Plattform veröffentlicht. KI-Anfragen laufen ausschließlich über die Plattform, mit dem nötigen Kontext und ohne Namen; VisuApps laden nichts von fremden Servern, sodass keine IP-Adresse an Dritte gelangt. Nutzende erhalten die Auskunft nach Art. 15 DSGVO direkt im Profil; eine Kontolöschung entfernt alle zugehörigen Daten, nachgewiesen durch eine Inventur über alle Tabellen. Ein Verarbeitungsverzeichnis und ein Vorgehensplan dokumentieren den Stand. Entwicklung und Produktion sind getrennte Projekte; Organisationen sind als Mandanten voneinander isoliert.

Die Mandantenfähigkeit hat auch eine Oberfläche: Eine Organisation verwaltet ihre Mitglieder, bearbeitet Beitrittsanfragen und spricht Einladungen aus; verwaltete Klassenkonten treten der Organisation des Kurses automatisch bei. Das Community-Board ist je Organisation getrennt, und wie weit ein Beitrag reicht, entscheidet der Kurs.

Der Betrieb finanziert sich über Stufen. Ein Konto liegt auf einer freien oder einer bezahlten Stufe, an die eine verbindliche Funktionsmatrix geknüpft ist; die Zahlung läuft über einen externen Dienstleister, die Freischaltung über dessen Rückmeldung an die Plattform. Zahlungsdaten werden nicht in heyprof gespeichert.

9 Einsatz und Zahlen

Im Sommersemester 2026 haben wir heyprof als experimentellen Prototypen in der eigenen Lehre eingesetzt. Die Zahlen in Tabelle 1 stammen aus diesem Betrieb und sind entsprechend zu lesen: Sie beschreiben die Nutzung eines Prototyps über ein Semester, nicht den Regelbetrieb einer eingeführten Plattform.

Die Plattform wird von der Studio Quant GmbH entwickelt und betrieben.

Tabelle 1: Nutzung der Produktionsinstanz, Stand 21. September 2026.

Konten (Lehrende / Studierende)	312 (19 / 293)
Kurse (davon veröffentlicht / mit Einschreibungen)	78 (44 / 31)
Einschreibungen	529
Studierende mit Aktivität in den letzten 90 Tagen	224
Materialien / indexierte Abschnitte	413 / 11 989
Übungsaufgaben	2 854
Übungsversuche (Studierende)	11 672 (181)
Vorlesungsaufnahmen / Live-Quiz	125 / 39
Community-Beiträge	191
Im VisuLab gebaute Visualisierungen	414
KI-Aufrufe seit 31. März 2026 (Nutzende)	40 416 (215)
davon Tutor-Dialoge (Studierende)	17 635 (167)
davon Textbewertungen / Fotobewertungen	8 237 / 3 938
davon Handschrifterkennungen	7 454
Verarbeitete Sprachmodell-Tokens	rund 83 Mio.

10 Ausblick

heyprof wird laufend weiterentwickelt; eine unvollständige Auswahl.

Adaptive Lernpfade. Die Lernlandkarte sammelt bereits Signale aus Übungen und VisuApps. Der nächste Schritt ist ein Empfehlungssystem, das aus der individuellen Historie die jeweils passend herausfordernde Aufgabe vorschlägt, mit Voraussetzungsbeziehungen in einem Kompetenzgraphen.

Native iPad-App. Eine eigenständige iPad-Anwendung ist in Arbeit. Sie führt die Weboberfläche in einer nativen Schale und ergänzt sie um das, was im Browser nicht geht: Handschrift über PencilKit mit dokumentweitem Rückgängig, Annotation von Folien direkt auf dem Gerät und einen nativen Präsentationsmodus für den Einsatz im Hörsaal.

Anbindung an Hochschulsysteme. Single-Sign-On und automatische Einschreibung über die Systeme des Prüfungsamts.

Weitere Welten. Das Grundschulprogramm wächst auf 100 Apps; die Kollektionen für Schule, Weiterbildung und lebenslanges Lernen werden ausgebaut.

11 Fazit

heyprof zeigt, dass KI in der Lehre nicht bedeuten muss, dass Lernende den Lernprozess umgehen. Ein Tutor, der tief in Material, Aufgabe und Visualisierung eingebettet ist, pädagogischen Kontext bekommt und auf sokratische Führung verpflichtet bleibt, macht aus einem potenziell schädlichen Werkzeug ein wirksames Lernmittel. Lernende erhalten rund um die Uhr sofortige, niveaugerechte Unterstützung, ohne den kognitiven Aufwand zu umgehen, den echtes Lernen erfordert. Lehrende gewinnen eine Autorenumgebung, automatische Rückmeldung, eine Bibliothek anschaulicher Visualisierungen und einen Blick auf den Stand ihrer Kohorte.

Nicht zuletzt ist heyprof selbst ein Produkt KI-gestützter Entwicklung: Ein großer Teil des Codes und der Dokumentation entstand mit Hilfe von Claude [5], ein praktischer Beleg dafür, dass KI-Assistenten, richtig eingesetzt, die Produktivität eines kleinen Teams erheblich steigern.

Literatur

- [1] Patrick Lewis u. a. „Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks“. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Bd. 33. 2020, S. 9459–9474.
- [2] John Biggs und Catherine Tang. *Teaching for Quality Learning at University*. 4. Aufl. Maidenhead: McGraw-Hill / Open University Press, 2011.
- [3] Lev S. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Hrsg. von Michael Cole u. a. Cambridge, MA: Harvard University Press, 1978.
- [4] K. Anders Ericsson, Ralf T. Krampe und Clemens Tesch-Römer. „The Role of Deliberate Practice in the Acquisition of Expert Performance“. In: *Psychological Review* 100.3 (1993), S. 363–406. DOI: [10.1037/0033-295X.100.3.363](https://doi.org/10.1037/0033-295X.100.3.363).
- [5] Anthropic. *The Claude 3 Model Family: Opus, Sonnet, Haiku*. Techn. Ber. Model Card. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf. Anthropic, 2024.