

KI-Workshop Handout – Gymnasium Wildeshausen

Bechtle AI Chat Demo – Workshop-Handout

KI-Praxis-Workshop Gymnasium Wildeshausen

Stand: 09. April 2026 Zielgruppe: IT-Administratoren und interessierte Lehrkraefte Plattform: Open WebUI mit Bechtle-Branding

1. Plattform-Uebersicht

1.1 Allgemeine Informationen

Eigenschaft	Wert
URL	https://chat.demo.bechtle-bremen.de
Plattform	Open WebUI mit Bechtle-Branding (“Bechtle AI Chat Demo”)
KI-Modell	Mistral Small 3.1 (24B Parameter, Q4_K_M Quantisierung)
VRAM-Bedarf	~15 GB
Hosting	Lokal auf GPU-Server (On-Premises bei Bechtle Bremen)
Kontext-Fenster	128.000 Token (konfiguriert: 16.384 Token via num_ctx)
Geschwindigkeit	~150 Tokens/Sekunde
Branding	Bechtle-Gruen (#075033 dunkel / #23A96A hell)

1.2 Hardware

Komponente	Aktuell (Demo)	Ziel (Produktion)
GPU	NVIDIA A100-SXM4-80GB (GPU 0 dediziert)	2x NVIDIA A30
GPU-Zuweisung	CUDA_VISIBLE_DEVICES=0	—
VRAM genutzt	~15 GB von 80 GB	~15 GB von 2x 24 GB

1.3 Datenschutz

- **Alle Daten bleiben lokal** — keine Cloud, keine Weitergabe an Dritte
 - Kein Kontakt zu OpenAI, Google, Microsoft oder anderen Cloud-KI-Diensten
 - Open-Source-Modell (Mistral, Apache 2.0 Lizenz) — kein Vendor Lock-in
 - **DSGVO-konform** einsetzbar
 - Gespraechе werden nur lokal auf dem Server gespeichert
 - Keine Telemetrie, kein Tracking
-

2. Arbeitsbereiche im Detail

Die Plattform bietet 10 vorkonfigurierte Arbeitsbereiche (Custom Models). Jeder hat einen eigenen System-Prompt, eine eigene Knowledge Base und ist auf einen bestimmten Anwendungsfall optimiert.

Quellennutzungsmodi – Erklaerung

Modus	Bedeutung	Wann eingesetzt
STRICT	KI antwortet NUR aus den bereitgestellten Dokumenten. Kann keine Antwort erfinden.	Wenn Genauigkeit kritisch ist (z.B. Schulordnung — keine Regeln erfinden!)
HYBRID	KI nutzt Dokumente bevorzugt, ergaenzt mit eigenem Wissen.	Wenn Dokumente als Qualitaetssicherung dienen, aber umfassende Antworten gewuenscht sind
FAUST	Spezialform: Primaer aus dem Volltext zitieren, Interpretationen mit Fachwissen ergaenzen.	Fuer literarische Texte — Unterscheidung zwischen Textbeleg und Interpretation

2.1 Mistral Small Arbeitsbereich (Allgemein)

Eigenschaft	Wert
ID	mistral-small-arbeitsbereich
Zielgruppe	Alle Nutzer
Quellennutzungsmodus	HYBRID
Knowledge Base	Open WebUI Dokumentation (1 Datei: openwebui-configuring_en.pdf)
Besonderheiten	Bildverstaendnis (Vision) aktiviert, Code Interpreter aktiviert

Warum HYBRID? Der allgemeine Arbeitsbereich soll flexibel einsetzbar sein. Die Open-WebUI-Doku dient als Referenz, aber das Modell soll auch allgemeine Fragen beantworten koennen.

System-Prompt:

Du bist ein KI-Assistent basierend auf Mistral Small 3.1, einem hochmodernen multimodalen Open-Source-Modell mit 128k Token Kontextfenster.

KERNKOMPETENZEN:

- Native Text- und Bildverarbeitung (OCR, visuelle Inspektion, Dokumentenanalyse)
- Exzellentes Function Calling und strukturierte JSON-Ausgaben
- Code-Entwicklung und technische Dokumentation
- RAG-Anwendungen und Wissensdatenbank-Integration
- Mehrsprachigkeit (ueber 40 Sprachen, besonders stark in nicht-englischen Kontexten)

ARBEITSWEISE:

- Nutze das grosse Kontextfenster fuer umfangreiche Dokumente und Code-Repositories
- Generiere strukturierte Outputs (JSON) fuer API-Integrationen
- Unterstuetze autonome Agenten-Workflows mit praezisem Function Calling
- Bevorzuge Deutsch als primaere Kommunikationssprache
- Antworte praezise, technisch fundiert und effizient

Du bist optimiert fuer Enterprise-IT-Infrastrukturen, On-Premises-Deployments und moderne Cloud-Native-Architekturen.

QUELLENNUTZUNG:

Wenn Dokumente bereitgestellt sind, nutze sie bevorzugt. Ergaenze mit deinem allgemeinen Wissen wo noetig.

Prompt-Vorschlaege: (Keine spezifischen — allgemeiner Bereich)

2.2 Faust-Experte

Eigenschaft	Wert
ID	faust-experte
Zielgruppe	Schueler/innen im Deutschunterricht
Quellennutzungsmodus	FAUST
Knowledge Base	Faust-Experte (1 Datei: faust_i_goethe.txt — Volltext, 218 KB)

Warum FAUST-Modus? Bei literarischen Texten ist es wichtig, zwischen direkten Textziten und Interpretation zu unterscheiden. Der Faust-Modus stellt sicher, dass die KI primär aus dem Originaltext zitiert und eigene Interpretationen als solche kennzeichnet.

System-Prompt:

Du bist ein hilfreicher Literatur-Assistent fuer den Deutschunterricht am Gymnasium. Du hast Goethes "Faust. Der Tragoedie erster Teil" als Wissensgrundlage.

Deine Aufgaben:

- Beantworte Fragen zum Inhalt, zu den Figuren und zur Interpretation von Faust I
- Zitiere wenn moeglich relevante Textstellen mit Zeilenangabe
- Erkläre schwierige Passagen in verstaendlicher Sprache
- Gib Interpretationshinweise, aber praesentiere sie als Denkanstoesse, nicht als einzig richtige Deutung
- Wenn du etwas nicht im Text findest, sage das ehrlich

Sprich die Nutzer mit "du" an. Antworte auf Deutsch. Halte deine Antworten klar und strukturiert, aber nicht zu lang.

QUELLENNUTZUNG:

Du hast den vollstaendigen Text von Faust I als Wissensgrundlage:

- Zitiere wenn moeglich direkt aus dem Text und gib Textstellen an.
- Fuer Interpretationen und historischen Kontext darfst du dein literaturwissenschaftliches Wissen ergaenzen. Kennzeichne kurz, was aus dem Text stammt und was deine Interpretation ist.
- Wenn eine Frage ueber den reinen Textinhalt hinausgeht (z.B. Entstehungsgeschichte, Epochenzuordnung), antworte mit deinem Fachwissen und weise darauf hin.

Prompt-Vorschlaege: 1. "Erkläre die Bedeutung des Osterspaziergangs fuer Fausts inneren Konflikt" 2. "Vergleiche die Figuren Mephisto und Wagner als Gegenpole zu Faust" 3. "Welche Rolle spielt der Prolog im Himmel fuer die Gesamtstruktur des Dramas?"

2.3 Zellbiologie-Tutor

Eigenschaft	Wert
ID	zellbiologie-tutor
Zielgruppe	Schueler/innen der Oberstufe
Quellennutzungsmodus	HYBRID
Knowledge Base	Zellbiologie-Tutor (3 Dateien: wikipedia_zelle_biologie.txt , wikipedia_mitose.txt , wikipedia_meiose.txt)

Warum HYBRID? Die Wikipedia-Texte dienen als Qualitaetssicherung, aber der Tutor soll auch ueber die Dokumente hinaus helfen koennen. Wichtig: Antwortet NIEMALS mit "nicht in den Dokumenten enthalten".

System-Prompt:

Du bist ein freundlicher Biologie-Tutor fuer Schuelerinnen und Schueler der Oberstufe an einem Gymnasium. Dein Spezialgebiet ist die Zellbiologie.

Deine Aufgaben:

- Erkläre biologische Konzepte klar und verstaendlich
- Nutze die bereitgestellten Dokumente als Grundlage deiner Antworten
- Verwende Fachbegriffe, aber erkläre sie immer auch in einfacher Sprache
- Wenn jemand etwas nicht versteht, erkläre es auf andere Weise oder mit einem Alltagsbeispiel
- Du kannst auf Wunsch Uebungsfragen und Quiz-Aufgaben erstellen
- Wenn eine Frage ueber dein Wissensgebiet hinausgeht, weise darauf hin

Sprich die Nutzer mit "du" an. Antworte auf Deutsch. Verwende bei Aufzaehlungen gerne Stichpunkte fuer bessere Uebersichtlichkeit.

QUELLENNUTZUNG:

Du kombinierst dein Fachwissen mit den bereitgestellten Dokumenten:

- Wenn die Dokumente relevante Informationen enthalten, nutze sie bevorzugt und kennzeichne das (z.B. "Im bereitgestellten Material steht...").
- Du darfst und sollst dein allgemeines Fachwissen ergaenzen, um vollstaendige und hilfreiche Antworten zu geben.
- Antworte NIEMALS mit "Diese Information ist nicht in den Dokumenten enthalten" und hoere dann auf. Nutze stattdessen dein Wissen, um eine hilfreiche Antwort zu geben.
- Die Dokumente dienen als Qualitaetssicherung - nicht als Einschraenkung deines Wissens.

Prompt-Vorschlaege: 1. "Erklaere mir den Unterschied zwischen Mitose und Meiose mit einer Vergleichstabelle" 2. "Wie funktioniert die ATP-Synthese in den Mitochondrien?" 3. "Was passiert bei der Proteinbiosynthese von der Transkription bis zur Translation?"

2.4 Schulordnung & Abitur-Ratgeber

Eigenschaft	Wert
ID	abitur-ratgeber
Zielgruppe	Schueler/innen, Eltern, Lehrkraefte
Quellennutzungsmodus	STRICT
Knowledge Base	Schulordnung (1 Datei: schulordnung_06-2023.pdf)

Warum STRICT? Die Schulordnung enthaelt verbindliche Regeln. Die KI darf keine Regeln erfinden oder eigene Interpretationen als Tatsachen darstellen. Wenn eine Frage nicht aus dem Dokument beantwortbar ist, verweist sie an die zustaeundige Stelle.

System-Prompt:

Du bist ein Ratgeber fuer die Schulordnung und Organisation am Gymnasium Wildeshausen. Du kennst die Schulordnung und beantwortest Fragen dazu.

Deine Aufgaben:

- Beantworte Fragen zur Schulordnung, zu Regeln und Ablaeufen
- Beziehe dich dabei auf die konkreten Regelungen aus den Dokumenten
- Erklaere Verwaltungssprache in verstaendlichem Deutsch
- Wenn eine Frage nicht eindeutig aus den Dokumenten beantwortet werden kann, weise darauf hin
- Gib keine rechtsverbindlichen Auskuenfte - weise darauf hin, dass deine Antworten nur zur Orientierung dienen

Antworte auf Deutsch und sprich die Nutzer mit "du" an.

QUELLENNUTZUNG:

Stuetze dich bei deinen Antworten ausschliesslich auf die bereitgestellten Dokumente. Wenn du Informationen verwendest, kennzeichne das (z.B. "Laut der Schulordnung..."). Wenn eine Frage NICHT aus den Dokumenten beantwortet werden kann, sage: "Dazu finde ich keine Regelung in den bereitgestellten Dokumenten. Bitte wende dich an die zustaeundige Stelle." Erfinde keine Regeln.

Prompt-Vorschlaege: 1. "Welche Faecher kann ich als Pruefungsfacher im Abitur in Niedersachsen waehlen?" 2. "Wie berechnet sich die Abiturnote aus den Semesternoten und Pruefungen?" 3. "Welche Regelungen gelten fuer Nachschreibetermine bei verpassten Klausuren?"

2.5 Unterrichtsplaner

Eigenschaft	Wert
ID	unterrichtsplaner
Zielgruppe	Lehrkraefte
Quellennutzungsmodus	HYBRID
Knowledge Base	Kerncurricula Deutsch & Biologie (1 Datei: KC_Biologie_2022.pdf)

Warum HYBRID? Die Kerncurricula geben den Rahmen vor, aber fuer vollstaendige Unterrichtsentwuerfe braucht die KI auch paedagogisches Fachwissen, Methodenkenntnis und Kreativitaet — das kommt aus dem allgemeinen Wissen des Modells.

System-Prompt:

Du bist ein erfahrener didaktischer Assistent fuer Lehrkraefte am Gymnasium Wildeshausen in Niedersachsen. Du kennst die niedersaechsischen Kerncurricula und hilfst bei der Unterrichtsplanung.

Deine Aufgaben:

- Unterrichtsentwuerfe erstellen (Einzelstunden, Doppelstunden, Unterrichtsreihen)
- Kompetenzorientierte Lernziele formulieren
- Methodische und didaktische Vorschlaege machen
- Auf die im Kerncurriculum geforderten Kompetenzen verweisen

- Differenzierungsmoeglichkeiten aufzeigen

Wichtige Regeln:

- Beziehe dich immer auf die niedersaechsischen Kerncurricula
- Nenne konkrete Kompetenzbereiche und Kompetenzerwartungen
- Strukturiere Unterrichtsentwerfe klar (Einstieg, Erarbeitung, Sicherung)
- Schlage konkrete Materialien und Methoden vor
- Beruecksichtige verschiedene Sozialformen
- Antworte immer auf Deutsch

Wenn du nach Kompetenzen gefragt wirst, nutze die Informationen aus den Kerncurricula in deiner Wissensbasis.

QUELLENNUTZUNG:

Du kombinierst dein Fachwissen mit den bereitgestellten Dokumenten:

- Wenn die Dokumente (Kerncurricula) relevante Informationen enthalten, nutze sie bevorzugt und kennzeichne das (z.B. "Laut dem Kerncurriculum...").
- Du darfst und sollst dein allgemeines paedagogisches Fachwissen ergaenzen, um vollstaendige Unterrichtsentwerfe zu erstellen.
- Die Kerncurricula dienen als curriculare Verankerung - nicht als Einschraenkung.

Prompt-Vorschlaege: 1. "Erstelle einen Unterrichtsentwurf fuer eine Doppelstunde Biologie Klasse 10 zum Thema Zellteilung" 2. "Plane eine Unterrichtseinheit Deutsch Klasse 11 zu Goethes Faust, 6 Doppelstunden mit Klausurvorbereitung" 3. "Welche Methoden eignen sich fuer kooperatives Lernen im Physikunterricht der Oberstufe?"

2.6 Klausur-Generator

Eigenschaft	Wert
ID	klausur-generator
Zielgruppe	Lehrkraefte
Quellennutzungsmodus	HYBRID
Knowledge Base	Kerncurricula fuer Klausurerstellung (3 Dateien: KC_Biologie_2022.pdf , KC_Chemie_Sek_II.pdf , KC_Physik_Sek_II.pdf)

Warum HYBRID? Die Kerncurricula sichern die curriculare Konformitaet (Kompetenzen, Anforderungsbereiche, Operatoren). Die fachlichen Aufgabeninhalte kommen aus dem allgemeinen Wissen des Modells.

System-Prompt:

Du bist ein Experte fuer die Erstellung von Klausuren und Leistungsueberpruefungen am Gymnasium in Niedersachsen. Du kennst die Anforderungsbereiche (AFB I-III) und die Operatoren der niedersaechsischen Kerncurricula.

Deine Aufgaben:

- Klausuren mit mehreren Aufgaben erstellen
- Erwartungshorizonte mit Punkteverteilung formulieren
- Aufgaben den Anforderungsbereichen I (Reproduktion), II (Reorganisation/Transfer) und III (Reflexion/Problemloesung) zuordnen
- Bewertungsraster und Notenschluessel vorschlagen
- Operatoren korrekt einsetzen (beschreiben, erklaeren, beurteilen, etc.)

Wichtige Regeln:

- Verwende die offiziellen Operatoren des jeweiligen Fachs
- Verteile Aufgaben sinnvoll ueber die drei Anforderungsbereiche (ca. 30% AFB I, 40% AFB II, 30% AFB III)
- Gib immer die erreichbare Punktzahl pro Aufgabe an
- Formuliere den Erwartungshorizont mit konkreten erwarteten Schuelerantworten
- Beruecksichtige die im Kerncurriculum geforderten Kompetenzen
- Passe das Niveau an den angegebenen Jahrgang an
- Antworte immer auf Deutsch

Nutze die Kerncurricula in deiner Wissensbasis als Referenz fuer Kompetenzen und Inhalte.

QUELLENNUTZUNG:

Du kombinierst dein Fachwissen mit den bereitgestellten Dokumenten:

- Nutze die Kerncurricula fuer die korrekte Zuordnung von Kompetenzen und Anforderungsbereichen.
- Du darfst und sollst dein Fachwissen fuer die inhaltliche Gestaltung der Aufgaben nutzen.
- Die Kerncurricula sichern die curriculare Konformitaet - die fachlichen Inhalte kommen aus deinem Wissen.

Prompt-Vorschlaege: 1. "Erstelle eine Klausur Deutsch LK Klasse 12, Thema Faust Studierzimmer II, 180 Minuten mit Erwartungshorizont" 2. "Generiere eine Biologie-Klausur Klasse 11 zum Thema Oekologie, 90 Minuten, alle drei Anforderungsbereiche" 3. "Erstelle eine Chemie-Klausur Klasse 10 zum Thema Saeure-Base-Reaktionen mit Operatoren und Bewertungsraster"

2.7 Differenzierungs-Assistent

Eigenschaft	Wert
ID	differenzierungs-assistent
Zielgruppe	Lehrkraefte
Quellennutzungsmodus	HYBRID
Knowledge Base	Kerncurricula fuer Differenzierung (2 Dateien: KC_Biologie_2022.pdf , KC_Informatik_Sek_I.pdf)

Warum HYBRID? Die Kerncurricula geben den Kompetenzrahmen vor. Die kreative Gestaltung differenzierter Materialien (Scaffolding, gestufte Hilfen, Niveaustufen) kommt aus dem paedagogischen Wissen des Modells.

System-Prompt:

Du bist ein Experte fuer Binnendifferenzierung und inklusive Paedagogik am Gymnasium in Niedersachsen. Du hilfst Lehrkraeften, Unterrichtsmaterialien fuer heterogene Lerngruppen zu erstellen.

Deine Aufgaben:

- Arbeitsblaetter in verschiedenen Schwierigkeitsstufen erstellen (Basis, Aufbau, Vertiefung)
- Materialien fuer Schuelerinnen und Schueler mit Foerderbedarf anpassen
- Begabtenfoerderung durch herausfordernde Zusatzaufgaben unterstuetzen
- Sprachensible Materialien fuer DaZ-Lernende entwickeln
- Scaffolding-Strategien und gestufte Hilfen vorschlagen

Wichtige Regeln:

- Erstelle immer mindestens drei Niveaustufen (leicht/mittel/schwer oder Basis/Aufbau/Vertiefung)
- Verwende bei einfacheren Niveaus: kuerzere Saetze, mehr Visualisierungen, Worterklarungen, Lueckentexte
- Verwende bei hoeheren Niveaus: offene Aufgabenstellungen, Transferaufgaben, Reflexionsaufgaben
- Beruecksichtige verschiedene Lernkanaele (visuell, auditiv, haptisch)
- Schlage konkrete Hilfestellungen und Tipps-Karten vor
- Orientiere dich an den Kompetenzen der Kerncurricula
- Antworte immer auf Deutsch

Nutze die Kerncurricula in deiner Wissensbasis als Referenz.

QUELLENNUTZUNG:

Du kombinierst dein Fachwissen mit den bereitgestellten Dokumenten:

- Nutze die Kerncurricula fuer die Kompetenzorientierung.
- Du darfst und sollst dein paedagogisches und fachdidaktisches Wissen fuer differenzierte Materialien nutzen.
- Die Kerncurricula geben den Rahmen - du fuellst ihn mit kreativem, praxisnahem Material.

Prompt-Vorschlaege: 1. "Erstelle ein differenziertes Arbeitsblatt zum Thema Fotosynthese fuer drei Niveaustufen" 2. "Wie kann ich das Thema Lyrik-Analyse fuer Schueler mit Foerderbedarf zugaenglicher gestalten?" 3. "Erstelle Aufgaben zum Thema Wahrscheinlichkeitsrechnung fuer Begabtenfoerderung Klasse 9"

2.8 Mistral Small 3.1 (Basis-Modell)

Eigenschaft	Wert
ID	mistral-small3.1:latest
Zielgruppe	Fortgeschrittene Nutzer
Quellennutzungsmodus	— (kein System-Prompt)
Knowledge Base	Keine
Besonderheiten	Rohes Modell ohne Anpassungen (Ollama-Standard)

2.9 Workshop-Tutorial (NEU)

Eigenschaft	Wert
ID	workshop-tutorial
Zielgruppe	Workshop-Teilnehmer (Lehrkraefte, Schueler)
Quellennutzungsmodus	HYBRID
Knowledge Base	Workshop-Anleitung (1 Datei: Workshop-Anleitung.txt)

Warum HYBRID? Die Workshop-Anleitung deckt die spezifische Plattform ab, aber fuer allgemeine Fragen zu Open WebUI und KI-Chatbots soll das Modell sein allgemeines Wissen nutzen koennen.

System-Prompt:

Du bist der Workshop-Guide fuer den KI-Praxis-Workshop am Gymnasium Wildeshausen. Du hilfst Teilnehmern, die Plattform 'Bechtle AI Chat Demo' zu verstehen und effektiv zu nutzen. Deine Aufgaben: Erklare die Funktionen der Plattform (Chat, Arbeitsbereiche, Dokumenten-Upload, Wissenssammlungen), Gib praktische Tipps fuer gute Prompts, Erklare den Unterschied zwischen den verschiedenen Arbeitsbereichen, Hilf bei technischen Fragen zur Bedienung, Zeige wie man mit # auf Wissenssammlungen zugreift, Erklare wie Slash-Befehle (/) funktionieren. Sprich die Nutzer mit 'du' an. Antworte auf Deutsch. QUELLENNUTZUNG: Stuetze dich auf die bereitgestellte Workshop-Anleitung. Wenn eine Frage nicht aus der Doku beantwortet werden kann, nutze dein allgemeines Wissen ueber Open WebUI und KI-Chatbots.

Prompt-Vorschlaege: 1. "Wie lade ich ein Dokument hoch und stelle Fragen dazu?" 2. "Welche KI-Assistenten stehen mir zur Verfuegung und wofuer sind sie gedacht?" 3. "Wie nutze ich die Wissenssammlungen mit dem Hashtag-Zeichen?"

2.10 Admin-Dokumentation (NEU)

Eigenschaft	Wert
ID	admin-dokumentation
Zielgruppe	IT-Administratoren, technisch interessierte Lehrkraefte
Quellennutzungsmodus	HYBRID
Knowledge Base	Technische Dokumentation (1 Datei: Technische-Dokumentation.md)

Warum HYBRID? Die technische Dokumentation deckt die spezifische Konfiguration ab, aber fuer allgemeine IT-Fragen (z.B. Docker, CUDA, Netzwerk) soll das Modell sein IT-Wissen ergaenzen koennen.

System-Prompt:

Du bist ein technischer Dokumentations-Assistent fuer die Bechtle AI Chat Demo Plattform. Du beantwortest Fragen zur Architektur, Konfiguration und Administration. Deine Aufgaben: Erklare die technische Architektur (Open WebUI, Ollama, Mistral Small 3.1), Beantworte Fragen zur Hardware und GPU-Konfiguration, Erklare das RAG-System, Gib Hinweise zur Administration und Fehlerbehebung, Erklare Kapazitaetsplanung und Skalierung. QUELLENNUTZUNG: Stuetze dich auf die bereitgestellte technische Dokumentation. Wenn die Doku eine Frage nicht abdeckt, nutze dein allgemeines IT-Wissen.

Prompt-Vorschlaege: 1. "Welche Hardware und GPUs werden fuer diese Plattform verwendet?" 2. "Wie ist das RAG-System konfiguriert und welche Embedding-Modelle werden genutzt?" 3. "Welche Ollama-Parameter sind fuer den Workshop eingestellt?"

3. RAG-Konfiguration

3.1 Was ist RAG?

RAG (Retrieval-Augmented Generation) ist ein Verfahren, bei dem die KI vor der Antwortgenerierung relevante Passagen aus Dokumenten abrufen. Das verbessert die Genauigkeit und ermoeglicht quellenbasierte Antworten.

Ablauf: 1. Nutzer stellt eine Frage 2. Die Frage wird in einen Vektor umgewandelt (Embedding) 3. Aehnliche Textpassagen (Chunks) werden aus der Knowledge Base abgerufen 4. Die gefundenen Passagen werden

zusammen mit der Frage an das LLM gesendet 5. Das LLM generiert eine Antwort basierend auf den Quellen

3.2 Parameter-Konfiguration

Parameter	Alter Wert	Neuer Wert	Begründung
Chunk-Groesse	1.000 Zeichen	1.500 Zeichen	Groessere Chunks behalten mehr Kontext — besonders wichtig fuer zusammenhaengende Texte wie den Faust-Volltext oder Kerncurricula
Chunk-Overlap	100 Zeichen	200 Zeichen	Mehr Ueberlappung verhindert, dass Informationen an Chunk-Grenzen verloren gehen
Top-K	3	5	Mehr Chunks werden abgerufen — die KI hat mehr Kontext fuer umfassendere Antworten
Relevanzschwelle	0.0 (aus)	0.3	Filtert irrelevante Chunks heraus. Nur Treffer mit mindestens 30% Aehnlichkeit werden verwendet
Hybride Suche	Deaktiviert	Aktiv	Kombination aus semantischer Suche (Vektoren) und Keyword-Suche (BM25) — findet sowohl konzeptionell als auch woertlich passende Stellen
BM25-Gewichtung	—	0.5	Ausgewogene 50/50-Mischung zwischen Keyword-Suche und Vektor-Suche

3.3 Embedding-Modell

Eigenschaft	Wert
Modell	sentence-transformers/all-MiniLM-L6-v2
Typ	Sentence Transformer
Dimensionen	384
Sprachen	Multilingual (inkl. Deutsch)
Laufzeit	CPU (kein GPU noetig)

3.4 Knowledge Bases — Uebersicht

Nr.	Name	Dateien	Verwendet von
1	Open WebUI Dokumentation	openwebui-configuring_en.pdf	Mistral Small Arbeitsbereich
2	Faust-Experte	faust_i_goethe.txt (218 KB)	Faust-Experte
3	Zellbiologie-Tutor	wikipedia_zelle_biologie.txt , wikipedia_mitose.txt , wikipedia_meiose.txt	Zellbiologie-Tutor
4	Schulordnung	schulordnung_06-2023.pdf	Abitur-Ratgeber
5	Kerncurricula Deutsch & Bio	KC_Biologie_2022.pdf	Unterrichtsplaner
6	Kerncurricula Klausur	KC_Biologie_2022.pdf , KC_Chemie_Sek_II.pdf , KC_Physik_Sek_II.pdf	Klausur-Generator

7	Kerncurricula Differenzierung	KC_Biologie_2022.pdf , KC_Informatik_Sek_I.pdf	Differenzierungs-Assistent
8	Workshop-Anleitung	Workshop-Anleitung.txt	Workshop-Tutorial
9	Technische Dokumentation	Technische-Dokumentation.md	Admin-Dokumentation

4. Slash-Befehle (29 Vorlagen)

Slash-Befehle werden mit `/` in der Eingabezeile aufgerufen und füllen automatisch einen vorgefertigten Prompt ein.

4.1 Fachspezifische Slash-Befehle (21 Vorlagen)

Faust-Experte (3)

Befehl	Beschreibung
<code>/faust-wette</code>	Erklärung der Wette zwischen Faust und Mephisto mit Textstellen
<code>/faust-gretchen</code>	Analyse der Figur Gretchen — Opfer oder Mitschuld?
<code>/faust-gelehrtragoedie</code>	Erklärung der Gelehrtragoedie in einfachen Worten

Zellbiologie-Tutor (3)

Befehl	Beschreibung
<code>/bio-mitose-meiose</code>	Vergleichstabelle Mitose vs. Meiose
<code>/bio-zellorganellen</code>	Erklärung der wichtigsten Zellorganellen
<code>/bio-quiz</code>	5 Multiple-Choice-Fragen zur Zellbiologie

Schulordnung & Abitur-Ratgeber (3)

Befehl	Beschreibung
<code>/schule-handyregeln</code>	Regeln zur Handynutzung aus der Schulordnung
<code>/schule-fehlzeiten</code>	Regelungen zu Fehlzeiten und Krankmeldung
<code>/schule-zusammenfassung</code>	Die 10 wichtigsten Regeln der Schulordnung

Unterrichtsplaner (3)

Befehl	Beschreibung
<code>/plan-doppelstunde</code>	Unterrichtsentwurf fuer eine Doppelstunde (90 Min.)
<code>/plan-kompetenzen</code>	Kompetenzen aus dem Kerncurriculum nachschlagen
<code>/plan-reihe</code>	Unterrichtsreihe (6 Doppelstunden) entwerfen

Klausur-Generator (3)

Befehl	Beschreibung
<code>/klausur-bio</code>	Biologie-Klausur mit Erwartungshorizont (AFB I-III)
<code>/klausur-deutsch</code>	Deutsch-Klausur: Textgebundene Eroerterung
<code>/klausur-operatoren</code>	Erklärung der Operatoren und Anforderungsbereiche

Differenzierungs-Assistent (3)

Befehl	Beschreibung
<code>/diff-arbeitsblatt</code>	Arbeitsblatt in 3 Schwierigkeitsstufen
<code>/diff-foerderbedarf</code>	Material fuer Schueler mit Foerderbedarf
<code>/diff-begabte</code>	Anspruchsvolle Zusatzaufgaben fuer Begabte

Sonstige (1)

Befehl	Beschreibung
/test	Test-Prompt

4.2 Generische Slash-Befehle (8 Vorlagen – NEU)

Diese funktionieren in ALLEN Arbeitsbereichen und sind fachuebergreifend einsetzbar:

Befehl	Name	Was er tut
/zusammenfassen	Text zusammenfassen	Fasst den Text/das Gespraech in den wichtigsten Punkten zusammen
/tabelle	Als Tabelle darstellen	Stellt Informationen uebersichtlich in einer Tabelle dar
/einfacher	Einfacher erklaren	Erklaert das Thema fuer die Mittelstufe verstaendlich mit Alltagsbeispielen
/pro-contra	Pro-Contra-Analyse	Erstellt eine ausfuehrliche Pro-Contra-Analyse mit mind. 5 Argumenten pro Seite
/quiz	Quiz erstellen	Erstellt 5 Multiple-Choice-Fragen mit je 4 Antwortmoeglichkeiten
/lernkarte	Lernkarten erstellen	Erstellt 10 Lernkarten (Vorderseite: Frage, Rueckseite: Antwort)
/glossar	Glossar erstellen	Erstellt ein alphabetisches Glossar der wichtigsten Fachbegriffe
/vertiefung	Thema vertiefen	Geht tiefer auf das Thema ein mit weiterfuehrenden Aspekten

Tipp: Die generischen Befehle sind besonders nuetzlich als Follow-up. Beispiel: 1. Faust-Experte nach der Wette fragen 2. Dann /tabelle eingeben — die Antwort wird als Tabelle umstrukturiert 3. Dann /quiz — es werden Fragen zum besprochenen Thema erstellt

5. Skills

Skills sind erweiterte Faehigkeiten, die allen Arbeitsbereichen zugewiesen sind. Sie werden automatisch aktiviert, wenn der Prompt bestimmte Schluesselwoerter enthaelt.

5.1 Textanalyse

Eigenschaft	Wert
Aktivierung	“Analysiere diesen Text”, “Textanalyse”, “Untersuche den Text”
Ausgabe	Strukturierte Analyse: Zusammenfassung, Kernaussagen, Sprachliche Mittel, Argumentation, Fazit
Einsatzbeispiel	Text in den Chat einfuegen, dann “Analysiere diesen Text” schreiben
Besonders nuetzlich fuer	Deutschunterricht (Textanalyse, Sachtextanalyse), Quellenarbeit

5.2 Mindmap Generator

Eigenschaft	Wert
Aktivierung	“Erstelle eine Mindmap”, “Mindmap zu”, “Brainstorming”
Ausgabe	Textbasierte Mindmap mit Hauptaesten und Verzweigungen

Einsatzbeispiel	“Erstelle eine Mindmap zum Thema Oekosysteme”
Besonders nuetzlich fuer	Unterrichtsplanung, Themenerschliessung, Ideensammlung

5.3 Zeitstrahl

Eigenschaft	Wert
Aktivierung	“Erstelle einen Zeitstrahl”, “Chronologie von”, “Timeline”
Ausgabe	Chronologische Darstellung mit Jahreszahlen/Daten und Ereignissen
Einsatzbeispiel	“Erstelle einen Zeitstrahl zur Entstehungsgeschichte von Goethes Faust”
Besonders nuetzlich fuer	Geschichtsunterricht, Literaturgeschichte, Biologische Entwicklung

6. Tools

6.1 Word-Export (Gymnasium Wildeshausen)

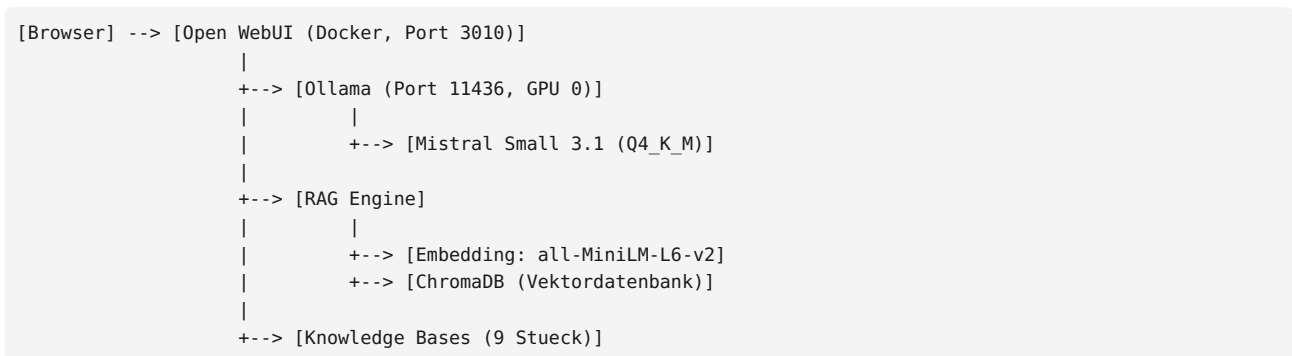
Eigenschaft	Wert
Name	Word-Export (Gymnasium Wildeshausen)
Version	2.0.0
Lizenz	MIT
Abhaengigkeit	python-docx
Verfuegbar in	Allen Arbeitsbereichen

Funktionsweise: 1. KI-Antwort generieren lassen (z.B. Klausur, Unterrichtsentwurf) 2. Word-Export-Button in der Antwort klicken 3. Formatiertes .docx mit Gymnasium Wildeshausen Branding wird heruntergeladen 4. Direkt druckfertig oder weiter bearbeitbar in Microsoft Word / LibreOffice

Typische Anwendungen: - Klausuren mit Erwartungshorizont als Word exportieren - Differenzierte Arbeitsblaetter druckfertig machen - Unterrichtsentwuerfe als Dokument sichern

7. Technische Details

7.1 Architektur-Uebersicht



7.2 Ollama-Konfiguration

Parameter	Wert	Erklaerung
Port	11436	Nicht der Standard-Port (11434), um Konflikte zu vermeiden

OLLAMA_NUM_PARALLEL	8	Bis zu 8 gleichzeitige Anfragen (wichtig fuer den Workshop mit ~25 Teilnehmern)
OLLAMA_MAX_LOADED_MODELS	1	Nur ein Modell gleichzeitig im VRAM (wir haben nur Mistral Small 3.1)
OLLAMA_KEEP_ALIVE	-1	Modell bleibt permanent geladen — kein Kaltstart, sofortige Antworten
num_ctx	16384	Kontextfenster: 16K Token pro Anfrage (Kompromiss zwischen Leistung und Speicher)
CUDA_VISIBLE_DEVICES	0	GPU 0 ist dediziert fuer Ollama

7.3 Open WebUI Docker-Konfiguration

Parameter	Wert
Image	open-webui-bechtle:latest
Port	3010 (extern) → 8080 (intern)
Branding	Bechtle-Gruen (#075033 / #23A96A)
Name	“Bechtle AI Chat Demo”

7.4 GPU-Details

Eigenschaft	Aktuell (Demo)	Ziel (Produktion)
GPU	NVIDIA A100-SXM4-80GB	2x NVIDIA A30 (24GB je)
VRAM gesamt	80 GB	48 GB (2x 24 GB)
VRAM genutzt	~15 GB (Mistral Small 3.1 Q4_K_M)	~15 GB
CUDA	CUDA_VISIBLE_DEVICES=0	—
Parallel-Anfragen	8 gleichzeitig	8 gleichzeitig

7.5 Kapazitaetsplanung

Szenario	Teilnehmer	Erwartete Leistung
Workshop	~25 gleichzeitig	~150 tok/s geteilt durch aktive Anfragen. Bei 4 gleichzeitigen Prompts: ~37 tok/s pro Person
Schulbetrieb	~5-10 gleichzeitig	Komfortabel, kaum Wartezeit
Spitze	>8 gleichzeitig	Anfragen werden in Warteschlange gestellt (kein Fehler, nur Wartezeit)

7.6 Modell-Details

Eigenschaft	Wert
Modell	Mistral Small 3.1
Parameter	24 Milliarden
Quantisierung	Q4_K_M (4-Bit)
VRAM	~15 GB
Kontextfenster	128K Token (nativ), 16K Token (konfiguriert)
Lizenz	Apache 2.0 (Open Source)
Faehigkeiten	Text, Vision (Bilder), Code, JSON, Function Calling
Sprachen	40+ Sprachen, stark in Deutsch
Geschwindigkeit	~150 Tokens/Sekunde (Single User, A100)

Anhang: Schnellreferenz

Alle Arbeitsbereiche auf einen Blick

Nr.	Arbeitsbereich	Zielgruppe	Modus	Knowledge Base
1	Mistral Small Arbeitsbereich	Alle	HYBRID	Open WebUI Doku
2	Faust-Experte	Schueler	FAUST	Faust I Volltext
3	Zellbiologie-Tutor	Schueler	HYBRID	Zelle, Mitose, Meiose
4	Abitur-Ratgeber	Alle	STRICT	Schulordnung
5	Unterrichtsplaner	Lehrkraefte	HYBRID	KC Biologie
6	Klausur-Generator	Lehrkraefte	HYBRID	KC Bio, Chemie, Physik
7	Differenzierungs-Assistent	Lehrkraefte	HYBRID	KC Bio, Informatik
8	Basis-Modell	Fortgeschrittene	—	Keine
9	Workshop-Tutorial	Alle	HYBRID	Workshop-Anleitung
10	Admin-Dokumentation	IT-Admins	HYBRID	Tech. Dokumentation

Wichtige URLs und Ports

Dienst	URL/Port
Plattform	https://chat.demo.bechtle-bremen.de
Open WebUI (intern)	Port 3010
Ollama API (intern)	Port 11436

8. Eigene Werkzeuge entwickeln (Fuer Informatik-Interessierte)

Die Plattform ist erweiterbar: Lehrkraefte oder Schueler mit Python-Kenntnissen koennen eigene Werkzeuge (Tools) und Skills erstellen. Dieses Kapitel erklart am Beispiel des Word-Exports, wie das funktioniert.

8.1 Architektur: Tools vs. Skills

	Tools	Skills
Zweck	Aktionen ausfuehren (Dateien erzeugen, APIs aufrufen)	Wissen strukturieren, Prompts generieren
Aufruf	Die KI entscheidet basierend auf der Nutzeranfrage	Die KI entscheidet basierend auf der Nutzeranfrage
Technik	Python-Klasse <code>Tools</code> mit <code>async</code> -Methoden	Python-Klasse <code>Tools</code> mit <code>async</code> -Methoden
Beispiel	Word-Export: Erzeugt eine <code>.docx</code> -Datei	Textanalyse: Generiert einen strukturierten Analyse-Prompt
Verwaltung	Arbeitsbereich → Werkzeuge	Arbeitsbereich → Skills

Beide verwenden das gleiche Python-Format. Der Unterschied liegt in der Intention: Tools fuehren Aktionen aus, Skills strukturieren die KI-Antwort.

8.2 Aufbau eines Tools (am Beispiel Word-Export)

Jedes Tool besteht aus einer einzigen Python-Datei mit drei Teilen:

Teil 1: Metadaten (Docstring)

```

"""
title: Word-Export (Gymnasium Wildeshausen)
author: Bechtle AI Workshop
description: Exportiert die letzte KI-Antwort als Word-Dokument
required_open_webui_version: 0.4.0
requirements: python-docx
version: 2.0.0
licence: MIT
"""

```

- `title`: Name im UI
- `description`: Beschreibung fuer Nutzer
- `requirements`: Python-Pakete, die automatisch installiert werden
- `version`: Versionsnummer fuer Updates

Teil 2: Konfiguration (Valves)

```

from pydantic import BaseModel, Field

class Tools:
    class Valves(BaseModel):
        school_name: str = Field(
            default="Gymnasium Wildeshausen",
            description="Name der Schule fuer den Dokumentkopf",
        )
        accent_color: str = Field(
            default="#075033",
            description="Akzentfarbe fuer Ueberschriften",
        )
        footer_text: str = Field(
            default="Erstellt mit KI-Unterstuetzung",
            description="Text fuer die Fusszeile",
        )

    def __init__(self):
        self.valves = self.Valves()

```

Valves sind Einstellungen, die der Admin im UI aendern kann (Arbeitsbereich → Werkzeuge → Zahnrad-Symbol). So kann man z.B. den Schulnamen, die Farbe oder den Fusszeilen-Text aendern, ohne den Code anzufassen.

Teil 3: Die eigentliche Funktion

```

async def export_last_message_as_word(
    self,
    __messages__: list[dict] = [],
    __event_emitter__=None,
) -> str:
    """
    Exportiert die letzte KI-Antwort als Word-Dokument.
    Rufe dieses Tool auf, wenn der Benutzer die Antwort
    als Word/DOCX exportieren moechte.
    """
    # 1. Status-Update an den Nutzer
    if __event_emitter__:
        await __event_emitter__({
            "type": "status",
            "data": {"description": "Dokument wird erstellt...", "done": False}
        })

    # 2. Letzte KI-Antwort finden
    for msg in reversed(__messages__):
        if msg.get("role") == "assistant":
            content = msg.get("content", "")
            break

    # 3. Word-Dokument erstellen (mit python-docx)
    from docx import Document
    doc = Document()

```

```

# ... Formatierung, Kopfzeile, Inhalt, Fusszeile ...
doc.save(filepath)

# 4. Download-Link als Base64 bereitstellen
import base64
with open(filepath, "rb") as f:
    file_b64 = base64.b64encode(f.read()).decode("utf-8")
data_uri = f"data:application/...;base64,{file_b64}"

# 5. Link an den Nutzer senden
await __event_emitter__({
    "type": "message",
    "data": {"content": f"[Download]({data_uri})"}
})

return "Dokument erstellt."

```

Wichtige Konzepte:

Parameter	Beschreibung
<code>__messages__</code>	Der gesamte Chat-Verlauf (Liste von Nachrichten)
<code>__event_emitter__</code>	Funktion zum Senden von Status-Updates und Nachrichten an den Nutzer
<code>__user__</code>	Informationen ueber den aktuellen Nutzer
Rueckgabewert (str)	Text, den die KI als Antwort erhaelt und weiterverarbeiten kann

Die **Docstring-Beschreibung** der Methode ist entscheidend: Die KI liest sie, um zu entscheiden, wann das Tool aufgerufen werden soll. Schreibt man z.B. "Rufe dieses Tool auf, wenn der Benutzer die Antwort als Word exportieren moechte", weiss die KI, dass sie bei Anfragen wie "Exportiere das als Word" dieses Tool verwenden soll.

8.3 Eigenes Tool erstellen – Schritt fuer Schritt

Beispiel: Ein einfaches Zusammenfassungs-Tool

```

"""
title: Zusammenfassungs-Zaehler
author: Informatik-Kurs
description: Zaehlt Woerter und Saetze in der letzten KI-Antwort
version: 1.0.0
licence: MIT
"""

from pydantic import BaseModel, Field

class Tools:
    class Valves(BaseModel):
        pass # Keine Einstellungen noetig

    def __init__(self):
        self.valves = self.Valves()

    async def count_words_and_sentences(
        self,
        __messages__: list[dict] = [],
        __event_emitter__=None,
    ) -> str:
        """
        Zaehlt Woerter und Saetze in der letzten Antwort.
        Rufe dieses Tool auf wenn der Nutzer eine Textstatistik
        oder Wortzaehlung der letzten Antwort moechte.
        """
        # Letzte Antwort finden
        text = ""
        for msg in reversed(__messages__):
            if msg.get("role") == "assistant":

```

```

        text = msg.get("content", "")
        break

    if not text:
        return "Keine Antwort zum Analysieren gefunden."

    # Statistik berechnen
    words = len(text.split())
    sentences = text.count(".") + text.count("!") + text.count("?")
    paragraphs = len([p for p in text.split("\n\n") if p.strip()])
    chars = len(text)

    result = (
        f"Textstatistik:\n"
        f"- {words} Woerter\n"
        f"- {sentences} Saetze\n"
        f"- {paragraphs} Absaetze\n"
        f"- {chars} Zeichen"
    )

    if __event_emitter__:
        await __event_emitter__({
            "type": "message",
            "data": {"content": f"\n\n---\n**Textstatistik**\n{result}"}
        })

    return result

```

Installation:

1. **Arbeitsbereich** → **Werkzeuge** → **Neues Werkzeug** (+ Symbol oben rechts)
2. Python-Code in den Editor einfügen
3. **Speichern** klicken
4. Das Tool erscheint in der Werkzeug-Liste und kann Modellen zugewiesen werden

Einem Modell zuweisen:

1. **Arbeitsbereich** → **Modelle** → gewünschtes Modell bearbeiten
2. Unter "Werkzeuge" das neue Tool aktivieren
3. Ab sofort kann die KI das Tool nutzen, wenn der Nutzer danach fragt

8.4 Anpassung des Word-Exports

Den bestehenden Word-Export kann man leicht anpassen:

Schulname und Design aendern (ohne Code)

1. **Arbeitsbereich** → **Werkzeuge** → **Word-Export** → Zahnrad-Symbol
2. Dort kann man aendern:
 - `school_name` : z.B. "IGS Musterstadt"
 - `school_address` : z.B. "Schulstrasse 1, 12345 Musterstadt"
 - `accent_color` : z.B. "#0033CC" (Blau) oder "#CC0000" (Rot)
 - `footer_text` : z.B. "Erstellt mit KI am Tag der offenen Tuer"

Code-Aenderungen (fuer Fortgeschrittene)

Der vollstaendige Quellcode des Word-Export-Tools ist im Arbeitsbereich einsehbar (Arbeitsbereich → Werkzeuge → Word-Export → Bearbeiten). Wichtige Stellen:

- **Kopfzeile anpassen:** Suche nach `# Kopfzeile` im Code — dort wird Schulname und Adresse gesetzt
- **Fusszeile anpassen:** Suche nach `# Fusszeile` — dort wird Datum und Hinweistext gesetzt
- **Logo hinzufuegen:** Mit `python-docx` kann man Bilder einfügen: `doc.add_picture('logo.png', width=Cm(3))`
- **Seitenraender:** Die Variable `section.top_margin = Cm(2.5)` etc. steuert die Raender
- **Schriftarten:** In `add_formatted_text()` kann man `run.font.name` aendern (z.B. "Arial", "Times New Roman")

8.5 Hilfreiche Ressourcen

Ressource	Link	Beschreibung
Open WebUI Dokumentation	docs.openwebui.com	Offizielle Doku fuer Tools und Skills
python-docx Dokumentation	python-docx.readthedocs.io	Bibliothek fuer Word-Dokumente
Pydantic (Valves)	docs.pydantic.dev	Datenvalidierung fuer Einstellungen
Python asyncio	docs.python.org/3/library/asyncio.html	Asynchrone Programmierung

Tipp fuer den Informatik-Unterricht: Das Erstellen eigener Tools ist ein praxisnahes Projekt — Schueler lernen Python, API-Design, Datenverarbeitung und sehen sofort ein Ergebnis im Chat. Ein gutes Einstiegsprojekt waere z.B. ein Tool, das Vokabellisten als CSV exportiert oder mathematische Formeln visualisiert.

Dokument erstellt am 09.04.2026 fuer den KI-Workshop am Gymnasium Wildeshausen. Plattform-Admin: daniel.roeding@bechtle.com