

KANTI BADEN

Kanti Koala

Die Lern- und Studienhilfsapp für Schüler:innen der Kantonsschule
Baden

Maturitätsarbeit, Kantonsschule Baden
Schriftlicher Kommentar

Erstbetreuer: Michael Schneider
Zweitbewerterin: Julia Smits

Geschrieben von: Aryan Anand (G22b), Simon Haddon (G22b)

Datum: 11. November 2025

Abstract

Diese Maturitätsarbeit behandelt die Konzeption und prototypische Umsetzung einer webbasierten Applikation für Schüler:innen der Kantonsschule Baden mit dem Ziel, schulische Termine, Aufgaben und Lernzeiten zentral organisierbar zu machen und regelmässige, abgegrenzte Lernphasen zu unterstützen. Die Zielsetzung umfasst keine Wirksamkeitsmessung, sondern die Bereitstellung funktionaler Grundlagen.

Zur Ableitung der Anforderungen wurden eine fokussierte Literatur- und Internetrecherche (Lernmethoden, Zeit- und Pausenorganisation, Stressaspekte), Interviews mit zwei PPP-Lehrpersonen sowie eine Umfrage unter Schüler:innen eingesetzt. Die Ergebnisse strukturierten die Funktionsprioritäten (Agenda, Lernblöcke, Noten, Timer, Tipps) und die Parametrisierung des Lernzeitalgorithmus.

Die Applikation wurde mit Python (Flask), einem relationalen Datenmodell und modularer Architektur implementiert. Kernfunktionen sind: eine Agenda mit manueller Ereigniserfassung, Import von `.ics`-Stundenplänen und algorithmischer Lernblock-Generierung nach Prioritäten; Notenverwaltung; ein konfigurierbarer Pomodoro-basiertes Lerntimer-Modul; kategorisierte tägliche Tipps, sowie auch allgemeine Lerntipps (Zeit-, Pausen-, Stressmanagement, Lernmethoden); Authentifizierung mit Basis-Schutzmassnahmen (u. a. Passwort Hashing). Der Lernzeitalgorithmus verteilt definierte Lernstunden rückwärts vom Prüfungstermin unter Berücksichtigung von Tageslimits, bevorzugten Zeiten und Konfliktvermeidung.

Die Arbeit zeigt die technische Machbarkeit eines integrierten, strukturierten Ansatzes zur Lern- und Organisationsunterstützung; Aussagen zur tatsächlichen Wirkung im Nutzungskontext stehen aus und bilden den Ge-

genstand zukünftiger Evaluation (z.B. Planungsaufwand, Terminnutzung, wahrgenommene Kontrolle). Der Prototyp bildet eine erweiterbare Grundlage für nachfolgende empirische Validierung.¹

¹ChatGPT (Version GPT-5): „Überarbeite den folgenden Text, damit er sprachlich und stilistisch den Standards einer wissenschaftlichen Maturitätsarbeit entspricht. Achte auf korrekte Grammatik, präzisen Ausdruck, logische Argumentation und sachlichen Stil. Behalte den ursprünglichen Sinn und Stil des Textes bei, aber formuliere ihn wissenschaftlicher und grammatikalisch korrekt. [...]“, 06.11.2025. Antwort ganz übernommen.

Inhaltsverzeichnis

Abstract	1
1 Einleitung – unsere Vision	5
1.1 Motivation und Relevanz	5
1.2 Fragestellung und Zielsetzung	6
1.3 Aufbau und Begründung des schriftlichen Kommentars	6
2 Recherche	7
2.1 Einführung	7
2.1.1 Warum brauchen wir eine Recherche?	7
2.1.2 Was recherchieren wir?	7
2.1.3 Wie führen wir die Recherche durch	8
2.2 Literaturstudie	9
2.2.1 Ziel	9
2.2.2 Vorgehensweise	9
2.2.3 Internet-Recherche	10
2.2.4 Literatur	11
2.3 Interviews	12
2.3.1 Vorgehensweise	12
2.3.2 Interviewfragebogen & Interviewfragen	13
2.3.3 Durchführung	16
2.3.4 Transkription	17
2.3.5 Analyse	18
2.4 Umfrage	21
2.4.1 Ziel	21

2.5	Fazit der Recherche	21
3	Methodik – Programmieren der Web-Applikation	21
3.1	Erste Entscheidungen	21
3.2	Anforderungen	22
3.3	Technische Dokumentation (System- und Datenstruktur) . . .	24
3.3.1	Darstellung der Systemarchitektur	24
3.3.2	Beschreibung der Datenbank(en) und Datenstruktur .	25
3.3.3	Code-Struktur	26
3.3.4	Frontend-Struktur (Templates und Statische Dateien) .	29
3.4	Features	29
3.4.1	Datenbank	30
3.4.2	Serververbindung	33
3.4.3	Authentifizierung	34
3.4.4	Agenda	35
3.4.5	Der Lernzeitalgorithmus	36
3.4.6	Daily Tipps	40
3.4.7	Notenorganisation	40
3.4.8	Lerntimer	41
3.4.9	Lerntipps	41
3.5	Sicherheitskonzept	41
3.5.1	Datenspeicherung und Passwort-Sicherheit	41
3.5.2	Transportverschlüsselung (HTTPS)	42
3.5.3	CSRF-Schutz	42
3.6	Tests	44
4	Schlussfolgerung und Ausblick	44
	Literaturverzeichnis	48
	Abbildungsverzeichnis	52

1. Einleitung – unsere Vision¹

Der Schulalltag an der Kantonsschule Baden ist durch eine hohe Arbeitsdichte geprägt. Neben zahlreichen Prüfungen, auf die vorzubereiten ist, fallen kontinuierlich Hausaufgaben an. In Kombination mit auserschulischen Verpflichtungen entstehen Anforderungen an Planung und Selbstorganisation, die ohne strukturierende Hilfsmittel nur begrenzt überschaubar bleiben.

1.1 Motivation und Relevanz

Aus unserer Erfahrung als Schüler der Kantonsschule Baden besteht ein wiederkehrendes Problem für Schüler:innen darin, Lernaufwand, Fristen und verfügbare Zeitblöcke koordiniert abzubilden. Fehlende Übersicht kann zu subjektivem Zeitdruck, Verdichtung einzelner Tage und Verschiebungen führen. Eine zentral organisierende Applikation, welche Lernzeiten, Termine und Noten gemeinsam verwaltet, ist daher relevant. In diesem Kontext wird unter Unterstützung des Selbstmanagements ausdrücklich Folgendes verstanden: (1) Reduktion des individuellen Planungsaufwands (Zeit zur manuellen Wochenplanung), (2) Erhöhung der Vorlauftransparenz (Sichtbarkeit bevorstehender Prüfungen in Tagen), (3) strukturierte Segmentierung von Lernblöcken (klar definierte Start-/Endzeiten). Die Plattformwahl (Web oder nativ) bleibt zum Untersuchungszeitpunkt offen; die Konzeption erfolgt plattformneutral.

¹ChatGPT (Version GPT-5): „Überarbeite den folgenden Text, damit er sprachlich und stilistisch den Standards einer wissenschaftlichen Maturitätsarbeit entspricht. Achte auf korrekte Grammatik, präzisen Ausdruck, logische Argumentation und sachlichen Stil. Behalte den ursprünglichen Sinn und Stil des Textes bei, aber formuliere ihn wissenschaftlicher und grammatikalisch korrekt. [...]“, 04.11.2025. Antwort ganz übernommen.

1.2 Fragestellung und Zielsetzung

Hauptfragestellung:

Kann eine digitale Applikation so konzipiert und prototypisch umgesetzt werden, dass sie Schüler:innen eine übersichtliche Organisation von Terminen, Aufgaben und Lernzeiten ermöglicht und zugleich Funktionen bereitstellt, die regelmässige Lerngewohnheiten unterstützen?

Begriffsklärung: Unter „Organisation erleichtern“ verstehen wir eine bessere Übersicht über anstehende Termine/Aufgaben und eine Verringerung des manuellen Planungsaufwands. „Lerngewohnheiten unterstützen“ meint Funktionen, die regelmässige, zeitlich abgegrenzte Lernphasen fördern (z. B. Lernblöcke und Timer).

Zielsetzung: Entwicklung eines Prototyps, der die zur Messung dieser Indikatoren notwendigen funktionalen Strukturen (Agenda mit Prioritäten, Lernblock-Generierung, Notenerfassung, Timer) bereitstellt.

1.3 Aufbau und Begründung des schriftlichen Kommentars

Der Bericht ist entlang der Entwicklungslogik aufgebaut und führt von der Ausgangslage über die Umsetzung zur Einordnung:

1. **Recherche:** TBA
2. **Methodik - Programmierung:** Systemarchitektur, Datenstrukturen, Darstellung der Features, Sicherheitsaspekte und schliesslich noch Testing.
3. **Schlussfolgerung und Ausblick:** Bilanz zur Fragestellung, gewonnene Erkenntnisse und offene Arbeitsschritte.

Die Gliederung macht nachvollziehbar, wie aus Problemstellung und Zielsetzung konkrete Komponenten abgeleitet, implementiert und dokumentiert wurden; eine empirische Wirkungsevaluation ist nicht Gegenstand dieses Berichts.

2. Recherche

2.1 Einführung

2.1.1 Warum brauchen wir eine Recherche?

Da wir eine Web-App erstellen wollen, welche möglichst gut an die Bedürfnisse von Schüler:innen angepasst ist, durften wir uns nicht nur auf unsere eigenen Erfahrungen als Schüler verlassen, sondern mussten auch ein gewisses Mass an Recherche erledigen, damit wir wichtige Entscheidungen sinnvoll begründen konnten. Zu diesem Ende haben wir uns entschieden, uns tiefgründig mit unserem Zielpublikum - Schüler:innen der Kantonsschule Baden - auseinanderzusetzen, indem wir Interviews mit PPP-Lehrpersonen führten und eine Umfrage für Schüler:innen gestalteten.

Die Recherche stellt hier nicht das Kernstück unserer Arbeit dar, sondern ist ein unterstützender, aber dennoch sehr wichtiger, Bestandteil für die Entwicklung der Web-Applikation, da sie uns hilft, uns in unsere Zielgrupe zu versetzen und ihre Bedürfnisse Wir setzten einen starken Wert auf begründete Entscheidungen und strebten eine möglichst hohe Qualität an.

2.1.2 Was recherchieren wir?

Nun ging es zuerst einmal darum, herauszufinden was wir überhaupt recherchieren wollten. Da unsere Web-Applikation das Lernen fördern soll, stand

der Kernpunkt, nämlich das Lernen, schon von Anfang an klar.

Um genaue Recherche-Themen auszusuchen, stellten wir einige W-Fragen zum Lernen:

- Wie oder warum lernt man gut?
- Wann lernt man gut oder nicht gut?
- Was hindert das Lernen?
- Was verursacht Stress beim Lernen?

Ebenso versuchten wir zu bedenken, was für eine Rolle unsere geplanten Features für die Web-Applikation spielen und was für Themenbereiche für sie wichtig sein könnten. Beispielsweise ist die Recherche zur Zeitplanung wichtig, damit wir unsere Agenda und den Lernzeitalgorithmus sinnvoll gestalten und an den Bedürfnissen von Schüler:innen anpassen können, Informationen zu Stress beim Lernen oder vor Prüfungen und die Vorbeugung davon können uns helfen, gute Lerntipps zu gestalten. Somit entstanden unsere vier Hauptbereiche, welche die obigen drei Fragen beinhalten: Lerntechniken, das Pausen- und Zeitmanagement und das Stressmanagement.

Auch bestand die Möglichkeit, dass aus der Recherche zu diesen Themenblöcken weitere Features entstehen könnten.

2.1.3 Wie führen wir die Recherche durch

Uns war klar, dass wir, da wir uns möglichst tief in unser Zielpublikum versetzen, uns hauptsächlich auf die Meinungen von Personen in und im Umfeld unserer Zielgruppe. Somit standen persönlichere Methoden, wie Interviews und Umfragen, schon früh in Erwägung.

Wir entschieden uns also nach Rücksprache mit unserem Erstbewerter, die Recherche in drei wesentliche Teile zu teilen:

- Eine begrenzte Literaturstudie, welche uns in das Thema einführen und ein wenig vertrauter mit der Materie machen soll.

- Darauf aufbauend führen wir Interviews mit Expert:innen, um diese Materie konkret zu vertiefen und, wenn möglich, auf das Zielpublikum zu beziehen.
- Durch Feedback aus den vorherigen Schritten bereiten wir eine Umfrage für unser Zielpublikum vor, welches uns Daten aus der Sicht der Schüler:innen liefern soll.

Somit können wir anhand dieser Strategie das Wissen, welche wir in der Literaturstudie auffinden, in den Interviews vertiefen, mit unserem gewünschten Umfeld vergleichen und dann gezielt in der Umfrage mit der Praxis vergleichen. Dies sollte uns erlauben, einiges an nützlichen Informationen für unsere Web-Applikation zu erhalten und daraus wichtige Entscheidungen zu treffen.

2.2 Literaturstudie

2.2.1 Ziel

Da die Literaturstudie als Unterstützung für die anderen, für uns weitaus wichtigeren Elemente unserer Recherche gedacht war, lag nicht besonders viel Fokus darauf. Ziel war, hier ein kleines Stückchen an grundlegendem Wissen zu erreichen ohne dass wir komplett in der Materie verloren gehen und die Übersicht über was tatsächlich für unsere Arbeit nötig ist verlieren.

2.2.2 Vorgehensweise

Die Literaturstudie ist hauptsächlich aus zwei Teilen aufgebaut: Eine breitere Internet-Recherche um ein gewisses Basiswissen zu erreichen, und einen tieferen Einblick in zwei Sachtexte zum Lernen. Dieses kombinierte Basiswissen fließt dann direkt in die Vorbereitung für die Interviews hinein.

2.2.3 Internet-Recherche

Wie erwähnt, ging es hier primär um den Aufbau eines Grundsatzes an Vorwissen, welches wir an unserer bisherigen Erfahrung als Kantischüler anhängen können. Somit fokussierten wir uns nicht darauf, möglichst breite und diverse Quellen einzuholen, sondern darauf, dass wir dieses Vorwissen einigermaßen effizient aufbauen können. Daraus merkten wir, dass nicht viel nötig war, um dies zu erlangen.

Lerntechniken und -methoden

Eines der ersten interessanten Einblicke welche wir fanden, war der Unterschied zwischen den Fachbegriffen „Lernmethode“ und „Lerntechnik“.

Nämlich besteht da der wesentliche Unterschied darin, dass Lerntechniken einzelne spezifische Schritte im Lernprozess sind, während eine Lernmethode eine Kollektion von Lerntechniken darstellt und die allgemeine Lernstrategie beschreibt.(Weber, 2023) Dabei wurden wir auf ein paar wenige Lernmethoden und -techniken aufmerksam, welche angeblich das Lernen vereinfachen sollten, wie beispielsweise die Lernmethoden SQ3R(Ritschel-Gotal, 2023) oder KWL(Ritschel-Gotal, 2023), während eine Lerntechnik beispielsweise das Erschaffen von Verknüpfung zu bestehendem Wissen darstellt. Diese wurden hauptsächlich von Ritschel-Gotal (2023) übernommen.

Zeit- & Pausenmanagement

Dieses Thema war schon von Anfang an wichtig für uns, vor allem wegen unseren Agenda- und Lerntimer-features, also haben wir hauptsächlich im Bereich des Lerntimers, auch bekannt als der „Pomodoro-Timer“, und die Zeiteinplanung recherchiert.

Zur erfolgreichen Zeiteinteilung gehört für uns auch die Fähigkeit, sinnvoll Pausen zu machen, weswegen wir auch Informationen zu wann und wie man Pausen machen soll recherchierten. Dies war für uns vor allem wichtig, da wir in unserem Umfeld auch dies immer wieder als Problem beobachtet haben.

Umfragedesign

Da wir später eine Umfrage für die Schüler:innen der Kantonsschule Baden erstellen wollten, war es für uns wichtig, dass wir auch das Umfragedesign berücksichtigen. Der Hauptteil der Informationen dafür kam nicht aus dem Internet, sondern aus einem Lehrmittel(Ludwig & Hartmeier, 2019), welches wir netterweise von den PPP-Lehrpersonen Frau Suter und Herr Schmocker erhalten hatten. Da war hauptsächlich das Kapitel „**Sozialwissenschaftliche Methoden**“(S. 191 – 218)" Ludwig und Hartmeier (2019) hatte uns hier schon sehr weitergeholfen, da es uns auch in das Konzept der Beibehaltung der Befragten eingeführt hatte. Eine Umfrage darf somit nicht zu komplex sein, da sonst die Befragten zu schnell das Interesse verlieren und die Umfrage nicht abschliessen.

Auch recherchierten wir ein wenig zur Farbpsychologie für Umfragen, welche begründet was für Farben man verwenden sollte, um eine ansprechbare Umfrage zu gestalten. Hierbei bezogen wir uns hauptsächlich auf Schmitz (2017) und wählten schlussendlich ein blaues Farbschema für unsere Umfrage aufgrund dessen angeblicher fokussierender Wirkung.

2.2.4 Literatur

Für die Literaturstudie liehen wir die folgenden zwei Bücher aus, welche uns einen differenzierten Standpunkt geben sollten:

- **Effektiver Lernen für Dummies** (2. Auflage) (Ebbert, 2019) von Dr. Birgit Ebbert
- **Lernpsychologie** (6. Auflage) (Edelmann, 2000) von Walter Edelmann

Diese zwei Bücher sollten uns einen guten Überblick über das Lernen aus zwei verschiedenen Perspektiven - der direkten Anwendung mit „Effektiver Lernen für Dummies“ und der wissenschaftlichen Perspektive mit „Lernpsychologie“ - geben. Dabei führten wir laufend Notizen und integrierten diese in unser Interviewfragendossier & unsere Umfrage, welche in **Abschnitt 2.3 & 2.4** näher beschrieben werden.

In den Büchern lernten wir viel über was einen guten Lernerfolg voraussetzt. Dazu gehören unter anderem verschiedene Lerntechniken & -strategien, wie man sich erfolgreich auf eine Prüfung vorbereitet und die Bedeutung von einem guten Lernumfeld, d.h. alle Faktoren um das éernen welche dies unterstützen. Unter anderem beinhaltet dies, dass man einen guten Schlafrhythmus hat, genug sinnvolle Pausen macht, und auch psychologische Faktoren wie ein gutes Mindset. Auch ein paar wenige wissenschaftliche Begriffe waren für uns wichtig, darunter das sogenannte „Assoziationslernen“.(Edelmann, 2000). Unter dem Assoziationslernen versteht Edelmann (2000) das Lernen durch der Schaffung von Verknüpfungen zu bisher gelerntem Wissen. Somit kann das Gehirn einfacher neues Wissen aufnehmen und verarbeiten.

Insgesamt konnten wir so unser Vorwissen zu unseren für uns wichtigen Themenbereichen noch um einiges ausbauen, damit wir uns noch besser auf die Interviews vorbereiten konnten.

2.3 Interviews

2.3.1 Vorgehensweise

Wir wussten, das ein wichtiger Aspekt unserer Recherche Interviews mit Expert:innen sein würden, da sie uns vermutlich am Besten weiterhelfen könnten, da sie sich gut mit dem Thema auskennen und viel persönliche Erfahrungen mitbringen. Somit können sie auch direkt auf unsere Fragen eingehen und uns auch für die Umfrage persönlich Feedback geben.

Zu diesem Ende wählten wir zwei PPP-Lehrpersonen der Kantonsschule Baden, Frau Suter und Herr Schmocker, aus. Sie haben beide Erfahrung mit der Lernpsychologie und, dank ihrer Tätigkeit als Lehrpersonen, auch viel Kontakt mit Kantischüler:innen. Deswegen stellten sie, unserer Meinung nach, gute Interviewpartner für uns dar. Wir hatten bereits durch unserem Erstbetreuer, Herr Schneider, zwei Theorie-Dokumente aus Ludwig und Hartmeier (2019) erhalten, welche uns mit der Umfragetheorie und der Durchführung von Interviews helfen sollte.

Themenwahl

Wir wollten uns hauptsächlich auf unsere am Anfang der Recherche festgelegten vier Themenbereiche konzentrieren:

- Lernmethoden & -techniken
- Stressmanagement
- Pausenmanagement
- Zeitmanagement

Unser Ziel mit den Interviews war es damit, möglichst vieles an nützlichen Informationen zu diesen Themenbereichen und geplanten Features für unsere Web-Applikation zu lernen. Darunter gehören auch zum Beispiel Tipps für unsere Daily-Tipps- und Lerntipps-Features, aber auch gute Ansätze für unsere Agenda, den Lernzeitalgorithmus und den Lerntimer, auch als Pomodoro-Timer bekannt.

Bei all diesen Themen haben wir neben dem Vorwissen, welches wir durch die Literaturstudie erworben haben, auch schon einen persönlichen Bezug dank unserer bisherigen Schulkarriere, und können uns so auch auf unsere eigenen Erfahrungen und Unsicherheiten stützen. Als Letztes haben wir dann auch nach Feedback für unsere Umfrage eingefügt, da wir professionelles Feedback dafür einholen wollten und das Interview dafür die beste Gelegenheit ist, vor allem da die PPP-Lehrpersonen uns schon das Theorie-Dokument zum Umfragedesign(Ludwig & Hartmeier, 2019) zur Verfügung gestellt hatten.

2.3.2 Interviewfragebogen & Interviewfragen

Um sowohl uns selbst als auch die Interviewpartner auf das Interview adequat vorzubereiten, erstellten wir einen Interviewfragebogen, in dem all unsere geplanten Fragen aufgelistet sind. Die Fragen wurden nach den vier Themenblöcken geordnet und nummeriert, um eine klare Struktur zu erstellen, in der das Interview verlaufen soll.

Die jeweiligen Fragen haben wir gesammelt, indem wir uns einerseits überlegten, wo wir nach der Literaturstudie Unsicherheiten sahen oder generell

mehr wissen wollte, andererseits wo ein genauer Bezug zu den Schüler:innen der Kantonsschule Baden oder die persönlichen Erfahrungen der Interviewpartner wichtig sein könnten. Ebenfalls benutzten wir am Anfang ChatGPT um ein paar Vorschläge zu generieren, jedoch sind alle Fragen selbstständig ausgedacht und formuliert worden. (Vergleich *KI Nachweis*)

Wir achteten uns immer darauf, dass die Fragen einen guten Bezug zu unserer Web-Applikation hatten und das Potenzial hatten, relevante Informationen zu liefern.

Interviewfragen: Lernverhalten

Als Erstes überlegten wir uns theoretische Fragen zum Lernverhalten allgemein, aufgeteilt in **Lernmethoden** & **Lerntechniken**. Hier wurde wieder der Unterschied zwischen den beiden Begriffen wichtig, welcher von Ritschel-Gotal (2023) erklärt wurde.

Lernmethoden

Zum Thema Lernmethoden überlegten wir uns zwei sehr spezifische Fragen zur Effektivität von Lernmethoden zum Lernen. Mit diesen Fragen wollten wir nach spezifischen Lernmethoden nachforschen, wie beispielsweise SQ3R und KWL(Ritschel-Gotal, 2023), und die Meinung der Interviewpartner:innen dazu herausfinden. Dies da, wenn sich solche als sinnvoll herausstellen würden, diese eventuell in die Web-Applikation integriert oder, beispielsweise, speziell erklärt werden könnten. Somit konnten wir auch die erwähnten Lernmethoden aus der Internet-Recherche hineinarbeiten.

Lerntechniken

Das Segment der Lerntechniken stellten mit fünf Fragen das umfangreichste Segment des Interviewfragebogens dar.

Hier ging es uns darum, anstatt spezifische Techniken, wie beispielsweise das Assoziationslernen(Edelmann, 2000), genauer zu erforschen, was für eine individuelle Person am Besten funktionieren würde und ob es gute, allgemeine Lösungen gibt. Jenachdem, wie unterschiedlich das Lernen für verschiedene Personen sein kann, hätte dies unseren Ansatz für die Lerntipps

der Web-Applikation stark verändern können. Das grundlegende Ziel war deswegen immernoch, herauszufinden, wie man dies in die Web-Applikation integrieren kann. Beispielsweise wäre es schlau, wenn nun ganz klar eine spezifische Technik empfohlen wird, diese Technik, ähnlich wie beispielsweise die Pomodoro-Technik für das Zeitmanagement, einzubauen.

Besonders interessiert waren wir am sogenannten „Mindset“ und den persönlichen Tipps der Interviewpartner:innen, da diese uns womöglich einen guten Einblick in die Materie aufgrund ihrer eigenen Erfahrung als Lehrperson geben könnten. Natürlich sind alle Fragen auch besonders relevant für unsere Lerntipps.

Interviewfragen: Pausenmanagement

Dieser Abschnitt ist vor allem für unseres geplantes Lerntimer-Feature und unsere Agenda wichtig, hat aber auch eine Relevanz für unsere Lerntipps. Somit stellten wir Fragen, welche die Meinung der Interviewpartner:innen zu Pausen, deren Dauer und wie man gute Pausen macht, erforschen. Beispielsweise könnte

Interviewfragen: Stressmanagement

Das Stressmanagement war ein für uns durch persönliche Erfahrungen bereits sehr vertrautes Problem und eines, welches wir so gut wie möglich in unsere Lerntipps einbauen wollten. Ob und wie andere Schüler:innen oft Stress empfinden, wollten wir mit der Umfrage herausfinden, weswegen es auch hier hauptsächlich um die Erfahrungen der Lehrpersonen mit ihren Schüler:innen und um ihre Empfehlungen, wie man Stress abbauen kann, geht. Prüfungs- und Lernstress vorzubeugen sind für uns wichtige Aspekte, welche wir auch so gut wie möglich mithilfe unseres Lernzeitalgorithmus und den Lerntipps in die Web-Applikation einbauen wollten.

Interviewfragen: Zeitmanagement

Als letztes Segment mit konkreten Fragen kam das Zeitmanagement. Auch dies stellte sich für uns als ein vertrautes Problemfeld dar, vor allem wegen der

Prokrastination, etwas mit dem wir schon seit langem oftmals kämpfen. Ob dies auch andere Schüler:innen persönlich betrifft, ist hauptsächlich Thema für die Umfrage. Hier ging es uns aber im Interview hauptsächlich darum, herauszufinden wie man die Zeit ausserhalb des Stundenplans einteilen sollte, was unter anderem für unseren Agenda-Algorithmus von sehr grosser Bedeutung ist. Auch fragten wir hier sehr spezifisch zur Pomodoro-Technik nach, da wir auch die persönlichen Meinungen der Interviewpartner:innen in Bezug nehmen wollten, da sie als Lehrpersonen das Umfeld an der Kanti Baden gut kennen sollten und die Nützlichkeit davon einschätzen könnten. Dies ist sehr wichtig für unseren Lerntimer, welcher auf der Pomodoro-Technik basiert.

Umfragedesign

Ganz am Schluss wollten wir noch das Umfragedesign besprechen. Hierzu gibt es keine konkreten Fragen, sondern es ging uns darum, genaues Feedback von den Interviewpartner:innen zu unserer bisher erstellten Umfrage einzuholen, damit wir diese so verbessern können. Dabei achteten wir uns vor allem auch auf das Umfrage-Layout und die Art der Fragen, welche wir in der Umfrage einbauten, da unser Ziel war, eine möglichst zugängliche und, für die Befragten, möglichst unkomplizierte Umfrage zu erstellen. Beispielsweise wollten wir wissen, ob unsere Wahl eines bestimmten Fragentyps, wie beispielsweise die sogenannten Likert-Fragen, und die Gliederung der Themen sinnvoll war, damit die Befragten die Umfrage auch abschliessen. Hier stützten wir uns sehr stark auf das erlernte Vorwissen aus Ludwig und Hartmeier (2019).

2.3.3 Durchführung

Wir wussten schon früh, wer wir als unsere Interviewpartner:innen haben wollten. Wir haben uns zwei PPP-Lehrpersonen der Kantonsschule Baden ausgesucht, nämlich Herr Schmocker und Frau Suter, von welchen wir auch durch unseren Erstbewerter Theoriedokumente zum Umfrage- und Interviewdesign erhalten haben (Ludwig & Hartmeier, 2019).

Somit luden wir diese zwei Lehrpersonen per E-Mail zum Interview ein, vereinbarten ein Datum und schickten ihnen jeweils etwa eine Woche vor dem

Interview den fertiggestellten Interviewfragebogen und einen Link zu unserer Umfrage, damit sie sich gut vorbereiten konnten. Die vereinbarten Daten waren der 24. April 2025 mit Frau Suter und der 8. Mai 2025 mit Herr Schmocker.

Die Interviews führten wir in einem reservierten Klassenzimmer an der Kantonsschule Baden, jeweils am Mittag um 12:15 durch, mit einer Dauer von je etwa einer Stunde. Diese wurden mit der Erlaubnis der Interviewpartner aufgenommen, damit sie später besser transkribiert und analysiert werden können und schrieben nebenbei reichliche Notizen. Ein Fehler, welcher uns hier unterlief, war, dass wir die Interviews auf Mundart führten, welches diese spätere Analyse um einiges erschwerte.

Nach dem Interview mit Frau Suter tauschten wir uns noch per E-Mail aus, um weiteres Feedback für die Umfrage, welche nach dem Feedback aus den Interviews ergänzt wurde, zu gewinnen.

2.3.4 Transkription

Als erster Schritt der formellen Analyse der Interviews transkribierten wir die jeweiligen Audio-Aufnahmen der Interviews auf Papier als Word-Dateien, um die spätere detaillierte Analyse zu erleichtern. Dies war, wie in der *Durchführung* erwähnt, aufgrund der Durchführung auf Mundart nicht sehr einfach, da dies aufgrund des Mangels an Mundart-Übersetzern nun komplett manuell vorlaufen musste.

Somit wurde das Gesprochene auf Hochdeutsch übersetzt und allfällige sprachliche Füller wie beispielsweise „ähm“ wurden entfernt. Wir achteten uns immer genau darauf, dass klar ist wer wann spricht. Zu diesem Ende wurden Textabschnitte geformt, indem alles, was eine Person zu einem Zeitpunkt ununterbrochen sagt, zusammengefügt wurde. Dies sah dann beispielsweise folgendermassen aus:

[3:27] **Frau Suter:** Von den effektiven Zeiten bin ich ein wenig überfragt. Ich kann mir vorstellen, dass es auch wieder draufankommt, um was es nun genau geht.

Die Zeitangabe signifiert, wann der gesprochene Abschnitt anfängt, damit man ihn zur Überprüfung leicht wiederfinden kann und die Person, wer die Aussage getroffen hat. Ebenso haben wir uns als Hilfe notiert, wann in etwa welche Frage / welches Thema diskutiert wird.

So arbeiteten wir uns durch beide Interviews durch und transkribierten sie vollständig. Die Transkriptionen sind im Anhang zu finden.

2.3.5 Analyse

Vorgehensweise

In der Analyse ging es uns um dreierlei: Direkte Antworten und Aussagen zu unseren Fragen, allgemeine nützliche Informationen zum Lernen und allgemeine nützliche Informationen zur Web-Applikation. Diese sammelten wir durch genaue Analyse der schriftlichen Transkription, dabei wurde immer für spätere Referenz auch auf die genaue Textquelle vermerkt. Auch wurden die jeweiligen Antworten von Frau Suter und Herr Schmocker zu den gegebenen Fragen verglichen.

Unser Ziel mit der Analyse bestand darin, möglichst viele konkrete Empfehlungen für unsere Web-Applikation zu erhalten, welche aus diesen nützlichen Informationen zusammengefasst wurden und nach den Features, für welche sie nützlich sind, geordnet wurden.

Das Interviewanalyse-Dokument samt allen Erkenntnissen ist im Anhang vorzufinden.

Ergebnisse

Zuerst fassen wir hier die wichtigsten Erkenntnisse zu unseren festgelegten Themenbereichen stark zusammen:

Lernverhalten

Bei den Lernmethoden und -techniken bestand die Hauptaussage der Interviewpartner:innen darin, dass es keine perfekte, allgemeingültige Methode oder Technik gibt, um das Lernen einfacher zu machen. Es geht darum, dass

das Gehirn gut aktiviert wird und auch im Idealfall mit mehreren „Sinnen“ lernt. Konzepte wie die „Lerntypen“ sind eigentlich veraltet und nicht wissenschaftlich belegt.

„Geheimtipps“ der Lehrpersonen empfehlen beispielsweise gute Reflektion über den Lernprozess selbst und dass, grundsätzlich, früheres Lernen besser ist. Auch hat die eigene Einstellung, also das „Mindset“, der Lernenden einen grossen Einfluss auf den Lernerfolg hat.

Dies bedeutete für unsere Web-Applikation, dass wir fürs Lernen selbst keine spezifischen Lernmethoden einbauen können ausser das der Lernzeitalgorithmus frühes und wiederholtes Lernen fördern sollte, jedoch haben wir reichlich Lerntipps daraus gewonnen.

Pausenmanagement

Bei den Pausen engagierte Frau Suter, dass man vor allem auf seinen Körper hören sollte. Das heisst, wenn man merkt, dass man zu müde wird um selbst einfache Aufgaben zu lösen, sollte man spätestens eine Pause machen. Herr Schmocker hingegen empfahl, dass fix regulierte Pausen, wie beispielsweise bei der Pomodoro-Technik, besser seien.

Beide aber betonen auch, dass Disziplin und Selbstregulation hier eine grosse Rolle spielen und gaben genauere Empfehlungen für wie lange Pausen sein sollten.

Diese Erkenntnisse helfen auch wieder hauptsächlich bei unseren Lerntipps, aber auch mit dem Lerntimer-Feature.

Stressmanagement

Hier wurde von beiden Interviewpartner:innen betont, dass Stress vor allem durch gute Planung verhindert werden kann, und auch genau der Mangel dessen oftmals hier an der Kantonsschule Baden zu diesem Stress führt, welches den Nutzen für unsere Agenda und den Lernzeitalgorithmus unterstützt. Sie betonten aber auch, dass es auch an den Lehrpersonen liegen kann, vor allem wenn diese Rücksichtslos Prüfungen und Termine einplanen, weswegen auch sie eine grosse Verantwortung tragen. Auch wurden, in Verbindung mit dem Pausenmanagement, gute Aktivitäten zum Stressabbau empfohlen, wie

beispielsweise Sport.

Somit haben wir auch hier reichlich Material für unsere Daily- und Lerntipps gewonnen, aber auch wichtige Validation für unsere Agenda. Der Lernzeitalgorithmus sollte so auch durch gute Planung zum Stressabbau beitragen können.

Zeitmanagement

Beide Interviewpartner:innen befanden die Pomodoro-Technik als eine gute Methode um die Disziplin beim Lernen zu fördern, sie brauchen die Technik auch oftmals selbst. Auch verstärkten sie den Nutzen von guter Planung mit Wochenplänen und Agenden, damit die verfügbare Zeit gut benutzt wird und eine gute Balance zwischen Lernen und Freizeit etabliert werden kann. Herr Schmocker beispielsweise empfiehlt, dass man mindestens einen arbeitsfreien Tag in der Woche haben sollte, egal wie beschäftigt man sonst ist.

Auch dieser Themenblock unterstützt so unsere Agenda und den Lerntimer und liefert uns einige gute Vorschläge, wie wir diese Features verbessern können.

Konkrete Empfehlungen für die Web-Applikation

Nun wurden diese Ergebnisse zu konkreten Empfehlungen für unsere Web-Applikation zusammengefasst, also wie genau gewisse Features gestaltet und ergänzt werden sollten. Diese Empfehlungen drehen sich hauptsächlich um die Lerntipps, die Agenda und dem Lernzeitalgorithmus, dem Lerntimer und ein paar extra Features.

Daily- und Lerntipps

Hier geht es hauptsächlich darum, dass Erkenntnisse welche sich sonst nicht direkt in die Web-Applikation einbauen lassen, den Endnutzer:innen vermittelt werden können. Davon haben wir einiges aus allen unserer Themenbereichen gesammelt, grundsätzlich alles, in dem es um die persönliche Einstellung oder Handlungen geht.

Empfohlene Tipps sind beispielsweise:

- Multisensorisches Lernen, da dies verschiedene Bereiche des Gehirns aktiviert, was zu effizienterem Lernen führt.
- Kurze Erklärungen bekannter Lernmethoden, wie die bereits erwähnten SQ3R- und KWL-Methoden(Ritschel-Gotal, 2023).

2.4 Umfrage

2.4.1 Ziel

Das wesentliche Ziel unserer Umfrage bestand darin, unsere Web-Applikation und unseren Features aus unserer Zielgruppe, welche ja dann auch die Web-Applikation benutzen sollten, so gut wie möglich zu validieren. Dazu gehört, zum Beispiel, herauszufinden ob unsere Annahmen über die Bedürfnisse der Schüler:innen tatsächlich stimmen, beispielsweise im Bereich Hier fließen die Erkenntnisse der Literaturstudie und der Interviews hinein, damit wir unsere gewählten Themenbereiche möglichst genau abdecken können.

2.5 Fazit der Recherche

3. Methodik – Programmieren der Web-Applikation

3.1 Erste Entscheidungen¹

¹ChatGPT (Version GPT-5): „Überarbeite den folgenden Text, damit er sprachlich und stilistisch den Standards einer wissenschaftlichen Maturitätsarbeit entspricht. Achte auf kor-

Zu Beginn stand die grundsätzliche Plattformwahl im Zentrum: native Applikation (z. B. für Smartphones) oder webbasierte Lösung. Unter Berücksichtigung von Geräteunabhängigkeit, Verteil- und Updateaufwand, Entwicklungsressourcen sowie des verfügbaren Zeitrahmens erwies sich eine Web-Applikation als zweckmässig. Sie ist plattformagnostisch im Browser nutzbar, benötigt keine Installation und lässt sich zentral aktualisieren. Zudem reduziert eine einheitliche Codebasis den Implementierungs- und Wartungsaufwand gegenüber mehreren nativen Anwendungen für unterschiedliche Betriebssysteme.

Auf Basis dieser Entscheidung fiel die Wahl der Technologie auf Python mit dem Microframework Flask. Ausschlaggebend waren vorhandene Vorkenntnisse. Als Entwicklungsumgebung wurde Visual Studio Code verwendet, da es durch integrierte Funktionen wie Code-Assistenz (z. B. Copilot) einen effizienten Entwicklungsprozess ermöglicht und schnelle Unterstützung bietet.

Für die kollaborative Arbeit kamen Git als Versionsverwaltung und GitHub als zentrales Remote-Repository zum Einsatz. Der Quellcode wurde dort gemeinsam versioniert und ausgetauscht; regelmässige Synchronisationen (Push/Pull) stellten einen konsistenten, aktuellen Projektstand sicher.

3.2 Anforderungen²

Bevor wir mit dem Programmieren der Web-Applikation beginnen konnten, mussten wir uns zuerst über die Anforderungen an die Applikation klar werden. Da es sich um eine Web-Applikation handelt, welche den Schüler:innen der Kantonsschule Baden helfen soll, mussten wir uns überlegen, welche Funktionen die Applikation beinhalten sollte und wie diese umgesetzt werden könnten.

Die Anforderungen an die Kanti Koala Web-Applikation sind wie folgt:

rekte Grammatik, präzisen Ausdruck, logische Argumentation und sachlichen Stil. Behalte den ursprünglichen Sinn und Stil des Textes bei, aber formuliere ihn wissenschaftlicher und grammatikalisch korrekt. [...]. “, 04.11.2025. Antwort ganz übernommen.

²ChatGPT (Version GPT-5): „Überarbeite den folgenden Text, damit er sprachlich und stilistisch den Standards einer wissenschaftlichen Maturitätsarbeit entspricht. Achte auf korrekte Grammatik, präzisen Ausdruck, logische Argumentation und sachlichen Stil. Behalte den ursprünglichen Sinn und Stil des Textes bei, aber formuliere ihn wissenschaftlicher und grammatikalisch korrekt. [...]. “, 04.11.2025. Antwort ganz übernommen.

- **Home-Screen:** Von dem Home-Screen sollte man auf seinen Account und die Agenda zugreifen können. Zusätzlich sollte hier jeden Tag ein allgemeiner Tipp für die Kantonsschule angezeigt werden.
- **Account Management:** Die Nutzer:innen sollten sich registrieren, einloggen, ihr Passwort zurücksetzen und ihre Account-Einstellungen ändern können. Sie sollten die Möglichkeit haben, ihr Passwort zu ändern und allfälligerweise ihr Account zu löschen.
- **Agenda:** Die Nutzer:innen sollten ihren Stundenplan eintragen können, sowohl manuell wie auch durch den Import einer .ics-Datei. Ebenso sollte man neue Ereignisse eintragen können. Die Ereignisse sollten veränderbar sein. Die Farbe der Ereignisse sollten auch frei bestimmbar sein. Die Agenda sollte auch einen Lernzeitalgorithmus beinhalten, welcher automatisch Lernzeiten basierend auf den eingetragenen Ereignissen und den Prioritätseinstellungen der Nutzer:innen plant.
- **Notenverwaltung:** Die Nutzer:innen sollten ihre Noten für jedes Fach eintragen können. Die Noten sollten veränderbar und löschar sein. Die Nutzer:innen sollten auch ihre Semester verwalten können, indem sie neue Semester hinzufügen, bestehende Semester bearbeiten und löschen können.
- **Lerntimer:** Die Nutzer:innen sollten einen Pomodoro-Timer verwenden können, um ihre Lernzeiten zu strukturieren. Der Timer sollte anpassbar sein, sodass die Nutzer:innen die Länge der Lern- und Pausenintervalle einstellen können.
- **UI:** Die Web-Applikation sollte ein benutzerfreundliches und ansprechendes UI haben, welches einfach zu navigieren ist.

3.3 Technische Dokumentation (System- und Datenstruktur)³

Dieser Abschnitt beschreibt die technische Grundlage der Kanti Koala Web-Applikation, einschliesslich der Systemarchitektur, der Datenstruktur und der Code-Struktur, wie es auch implementiert wurde.

3.3.1 Darstellung der Systemarchitektur

Die Kanti Koala App ist als monolithische Webanwendung konzipiert, die auf einem zentralen Backend-Server läuft.

- **Frameworks:** Das Kernstück der Anwendung ist das Python-Microframework Flask. Es steuert das Routing (die Zuordnung von URLs zu Funktionen), verarbeitet HTTP-Anfragen (GET, POST, usw.) und rendert die HTML-Templates für den Benutzer.
- **Komponentenübersicht (Wichtige Pakete):**
 - **Flask-SQLAlchemy:** Dient als Object-Relational Mapper (ORM) für die Datenbank. Es ermöglicht die Definition von Datenbanktabellen als Python-Klassen (Models) und vereinfacht Datenbankabfragen.
 - **Flask-Bcrypt:** Wird für die Sicherheit der Benutzerpasswörter eingesetzt. Es hasht und verifiziert Passwörter mithilfe des bcrypt-Algorithmus.
 - **Flask-Migrate:** Erleichtert Schema-Migrationen der Datenbank, wenn sich die Modelle (Tabellenstruktur) ändern.
 - **Resend:** Dient als E-Mail-API für den Versand von systemgenerierten E-Mails, insbesondere für die „Passwort vergessen“-Funktion.

³ChatGPT (Version GPT-5): „Überarbeite den folgenden Text, damit er sprachlich und stilistisch den Standards einer wissenschaftlichen Maturitätsarbeit entspricht. Achte auf korrekte Grammatik, präzisen Ausdruck, logische Argumentation und sachlichen Stil. Behalte den ursprünglichen Sinn und Stil des Textes bei, aber formuliere ihn wissenschaftlicher und grammatikalisch korrekt. [...]“, 04.11.2025. Antwort ganz übernommen.

- **icalendar**: Eine Python-Bibliothek, die zum Parsen und Importieren von `.ics`-Kalenderdateien verwendet wird, um den Schulnetz-Stundenplan zu importieren.
- **itsdangerous**: Wird verwendet, um sichere, zeitlich begrenzte Tokens zu generieren, die für die „Passwort zurücksetzen“-Links benötigt werden.
- **Server-Setup:**
 - Die Anwendung ist für den Betrieb auf DigitalOcean, einer Cloud-Plattform, ausgelegt.
 - Die Datenbankkonfiguration und verschiedene Schlüssel/API Keys werden dynamisch über Umgebungsvariablen geladen.
 - Der Code unterstützt sowohl PostgreSQL (für die Produktion auf Cloud-Diensten) als auch SQLite (für wenn wir lokal entwickeln).

3.3.2 Beschreibung der Datenbank(en) und Datenstruktur

Die Datenstruktur ist in der Datei `models.py` durch SQLAlchemy-Modelle definiert. Eine detaillierte Beschreibung der einzelnen Tabellen (`User`, `Settings`, `PrioritySetting`, `Event`, `Semester`, `Subject`, `Grade`) und ihrer Attribute ist im Abschnitt „Datenbankmodelle und Schema“ ausführlicher dokumentiert.

- **Tabellen:** Die Datenbank besteht aus sieben Haupttabellen: `User`, `Settings`, `PrioritySetting`, `Event`, `Semester`, `Subject` und `Grade`, die direkt den SQLAlchemy-Klassen in `models.py` entsprechen.
- **Beziehungen (Relationships):** Die Beziehungen werden durch `db.relationship` und `db.ForeignKey` in den Modellen verwaltet.
 - **User-zentrierte Struktur:** Der `User` ist das zentrale Modell. Alle anderen Hauptdaten sind direkt oder indirekt mit ihm verknüpft:
 - * `User (1) → Settings (1)`

- * **User** (1) → **Event** (N)
- * **User** (1) → **Semester** (N)
- **Hierarchische Beziehungen:**
 - * **Settings** (1) → **PrioritySetting** (N): Jede Einstellung hat mehrere Prioritätsregeln.
 - * **Semester** (1) → **Subject** (N): Jedes Semester hat mehrere Fächer.
 - * **Subject** (1) → **Grade** (N): Jedes Fach hat mehrere Noten.
- **Kaskadierendes Löschen:** Die Beziehungen sind mit `cascade="all, delete-orphan"` konfiguriert. Das bedeutet, wenn ein übergeordnetes Objekt (z.B. ein **User** oder ein **Semester**) gelöscht wird, werden alle zugehörigen untergeordneten Objekte (z.B. alle **Events** und **Settings** des Users) automatisch mitgelöscht. Dies stellt die Datenintegrität sicher.

3.3.3 Code-Struktur

Um die Wartbarkeit und Skalierbarkeit der Anwendung zu verbessern, wurde die ursprüngliche Code-Struktur von einer einzigen `app.py`-Datei in ein modulares Python-Paket namens `kkoala` umstrukturiert. Dieser Ansatz folgt dem „Application Factory“-Pattern, einer bewährten Methode für Flask-Anwendungen. Zudem ist diese Anwendung auch die Best Practice für Flask-Anwendungen. (Muneeb, 2025)

1. **Das „Application Factory“-Pattern (`kkoala/__init__.py`):** Das Herzstück des Pakets ist die `create_app`-Funktion. Anstatt einer globalen App-Instanz wird die Anwendung durch diesen „Factory“-Aufruf erzeugt. Dies ermöglicht es, verschiedene Konfigurationen (z.B. für Entwicklung, Test oder Produktion) dynamisch zu laden und macht die Anwendung robuster. In dieser Datei werden auch die Flask-Erweiterungen initialisiert und die Blueprints registriert.

2. **Konfiguration (`kkoala/config.py`):** Diese Datei enthält Konfigurationsklassen (z.B. `DevConfig`, `ProdConfig`). Sie verwalten wichtige Einstellungen wie den `SECRET_KEY`, die Datenbank-URL und API-Schlüssel. Die Konfiguration wird je nach Umgebungsvariable beim Start der App ausgewählt.
3. **Blueprints für Routen (`kkoala/routes/`):** Die Routen der Anwendung sind in „Blueprints“ aufgeteilt, die eine Gruppierung von zusammengehörigen Endpunkten ermöglichen. Dies sorgt für eine saubere Trennung der Anwendungslogik:
 - **`auth.py`:** Enthält alle Routen für die Benutzerauthentifizierung (Login, Registrierung, Passwort zurücksetzen).
 - **`events.py`:** Verwaltet die API-Endpunkte für die Agenda, einschliesslich des Erstellens, Bearbeitens und Löschens von Kalendereinträgen sowie den Start des Lernalgorithmus.
 - **`grades.py`:** Beinhaltet die API für das Notenmanagement.
 - **`main.py`:** Definiert die Hauptrouten der Webseite, wie die Startseite.
 - **`settings.py`:** Steuert die Einstellungsseite und die zugehörige Speicherlogik.
4. **Datenbankmodelle (`kkoala/models.py`):** Alle SQLAlchemy-Datenbankmodelle (z.B. `User`, `Event`, `Semester`) sind zentral in dieser Datei definiert. Dies erleichtert die Verwaltung der Datenstruktur und Beziehungen.
5. **Kernlogik und Hilfsfunktionen:**
 - **`kkoala/algorithms.py`:** Eine dedizierte Datei, die ausschliesslich die komplexe Logik des Lernzeitalgorithmus (LZA) enthält.
 - **`kkoala/utills.py`:** Beinhaltet wiederverwendbare Hilfsfunktionen und sogenannte „Decorators“. Die wichtigsten sind:

- **@login_required**: Dieser Decorator wird über Routen platziert, die nur von angemeldeten Benutzern aufgerufen werden dürfen. Er prüft automatisch, ob ein Benutzer in der aktuellen Sitzung (Session) angemeldet ist. Falls nicht, wird der Benutzer zur Login-Seite umgeleitet.
- **@csrf_protect**: Schützt Formulare und API-Endpunkte vor Cross-Site Request Forgery Angriffen, wie im Abschnitt „CSRF-Schutz“ beschrieben.
- **kkoala/extensions.py**: Hier werden die Flask-Erweiterungen (SQLAlchemy, Bcrypt, usw.) initialisiert, um zirkuläre Importfehler zu vermeiden.

6. **Startpunkt (wsgi.py) und die WSGI-Schnittstelle**: Die Datei `wsgi.py` im Hauptverzeichnis ist der standardisierte Einstiegspunkt für den Webserver. Ihre einzige Aufgabe ist es, die `create_app`-Factory zu importieren und das daraus resultierende Flask-`application`-Objekt zu erstellen.

Dieses Objekt ist entscheidend, da es der WSGI (Web Server Gateway Interface) Spezifikation entspricht. WSGI ist ein Python-Standard, der als universelle Schnittstelle oder „Brücke“ zwischen dem Webserver (der Anfragen aus dem Internet empfängt) und der Webanwendung (unserem Flask-Code) dient. (Srivastav, 2022)

Für den produktiven Einsatz unserer App verwenden wir Gunicorn („Green Unicorn“), einen robusten und weit verbreiteten WSGI-HTTP-Server. Während der eingebaute Entwicklungsserver von Flask für das Testen ausreicht, ist er nicht dafür ausgelegt, eine hohe Anzahl von Anfragen zu bewältigen. Gunicorn agiert hier als leistungsfähiger „Middleman“ zwischen dem Internet und unserer Flask-Anwendung. Er kann mehrere Anfragen gleichzeitig bearbeiten, indem er mehrere „Worker“-Prozesse verwaltet, was die Leistung und Stabilität der Anwendung unter Last sicherstellt. („Launching a Flask Application with Gunicorn“, o. D.) Wenn wir Gunicorn starten, geben wir ihm den Befehl `gunicorn wsgi:application`. Er weiss dann, dass er in der Datei `wsgi.py` nach

dem `application`-Objekt suchen und dieses als Startpunkt für die Anwendung verwenden muss.

3.3.4 Frontend-Struktur (Templates und Statische Dateien)

Die Benutzeroberfläche der Kanti Koala App wird dynamisch auf dem Server generiert und als fertige HTML-Seiten an den Browser des Nutzers gesendet.

- **Templates (`kkoala/templates/`):** In diesem Verzeichnis befinden sich alle HTML-Dateien der Anwendung. Flask verwendet die Template-Engine Jinja, um diese Dateien zu verarbeiten. Jinja ermöglicht es, Python-Code direkt in HTML einzubetten, zum Beispiel um Schleifen zu erstellen (`{% for item in items %}`), bedingte Blöcke anzuzeigen (`{% if user.is_authenticated %}`) oder Variablen aus dem Backend auszugeben (`{{ user.username }}`). Dies macht die Seiten dynamisch und personalisiert. („Template Designer Documentation — Jinja Documentation“, o. D.; „Templates — Flask Documentation“, o. D.)
- **Statische Dateien (`kkoala/static/`):** Dieses Verzeichnis enthält alle statischen Assets, die vom Browser direkt geladen werden, ohne dass der Server sie verarbeiten muss. Dazu gehören:
 - **CSS-Dateien:** Für das Styling und das visuelle Design der Anwendung.
 - **Bilder:** Logos und andere grafische Elemente.

3.4 Features

Zunächst werden alle Entscheidungen über die Features erklärt. Natürlich werden die Features auch erklärt.

3.4.1 Datenbank

Da es sich um eine Webanwendung handelt, können nicht alle Informationen des Benutzers lokal gespeichert werden. Das bedeutet erstens, dass alle Informationen extern auf einem Server gespeichert werden müssen. Zweitens erfordert der Mehrbenutzerbetrieb eine nutzerspezifisch getrennte, zentrale Datenhaltung; die ausschliessliche Speicherung der Daten eines einzelnen Nutzers, wie bei lokaler Nutzung, ist nicht ausreichend. Das bedeutete für uns zwei Dinge. Wir müssen einen Weg finden, die Informationen zu speichern, und wir müssen herausfinden, wo diese Informationen gespeichert werden sollen.

Um die Informationen zu speichern, haben wir uns für SQL-Datenbanken entschieden, da diese am einfachsten mit Flask zu verwenden sind. („Define and Access the Database“, o. D.) Wenn wir lokal arbeiten, können wir SQLite verwenden, und wenn es sich um die Produktionsumgebung handelt, können wir PostgreSQL verwenden, welches viel besser ist für eine solche Umgebung („PostgreSQL vs SQLite: The Ultimate Database Showdown“, 2025).

Die Struktur dieser Tabellen ist im Folgenden dargestellt:

Datenbankmodelle und Schema⁴

Die Datenbank besteht aus sieben Hauptmodellen, die die Nutzerdaten und die Planungslogik abbilden.

User

Dieses Modell speichert die Authentifizierungsdetails und dient als zentraler Ankerpunkt für alle anderen Daten des Nutzers.

id Eindeutige ID des Nutzers (Primary Key).

username Der gewählte Benutzername (eindeutig, notwendig).

password Das gehashte Passwort (notwendig).

email Die E-Mail-Adresse des Nutzers (eindeutig, notwendig).

⁴ChatGPT (Version GPT-5): „Describe the models from models.py in LaTeX in German, without using tables. Just use text. “, 04.11.2025. Antwort ganz übernommen.

Settings

Speichert globale Einstellungen für den Lernalgorithmus, die dem **User** zugeordnet sind.

id Eindeutige ID (Primary Key).

user_id Fremdschlüssel zur **User**-Tabelle (notwendig).

learn_on_saturday Boolesche Variable, ob am Samstag gelernt werden soll (Standard: False).

learn_on_sunday Boolesche Variable, ob am Sonntag gelernt werden soll (Standard: False).

preferred_learning_time Bevorzugte Startzeit für Lernblöcke (Standard: 18:00).

study_block_color Hex-Code für die Farbe der Lernblöcke (Standard: #0000FF).

PrioritySetting

Definiert die spezifischen Parameter für jede Prioritätsstufe des Lernalgorithmus.

id Eindeutige ID (Primary Key).

settings_id Fremdschlüssel zur **Settings**-Tabelle (notwendig).

priority_level Die Prioritätsstufe (Integer, notwendig).

color Die dem Prioritätslevel zugeordnete Farbe (Hex-Code, notwendig).

max_hours_per_day Maximale Lernstunden pro Tag für diese Priorität.

total_hours_to_learn Die gesamte zu lernende Stundenanzahl für diese Priorität.

Event

Speichert Kalendereinträge des Nutzers sowie Metadaten für den Planungsalgorithmus.

id Eindeutige ID (Primary Key).

user__id Fremdschlüssel zur **User**-Tabelle (notwendig).

title Titel des Ereignisses.

start Startzeitpunkt im ISO-Format (notwendig).

end Endzeitpunkt im ISO-Format (optional).

color Farbe des Ereignisses.

priority Prioritätsstufe (Integer).

recurrence Wiederholungsregel des Ereignisses.

recurrence__id Eindeutige ID zur Gruppierung wiederkehrender Ereignisse.

all__day Boolesche Variable, ob das Ereignis ganztägig ist (Standard: False).

locked Boolesche Variable für den Algorithmus; **True** bedeutet, das Ereignis ist fixiert (Standard: True).

exam__id ID des zugehörigen Examens, falls zutreffend.

Semester, Subject und Grade

Diese Modelle bilden die akademische Hierarchie ab.

Semester Speichert akademische Abschnitte. Enthält **user__id** (Fremdschlüssel) und **name**.

Subject Speichert Fächer innerhalb eines Semesters. Enthält **semester__id** (Fremdschlüssel) und **name**.

Grade Speichert Bewertungen für Fächer. Enthält **subject__id** (Fremdschlüssel), **name**, **value**, **weight** und **counts**.

Die Daten in der Datenbank werden unten erklärt. Was wichtig bei den Datenbanken ist, ist dass die beiden Datenbanken zusammen verbunden sind, mithilfe eines Foreign Keys. Dieser Foreign Key befindet sich in der Eventdatenbank, unter dem Name „user_id“. Dieser sagt uns, welcher Event zu welchem User gehört.

Beziehungsstruktur Die Abhängigkeiten und Kaskadenlöschungen (z.B. ein gelöschter **User** löscht alle seine **Events**, **Semesters** und **Settings**) sind über Fremdschlüsselverweise in allen untergeordneten Tabellen implementiert. Die zentralen Verbindungen sind:

- **User** → **Settings** (1:1), **Events** (1:n), **Semester** (1:n)
- **Settings** → **PrioritySetting** (1:n)
- **Semester** → **Subject** (1:n)
- **Subject** → **Grade** (1:n)

3.4.2 Serververbindung⁵

Da es um eine Web-App handelt, müssen die SQL-Datenbanken irgendwo extern gespeichert werden, wo man sie jederzeit abrufen kann. Das heisst, die SQL-Datei muss auf ein Server gespeichert werden. Wir haben uns schliesslich für die Cloud-Applikation „DigitalOcean“ entschieden. Der Grund dafür war, weil wir erhalten gratis Credits mit einem Student Developer Pack. Gleichzeitig konnten wir auch die Webseite auf derselben Plattform hosten, welches den Prozess vereinfacht hat, da wir alles intern verbinden konnten.

Die Domain, die wir kauften, wurde wegen dem gleichen Grund gekauft, nämlich, dass wir mit demselben Student Developer Pack die Top-Level-Domain .app gratis erhalten haben. Die Domain heisst kantikoala.app.

⁵ChatGPT (Version GPT-5): „Korrigiere Grammatik und Rechtschreibfehler im folgenden Text. [...]“, 04.11.2025. Antwort ganz übernommen.

3.4.3 Authentifizierung⁶

Um eine App mit verschiedenen Nutzern zu haben, brauchen wir ein gutes Authentifizierungssystem. Das bedeutet, dass es eine Anmelde- und Registrierungsfunktion sowie eine Option zum Vergessen des Passworts, eine Option zum Ändern des Passworts und schliesslich auch eine Option zum Löschen des Kontos geben muss.

Natürlich können wir ein Passwort nicht im Klartext speichern, denn das wäre ein Sicherheitsrisiko und ein ethisches Risiko, weil wir als Entwickler dann die Passwörter der einzelnen Konten einsehen können. Die einfachste Lösung für dieses Problem besteht darin, das Kennwort zu hashen. Unter Passwort-Hashing versteht man die algorithmische Umwandlung eines Passworts in Chiffretext oder eine unumkehrbar verschleierte Version seiner selbst („What is Password Hashing?“, 2022). Ein anderes wichtiges Konzept ist das Salting. Salting ist die Praxis, zufällige Daten (ein „Salt“) zu einem Passwort hinzuzufügen, bevor es gehasht wird. Dies verhindert Angriffe mit vorgefertigten Tabellen (Rainbow Tables), da das gleiche Passwort mit unterschiedlichen Salts zu unterschiedlichen Hashes führt („Rainbow Table Attacks: How They Work and How to Defend Against Them“, o. D.). Glücklicherweise kümmert sich das Flask-Bcrypt-Modul um all diese Dinge für uns, sodass wir uns keine Sorgen machen müssen, wie wir das Passwort hashen und salten. („Flask-Bcrypt“, o. D.)

Sobald man sich anmeldet oder registriert, wird man auf die Startseite der Website weitergeleitet.

Sehr wichtig bei jedem Anmeldesystem ist natürlich eine Option das Passwort zurückzusetzen wenn man es vergisst. Um so ein System zu haben, ist es wichtig, dass man den Link, um das Passwort zurückzusetzen, nur einmal verwenden kann. Um dies zu erreichen, haben wir den Token, welches gebraucht wird, um zu überprüfen, dass das Passwort für den richtigen User

⁶ChatGPT (Version GPT-5): „Überarbeite den folgenden Text, damit er sprachlich und stilistisch den Standards einer wissenschaftlichen Maturitätsarbeit entspricht. Achte auf korrekte Grammatik, präzisen Ausdruck, logische Argumentation und sachlichen Stil. Behalte den ursprünglichen Sinn und Stil des Textes bei, aber formuliere ihn wissenschaftlicher und grammatikalisch korrekt. [...]“, 04.11.2025. Antwort als Basis.

zurückgesetzt wird, mit dem Hash vom alten Passworts generiert. Wir sind dank Custic (2024) auf diese Idee gekommen. Das Prinzip funktioniert, da das Passwort sich ändern wird, und somit kann man den gleichen Token nicht wieder verwenden. Zudem brauchen wir, um die E-Mail zu verschicken, eine API, damit wir sie mit unserer eigenen Domäne verschicken können. Wir haben uns für die API von Resend entschieden, da sie eine kostenlose Stufe hat, die für unser Projekt ausreicht.

Im Falle, dass ein Nutzer zum Beispiel mit einer E-Mail, die schon gebraucht wird, sich registrieren will, oder mit einem falschen Passwort sich anmelden will, bekommt der Nutzer eine Fehlermeldung angezeigt, die ihm sagt, was falsch gelaufen ist, mithilfe von Flash-Nachrichten, welche eine Funktion von Flask ist (Acsany, o. D.).

3.4.4 Agenda⁷

Die Agenda ist eine zentrale Kernfunktion der Anwendung. Sie muss intuitiv bedienbar sein und mindestens die folgenden Operationen unterstützen: das Erstellen, Bearbeiten und Löschen von Terminen.

Jedes Ereignis umfasst die Attribute Titel, Start- und Endzeit, eine optionale Wiederholungsregel, eine Farbe sowie eine Priorität, die für den Lernzeitalgorithmus relevant ist. Zusätzlich kann ein Ereignis als ganztätig gekennzeichnet werden. Die Zuordnung zum jeweiligen Konto erfolgt über die aktuell authentifizierte Sitzung; die Benutzer-ID wird beim Anlegen eines Ereignisses automatisch hinterlegt.

Für die Darstellung setzen wir die JavaScript-Bibliothek FullCalendar ein. Wir haben uns für dieses Modul entschieden, weil es das erste Ergebnis war, das bei der Suche nach einer Bibliothek mit seinen Fähigkeiten auftauchte.

Zudem unterstützt die Agenda den Import des individuellen Stundenplans aus **schulnetz**, der von der Kantonsschule zur Anzeige von Stundenplänen

⁷ChatGPT (Version GPT-5): „Überarbeite den folgenden Text, damit er sprachlich und stilistisch den Standards einer wissenschaftlichen Maturitätsarbeit entspricht. Achte auf korrekte Grammatik, präzisen Ausdruck, logische Argumentation und sachlichen Stil. Behalte den ursprünglichen Sinn und Stil des Textes bei, aber formuliere ihn wissenschaftlicher und grammatikalisch korrekt. [...]“, 04.11.2025. Antwort als Basis.

er die formalen Kriterien eines Algorithmus erfüllt: Jeder Planungsschritt ist ausführbar (existierende Funktionen), deterministisch und determiniert (gleiche Eingabedaten führen stets zur gleichen Priorisierung und Planung). Zudem ist die Anzahl der zur Erstellung des Lernplans notwendigen Schritte endlich (Finitheit), wodurch der Mechanismus garantiert terminiert und eine strukturierte Ausgabe (den Lernplan) basierend auf den Eingabedaten (Benutzereinstellungen und Prüfungstermine) liefert. („Algorithmus“, o. D.)

Eingabeparameter und Planungsziel

Der Lernzeitalgorithmus (LZA) verwendet globale Benutzereinstellungen sowie prüfungsspezifische Prioritätseinstellungen als Eingabeparameter, um eine optimale Lernplanung zu gewährleisten.

Das zentrale Planungsziel des LZA ist es, die definierten totale Lernstunden für jede Prüfung innerhalb des gültigen Lernfensters zu erreichen, während das tägliche Maximum und alle bestehenden Kalenderkonflikte des Nutzers strikt eingehalten werden.

Globale Parameter

Diese Einstellungen gelten für den gesamten Planungszeitraum:

- **Lernen am Samstag**
Definiert, ob der Algorithmus Lernblöcke an Samstagen planen darf.
- **Lernen am Sonntag**
Definiert, ob der Algorithmus Lernblöcke an Sonntagen planen darf.
- **Bevorzugte Lernzeit**
Die bevorzugte Uhrzeit am Tag, zu der die Platzierung von Lernblöcken primär angestrebt wird.

Prüfungsspezifische Parameter (Pro Prioritätstufe)

Diese Werte werden basierend auf der Priorität jeder Prüfung zugewiesen:

- **Tägliches Maximum**

Die maximale Stundenzahl, die pro Tag für Prüfungen dieser Priorität geplant werden darf.

- **Total Lernstunden**

Die gesamte Anzahl an Lernstunden, die für Prüfungen dieser Priorität absolviert werden muss.

Ablauf und Planungsstrategie

Der Algorithmus arbeitet iterativ und bearbeitet alle als Prüfung markierten Ereignisse in aufsteigender Reihenfolge ihrer Priorität. Eine niedrigere Prioritätsnummer kennzeichnet dabei eine höhere Wichtigkeit, um eine optimale Ressourcenverteilung zu gewährleisten.

1. Zyklische Neuberechnung der Anforderungen (Recycling):

- **Flexibilitätsbereinigung:** Alle vom System selbst geplanten, aber nicht gesperrten (gesperrt sind die Blöcke, die vom Nutzer bearbeitet worden sind) Lernblöcke für die aktuelle Prüfung werden gelöscht. Dies ermöglicht eine Neuplanung, falls sich die Rahmenbedingungen (z.B. neue Events, geänderte Prioritätseinstellungen) geändert haben.
- **Soll-Stunden-Ermittlung:** Die noch zu erbringende Lernzeit wird neu berechnet. Hierbei werden alle bereits absolvierten Stunden sowie Stunden, die durch manuelle oder vom Nutzer gesperrte Blöcke abgedeckt sind, von den Gesamtsollstunden abgezogen.

2. Rückwärts-Iterative Planung:

- Die Planungsstrategie ist eine Rückwärtsiteration: Sie beginnt beim Prüfungstermin und arbeitet sich tageweise, jedoch maximal drei Wochen, bis zum aktuellen Datum vor. Diese Zeitspanne wurde auf Basis unserer Recherche als optimaler Zeitraum für eine effektive Prüfungsvorbereitung festgelegt. Dieses Vorgehen stellt sicher,

dass die Lernblöcke mit höchster Dringlichkeit (die Tage, die am nächsten zur Prüfung liegen) zuerst belegt werden.

- An jedem Tag wird die maximale Lernzeit für diese spezifische Prüfung ermittelt. Diese ergibt sich aus der Differenz zwischen dem täglichen Maximum und den Stunden, die bereits an diesem Tag für die Prüfung geplant wurden. Dadurch wird das tägliche Zeitlimit zuverlässig eingehalten und eine Überlastung vermieden.

3. Platzierung und strikte Konfliktvermeidung:

- **Bevorzugter Slot:** Es wird primär versucht, einen Lernblock in der vom Nutzer festgelegten bevorzugten Lernzeit zu platzieren.
- **Konfliktprüfung:** Die Verfügbarkeit des Slots wird gegen alle Kalendereinträge des Nutzers für den aktuellen Tag geprüft. Dabei wird ein 30-minütiger Puffer vor und nach jedem bestehenden Ereignis (wie Sport oder Arzttermin) beachtet, um knappe Überlappungen und unnötigen Stress zu vermeiden.
- **Alternative Slots:** Falls die bevorzugte Zeit belegt ist, sucht eine dedizierte Funktion den grössten verfügbaren, konfliktfreien Zeitabschnitt des Tages, um die Platzierung zu maximieren.
- **Echtzeit-Aktualisierung:** Nach der erfolgreichen Generierung und Speicherung eines Lernblocks wird dieser sofort zur Liste der aktuellen Kalenderereignisse hinzugefügt. Dieser Mechanismus ist entscheidend, um sicherzustellen, dass alle unmittelbar nachfolgenden Planungsversuche am selben Tag diesen neu erstellten Block als bereits belegt berücksichtigen und somit Überlappungen ausgeschlossen sind.

4. ErgebnISRückgabe und Zusammenfassung:

- Nachdem alle Prüfungen bearbeitet wurden, gibt der Algorithmus eine detaillierte Zusammenfassung des Planungsvorgangs zurück.

- Diese Zusammenfassung informiert den Nutzer in einem Popup über die Gesamtzahl der hinzugefügten Lernblöcke und die Gesamtstunden, die erfolgreich geplant wurden.
- Zusätzlich wird für jede einzelne Prüfung der Planungsstatus (erfolgreich / nicht erfolgreich) und die geplante Stundenanzahl angezeigt.

3.4.6 Daily Tipps

Ein relativ wichtiges Feature unserer App sind die Daily Tipps. Es sollte jeden Tag ein neuer Tipp an den Nutzer gezeigt werden auf der Startseite, entweder über die Kantonsschule oder allgemeine Lerntipps. Die einfachste Möglichkeit, dies zu implementieren, ist eine simple Modulo-Rechnung.

Tipp des Tages (Index von der Liste) = (Tag des Jahres) mod (Anzahl der Tipps)

Somit wird an einem bestimmten Tag nur einen Tipp gezeigt und über das ganze Jahr sollten alle Tipps in einem Zyklus gezeigt werden (da wir sowieso weniger als 365 Tipps haben). Am ersten Tag wird der zweite Tipp der Liste gezeigt, da natürlich der Index bei 0 anfängt, aber der Tag des Jahres bei 1 anfängt. Es macht jedoch keinen Unterschied, ob der erste Tipp am ersten Tag oder der zweite Tipp am ersten Tag gezeigt wird, da es ja nur darum geht, dass jeden Tag ein anderer Tipp gezeigt wird.

3.4.7 Notenorganisation

In der Kantonsschule muss man immer wieder Prüfungen schreiben. Die Noten, die man erhält, sind dann wichtig für die Promotion in die nächste Stufe. Deswegen haben wir ein Feature, in dem man seine Noten pro Semester speichern kann. Jedes Semester hat schon die jeweiligen Fächer, die man dann in diesem Semester haben würde, geladen. Man kann natürlich immer noch Fächer löschen und hinzufügen. Man kann mit diesem Feature dann seine Durchschnitte pro Fach und Semester sehen. Ebenfalls kann man mit dem Notenrechner sehen, welche Note man in einem Fach brauchen würde,

um einen bestimmten Schnitt in diesem Fach zu haben. Die Idee von diesem Feature stammt aus der App „Pluspoints“ (Junker, 2025). Das Feature, das uns dort fehlte, war die automatische Fächeraddition pro Semester, welches wir in unserer App implementiert haben.

3.4.8 Lerntimer

Ein weiteres wichtiges Feature unserer App ist der Lerntimer. Dieser Timer basiert auf der Pomodoro-Technik, welche im Recherche Teil dieses Bericht erklärt wurde. Mit der Recherche, die wir führten, fanden wir es wichtig, ein solcher Lerntimer zu implementieren, da es eine sehr effektive Lerntechnik ist. Der Timer hat die Standard-Einstellungen von 25 Minuten Lernen und 5 Minuten Pause, welche man aber auch ändern kann, falls man das möchte.

3.4.9 Lerntipps

Ein weiteres Feature unserer App sind die Lerntipps. Diese sind in verschiedene Kategorien aufgeteilt, wie zum Beispiel Zeitmanagement, Pausenmanagement, Stressmanagement und Lernmethoden. In jeder Kategorie gibt es verschiedene Tipps, welche wir aus unserer Recherche und den Interviews gesammelt haben. Diese Tipps sollen den Nutzer:innen helfen, ihr Lernverhalten zu verbessern.

3.5 Sicherheitskonzept⁹

Neben der reinen Authentifizierung wurden weitere grundlegende Sicherheitsmassnahmen implementiert, um die Daten der Nutzer und die Integrität der Anwendung zu schützen.

3.5.1 Datenspeicherung und Passwort-Sicherheit¹⁰

⁹ChatGPT (Version GPT-5): „Korrigiere Grammatik und Rechtschreibfehler im folgenden Text. [...]“, 04.11.2025. Antwort ganz übernommen.

¹⁰ChatGPT (Version GPT-5): „Korrigiere Grammatik und Rechtschreibfehler im folgenden Text. [...]“, 04.11.2025. Antwort ganz übernommen.

Die Sicherheit der Benutzerdaten hat höchste Priorität. Wie im Abschnitt zur Authentifizierung beschrieben, werden Passwörter niemals im Klartext gespeichert. Stattdessen wird die Flask-Bcrypt-Bibliothek verwendet, um von jedem Passwort einen kryptografischen Hash zu erzeugen. Beim Login-Vorgang wird das eingegebene Passwort ebenfalls gehasht und dieser Hash wird mit dem in der Datenbank gespeicherten Hash verglichen. Da dieser Prozess unumkehrbar ist, kann selbst bei einem direkten Zugriff auf die Datenbank das ursprüngliche Passwort nicht wiederhergestellt werden.

3.5.2 Transportverschlüsselung (HTTPS)¹¹

Die gesamte Kommunikation zwischen dem Browser des Nutzers und unserem Server wird durch das HTTPS-Protokoll verschlüsselt. Dies wird durch ein SSL/TLS-Zertifikat realisiert, das auf unserem Server bei DigitalOcean installiert ist. Die Verschlüsselung stellt sicher, dass alle übertragenen Daten, von Login-Informationen über Kalendereinträge bis hin zu Noten, vor dem Abhören durch Dritte geschützt sind. Ein Angreifer in einem öffentlichen WLAN könnte beispielsweise die Daten nicht mitlesen. Der Browser zeigt dies durch ein Schlosssymbol in der Adressleiste an und garantiert so eine sichere Verbindung zur Domain `kantikoala.app`. („What is HTTPS?“, o. D.)

3.5.3 CSRF-Schutz¹²

Neben der reinen Authentifizierung ist es entscheidend, die Aktionen eines angemeldeten Benutzers abzusichern. Eine häufigste Schwachstelle in Webanwendungen ist die Cross-Site Request Forgery (CSRF) („Cross-Site Request Forgery (CSRF, XSRF)“, o. D.). Bei einem CSRF-Angriff bringt ein Angreifer den Browser eines authentifizierten Benutzers dazu, eine unerwünschte Aktion in einer Webanwendung auszuführen, bei der der Benutzer gerade angemeldet ist. Dies geschieht, ohne dass der Benutzer es merkt. So könnte ein Angreifer

¹¹ChatGPT (Version GPT-5): „Korrigiere Grammatik und Rechtschreibfehler im folgenden Text. [...]“, 04.11.2025. Antwort ganz übernommen.

¹²ChatGPT (Version GPT-5): „Korrigiere Grammatik und Rechtschreibfehler im folgenden Text. [...]“, 04.11.2025. Antwort ganz übernommen.

beispielsweise einen Benutzer dazu verleiten, auf einen bösartigen Link zu klicken, der im Hintergrund unbemerkt das Passwort des Benutzers ändert oder sein Konto löscht. (Shaji, 2022)

Um dies zu verhindern, haben wir das „Synchronizer Token Pattern“ implementiert, eine von „Cross-Site Request Forgery Prevention Cheat Sheet“ (o. D.) empfohlene Methode. Das Prinzip ist einfach, aber sehr wirksam:

1. Für jede Benutzersitzung wird ein einzigartiges, geheimes und unvorhersehbares Token generiert und auf dem Server gespeichert.
2. Dieses Token wird in alle Formulare, die eine Zustandsänderung bewirken (z.B. das Ändern von Einstellungen oder das Löschen eines Kontos), als verstecktes Feld eingebettet.
3. Wenn der Benutzer das Formular abschickt, wird das Token zusammen mit den anderen Formulardaten an den Server gesendet.
4. Der Server vergleicht das vom Client gesendete Token mit dem in der Sitzung gespeicherten Token. Stimmen die beiden nicht überein, wird die Anfrage abgelehnt.

Da ein Angreifer auf einer fremden Website dieses geheime Token nicht kennen kann, schlägt der Fälschungsversuch fehl.

In unserer Flask-Anwendung haben wir diese Logik mithilfe eines eigenen Decorators (`@csrf_protect`) umgesetzt. Dieser Decorator wird auf alle Routen angewendet, die Daten durch POST-, PUT- oder DELETE-Anfragen ändern. Bei Standard-HTML-Formularen wird das Token als verstecktes `<input>`-Feld übergeben. Für unsere dynamischen Agenda-Funktionen, die auf AJAX basieren, wird das Token aus einem Meta-Tag ausgelesen und in einem benutzerdefinierten HTTP-Header (`X-CSRF-Token`) mit jeder Anfrage gesendet. Dies stellt sicher, dass jede datenverändernde Aktion, die in unserer Applikation ausgeführt wird, legitim vom Benutzer und von unserer eigenen Webseite stammt.

Eine wichtige Anmerkung ist, dass es schon Module wie `Flask-WTF` gibt, die CSRF-Schutz bieten. Wir haben uns jedoch entschieden, unseren eigenen

Decorator zu schreiben, um ein tieferes Verständnis für die Funktionsweise von CSRF-Schutzmechanismen zu erlangen.

3.6 Tests

Um die Qualität und Zuverlässigkeit der Kanti Koala Web-App sicherzustellen, wurden verschiedene Testfälle erstellt. Diese Tests decken sowohl die Backend-Logik als auch die Frontend-Funktionalität ab. Die genauen Testfälle und deren Ergebnisse sind im Anhang dokumentiert.

4. Schlussfolgerung und Ausblick¹

Zusammenfassung

Ausgehend von der in der Einleitung formulierten Zielsetzung – Übersicht über Termine/Aufgaben/Lernzeiten schaffen und regelmässige, abgegrenzte Lernphasen unterstützen – wurde ein prototypischer Web-Ansatz umgesetzt. Die zuvor durch Literaturstudie (Lern-, Zeit-, Pausen-, Stressaspekte), Interviews mit PPP-Lehrpersonen und eine Schüler:innen-Umfrage gewonnenen Erkenntnisse dienten der Ableitung der funktionalen Kernanforderungen (zentrale Agenda, automatische Lernblock-Erzeugung, strukturierte Zeitsegmente, einfache Notenerfassung, Basis-Schutzmechanismen).

¹ChatGPT (Version GPT-5): „Überarbeite den folgenden Text, damit er sprachlich und stilistisch den Standards einer wissenschaftlichen Maturitätsarbeit entspricht. Achte auf korrekte Grammatik, präzisen Ausdruck, logische Argumentation und sachlichen Stil. Behalte den ursprünglichen Sinn und Stil des Textes bei, aber formuliere ihn wissenschaftlicher und grammatikalisch korrekt. [...]“, 06.11.2025. Antwort ganz übernommen.

Umgesetzt wurden:

- **Agenda** als zentrales Organisationsinstrument mit manueller Event-Erfassung, Import von .ics-Stundenplänen und Prioritätsattributen als Grundlage für Lernblöcke (reduziert Mehrfacherfassung).
- **Lernzeitalgorithmus** mit rückwärtsgerichteter, konfliktvermeidender Platzierung von Lernblöcken innerhalb täglicher Obergrenzen und gesetzter Prioritäten (unterstützt regelmässige, zeitlich definierte Phasen).
- **Notenverwaltung** (Semester-Fach-Note) zur strukturierten Ablage schulischer Leistungsdaten (fördert konsistente Dokumentation).
- **Pomodoro-basierter Lerntimer** zur zeitlichen Segmentierung von Lernintervallen und Pausen (Rahmen für wiederholbare Lerngewohnheiten).
- **Tägliche Tipps** aus recherchierten Bereichen (Zeit-, Pausen-, Stressmanagement, Lernmethoden) als ergänzende Impulse.
- **Authentifizierung** (Registrierung, Login, Passwort-Reset) mit gehashten/gesalzten Passwörtern (bcrypt) zur sicheren Zugangstrennung.
- **Basis-Sicherheitsmassnahmen** (CSRF-Token, HTTPS-Kommunikation) zur Integrität von Sitzungsaktionen und Schutz der Transportstrecke.

Damit liegt ein kohärenter Prototyp vor, der die in der Einleitung definierten organisatorischen und lerngewohnheitsbezogenen Zielkomponenten funktional realisiert; Aussagen zur tatsächlichen Wirkung im Nutzungskontext stehen aus.

Bilanz zur Fragestellung

Die Hauptfragestellung (Erleichterung von Organisation und Unterstützung regelmässiger Lerngewohnheiten) wurde technisch adressiert:

- **Organisation erleichtern:** Zentrale Erfassung von Ereignissen, algorithmische Erstellung von Lernblöcken, Prioritätssteuerung und .ics-Import reduzieren manuellen Planungsaufwand (keine Mehrfacherfassung, automatische zeitliche Segmentierung vom Algorithmus).
- **Unterstützung von Lerngewohnheiten:** Strukturierte Lernblöcke (mit Start-/Endzeiten) und konfigurierbarer Lerntimer bilden wiederkehrende Einheiten ab; Lerntipps liefern ergänzende Handlungsimpulse.

Nicht beantwortet ist die empirische Wirksamkeit (keine Nutzerstudie, keine Auswertung von Termintreue oder wahrgenommener Kontrolle). Die Plattformfrage wurde faktisch zugunsten einer Web-Variante getroffen; eine systematische Gegenüberstellung zu einer nativen Umsetzung steht aus.

Erkenntnisse aus dem Arbeitsprozess

- Früh definierte Datenbeziehungen erleichtern Erweiterungen (z.B. zusätzliche Leistungsmetriken).
- Iterative Anpassung des Lernzeitalgorithmus notwendig (Rückwärtsplanung erwies sich zur Priorisierung naher Prüfungen als praktikabel; Konfliktpuffer reduzierten Engstellen).
- Klare Prioritätsparameter (tägliches Maximum, Gesamtstunden) verhindern Überlastung, erforderten aber konsistente Validierung.
- Modularisierung (Blueprints, `extensions.py`, Factory) minimierte Kopplung und vereinfachte Tests.
- Sicherheitsthemen (CSRF, Passwort-Hashing) müssen früh integriert werden, um Nacharbeiten zu vermeiden.
- Transkription in Mundart führte zu hohem Aufwand; Standardisierung der Interviewsprache wäre effizienter.
- Heuristische Betrachtung der Bedienlogik ersetzt keine Nutzerbeobachtung.

Ausblick und offene Punkte

- **Empirische Evaluation:** Erhebung definierter Indikatoren (Planungsaufwand, Termintreue, wahrgenommene Kontrolle) durch Tagebuch- oder Log-Daten und Fragebögen.
- **Algorithmus-Erweiterung:** Adaptive Verteilung basierend auf Notenfortschritt.
- **Plattformvergleich:** Systematische Analyse einer nativen oder hybriden App (Offline-Fähigkeit, Push-Benachrichtigungen).
- **Zusätzliche Sicherheitsmassnahmen:** 2FA, Ratenbegrenzung bei Anmeldeversuchen, E-Mail-Verifizierung.
- **Zusätzliche Features:** Integration von To-Do-Listen, Fortschrittsvisualisierung, soziale Funktionen (z.B. Lerngruppen).

Fazit

Der Prototyp schafft eine funktionale Grundlage zur strukturierten Erfassung und Planung von schulischen Aktivitäten und Lernzeiten. Die technische Machbarkeit der Kernfunktionen ist gezeigt; die tatsächliche Wirkung auf Lernorganisation und Gewohnheiten bleibt Gegenstand zukünftiger Evaluation. Damit liegt eine belastbare Ausgangsbasis für eine empirische Validierungs- und Erweiterungsphase vor.

Literaturverzeichnis

- Acsany, P. (o. D.). *Add Logging and Notification Messages to Flask Web Projects*. Real Python. Verfügbar 2. November 2025 unter <https://realpython.com/flask-logging-messages/>
- Algorithmus. (o. D.). Studyflix. Verfügbar 4. November 2025 unter <https://studyflix.de/informatik/algorithmus-4244>
- Blakley, J. (2024, 15. März). *Burnout on the Rise: Global Survey Exposes Alarming Trends in Workplace Stress*. The Alert Program. Verfügbar 10. Oktober 2025 unter <https://alertprogram.com/burnout-on-the-rise/#:~:text=Burnout%20is%20defined%20by%20the,worsening%20trend%20in%20mental%20exhaustion>.
- Contributors, W. (2019, 20. September). *Pomodoro Technique*. Wikimedia Foundation. Verfügbar 10. Oktober 2025 unter https://en.wikipedia.org/wiki/Pomodoro_Technique
- Cross-Site Request Forgery (CSRF, XSRF)*. (o. D.). Rapid7. Verfügbar 6. November 2025 unter <https://www.rapid7.com/de/cybersecurity-grundlagen/cross-site-request-forgery/>
- Cross-Site Request Forgery Prevention Cheat Sheet*. (o. D.). Verfügbar 21. Oktober 2025 unter https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
- Custic, P. (2024, 16. Januar). *Yet Another Password Reset Tutorial in Flask*. Verfügbar 7. Mai 2025 unter <https://freelancefootprints.substack.com/p/yet-another-password-reset-tutorial>
- Define and Access the Database*. (o. D.). Flask Pallets Projects. Verfügbar 18. Mai 2025 unter <https://flask.palletsprojects.com/en/stable/tutorial/database/>

- Documentation*. (o. D.). Verfügbar 26. März 2025 unter <https://fullcalendar.io/docs>
- Ebbert, D. B. (2019). *Effektiver Lernen für Dummies (2. Auflage)*. WILEY-VCH Verlag GmbH & Co. KGaA.
- Edelmann, W. (2000). *Lernpsychologie (6. Auflage)*. Beltz.
- Flask-Bcrypt*. (o. D.). Read the Docs. Verfügbar 4. November 2025 unter <https://flask-bcrypt.readthedocs.io/en/1.0.1/>
- How to Deploy a Flask App and Postgres Database to Render*. (2023, 13. Januar). Pretty Printed. Verfügbar 18. Mai 2025 unter https://www.youtube.com/watch?v=IBfj_0Zf2Mo
- icalendar*. (o. D.). Verfügbar 2. April 2025 unter <https://pypi.org/project/icalendar/>
- Junker, P. (2025, 26. September). *PlusPoints - grade management* [Google Play Store Eintrag]. Verfügbar 6. November 2025 unter <https://play.google.com/store/apps/details?id=ch.fidelisfactory.pluspoints>
- Launching a Flask Application with Gunicorn*. (o. D.). CodeSignal. Verfügbar 31. Oktober 2025 unter <https://codesignal.com/learn/courses/introduction-to-flask-basics/lessons/launching-a-flask-application-with-gunicorn>
- Lerntechniken: Die 10 Erfolgreichsten Methoden!* (2017, 7. Juli). mystipendium. Verfügbar 10. Oktober 2025 unter <https://www.mystipendium.de/studium/lerntechniken>
- Lopez, M. (2025, 24. Februar). *12 Tips To Avoid Academic Burnout*. Rochester Institute of Technology. Verfügbar 10. Oktober 2025 unter <https://www.rit.edu/admissions/blog/12-tips-avoid-academic-burnout>
- Ludwig, M., & Hartmeier, G. (2019). *Forschen, aber wie? (1. Auflage)*. hep Verlag.
- Muneeb. (2025, 12. Juni). *Flask Project Structure: Best Practices with Blueprints & Application Factory Pattern*. Verfügbar 31. Oktober 2025 unter <https://muneebdev.com/flask-project-structure-best-practices/>
- PostgreSQL vs SQLite: The Ultimate Database Showdown*. (2025, 17. Februar). Astera. Verfügbar 31. Oktober 2025 unter <https://www.astera.com/knowledge-center/postgresql-vs-sqlite/>

- Rainbow Table Attacks: How They Work and How to Defend Against Them.* (o. D.). Netwrix. Verfügbar 4. November 2025 unter <https://netwrix.com/en/cybersecurity-glossary/cyber-security-attacks/rainbow-table-attack/>
- Ritschel-Gotal, A. D. (2023, 24. Mai). *Die Top 5 Lernmethoden & Top 15 Lerntechniken für berufliche Weiterbildungen.* evrlearn. Verfügbar 10. Oktober 2025 unter <https://www.evrlearn.ch/blog/2023/05/24/die-top-5-lernmethoden-top-15-lerntechniken-fur-berufliche-weiterbildungen/>
- Schmitz, C. (2017, 25. August). *The psychology of colours.* Limesurvey. Verfügbar 10. Oktober 2025 unter <https://www.limesurvey.org/blog/knowledge/colour-psychology-in-survey-design>
- Shaji, A. (2022, 14. Oktober). *CSRF Protection in Flask.* Verfügbar 21. Oktober 2025 unter <https://testdriven.io/blog/csrf-flask/>
- Srivastav, C. (2022, 5. August). *Flask Framework: WSGI Explained.* Medium. Verfügbar 31. Oktober 2025 unter <https://chaitanya-srivastav.medium.com/flask-framework-wsgi-explained-669753ca2b72>
- Template Designer Documentation — Jinja Documentation.* (o. D.). The Pallets Projects. Verfügbar 31. Oktober 2025 unter <https://jinja.palletsprojects.com/en/stable/templates/>
- Templates — Flask Documentation.* (o. D.). The Pallets Projects. Verfügbar 31. Oktober 2025 unter <https://flask.palletsprojects.com/en/stable/templating/>
- Tomazic, N. (2022, 8. November). *Heroku Alternatives for Python-based Applications.* Verfügbar 18. Mai 2025 unter <https://testdriven.io/blog/heroku-alternatives/>
- Weber, B.
bibinitperiod R. (2023, 17. September). *Lernmethoden und Lerntechniken.* Bernd Weber Team. Verfügbar 10. Oktober 2025 unter <https://www.diebegabungsspezialisten.de/2023/09/17/lernmethoden-und-lerntechniken/>
- What is HTTPS?* (o. D.). Cloudflare. Verfügbar 31. Oktober 2025 unter <https://www.cloudflare.com/learning/ssl/what-is-https/>

What is Password Hashing? (2022, 27. Juli). Stytch. Verfügbar 10. April 2025
unter <https://stytch.com/blog/what-is-password-hashing/>

Abbildungsverzeichnis

1	Logo der Kantonsschule Baden. Quelle: Wikipedia	1
3.1	Beispiel einer Agenda mit Lernblöcken der LZA. Screenshot Kanti Koala, 31.10.2025	36

Anhang

- **Code:** Der vollständige Code der Kanti Koala Web-App ist auf GitHub verfügbar unter: <https://github.com/CoderAryanAnand/lernapp>.
- **KI-Nachweis**
- **Tests**
- **Umfragen und Interviews**
- **Interviewfragebogen**
- **Interviewtranskripte**
- **Interview- und Umfrageanalysen**