

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 Umsetzung einer webbasierten Applikation für Schüler:innen der Kantonsschule Baden. Diese Applikation hat das Ziel, schulische Termine, Aufgaben und Lernzeiten zentral organisierbar zu machen und regelmässige, abgegrenzte Lernphasen zu unterstützen. Eine umfassende Wirkungskontrolle war nicht Teil der Zielsetzung; stattdessen wurde die Bedienbarkeit in einem Usability-Test geprüft. Die Ergebnisse bestätigten grundlegende Bedienbarkeit und identifizierten kleinere Optimierungspunkte.

Zur Ableitung der Anforderungen und Validation unserer Arbeit wurden eine fokussierte Literatur- und Internetrecherche, Interviews mit zwei PPP-Lehrpersonen sowie eine Umfrage unter Schüler:innen eingesetzt. Die Ergebnisse strukturierten die Funktionsprioritäten und die Parametrisierung des Lernzeitalgorithmus, welcher Lernzeiten regelbasiert verteilt.

Die Applikation wurde mit Flask, ein Python-Modul, SQL, einem relationalen Datenmodell, und modularer Architektur implementiert. Kernfunktionen sind eine Agenda mit manueller Ereigniserfassung, Import, sowie Export, von .ics-Stundenplänen und algorithmischer Lernblock-Generierung nach Prioritäten; Notenverwaltung; ein konfigurierbarer Pomodoro-basiertes Lerntimer-Modul; tägliche Tipps, sowie auch allgemeine Lerntipps; Authentifizierung mit Basis-Schutzmassnahmen, sowie Schutzmassnahmen für die ganze App.

Die Arbeit zeigt die technische Machbarkeit eines integrierten, strukturierten Ansatzes zur Lern- und Organisationsunterstützung; eine weitergehende Wirkungsevaluation (über die durchgeföhrte Usability-Prüfung hinaus) steht aus und bildet zukünftige Arbeit. Das Produkt bildet eine erweiterbare

Grundlage für nachfolgende empirische Validierung.¹

¹ChatGPT (Model GPT-5): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 06.11.2025. Antwort als Basis.

Inhaltsverzeichnis

Abstract	1
1 Einleitung – unsere Vision	6
1.1 Fragestellung und Zielsetzung	6
1.2 Motivation und Relevanz	7
1.3 Name und Logo	7
1.4 Aufbau und Begründung des schriftlichen Kommentars	8
2 Recherche	9
2.1 Einführung	9
2.1.1 Warum brauchten wir eine Recherche?	9
2.1.2 Was recherchierten wir?	10
2.1.3 Wie führten wir die Recherche durch?	10
2.2 Literaturstudie	11
2.2.1 Ziel	11
2.2.2 Vorgehensweise	12
2.2.3 Internetrecherche	12
2.2.4 Literatur	13
2.3 Interviews	15
2.3.1 Vorgehensweise	15
2.3.2 Interviewfragebogen & Interviewfragen	16
2.3.3 Durchführung	19
2.3.4 Transkription	20
2.3.5 Analyse	20
2.4 Umfrage	25

2.4.1	Ziel	25
2.4.2	Entwicklung	25
2.4.3	Durchführung	31
2.4.4	Analyse	32
2.5	Fazit der Recherche	40
3	Methodik – Programmieren der Web-Applikation	42
3.1	Technologie- und Plattformwahl	42
3.2	Anforderungsanalyse und -definition	43
3.3	Systemarchitektur und Datenmodell	45
3.3.1	Systemarchitektur und Framework-Wahl	45
3.3.2	Datenmodell und -struktur	48
3.4	Implementierung der Kernfunktionen	52
3.4.1	Authentifizierung und Account Management	52
3.4.2	Homescreen / Dashboard	53
3.4.3	Agenda und Kalenderverwaltung	56
3.4.4	Der Lernzeitalgorithmus	56
3.4.5	Notenverwaltung	60
3.4.6	Lerntools (Timer und Tipps)	61
3.4.7	Benutzerkonfiguration (Einstellungen)	63
3.4.8	Informationsseiten	64
3.4.9	To-Do Liste	67
3.5	Übergreifende Aspekte: Sicherheit und Design	68
3.5.1	Sicherheitskonzept	68
3.5.2	Design und Benutzeroberfläche	70
3.6	Testing und Usability-Evaluation	73
3.6.1	Funktionale Tests	73
3.6.2	Formativer Usability-Test	73
4	Reflexion zum Entwicklungsprozess	75
4.1	Zeitmanagement und Projektplanung	76
4.2	Technische Herausforderungen	77
4.3	Architekturentscheidungen	78

4.4	Usability-Testing	78
4.5	Sicherheitsimplementierung	78
4.6	Zusammenarbeit und Arbeitsteilung	79
4.7	Erkenntnisse für zukünftige Projekte	79
5	Schlussfolgerung und Ausblick	80
5.1	Zusammenfassung der Arbeit	80
5.2	Beantwortung der Fragestellung	81
5.3	Ausblick und weiterführende Forschung	83
5.3.1	Empirische Wirksamkeitsevaluation	83
5.3.2	Funktionale Erweiterungen	84
5.3.3	Technische Weiterentwicklungen	84
5.3.4	Skalierung und Verbreitung	85
5.3.5	Offene Probleme	85
5.4	Schlussbemerkung	85
	Literaturverzeichnis	87
	Abbildungsverzeichnis	93
	Anhang	93

1. Einleitung – unsere Vision¹

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

1.1 Fragestellung und Zielsetzung

Hauptfragestellung:

Kann eine digitale Applikation so konzipiert und 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).

¹ChatGPT (Model GPT-5): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 08.11.2025. Antwort als Basis.

Zielsetzung: Entwicklung einer Applikation, die die zur Messung dieser Indikatoren notwendigen funktionalen Strukturen (Agenda mit algorithmischer Lernblock-Generierung, Lerntimer und -tipps, etc.) bereitstellt.

1.2 Motivation und Relevanz

Aus unserer eigenen Erfahrung als Schüler:innen an der Kantonsschule Baden besteht ein wiederkehrendes Problem: Lernaufwand, Fristen und verfügbare Zeitfenster lassen sich nur schwer koordinieren. Die Recherche bestätigte diese Beobachtungen und in den Interviews und den Umfragen wurden ähnliche Herausforderungen identifiziert.

Eine zentral organisierende Anwendung, die Lernzeiten, Termine und Noten gemeinsam verwaltet, ist daher relevant. Solch eine Applikation könnte die Selbstorganisation unterstützen, damit Stress reduzieren und die Lernmotivation steigern. Die Selbstorganisation wird unterstützt, indem die Planung der Lernzeiten nicht manuell erfolgen muss, sondern durch einen Algorithmus, der die Lernzeiten priorisiert, nach Präferenz, und verteilt.

Ob das Produkt als Web-Anwendung oder als native App umgesetzt wird, blieb zum Zeitpunkt der Untersuchung offen; die Konzeption erfolgte plattformneutral.

1.3 Name und Logo

Der Name „KantiKoala“ verbindet „Kanti“ (Kurzform von Kantonsschule) mit dem Koala als sympathisches Maskottchen. Die Alliteration macht den Namen einprägsam; den Koala haben wir als Tier gerne, deswegen wurde er gewählt. Das Logo wurde bewusst schlicht gehalten, um freundlich und zugänglich zu wirken und gleichzeitig in der Kalender- und UI-Darstellung gut lesbar zu bleiben.



Abbildung 1.1: Logo der Applikation KantiKoala

1.4 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; ein Usability-Test der Bedienbarkeit und Verständlichkeit ist auch Bestandteil der Arbeit und wurde durchgeführt.

2. Recherche

2.1 Einführung¹

2.1.1 Warum brauchten wir eine Recherche?

Da wir eine Web-App erstellen wollten, die gut an die Bedürfnisse von Schüler:innen angepasst ist, durften wir uns nicht nur auf unsere eigenen Erfahrungen als Schüler verlassen. Ein gewisses Mass an Recherche war notwendig, damit wir wichtige Entscheidungen sinnvoll begründen konnten. Zu diesem Zweck 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 stellte zwar nicht das Kernstück unserer Arbeit dar, jedoch einen unterstützenden, aber dennoch sehr wichtigen, Bestandteil für die Entwicklung der Web-Applikation. Sie ermöglichte es, dass wir uns in unsere Zielgruppe versetzen und ihre Bedürfnisse verstehen konnten. Wir setzten grossen Wert auf begründete Entscheidungen und strebten eine hohe Qualität an.

¹Claude (Model Sonnet 4.5): „Korrigiere bitte diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. [...].“, 10.11.2025. Antwort als Basis.

2.1.2 Was recherchierten wir?

Zuerst mussten wir festlegen, welche Themen wir überhaupt recherchieren wollten. Da unsere Web-Applikation das Lernen fördern sollte, stand der Kernpunkt, nämlich das Lernen, schon von Anfang an fest.

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 überlegten wir, welche Rolle unsere geplanten Features für die Web-Applikation spielen und welche Themenbereiche für sie wichtig sein könnten. Beispielsweise war die Recherche zur Zeitplanung wichtig, damit wir unsere Agenda und den Lernzeitalgorithmus sinnvoll gestalten und an die Bedürfnisse von Schüler:innen anpassen konnten. Informationen zu Stress beim Lernen oder vor Prüfungen und dessen Vorbeugung könnten uns helfen, gute Lerntipps zu gestalten. Somit entstanden unsere vier Hauptbereiche, welche die obigen vier Fragen abdecken: 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 führten wir die Recherche durch?

Uns war klar, dass wir, um uns tief in unser Zielpublikum hineinzuversetzen, uns hauptsächlich auf die Meinungen von Personen in und im Umfeld unserer Zielgruppe fokussieren sollten. Somit zogen wir persönlichere Methoden, wie Interviews und Umfragen, schon früh in Betracht.

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

- Eine begrenzte Literaturstudie, welche uns in das Thema einführen und ein wenig vertrauter mit der Materie machen sollte.
- Darauf aufbauend führten 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 bereiteten wir eine Umfrage für unser Zielpublikum vor, welches uns Daten aus der Sicht der Schüler:innen liefern sollte.

Somit konnten wir mithilfe dieser Strategie das Wissen, das wir in der Literaturstudie auffanden, in den Interviews vertiefen, mit unserem gewünschten Umfeld vergleichen und dann gezielt in der Umfrage mit der Praxis abgleichen. Dies sollte es uns erlauben, zahlreiche nützliche Informationen für unsere Web-Applikation zu gewinnen und darauf basierend wichtige Entscheidungen zu treffen.

Das Ziel der Interviews und der Umfrage bestand auch wesentlich darin, konkrete Empfehlungen für unsere Web-Applikation zu erhalten. Diese konnten dann auch als „Must“-Anforderungen, also Features, welche zwingend empfehlenswert für die Implementation sind, und „Kann“-Anforderungen, welche definitiv nützlich aber nicht dringend wären, kategorisiert werden.

2.2 Literaturstudie²

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 erlangen ohne dass wir komplett in der Materie verloren gingen und die Übersicht darüber verloren, was tatsächlich für unsere Arbeit nötig war.

²Claude (Model Sonnet 4.5): „Korrigiere bitte diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. [...].“, 10.11.2025. Antwort als Basis.

2.2.2 Vorgehensweise

Die Literaturstudie besteht hauptsächlich aus zwei Teilen: Eine breitere Internetrecherche um ein gewisses Basiswissen zu erlangen, und einen vertieften Einblick in zwei Sachtexte zum Lernen. Dieses kombinierte Basiswissen floss anschliessend direkt in die Vorbereitung der Interviews hinein.

2.2.3 Internetrecherche

Wie erwähnt, ging es hier primär um den Aufbau eines Grundwissens, das wir an unserer bisherigen Erfahrung als Kantischüler aufbauen können. Somit fokussierten wir uns nicht darauf, möglichst breite und diverse Quellen einzuholen, sondern darauf, dieses Vorwissen einigermassen effizient aufzubauen können. Daraus erkannten wir, dass nicht viel nötig war, um dies zu erreichen.

Lerntechniken und -methoden

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

Der wesentliche Unterschied besteht 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, die 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 Verknüpfungen 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 der Zeiteinteilung recherchiert.

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

So lernten wir ein wenig über die allgemeine Funktionsweise des Pomodoro-Timers (Wikipedia Contributors, 2019), ebenso wie einige Tipps dazu, wie man seine Zeit aufteilt und Pausen strukturiert (Lopez, 2025).

Umfragedesign

Da wir später eine Umfrage für die Schüler:innen der Kantonsschule Baden erstellen wollten, war es für uns wichtig, auch das Umfragedesign zu berücksichtigen. Der Hauptteil der Informationen dafür kam dann aber unerwarteterweise nicht aus dem Internet, sondern aus einem Lehrmittel. Dies wird in der Literatur näher erläutert.

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

2.2.4 Literatur

Für die Literaturstudie liehen wir die folgenden zwei Bücher aus, die uns unterschiedliche Perspektiven vermitteln 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 - einer anwendungsorientierten „Effektiver

Lernen für Dummies“ und einer wissenschaftlichen „Lernpsychologie“. Während der Lektüre führten wir laufend Notizen und integrierten diese in unser Interviewfragendossier und unsere Umfrage, die in **Abschnitt 2.3 und 2.4** näher beschrieben werden.

In den Büchern lernten wir viel darüber, was einen guten Lernerfolg voraussetzt. Dazu gehören unter anderem verschiedene Lerntechniken und -strategien, wie man sich erfolgreich auf eine Prüfung vorbereitet und die Bedeutung eines guten Lernumfelds, d.h. alle Faktoren um das Lernen, 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 berücksichtigt. 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 die Schaffung von Verknüpfungen zu bisher gelerntem Wissen. Somit kann das Gehirn einfacher neues Wissen aufnehmen, abspeichern und verarbeiten.

Insgesamt konnten wir so unser Vorwissen in den für uns relevanten Themenbereichen deutlich erweitern, damit wir uns noch besser auf die Interviews vorbereiten konnten.

Umfragedesign

Wie in der Internetrecherche erwähnt, kam ein Grossteil unseres Vorwissen zum Umfragedesign aus (Ludwig & Hartmeier, 2019), was wir netterweise von den PPP-Lehrpersonen Frau Suter und Herr Schmocker erhalten hatten. Da war hauptsächlich das Kapitel „**Sozialwissenschaftliche Methoden**“ (S. 191 – 218) von Relevanz. Dieses Dokument 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.

2.3 Interviews³

2.3.1 Vorgehensweise

Wir wussten, dass Interviews mit Expert:innen ein zentraler Bestandteil unserer Recherche sein würde, da sie uns vermutlich am besten weiterhelfen könnten, weil sie sich gut mit dem Thema auskennen und viel persönliche Erfahrung mitbringen. Dadurch 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 Herrn Schmocker, aus. Sie haben beide Erfahrung mit der Lernpsychologie und, dank ihrer Tätigkeit als Lehrpersonen, auch viel Kontakt mit Kantischüler:innen. Daher stellten sie, unserer Meinung nach, gute Interviewpartner für uns dar. Wir hatten bereits durch unserem Erstbetreuer, Herrn Schneider, zwei Theorie-Dokumente aus Ludwig und Hartmeier (2019) von ihnen erhalten, die uns bei der Umfragetheorie und der Durchführung von Interviews helfen sollten.

Themenwahl

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

- Lernmethoden und -techniken
- Stressmanagement
- Pausenmanagement
- Zeitmanagement

Unser Ziel mit den Interviews war es damit, viele nützliche Informationen zu diesen Themenbereichen und zu den geplanten Features unserer Web-Applikation zu gewinnen. Darunter gehören auch beispielsweise Tipps für

³Claude (Model Sonnet 4.5): „Korrigiere bitte diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. [...].“, 10.11.2025. Antwort als Basis.

unsere Daily-Tipps- und Lerntipps-Features, aber auch gute Ansätze für unsere Agenda, den Lernzeitalgorithmus und den Lerntimer.

Bei all diesen Themen hatten wir neben dem Vorwissen, welches wir durch die Literaturstudie erworben haben, auch schon einen persönlichen Bezug dank unserer bisherigen Schulkarriere, und konnten uns so auch auf unsere eigenen Erfahrungen und Unsicherheiten stützen. Als Letztes haben wir dann auch nach Feedback für unsere Umfrage gefragt, da wir professionelles Feedback dafür einholen wollten und das Interview dafür die beste Gelegenheit ist.

2.3.2 Interviewfragebogen & Interviewfragen

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

Die jeweiligen Fragen haben wir gesammelt, indem wir uns einerseits überlegten, in welchen Bereichen nach der Literaturstudie noch Unsicherheiten bestanden oder wir generell mehr wissen wollten, andererseits wo ein genauer Bezug zu den Schüler:innen der Kantonsschule Baden oder den persönlichen Erfahrungen der Interviewpartner wichtig sein könnte. Ebenfalls benutzten wir am Anfang ChatGPT, um einige Vorschläge zu generieren, jedoch sind alle Fragen selbstständig formuliert und entwickelt worden. (Vergleich *KI Nachweis*)

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

Der gesamte Interviewfragebogen ist im Anhang vorzufinden.

Interviewfragen: Lernverhalten

Als Erstes überlegten wir uns theoretische Fragen zum Lernverhalten allgemein, aufgeteilt in **Lernmethoden** und **-techniken**. Hier wurde wieder der

Unterschied zwischen den beiden Begriffen wichtig, der von Ritschel-Gotal (2023) erklärt wurde.

Lernmethoden

Zum Thema Lernmethoden überlegten wir uns zwei sehr spezifische Fragen zur Effektivität von Lernmethoden beim 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 Internetrecherche einarbeiten.

Lerntechniken

Das Segment der Lerntechniken stellte 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. Je nachdem, wie unterschiedlich das Lernen für verschiedene Personen sein kann, hätte dies unseren Ansatz für die Lerntipps der Web-Applikation stark verändert können. Das grundlegende Ziel war deswegen immer noch, 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 Erfahrung als Lehrpersonen geben könnten. Natürlich sind alle Fragen auch besonders relevant für unsere Lerntipps.

Interviewfragen: Pausenmanagement

Dieser Abschnitt ist vor allem für unser geplantes Lerntimer-Feature und unsere Agenda wichtig, war aber auch relevant für unsere Lerntipps. Somit stellten wir Fragen, die die Meinung der Interviewpartner:innen zu Pausen, deren Dauer und dazu, wie man gute gestalten kann, erforschen.

Interviewfragen: Stressmanagement

Das Stressmanagement war ein für uns durch persönliche Erfahrungen bereits sehr vertrautes Problem und eines, das 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, weshalb 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 ist für uns wichtige Aspekte, den wir auch so gut wie möglich mithilfe unseres Lernzeitalgorithmus und der 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, ein Problem, mit dem wir selbst schon seit Langem häufig zu kämpfen haben. 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 war. Auch fragten wir hier sehr spezifisch zur Pomodoro-Technik nach, da wir auch die persönlichen Meinungen der Interviewpartner:innen einholen wollten, da sie als Lehrpersonen das Umfeld an der Kanti Baden gut kennen sollten und die Nützlichkeit dieser Methode einschätzen könnten. Diese Informationen waren zentral 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, konkretes Feedback von den Interviewpartner:innen zu unserer bisher erstellten Umfrage einzuholen, damit wir diese gezielt verbessern können. Dabei achteten wir uns vor allem auf das Umfrage-Layout und die Art der Fragen, welche wir in der Umfrage verwendet hatten, da unser Ziel war, eine für die Befragten zugängliche und unkomplizierte Umfrage zu erstellen. Beispielsweise wollten wir wissen, ob unsere Wahl eines bestimmten Fragentyps, wie etwa 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 vorhandene Wissen aus Ludwig und Hartmeier (2019).

2.3.3 Durchführung

Wir wussten schon früh, wer wir als unsere Interviewpartner:innen auswählen wollten. Dies waren, wie bereits erwähnt, Herr Schmocker und Frau Suter. Somit luden wir diese zwei Lehrpersonen per E-Mail zum Interview ein und vereinbarten ein Datum. Danach schickten wir ihnen jeweils etwa eine Woche vor dem Interview den fertiggestellten Interviewfragebogen und einen Link zu unserer Umfrage, damit sie sich gut vorbereiten konnten.

Die Interviews führten wir in einem reservierten Klassenzimmer an der Kantonsschule Baden 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 konnten und wir machten zusätzlich umfangreiche Notizen. Wir erhielten auch ihre Erlaubnis, ihre Aussagen direkt zu verwenden. Ein Fehler, der uns unterlief, war, dass wir die Interviews auf Mundart führten, welches die 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, die 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 in Word-Dateien, um die spätere detaillierte Analyse zu erleichtern. Dies war, wie im obigen Abschnitt erwähnt, aufgrund der Durchführung auf Mundart nicht sehr einfach, da dies aufgrund des Mangels an Mundart-Übersetzern vollständig manuell vorlaufen musste.

Somit wurde das Gesprochene auf Hochdeutsch übersetzt und allfällige sprachliche Füller wie beispielsweise „ähm“ wurden entfernt. Wir achteten stets genau darauf, klar zu kennzeichnen, 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 zeigt an, wann der gesprochene Abschnitt anfängt, damit man ihn zur Überprüfung leicht wiederfinden kann sowie 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 hindurch 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, wobei dabei stets für spätere Referenz die genaue Textquelle vermerkt wurde. Auch wurden die

jeweiligen Antworten von Frau Suter und Herrn Schmocker zu den gegebenen Fragen verglichen.

Unser Ziel mit der Analyse bestand darin, möglichst viele konkrete Empfehlungen für unsere Web-Applikation zu gewinnen. Diese wurden aus den nützlichen Informationen zusammengefasst und nach den Features, für die sie relevant sind, geordnet.

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

Ergebnisse

Zuerst fassten 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, dass der Lernzeit-algorithmus frühes und wiederholtes Lernen fördern sollte, jedoch gewannen wir reichlich Lerntipps daraus.

Pausenmanagement

Bei den Pausen betonte 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 sind.

Beide betonten aber auch, dass Disziplin und Selbstregulation hier eine grosse Rolle spielen und gaben genauere Empfehlungen zur Pausendauer.

Diese Erkenntnisse helfen auch hauptsächlich bei unseren Lerntipps, helfen aber auch beim Lerntimer-Feature.

Stressmanagement

Hier wurde von beiden Interviewpartner:innen betont, dass Stress vor allem durch gute Planung verhindert werden kann, und dass genau dessen Mangel oftmals hier an der Kantonsschule Baden zu diesem Stress führt, was 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 sie auch eine gewisse 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 Validierung 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 verwendeten die Technik auch oftmals selbst. Auch unterstrichen sie den Nutzen einer guten Planung mit Wochenplänen und Agenden, damit die verfügbare Zeit gut genutzt 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 solle, egal wie beschäftigt man sonst ist.

Auch dieser Themenblock unterstützte so unsere Agenda und den Lerntimer und lieferte uns einige gute Vorschläge, wie wir diese Features verbessern

konnten.

Konkrete Empfehlungen für die Web-Applikation

Darauf 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, den Lernzeitalgorithmus, den Lerntimer und ein paar zusätzliche Features.

Alle diese Empfehlungen sind im Anhang in unserem Interviewanalyse-Dokument genauer beschrieben und wurden ausschliesslich aus den Erkenntnissen der Interviews abgeleitet. Die folgende Liste ist nicht vollständig, sondern enthält nur ein paar Beispiele davon. Dort wurde auch detailliert aufgelistet, welche dieser Empfehlungen für uns als „Muss“-Anforderungen und welche als „Kann“-Anforderungen gelten, wie zuvor in der Vorgehensweise der Recherche definiert.

Daily- und Lerntipps

Hier geht es hauptsächlich darum, dass Erkenntnisse, die sich sonst nicht direkt in die Web-Applikation einbauen liessen, den Endnutzer:innen vermittelt werden konnten. Dafür haben wir einiges aus allen unseren Themenbereichen gesammelt, grundsätzlich alles, was die persönliche Einstellung oder Handlungen betrifft.

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.
- Selbstreflektion über den Lernprozess und Selbsttests sind wichtig, ebenso wie externes Feedback.
- Tipps zum „richtigen“ Mindset und Motivationssprüche.

- Tipps für eine gute Zeit- und Pausenplanung.
- Tipps zum Stressabbau, wie Sport und Entspannungstechniken.

Diese Tipps werden dann, gestützt auf den Aussagen von Frau Suter und Herr Schmocker, in vollständige Sätze formuliert und in die Daily- und Lerntipps-Features integriert.

Agenda und Lernzeitalgorithmus

Im Laufe der Interviews kamen einige gute Empfehlungen für die Agenda und den Lernzeitalgorithmus zustande, hauptsächlich betreffend wann man lernen soll und wann nicht. Diese Empfehlungen sind für uns sehr wichtig, da diese Features den Kern unserer Web-Applikation darstellen. So sollte der/die Endnutzer:in beispielsweise einstellen können, wie viel vom Wochenende vom Lernzeitalgorithmus aufgebraucht werden darf, bis wie spät am Abend gelernt werden darf, und ob es einen idealen Lernzeitpunkt gibt, der zuerst gefüllt werden sollte. Auch soll der Lernzeitalgorithmus ebenso eine Warnung geben, wenn ein Tag zu voll wird, und möglichst wiederholtes Lernen fördern.

Dies und Weiteres soll den Benutzer:innen helfen, ihre Zeit sinnvoll mithilfe unserer Agenda einzuteilen und Lernstress zu vermeiden.

Lerntimer

Der Lerntimer, der auf der Pomodoro-Technik basiert, wurde auch durch die Interviews stark validiert, was an sich schon eine gute Empfehlung darstellt. Jedoch könnten wir aufgrund diesen Erkenntnissen auch auf der Pomodoro-Technik aufbauen und das Konzept etwas weiterentwickeln. Eine Empfehlung ist beispielsweise, dass der Timer auch Lernintervalle bis um eine gewisse Uhrzeit unterstützen soll, damit längere Aufgaben wo man nicht ständig unterbrochen werden will und sich gut konzentrieren kann, durchgeführt werden können.

Miscellaneous

Es gab auch ein paar wenige Empfehlungen, welche nicht direkt in die obigen

Kategorien passen. Diese zwei Empfehlungen schlugen neue Features vor, nämlich:

- Ein **To-Do Liste** mit dem man sich Aufgaben aufschreiben und erledigen können soll.
- Eine **KI-Page**, die einen gesunden Umgang mit KI empfiehlt und auch vor den Gefahren für den Lernerfolg warnt, wenn man sich zu sehr darauf verlässt.

Da dies beides nicht sehr wichtige Empfehlungen sind, wurden sie auch als 'Kann'-Anforderungen klassifiziert und sind somit nicht so dringend wie die Muss-Anforderungen.

2.4 Umfrage⁴

2.4.1 Ziel

Das Hauptziel unserer Umfrage war die Validierung unserer Web-Applikation und ihre Features durch unsere Zielgruppe, welche ja diese dann auch später nutzen soll. Dazu gehört, zum Beispiel, herauszufinden ob unsere Annahmen über die Bedürfnisse der Schüler:innen tatsächlich stimmen, beispielsweise ob sie oftmals an Prüfungsstress leiden. Die Erkenntnisse der Literaturstudie und der Interviews flossen hierbei ein, damit wir unsere gewählten Themenbereiche vollständig abdecken zu können.

2.4.2 Entwicklung

Die Umfrage deckte im Wesentlichen dieselben Themenbereiche ab wie bei den anderen Recherche-Segmenten, jedoch mit leichten Änderungen. Hier interessierte uns hauptsächlich, ob die Probleme, welche unser Produkt lösen soll, tatsächlich in unserer Zielgruppe existieren, d.h. wir suchten eine Validierung

⁴Claude (Model Sonnet 4.5): „Korrigiere bitte diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. [...].“, 10.11.2025. Antwort als Basis.

für unsere Web-Applikation. Deshalb waren die Fragen auf das Verhalten der Befragten gerichtet, wie beispielsweise, ob sie oftmals Prüfungsstress erleben und weshalb, wie früh sie mit dem Lernen anfangen, ob sie regelmässig Pausen machen und ob sie Interesse an einer Lern-Applikation hätten.

Zuerst mussten wir jedoch wählen, mit was für einem Programm wir die Umfrage erstellen wollten. Als Host für die Umfrage wählten wir Microsoft Forms, da wir mit dem Programm bereits vertraut waren. Dies machte Kollaboration einfach, da wir unsere Microsoft-Accounts von der Schule brauchen konnten. Zudem ermöglichte Microsoft Forms die einfache Übertragung der Daten in eine verknüpfte Excel-Datei, was die spätere Analyse vereinfachte.

Somit konnten wir mit der Entwicklung der Umfrage beginnen. Wir stützten uns dabei stark auf das bisherig erwähnte Theorie-Dokument zum Umfragendesign ((Ludwig & Hartmeier, 2019)). Dieses Dokument vermittelte uns einige Konzepte, darunter, wie man die Beteiligung von den Befragten erhöht, indem die Umfrage möglichst kurz und möglichst unkompliziert gehalten wird und für eine gute Vergleichbarkeit möglichst geschlossene Fragen verwendet werden. Aber es brachte uns auch spezifische Fragentypen bei, wie die sogenannte „Likert“-Frage, bei welcher man mehrere Aussagen mit einer Wahl von, typischerweise, „Nie“ bis „Immer“ bewertet, wie oft sie vorkommen. Deswegen lagen unsere Kernziele für das Design der Umfrage folgendermassen:

- Die Umfrage ist unkompliziert, sodass die Motivation der Befragten die Umfrage auszufüllen und somit die Rücklaufquote hoch bleibt, und sodass wenig missverstanden werden kann.
- Eine klare Gruppierung der Themenbereiche.
- Keine unnötigen Fragen - die Fragen sind stets nach unseren Zielen in der Recherche gerichtet.

Grundlegende Informationen

Als Erstes mussten wir natürlich ein paar grundlegende Daten über unsere Befragten sammeln, damit wir gut Vergleiche zwischen Gruppen ziehen können. Die Umfrage ist jedoch anonym - die einzigen Informationen, welche wir hier

sammeln wollten, waren die Mittelschule, also Gymnasium, WMS & IMS, und die Stufe (1. - 4. Klasse) in der die Befragten zum Zeitpunkt des Ausfüllens der Umfrage waren. Auf diese Weise sollte später ein Vergleich möglich sein, um festzustellen, wie sich beispielsweise das Lernverhalten zwischen den verschiedenen Stufen und Mittelschulen unterscheidet.

Lernverhalten

Hier interessierte uns hauptsächlich, wann die Schüler:innen mit dem Lernen anfangen, da dies klar mit der Theorie aus der Literaturstudie und den Interviews vergleichbar ist. Obwohl die Theorie aus Literatur und Interviews besagt, dass früheres und wiederholtes Lernen zu besseren Ergebnissen führt, wussten wir aus eigener Erfahrung, dass der damit verbundene Aufwand die Versuchung, erst sehr spät zu beginnen, stark erhöht. Gleich danach stellten wir zwei einfache Fragen, wie zufrieden die Befragten mit ihrem Lernaufwand seien und was sich ändern müsste, damit sie zufriedener wären. Somit wollten wir vergleichen können, ob es eine Korrelation zwischen der Lernzufriedenheit und dem Lernzeitpunkt gibt, und was die Schüler:innen selbst denken, dass sie machen könnten bzw. bräuchten.

Stress- und Pausenmanagement

Wir haben in den Interviews zum Stressmanagement hauptsächlich einiges über Methoden, um Stress abzubauen, gelernt, und kannten die Perspektiven der Lehrpersonen hinsichtlich der häufigsten Gründe für Prüfungsstress. Hier konnten wir nun aber selbst nachfragen, ob die Schüler:innen oftmals sich vor Prüfungen gestresst fühlen und weshalb dies so sei. So erhofften wir uns herauszufinden, was die häufigsten Motivationen für diesen Stress sind, damit wir uns auch Gedanken machen konnten, ob und wie das in der Applikation adressiert werden könnte. Wir benutzten hierfür eine Likert-Frage, wo diese Aussagen mit „Nie“, „Selten“, „Manchmal“, „Oft“ und „Immer“ beantwortet werden können.

Da wir möglichst geschlossene Fragen haben wollten, gaben wir vier für uns plausible Gründe für Prüfungsstress basiert auf den Interviews und unseren

Welche der folgenden Gründe für Prüfungsstress treffen auf Sie zu?

	Nie	Selten	Manchmal	Oft	Immer
Zu wenig gelernt	<input type="radio"/>				
Nicht genug Zeit, um zu Lernen	<input type="radio"/>				
Druck von Aussen, gute Leistungen zu erzielen	<input type="radio"/>				
Interner Druck, gute Leistungen zu erzielen	<input type="radio"/>				
Weiteres (Bitte in der nächsten Frage angeben)	<input type="radio"/>				

Welche weiteren Gründe für Prüfungsstress treffen auf Sie zu? Beschreiben Sie, was Sie genau stressst.

Enter your answer

Abbildung 2.1: Darstellung einer Likert-Frage zu den Gründen für Prüfungsstress

persönlichen Erfahrungen. Likert-Fragen waren hierfür besonders nützlich, da sie trotz ihrer Geschlossenheit mehr Nuance als reine Ja/Nein-Fragen ermöglichen und sich dennoch einfach interpretieren lassen - im Gegensatz zu offenen Fragen. Aufgrund ihres Designs konnten wir auch sehr einfach mehrere Fragen in ein „Gerüst“ kombinieren. Nach Feedback von Frau Suter beschlossen wir uns jedoch auch, ein „Weiteres“-Feld einzubauen, da es wohl wahrscheinlich noch andere Gründe für Prüfungsstress geben könnte.

Da das Stressmanagement kein besonders grosses Feld darstellte, beschlossen wir uns, das Pausenmanagement auf die gleiche Seite zu packen, da Pausen als Entspannung und Stressabbau für uns eine Antithese zum Stress darstellten. So wollten wir auch untersuchen, ob und wie die Befragten bisher Pausen machten, da dies für unsere Lerntipps und unseren Lerntimer von Relevanz hätte sein können, und als Vergleich mit den theoretischen Grundlagen aus den Interviews interessant hätte sein können.

Wegen unserem Ziel, die Umfrage unkompliziert darzustellen, beschlossen wir uns, das sogenannte „Branching“ zu verwenden, damit nur die Befragten, welche die ersten Ja/Nein-Fragen zum Prüfungsstress und dem Pausenma-

gement mit „Ja“ beantworten, überhaupt die weiterführenden Fragen sehen. Die Abbildungen 2.2 und ?? zeigen, wie solches „Branching“ aussieht.

KantiKoala - Umfrage Lernverhalten
Kantonsschule Baden

Stress- & Pausenmanagement

Fühlen Sie sich vor Prüfungen oft gestresst? Ja Nein

Machen Sie während des Lernens regelmässig Pausen? Ja Nein

Wie verbringen Sie meistens Ihre Lernpausen?

Enter your answer

Back Next Page 3 of 6

Abbildung 2.2: Ansicht der Seite wenn bei „Fühlen Sie sich vor Prüfungen oft gestresst?“ mit „Nein“ beantwortet wird.

KantiKoala - Umfrage Lernverhalten
Kantonsschule Baden

Stress- & Pausenmanagement

Fühlen Sie sich vor Prüfungen oft gestresst? Ja Nein

Welche der folgenden Gründe für Prüfungsstress treffen auf Sie zu?

	Nie	Selten	Manchmal	Oft	Immer
Zu wenig gelernt	<input type="radio"/>				
Nicht genug Zeit, um zu Lernen	<input type="radio"/>				
Druck von Aussen, gute Leistungen zu erzielen	<input type="radio"/>				
Interner Druck, gute Leistungen zu erzielen	<input type="radio"/>				
Weiteres (Bitte in der nächsten Frage angeben)	<input type="radio"/>				

Abbildung 2.3: Ansicht der Seite wenn mit „Ja“ beantwortet wird.

Zeitmanagement

Dieser Themenbereich betrachteten wir als essenziell, um unsere Web-Applikation zu validieren. Wir waren persönlich sehr vertraut mit dem Problem der Prokrastination, das eines der Hauptbegründungen für den Nutzen unserer Web-App darstellt. Wir konnten aber erst hier tatsächlich herausfinden, ob und wie breit dies auf unser Zielpublikum zutrifft. Ebenso wollten wir herausfinden, ob unser Zielpublikum bereits Hilfsmittel verwendet, um das Lernen zu unterstützen, oder nicht.

Hierfür verwendeten wir ebenfalls wieder Likert-Fragen.

Musik

Ein weiteres Feature, welches uns während der Erstellung der Umfrage interessierte, war die Option, Musik in die Lernapp zu integrieren, welche man beim Lernen hören könnte, ähnlich wie diese „Lofi hip hop radio beats to relax/study to“ videos, welche man auf YouTube vorfinden kann (Lofi Girl, 2022). Die These, dass Musik störend für das Lernen ist, war eine sehr kontroverse These (Weller, o. D.). Da wir dies aber selbst gern machen, wollten wir untersuchen, ob Musik von unserem Zielpublikum selbst beim Lernen gebraucht, und ob sie als störend oder nicht störend empfunden wird.

Lern-App

Als letztes wollten wir das allgemeine Interesse an unserer Applikation und unseren geplanten Features direkt untersuchen. So wollten wir besser einschätzen, ob der Bedarf für so ein Produkt überhaupt in unserer Zielgruppe vorhanden war, was ebenfalls eine sehr wichtige Validierung für unsere Arbeit darstellt und deswegen von kritischer Bedeutung für uns war.

Auch wollten wir genauer nachfragen, indem wir mithilfe einer Multiple-Choice-Frage das Interesse zu spezifischen Features untersuchten, und dank einer „Weiteres“-Option Vorschläge für andere Features erlaubten. So erhofften wir uns, direkt zu sehen welche Vorschläge populär bzw. nicht populär sind.

Schlussendlich wollten wir für unsere Lern- und Daily-Tipps auch noch unser Zielpublikum direkt einbeziehen. Da wir ja eine Web-Applikation von

Schülern für Schüler:innen machen, machte es für uns Sinn, dass unser Zielpublikum wahrscheinlich noch gute Ideen für relevante Lerntipps haben könnte.

Fertigstellung

Nach den Interviews konnten wir, basierend auf dem Feedback unserer Interviewpartner:innen, noch einiges an Feedback integrieren. Dies umfasste beispielsweise einige Korrekturen und Empfehlungen, wie Fragen einheitlicher und effizienter formattiert werden könnten. Ebenso liessen wir ein paar wenige Kollegen die Umfrage ausfüllen um zu testen, dass die Endnutzer-Erfahrung etwa so war, wie wir sie wollten. Wir waren, schlussendlich, sehr zufrieden mit dem Ergebnis aus all dem.

Ein Link zur fertiggestellten Umfrage kann im Anhang vorgefunden werden.

2.4.3 Durchführung

Nun ging es darum, die Umfrage tatsächlich an unser Zielpublikum zu bringen. Dies erreichten wir hauptsächlich, indem wir die Umfrage mittels Sammelmail an die gesamte Schule verschicken liessen. Nach einem kurzen Hin und Her mit der Prorektorin Frau Hofmann, konnte dies dann auch eine Woche, nachdem unsere Umfrage fertiggestellt war, umgesetzt werden und die Umfrage wurde für die Schüler:innen der Kantonsschule Baden frei zugänglich. Ebenso verbreiteten wir die Umfrage selbst auf privaten Kommunikationskanälen und baten diese darum, sie ebenfalls an andere Kantischüler weiterzuschicken.

Nach vier Wochen hatten wir schlussendlich **84** Antworten eingesammelt, was wir als eine gute repräsentative Stichprobe in unserer Zielgruppe betrachteten. Danach kamen keine weiteren Antworten in die Umfrage hinein, also stellte dies unser gesamtes Datenset dar. Diese Antworten sollten uns wertvolle Einblicke für unsere Arbeit liefern.

2.4.4 Analyse

Vorgehensweise

Nachdem die Erhebungszeit erfolgreich verlaufen war, mussten wir die Umfrage analysieren. Dies erfolgte hauptsächlich über Excel und Microsoft Forms, die uns die grafische Darstellung der gesammelten Daten ermöglichten. Dies half uns, konkrete Schlüsse aus unseren Fragestellungen zu ziehen und qualitative Vergleiche anzustellen. So arbeiteten wir uns folglich Schritt für Schritt durch die jeweiligen Daten und werteten diese aus.

Nachdem wir diese graphischen Vergleiche gezogen hatten, sammelten wir diese zu Erkenntnissen zum Lernen und für die Web-Applikation zusammen und achteten dabei jeweils darauf, ob diese Ergebnisse unsere Web-Applikation validieren oder nicht. Schlussendlich formulierten wir daraus konkrete Empfehlungen, ähnlich wie bei der Interview-Analyse. Diese konkreten Empfehlungen sollten uns tiefgründig über die Bedürfnisse unseres Zielpublikums informieren. Auch hier wurde wieder in „Muss“- und „Kann“- Anforderungen aufgeteilt.

Die vollständige Analyse kann im Umfrage-Analyse-Dokument im Anhang eingesehen werden.

Ergebnisse

Nun präsentieren wir die wesentlichen Ergebnisse unserer Umfrage, analysiert wie im obigen Abschnitt beschrieben.

Generelle Informationen

Insgesamt gab es, wie schon erwähnt, **84** Antworten (Siehe Abbildung 2.4). Davon kamen 76 aus dem Gymnasium, vier aus der IMS und vier aus der WMS. Aufgrund dieser Verteilung waren qualitative Vergleiche nur innerhalb des Gymnasiums oder für die Gesamtstichprobe möglich, nicht jedoch zwischen den einzelnen Mittelschulformen. Zu diesem Ende wurden die meisten Vergleiche zwischen den einzelnen Gymnasiums-Stufen geführt. Die Klassenverteilung war ein wenig einheitlicher (Siehe Abbildung 2.5), leider gab es aber keine Antworten aus der damaligen 4. Klasse.



Abbildung 2.4: Antworten nach Mittelschule



Abbildung 2.5: Antworten nach Klasse

Lernverhalten

Für die Abbildung 2.6 gilt: Je heller, desto früher, und demnach grundsätzlich auch besser (Da, wie schon erwähnt, wiederholtes Lernen zu einem längerfristigen / festeren Lernerfolg führt).

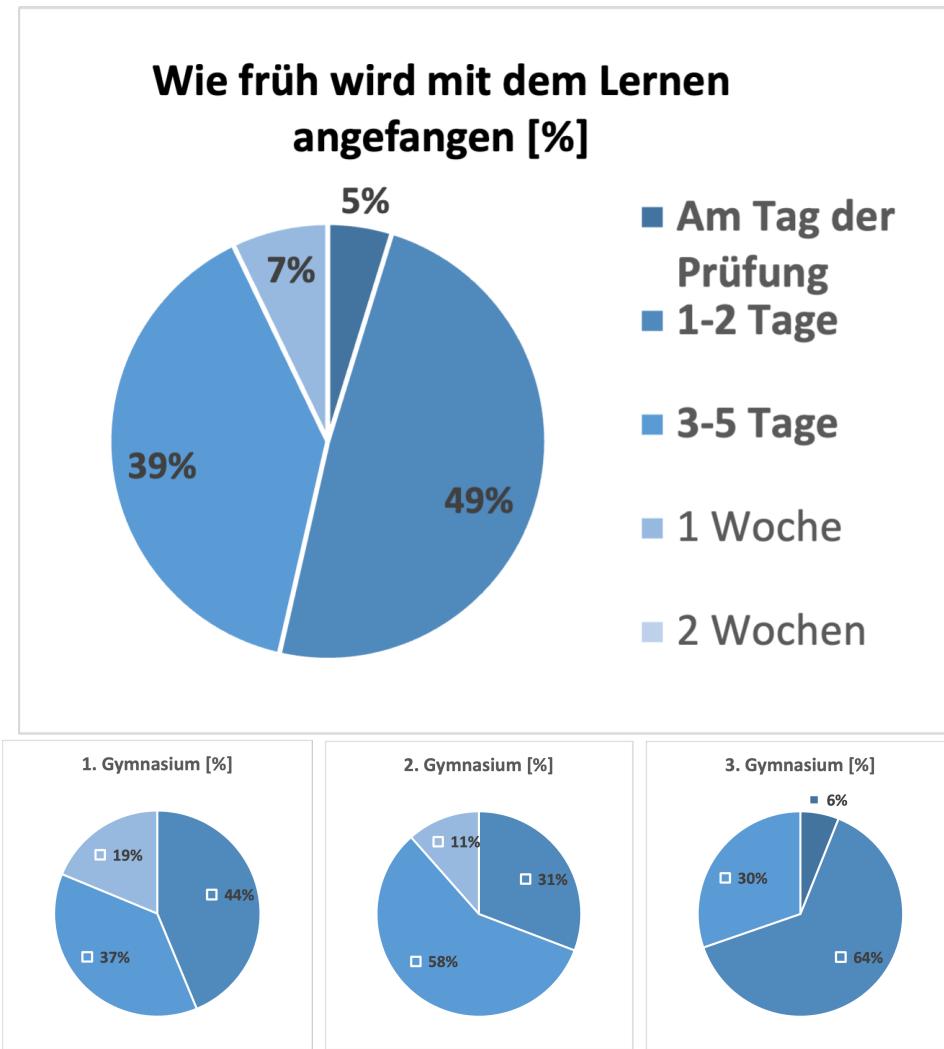


Abbildung 2.6: Antworten nach wie früh gelernt wird, alle Antworten und aufgebrochen in Gym. Stufe

Wie in Abbildung 2.6 sichtbar, fängt die Mehrheit aller befragten Schüler:innen erst sehr spät, nämlich regelmässig nicht früher als max. zwei Tage vor der Prüfung, mit dem Lernen an. Gewisse fangen sogar erst am Tag der

Prüfung mit dem Lernen an. Dies zeigte uns sehr klar, dass Verbesserungsbedarf in diesem Bereich besteht. Dies sahen die Befragten auch ein, denn eines der häufigsten genannten Verbesserungsvorschläge war tatsächlich „Früher anfangen“ (Siehe Abbildung 2.7).

5. Was müsste sich ändern, damit Sie zufriedener sind?

[More details](#)

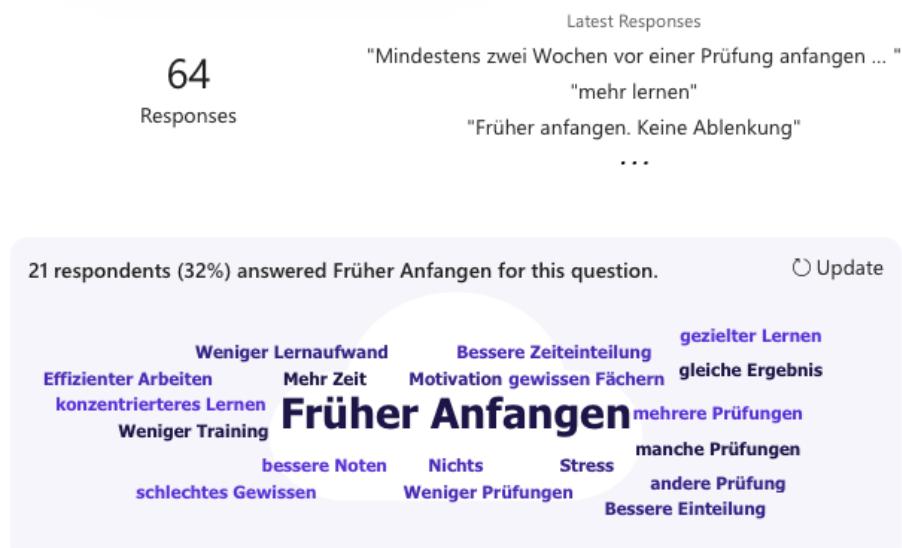


Abbildung 2.7: Die Häufigsten Antworten zur Frage „Was müsste sich ändern, damit Sie zufriedener sind?“

Stressmanagement

Dieser Abschnitt validiert unsere Vermutungen zum Lernstress stark, nämlich dass Lernstress unter den Schüler:innen oft vorkommt. Tatsächlich gaben rund **57%** aller Schüler:innen (Siehe Abbildung 2.8) an, dass sie oftmals Stress empfinden, und dieser Prozentsatz steigt in höheren Stufen und ist nie geringer als **50%**.

Fühlen Schüler:innen sich oft gestresst? [%]



Abbildung 2.8: Ja/Nein-Antworten zur Frage „Fühlen Schüler:innen sich oft gestresst?“, dargestellt pro Stufe und Insgesamt, in Prozent.

Auch war der am häufigsten genannte Grund für Prüfungsstress laut der Umfrage (Siehe Abbildung 2.9) interner Druck, während äusserer Druck am wenigsten häufig vorkam.

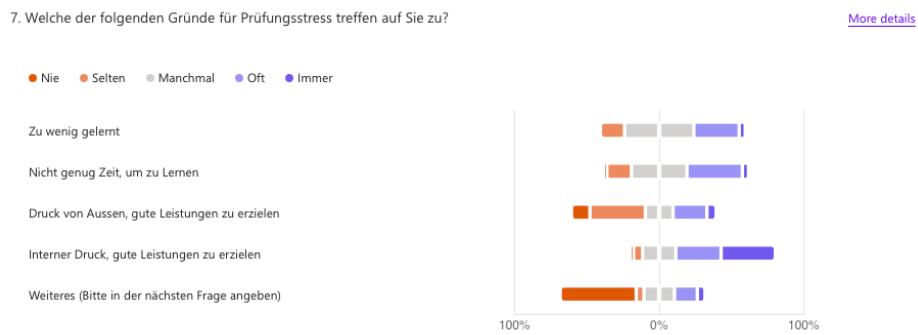


Abbildung 2.9: Wie häufig gewisse Gründe für Stress wahrgenommen werden. Hier gilt: je blauer, desto häufiger.

Als „Weitere Gründe“ wurden auch Gründe wie beispielsweise die Matura genannt, was für uns Sinn ergab, da dies in unserem Umfeld als ein sehr wichtiger Aspekt für die Zukunft gesehen wird.

Pausenmanagement

Für das Pausenmanagement fanden wir heraus, dass rund **71%** aller Befragten regelmässig Lernpausen machen, was scho nmal für uns gut aussah. Aber diese Lernpausen wurden laut der Umfrage oftmals nicht sinnvoll verbracht. So waren prominente Aktivitäten während einer Lernpause, laut der Umfrage, auf Sozialen Medien (bspw. Instagram, TikTok oder ähnliches) aktiv zu sein, was laut unseren Interviews kontraproduktiv für die Lernqualität ist. Manche verbrachten ihre Pausen jedoch auch sinnvoll, indem sie oft an die Luft gingen, etwas assen oder Sport trieben.

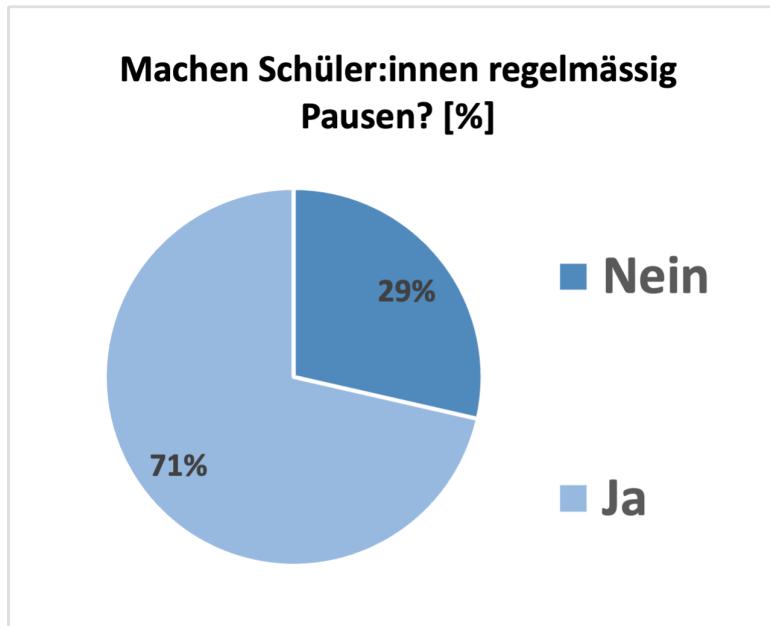


Abbildung 2.10: Antworten in Prozent zu ob Schüler:innen regelmässig Pausen machen.

Zeitmanagement

Die Untersuchung zur Prokrastination in diesem Abschnitt stellte eine der wichtigsten Validierungen für unsere Web-Applikation dar. Wie es sich hier herausstellte, gaben sehr viele unserer Befragten an (Siehe Abbildung 2.11), dass sie oft oder sogar immer prokrastinieren, und nur sehr wenige benutzten regelmässig Hilfsmittel, um ihre Zeit zu planen. Dies legte einen akuten Mangel dar, welcher durch unsere Lern-Applikation gefüllt werden kann.



Abbildung 2.11: Wie häufig prokrastiniert wird & Hilfsmittel für die Zeitplanung verwendet werden. Hier gilt ebenfalls: je blauer, desto häufiger.

Die Web-Applikation und Musik

Unsere Fragen zur Musik waren extrem kontrovers. Bei beiden Teilen der Likert-Frage waren die Antworten sehr ausgeglichen oder wiesen eine sehr leichte Tendenz ins Negative auf. Somit zeigte dies uns, dass Musik kein besonders begehrtes Feature sein würde, vor allem wenn viele diese als ablenkend empfinden.

Das Konzept einer Web-Applikation selbst wurde aber sehr begrüßt, mit „Ja“ als rund **77%** (Siehe Abbildung 2.12) aller Antworten. Auch waren alle vorgeschlagenen Features sehr beliebt, mit der Agenda an der Spitze mit **55** Stimmen und der Notenorganisation mit den wenigsten Stimmen (**40**). Aber dennoch wurden fast alle Features von mehr als der Hälfte der Befragten gewünscht.

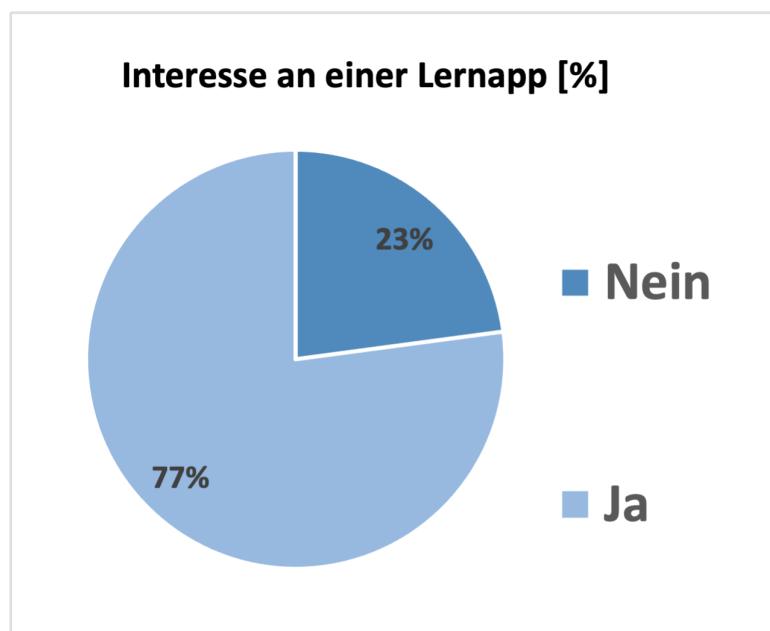


Abbildung 2.12: Interesse an einer Lernapp in Prozent.

Leider gab es hier keine nützlichen Feature-Vorschläge, aber wir bekamen insgesamt **35** seriöse Tipp-Vorschläge für unsere Web-Applikation, was für uns einen grossen Erfolg darstellt.

Erkenntnisse

Aus der Umfrage kamen hauptsächlich einige Erkenntnisse zur Web-Applikation heraus, und nur wenige zum Lernverhalten, welche jedoch hauptsächlich auch schon alle in den vorherigen Abschnitten aufgeführt wurden.

Wichtige Erkenntnisse für die Web-Applikation beinhalteten unter anderem:

- Prüfungsstress tritt in unserer Zielgruppe tatsächlich häufig auf und kommt am meisten von innen.
- Die Agenda war indes das meistgewollte Feature für unsere Web-Applikation, aber alle unsere Vorschläge in der Umfrage sind populär.
- Wir konnten einige gute Tipps direkt von den Befragten gewinnen.

Empfehlungen

Somit konnten wir nun auch als Letztes konkrete Empfehlungen formulieren. Diese bestanden zum grössten Teil aus den Vorschlägen für Tipps aus der Umfrage, aber auch aus der Erkenntnis, dass ein Feature gegen Prokrastination (beispielsweise Lern-Erinnerungen) nützlich sein könnte und dass die Agenda unsere Priorität sein sollte.

Also hatten wir schlussendlich auch aus der Umfrage-Analyse, neben den Validierungen für unsere Arbeit, auch ein paar konkrete Empfehlungen gewonnen.

2.5 Fazit der Recherche

Die durchgeführte Recherche konnte ihren Zweck als Grundlage für unsere Web-Applikation gut erfüllen, indem sie uns dank der drei Hauptelemente - die Literaturstudie, die Interviews und die Umfrage - einen differenzierten Standpunkt aus mehreren Perspektiven gab und unsere Ausgangsannahmen grundsätzlich validierte. Unsere vier Hauptthemenbereiche wurden gut abgedeckt und ebenfalls von mehreren Winkeln betrachtet, was uns viele differenzierte Erkenntnisse und viel Wissen lieferte. Somit ergaben sich einige

gute Anforderungen und Empfehlungen an die Web-Applikation, welche wir einbauen konnten, um die Bedürfnisse unserer Zielgruppe besser anzusprechen.

3. Methodik – Programmieren der Web-Applikation

3.1 Technologie- und Plattformwahl¹

Zu Beginn stand die grundsätzliche Plattformwahl im Zentrum: eine native Applikation (z. B. für Smartphones) oder eine webbasierte Lösung. Unter Berücksichtigung der Geräteunabhängigkeit, dem Verteil- und Updateaufwand, Entwicklungsressourcen sowie dem verfügbaren Zeitrahmen 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 des Technologie-Stacks auf Python mit dem Microframework Flask („Flask Documentation“, o. D.). Entscheidend waren vorhandene Vorkenntnisse, welche eine einfachere Umsetzung ermöglichen. 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 Code wurde dort

¹Gemini (Model 2.5 Pro): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 04.11.2025. Antwort als Basis.

gemeinsam versioniert und ausgetauscht; regelmässige Synchronisationen (Push/Pull) stellten einen konsistenten, aktuellen Projektstand sicher.

Für das Hosting der Produktivumgebung fiel die Wahl auf DigitalOcean. Ausschlaggebend waren die Verfügbarkeit kostenloser Credits über das GitHub Student Developer Pack, integrierte PostgreSQL-Datenbank-Hosting-Optionen und Unterstützung für Gunicorn/WSGI-Deployment. Die Domain kantikoala.app wurde ebenfalls über das Student Pack mit kostenloser .app-TLD registriert.

3.2 Anforderungsanalyse und -definition²

Die im Folgenden definierten Anforderungen sind das direkte Resultat der in Kapitel 2 (Recherche) durchgeföhrten Analyse. Die Interviews mit den PPP-Lehrpersonen und die Umfrage unter den Schüler:innen lieferten die „Must“- und „Kann“-Anforderungen (vgl. Abschnitt 2.3.5), welche die Basis für die funktionale Spezifikation der Applikation bildeten. 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 abgeleiteten Anforderungen an die Kanti Koala Web-Applikation sind wie folgt:

- **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

²Gemini (Model 2.5 Pro): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 04.11.2025. Antwort als Basis.

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. Natürlich sollte man die Agenda auch als .ics-Datei exportieren können. Ebenso sollte man neue Ereignisse eintragen können. Die Ereignisse sollten veränderbar sein. Die Farbe der Ereignisse sollten auch frei bestimbar 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 lösbar 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 ansprechendes UI haben, welches die Kernfunktionen (Agenda, Noten, Timer) ohne Schulung zugänglich macht.
- **To-Do Liste:** Basierend auf den Empfehlungen aus der Recherche (vgl. Abschnitt 2.3.5) sollten Nutzer:innen die Möglichkeit haben, Aufgaben in Kategorien zu organisieren, hinzuzufügen und zu löschen.

3.3 Systemarchitektur und Datenmodell³

Dieser Abschnitt beschreibt die technische Grundlage der Kanti Koala Web-Applikation, einschliesslich der Systemarchitektur, der Code-Struktur und der zugrundeliegenden Datenstruktur.

3.3.1 Systemarchitektur und Framework-Wahl

Die Kanti Koala App ist als Webanwendung konzipiert, die auf einem zentralen Backend-Server läuft. Das Kernstück der Anwendung ist das Python-Microframework Flask („Flask Documentation“, o. D.). Es steuert das Routing (die Zuordnung von URLs zu Funktionen), verarbeitet HTTP-Anfragen (GET, POST, etc.) und rendert die HTML-Templates für den Benutzer.

Wichtige Komponenten und Pakete

- **Flask-SQLAlchemy** („Flask-SQLAlchemy Documentation“, o. D.): Dient als Object-Relational Mapper (ORM) für die Datenbank. Es ermöglicht die Definition von Datenbanktabellen als Python-Klassen (Models) und vereinfacht Datenbankabfragen.
- **Flask-Bcrypt** („Flask-Bcrypt“, o. D.): Wird für die Sicherheit der Benutzerpasswörter eingesetzt. Es hasht und verifiziert Passwörter mithilfe des bcrypt-Algorithmus.
- **Flask-Migrate** („Flask-Migrate Documentation“, o. D.): Erleichtert Schema-Migrationen der Datenbank, wenn sich die Modelle (Tabellenstruktur) ändern.
- **Resend** („Resend Documentation“, o. D.): Dient als E-Mail-API für den Versand von systemgenerierten E-Mails, insbesondere für die „Passwort vergessen“-Funktion.

³Gemini (Model 2.5 Pro): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 04.11.2025. Antwort als Basis.

- **icalendar** („icalendar – iCalendar parser/generator“, o. D.): Eine Python-Bibliothek, die zum Parsen und Importieren von .ics-Kalenderdateien verwendet wird, um den Schulnetz-Stundenplan zu importieren.
- **itsdangerous** („ItsDangerous Documentation“, o. D.): Wird verwendet, um sichere, zeitlich begrenzte Tokens zu generieren, die für die „Passwort zurücksetzen“-Links benötigt werden.

Code-Struktur und Application Factory Pattern

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 (Muneeb, 2025).

Das Herzstück des Pakets ist die `create_app`-Funktion in `kkoala/__init__.py`. 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. In dieser Datei werden auch die Flask-Erweiterungen initialisiert und die Blueprints registriert.

Die Datei `kkoala/config.py` enthält Konfigurationsklassen (z.B. `DevConfig`, `ProdConfig`), welche wichtige Einstellungen wie den `SECRET_KEY`, die Datenbank-URL und API-Schlüssel verwalten. Die Konfiguration wird je nach Umgebungsvariable beim Start der App ausgewählt.

Die Routen der Anwendung sind in Blueprints aufgeteilt (`kkoala/routes/`), die eine Gruppierung von zusammengehörigen Endpunkten ermöglichen:

- **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.

Die Kernlogik ist ebenfalls modularisiert:

- **kkoala/algorithms.py**: Enthält ausschliesslich die komplexe Logik des Lernzeitalgorithmus (LZA).
- **kkoala/utils.py**: Beinhaltet wiederverwendbare Hilfsfunktionen und Decorators wie `@login_required` (prüft ob ein Benutzer angemeldet ist) und `@csrf_protect` (Schutz vor Cross-Site Request Forgery Angriffen).
- **kkoala/extensions.py**: Initialisiert Flask-Erweiterungen, um zirkuläre Importfehler zu vermeiden.

Deployment und WSGI-Schnittstelle

Die Datei `wsgi.py` im Hauptverzeichnis ist der standardisierte Einstiegspunkt für den Webserver. Sie importiert die `create_app`-Factory und erstellt das Flask-application-Objekt, das der WSGI (Web Server Gateway Interface) Spezifikation entspricht (Srivastav, 2022). Für den produktiven Einsatz verwenden wir Gunicorn („Green Unicorn“), einen robusten WSGI-HTTP-Server. Während der eingebaute Entwicklungsserver von Flask für Tests ausreicht, ist er nicht für hohe Last ausgelegt. Gunicorn agiert als leistungsfähiger „Middleman“ zwischen dem Internet und unserer Flask-Anwendung, verwaltet mehrere Worker-Prozesse und stellt so Leistung und Stabilität sicher („Launching a Flask Application with Gunicorn“, o. D.).

Frontend-Struktur

Die Benutzeroberfläche wird dynamisch auf dem Server generiert und als fertige HTML-Seiten an den Browser gesendet.

- **Templates** (`kkoala/templates/`): Enthalten alle HTML-Dateien. Flask verwendet die Template-Engine Jinja, um Python-Code direkt in HTML einzubetten (Schleifen, bedingte Blöcke, Variablenausgabe), was die Seiten dynamisch und personalisiert macht („Template Designer Documentation — Jinja Documentation“, o. D.; „Templates — Flask Documentation“, o. D.).
- **Statische Dateien** (`kkoala/static/`): Enthält CSS-Dateien für das Styling, JavaScript-Dateien für clientseitige Logik sowie Bilder (Logos, grafische Elemente), die vom Browser direkt geladen werden.

3.3.2 Datenmodell und -struktur

Datenbankwahl und Konfiguration

Die Datenstruktur ist in der Datei `models.py` durch SQLAlchemy-Modelle definiert. Für die lokale Entwicklung wird SQLite verwendet, da es einfach einzurichten ist. Für die Produktionsumgebung kommt PostgreSQL zum Einsatz, das besser für Mehrbenutzerbetrieb und hohe Last geeignet ist („Define and Access the Database“, o. D.; „PostgreSQL vs SQLite: The Ultimate Database Showdown“, 2025).

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).

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.

password Das gehashte Passwort (notwendig).

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

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 (notwendig).

total_hours_to_learn Die gesamte zu lernende Stundenanzahl für diese Priorität (notwendig).

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 (notwendig).

priority Prioritätsstufe (Integer, notwendig).

recurrence Wiederholungsregel des Ereignisses.

recurrence_id Eindeutige ID zur Gruppierung wiederkehrender Ereignisse.

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

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), **name** und **counts_towards_average**.

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

ToDoCategory

Speichert benutzerdefinierte Kategorien für die To-Do-Liste.

id Eindeutige ID (Primary Key).

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

name Name der Kategorie (notwendig).

ToDoItem

Speichert einzelne Aufgaben (To-Dos) innerhalb einer Kategorie.

id Eindeutige ID (Primary Key).

category_id Fremdschlüssel zur **ToDoCategory**-Tabelle (notwendig).

description Beschreibung der Aufgabe (notwendig).

Beziehungsstruktur⁵

Die Tabellen in der Datenbank sind durch Fremdschlüssel (Foreign Keys) miteinander verbunden, um die Beziehungen zwischen den verschiedenen Datenmodellen abzubilden. Beispielsweise verknüpft der Fremdschlüssel **user_id** in der **Event**-Tabelle jedes Ereignis mit dem zugehörigen Nutzer. Dadurch wird sichergestellt, dass jeder Kalendertermin eindeutig einem Benutzerkonto zugeordnet ist. Ähnliche Verknüpfungen existieren auch zwischen **Settings** und **User**, **PrioritySetting** und **Settings**, **Semester** und **User**, **Subject** und **Semester** sowie **Grade** und **Subject**. Die Abhängigkeiten und Kaskadenlöschen (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), **ToDoCategory** (1:n)

⁵Gemini (Model 2.5 Pro): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 04.11.2025. Antwort als Basis.

- `Settings` → `PrioritySetting` (1:n)
- `Semester` → `Subject` (1:n)
- `Subject` → `Grade` (1:n)
- `ToDoCategory` → `ToDoItem` (1:n)

3.4 Implementierung der Kernfunktionen

3.4.1 Authentifizierung und Account Management⁶

Ein sicheres Authentifizierungssystem ist die Voraussetzung für eine personalisierte App. Dies umfasst Anmelde- und Registrierungsfunktionen, eine „Passwort vergessen“-Option, eine Option zum Ändern des Passworts und zum Löschen des Kontos.

Sicherheit der Passwörter: Passwörter dürfen aus Sicherheits- und ethischen Gründen niemals im Klartext gespeichert werden. Die gewählte Lösung ist das Passwort-Hashing. Darunter versteht man die algorithmische Umwandlung eines Passworts in eine unumkehrbar verschleierte Version (Chiffertext) („What is Password Hashing?“, 2022).

Zusätzlich wird Salting angewendet: Vor dem Hashing werden zufällige Daten (ein „Salt“) zum Passwort hinzugefügt. 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.). Das Modul Flask-Bcrypt übernimmt diese Aufgaben automatisch („Flask-Bcrypt“, o. D.). 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.

⁶ChatGPT (Model GPT-5): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 04.11.2025. Antwort als Basis.

„Forgot-Password“: Eine wesentliche Funktion ist das Zurücksetzen des Passworts, falls man ihn vergessen hat. Um den Reset-Link nur einmal verwendbar zu machen, wird der Token (der die Identität des Nutzers prüft) mit dem Hash des alten Passworts generiert. Diese Idee stammt von Cistic (2024). Da das Passwort beim Zurücksetzen geändert wird, ändert sich auch der Hash, wodurch der alte Token sofort ungültig wird.

Für den E-Mail-Versand (z.B. des Reset-Links) wird die API von Resend genutzt, da sie eine für unser Projekt ausreichende kostenlose Stufe bietet und den Versand über unsere eigene Domain kantikoala.app erlaubt.

Bei Fehlern (z.B. E-Mail bereits vergeben, falsches Passwort) erhält der Nutzer eine spezifische Rückmeldung über „Flash-Nachrichten“, eine Funktion von Flask (Acsany, o. D.).

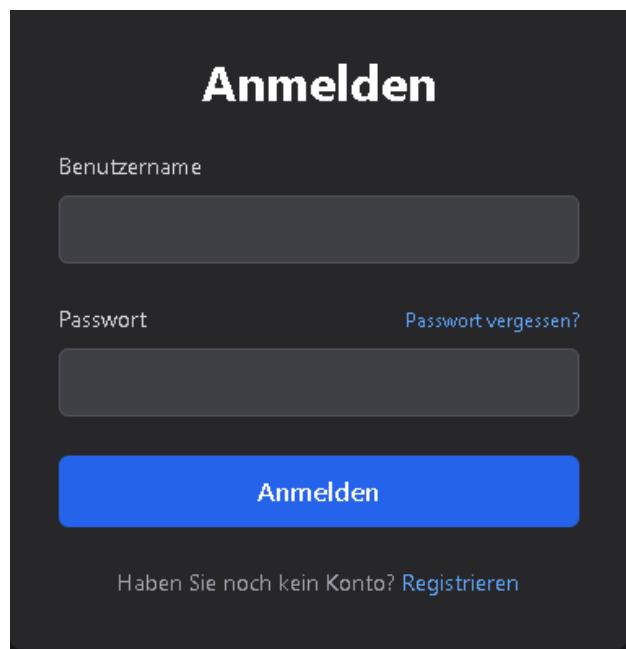


Abbildung 3.1: Screenshot des Anmeldeprozesses

3.4.2 Homescreen / Dashboard⁷

⁷ChatGPT (Model GPT-5): „Korrigiere Grammatik und Rechtschreibfehler im folgenden Text. [...].“, 09.11.2025. Antwort als Basis.

Der Homescreen (Dashboard) ist die zentrale Einstiegsseite nach dem Login. Er fasst die wichtigsten Informationen kompakt zusammen, um Nutzer:innen einen schnellen Überblick über ihren aktuellen Stand zu geben. Der Home-screen ist auch für nicht angemeldete Besucher:innen verfügbar. In diesem öffentlichen Modus wird eine gekürzte Ansicht präsentiert, die prominente Aufrufe zum Anmelden oder Registrieren enthält. Der tägliche Tipp wird angezeigt, persönliche Daten wie Noten, Termine oder Prüfungen bleiben jedoch ausgeblendet.

Aufbau

Das Dashboard ist in mehrere Kacheln (Cards) gegliedert, die folgende Informationen anzeigen:

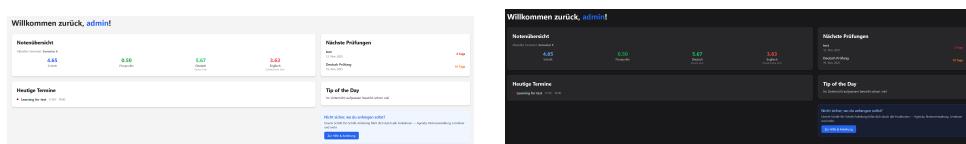
- **Notenübersicht:** Zeigt eine Zusammenfassung der schulischen Leistung des aktuellsten Semesters (aktueller Schnitt, Pluspunkte sowie bestes und schwächstes Fach). Sind noch keine Noten eingetragen, erscheint ein Hinweis mit Aufforderung zur Eingabe.
- **Heutige Termine:** Listet alle für den aktuellen Tag im Kalender eingetragenen Ereignisse mit Titel und Uhrzeit auf. Dies dient auch zur Erinnerung an Lernblöcke, falls welche am Tag stattfinden, was laut der Recherche von Vorteil ist (vgl. Abschnitt 2.4.4).
- **Nächste Prüfungen:** Zeigt die anstehenden Prüfungen der nächsten 21 Tage. Findet eine Prüfung am heutigen Tag statt, wird sie in einem dunklen Rot hervorgehoben; Prüfungen innerhalb der nächsten sieben Tage erscheinen in Rot, spätere Prüfungen werden orange markiert.
- **Tip of the Day:** Ein Bereich mit täglich wechselnden, kurzen Hinweisen — dieser Bereich ist sowohl für eingeloggte als auch für nicht eingeloggte Nutzer:innen sichtbar.
- **Hilfe-Hinweis:** Eine Kachel, die Nutzer:innen auf die Hilfe-Seite ([/hilfe](#)) aufmerksam macht, um den Einstieg zu erleichtern.

Tip of the Day

Eine Kernanforderung der App ist die tägliche Bereitstellung eines neuen Tipps auf der Startseite, etwa zu schulbezogenen oder allgemeinen Lernthemen. Die Implementierung hierfür nutzt eine einfache Modulo-Rechnung:

$$\text{Tip of the Day (Index)} = (\text{Tag des Jahres}) \bmod (\text{Anzahl der Tipps})$$

So wird an jedem Tag genau ein Tipp angezeigt und über das Jahr hinweg zyklisch alle Tipps durchlaufen. Der Titel dieses Features steht als englischer UI-Anker („Tip of the Day“) in der Oberfläche, weil der Ausdruck als prägnanter UI-Anker wiedererkennbar ist; der eigentliche Inhalt (der Tipp) bleibt jedoch vollständig auf Deutsch, um Verständlichkeit und lokalen Bezug sicherzustellen.



(a) Homescreen – Light Mode

(b) Homescreen – Dark Mode

Abbildung 3.2: Homescreen / Dashboard in Light (links) und Dark Mode (rechts). Die Darstellung zeigt die Kacheln für Notenübersicht, heutige Termine, nächste Prüfungen und den „Tip of the Day“.



Abbildung 3.3: Homescreen für nicht angemeldete Nutzer:innen im Dark Mode.

3.4.3 Agenda und Kalenderverwaltung⁸

Die Agenda ist eine zentrale Kernfunktion der Anwendung. Sie ermöglicht das Erstellen, Bearbeiten und Löschen von Terminen mit minimalem Aufwand. 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 ganztägig 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 („Documentation“, o. D.). Diese Wahl wurde getroffen, da sie etabliert, funktionsreich und gut dokumentiert ist.

Import und Export von .ics-Dateien

Die Agenda unterstützt den Import des individuellen Stundenplans aus **schulnetz**, der von der Kantonsschule zur Anzeige von Stundenplänen genutzt wird. Da **schulnetz** Kalenderdaten im **.ics**-Format exportiert, kann die Datei zunächst dort exportiert und anschliessend in unsere Anwendung importiert werden. Dies reduziert manuellen Erfassungsaufwand erheblich. Zusätzlich besteht die Möglichkeit, die gesamte Agenda als **.ics**-Datei zu exportieren, um sie in anderen Kalenderanwendungen zu nutzen. Wenn man den Kalendar aus unserer Applikation exportiert und dann auch wieder in unserer Applikation importiert, bleiben Priorität und Farbe erhalten.

3.4.4 Der Lernzeitalgorithmus⁹

Der Lernzeitalgorithmus (LZA) ist der Kern unserer Web-App. Er automatisiert die Planung der notwendigen Lernzeiten für die anstehenden Prüfungen

⁸ChatGPT (Model GPT-5): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 04.11.2025. Antwort als Basis.

⁹ChatGPT (Model GPT-5): „Korrigiere Grammatik und Rechtschreibfehler im folgenden Text. [...].“, 04.11.2025. Antwort als Basis.

04:00									
05:00									
06:00									
07:00									
08:00	08:25 - 09:10 ENG-G4b-ToV			07:30 - 08:15 DEU-G4b-ScB		07:30 - 08:15 Spam.4.1b			
09:00	09:20 - 10:05 ENG-G4b-ToV	09:20 - 10:05 Einf.4.1		08:25 - 09:10 DEU-G4b-ScB	08:25 - 09:10 Exkursion D: Besuch ch-MC Leitung: Mirko	08:25 - 09:10 Spam.4.1b	08:25 - 09:10 Spam.4.1b	09:20 - 10:05 fra-F.4.b	
10:00	10:25 - 11:10 ENG-G4b-ToV	10:25 - 11:10 Einf.4.1 SO-	11:00 - 13:00 Sprechst. Abt. G4b-ABT-G4 Paola Gallati Psycholo	10:25 - 11:10 PHY-G4b-GaA	10:25 - 11:10 PHY-G4b-GaA	10:15 - 13:15 Perspektiven auf die Berufswelt. Fremde Veranstaltung zur Studienvahl (Rotary)	10:15 - 13:15 MAT-G4b-HaS 11:00 - 13:30 Sprechstund MAT-G4b-Ha Paola Gallati Psychologin	10:25 - 11:10 MAT-G4b-HaS 11:00 - 13:30 Sprechstund MAT-G4b-Ha Paola Gallati Psychologin	
11:00				11:20 - 12:10 12:35 - 13:20 PHY-G4b-GaA	11:20 - 12:10 spm.4.bc				
12:00									
13:00									
14:00	13:30 - 14:15 Spam.4.1a	13:30 - 14:15 Spam.4.1a	13:30 13:30 13:30 13:30 13:30 spk.4spk.4spk.4spk.4spk.4spk.4		13:30 - 14:15 MAT-G4b-HaS		13:30 - 14:13 13:30 - 14:13 Einf.4.1 Einf.4.1	13:30 - 14:13 13:30 - 14:13 Prüfung	
15:00	14:25 - 15:10 Spam.4.1a	14:25 - 15:10 Spam.4.1a	14:25 14:25 14:25 14:25 14:25 spk.4spk.4spk.4spk.4spk.4		14:25 - 15:10 fra-F.4.b		14:25 - 15:10 Einf.4.1	14:25 - 15:10 Einf.4.1	14:25 - 15:10 Prüfung
16:00	15:20 - 16:05 Spam.4.1a	15:20 - 16:05 Spam.4.1a	15:20 - 16:05 DEU-G4b-ScB	15:20 - 16:05 mar.4.1	15:20 - 16:05 fra-F.4.b		15:20 - 16:05 GSC-G4b-TeD		16:15 - 17:00 GSC-G4b-TeD
17:00				16:15 - 17:00 DEU-G4b-ScB	16:15 - 17:00 mar.4.1				
18:00				17:30 - 19:00 Learning for fra-F.4.b		16:00 - 20:00 Learning for fra-F.4.b	16:00 - 20:00 Learning for fra-F.4.b		18:00 - 00:00 Exkursion 3. und 4. Klasse, Lange Nacht der Mathematik, Leitung: Asako Frey
19:00									
20:00									
21:00									
22:00									

Abbildung 3.4: Beispiel einer Agenda mit Lernblöcken der LZA

eines Nutzers. Wir bezeichnen unseren Mechanismus als Algorithmus, da 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 LZA verwendet globale Benutzereinstellungen sowie prüfungsspezifische Prioritätseinstellungen als Eingabeparameter, um eine regelbasierte Lernplanung zu ermöglichen. Diese Parameter wurden auf Basis der Ergebnisse aus der Recherche (Kapitel 2) als wichtig erachtet, um eine flexible und realistische Lernplanung zu ermöglichen.

Das zentrale Planungsziel des LZA ist es, die definierten totalen 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.

Hinweis zum Umgang mit fehlenden Endzeiten: Falls ein Ereignis keine Endzeit besitzt, behandelt der Algorithmus dieses Ereignis als sogenanntes *Punkt-Ereignis*, d. h. das Start- und Endzeitpunkt sind identisch. Dies ist insbesondere relevant, da importierte Kalenderdaten (z.B. aus **schulnetz**) manchmal keine Endzeit angeben. Eine Korrektur dieses Verhaltens bereits beim Import könnte zu Darstellungsproblemen in der Kalenderansicht führen. Daher wurde entschieden, diese Anpassung erst im Algorithmus selbst vorzunehmen, um sowohl die Integrität der Planung als auch die Kompatibilität mit der Kalenderdarstellung zu gewährleisten.

1. Zyklische Neuberechnung der Anforderungen (Recycling):

- **Flexibilitätsbereinigung:** Alle vom System selbst geplanten, aber nicht gesperrten Lernblöcke für die aktuelle Prüfung werden gelöscht. Dies ermöglicht eine Neuplanung, falls sich die Rahmenbedingungen geändert haben. Ein Block wird gesperrt, wenn der/die Nutzer:in ihn manuell verändert hat.
- **Soll-Stunden-Ermittlung:** Die noch zu erbringende Lernzeit wird neu berechnet unter Berücksichtigung bereits absolvieter oder manuell gesperrter Stunden.

2. Rückwärts-Iterative Planung:

- Die Planungsstrategie beginnt beim Prüfungstermin und arbeitet sich tageweise, jedoch maximal drei Wochen, bis zum aktuellen

Datum vor. Dies stellt sicher, dass Lernblöcke mit höchster Dringlichkeit (nächste zur Prüfung) zuerst belegt werden.

- An jedem Tag wird die maximale Lernzeit für diese spezifische Prüfung ermittelt, um das tägliche Zeitlimit einzuhalten.

3. Platzierung und strikte Konfliktvermeidung:

- **Bevorzugter Slot:** Primär wird versucht, einen Lernblock in der vom Nutzer festgelegten bevorzugten Lernzeit zu platzieren.
- **Konfliktprüfung:** Die Verfügbarkeit wird gegen alle Kalendereinträge geprüft. Ein 30-minütiger Puffer vor und nach jedem Ereignis verhindert zeitliche Überlappungen.
- **Alternative Slots:** Falls die bevorzugte Zeit belegt ist, sucht eine dedizierte Funktion den grössten verfügbaren, konfliktfreien Zeitabschnitt.
- **Echtzeit-Aktualisierung:** Nach erfolgreicher Generierung wird der Lernblock sofort zur Liste der aktuellen Ereignisse hinzugefügt, um nachfolgende Überlappungen auszuschliessen.

4. Ergebnisrückgabe und Zusammenfassung:

- Der Algorithmus gibt eine detaillierte Zusammenfassung zurück: Gesamtzahl hinzugefügter Lernblöcke, geplante Gesamtstunden und Planungsstatus pro Prüfung.

3.4.5 Notenverwaltung

In der Kantonsschule sind die Noten entscheidend für die Promotion in die nächste Stufe. Aus diesem Grund wurde ein Feature implementiert, in dem man seine Noten pro Semester speichern kann. Jedes Semester wird beim Erstellen automatisch mit den jeweiligen Fächern des entsprechenden Semesters an der Kanti Baden vorbefüllt. Nutzer:innen können Fächer löschen und hinzufügen. Das Feature berechnet automatisch die Durchschnitte pro Fach und Semester.

Ebenfalls ist ein Notenrechner integriert, mit dem man ermitteln kann, welche Note man in einem Fach benötigen würde, um einen bestimmten Zielschnitt zu erreichen. Die Idee zu diesem Feature stammt von der App „Pluspoints“ (Junker, 2025). Ein Manko dieser App war jedoch das Fehlen einer automatischen Fächeraddition pro Semester, welches wir in unserer App implementiert haben. Da die App an Kanti-Schüler:innen gerichtet ist, ist die Anzahl der Semester auf acht begrenzt. Neue Semester werden in der Anzeige stets oben hinzugefügt, um das aktuellste Semester zuerst anzuzeigen.

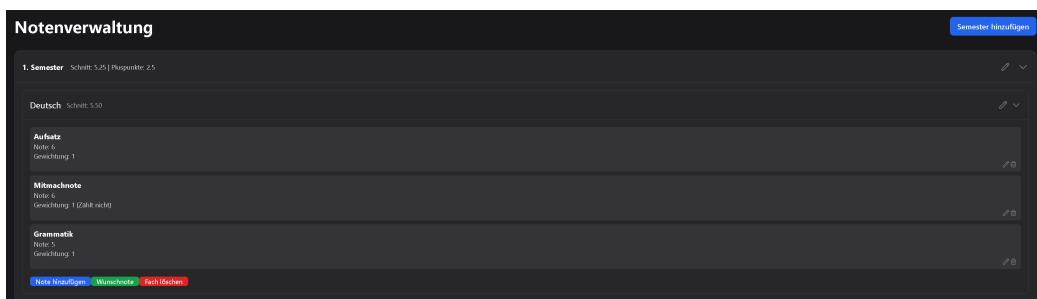


Abbildung 3.5: Screenshot der Notenverwaltungs-Seite

3.4.6 Lerntools (Timer und Tipps)

Lerntimer

Ein weiteres wichtiges Feature ist der Lerntimer. Dieser Timer basiert auf der Pomodoro-Technik, welche in Kapitel 2 (Recherche) erläutert wurde. Die Recherche, insbesondere die Interviews, validierte die Nützlichkeit dieser Technik zur Förderung der Disziplin (vgl. Abschnitt 2.3.5), weshalb die Implementierung eines solchen Timers als sinnvoll erachtet wurde. Der Timer hat die Standard-Einstellungen von 25 Minuten Lernen und 5 Minuten Pause, welche vom Nutzer individuell angepasst werden können.



Abbildung 3.6: Screenshot der Lerntimer-Seite

Lerntipps

Ein weiteres Feature unserer App sind die Lerntipps. Diese sind in verschiedene Kategorien aufgeteilt, u. a. Zeitmanagement, allgemeine Tipps der Kanti Baden, Stressmanagement und Lernmethoden. In jeder Kategorie gibt es verschiedene Tipps, welche wir direkt aus unserer Recherche (Interviews, Umfrage, Literatur) gesammelt haben. Diese Tipps sollen den Nutzer:innen helfen, ihr Lernverhalten zu verbessern. Beim Design für dieses Feature haben wir uns entschieden, dass die Bullet-Points der Tipps ein kleines Koala-Symbol sind, um das Thema der App widerzuspiegeln. Der Koala wurde von uns selber entworfen. Basierend auf dem Feedback aus dem Usability-Test (siehe Abschnitt 3.7.2, Person 2) haben wir auch einen „Scroll-To-Top“-Button implementiert, um die Navigation auf dieser langen Seite zu erleichtern.



Abbildung 3.7: Koala Bullet Point für Lerntipps

A screenshot of the 'Lerntipps' section in the app interface. The title 'Lerntipps' is at the top. Below it are two dropdown menus: 'Lernen' and 'Lernmethoden'. Under 'Lernen', there is a tip about SQ3R. Under 'Lernmethoden', there are three tips: KWL, SQ3R, and LOCI. Each tip has a small icon and a brief description.

Abbildung 3.8: Screenshot der Lerntipps-Seite

3.4.7 Benutzerkonfiguration (Einstellungen)

In den Einstellungen kann der Nutzer verschiedene globale Einstellungen für die App vornehmen. Dazu gehören die globalen Parameter für den LZA (bevorzugte Lernzeit, ob am Wochenende gelernt werden soll) und die Farben

der Prioritätsstufen. Diese Einstellungen werden in der Datenbank gespeichert und beim Laden der App abgerufen, um eine personalisierte Erfahrung zu bieten. Zusätzlich kann man hier auch sein Konto löschen und sein Passwort ändern. Wie im Abschnitt 3.7.2 (Feedback Person 2) erwähnt, wurde hier auch die Einstellung für das Farbschema (hell, dunkel, automatisch) implementiert.

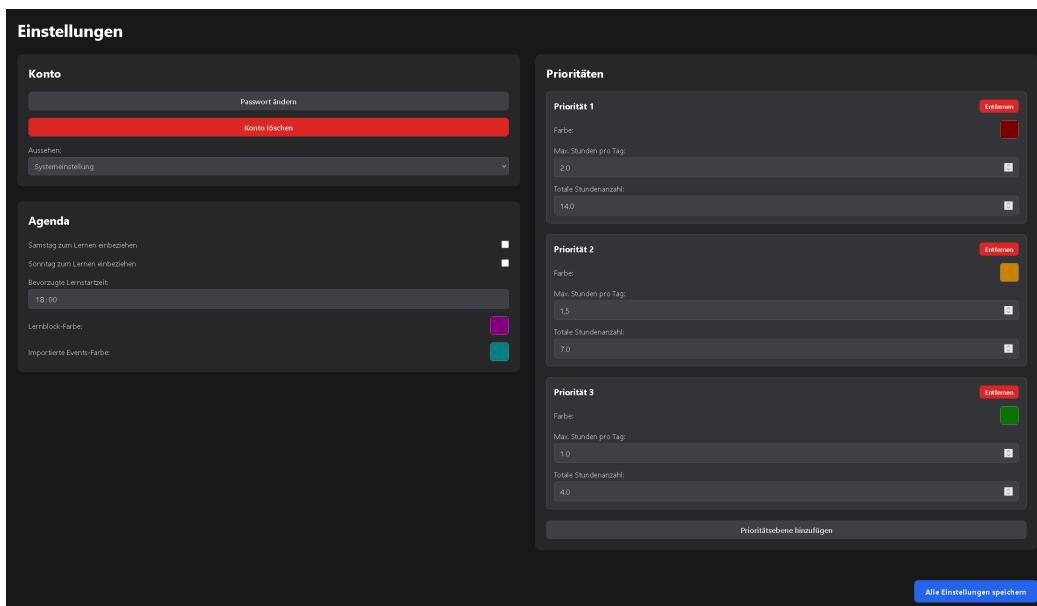


Abbildung 3.9: Screenshot der Einstellungen-Seite

3.4.8 Informationsseiten¹⁰

Über uns-Seite

Auf der „Über uns“-Seite stellen wir das Team hinter KantiKoala vor. Wir geben einen Einblick in unsere Motivation, die App zu entwickeln, und erläutern die Ziele, die wir mit KantiKoala verfolgen. Ausserdem erklären wir,

¹⁰Claude (Model Sonnet 4.5): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 09.11.2025. Antwort als Basis.

worin sich die KantiKoala-Web-App von anderen Lern-Apps unterscheidet. Zudem bieten wir Kontaktinformationen an, damit Nutzer:innen uns bei Fragen oder Feedback erreichen können. Als kleines „Easter Egg“ haben wir auf der Teamseite ausserdem die Kanti-Katze als Mitgründer hinzugefügt, um zu weiter den Punkt zu stärken, dass die App von Schüler entwickelt wurde.

Dein KantiKoala Team

KantiKoala wurde als Maturitätsarbeit an der Kantonsschule Baden entwickelt. Das Projekt vereint Forschung, Softwareentwicklung und User Experience Design.



Aryan Anand
Mitgründer



Simon Haddon
Mitgründer



Kanti-Chatz
Moralische Unterstützung (Mitgründer, inoffiziell)

Unser Ansatz ist praxisorientiert: Von Schüler, für Schüler:innen – mit dem Ziel, Lösungen für alltägliche Herausforderungen im Schulalltag zu schaffen.

Fotos: © Team KantiKoala.

Abbildung 3.10: Screenshot des Easter-Eggs auf der Über uns-Seite

Hilfe-Seite

Basierend auf dem Feedback aus den Usability-Tests (siehe Abschnitt 3.7.2, Person 1) wurde deutlich, dass eine Anleitung für die App fehlte. Deshalb haben wir die Hilfeseite angelegt, die eine vollständige Anleitung zur Nutzung der Anwendung enthält. Sie erklärt alle Features, bietet einen Schnellstart und Best Practices zur optimalen Nutzung der App.

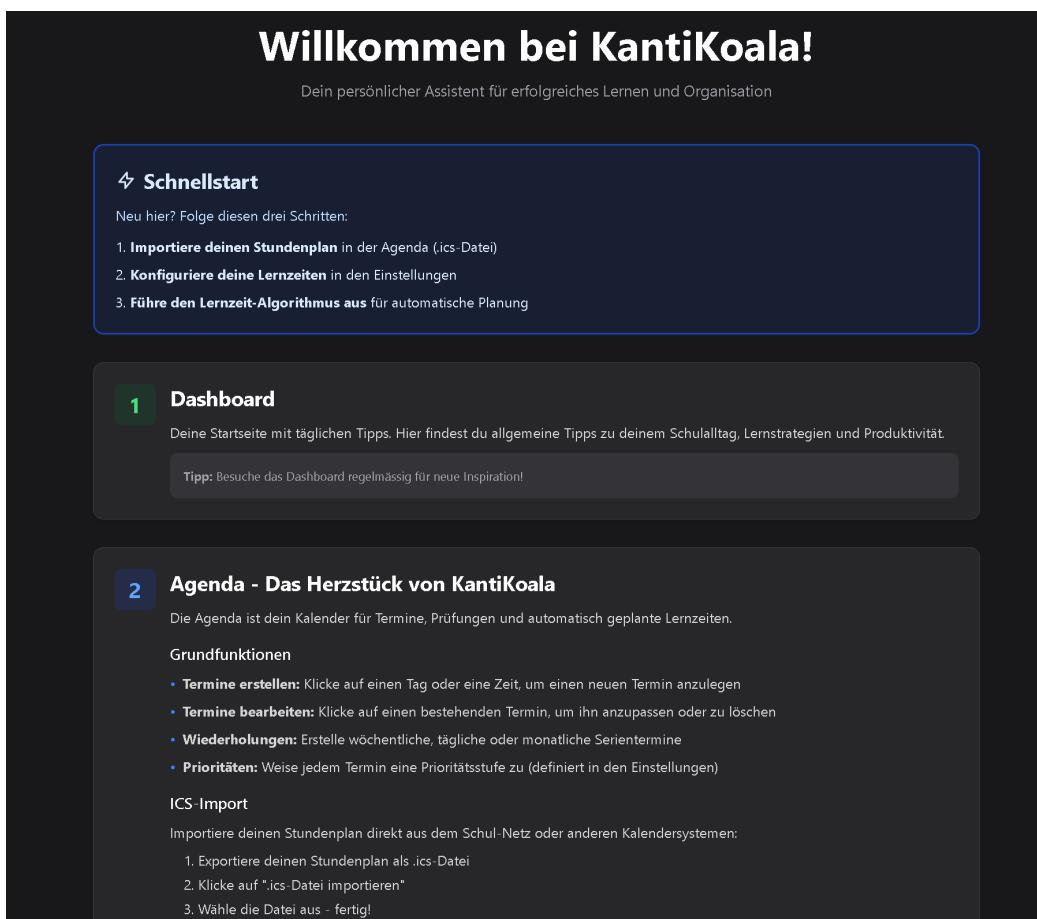


Abbildung 3.11: Screenshot der Hilfe-Seite

Datenschutz-Seite

Als rechtlich verpflichtender Bestandteil der Web-Applikation wurde eine Datenschutzerklärung gemäss den Vorgaben des Eidgenössischen Datenschutz- und Öffentlichkeitsbeauftragten (EDÖB) erstellt („Datenschutzerklärungen im Internet“, o. D.).

Die Datenschutz-Seite informiert die Nutzer:innen über die Erhebung, Verarbeitung und Speicherung ihrer personenbezogenen Daten. Sie ist über den Footer der Webseite sowie unter [/datenschutz](#) jederzeit zugänglich und erfüllt die gesetzlichen Transparenzpflichten.

Wichtige Inhalte der Datenschutzerklärung umfassen:

- Verantwortliche Stelle (Betreiber der Applikation)
- Art der erhobenen Daten (Kontodaten, Kalendereinträge, Noten, etc.)
- Rechtsgrundlage und Zweck der Verarbeitung
- Speicherort (DigitalOcean-Server in Frankfurt) und Speicherdauer
- Verwendung von Cookies
- Betroffenenrechte (Auskunft, Berichtigung, Löschung, Datenportabilität)
- Kontaktmöglichkeit für Datenschutzanfragen

3.4.9 To-Do Liste¹¹

Als direkte Umsetzung einer „Kann“-Anforderung aus den Interviews (vgl. Abschnitt 2.3.5) wurde eine To-Do-Listen-Funktion implementiert. Diese Funktion dient der einfachen Organisation von Hausaufgaben oder anderen anstehenden Aufgaben.

Die Funktionalität ermöglicht es Nutzer:innen:

¹¹Claude (Model Sonnet 4.5): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 10.11.2025. Antwort als Basis.

- **Kategorien zu erstellen:** Um Aufgaben thematisch zu gruppieren (z.B. nach Schule, Haushalt oder anderen Lebensbereichen).
- **Aufgaben hinzuzufügen:** Zu jeder Kategorie können einzelne ToDoItem-Einträge mit einer Beschreibung hinzugefügt werden.
- **Aufgaben zu löschen:** Sowohl einzelne Aufgaben als auch ganze Kategorien (mitsamt allen enthaltenen Aufgaben) können entfernt werden, sobald die Nutzer:innen dies erledigt haben.

Dies bietet eine grundlegende Struktur zur Aufgabenverwaltung, wie sie in der Recherche als nützlich identifiziert wurde.

3.5 Übergreifende Aspekte: Sicherheit und Design¹²

Neben den Kernfunktionen wurden übergreifende Aspekte wie Sicherheit und ein konsistentes Design implementiert, die die gesamte Anwendung betreffen.

3.5.1 Sicherheitskonzept

Zusätzlich zur Passwortsicherheit (behandelt in Abschnitt 3.4.1) wurden weitere Massnahmen zum Schutz der Nutzerdaten und der Anwendungsintegrität implementiert.

Transportverschlüsselung (HTTPS)

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

¹²Claude (Model Sonnet 4.5): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 10.11.2025. Antwort als Basis.

– 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 stellt so eine verschlüsselte Verbindung zur Domain `kantikoala.app` sicher („What is HTTPS?“, o. D.).

CSRF-Schutz

Neben der reinen Authentifizierung ist es entscheidend, die Aktionen eines angemeldeten Benutzers abzusichern. Eine häufige Schwachstelle ist 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 auszuführen (z.B. Konto löschen), ohne dass der Benutzer es merkt, etwa durch Klick auf einen bösartigen Link (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.

1. Für jede Benutzersitzung wird ein einzigartiges, geheimes Token generiert und auf dem Server gespeichert.
2. Dieses Token wird in alle Formulare, die eine Zustandsänderung bewirken (z.B. Einstellungen ändern), als verstecktes Feld eingebettet.
3. Beim Abschicken vergleicht der Server das vom Client gesendete Token mit dem in der Sitzung gespeicherten Token.
4. Stimmen sie 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 (AJAX) wird das Token aus einem Meta-Tag ausgelesen und in einem HTTP-Header (`X-CSRF-Token`) mitgesendet. Dies stellt sicher, dass jede datenverändernde Aktion legitim vom Benutzer und von unserer eigenen Webseite stammt.

Obwohl es fertige Module wie `Flask-WTF` gibt, haben wir uns entschieden, unseren eigenen Decorator zu schreiben, um ein tieferes Verständnis für die Funktionsweise von CSRF-Schutzmechanismen zu erlangen.

3.5.2 Design und Benutzeroberfläche

Das Design der Web-App haben wir mit der CSS-Bibliothek Tailwind CSS umgesetzt („Get started with Tailwind CSS“, o. D.). Diese Wahl wurde getroffen, da Tailwind es uns ermöglicht, schnell und konsistent ansprechende Oberflächen zu gestalten, indem wir vordefinierte Utility-Klassen verwenden. Dies erleichtert die Umsetzung eines einheitlichen Designs und beschleunigt den Entwicklungsprozess erheblich.

Farbkonzept und Theming

Die Farbwahl folgt praktischen Erwägungen: gut unterscheidbare Zustände, ruhige Grundflächen und konsistente Akzente.

- Neutrale Hintergründe (hell/dunkel): dezente Grau-/Zinc-Töne als Bühne für Inhalte.
- Akzente: Primärfarbe Blau und Grün für interaktive Elemente (Buttons, Links) zur Hervorhebung.
- Rückmeldungen: Grün für erfolgreiche Aktionen, Rot für Fehlerzustände.

Die Töne sind bewusst leicht gedämpft gewählt, um eine ruhige, einheitliche Anmutung über helle und dunkle Oberflächen zu erhalten.

Ein spezifisches Farbkonzept wurde für die Agenda (vgl. `kkoala/consts.py`) entwickelt:

Importierte Ereignisse (Stundenplan)

Beim Import von .ics-Dateien werden alle Ereignisse automatisch mit der Standardfarbe #6C757D (mittleres Grau) eingefärbt. Grau ist neutral und tritt gegenüber den farbcodierten Prioritätsstufen zurück, sodass wichtigere Ereignisse visuell dominieren. Es bietet dennoch ausreichenden Kontrast zu weissen Hintergründen. Diese Standardfarbe kann in den Einstellungen global angepasst werden. Alle bereits importierten Ereignisse übernehmen dann die neue Farbe, was eine konsistente visuelle Darstellung des gesamten Stundenplans gewährleistet.

Prüfungsergebnisse nach Prioritätsstufen

Prüfungen werden basierend auf ihrer Prioritätsstufe automatisch eingefärbt, um die Dringlichkeit visuell zu kommunizieren (vgl. `kkoala/consts.py`). Die Farbwahl folgt einem Ampel-System:

- **Priorität 1 (Höchste Dringlichkeit): #770000 – Dunkles Rot**
(Dunkles Rot ist weniger intensiv als reines Rot, bleibt aber gut sichtbar).
- **Priorität 2 (Mittlere Dringlichkeit): #ca8300 – Orange** (Bildet die Abstufung zwischen Rot und Grün).
- **Priorität 3 (Niedrigste Dringlichkeit): #097200 – Dunkles Grün**
(Dunkles Grün bietet ausreichenden Kontrast).

Alle Prioritätsfarben können in den Einstellungen angepasst werden.

Algorithmisch generierte Lernblöcke

Die vom LZA erzeugten Lernblöcke erhalten standardmäßig die Farbe #0000FF (reines Blau). Blau liegt ausserhalb des Ampel-Farbspektrums (Rot-Orange-Grün) und der Stundenplanfarbe (Grau), wodurch Lernblöcke klar als eigene Kategorie erkennbar sind. Auch diese Farbe kann in den Einstellungen angepasst werden.

Dark Mode

Basierend auf dem Feedback aus den Usability-Tests (Abschnitt 3.7.2, Person 2) wurde ein wechselbarer Dark Mode implementiert. Nutzer:innen können in den Einstellungen (Abschnitt 3.4.7) zwischen Light Mode, Dark Mode und automatischem Modus (abhängig von Systemeinstellungen) wählen. Das Design wurde so angepasst, dass alle Elemente in beiden Modi gut lesbar und ansprechend sind. Tailwind CSS erleichtert die Umsetzung durch integrierte Dark-Mode-Klassen. Flask kann mit einem Content Processor so konfiguriert werden, dass jede Seite die gebrauchte Einstellung erhält.

Fehlerbehandlung (404/500)

Spezielle Fehlerseiten wurden für die HTTP-Statuscodes 404 (Nicht gefunden) und 500 (Interner Serverfehler) gestaltet. Die Seiten sind bewusst reduziert gestaltet:

- Zentrale Karte mit Statuscode, kurzer Erklärung und primärer Aktion (zurück zur Startseite).
- Konsistentes Branding (Logo, Basisfarben) für Wiedererkennung.
- Keine technischen Details; klare Orientierung statt Verunsicherung.

Ziel ist, Nutzer:innen schnell zurück in einen funktionierenden Kontext zu führen und gleichzeitig sensible Interna bei Serverfehlern nicht offenzulegen.



Abbildung 3.12: Screenshot der 404-Fehlerseite

3.6 Testing und Usability-Evaluation¹³

Um die Qualität und Zuverlässigkeit der Kanti Koala Web-App sicherzustellen, wurden zwei Arten von Tests durchgeführt: funktionale Tests zur Überprüfung der korrekten Logik und ein formativer Usability-Test zur Evaluation der Bedienbarkeit.

3.6.1 Funktionale Tests

Die funktionalen Tests decken Backend-Logik und Frontend-Funktionalität ab (Authentifizierung, Agenda, Notenverwaltung, Lerntools, UI). Die vollständigen Testprotokolle der funktionalen Tests sind im Anhang dokumentiert.

3.6.2 Formativer Usability-Test

Um die Anonymität der Tester:innen zu gewährleisten, lassen wir die Namen weg. Insgesamt nahmen drei Personen am Test teil, alle Schüler:innen der

¹³Claude (Model Sonnet 4.5): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 10.11.2025. Antwort als Basis.

Kantonsschule. Die Tester:innen wurden gebeten, verschiedene Aufgaben in der App zu erledigen und danach qualitatives Feedback zu geben.

Testbericht Person 1 – 08.11.2025

Pros:

- Alles an einem Ort, was ich für mein Lernen brauche. Ich muss nicht 100 Tabs offen haben.
- Ein schönes User-Interface.

Cons:

- Es fehlt mir eine Introduktion in die App. So eine Anleitung, wie man es benutzt und was man machen kann.

Testbericht Person 2 – 08.11.2025

Pros:

- Simples und intuitives Design.
- Design ist sehr schön.

Cons:

- Ich fände es cool, zwischen Light und Dark Mode wechseln zu können, anstatt dass es nur automatisch geht.
- Ich fände es cool, wenn ihr irgendwie ein Bild von euch in der „Über Uns“ Page hättest.
- Ich würde auch gerne ein Scroll-To-Top Button bei den Lerntipps haben.
- Mir ist es unklar, wieso es bei den Lerntipps ein Section „Lernen“ und dann noch „Lernmethoden“ gibt. Es fühlt sich wie das Gleiche an.

Testbericht Person 3 – 09.11.2025

Pros:

- Schönes Design.
- Gute Lerntipps.

Cons:

- Keine, eigentlich, ich find das ganze super.

Ableitungen aus den Tests

Das Feedback aus dieser formativen Evaluation war entscheidend für die Finalisierung der App. Die wichtigsten „Cons“ wurden direkt adressiert und umgesetzt:

- (Person 1) Die fehlende Anleitung führte zur Erstellung der Hilfe-Seite (siehe Abschnitt 3.4.8).
- (Person 2) Der Wunsch nach einem manuellen Dark-Mode-Wechsel wurde in den Einstellungen implementiert (Abschnitt 3.4.7 und 3.5.2).
- (Person 2) Der Scroll-To-Top Button wurde auf der Seite „Lerntipps“ hinzugefügt (Abschnitt 3.4.6).
- (Person 2) Die unklare Sektionsaufteilung bei den Lerntipps wurde überprüft und die Kategorien wurden klarer benannt (z.B. „Allgemeine Lerntipps“ statt „Lernen“).

4. Reflexion zum

Entwicklungsprozess¹

Die Entwicklung der Kanti Koala Web-Applikation erstreckte sich über einen Zeitraum von mehreren Monaten und durchlief verschiedene Phasen, die sowohl Lernerfahrungen als auch Herausforderungen mit sich brachten.

4.1 Zeitmanagement und Projektplanung

Die ursprüngliche Zeitplanung erwies sich als zu optimistisch. Insbesondere die Recherche (Kapitel 2) war zuerst nur als kleiner Teil dieser Maturaarbeit gedacht, jedoch nahm sie schnell einen viel grösseren Umfang an, als wir gedacht hatten. Eigentlich wollten wir die Recherche in den ersten paar Monaten vorüber haben, jedoch war die letzte Analyse erst im Oktober fertig. Sie nahm viel mehr Zeit und Effort in gebrauch, als wir geplant haben und wir fielen so leicht hinter unseren Zeitplan.

Jedoch hat sich der Aufwand gelohnt. Die Recherche validiert unsere Web-Applikation wunderbar und zeigt auf, dass alle Probleme, aufgrund dessen wir die Web-Applikation entworfen haben, auch tatsächlich in unserer Zielgruppe existieren. Ebenso erwarben wir mithilfe dieser Recherche sehr vieles an nützlichen Wissen, Tipps und Erkenntnissen, sodass dies unser Produkt erheblich verbessert und uns solide Grundlagen gibt, um die Web-Applikation gut weiterzuentwickeln. Dies war, unserer Meinung nach, den extra Aufwand absolut wert. Wir lernten vieles darüber, wie man eine sinnvolle und ziel-orientierte Recherche durchführt und auch, was für ein Zeitaufwand dies sein kann.

Würden wir diese Recherche noch einmal durchführen, würden wir uns definitiv mehr darauf achten, dass genug Zeit dafür ausgelegt ist und dass uns gewisse Fehler, wie zum Beispiel dass die Interviews auf Mundart durchgeführt

¹Claude (Model Sonnet 4.5): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 10.11.2025. Antwort als Basis.

wurden, nicht mehr unterlaufen, da dies noch einmal viel mehr Aufwand beim transkribieren bedeutete. Somit sind wir, im grossen Ganzen, dennoch sehr zufrieden mit dem Verlauf der Recherche und der Ergebnisse, die sie uns geliefert hat.

4.2 Technische Herausforderungen

Die Implementierung des Lernzeitalgorithmus stellte ebenfalls eine grosse technische Herausforderung dar. Mehrere Iterationen waren notwendig, um einen funktionsfähigen Algorithmus zu entwickeln, der alle Anforderungen erfüllte:

- Die anfängliche Vorwärtsplanung (vom heutigen Datum zum Prüfungstermin) führte zu suboptimalen Ergebnissen, da Lernblöcke zu früh platziert wurden und nähere Prüfungen nicht ausreichend priorisiert wurden. Die Umstellung auf eine rückwärtsgerichtete Planung (vom Prüfungstermin zum aktuellen Datum) löste dieses Problem.
- Die Konfliktprüfung musste mehrfach überarbeitet werden, um zeitliche Überlappungen zuverlässig zu vermeiden. Die Einführung eines 30-minütigen Puffers vor und nach jedem Ereignis erwies sich als praktikable Lösung.
- Die Behandlung von Zeitzonen erwies sich als komplexer als erwartet. Man musste immer sicherstellen, dass alle Zeitangaben konsistent in der gleichen Zeitzone verarbeitet wurden, um Fehler bei der Planung zu vermeiden.

Diese iterative Entwicklung war lehrreich, verbrauchte aber erhebliche Entwicklungszeit. In zukünftigen Projekten würden wir versuchen, solche komplexen Algorithmen früher zu prototypisieren und zu testen, um technische Risiken besser einschätzen zu können.

4.3 Architekturentscheidungen

Die anfängliche Code-Struktur mit einer einzigen `app.py`-Datei erwies sich schnell als unübersichtlich. Die Umstrukturierung in ein modulares Python-Paket mit dem Application Factory Pattern war zeitaufwendig, verbesserte aber die Wartbarkeit erheblich. Diese Entscheidung hätte früher im Projekt getroffen werden sollen, um Refactoring-Aufwand zu minimieren.

Die Wahl von Flask als Framework war grundsätzlich richtig, da unsere Vorkenntnisse eine schnellere Entwicklung ermöglichten.

4.4 Usability-Testing

Der formative Usability-Test mit drei Personen lieferte wertvolles Feedback, das direkt umgesetzt werden konnte (Hilfe-Seite, manueller Dark-Mode-Wechsel, Scroll-To-Top-Button). Die Stichprobengrösse von drei Testern ist jedoch für eine umfassende Usability-Evaluation zu klein. Ein grösserer und diverserer Testerpool hätte möglicherweise weitere Schwachstellen identifiziert.

Positiv zu vermerken ist, dass die identifizierten Mängel zeitnah behoben werden konnten, was den iterativen Entwicklungsansatz bestätigte. Allerdings erfolgte der Test relativ spät im Projekt, wodurch grössere strukturelle Änderungen schwieriger umsetzbar gewesen wären, welche aber glücklicherweise nicht notwendig waren.

4.5 Sicherheitsimplementierung

Die Implementierung von Sicherheitsmassnahmen (CSRF-Schutz, Passwort-Hashing, HTTPS) erfolgte teilweise parallel zur Funktionsentwicklung. Rückblickend wäre es effizienter gewesen, diese Aspekte von Anfang an systematisch zu integrieren, anstatt sie nachträglich hinzuzufügen. Dies hätte Refactoring-Aufwand reduziert.

Die Entscheidung, einen eigenen CSRF-Decorator zu implementieren anstatt Flask-WTF zu verwenden, war aus Lernperspektive wertvoll, erhöhte

aber den Entwicklungsaufwand. Für ein produktives System wäre die Verwendung etablierter Bibliotheken möglicherweise vorzuziehen.

4.6 Zusammenarbeit und Arbeitsteilung

Die Zusammenarbeit über Git und GitHub funktionierte grundsätzlich gut, erforderte aber Einarbeitungszeit. Merge-Konflikte traten gelegentlich auf, konnten aber durch regelmässige Synchronisation minimiert werden. Die Aufteilung der Arbeitspakete basierte auf individuellen Stärken, was die Effizienz steigerte. Dennoch zeigte sich, dass eine klarere Aufgabenverteilung und regelmässigere Abstimmungen den Fortschritt weiter hätten optimieren können.

4.7 Erkenntnisse für zukünftige Projekte

Aus dem Arbeitsprozess lassen sich folgende Erkenntnisse für zukünftige Projekte ableiten:

- Realistische Zeitplanung mit ausreichenden Puffern ist essentiell.
- Interviews sollten in der Zielsprache des Berichts geführt werden, um Transkriptionsaufwand zu reduzieren.
- Architekturentscheidungen (Modularisierung, Design Patterns) sollten früh getroffen werden.
- Sicherheitsaspekte sollten von Anfang an systematisch integriert werden.
- Usability-Tests sollten früher und mit grösserer Stichprobe durchgeführt werden.
- Regelmässige Code-Reviews und Refactoring-Phasen sollten eingeplant werden.

Trotz der genannten Herausforderungen konnten nicht nur alle Kernfunktionen erfolgreich implementiert werden, sondern auch weitere nützliche

Features. Die gewonnenen Erkenntnisse tragen zur persönlichen Entwicklung im Bereich Projektmanagement und Software-Engineering bei und bieten eine solide Grundlage für zukünftige Projekte.

5. Schlussfolgerung und Ausblick¹

5.1 Zusammenfassung der Arbeit

Diese Maturitätsarbeit hatte zum Ziel, eine webbasierte Applikation zu konzipieren und umzusetzen, die Schüler:innen der Kantonsschule Baden eine übersichtliche Organisation von Terminen, Aufgaben und Lernzeiten ermöglicht und Funktionen bereitstellt, die regelmässige Lerngewohnheiten unterstützen.

Die Arbeit gliederte sich in zwei Hauptphasen:

1. **Recherchephase (Kapitel 2):** Durch eine kombinierte Methodik aus Literaturstudie, Interviews mit zwei PPP-Lehrpersonen und einer Umfrage unter Schüler:innen der Kantonsschule Baden wurden die Bedürfnisse der Zielgruppe systematisch erhoben. Die Recherche identifizierte vier zentrale Themenbereiche (Lernverhalten, Pausen-, Stress- und Zeitmanagement) und lieferte konkrete Anforderungen für die Applikation.
2. **Entwicklungsphase (Kapitel 3):** Basierend auf den Rechercheergebnissen wurde eine Flask-basierte Web-Applikation mit folgenden Kernfunktionen implementiert:

¹Claude (Model Sonnet 4.5): „Korrigiere diesen Abschnitt - behalte den Text so gut bei wie möglich, aber korrigiere alle grammatischen Fehler und markiere die Korrekturen. Mach auch Vorschläge, wie man problematische Sätze besser formulieren könnte. Schau, dass es wissenschaftlich formuliert ist. [...].“, 10.11.2025. Antwort als Basis.

- Eine zentrale Agenda mit Import-/Export-Funktionalität für .ics-Dateien
- Ein Lernzeitalgorithmus (LZA), der auf Basis von Prioritätseinstellungen und Benutzerparametern algorithmisch Lernblöcke plant
- Eine Notenverwaltung
- Ein konfigurierbarer Pomodoro-Timer
- Lerntipps und tägliche Tipps, basierend auf den Rechercheergebnissen
- Ein Authentifizierungssystem
- Eine To-Do-Liste

5.2 Beantwortung der Fragestellung

Die zentrale Fragestellung lautete:

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

Diese Fragestellung kann auf technischer Ebene bejaht werden. Die entwickelte Applikation stellt die notwendigen funktionalen Strukturen bereit:

- **Organisation von Terminen, Aufgaben und Lernzeiten:**
 - Die zentrale Agenda ermöglicht die Erfassung aller relevanten Termine an einem Ort.
 - Der Import von Stundenplänen als .ics-Datei reduziert den manuellen Erfassungsaufwand erheblich.
 - Der Lernzeitalgorithmus generiert algorithmisch Lernblöcke basierend auf Prüfungsterminen und Prioritätseinstellungen, wodurch die Planung von Lernzeiten erheblich vereinfacht wird.

- Die Notenverwaltung bietet eine strukturierte Übersicht über schulische Leistungen.
- Die To-Do-Liste unterstützt die Verwaltung von Aufgaben ausserhalb der Agenda.

- **Unterstützung regelmässiger Lerngewohnheiten:**

- Der Lernzeitalgorithmus platziert Lernblöcke mit festen Start- und Endzeiten, was zeitlich abgegrenzte Lernphasen schafft.
- Der Pomodoro-Timer strukturiert Lernintervalle und Pausen, was die Etablierung regelmässiger Lernroutinen unterstützt.
- Tägliche Tipps liefern kontinuierliche Impulse zu verschiedenen Lernaspekten.
- Die Schnellansicht der heutigen Ereignisse auf dem Dashboard erinnert an bevorstehende Termine und allfällige Lernblöcke.

- **Einschränkungen der Beantwortung:**

Es ist jedoch wichtig zu betonen, dass diese Arbeit die Frage nur auf der Ebene der technischen Umsetzbarkeit und der grundlegenden Bedienbarkeit beantwortet. Folgende Aspekte wurden nicht untersucht:

- **Tatsächliche Nutzung:** Es liegen keine Daten über die Langzeitnutzung der Applikation vor. Ob Schüler:innen die App regelmässig verwenden würden, ist nicht belegt.
- **Verhaltensänderung:** Ob die Nutzung der App tatsächlich zu einer Verbesserung des Lernverhaltens führt (z.B. früherer Lernbeginn, regelmässigere Lernphasen, reduzierter Stress), wurde nicht empirisch überprüft.
- **Vergleichsstudien:** Ein Vergleich mit anderen Organisationsmethoden oder -apps wurde nicht durchgeführt.
- **Wirksamkeitsevaluation:** Eine systematische Evaluation der Wirksamkeit einzelner Features (z.B. des Lernzeitalgorithmus) auf das Lernverhalten steht aus.

Die durchgeführte Umfrage validierte das Vorhandensein der Probleme (Prokrastination, Stress, spätes Lernen) in der Zielgruppe und das grundsätzliche Interesse an einer solchen Applikation. Dies belegt jedoch nicht die tatsächliche Wirksamkeit der implementierten Lösung.

Der formative Usability-Test bestätigte die grundsätzliche Bedienbarkeit der Applikation, jedoch mit einer sehr kleinen Stichprobe. Eine umfassendere Usability-Evaluation mit grösserer und diverserer Nutzergruppe wäre für eine fundierte Aussage zur Gebrauchstauglichkeit notwendig.

Zusammenfassend lässt sich sagen: Die Fragestellung wurde hinsichtlich der technischen Konzeption und Umsetzung beantwortet. Die Applikation enthält die notwendigen Funktionen und wurde als grundsätzlich bedienbar evaluiert. Ob sie die angestrebten Ziele (bessere Organisation, verbesserte Lerngewohnheiten) in der tatsächlichen Nutzung erreicht, bleibt eine offene empirische Frage, die weiterführende Forschung erfordert.

5.3 Ausblick und weiterführende Forschung

Die vorliegende Arbeit bietet eine solide technische Grundlage, wirft aber gleichzeitig mehrere Fragen auf, die weiterführende Forschung erfordern:

5.3.1 Empirische Wirksamkeitsevaluation

Die wichtigste offene Frage ist die tatsächliche Wirksamkeit der Applikation in der Praxis:

- **Langzeit-Nutzungsstudie:** Eine Studie über mindestens ein Semester mit einer grösseren Nutzergruppe könnte erfassen, ob und wie regelmässig die App genutzt wird, welche Features am häufigsten verwendet werden und welche vernachlässigt werden.
- **Verhaltensänderung:** Vergleichsstudien (z.B. Kontrollgruppenvergleich) könnten untersuchen, ob die Nutzung der App tatsächlich zu früherem Lernbeginn, regelmässigeren Lernphasen oder reduziertem Stress führt.

- **Lernzeitalgorithmus-Evaluation:** Spezifische Untersuchung, ob und wie die algorithmisch generierten Lernblöcke genutzt werden. Werden sie eingehalten? Werden sie manuell angepasst? Korreliert ihre Nutzung mit Lernerfolg?

5.3.2 Funktionale Erweiterungen

Basierend auf den gewonnenen Erkenntnissen und dem Feedback bieten sich mehrere Erweiterungsmöglichkeiten an:

- **Dynamische Prioritätsanpassung:** Automatische Anpassung der Lernblockpriorität basierend auf Notenentwicklung oder verbleibender Zeit bis zur Prüfung.
- **Kollaborative Funktionen:** Möglichkeit zum Teilen von Lernplänen oder Tipps zwischen Nutzer:innen, Lerngruppen-Features.
- **Erweiterte Reminder-Funktionen:** Push-Benachrichtigungen für anstehende Lernblöcke oder Prüfungen (würde native App oder Web-Push erfordern).
- **KI-Integration:** Wie in den Interview-Empfehlungen vorgeschlagen, könnte eine Seite über den bewussten Umgang mit KI-Tools beim Lernen integriert werden.

5.3.3 Technische Weiterentwicklungen

- **Native App-Entwicklung:** Evaluation, ob eine native App für iOS/Android Vorteile bietet (Offline-Funktionalität, bessere Push-Notifications, App-Store-Präsenz).
- **Performance-Optimierung:** Bei steigender Nutzerzahl und Datenvolumen könnten Optimierungen notwendig werden.

5.3.4 Skalierung und Verbreitung

Man könnte die App, nach Bedarf, auch auf andere Schulen oder Bildungseinrichtungen ausweiten.

5.3.5 Offene Probleme

Mehrere identifizierte Probleme konnten im Rahmen dieser Arbeit nicht gelöst werden:

- **Motivations-Problematik:** Die App kann Tools bereitstellen, aber die intrinsische Motivation zum Lernen nicht erzeugen. Wie kann die App Anreize schaffen, sie tatsächlich zu nutzen?
- **Individuelle Unterschiede:** Die Recherche zeigte, dass Lernstrategien individuell sehr unterschiedlich sind. Wie kann die App personalisierbar bleiben, ohne überkomplex zu werden?

5.4 Schlussbemerkung

Diese Maturitätsarbeit hat gezeigt, dass die Konzeption und technische Umsetzung einer Lern- und Organisationsapplikation für Kantonsschüler:innen machbar ist. Die entwickelte Web-Applikation Kanti Koala bietet eine kohärente Lösung zur zentralen Verwaltung von Terminen, automatisierten Planung von Lernzeiten und Unterstützung strukturierter Lernphasen.

Die durchgeführte Recherche validierte die Relevanz der adressierten Probleme (Prokrastination, Stress, spätes Lernen) in der Zielgruppe und lieferte fundierte Anforderungen für die Implementierung. Die technische Umsetzung demonstriert die Funktionsfähigkeit der Kernkomponenten, insbesondere des Lernzeitalgorithmus als zentralem Alleinstellungsmerkmal.

Gleichzeitig wurde deutlich, dass die Frage nach der tatsächlichen Wirksamkeit einer solchen Applikation – also ob sie das Lernverhalten von Schüler:innen nachhaltig verbessert – über den Rahmen dieser Arbeit hinausgeht. Eine umfassende empirische Evaluation über einen längeren Zeitraum mit einer grösseren Nutzerbasis wäre notwendig, um diese Frage zu beantworten.

Die vorliegende Arbeit schafft somit eine belastbare Ausgangsbasis für weiterführende Forschung und Entwicklung. Sie demonstriert die technische Machbarkeit, identifiziert offene Fragen und bietet konkrete Ansatzpunkte für Verbesserungen und Erweiterungen.

Der Arbeitsprozess selbst war lehrreich und forderte sowohl in methodischer (Recherche, Interviews, Umfragen) als auch in technischer Hinsicht (Software-Architektur, Algorithmenentwicklung, Sicherheit). Die gewonnenen Erkenntnisse zu Projektplanung, kollaborativer Softwareentwicklung und wissenschaftlicher Arbeitsweise bieten eine wertvolle Grundlage für zukünftige Projekte und das weitere Studium.

Abschliessend lässt sich festhalten: Kanti Koala ist ein funktionsfähiges Produkt, der die identifizierten Bedürfnisse technisch adressiert. Ob er sein Potenzial zur Verbesserung von Lernorganisation und -gewohnheiten in der Praxis entfalten kann, bleibt eine spannende Frage für zukünftige Untersuchungen.

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.>
- 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>
- Datenschutzerklärungen im Internet. (o. D.). Eidgenössischer Datenschutz- und Öffentlichkeitsbeauftragter (EDÖB). Verfügbar 10. November 2025 unter <https://www.edoeb.admin.ch/de/datenschutzerklaerungen-im-internet>

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 Documentation. (o. D.). Pallets Projects. Verfügbar 7. November 2025 unter <https://flask.palletsprojects.com/>

Flask-Bcrypt. (o. D.). Read the Docs. Verfügbar 4. November 2025 unter <https://flask-bcrypt.readthedocs.io/en/1.0.1/>

Flask-Migrate Documentation. (o. D.). Read the Docs. Verfügbar 7. November 2025 unter <https://flask-migrate.readthedocs.io/>

Flask-SQLAlchemy Documentation. (o. D.). Pallets Projects. Verfügbar 7. November 2025 unter <https://flask-sqlalchemy.palletsprojects.com/>

Get started with Tailwind CSS. (o. D.). Tailwindcss. Verfügbar 8. November 2025 unter <https://tailwindcss.com/docs/installation/using-postcss>

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/>

icalendar – iCalendar parser/generator. (o. D.). Read the Docs. Verfügbar 7. November 2025 unter <https://icalendar.readthedocs.io/>

ItsDangerous Documentation. (o. D.). Pallets Projects. Verfügbar 7. November 2025 unter <https://itsdangerous.palletsprojects.com/>

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>
- Lofi Girl. (2022, 12. Juli). *Lofi hip hop radio beats to relax/study to [Video]*. YouTube. <https://www.youtube.com/watch?v=jfKfPfyJRdk>
- 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/>
- Resend Documentation*. (o. D.). Resend. Verfügbar 7. November 2025 unter <https://resend.com/docs>
- 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/>

- SQLAlchemy Documentation*. (o. D.). Verfügbar 7. November 2025 unter <https://www.sqlalchemy.org/>
- 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/>
- Weller, D. (o. D.). *Hilft Musik hören beim Lernen?* Barmer. Verfügbar 9. November 2025 unter <https://www.barmer.de/gesundheit-verstehen/wissen/alle-themen/hilft-musik-hoeren-beim-lernen--1134038>
- 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/>
- Wikipedia Contributors. (2019, 20. September). *Pomodoro Technique*. Wikimedia Foundation. Verfügbar 10. Oktober 2025 unter https://en.wikipedia.org/wiki/Pomodoro_Technique

Abbildungsverzeichnis

1	Logo der Kantonsschule Baden. Quelle: Wikipedia	1
1.1	Logo der Applikation KantiKoala. Eigene Darstellung, 22.05.2025.	8
2.1	Darstellung einer Likert-Frage zu den Gründen für Prüfungsstress. Eigener Screenshot von KantiKoala, 9.11.25 . . .	28
2.2	Demonstration des Branching-Features von Microsoft Forms in unserer Umfrage (Bild 1). Eigener Screenshot von KantiKoala, 9.11.25	29
2.3	Demonstration des Branching-Features von Microsoft Forms in unserer Umfrage (Bild 2). Eigener Screenshot von KantiKoala, 9.11.25	29
2.4	Pie-Chart der Antworten nach Mittelschule. Eigener Screenshot von KantiKoala, 9.11.25	33
2.5	Pie-Chart der Antworten nach Klasse. Eigener Screenshot von KantiKoala, 9.11.25	33
2.6	Pie-Chart für wie früh mit Lernen angefangen wird (Insgesamt & pro Stufe). Eigener Screenshot von KantiKoala, 9.11.25 . .	34
2.7	Ein Cluster aus den häufigsten Vorschlägen, was sich für eine bessere Lernzufriedenheit änder müsste. Eigener Screenshot von KantiKoala, 9.11.25	35
2.8	Pie-Chart für wie viele der Schüler oft Lernstress empfinden (Insgesamt & pro Stufe). Eigener Screenshot von KantiKoala, 9.11.25	36

2.9	Resultate einer Likert-Frage über wie häufig gewisse Gründe wahrgenommen werden. Eigener Screenshot von KantiKoala, 9.11.25	37
2.10	Pie-Chart der Antworten zu ob Schüler:innen regelmässig Pausen machen. Eigener Screenshot von KantiKoala, 9.11.25	38
2.11	Resultate einer Likert-Frage über wie oft prokrastiniert wird und . Eigener Screenshot von KantiKoala, 9.11.25	38
2.12	Pie-Chart der Interesse an einer Lernapp (Ja / Nein). Eigener Screenshot von KantiKoala, 9.11.25	39
3.1	Screenshot des Anmeldeprozesses. Eigener Screenshot von Kanti Koala, 08.11.2025.	53
3.2	Homescreen für angemeldete Nutzer:innen in Light und Dark Mode. Eigene Screenshots von KantiKoala, 09.11.2025.	55
3.3	Homescreen für nicht angemeldete Nutzer:innen im Dark Mode. Eigener Screenshot von KantiKoala, 09.11.2025.	55
3.4	Beispiel einer Agenda mit Lernblöcken der LZA. Eigener Screenshot von Kanti Koala, 31.10.2025.	57
3.5	Screenshot der Notenverwaltungs-Seite. Eigener Screenshot von Kanti Koala, 08.11.2025.	61
3.6	Screenshot der Lerntimer-Seite. Eigener Screenshot von Kanti Koala, 08.11.2025.	62
3.7	Koala Bullet Point für Lerntipps. Eigene Darstellung, 06.11.2025.	63
3.8	Screenshot der Lerntipps-Seite. Eigener Screenshot von Kanti Koala, 08.11.2025.	63
3.9	Screenshot der Einstellungen-Seite. Eigener Screenshot von Kanti Koala, 08.11.2025.	64
3.10	Screenshot des Easter-Eggs auf der Über uns-Seite. Eigener Screenshot von Kanti Koala, 09.11.2025.	65
3.11	Screenshot der Hilfe-Seite. Eigener Screenshot von Kanti Koala, 09.11.2025.	66
3.12	Screenshot der 404-Fehlerseite. Eigener Screenshot von Kanti Koala, 08.11.2025.	73

Anhang

- **Code:** Der vollständige Code der Kanti Koala Web-App ist auf GitHub verfügbar unter: <https://github.com/CoderAryanAnand/lernapp>. Der Code wird aber auch noch separat beigelegt.
- **KI-Nachweis**
- **Tests**
- **Unsere Umfrage**
- **Interviewfragebogen**
- **Interviewtranskripte**
- **Interview- und Umfrageanalysen**