



Zahlen, Sprachen, Erfolg:

Eine Analyse der Vokabelverschiebung mittels Markovketten in einem selbstprogrammierten Vokabeltrainer nach dem Leitner System für effektives Spaced Repetition-Training.

Facharbeit Qualifikationsphase

Lehrkraft:	Sven Kirchner, Henning Ainödhofer
Fach:	Mathematik, Informatik
Autor:	Fabian Skutta
Adresse:	Mühlenweg 23, 48324 Sendenhorst
Abgabedatum:	26.02.2024

Inhalt

1	Einleitung.....	2
2	Download.....	2
3	Einführung in das Leitner System.....	2
4	Vokabeltrainer – Vocabify.....	3
4.1	Lern-Benutzeroberfläche.....	3
4.2	Auswahl des Technologiestacks.....	4
4.3	Datenbank	4
4.3.1	Beispiel Daten.....	5
4.4	Datenbank-Kommunikation	5
4.5	Algorithmus für die Vokabel-Abfrage.....	6
4.5.1	Auswahl der nächsten Vokabel.....	6
4.5.2	Datenbank-Aktualisierung und visuelles Feedback	7
5	Einführung in Markov-Ketten.....	7
5.1	Die Markov-Eigenschaft	7
5.2	Herleitung anhand eines Baumdiagramms.....	7
5.2.1	Prozessdiagramme	8
5.2.2	Übergangsmatrix.....	8
6	Modellierung des Vokabeltrainer mittels Markov-Ketten.....	9
6.1	Modellierung der Fächer als Zustände.....	9
6.1.1	Modellierung einer gleichbleibenden Wahrscheinlichkeit	9
6.1.2	Modellierung einer kontinuierlich ansteigenden Wahrscheinlichkeit	9
6.2	Erstellen von Übergangsmatrizen.....	9
7	Simulation und Analyse.....	11
7.1	Verteilung nach 10 Durchgängen	11
7.2	Verteilung nach 100 Durchgängen	12
7.3	Grenzwert.....	12
8	Vergleich	13
8.1	Vergleich zwischen den beiden Modellannahmen	13
8.2	Vergleich mit dem Leitner-System.....	13
9	Auswertung und Erweiterung	14
9.1	Auswertung der Analyse mittels Markov-Ketten	14
9.2	Erweiterung des Vokabeltrainers.....	14
10	Abbildungsverzeichnis	15
11	Literaturverzeichnis.....	16
12	Versicherung	17

1 Einleitung

Diese Facharbeit beschäftigt sich mit der Implementierung des Leitner Systems in einem web-basierten Vokabeltrainer, welcher für effektives Spaced Repetition-Training programmiert wurde, sowie einer vereinfachten Modellierung und Analyse der Vokabelverschiebung mittels Markov-Ketten.

In der Schule spielt Vokabellernen eine entscheidende Rolle. Die Tage des simplen Karteikartensystems oder des stumpfen Auswendiglernens sind vorbei. Der Fortschritt der Technologie ermöglichen uns ein effizienteres Lernen. Hierfür gibt es zahlreiche Ansätze, aber das Leitner System hebt sich durch seine Effektivität und durch seine doch einfach zu verstehende Struktur hervor (Settles & Meeder, 2016).

Das Hauptziel dieser Facharbeit ist es, ein effektives Vokabeltraining durch die Implementierung des Leitner Systems zu ermöglichen. Dabei sollen die Vorteile der Frameworks Vue und Nuxt für eine benutzerfreundliche Oberfläche sowie die Vorteile von Supabase, ein Backend as a Service (BaaS) genutzt werden, um eine Web-Applikation mit einem möglichst modernen Front- und Backend zu erstellen. Wie effektiv ein solcher Vokabeltrainer wirklich ist, zeigt sich durch die mathematische Analyse der Übergangswahrscheinlichkeiten. Die Verwendung von Markov-Ketten bietet eine leistungsstarke mathematische Methode, um solche Verschiebungen zu modellieren, statistisch auszuwerten und letztendlich zu verstehen.

Diese Arbeit ist von Bedeutung, da sie nicht nur eine technologische Umsetzung präsentiert, sondern auch eine Analyse der Vokabelverschiebung bietet. Die gewonnenen Erkenntnisse sind von Relevanz für die Optimierung von Lernprozessen und können sowohl für Lehrer als auch für Schüler wertvolle Einblicke liefern, um das Vokabellernen effizienter zu gestalten.

2 Download

Das Projekt, das im Rahmen der Facharbeit entwickelt wurde, ist auf GitHub unter <https://github.com/fabianskutta/Vocabify> verfügbar. Eine Online-Version kann unter <https://vocabify.fabian.lol> aufgerufen werden.

Um eine lokale Entwicklungsumgebung zu starten, muss Node.js installiert werden. Anschließend öffnet man eine Konsole in dem Repository und installiert alle Dependencies mit dem Befehl `npm install`. Um den Entwicklungsserver zu starten, führt man `npm run dev` aus. Nun ist das Projekt unter `http://localhost:3000` erreichbar. Eine `.env`-Datei mit allen wichtigen Zugangsdaten ist in der öffentlichen Version nicht vorhanden. Es kann dann keine Verbindung zu der Datenbank aufgebaut werden.

3 Einführung in das Leitner System

Das Prinzip des Leitner Systems besteht darin, mehrere Boxen mit unterschiedlichen Übungsintervallen zu haben, welches mit der Funktion $I = F^2$ beschrieben werden kann. Dabei steht I für das Intervall und F für das Fach.

Alle Vokabeln beginnen erstmal in der 1-Tages-Box. Wenn der Schüler sich nach einem Tag an den richtigen Begriff erinnern kann, wird die Vokabel in die 2-Tages-Box hochgestuft. Wenn er sich 2 Tage später erneut richtig erinnern kann, wird die Vokabel in das 4-Tages-Kästchen verschoben. Umgekehrt wird die Vokabel bei einer falschen Antwort in ein das

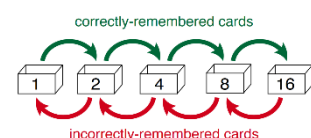


Abbildung 1 Das Leitner-System (Settles & Meeder, 2016)

nächstkleinere Intervallkästchen herabgestuft. Dieser Prozess kann so, wie in Abbildung 1 gezeigt, weitergeführt werden (Settles & Meeder, 2016).

Nach diesem Prinzip wurde folgender Vokabeltrainer programmiert:

4 Vokabeltrainer – Vocabify

Der Vokabeltrainer „Vocabify“ ist eine webbasierte Anwendung, welche auch als Webanwendung bezeichnet wird (Rouse, 2023). Benutzer können mit einem Computer oder einem Handy, welches mit dem Internet verbunden ist, mit einem Standardbrowser auf die Anwendung zugreifen. Ein Herunterladen der Anwendung ist nicht notwendig.

4.1 Lern-Benutzeroberfläche

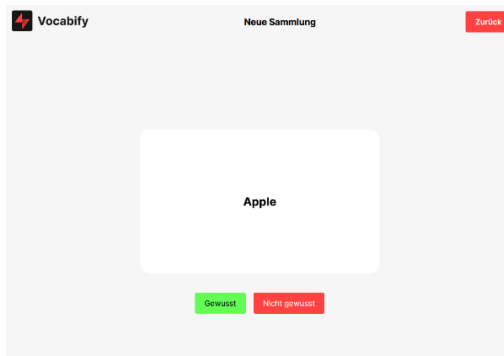


Abbildung 3 Kartenansicht (2024, [https://vocabify.fabian.lol/learn/\[id\]](https://vocabify.fabian.lol/learn/[id]))

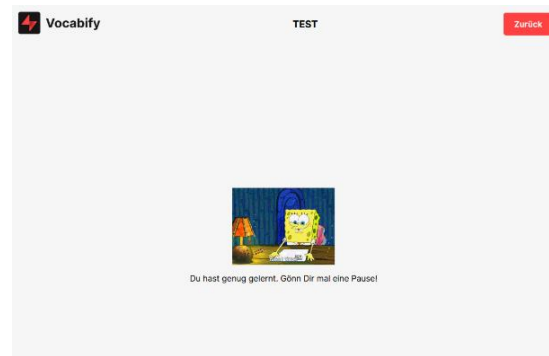


Abbildung 2 Abschlussnachricht (2024, [https://vocabify.fabian.lol/learn/\[id\]](https://vocabify.fabian.lol/learn/[id]))

Eine zentrale Komponente ist die Navigation, die durch eine übersichtliche Navigationsleiste erleichtert wird. Hier findet man einen Rückwärtsbutton für die einfache Rückkehr zu der vorherigen Ansicht.

Die Kartenansicht (Abb. 2) bildet das Herzstück der Anwendung. Hier werden die Vokabeln angezeigt, wobei auf der Vorderseite die Vokabel selbst und auf der Rückseite die Bedeutung angezeigt wird. Durch einen simplen Klick auf die Karte kann der Nutzer diese umdrehen, um die andere Seite zu sehen und sein Wissen zu überprüfen. Eine Animation beim Umblättern unterstützt das Umdrehen.

Unterhalb der Karte befinden sich zwei Buttons, die es dem Nutzer ermöglichen, seine Kenntnisse zu bewerten. Der grüne „Gewusst“ Button und der rote „Nicht gewusst“ Button ermöglichen eine schnelle Rückmeldung über den Wissenszustand der Vokabel und sind farblich deutlich voneinander abgesetzt, um eine intuitive Bedienung zu gewährleisten. Eine Konfetti-Animation unterstützt eine positive Rückmeldung.

Als Belohnung für das erfolgreiche Lernen aller Vokabeln wird dem Nutzer eine Abschlussnachricht präsentiert (Abb. 3), die ihm signalisiert, dass er genug gelernt hat und eine Pause machen kann. Ein GIF unterstützt diese Botschaft visuell und lädt den Nutzer dazu ein, sich eine wohlverdiente Auszeit zu nehmen.

Es wurde außerdem ein Dark- und Light-Mode implementiert, welcher sich an die Geräteeinstellung hält. Weiterhin wurde kein Frontend-CSS-Framework wie Bootstrap genutzt. Das ganze UI (User Interface) wurde selbst designend und schließlich in HTML und CSS implementiert.

4.2 Auswahl des Technologiestacks

Die Auswahl einer geeigneten Programmiersprache sowie eines passenden Frameworks bilden das Fundament für die Entwicklung einer Web-Anwendung (SoftGuide GmbH & Co. KG, 2024).

Für das Frontend verwendet die Anwendung das Vue.js-Framework, welches hauptsächlich auf die Kombination der Programmiersprachen JavaScript, TypeScript, HTML und CSS setzt. Vue.js wurde aufgrund seiner klaren Struktur und Flexibilität ausgewählt (Vue.js contributors, 2023). Diese Eigenschaften ermöglichen eine effiziente Entwicklung der Benutzeroberfläche, die entscheidend für eine möglichst einfache Programmierung sowie einer positiven Benutzererfahrung ist.

Um die Vorteile des serverseitigen Renderings (SSR) zu nutzen und gleichzeitig die Entwicklung zu vereinfachen, wurde Nuxt.js als Framework für Vue.js ausgewählt. Nuxt.js ermöglicht die serverseitige Vorrenderung der Seiten, was zu verbesserten SEO-Ergebnissen und einer besseren Website-Performance führt. Außerdem bietet Nuxt.js im Vergleich zu Web-Anwendungen, die ausschließlich auf HTML, CSS und JS setzen, weitere Vorteile wie ein vereinfachtes Routing der verschiedenen Seiten einer Website (Nuxt contributors, 2023).

Die Kombination von Vue.js und Nuxt.js schafft eine technologische Basis für den Vokabeltrainer. Diese Auswahl berücksichtigt nicht nur die Anforderungen der Anwendung, sondern auch moderne „Best Practices“ in der Webentwicklung, um eine effiziente, gut strukturierte und performante Lösung anzubieten (Mirzaei, 2023).

Für die Datenbankkomponente fiel die Entscheidung auf Supabase, eine Datenbanklösung, welche auf PostgreSQL basiert. Supabase ist ein BaaS und bietet nicht nur eine leistungsstarke Datenbankinfrastruktur, sondern auch eine benutzerfreundliche Oberfläche für die Verwaltung der Datenbanken. Außerdem integriert Supabase auch verschiedene Authentifizierungs- und Autorisierungsfunktionen, um eine sichere Benutzerauthentifizierung in der Anwendung bereitzustellen (Wemakefuture AG, 2024).

Insbesondere mit der Verwendung des entsprechenden Vue-Moduls, werden die genannten Vorteile im Vergleich zu einer herkömmlichen SQL-Datenbank deutlich.

4.3 Datenbank

Die Datenbank für den Vokabeltrainer besteht aus drei Haupttabellen: boxes, words und users.

Die boxes-Tabelle speichert die Vokabelboxen und beinhaltet die Spalte name für den Boxnamen, user_id für die eindeutige Benutzerzuordnung mit Verweis auf den Fremdschlüssel, created_at für den Erstellungszeitpunkt und id als eindeutige Kennung (uuid), auch Primärschlüssel genannt.

Die words-Tabelle speichert die Vokabeleinträge. Dabei beinhaltet created_at den Zeitpunkt der Vokabelerstellung, term die eigentliche Vokabel und definition die zugehörige Definition oder Übersetzung. Des Weiteren befindet sich in dieser Tabelle eine id als eindeutige Vokabelkennung (uuid), eine box_id zur Verknüpfung mit der zugehörigen Vokabelbox, das level für das Lernniveau von 1-16 und LastLearned als Zeitpunkt der letzten Lernaktivität.

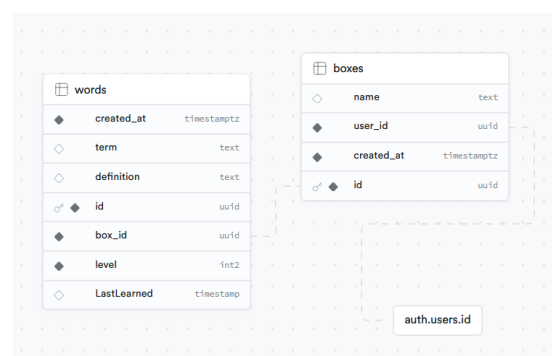


Abbildung 4 Datenbank Schema (2024, Schema Visualizer, supabase.com)

Die `users`-Tabelle ist bereit von Supabase implementiert und ist Teil der Authentication. Diese Datenbank enthält Informationen zu den Benutzern, darunter `eMail` für die E-Mail-Adresse, `created_at` für den Zeitpunkt der Benutzererstellung, `last_SignIn` für die letzte Anmeldung und `user_UID` als eindeutige Benutzerkennung. In der Abbildung 4 wird diese Datenbank nur durch die notwendige Beziehung `auth.users.id` angedeutet.

4.3.1 Beispiel Daten

Ein konkretes Beispiel verdeutlicht die Struktur:

<code>id</code> uuid	<code>user_id</code> uuid	<code>created_at</code> timestampz	<code>name</code> text
cc7d581a-a493-4a...	d00587e8-70a7-470d-bf26-0b687...	2023-10-30 22:30:54.503193+0	New Box
4c240275-ea7e-46...	bffeb6a0-eff3-42ed-9db8-557844...	2023-10-31 06:53:12.901206+0	New Box

Abbildung 5 boxes-Tabelle (2024, Table Editor, supabase.com)

<code>id</code> uuid	<code>created_at</code> timestampz	<code>box_id</code> uuid	<code>term</code> text	<code>definition</code> text	<code>level</code> int2	<code>LastLearned</code> timestamp
9d695ab7-bfba-43...	2024-02-20 14:33:07.942297+0	15ccb739-ac43-...	Apple	Apfel	1	2024-02-20 14:34:31.85
afcd8ba9-9e51-49...	2024-02-20 14:33:07.942297+0	15ccb739-ac43-...	Algorithm	Algorithmus	1	2024-02-20 14:34:38.252
3d14aabf-9371-46...	2024-02-20 14:33:07.942297+0	15ccb739-ac43-...	Sustainability	Nachhaltigkeit	1	2024-02-20 14:34:39.222

Abbildung 6 words-Tabelle (2024, Table Editor, supabase.com)

4.4 Datenbank-Kommunikation

Die Datenbank-Kommunikation verläuft über ein Vue-Modul (Supabase Client Library). Durch die Verwendung von beispielsweise `client.from('boxes').select('*').eq('id', id)` wird die Implementierung der Datenbankabfrage für den Entwickler erleichtert, da dieser sich nicht um die SQL-Abfrage kümmern muss. Der Code ist für Entwickler, die mit Vue.js vertraut sind außerdem leichter lesbar als eine komplexe SQL-Abfrage. Die Verwendung von Funktionen wie `eq()` (Equal) für Bedingungen verbessert die Lesbarkeit und strukturiert auch den Code übersichtlicher.

Die Supabase Client Library und auch das Supabase Studio, welches als Verwaltung der Datenbank fungiert, bauen eine Verbindung zu einem API-Gateway (Eine Programmierschnittstelle) auf, welches die Anfragen auf die anderen Systeme, wie die SQL-Datenbank Postgress und den Storage Provider AWS (Amazon) auf (Roos, 2022). Eine vereinfachte Darstellung dieser Kommunikation ist in Abbildung 7 dargestellt.

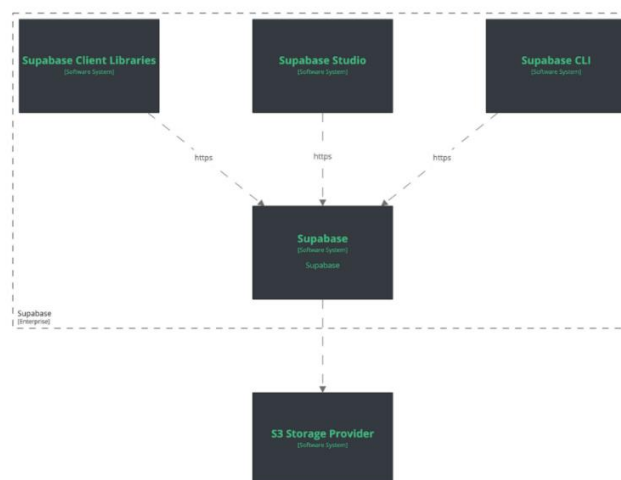


Abbildung 7 Supabase Tech-Stack (2022, Patrick Roos, <https://www.workingsoftware.dev/tech-stack-and-architecture-of-supabase>)

4.5 Algorithmus für die Vokabel-Abfrage

4.5.1 Auswahl der nächsten Vokabel

Für die Auswahl der angezeigten Vokabel wurde ein Algorithmus implementiert, welcher auf der Basis des Leitner Systems funktioniert:

Es werden alle Vokabeln aus der `words`-Tabelle selektiert, welche mit der ausgewählten `box_id` übereinstimmen (Abb. 8 Z. 2). Außerdem wird das aktuelle Datum auf der Konstanten `currentDate` gespeichert (Abb. 8 Z. 3).

Für jeden Wort-Datenpunkt, der in den selektierten Daten vorhanden ist, wird das Wort `level` definiert und folgende Logik angewendet (Abb. 8 Z. 5):

Wenn die Vokabel noch nie gelernt wurde und das Wort `level` somit `NULL` ist (nicht definiert), wird diese Vokabel direkt zurückgegeben (Abb. 8 Z. 8-10). Sollte das aktuelle Datum (`currentDate`) größer sein als das ausgerechnete nächste Fälligkeitsdatum (`nextDueDate`), wird diese Vokabel auch zurückgegeben (Abb. 8 Z. 18-20).

```
pages/learn/[id].vue
1 const { data: word, refresh: refreshWords } = await useAsyncData("words", async () => {
2   const { data } = await client.from("words").select("*").eq("box_id", id);
3   const currentDate = new Date();
4
5   for (const wordItem of data) {
6     const level = wordItem.level;
7
8     if (level === null) {
9       return wordItem;
10    }
11
12    const lastLearnedDate = new Date(wordItem.lastLearned);
13    const intervalDays = Math.pow(level, 2);
14
15    const nextDueDate = new Date(lastLearnedDate);
16    nextDueDate.setDate(lastLearnedDate.getDate() + intervalDays);
17
18    if (currentDate > nextDueDate) {
19      return wordItem;
20    }
21  }
22 },
23 );
```

Abbildung 8 Ausschnitt aus dem script-Teil der Datei `pages/learn/[id].vue` (2024, Erstellt mit <https://codeimg.io/>)

Um das nächste Fälligkeitsdatum auszurechnen, wird der Zeitpunkt der letzten Lernaktivität als Datum definiert und das passende Intervall (`intervalDays`) des jeweiligen Levels nach dem Leitner System berechnet. Dazu wird das Level (klassisch: das Fach) mithilfe von `Math.pow(level, 2)` quadriert und anschließend mit dem Zeitpunkt der letzten Lernaktivität addiert (Abb. 8 Z. 13).

Beispielsweise wurde eine Vokabel mit dem Level 2 am 2024-02-02 zuletzt gelernt. Das Level wird quadriert ($2^2 = 4$) und mit dem Zeitpunkt der letzten Lernaktivität addiert:

$$2024 - 02 - 02 + 4 = 2024 - 02 - 06$$

Die Vokabel ist also überfällig, wenn das aktuelle Datum älter ist als der 2024-02-06.

4.5.2 Datenbank-Aktualisierung und visuelles Feedback

In dem UI (User Interface) des Vokabeltrainers gibt es zwei mögliche Optionen in dem Lern-Interface: Gewusst und nicht gewusst. Diese Buttons führen bei einem Klick entweder die Funktion `yes(id, level)` oder `no(id, level)` aus. Dabei wird die `id` der aktuellen Vokabel sowie das aktuelle `level` übergeben. Sollte die `yes`-Funktion ausgeführt werden, wird zusätzlich auch eine Konfetti-Animation abgespielt, welche durch die Funktion `confetti()` aus dem Modul `canvas-confetti` bereitgestellt wird (Abb. 9 Z. 12-18).

Durch Aufrufen der Funktion `updateLevel(id, levelChange)` (Abb. 9 Z. 1) mit dem entsprechenden neuen Level-Wert und der Vokabel-Id wird nun der Eintrag in der Datenbank aktualisiert: Als erstes wird überprüft, ob das neue übergebende Level der Vokabel in dem vorgeschriebenen Bereich zwischen 1 und 16 liegt. Ist dies nicht der Fall (bei einem Level von 0 oder 17), wird auf das Minimum bzw. Maximum zurückgegriffen (Abb. 9 Z. 2).

Nun folgt die Datenbank-Aktualisierung, welche durch die bereits beschriebene Supabase API erfolgt. Hierbei wird in der `words` Tabelle das `level` auf das neue Level (`newLevel`) geändert und auch der Zeitpunkt der letzten Lernaktivität (`LastLearned`) mit dem aktuellen Datum übergeben. Damit diese Änderung nur bei der passenden Vokabel ausgeführt wird, sucht die `eq()` Funktion die Vokabel raus, welche die übergebende Vokabel-Id besitzt (Abb. 9 Z. 3-7).

Anschließend wird die nächste Vokabel mit der `refreshWords()`-Funktion aus der Datenbank geladen und die CSS Klasse `isFlipped` aus dem `card-inner` div (Content Division element) Container entfernt, damit in jeden Fall die Definition einer Vokabel nicht sichtbar ist bzw. auf der Rückseite der Karte ist (Abb. 9 Z. 8-9).

```
pages/learn/[id].vue
1 async function updateLevel(id, levelChange) {
2   const newLevel = Math.min(Math.max(1, levelChange), 16);
3   const { data, error } = await client
4     .from('words')
5     .update({ level: newLevel, LastLearned: new Date() })
6     .eq('id', id)
7     .select();
8   refreshWords();
9   isFlipped.value = false;
10 }
11
12 async function yes(id, level) {
13   await updateLevel(id, level + 1);
14   confetti();
15 }
16
17 async function no(id, level) {
18   await updateLevel(id, level - 1);
19 }
```

Abbildung 9 Ausschnitt aus dem script-Teil der Datei `pages/learn/[id].vue` (Erstellt mit <https://codeimg.io/>)

5 Einführung in Markov-Ketten

Um die Modellierung und Analyse des Vokabeltrainer mittels Markov-Ketten zu verstehen, folgt eine Herleitung zu Markov-Ketten anhand eines Baumdiagramms.

5.1 Die Markov-Eigenschaft

Um eine Markov-Kette zu modellieren, muss ein Zufallsexperiment die Markov-Eigenschaft besitzen. Sie besagt, dass das zukünftige Verhalten eines Systems oder Prozesses nur vom aktuellen Zustand abhängt und nicht von der gesamten Vergangenheit. Das bedeutet, dass, wenn man den aktuellen Zustand kennt, die Vergangenheit keine zusätzliche Information für die Vorhersage der Zukunft liefert (Krühler, 2013).

5.2 Herleitung anhand eines Baumdiagramms

In einem Baumdiagramm bleiben die Übergangswahrscheinlichkeiten von einer zur nächsten Stufe meistens gleich. Nach einer bestimmten Anzahl von Zufallsvorgängen, bieten Baumdiagramme keinen übersichtlichen Weg, Zufallsvorgänge darzustellen, da eine Beschreibung

durch Baumdiagramme zu umfangreich wäre. Bei einem endlichen Zufallsvorgang kann man solche Vorgänge in einem Prozessdiagramm darstellen.

5.2.1 Prozessdiagramme

Bei einem Stochastischen Prozess lässt sich die gegebene Situation in verschiedene Zustände modellieren. Die Summe der Wahrscheinlichkeiten der abgehenden Pfeile (bei jedem Zustand) muss dabei 1 sein. Die Randzustände erkennt man an einem Ringpfeil, welcher wieder auf sich selbst verweist.

Wenn diese Zustände keinen Rückverweis aufweisen, sind dies Endzustände, welche man nicht mehr verlassen kann. Man nennt diese Zustände auch absorbierend. Sollten Rückverweise vorhanden sein, sind die Randzustände also nicht absorbierend. Die anderen Zustände bezeichnet man als innere Zustände. In der Abbildung 10 (Fig. 2) wird ein absorbierender Prozess dargestellt (Brand & et al., 2015).

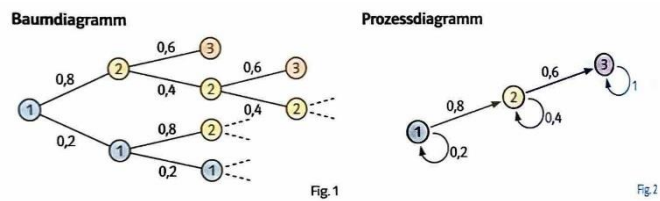


Abbildung 10 Baumdiagramm & Prozessdiagramm (S. 352, Lambacher Schweizer Mathematik Qualifikationsphase - Ausgabe Nordrhein-Westfalen, 1. Auflage 2015, Ernst Klett Verlag)

5.2.2 Übergangsmatrix

Mithilfe Stochastischer Matrizen kann man die Übergänge aus dem Prozessdiagramm wie folgt in einer beispielhaften 3×3 Übergangsmatrix nach Abbildung 10 darstellen. Hierbei steht M für die Übergangsmatrix und V_n für den Durchlauf n . Die blauen Notizen dienen zum besseren Verständnis und sind nicht Teil einer Übergangsmatrix bzw. eines Vektors.

von nach		1	2	3
$M =$	1	$\begin{pmatrix} 0,2 & 0 & 0 \\ 0,8 & 0,4 & 0 \\ 0 & 0,6 & 1 \end{pmatrix}$		
	2			
	3			
Spaltensumme		1	1	1

Nun lässt sich der 1. Zustand (V_1) als Vektor notieren. Beispielsweise mit einem Wert von 1 in dem 1. Fach:

$$V_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}$$

Um die Folgezustände zu berechnen, multipliziert man V_1 mit der Übergangsmatrix:

$$V_2 = M \times V_1 = \begin{pmatrix} 0,2 & 0 & 0 \\ 0,8 & 0,4 & 0 \\ 0 & 0,6 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Dabei geht man nach dem Prinzip „Zeile mal Spalte“ vor. Die erste Zeile der Matrix multipliziert man mit dem Spaltenvektor der zweiten Matrix: $0,2 \times 1 + 0 \times 0 + 0 \times 0 = 0,2$

Die zweite Zeile der ersten Matrix multipliziert man mit dem Spaltenvektor der zweiten Matrix: $0,8 \times 1 + 0,4 \times 0 + 0 \times 0 = 0,8$

Die dritte Zeile der ersten Matrix multipliziert man mit dem Spaltenvektor der zweiten Matrix: $0 \times 1 + 0,6 \times 0 + 1 \times 0 = 0$

$$\text{Somit ergibt } V_2 = M \times V_1 = \begin{pmatrix} 0,2 \\ 0,8 \\ 0 \end{pmatrix}$$

6 Modellierung des Vokabeltrainer mittels Markov-Ketten

Bei der folgenden Modellierung handelt es sich um eine vereinfachte Darstellung des Vokabeltrainers, welche den Zeitpunkt der letzten Lernaktivität mathematisch nicht berücksichtigt. Die Werte-Tage einer Vokabel werden somit nicht als Zustände modelliert, um die Analyse verständlicher zu halten. Die Analyseergebnisse können trotzdem zum besseren Verständnis des Vokabeltrainers genutzt werden.

6.1 Modellierung der Fächer als Zustände

In dem programmierten Vokabeltrainer gibt es 16 Fächer, welche technisch als Level implementiert wurden. Diese lassen sich wie folgt in einem Prozessdiagramm darstellen. Anders als in Abbildung 10, werden hier nicht-absorbierende Prozesse dargestellt, da sich die Vokabeln auch Rückwärts bewegen können. Es wurden zwei verschiedene Modellannahmen getätigt.

6.1.1 Modellierung einer gleichbleibenden Wahrscheinlichkeit

Bei der ersten Modellannahme wird davon ausgegangen, dass man eine Vokabel aus Fach 1 zu 70% richtig beantwortet und zu 30% falsch. Die Übergangswahrscheinlichkeiten ändern sich in diesem Modell nicht und bleiben bei jedem Zustand identisch. Das bedeutet, dass die Wahrscheinlichkeit, eine Vokabel von einem Fach in ein anderes zu bewegen, unabhängig von der Anzahl der Versuche oder dem Lernfortschritt ist. Das Prozessdiagramm in Abbildung 11 beschreibt dieses Modell.

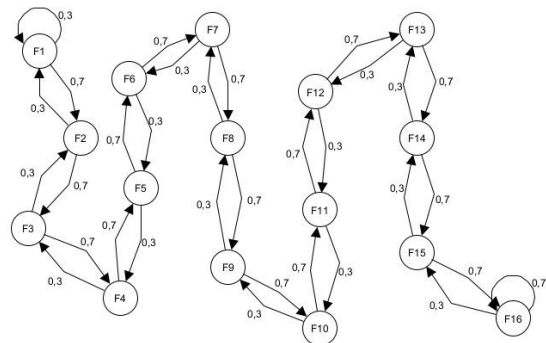


Abbildung 11 Prozessdiagramm bei einer gleichbleibenden Wahrscheinlichkeit (2024, Eigenwerk)

6.1.2 Modellierung einer kontinuierlich ansteigenden Wahrscheinlichkeit

In der zweiten Modellannahme wird davon ausgegangen, dass man eine Vokabel aus Fach 1 zu 20% richtig beantwortet und zu 80% falsch, da man sie zum ersten Mal sieht. Die Wahrscheinlichkeit eine Vokabel richtig zu beantworten, steigt nach jedem Zustand um 5% an. Die Wahrscheinlichkeit eine Vokabel falsch zu beantworten, sinkt nach jedem Zustand um 5%. Dies ist eine Modellierung, in der die Lernenden mit zunehmender Erfahrung besser werden. Dieses Modell wird durch das Prozessdiagramm in Abbildung 12 beschrieben.

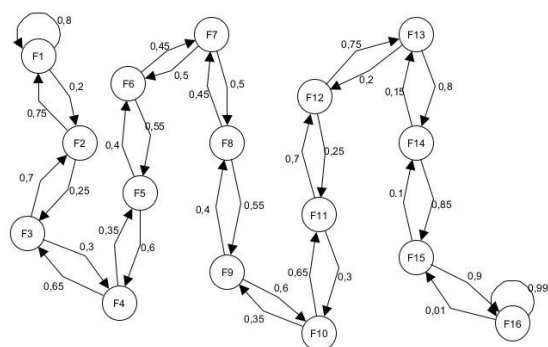


Abbildung 12 Prozessdiagramm bei einer kontinuierlich ansteigenden Wahrscheinlichkeit (2024, Eigenwerk)

6.2 Erstellen von Übergangsmatrizen

Der Prozess bei einer gleichbleibenden Wahrscheinlichkeit (7.1.1) wird fortlaufend auch G genannt. Der Prozess einer kontinuierlich ansteigenden Wahrscheinlichkeit (7.1.2) K.

Mit den Übergangswahrscheinlichkeiten aus den beiden Prozessdiagrammen in Abbildung 9 und 10 wird jeweils eine 16x16 Matrix erstellt. Die Spaltensumme beträgt immer 1 (100%).

[illegible]

[illegible]

Zum Start wurde eine Anfangsverteilung von 658 Vokabeln in F1 gewählt und als Vektor dargestellt. Diese Menge kommt ungefähr in einem Englischbuch vor (Rau, 2022).

[illegible]

7 Simulation und Analyse

Nun werden die beiden Modellannahmen analysiert, um die Verteilung der Vokabeln in den verschiedenen Fächern über die Zeit zu verstehen. Dies ermöglicht Rückschlüsse auf die Modellannahmen zu ziehen und potenzielle Verbesserungen zu identifizieren.

Für die weitere Analyse wird neben einem GTR hauptsächlich ein Python-Script verwendet, welches auch eine bessere Visualisierung der Ergebnisse bietet. Dieses ist ebenfalls in dem Github-Repository zu finden.

7.1 Verteilung nach 10 Durchgängen

$$G_{10} = M_g^{10} \times V_0 \approx \begin{pmatrix} 26,18 \\ 34,75 \\ 81,09 \\ 57,52 \\ 134,21 \\ 57,12 \\ 133,28 \\ 32,19 \\ 75,11 \\ 7,97 \\ 18,59 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad K_{10} = M_k^{10} \times V_0 \approx \begin{pmatrix} 466,35 \\ 121,23 \\ 41,31 \\ 15,57 \\ 7,54 \\ 3,02 \\ 1,81 \\ 0,6 \\ 0,42 \\ 0,08 \\ 0,07 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Der 10. Durchgang gibt einen Einblick in die Verteilung der Vokabeln nach 10 Lernzyklen gemäß den beiden Modellen (G und K).

Diese Ergebnisse zeigen unterschiedliche Verteilungen der Vokabeln zwischen den beiden Modellen. Während sich die Vokabeln im Modell mit gleichbleibenden Wahrscheinlichkeiten über die ersten Fächer 1 bis 11 verteilen, konzentrieren sich im Modell mit kontinuierlich ansteigenden Wahrscheinlichkeiten die Vokabeln auf die ersten Fächer, wobei die Anzahl stark abnimmt, je höher das Fach ist.

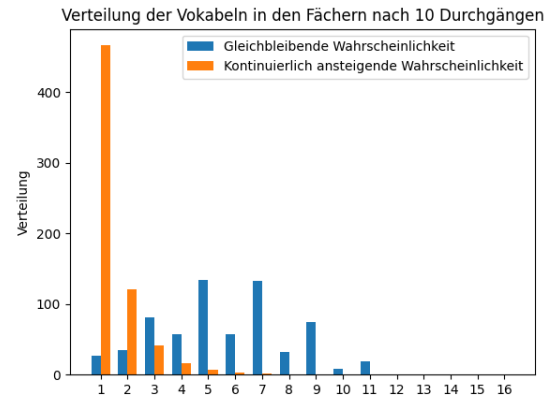


Abbildung 13 Verteilung der Vokabeln in den Fächern nach 10 Durchgängen (2024, Eigenwerk)

7.2 Verteilung nach 100 Durchgängen

$$G_{100} = M_g^{100} \times V_0 \approx \begin{pmatrix} 0 \\ 0 \\ 0,01 \\ 0,01 \\ 0,02 \\ 0,04 \\ 0,09 \\ 0,2 \\ 0,44 \\ 1,03 \\ 2,35 \\ 5,49 \\ 29,66 \\ 69,05 \\ 161,11 \\ 375,8 \end{pmatrix} \quad K_{100} = M_k^{100} \times V_0 \approx \begin{pmatrix} 415,73 \\ 110,28 \\ 38,61 \\ 16,91 \\ 8,86 \\ 5,33 \\ 3,55 \\ 2,56 \\ 1,96 \\ 1,57 \\ 1,29 \\ 1,1 \\ 0,96 \\ 0,89 \\ 1,22 \\ 47,19 \end{pmatrix}$$

Im Modell mit gleichbleibenden Wahrscheinlichkeiten (G) gibt es eine allmähliche Zunahme der Anzahl der Vokabeln ab dem 9. Fach, gefolgt von einem Plateau und schließlich einem rapiden Anstieg in den letzten Fächern. Dies deutet darauf hin, dass die Vokabeln mit der Zeit in höhere Fächer gelangen.

Im Modell mit kontinuierlich ansteigenden Wahrscheinlichkeiten (K) konzentrieren sich die meisten Vokabeln weiterhin auf die ersten Fächer, aber im Vergleich zu G steigt die Anzahl der Vokabeln in den höheren Fächern langsamer an. Dies bedeutet, dass das Modell eine weniger abrupte Zunahme in den höheren Fächern aufweist.

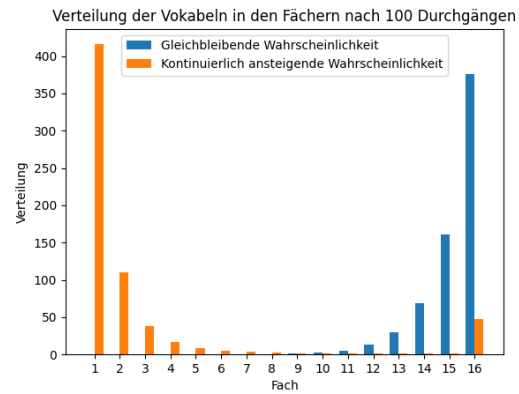


Abbildung 14 Verteilung der Vokabeln in den Fächern nach 100 Durchgängen (2024, Eigenwerk)

7.3 Grenzwert

$$G_{10000} = M_g^{10000} \times V_0 \approx \begin{pmatrix} 0 \\ 0 \\ 0,01 \\ 0,01 \\ 0,03 \\ 0,08 \\ 0,18 \\ 0,43 \\ 1 \\ 2,33 \\ 5,44 \\ 12,68 \\ 29,6 \\ 69,06 \\ 161,14 \\ 376 \end{pmatrix} \quad K_{10000} = M_k^{10000} \times V_0 \approx \begin{pmatrix} 0,17 \\ 0,05 \\ 0,02 \\ 0,01 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0,01 \\ 0,02 \\ 0,04 \\ 0,16 \\ 0,85 \\ 7,22 \\ 649,45 \end{pmatrix}$$

Diese Verteilungen zeigen den Grenzwert nach einer sehr großen Anzahl von Lernzyklen. In diesem Fall nähern sich die Verteilungen der Vokabeln in beiden Modellen ihren stabilen Zuständen an. Die Anzahl der Vokabeln nehmen in jedem Fach nicht mehr signifikant zu oder ab, sondern bleiben konstant.

Im Modell mit gleichbleibenden Wahrscheinlichkeiten (G) gibt es eine kontinuierliche Zunahme der Anzahl der Vokabeln ab dem 10. Fach, gefolgt von einem Plateau und schließlich einem Anstieg in den letzten Fächern. Dies deutet darauf hin, dass die Vokabeln mit der Zeit in höhere Fächer gelangen, aber der Anstieg immer noch relativ moderat bleibt. Es gibt immer wieder Vokabeln, die ein Fach absteigen.

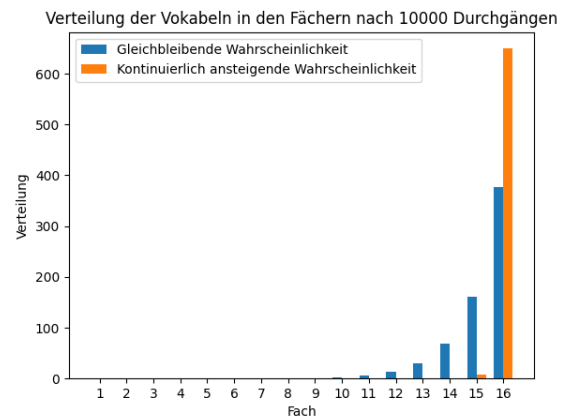


Abbildung 15 Verteilung der Vokabeln in den Fächern nach 10000 Durchgängen (2024, Eigenwerk)

Im Modell mit kontinuierlich ansteigenden Wahrscheinlichkeiten (K) zeigt sich eine stark ausgeprägte Konzentration der Vokabeln auf das letzte Fach. Dies deutet darauf hin, dass Vokabeln, die das letzte Fach erreichen, dieses kaum wieder verlassen.

8 Vergleich

8.1 Vergleich zwischen den beiden Modellannahmen

Die beiden Modelle zeigen über die Durchgänge hinweg unterschiedliche Entwicklungen in der Verteilung der Vokabeln über die Zeit.

Das Modell mit gleichbleibenden Wahrscheinlichkeiten zeigt eine allmähliche Zunahme der Vokabeln in höhere Fächer. Das Modell mit kontinuierlich ansteigenden Wahrscheinlichkeiten konzentriert sich am Anfang stärker auf niedrigere Fächer, wobei die Vokabeln langsamer in höhere Fächer aufsteigen, wo sie schließlich in den höchsten Fächern bleiben.

Modell G stellt einen schnellen Lernerfolg dar, doch die Vokabeln kommen jedoch nicht in das Langzeitgedächtnis. Es werden immer wieder Vokabeln vergessen. In Modell K dauert der Lernerfolg viel länger, doch die Vokabeln bleiben im Langzeitgedächtnis.

8.2 Vergleich mit dem Leitner-System

Insgesamt scheint das Modell mit kontinuierlich ansteigenden Wahrscheinlichkeiten (K) besser zu dem Leitner-System zu passen, da es die Konzepte des Intervall-basierten Lernens und des Verschiebens von Vokabeln basierend auf ihrem Lernfortschritt vergleichbar abbildet.

Im Leitner-System werden Vokabeln basierend auf ihrem Erinnerungsintervall in verschiedene Boxen verschoben. Dies spiegelt sich in dem kontinuierlichen Anstieg der Wahrscheinlichkeiten wieder, da die Wahrscheinlichkeit für das Verschieben in ein höheres Fach im Verlauf des Lernprozesses zunimmt.

Die Funktion $I = F^2$ beschreibt ein exponentielles Wachstum der Intervalle von jedem Fach. Dies ähnelt der Idee, dass die Wahrscheinlichkeiten im Modell K kontinuierlich ansteigen, was zu einem langsameren Wachstum der Vokabelverteilung in höhere Fächer führt.

9 Auswertung und Erweiterung

9.1 Auswertung der Analyse mittels Markov-Ketten

Die Analyse der beiden Modellannahmen hat interessante Einblicke in die Verteilung der Vokabeln ermöglicht. Es wurde festgestellt, dass das Modell mit kontinuierlich ansteigenden Wahrscheinlichkeiten (Modell K) dazu führt, dass die Vokabeln im Langzeitgedächtnis bleiben, während das Modell mit gleichbleibenden Wahrscheinlichkeiten (Modell G) eher zu einem schnelleren kurzfristigen Lernerfolg führt, jedoch mit einer höheren Vergessens-Rate.

Es ist jedoch anzumerken, dass die gewählten Wahrscheinlichkeiten in Modell K möglicherweise nicht optimal sind. Eine Verbesserung dieses Modells könnte durch eine bessere Auswahl der Übergangswahrscheinlichkeiten erfolgen, die eine realistischere Abbildung des Lernprozesses darstellen. Hierbei könnten Studien über das Lernverhalten von Personen genutzt werden, um die Wahrscheinlichkeiten genauer zu bestimmen und anzupassen.

Insgesamt bieten die vorgestellten Modelle eine solide Grundlage für die Analyse der Vokabelverschiebung, jedoch gibt es noch Raum für weitere Forschung und Entwicklung, um das Modell K zu verbessern.

9.2 Erweiterung des Vokabeltrainers

Eine der möglichen Erweiterungen ist die Einbeziehung von multimedialen Inhalten wie Bilder oder Audioaufnahmen. Sie können die Effektivität des Lernens steigern, indem verschiedene Sinnesmodalitäten angesprochen und die Verknüpfung von Begriffen mit konkreten Situationen gefördert wird (Stork, 2006).

Des Weiteren können Interaktive Übungen und Spiele das Lernen zusätzlich unterhaltsam gestalten und die Motivation der Benutzer steigern. Durch Quiz-Spiele, Lückentextübungen oder Drag-and-Drop-Übungen, welche beispielsweise bei dem Vokabeltrainer „Quizlet“ zu finden sind, wird das Lernen zu einer interaktiven und spannenden Erfahrung (Sanosi, 2018).

Außerdem kann die Verfolgung des Lernfortschritts mittels Statistiken, Durchschnittswerten und der Implementierung eines Belohnungssystems die Benutzer motivieren und ihnen helfen, ihre Ziele zu erreichen. Eine visuelle Darstellung des Fortschritts könnte es den Benutzern ermöglichen, ihre Entwicklung zu verfolgen und ihre Erfolge zu feiern.

10 Abbildungsverzeichnis

Abbildung 1 Das Leitner-System (Settles & Meeder, 2016)	2
Abbildung 2 Abschlussnachricht (2024, https://vocabify.fabian.lol/learn/[id])	3
Abbildung 3 Kartenansicht (2024, https://vocabify.fabian.lol/learn/[id])	3
Abbildung 4 Datenbank Schema (2024, Schema Visualizer, supabase.com)	4
Abbildung 5 boxes-Tabelle (2024, Table Editor, supabase.com)	5
Abbildung 6 words-Tabelle (2024, Table Editor, supabase.com)	5
Abbildung 7 Supabase Tech-Stack (2022, Patrick Roos, https://www.workingsoftware.dev/tech-stack-and-architecture-of-supabase)	5
Abbildung 8 Ausschnitt aus dem script-Teil der Datei pa-ges/learn/[id].vue (2024, Erstellt mit https://codeimg.io/)	6
Abbildung 9 Ausschnitt aus dem script-Teil der Datei pages/learn/[id].vue (Erstellt mit https://codeimg.io/)	7
Abbildung 10 Baumdiagramm & Prozessdiagramm (S. 352, Lambacher Schweizer Mathematik Qualifikationsphase - Ausgabe Nordrhein-Westfalen, 1. Auflage 2015, Ernst Klett Verlag)	8
Abbildung 11 Prozessdiagramm bei einer gleichbleibenden Wahrscheinlichkeit (2024, Eigenwerk)	9
Abbildung 12 Prozessdiagramm bei einer kontinuierlich ansteigenden Wahrscheinlichkeit (2024, Eigenwerk)	9
Abbildung 13 Verteilung der Vokabeln in den Fächern nach 10 Durchgängen (2024, Eigenwerk)	11
Abbildung 14 Verteilung der Vokabeln in den Fächern nach 100 Durchgängen (2024, Eigenwerk)	12
Abbildung 15 Verteilung der Vokabeln in den Fächern nach 10000 Durchgängen (2024, Eigenwerk)	13

11 Literaturverzeichnis

- Brand, D., & et al. (2015). *Lambacher Schweizer Mathematik Qualifikationsphase - Ausgabe Nordrhein-Westfalen, 1. Auflage*. Stuttgart, Leipzig: Ernst Klett Verlag.
- Krühler, M. (24. Oktober 2013). *Endliche Markov-Ketten*. Von Universität Münster: <https://www.uni-muenster.de/Stochastik/lehre/WS1314/BachelorWT/Daten/Kruehler.pdf> abgerufen
- Mirzaei, A. (08. Oktober 2023). *Nuxt.js Structure Best Practices: Building Robust Vue.js Applications*. Von Medium: https://medium.com/@ali_mirzaei/nuxt-js-structure-best-practices-building-robust-vue-js-applications-625ed4e782bb abgerufen
- Nuxt contributors. (18. Oktober 2023). *Introduction · Get Started with Nuxt*. Von Nuxt: <https://nuxt.com/docs/getting-started/introduction> abgerufen
- Rau, T. (12. August 2022). *Englisch-Vokabeln laut Schulbuch*. Von Herr Rau: <https://www.herr-rau.de/wordpress/2022/08/englisch-vokabeln-laut-schulbuch.htm> abgerufen
- Roos, P. (26. Juni 2022). *Under the hood: Architecture and Technology Stack of Supabase*. Von workingsoftware.dev: <https://www.workingsoftware.dev/tech-stack-and-architecture-of-supabase/> abgerufen
- Rouse, M. (16. Oktober 2023). *Webbasierte Anwendung*. Von Techopedia: <https://www.techopedia.com/de/definition/webbasierte-anwendung> abgerufen
- Sanosi, A. B. (04. August 2018). *The Effect of Quizlet on Vocabulary Acquisition*. Von ResearchGate GmbH: https://www.researchgate.net/profile/Abdulaziz-Sanosi/publication/327108959_The_Effect_of_Quizlet_on_Vocabulary_Acquisition/links/5b7a8aeb92851c1e12219602/The-Effect-of-Quizlet-on-Vocabulary-Acquisition.pdf abgerufen
- Settles, B., & Meeder, B. (7-12. August 2016). *A Trainable Spaced Repetition Model for Language Learning*. Von ACL Anthology: <https://aclanthology.org/P16-1174.pdf> abgerufen
- SoftGuide GmbH & Co. KG. (05. Januar 2024). *Auswahlkriterien für Frameworks*. Von SoftGuide: <https://www.softguide.de/software-tipps/grundlagen-und-funktionen-frameworks> abgerufen
- Stork, A. (2006). *Ein Bild sagt mehr als Tausend Worte” – auch beim Vokabellernen?* Von Adam Mickiewicz University Repository: <https://repozytorium.amu.edu.pl/server/api/core/bitstreams/918e8e2c-ffad-4c80-a523-7e9b89ca7c8e/content> abgerufen
- Vue.js contributors. (31. Dezember 2023). *Introduction*. Von Vue.js: <https://vuejs.org/guide/introduction.html> abgerufen
- Wemakefuture AG. (12. Februar 2024). *Mach deine Anwendungen stark mit Supabase*. Von Wemakefuture AG: <https://www.wemakefuture.com/tools/supabase> abgerufen

12 Versicherung

Hiermit versichere ich, dass ich die Arbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und die Stellen der Facharbeit, die im Wortlaut oder im wesentlichen Inhalt aus anderen Werken oder Quellen entnommen wurden, mit genauer Quellenangabe kenntlich gemacht habe. Ich bin damit einverstanden, dass die von mir verfasste Facharbeit in der Schulbibliothek anderen zugänglich gemacht wird.

Sendenhorst, 25.02.2024 _____

Ort, Datum Unterschrift