

Projektarbeit von

**Justin Klaußnitzer, Marvin Eckert, Eric Naumburger,  
Tom Heinzig**

**IT-Fachenglisch Lernapplikation**

Prüfungsperiode: 12.07.2021 – 23.07.2021

Projektbetreuer: Herr J. Seidel

# Inhaltsverzeichnis

	<b>Abkürzungsverzeichnis</b>	<b>III</b>
	<b>Vorwort</b>	<b>IV</b>
<b>1</b>	<b>Planung und Durchführung des Projektes</b>	<b>1</b>
1.1	Problemstellung	1
1.2	Aufgabenverteilung	2
1.3	Zeiteinteilung	3
1.4	Verwendete Umgebungen und Frameworks	4
<b>2</b>	<b>Erstellung der Webapplikation</b>	<b>5</b>
2.1	Erstellung und Design des Frontend	5
2.2	Programmierung des Backend	7
2.3	Erstellen und Einbinden der Datenbank	9
2.4	API Programmierung	12
<b>3</b>	<b>Zusammenfassung</b>	<b>14</b>
	<b>Literaturverzeichnis</b>	<b>15</b>
	<b>Anlagenverzeichnis</b>	<b>16</b>

## Abkürzungsverzeichnis

API	Application Programming Interface
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
AI	Artificial Intelligence
CSV	Character-separated Values
JSON	JavaScript Object Notation

## **Vorwort**

Die zweiwöchige Projektarbeit wurde von vier Schülern des Beruflichen Schulzentrums für Technik I - Industrieschule Chemnitz durchgeführt. Jedes Gruppenmitglied befindet sich im zweiten Lehrjahr der Ausbildung zum Fachinformatiker für Anwendungsentwicklung und hat einen Ausbildungsvertrag mit der Volkswagen Sachsen GmbH. Das Thema der Projektarbeit ist die Erstellung einer interaktiven Webapplikation, welche Schülern beim Erlernen von englischen Vokabeln und typischen Phrasen im Bereich IT helfen soll. Dafür wurde sich an einen Englischlehrer der Berufsschule gewendet, welcher dieses Projekt betreute und in Zukunft auch nutzen kann. Das Projekt umfasst die Erstellung der Applikation und eine darauffolgende Verteidigung in Form einer zwanzig-minütigen Präsentation.

# **1. Planung und Durchführung**

## **1.1 Problemstellung**

Die Frage welche sich für diese Projektarbeit gestellt wurde: Wie kann man das Erlernen eines englischen Wortschatzes für den Bereich IT interaktiv und einfach zugänglich für Berufsschüler gestalten? Da der Projektbetreuer für dieses Vorhaben ein Fachenglischlehrer der Industrieschule Chemnitz war, wurden Lehrbücher vorgegeben, welche Vokabelübungen und IT-spezifische Dialoge enthalten. Die Übungen aus diesen Büchern sollen abgewandelt in digitaler Form durchführbar sein. Um den einfachen Zugang für Berufsschüler zu gewährleisten, hat sich das Projektteam dazu entschieden, eine Webanwendung zu erstellen. Die User sollen durch die Anwendung in der Lage sein, verschiedene Übungen auswählen und durchführen zu können. Durch die Übungen soll das IT-spezifische, englische Vokabular gefestigt und branchen-typische Sätze und Phrasen erlernt werden. Ein Grund warum sich das Team für eine Webanwendung zum Erlernen von Fachenglisch entschieden hat, ist die einfache Aufteilbarkeit der Aufgaben. Eine Webanwendung, wie sie sich das Team vorgestellt hat, ist klar unterteilt in ein Frontend, Backend, einer API und einer Datenbank. Durch den Aufbau der Applikation mit verschiedenen Übungsarten, ist das Projekt insgesamt sehr modular aufgebaut, was es einfach macht, je nach verbleibender Projektzeit zu entscheiden, welche Funktionen noch umgesetzt werden.

Hauptbestandteil des Projektes ist die Erstellung einer Vokabel- und Satzdatenbank mit englischen Worten und deren Übersetzungen. Aus dieser Datenbank sollen zufällig Vokabeln ausgewählt und abgefragt werden. Dabei ist jede Vokabel einem Thema zugeordnet, damit der User selbst auswählen kann, aus welchem Bereich der IT noch Übungsbedarf besteht. Optional sind Hörübungen geplant, bei denen der User englische Sätze vorgesprochen bekommt und diese in einem Textfeld wiedergeben muss.

Durch diese Übung soll das Hörverstehen der englischen Sprache gefestigt werden. Um diese Art an Aufgaben zu realisieren, müssen englische Sätze aus der angefertigten Datenbank entnommen werden und über einen Text-to-Speech Synthesizer in eine abspielbare Audiodatei umgewandelt werden.

Die letzte Übungsart die optional umgesetzt werden sollte, war das sogenannte „Satzpuzzle“. In dieser bekommt der User einen englischen Satz, und muss die deutsche Übersetzung aus vorgegebenen Satzbruchstücken zusammengesetzt werden.

## **1.2 Aufgabenverteilung**

Das Projekt wurde von vier Teammitgliedern bearbeitet, welche prinzipiell an drei verschiedenen Aufgaben gearbeitet haben. Ein Schwerpunkt war die Erstellung des Frontend bzw. der Benutzeroberfläche durch ein Mitglied der Gruppe. Es musste eine simple aber gut designte Oberfläche erstellt werden, in welcher der User einfach zwischen verschiedenen Aufgabentypen navigieren kann. Insbesondere beim Vokabeltest muss ein Benutzer unterschiedliche Einstellungen schnell und einfach tätigen können, um die Übungen auf die eigenen Bedürfnisse anzupassen. Der Rest des Teams hat regelmäßig die Ergebnisse des Frontend-Designers überprüft und Feedback zu den Gestaltungsideen und Design-Entscheidungen gegeben. Es war ebenfalls Teil der Aufgabe ein passendes Farbschema zu finden, welches sich durch alle Seiten der Webapplikation zieht. Es mussten auch passende, urheberrechtsfreie Bilder für die Hintergründe gefunden werden. Ein weiteres Teammitglied war damit beschäftigt die Datenbanken zu füllen, welche später für die Erstellung des Backend benötigt wurden. Die Aufgabe bestand darin Vokabellisten aus den vom Projektbetreuer gestellten Englisch-Fachbüchern zu entnehmen und in einer Excel-Tabelle zu speichern.

Ebenfalls müssen englische Sätze und Phrasen aus den Lehrbüchern entnommen werden, und mit einer entsprechenden Übersetzung in einer weiteren Tabelle gespeichert werden. Dasselbe Teammitglied war Hauptverantwortlicher für das Schreiben der Projektdokumentation und die Erstellung der Präsentation. Die verbleibenden beiden Mitglieder der Gruppe programmierten das Backend und die API der Webapplikation. Das Testen der Applikation wurde abwechselnd vom gesamten Team unternommen. Auch bei den Treffen mit dem Projektbetreuer waren alle Gruppenmitglieder anwesend.

### **1.3 Zeiteinteilung**

Das Projekt wurde in einem Zeitraum von zwei Wochen bearbeitet. Montag bis Donnerstag stand dem Team jeweils acht Stunden für die Bearbeitung der Problemstellung zur Verfügung. Freitags jeweils nur sechs Stunden. Die gesamte erste Woche vom 12.07.2021 bis zum 16.07.2021 fokussierte sich das Team auf die Erstellung der Webapplikation. Es wurde parallel an der Erstellung des Frontend, Backend und der Datenbanken gearbeitet. Pro Tag wurde jeweils eine Stunde dem Schreiben der Dokumentation gewidmet. Erst ab der zweiten Woche bekam die Dokumentation und das Erstellen der Präsentation Priorität. Da am Donnerstag, dem 22.07.2021, eine Probe für die Präsentation stattfand, mussten die Folien und das Skript für den Vortrag bis dahin fertig sein. Gleichzeitig hat das Programmiererteam den eigenen Code auf Fehler getestet und auftretende Schwierigkeiten beseitigt. Am Freitag dem 23.07.2021 wurde das Projekt mit einer Präsentation in der Aula der Industrieschule Chemnitz abgeschlossen.

## 1.4 Verwendete Umgebungen und Frameworks

Der gesamte Quellcode der Webapplikation wurde mithilfe des kostenlosen Quelltext-Editors Microsoft Visual Studio Code erstellt. Für das Design wurde die Auszeichnungssprache HTML, die Formatierungssprache CSS und das freie CSS-Framework Bootstrap genutzt. Zusätzlich kam JavaScript für die Funktionalität des Frontend hinzu. Das Backend wurde mit dem Webframework Flask in der Programmiersprache Python geschrieben. Dementsprechend hat das Programmier-Team für die API die Flask Erweiterung Flask-RESTful genutzt. Die Datenbank wurde mit dem Tabellenkalkulationsprogramm Microsoft Excel vorbereitet. Die eingetragenen Vokabeln und Sätze wurden in eine SQLite Datenbank migriert. Auf diese wird mit dem DB Browser für SQLite zugegriffen. Um die Vokabeln schnell aus den Lehrbüchern übernehmen zu können, wurde die Applikation Adobe Scan genutzt, um die Vokabellisten abzufotografieren, in PDF Dokumenten zu konvertieren und per E-Mail an unsere Arbeitscomputer zu schicken. Die Vokabellisten wurden mithilfe der kostenlosen AI-Tabellenextraktionswebsite Parsel.ai aus den PDF-Dokumenten entnommen und in einfach zu bearbeitende Textdateien umgewandelt.



## **2 Erstellung der Web-Applikation**

### **2.1 Erstellung des Frontend**

Beim Frontend handelt es sich um das grafische Interface mit dem ein Benutzer interagiert. Die Unterseiten wurden in der Reihenfolge erstellt, in der ein potentieller Benutzer auch durch die Menüs navigieren würde. Das heißt, der Verantwortliche für das Frontend hat mit dem Design der Startseite begonnen, und sich dann Schritt für Schritt durch die Auswahlmöglichkeiten gearbeitet. Das Design der Startseite kann in Abbildung A1 im Anhang zu sehen.

Nach der ersten Seite folgt ein Auswahlmenü, bei dem der Nutzer eine Englisch-Übung aussuchen kann. (Siehe Abbildung A2). Diese Seite ist die Grundlage für viele der folgenden Seiten, da diese immer nach dem gleichen Schema aufgebaut sind. Es wurde darauf geachtet, dass diese Seiten sehr übersichtlich und einfach zu bedienen sind. Um dies umzusetzen, wurde mit großen Schaltflächen gearbeitet, welche mit einer Beschriftung und einem kleinen Symbol ausgestattet sind. Bei den Symbolen wurde die Website „fontawesome.com“ genutzt. Diese stellt Symbole kostenfrei zur Verfügung, solange die Quellen richtig angegeben werden.

Als Designphilosophie wurde immer wieder betont, dass die Menüs einfach zu navigieren sein sollen. Von einem Benutzer werden deshalb keine eigenen Eingaben erfordert. Es ist lediglich nötig Schaltflächen anzuklicken, um die gewünschten Optionen für eine Englischübung zusammenzustellen. Sobald der Benutzer sich innerhalb einer Übung befand, wurden alle zusätzlichen Steuerelemente in eine Navigationsleiste ausgelagert. Jedoch wurde diese Idee wieder verworfen und nun erfolgt die Navigation über Schaltflächen unterhalb der Übung.

Für die Hintergründe wurden ebenfalls schlichte Bilder mit wenigen Farben genutzt, damit diese nicht zu sehr vom eigentlichen Inhalt der Seite ablenken. Wie bei den Symbolen handelt es sich bei den Bildern um lizenzfreie und kostenlos nutzbare Hintergründe.

Diese wurden mit einem leichten Unschärfefeffer verfehen, damit die bedienbaren Elemente im Vordergrund stärker hervorgehoben werden. Um die Überschriften gut vom Hintergrund abzuheben, wurden diese mit einem Schatten versehen. Gleichzeitig erzielt dies einen 3D-Effekt, welcher die Seiten plastischer wirken lässt.

Das Farbschema war innerhalb des Teams lange umstritten, da sich nicht auf eine Farbkombination geeinigt werden konnte, die für eine IT-spezifische Englischlernplattform geeignet ist. Am Ende wurde sich auf eine blau-grüne Farbpalette geeinigt.

Für die Erstellung des Frontend wurde hauptsächlich HTML und CSS verwendet. Da viel Arbeitszeit aufgewendet wurde, die Elemente jeder Seite dynamisch an die Bildschirmgröße verschiedener Geräte anzupassen, wurde sich später dazu umentschieden Bootstrap zu verwenden.

Bei Bootstrap handelt es sich um ein freies CSS-Framework, welches es deutlich vereinfacht ein leicht anpassbares, grafisches Interface zu erstellen. Für dieses Framework gibt es bereits viele Designvorlagen. Durch das sogenannte Grid-System lässt sich zudem ein simpel strukturiertes Layout erstellen.

Besonders herausfordernd war es dynamische Menüs zu erstellen, die unterschiedlich große Schaltflächen besitzen, je nachdem wie viele Kapitel beispielsweise ein Englischbuch hat. Der User kann nach der Auswahl des Vokabeltests bestimmen, aus welchem Buch er abgefragt werden möchte. Da die Bücher unterschiedlich viele Kapitel haben, muss die Website flexibel darauf reagieren und in allen Fällen die Schaltflächen groß und übersichtlich darstellen. Um das umzusetzen musste Frontend und Backend zusammengeführt werden, was sich ebenfalls als Herausforderung herausstellte, da diese mit unterschiedlichen Sprachen erstellt wurden. Das Backend wurde in Python geschrieben. Es erstellt unter anderem Listen, die anzeigen wie viele Schaltflächen dynamisch erzeugt werden müssen. Dafür werden diese Listen in HTML-Text konvertiert und vom Frontend per JavaScript extrahiert. Dieses konvertieren zwischen verschiedenen Sprachen ist eher unelegant, weshalb sich das Team später dafür entschied das Ganze mit Bootstrap zu lösen.

## 2.2 Programmierung des Backend

Beim Backend handelt es sich um den funktionalen Teil der Webapplikation, in welchem aus den Daten der Datenbank dynamisch Elemente für das Frontend erzeugt werden. Dafür kommuniziert das Backend mit der API. Wenn der Nutzer im Frontend durch die Menüs navigiert, und beispielsweise eine bestimmte Übungsart auswählt, wird diese Information im Backend in der sogenannten „Session“ abgespeichert. Eine Session besteht während der User mit der Website kommuniziert und interagiert.

Jedes Mal, wenn innerhalb der Webapplikation eine Usereingabe entgegengenommen wird, wird diese Information über einen Link an das Backend weitergegeben und in der Session gespeichert. Danach nutzt das Backend die Informationen innerhalb der Session, um eine Anfrage an die API zu schicken, welche eine Antwort im JSON-Format schickt. Innerhalb dieser JSON befinden sich Datensätze, die aus der Vokabel- und Phrasendatenbank abgerufen wurden. Wenn der User zum Beispiel auswählt, aus welchem Buch er Vokabelübungen bekommen möchte, speichert sich die Session den Namen des Buches, und fragt danach die API, welche Kapitel in diesem Buch vorhanden sind. Die API durchsucht die Datenbank nach allen Kapiteln, die im Buch vorhanden sind, speichert die Ergebnisse im oben genannten Format und schickt diese zurück an das Backend.

Im nächsten Schritt generiert das Backend aus der JSON eine Liste, welche dann von einer Schleife durchlaufen wird. Durch diesen Vorgang können dynamisch HTML Elemente erzeugt werden, mit welchen der User dann weitere Auswahlmöglichkeiten treffen kann. Also je nach dem welches Buch vom User gewählt wurde, werden vom Backend dynamisch weitere Buttons erzeugt, welche die Kapitel im Buch anzeigen. In der aktuellen Version des Projekts speichert die Session nur die neusten Informationen über die Übungen, welche der User ausgewählt hat.

Das heißt: schließt der User eine Übung ab, geht zum Hauptmenü zurück und wählt einen weiteren Vokabeltest, „vergisst“ die Session welche Vokabeltests bereits abgeschlossen wurden. In Zukunft wäre es von Vorteil, wenn sich die Webapplikation merkt, welche Übungen bereits erfolgreich abgeschlossen wurden. Dies würde über Cookies realisiert werden, die nicht innerhalb der Projektzeit eingebaut werden konnten.

Während der Bearbeitung des Backend kam es immer wieder zu Verzögerungen, da dieser Teil des Projekts sehr abhängig vom Frontend war. Da die Entwicklung des Frontend stellenweise länger als erwartet gedauert hat, mussten die Funktionen der Applikation erstellt werden, ohne dass es die dazu passenden HTML-Seiten gab. Dies führte auch dazu, dass die Funktionen des Backend erst einige Zeit nach ihrer Erstellung getestet werden konnten.

Für die Entwicklung wurde sich für die Programmiersprache Python entschieden. Für diese Sprache wurde sich hauptsächlich entschieden, weil ein Großteil des Teams bereits Erfahrung mit Python hatte, und das Testen des Programmcodes schnell und einfach von statten geht. Das liegt vor allem an der interaktiven Konsole namens REPL. Mit dieser lässt sich Code testen und Änderungen ausprobieren, ohne dass der eigentliche Programmcode erst umgeschrieben und kompiliert werden muss. Unterstützend für die Webentwicklung wurde das Framework Flask eingesetzt. Mithilfe dieses Frameworks kann schon mit wenig Code eine simple Webapplikation erstellt werden. Auch in diesem Fall wurde sich für Flask entschieden, weil das Team bereits Erfahrung in vergangenen Projekten gesammelt hat. Im Gegensatz zu anderen Frameworks, beschränkt sich Flask nur auf die essentiellen Funktionen, die für die Webentwicklung benötigt werden.

Zusätzlich bietet das Framework eine sogenannte Templating Engine. Diese erlaubt es serverseitig eine dynamische Website zusammenstellen zu lassen, bevor diese zum User geschickt wird.

Das verringert die Zugriffszeit auf die einzelnen Unterseiten und der User muss nicht mit ansehen, wie sich eine Seite Schritt für Schritt zusammenbaut, sondern bekommt die komplette Seite auf einmal.

Des Weiteren ist es nicht nötig die dynamischen Funktionen der Website clientseitig mit JavaScript zu implementieren, was den Code unübersichtlicher machen würde. Stattdessen ist durch Flask Design und Funktion klar getrennt, was es auch einfacher macht, die Aufgaben aufzuteilen und getrennt an verschiedenen Teilen des Projekts zu arbeiten.

## **2.3 Erstellen und Einbinden der Datenbank**

Die Webapplikation nutzt eine Datenbank in welcher englische Vokabeln und Sätze mit ihrer dementsprechenden Übersetzung abgespeichert sind. Für die verschiedenen Übungsarten wird eine vom User ausgewählte Anzahl an Vokabeln oder Sätzen zufällig aus der Datenbank entnommen und abgefragt. Das Problem vor das die Gruppe gestellt war, bestand darin, dass die vorgegebenen Vokabeln und Sätze aus Lehrbüchern stammen mussten. Der Projektbetreuer stellte zwei Englisch-Lehrbücher für IT-Berufe zur Verfügung, welche im Literaturverzeichnis aufgeführt sind. In beiden Büchern befinden sich für jedes Kapitel – in den Büchern Units oder Modules genannt – Vokabellisten. Ziel war es diese in einem ersten Schritt in eine Excel-Tabelle zu übertragen.

Der erste Ansatz bestand darin ein Teammitglied dafür einzusetzen die Wörter mit ihren Übersetzungen abschreiben zu lassen. Dabei handelte es sich um einen sehr ineffektiven und zeitaufwendigen Weg, die über 1000 Vokabeln in eine Tabelle zu übernehmen. Auf digitale Versionen hatte die Projektgruppe keinen Zugriff, weshalb es keine andere Alternative gab, als mit den physischen Büchern zu arbeiten.

Um Zeit einzusparen wurde dazu übergegangen die Vokabellisten mit der Smartphone Applikation „Adobe Scan“ abzufotografieren und in PDF-Dateien umzuwandeln. Die erzeugten Dokumente wurden dann per E-Mail an unsere Arbeitsrechner geschickt. Nun lagen die Vokabellisten in digitaler Form vor, mussten aber immer noch in die Excel Tabelle eingefügt werden.

Die neue Arbeitsweise war nun die englischen Wörter nacheinander aus dem PDF-Dokument in die Excel-Tabelle zu kopieren. Da die Texterkennung durch „Adobe Scan“ nicht fehlerfrei war und das Layout der Vokabellisten es nicht ermöglichte mehr als ein englisches Wort auf einmal zu kopieren, war das Erstellen der Excel Tabelle leider immer noch zu zeitintensiv. Nun musste jedes Wort einzeln aus einer PDF kopiert werden, was nur minimal schneller war, als die Wörter per Hand aus den Büchern abzuschreiben.

Für tausende Vokabeln war dies immer noch zu aufwendig und hätte schätzungsweise die gesamte erste Woche der Projektzeit in Anspruch genommen. Ein neuer Lösungsansatz bestand darin, die Vokabelliste spaltenweise abzufotografieren und per Adobe Scan in PDF-Dokumente umzuwandeln. Wie dabei vorgegangen wurde ist in Abbildung A3 im Anhang zu sehen. Daraufhin wurden die PDF-Dokumente auf der Seite [parsel.ai](https://parsel.ai) hochgeladen und in einfache Textdateien konvertiert.

Dank dieser Konvertierung konnten nun ganze Blöcke an Vokabeln kopiert und in die Excel-Tabelle eingefügt werden. Da hinter den englischen Worten oftmals in eckigen Klammern die Aussprache in Lautschrift steht, mussten die Vokabeln per „Suchen und Ersetzen“-Befehl so zugeschnitten werden, dass nur noch die Wörter in der Tabelle stehen.

Neben den beiden Spalten für englische Wörter bzw. Sätze und deren Übersetzungen, waren für die spätere Umsetzung als Datenbank noch weitere Informationen nötig. Für jede Vokabel wurde festgehalten aus welchem Buch sie stammt, zu welchem Kapitel sie gehört (in dem Fall „Unit“ oder „Module“), und welcher Kategorie das Wort angehört (in der Datenbank dann „Topic“ genannt). Bei den Kategorien handelt es sich um kapitel-übergreifende Themen, wie zum Beispiel „Software“ oder „Hardware“

Die Tabelle für englische Sätze bzw. Phrasen beinhaltet neben der deutschen Übersetzung eine Spalte namens „Audio“. In dieser standen zunächst erneut die englischen Sätze, exakt so wie sie aus den Büchern übernommen wurden.

Aus dieser Spalte sollte später ein Text-to-Speech Synthesizer die englischen Texte auslesen und in gesprochene Audiodateien umwandeln. Diese zusätzliche Spalte dient dazu eventuelle Fehler bei der Aussprache des Text-to-Speech Synthesizers zu korrigieren, indem falsch ausgesprochene Worte bewusst falsch geschrieben werden, um eine richtige Aussprache zu erzwingen.

Der nächste Arbeitsschritt war es nun die Excel-Tabellen in eine SQLite Datenbank zu migrieren. Dafür wurde zunächst die Excel-Tabelle in eine CSV-Datei konvertiert. CSV-Dateien haben den Vorteil, dass sie einfach verarbeitet werden können. Als Vorbereitung wurde mit dem „DB Browser for SQLite“ verschiedene Tabellen über SQL-Befehle erstellt. Um eine Datenbank zu erstellen, die alle Normalformen erfüllt, wurden eigene Tabellen für „Buch“, „Unit“ und „Topic“ angelegt. Zum Problem wurde, dass eine englische Vokabel mehrere richtige, deutsche Übersetzungen hat.

Ein User hat beim Vokabeltest also mehrere richtige Antwortmöglichkeiten, die alle als richtig gewertet werden müssen. In der Datenbank müssen also pro englisches Wort mehrere dazu passende deutsche Übersetzungen abgespeichert werden. Um dies zu realisieren, hat jedes englische Wort eine eigene ID bekommen. Und jede richtige Übersetzung bekam dieselbe ID. Gibt ein User nun beim Vokabeltest eine Lösung ein, wird die gesamte Datenbank nach der ID des englischen Worts durchsucht, jedes deutsche Wort mit derselben ID extrahiert und verglichen, ob eines der Wörter aus der Datenbank mit der Usereingabe übereinstimmt. Die fertige SQLite Datenbanktabelle ist in Abbildung A4 im Anhang zu sehen.

Die Tabellen „Buch“, „Unit“ und „Topic“ wurden per Hand in die Datenbank eingefügt, da es sich bei diesen um sehr kleine Datensätze handelt. Wie schon am Anfang dieses Abschnitts deutlich wurde, handelt es sich bei den Vokabeln an sich jedoch um einen sehr großen Datensatz. Diesen per Hand in die Datenbank zu übernehmen wäre mühselig und ineffizient. Die aus den Excel-Tabellen erstellten CSV-Dateien werden nun benötigt.

Über ein Skript das mit Python geschrieben wurde, wird die CSV-Datei Zeile für Zeile durchgegangen und in einzelne Datensätze separiert. Danach wird jede Zeile in ihre Spalten aufgetrennt.

Diese Separierung der Daten ist einfach möglich, da in einer CSV-Datei alle Daten mit einem Semikolon getrennt wurden, und einzelne Zeilen durch einen Zeilenumbruch gekennzeichnet sind. Aus jeder Zeile werden die englischen Wörter und ihre deutschen Übersetzungen gefiltert. Hier werden die Wörter auch mit einer ID versehen. In einer Schleife werden die Daten schlussendlich mithilfe von dynamischen SQL Befehlen in die SQLite Datenbank eingespeist.

## 2.4 API Programmierung

Bei der API im Projekt „IT-Fachenglisch Lernapplikation“ handelt es sich um eine Schnittstelle zwischen der Vokabel-Datenbank und dem Backend der Webapplikation. Die Schnittstelle wartet auf Anfragen der Website und leitet diese in Form von SQL-Befehlen an die Datenbank weiter. Durch die SQL-Befehle können Daten in Form von Vokabeln oder englischen Sätzen aus der Datenbank abgerufen werden. Die API nimmt diese Datensätze entgegen und leitet diese an das Backend der Website weiter. Dabei werden die angefragten Daten in das JSON-Format umgewandelt, da dies vom Backend leicht verarbeitet werden kann. Das Zusammenspiel aller Komponenten lässt sich durch die selbst erstellte Grafik in Abbildung A5 gut nachvollziehen. Das Nutzen einer API war eine Entscheidung die getroffen wurde, weil sich das Projekt dadurch besser aufteilen ließ. Außerdem sorgt eine extra Schnittstelle für Datenbankzugriffe dafür, dass es nicht zu sogenannten SQL-Injections kommen kann.

Dabei handelt es sich um Sicherheitslücken, bei denen ein User Datenbankbefehle manipulieren kann, um zum Beispiel Datensätze aus der Datenbank zu löschen, oder diese zu verändern.<sup>1</sup>

---

<sup>1</sup> (Myra Security. (o.J.). Abgerufen am 16.07.2021, von [myrasecurity.com/de/was-ist-sql-injection/](https://myrasecurity.com/de/was-ist-sql-injection/))



Ein weiterer Vorteil, der dazu geführt hat, dass sich für die Nutzung einer API entschieden wurde, ist die einfache Erweiterbarkeit und Portabilität dieser. Jeder Unterseite der Webapplikation kann auf die API zugreifen und Datenbankabfragen an die Schnittstelle weitergeben. Sollten weitere Übungsarten in Zukunft hinzugefügt werden, muss nur die API angepasst werden, und nicht jede Unterseite der Webapplikation. Auch wenn die Englisch-Lernapplikation in Zukunft auf anderen Plattformen laufen soll, wird dies durch eine Datenbankschnittstelle begünstigt. Die API wurde mit der Programmiersprache Python erstellt, aus denselben Gründen, wie in 2.2 nachzulesen ist.

In einem Projekt, welches bereits im Rahmen der Berufsausbildung erstellt wurde, kam eine ähnliche Datenbankschnittstelle zum Einsatz. Deshalb konnte das Programmiererteam auf Code-Beispielen und Erfahrungen aus vergangenen Projekten zugreifen und bereits angeeignetes Wissen wiederverwenden. Durch das Nutzen von Python konnte die Bibliothek SQLite verwendet werden, welche Datenbankzugriffe stark vereinfacht. Das Programmiererteam hat sich außerdem das Ziel gesetzt das Abrufen von Datensätzen aus der Datenbank möglichst mit einer einzigen SQL-Abfrage durchzuführen. Dadurch ist die Webapplikation performanter, falls sie jemals mit sehr großen Datenmengen arbeiten muss. Das spielt für die aktuelle Projektgröße keine große Rolle, sollte die Englisch-Lernapplikation in Zukunft jedoch um viele Datensätze erweitert werden, ist es von Vorteil, wenn Datenbankabfragen auf ressourcenschonende Art und Weise durchgeführt werden.

### **3. Zusammenfassung**

Am Ende der Projektzeit wurden von den drei geplanten Englischübungen zwei vollständig implementiert. Das Ziel die Applikation für Berufsschüler einfach erreichbar zu machen, ist noch nicht voll realisiert, da die Website nicht online auf einem Server zur Verfügung gestellt wurde.

Dabei handelt es sich um ein Vorhaben, welches noch weitere Arbeitszeit benötigen würde und von einem Projektteam betreut werden müsste, welches die Website regelmäßig wartet, erweitert und Fehler behebt. Dennoch lässt sich die Webapplikation prinzipiell offline für Übungszwecke einsetzen. Der Hauptnutzen, welcher aus dem Projekt gezogen wurde, ist die Vorbereitung und Übung für die zukünftige Abschlussarbeit der Berufsausbildung.

Das Englisch-Projekt hat geholfen, zu erlernen, den Zeitaufwand hinter einem Projekt besser einzuschätzen.

Wie schon in der Problemstellung beschrieben, wurde das Projekt in verschiedene Aufgabenbereiche eingeteilt.

Es hat sich jedoch herausgestellt, dass die einzelnen Bestandteile einer Webapplikation unterschiedlich lange Bearbeitungszeiten benötigen, die zusätzlich von den Fähigkeiten und Erfahrungen des Teammitglieds abhängen. Während der Projektarbeitszeit kam es immer wieder zu Verzögerungen für manche Gruppenmitglieder, da für das Weiterarbeiten benötigte Bestandteile des Projekts noch nicht fertiggestellt wurden. Wenn die fehlenden Features umgesetzt werden würden, könnte die fertige Webapplikation durchaus schulintern genutzt werden, um beim lernen von Fachenglisch Vokabeln und Sätzen zu helfen. Als Applikation im Internet würde sich das Programm nicht nutzen lassen, da es bessere Alternativen gibt, die jedoch nicht auf Fachenglisch für IT-Berufe spezialisiert sind.

## Literatur- und Quellenverzeichnis

Courtney, B., Kleinschroth, R., Williams, I., (2018). *IT Matters. Englisch für IT-Berufe* (3. Aufl.). Berlin: Cornelsen Verlag GmbH

Dr. Schäfer, W., Eifler, J., Humphreys, A., Humphreys, J., Roth, T., Schäfer, C., Schäfer, M. (2013). *IT Milestones. Englisch für IT-Berufe* (1. Aufl.). Stuttgart, Leipzig: Ernst Klett Verlag

Myra Security. (o.J.). Abgerufen am 16.07.2021, von  
[myrasecurity.com/de/was-ist-sql-injection/](https://myrasecurity.com/de/was-ist-sql-injection/)

## Abbildungsverzeichnis

Abbildung	Titel	Seite
A1	Front Page der Webapplikation	17
A2	Auswahlmenü Übungen	17
A3	Screenshot von Adobe Scan	18
A4	Screenshot SQLite Datenbank	19
A5	Komponenten Übersicht	20

## A Anhang



Abbildung A1 - Front Page der Webapplikation (Quelle: Autor)

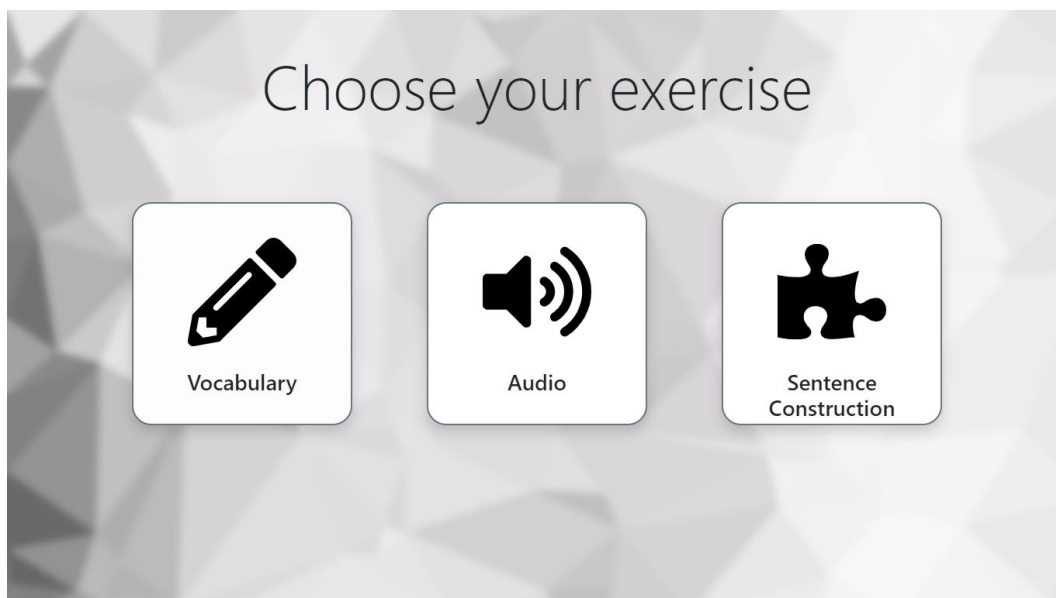


Abbildung A2 - Menü zum Auswählen der Übungen. (Quelle: Autor)

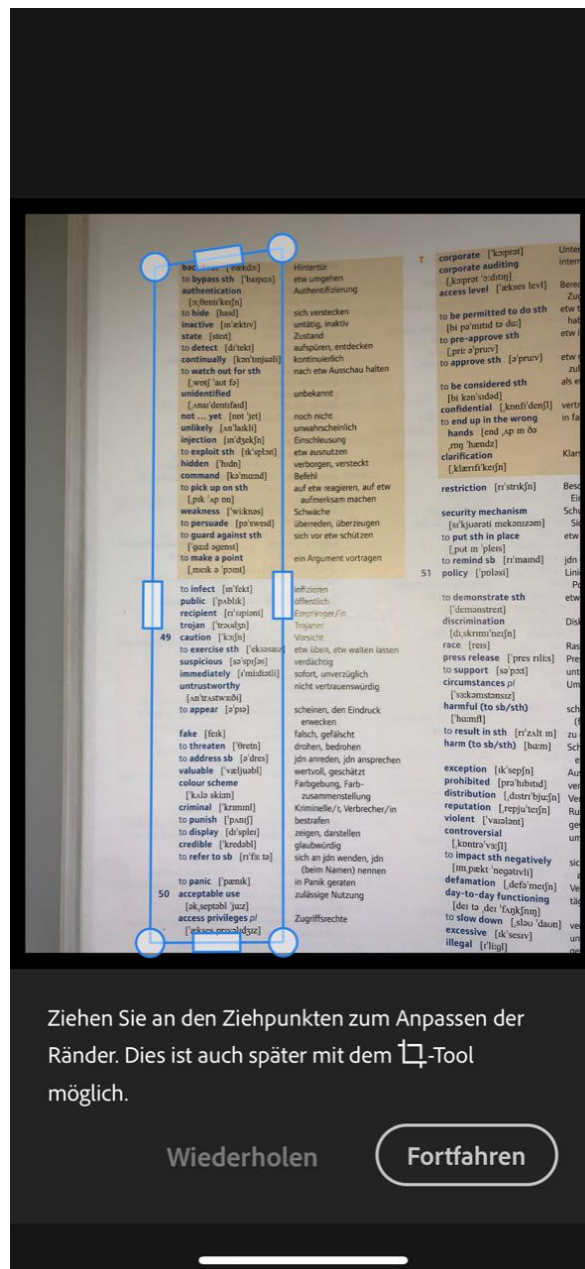


Abbildung A3 - Screenshot aus der Applikation Adobe Scan. Es ist zu sehen, wie eine einzelne Spalte gescannt wird, damit diese später in ein PDF konvertiert und einfach in eine Excel Tabelle übernommen werden kann. (Quelle: Autor)

	book	unit	topic	language	word	translation
1	IT Matters	My Company	Company	English	foundation	1
2	IT Matters	My Company	Company	German	Grundlage	1
3	IT Matters	My Company	Company	English	company	2
4	IT Matters	My Company	Company	German	Unternehmen	2
5	IT Matters	My Company	Company	English	profile	3
6	IT Matters	My Company	Company	German	Porträt	3
7	IT Matters	My Company	Company	German	Beschreibung	3
8	IT Matters	My Company	Company	German	Profil	3
9	IT Matters	My Company	Company	English	apprentice	4
10	IT Matters	My Company	Company	German	Auszubildender	4
11	IT Matters	My Company	Company	German	Auszubildende	4
12	IT Matters	My Company	Company	German	Lehrling	4
13	IT Matters	My Company	Company	English	IT administrator	5
14	IT Matters	My Company	Company	German	IT-Kaufmann	5
15	IT Matters	My Company	Company	German	IT-Kauffrau	5
16	IT Matters	My Company	Company	English	trainee	6
17	IT Matters	My Company	Company	German	Auszubildender	6
18	IT Matters	My Company	Company	German	Auszubildende	6
19	IT Matters	My Company	Company	German	Praktikant	6
20	IT Matters	My Company	Company	German	Praktikantin	6
21	IT Matters	My Company	Company	English	from abroad	7
22	IT Matters	My Company	Company	German	aus dem Ausland	7
23	IT Matters	My Company	Company	English	network	8
24	IT Matters	My Company	Company	German	Netzwerk	8
25	IT Matters	My Company	Company	English	component	9

*Abbildung A4 – Screenshot der fertigen Datenbanktabelle. Es sind die vom Projektteam angelegten Spalten zu sehen. Es ist außerdem zu erkennen, dass englische Worte und ihre Übersetzung dieselbe translation-ID haben. (Quelle: Autor)*

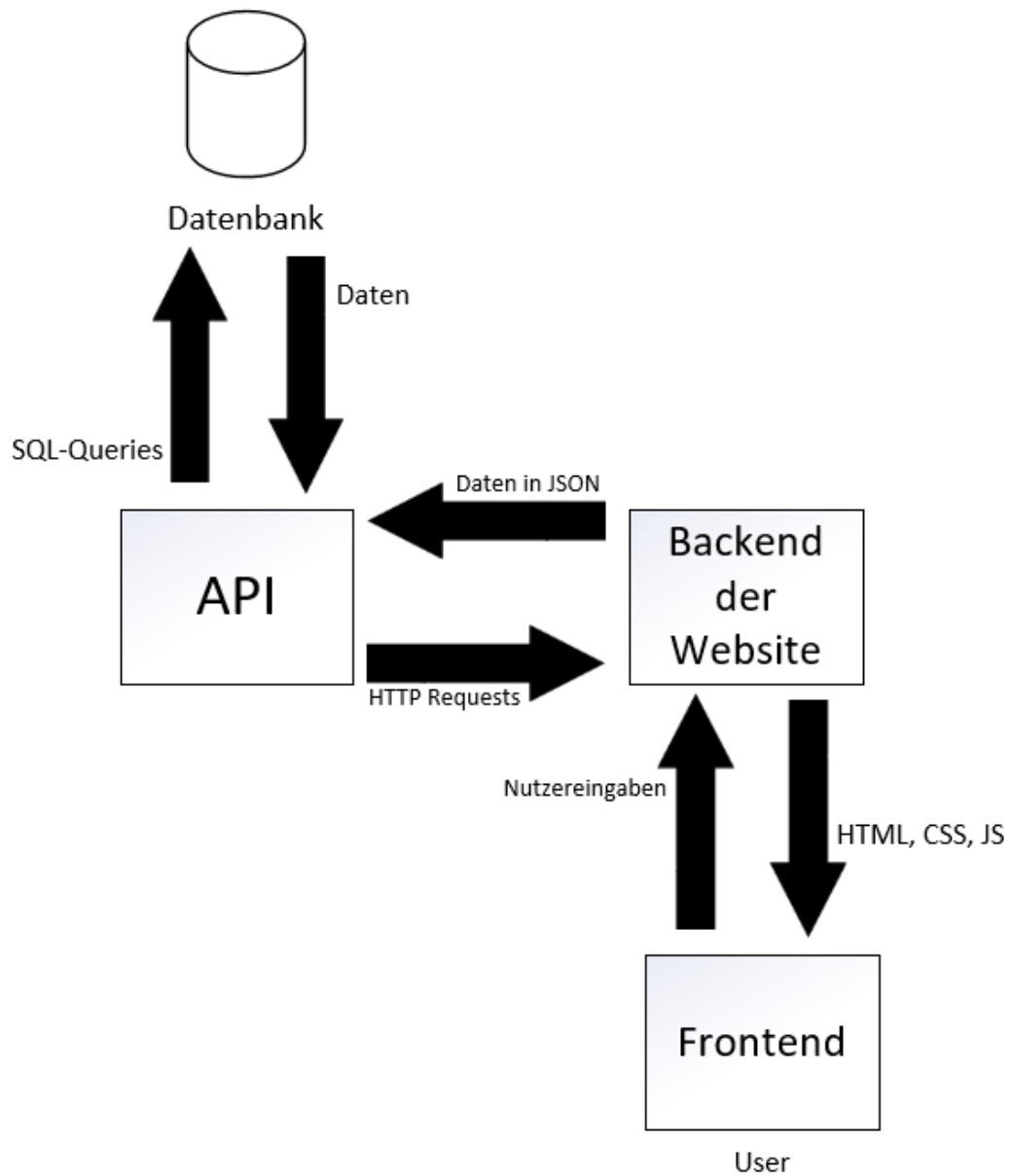


Abbildung A5 - Grafik welche das Zusammenspiel der Komponenten darstellt.  
(Quelle: Autor)



## **Erklärung**

Wir versichern durch unsere Unterschriften, dass wir diese Projektarbeit mit dem Thema „IT-Fachenglisch Lernapplikation“ selbstständig, ohne fremde Hilfe angefertigt, alle Stellen, die wir wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen, als solche kenntlich gemacht und uns auch keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient haben. Die Projektarbeit wurde in dieser oder ähnlicher Form noch nicht im Beruflichen Schulzentrum für Technik I - Industrieschule Chemnitz eingereicht. Uns ist bekannt, dass gemäß der Prüfungsordnung für die Durchführung von Abschlussprojekten der Industrieschule Chemnitz durch Täuschungshandlungen die Prüfung als nicht bestanden erklärt werden kann.

Ort, Datum, Unterschrift (mit Vor- und Nachnamen)