

## Exposé

# Entwicklung einer E-Learning-Webapp

Projektbasiert Programmieren lernen

Björn Kawecki

September 2024

## 1 Einleitung

Während meines Studiums stellte ich fest, dass es für Russisch kaum Lernmaterial gibt, um effizient Vokabeln zu lernen und zu wiederholen, auch nicht im Onlinesegment. Der Grund hierfür ist offensichtlich, dass Russisch keine populäre Fremdsprache ist und sich die aufwendige Herstellung spezifischer Bücher oder Software kommerziell nicht lohnt.

Anfang 2020, im letzten Jahr meines Studiums, hatte ich daher die Idee, selbst diesem Problem Abhilfe zu schaffen und ein entsprechendes Programm zu schreiben. Die Webapp, die ich entwickeln wollte, sollte Nutzern die Möglichkeit bieten, systematisch eine umfassende Sammlung russischer Vokabeln zu lernen.

Näheres zu meinen Überlegungen zum Thema digitales Lernmaterial finden Sie auf meiner Webseite [bjoernkawecki.de](http://bjoernkawecki.de).

## 2 Planung

Zuerst hatte ich zwei grundsätzliche Fragen zu klären: Wie plant man ein Projekt und was zeichnet moderne Webentwicklung aus? Die Antworten auf diese Fragen bezog ich aus überwiegend englischsprachigen Online-Quellen. Die Ergebnisse dieser Recherche mündeten in einen Projektplan mit Entwicklungszielen, User Journey, Empathy Map, technologischen Anforderungen und zu berücksichtigenden Ansätzen im Bereich UX/UI.

Über Video-Tutorials eignete ich mir die Grundlagen in HTML, CSS, JavaScript, Python, SQL, Git und Docker an. Zur Vertiefung las ich Fachliteratur und führte kleinere Projekte durch. Da ich schnell die Komplexität moderner Webentwicklung begriff, beschäftigte ich mich früh mit möglichen Frameworks, darunter Angular, React, NestJS und Django.

Über das Domäne-Wissen im Bereich russische Philologie verfügte ich bereits durch mein Studium, sodass ich früh mit der Gestaltung des Datenmodells sowie der Erstellung von Entitäts-Beziehungs-Diagrammen beginnen konnte.

### 3 Entwicklung

Mit der Festlegung auf das Python-Framework Django begann im Sommer 2023 die Hauptphase der Entwicklung. Django bot den Vorteil, dass typische Anforderungsbereiche der Anwendungsentwicklung bereits zur Verfügung standen, darunter Authentifizierung, Content-Administration, E-Mail, ORM, Logging und Schutz vor XSS und CSRF. Zur Umsetzung von AJAX und SSE im Frontend entschied ich mich für die leichtgewichtige JavaScript-Bibliothek htmx.

Mit der Wahl dieses Tech-Stacks konnte ich überwiegend in Python und relativ schnell eine komplexe Anwendung entwickeln, die ich später und bei Bedarf mit dem Django-REST-Framework in eine REST-API umschreiben könnte.

Weitere Technologien, die zum Einsatz kamen, waren Docker mit Docker-Compose, Git, Celery für asynchrone Aufgaben mit Redis als Broker, die JavaScript-Diagramm-Bibliothek ApexCharts und die API des Bezahl Dienstleisters Stripe.

### 4 Produktion

Bei Fertigstellung umfasste das Minimalprodukt im Mai 2024 eine Landing-Seite mit Registrierfunktion sowie einen geschützten Bereich mit der Geschäftslogik, darunter ein SSE-Feed, zeit- und datenbanksensible Widgets und mehrere Listen- und Detailansichten der App- bzw. Nutzerdaten mit graphischer Visualisierung.

Für das Hosten der Anwendung entschied ich mich für die App-Plattform und eine gemanagte Datenbank des Cloud-Service-Anbieters DigitalOcean. Dies hatte den Vorteil, dass ich zwei fehleranfällige Bereichen, Server- und Datenbankmanagement, auslagern und Updates für meine App komfortabel per GitHub-Push auslösen konnte.

Um Produktionskosten für das Minimalprodukt zu sparen, entfernte ich etwas später Celery, da ich hierfür eine zweite, zum jetzigen Zeitpunkt verzichtbare Datenbank benötigte.

### 5 Resümee

Eine große Herausforderung war, alle Rollen, die in einem Projekt anfallen, selbst auszufüllen. Für die Erstellung von Audiodateien und die Umsetzung des Logos und der Illustration auf der Landing-Seite gab es Zuarbeit aus meinem privaten Umfeld. Die Zusammenarbeit mit einer Web-Designerin, die ich für den Entwurf eines geeigneten Layouts mit ansprechender Farbgebung engagierte, führte hingegen nicht zum gewünschten Ergebnis, und die Aufgabe fiel zurück an mich.

Die wichtigste Erfahrung war für mich zu sehen, wie viel ich mit der genügend Zeit und Motivation erreichen konnte. Die einschlägigen Debugging-Foren, Blog-Beiträge, Tutorials und Dokumentationen waren auf diesem Weg unersetzlich, wenn auch der persönliche Austausch mit anderen Entwicklern manchmal hilfreich gewesen wäre. Umso erfreulicher waren daher die Momente, wenn man sich selbst überraschen und feststellen konnte, dass man an den selbst gestellten Aufgaben über sich hinausgewachsen war.