

Winterprüfung 2025

Ausbildungsberuf

Fachinformatiker/Fachinformatikerin (VO 2020) Fachrichtung:
Anwendungsentwicklung

Prüfungsbezirk

1201 FIA 4 (AP T2V1)

Tim Petersdorff

Identnummer: 840296-92

E-Mail: tim.petersdorff@icloud.com, Telefon: +49 176 23921074

Ausbildungsbetrieb: Lehrwerk GmbH

Projektbetreuer: Ulrich Meißner

E-Mail: Ulrich.Meissner@lehrwerk.de, Telefon: +49 800 590 631 00

Thema der Projektarbeit

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen
Präsentation und Steuerung interner Abläufe

1 Thema der Projektarbeit

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

2 Geplanter Bearbeitungszeitraum

Beginn: 11.10.2025

Ende: 19.11.2025

3 Ausgangssituation

Die Lehrwerk GmbH bietet Umschulungen und Weiterbildungen im IT-Bereich an. Die aktuelle Webseite ist technisch und inhaltlich veraltet, verfügt über keine interaktiven Funktionen oder direkten Kontaktmöglichkeiten, was zu erhöhtem Kommunikationsaufwand über andere Kanäle führt.

Aktuelle Probleme:

- Veraltete Webpräsenz ohne moderne Funktionen und SEO-Optimierung
- Keine integrierten Kontakt- oder Terminbuchungsmöglichkeiten
- Hoher manueller Aufwand bei der Bearbeitung von Anfragen
- Fehlende digitale Bereitstellung von Lernmaterialien und strukturierte FAQ-Sektion
- Keine zentrale Übersicht über Kurszeiten, Fördermöglichkeiten und Angebote

4 Projektziel

4.1 Meine Aufgaben im Projekt

Frontend (Öffentlicher Bereich) entwickeln und programmieren

Öffentliche Plattform mit Bildungsangeboten und relevanten Informationen:

- HTML5, Tailwind CSS für responsives Design, Vanilla JavaScript für Interaktivität

Möglichkeit zur Kontaktaufnahme und Terminvereinbarung:

- HTML-Formulare mit JavaScript-Validierung, Express.js Backend, MariaDB-Speicherung

Ergänzende Bereiche wie Downloads und weitere Informationsinhalte:

- Node.js mit Multer für Datei-Upload, MariaDB für Metadaten-Verwaltung

Backend (Administrationsbereich) entwickeln und programmieren

Content-Management-System zur Pflege der Inhalte:

- Eigenentwickeltes CMS in HTML/JavaScript, Node.js/Express.js mit Session-Authentifizierung, REST-API, MariaDB

Upload und Organisation von Lernmaterialien:

- Node.js Multer für Uploads, MariaDB für Kategorisierung, Express.js Routing

Verwaltung von Terminanfragen, Kontaktanfragen und Benutzern:

- Express.js API-Endpunkte, MariaDB-Strukturen, Admin-Interface mit JavaScript

4.2 Welche Anforderungen müssen erfüllt sein?

Funktionale Anforderungen:

- Responsives, barrierefreies Design
- DSGVO-konform
- Plattformunabhängig mit breiter Browserkompatibilität
- Gestaltung gemäß Corporate Identity

Technische Anforderungen:

- Schnelle Ladezeiten und skalierbare Architektur

4.3 Welche Einschränkungen müssen berücksichtigt werden?

- Integration in die vorhandene IT-Infrastruktur
- Mitarbeiterschulung außerhalb der Projektphase

5 Zeitplanung

5.1 Geplante Projektdauer anhand der Projektphasen

- Analyse – 6 Stunden
- Entwurf – 14 Stunden
- Entwicklung & Implementierung – 40 Stunden
- Test & Abnahme – 7 Stunden
- Deployment & Einführung – 4 Stunden
- Dokumentation – 9 Stunden.

5.2 Arbeitsschritte nach Projektphasen

Analyse

- Ist-Analyse durchführen
- Anforderungen mit Stakeholdern klären
- Pflichtenheft erstellen
- Technologien auswählen

Entwurf

- Systemarchitektur erstellen
- Datenbankmodell (ERM) entwerfen
- UI/UX-Design
- Sicherheitskonzept entwickeln

Entwicklung & Implementierung

- Entwicklungsumgebung einrichten
- Datenbank erstellen und implementieren
- Backend entwickeln und programmieren
- Frontend entwickeln
- Sicherheitsfunktionen implementieren

Test & Abnahme

- Umsetzung anhand der Anforderungen prüfen

Deployment & Einführung

- Produktivumgebung einrichten
- Anwendung bereitstellen

Dokumentation

- Benutzerdokumentation verfassen
- Projektdokumentation verfassen

6 Anlagen

keine

7 Präsentationsmittel

Notebook und Beamer
Rüstzeit: 10min

8 Hinweis!

Ich bestätige, dass der Projektantrag dem Ausbildungsbetrieb vorgelegt und vom Ausbildenden genehmigt wurde. Der Projektantrag enthält keine Betriebsgeheimnisse. Soweit diese für die Antragstellung notwendig sind, wurden nach Rücksprache mit dem Ausbildenden die entsprechenden Stellen unkenntlich gemacht.

Mit dem Absenden des Projektantrages bestätige ich weiterhin, dass der Antrag eigenständig von mir angefertigt wurde. Ferner sichere ich zu, dass im Projektantrag personenbezogene Daten (d. h. Daten über die eine Person identifizierbar oder bestimmbar ist) nur verwendet werden, wenn die betroffene Person hierin eingewilligt hat.

Bei meiner ersten Anmeldung im Online-Portal wurde ich darauf hingewiesen, dass meine Arbeit bei Täuschungshandlungen bzw. Ordnungsverstößen mit „null“ Punkten bewertet werden kann. Ich bin weiter darüber aufgeklärt worden, dass dies auch dann gilt, wenn festgestellt wird, dass meine Arbeit im Ganzen oder zu Teilen mit der eines anderen Prüfungsteilnehmers übereinstimmt. Es ist mir bewusst, dass Kontrollen durchgeführt werden.

9 Grund für „mit Auflage genehmigt“

Guten Tag Tim Petersdorff,
bitte erläutern Sie in Ihrer Make-Or-Buy-Entscheidung, warum Sie die Anwendung selbst entwickelt haben und nicht auf eine fertige Lösung aufsetzen.

Mit freundlichen Grüßen
Ihr Prüfungsausschuss



Bestätigung über die durchgeführte Projektarbeit

(Diese Bestätigung ist mit der Projektdokumentation einzureichen!)

Berufsbezeichnung/Fachrichtung/Einsatzgebiet/Fachbereich:

Fachinformatiker für Anwendungsentwicklung

Prüfungsteilnehmer/-in: Tim Petersdorff

Ausbildungsbetrieb oder Umschulungsträger: Lehrwerk GmbH

Azubi-Identnummer:

Abschlussprüfung Sommer Winter 2025

Projektbezeichnung: Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Projektbeginn: 11.10.2025 Projektfertigstellung: 19.11.2025 Zeitaufwand in Stunden: 80

Bestätigung der Ausbildungsfirma: LEHRWERK GmbH

Wir bestätigen, dass der/die Auszubildende das oben bezeichnete Projekt einschließlich der Dokumentationselbstständig ausgeführt hat.

17.11.2025

38114 Braunschweig
Tel.: +49 531 590 631 00
Fax: +49 531 590 630 98
info@lehrwerk.de



Datum

Stempel der Firma

Unterschrift

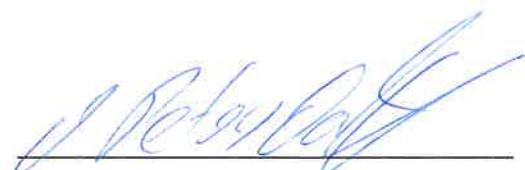
Die Dokumentation darf zu Unterrichtszwecken an der Berufsschule verwendet werden: JA NEIN

Eidesstattlich Erklärung:

Ich versichere, dass ich das Projekt und die dazugehörige Dokumentation selbstständig erstellt habe,

ATBS, 17.11.2025

Ort und Datum



Unterschrift des Prüflings



Abschlussprüfung Winter 2025

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

Lehrwerk Webapp

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Abgabedatum: Braunschweig, den 17.11.2025

Prüfungsbewerber:

Tim Petersdorff
Am Lehmann 1 B
38120 Braunschweig

Ausbildungsbetrieb:

Lehrwerk GmbH
Trautenastraße 10
38114 Braunschweig



LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis.....	I
Abbildungsverzeichnis.....	IV
Tabellenverzeichnis.....	V
Abkürzungsverzeichnis.....	VI
1 Einleitung	1
1.1 Projektumfeld	1
1.2 Projektziel	1
1.3 Projektbegründung.....	1
1.4 Projektschnittstellen	1
1.5 Projektabgrenzung.....	2
2 Projektplanung	2
2.1 Projektphasen.....	2
2.2 Abweichungen vom Projektantrag.....	3
2.3 Ressourcenplanung	3
2.4 Entwicklungsprozess	3
3 Analysephase.....	4
3.1 Ist-Analyse	4
3.2 Wirtschaftlichkeitsanalyse	4
3.2.1 Make-or-Buy-Entscheidung	4
3.2.2 Projektkosten	5
3.2.3 Amortisationsdauer	6
3.3 Nutzwertanalyse	6
3.4 Anwendungsfälle.....	7
3.5 Qualitätsanforderungen.....	7
3.6 Lastenheft/Fachkonzept.....	7

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Inhaltsverzeichnis

4	Entwurfsphase	8
4.1	Zielplattform	8
4.2	Architekturdesign	8
4.3	Entwurf der Benutzeroberfläche	8
4.4	Datenmodell	8
4.5	Geschäftslogik	9
4.6	Maßnahmen zur Qualitätssicherung	9
4.7	Pflichtenheft/Datenverarbeitungskonzept	9
5	Implementierungsphase	10
5.1	Implementierung der Datenstrukturen	10
5.2	Implementierung der Benutzeroberfläche	10
5.3	Implementierung der Geschäftslogik	10
6	Test- & Abnahmephase	11
7	Einführungsphase	11
8	Dokumentation	11
9	Fazit	12
9.1	Soll-/Ist-Vergleich	12
9.2	Lessons Learned	13
9.3	Ausblick	13
	Literaturverzeichnis	14
	Eidesstattliche Erklärung	15
	Anhang	i
A1	Detaillierte Zeitplanung	i
A2	Use-Case-Diagramm	iii
A3	Lastenheft	iv
A4	Sequenzdiagramm – Zugangscode eingeben	vi

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Inhaltsverzeichnis

A5	Pflichtenheft.....	vii
A6	Entity-Relationship-Model	x
A7	Relationales Datenmodell	xi
A8	Wireframes - Desktop und Mobile	xii
A9	Screenshots Frontend.....	xiv
A10	Screenshots Contentmanagement System	xviii
A11	EPK – Inhalte speichern im CMS	xxii
A12	SQL-Auszug Datenbankstruktur.....	xxiii
	Tabelle: courses.....	xxiii
	Tabelle: materials.....	xxiii
	Tabelle: users.....	xxiii
	Tabelle: content_entries	xxiv
	Tabelle: contacts	xxiv
	Tabelle: appointments	xxiv
A13	Verzeichnisstruktur – Root/Frontend	xxv
A14	Verzeichnisstruktur – Backend.....	xxvi
A15	Codebeispiel: JWT-Authentifizierung (requireAuth.js)	xxvii
A16	Codebeispiel: Spam-Schutz (spamGuard.js).....	xxviii
A17	Sicherheitstest / Rate-Limit-Test	xxix
A18	Integrationstest (Kontaktformular)	xxx
A19	Benutzerdokumentation (Auszüge) – Lehrwerk Webapp CMS.....	xxxi

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Abbildungsverzeichnis**Abbildungsverzeichnis**

Abbildung 1: Use-Case-Diagramm	iii
Abbildung 2: Sequenzdiagramm.....	vi
Abbildung 3: Entity-Relationship-Modell	x
Abbildung 4: Relationales Datenmodell	xi
Abbildung 5: Wireframe Desktop	xii
Abbildung 6: Wireframe Mobil.....	xiii
Abbildung 7: Heropage Desktop.....	xiv
Abbildung 8: Heropage Mobil	xv
Abbildung 9: Heropage Responsive	xvi
Abbildung 10: Kursangebote Übersicht Desktop	xvii
Abbildung 11: Zugangscodeüberprüfung Desktop	xvii
Abbildung 12: Loginmaske CMS	xviii
Abbildung 13: Benachrichtigungen im CMS.....	xix
Abbildung 14: Content bearbeiten im CMS.....	xix
Abbildung 15: Bilder verwalten im CMS.....	xx
Abbildung 16: Materialien verwalten im CMS	xx
Abbildung 17: Möglichkeit zur Passwortänderung im CMS.....	xxi
Abbildung 18: Ereignisprozesskette.....	xxii
Abbildung 19: Projektstruktur Root	xxv
Abbildung 20: Projektstruktur Backend.....	xxvi
Abbildung 21: Codebeispiel Authentifizierung.....	xxvii
Abbildung 22: Codebeispiel Spamschutz	xxviii
Abbildung 23: Sicherheitstest Rate-Limit	xxix
Abbildung 24: Eingabe und Absenden der Formulardaten.....	xxx
Abbildung 25: Protokoll des erfolgreichen Datentransfers	xxx

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Tabellenverzeichnis

Tabellenverzeichnis

Tabelle 1: Grobe Zeitplanung	2
Tabelle 2: Kostenaufstellung	5
Tabelle 3: Entscheidungsmatrix.....	6
Tabelle 4: Soll-/Ist-Vergleich.....	12
Tabelle 5: Detaillierte Zeitplanung	ii

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Abkürzungsverzeichnis

Abkürzungsverzeichnis

API	Application Programming Interface
CMS	Content-Management-System
CORS	<i>Cross-Origin Resource Sharing</i>
CRUD	<i>Create, Read, Update, Delete</i>
CSS	<i>Cascading Style Sheets</i>
DB	<i>Database</i>
EPK	<i>Ereignisgesteuerte Prozesskette</i>
ERM	<i>Entity Relationship Model</i>
HTML	<i>Hypertext Markup Language</i>
JS	<i>JavaScript</i>
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Token</i>
MVC	<i>Model View Controller</i>
NPM	<i>Node Package Manager</i>
REST	<i>Representational State Transfer</i>
SQL	<i>Structured Query Language</i>
UI	<i>User Interface</i>
UX	<i>User Experience</i>
VPS	<i>Virtual Private Server</i>
Express.js	Minimalistisches und flexibles Webframework
Node.js	<i>Serverseitige JavaScript-Laufzeitumgebung</i>
TailwindCSS	<i>Utility-first CSS-Framework</i>
Vite	<i>Frontend-Build-Tool für moderne Webanwendungen</i>

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Einleitung

1 Einleitung

1.1 Projektumfeld

Die Lehrwerk GmbH ist ein Bildungsträger aus Braunschweig, der IT-Umschulungen und Weiterbildungen in den Fachrichtungen der Fachinformatik sowie im kaufmännischen IT-Bereich anbietet. Das Unternehmen ist klein strukturiert und beschäftigt weniger als zehn Mitarbeitende. Im Zuge der digitalen Modernisierung wurde beschlossen, die bisherige Onlinepräsenz und interne Abläufe zu optimieren. Der Auftrag zur Umsetzung erfolgte durch den Geschäftsführer und Dozenten der Lehrwerk GmbH, Herrn Ulrich Meißner.

1.2 Projektziel

Ziel ist die Entwicklung einer modernen Full-Stack-Webanwendung für die Lehrwerk GmbH. Sie dient als zentrale Plattform für Interessenten und Teilnehmende. Interessenten informieren sich über Bildungsangebote, stellen Kontaktanfragen und buchen Termine. Teilnehmende erhalten Zugriff auf digitale Lehrmaterialien. Das integrierte CMS ermöglicht es den Mitarbeitern, Inhalte selbstständig zu pflegen und eingegangene Anfragen zu verwalten. Damit vereinfacht sich die Kommunikation und Verwaltung für alle Beteiligten.

1.3 Projektbegründung

Die bisherige Website bot keine Möglichkeit zur zentralen Inhaltsverwaltung oder automatisierten Anfrageverarbeitung. Dies führte zu erhöhtem Verwaltungsaufwand und umständlicher Kommunikation. Die neue Anwendung automatisiert diese Prozesse und präsentiert die Lehrwerk GmbH zeitgemäß. Das Projekt bietet zudem die Gelegenheit, moderne Webtechnologien praktisch umzusetzen und ein vollständiges Full-Stack-System eigenständig zu entwickeln.

1.4 Projektschnittstellen

Die Anwendung basiert auf einer dreischichtigen Architektur. Das Frontend wird mit HTML, Tailwind CSS und Vite im Browser bereitgestellt. Das Backend läuft auf Node.js mit Express.js und verwaltet eine MariaDB-Datenbank. Kontakt- und Terminformulare werden direkt in der Datenbank gespeichert und sind im CMS bearbeitbar. Das Projekt wurde auf einem Linux-VPS entwickelt und wird auf dem Server der Lehrwerk GmbH deployed. Nutzergruppen sind

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Projektplanung

Interessenten, Teilnehmende und Administratoren, die über das CMS Inhalte und Anfragen verwalten.

1.5 Projektabgrenzung

Das Projekt beschränkt sich auf die Entwicklung der Webanwendung mit integriertem CMS. Erweiterungen, externe Schnittstellen und spätere Wartungsaufgaben sind nicht Bestandteil des Projekts.

2 Projektplanung

2.1 Projektphasen

Das Projekt wird im Zeitraum vom 11.10.2025 bis 19.11.2025 durchgeführt. Die Gesamtbearbeitungszeit beträgt 80 Stunden, die flexibel über diesen Zeitraum verteilt werden. Die Einteilung orientiert sich an den im Projektantrag definierten Phasen und folgt einer klaren zeitlichen Struktur. Eine detailliertere Zeitplanung ist in Tabelle 5 in Anhang A1 zu sehen.

Projektphase	Geplante Zeit
Analyse	6 h
Entwurf	14 h
Implementierung	40 h
Test & Abnahme	7 h
Deployment & Einführung	4 h
Dokumentation	9 h
Gesamt	80 h

Tabelle 1: Grobe Zeitplanung

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Projektplanung

2.2 Abweichungen vom Projektantrag

Die Projektdurchführung erfolgte im Wesentlichen entsprechend dem genehmigten Projektantrag. Es ergaben sich lediglich leichte zeitliche Abweichungen bei einzelnen Aufgabenpaketen. Eine inhaltliche Abweichung betrifft die Einführungsphase. Das Deployment auf den produktiven Server konnte aufgrund von Verzögerungen bei der Infrastrukturbereitstellung nicht durchgeführt werden. Die Anwendung ist jedoch vollständig produktionsreif und kann unmittelbar nach Verfügbarkeit des Servers deployed werden. Die detaillierte Gegenüberstellung der geplanten und der tatsächlich benötigten Zeiten ist dem Soll-Ist-Vergleich in Kapitel 9.1 zu entnehmen.

2.3 Ressourcenplanung

Für die Umsetzung stand die erforderliche Hard- und Software bereit. Entwickelt wurde auf einem Linux-VPS mit Node.js und MariaDB. Als Tools dienten Visual Studio Code, Vite und Tailwind CSS im Frontend, sowie Express.js im Backend. Die Datenbankadministration erfolgte per SSH/Bash-Shell. Arbeitsplatz, Geräte und Internetzugang stellte die Lehrwerk GmbH, organisatorische Unterstützung erfolgte durch Herrn Ulrich Meißner.

2.4 Entwicklungsprozess

Das Projekt wurde überwiegend nach dem Wasserfallmodell umgesetzt und durch einzelne agile Elemente ergänzt. Die Bearbeitung erfolgte in klar abgegrenzten Phasen: Analyse, Entwurf, Implementierung, Test und Einführung. Zur Qualitätssicherung und frühzeitigen Berücksichtigung von Anpassungen fanden wöchentliche Feedback- und Abstimmungsrunden mit dem Auftraggeber statt. Dieses Vorgehen verband die Vorteile strukturierter Planung mit der notwendigen Flexibilität agiler Methoden.

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Analysephase

3 Analysephase

3.1 Ist-Analyse

Vor Projektbeginn verfügte die Lehrwerk GmbH über eine veraltete Website auf statischen HTML-Seiten ohne serverseitige Logik oder Datenbankanbindung. Inhalte konnten nicht zentral verwaltet werden und Aktualisierungen erforderten manuelle Eingriffe durch technisch versierte Personen. Eine regelmäßige Pflege durch Mitarbeitende war praktisch unmöglich. Kontaktanfragen und Terminvereinbarungen liefen ausschließlich über E-Mail oder Telefon ab, was zu Verzögerungen und erhöhtem Verwaltungsaufwand führte. Lehrmaterialien waren nicht digital verfügbar. Der digitale Auftritt entsprach weder modernen technischen Standards noch den Anforderungen an Benutzerfreundlichkeit und Effizienz.

3.2 Wirtschaftlichkeitsanalyse

Das Projekt bietet der Lehrwerk GmbH einen deutlichen wirtschaftlichen Nutzen. Durch die zentrale Verwaltung von Inhalten und Anfragen wird der organisatorische Aufwand reduziert. Wiederkehrende Aufgaben wie das manuelle Bereitstellen von Materialien entfallen. Die eigenentwickelte Lösung verursacht keine laufenden Lizenzkosten, kann flexibel erweitert werden und steigert langfristig die Außenwirkung des Unternehmens.

3.2.1 Make-or-Buy-Entscheidung

Für die neue Webanwendung wurde eine Eigenentwicklung gewählt, da fertige CMS-Systeme und Baukastentools die spezifischen Anforderungen der Lehrwerk GmbH nur eingeschränkt erfüllen. Kommerzielle Lösungen bieten meist keine flexible Integration von individueller Termin- und Kontaktverwaltung, Datenbankstrukturen oder anpassbarer Materialbereitstellung. Wesentliche Nachteile fertiger Lösungen sind zudem die laufenden Lizenzkosten und die damit verbundenen Abhängigkeiten von externen Anbietern, was das Risiko von Preisanpassungen oder Funktionseinschränkungen birgt. Durch die Eigenentwicklung kann die Anwendung exakt auf interne Abläufe und Bedürfnisse zugeschnitten werden. Dies ermöglicht die nahtlose Integration von Funktionen und die vollständige Datenhoheit, welche die Einhaltung der DSGVO-Anforderungen gewährleistet. Zudem ist die Lösung langfristig kosteneffizienter und schafft eine zukunftssichere, skalierbare Basis, da Änderungen und Wartung künftig eigenständig durchgeführt werden können.

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Analysephase

3.2.2 Projektkosten

Da der Lehrwerk GmbH durch die externe Finanzierung keine direkten Personalkosten entstehen, werden die Kosten fiktiv berechnet. Dies dient der realistischen Einschätzung des wirtschaftlichen Aufwands der Eigenentwicklung. Die Kalkulation basiert auf dem zeitlichen Aufwand (inkl. Abstimmungen), der Infrastruktturnutzung und dem Tariflohn eines Auszubildenden im dritten Lehrjahr (1200€ Brutto/Monat).

$$8 \frac{\text{h}}{\text{Tag}} \cdot 220 \frac{\text{Tage}}{\text{Jahr}} = 1.760 \frac{\text{h}}{\text{Jahr}}$$

$$1.200 \frac{\text{€}}{\text{Monat}} \cdot 13,3 \frac{\text{Monate}}{\text{Jahr}} = 15.960 \frac{\text{€}}{\text{Jahr}}$$

$$\frac{15.960 \frac{\text{€}}{\text{Jahr}}}{1.760 \frac{\text{h}}{\text{Jahr}}} \approx 9,07 \frac{\text{€}}{\text{h}}$$

Die Gesamtkosten des 80-stündigen Projekts betragen 2.090 EUR (siehe Tabelle 2). Diese Kalkulation basiert auf einem Stundensatz von 9,07 EUR für die Durchführung sowie Pauschalsätze von 15 EUR für die Ressourcennutzung und 35 EUR für andere Mitarbeiter.

Vorgang	Zeit	Kosten / Stunde	Kosten
Analyse, Entwurf, Implementierung, Test	80 h	$9 \text{ €} + 15 \text{ €} = 24 \text{ €}$	1.920,00 €
Betreuung durch Dozenten	2 h	35€	70,00€
Abnahme Auftraggeber	2 h	$35 \text{ €} + 15 \text{ €} = 50 \text{ €}$	100,00€
Gesamt			2.090,00€

Tabelle 2: Kostenaufstellung

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Analysephase

3.2.3 Amortisationsdauer

Die automatisierte Verwaltung von Anfragen, Terminen und Materialien spart dem Sekretariat monatlich etwa 5 Stunden. Hinzu kommt die Einsparung von Lizenzkosten. Damit amortisiert sich die Investition wie folgt:

$$5 \frac{\text{h}}{\text{Monat}} * 12 = 60 \frac{\text{h}}{\text{jahr}} \quad 60\text{h} \cdot 25 \frac{\text{€}}{\text{h}} = 1500\text{€}$$

$$\frac{2.090 \text{ €}}{1.500 \text{ €}} \approx 1,4 \text{ Jahre}$$

3.3 Nutzwertanalyse

Es wurde eine Nutzwertanalyse durchgeführt, um die Entscheidung zwischen einer Eigenentwicklung und einem fertigen CMS-System unter nicht-monetären Gesichtspunkten zu begründen. Die Bewertung erfolgt auf einer Skala von 0 bis 3 Punkten. Der Nutzenwert berechnet sich abschließend aus dem Produkt von Gewichtung × Bewertungspunkte (siehe Tabelle 3).

Eigenschaft	Gewichtung	Fertiges CMS	Eigenentwicklung
Anpassbarkeit	6	6	18
Datenschutz/	5	10	15
Kontrolle			
Skalierbarkeit	5	10	15
Technische Abhängigkeit	4	4	12
Wartungsaufwand	2	6	4
Lern- /	2	4	2
Schulungsaufwand			
Gesamt	24	40	66
Nutzwert			

Tabelle 3: Entscheidungsmatrix

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Analysephase

Der Nutzwert ergibt sich aus der gewichteten Bewertung der Kriterien. Die Eigenentwicklung erreicht mit 66 Punkten einen deutlich höheren Wert als das fertige CMS mit 40 Punkten. Ausschlaggebend ist die bessere Anpassbarkeit, die vollständige Kontrolle über Datenschutz und Datenhaltung sowie die hohe Skalierbarkeit und Unabhängigkeit von externen Systemen. Damit bestätigt die Nutzwertanalyse die Entscheidung für eine eigene Full-Stack-Entwicklung, da sie langfristig mehr Flexibilität und Zukunftssicherheit bietet.

3.4 Anwendungsfälle

Die zentralen Anwendungsfälle der Lehrwerk Webapp sind im Use-Case-Diagramm (siehe Anhang A2) dargestellt. Interessenten können Kursinformationen einsehen und Kontakt- oder Terminanfragen absenden. Teilnehmer authentifizieren sich über einen Zugangscode, um auf geschützte Lehrmaterialien zuzugreifen. Bei fehlerhaften Eingaben erhalten sie eine Fehlermeldung. Administratoren verwalten Inhalte, Anfragen und Materialien zentral über das CMS. Validierung, Datenverarbeitung und Speicherung sind dabei zentrale Systemfunktionen.

3.5 Qualitätsanforderungen

Bei der Entwicklung wurde Wert auf Stabilität, Effizienz und Benutzerfreundlichkeit gelegt. Die Anwendung validiert Eingaben, schützt Zugriffe und verarbeitet Daten sicher. Das Frontend ist responsiv und lädt schnell durch optimierte Datenbankabfragen und asynchrone Kommunikation. Die modulare Backend-Struktur ermöglicht einfache Wartung und Erweiterungen.

3.6 Lastenheft/Fachkonzept

Das Lastenheft (siehe Anhang A3) definiert die fachlichen und funktionalen Anforderungen an die Lehrwerk Webapp aus Sicht des Auftraggebers. Das Dokument bildet die Grundlage für die technische Planung und die Erstellung des Pflichtenhefts in der Entwurfsphase.

Entwurfsphase

4 Entwurfsphase

4.1 Zielplattform

Die Lehrwerk Webapp ist eine plattformunabhängige Webanwendung, die über gängige Browser auf Desktop- und Mobilgeräten nutzbar ist. Das Backend basiert auf Node.js mit Express.js, das Frontend auf HTML, Tailwind CSS und Vite. Die Datenbank ist in MariaDB realisiert. Die Anwendung folgt dem Client-Server-Prinzip und stellt eine REST-API bereit, über die Clients per HTTPS auf Daten zugreifen. Systemvoraussetzungen: Linux- oder Windows-Server mit Node.js, 2 CPU-Kernen und 2 GB RAM.

4.2 Architekturdesign

Die Lehrwerk Webapp basiert auf einer modularen MVC-Architektur mit Express.js und Node.js. Controller verarbeiten Anfragen und liefern Daten als JSON. Middleware-Komponenten gewährleisten Authentifizierung und Zugriffskontrolle. Die Datenhaltung erfolgt in einer relationalen MariaDB-Datenbank. Das Frontend nutzt HTML mit Tailwind CSS und Vite und lädt Daten über REST-API-Endpunkte.

4.3 Entwurf der Benutzeroberfläche

Die Benutzeroberfläche wurde als moderne, klare Webanwendung konzipiert, die in allen gängigen Browsern nutzbar ist. Die Navigation umfasst die Hauptbereiche Start, Kursangebote, Über-uns, Kontakt und FAQ. Ein prominenter Button auf der Startseite führt zum geschützten Materialbereich für Teilnehmer. Das Design folgt dem Corporate-Design der Lehrwerk GmbH und wurde mit Tailwind CSS umgesetzt. Hohe Kontraste, große Klickflächen und klare Typografie sichern Lesbarkeit und Bedienbarkeit auf allen Endgeräten. Der Administrationsbereich unter /admin.html ist nur für autorisierte Benutzer zugänglich. Screenshots und Entwürfe der Oberfläche befinden sich in Anhang A8–A10.

4.4 Datenmodell

Das Datenmodell basiert auf einer relationalen MariaDB-Datenbank mit logisch getrennten Tabellen. Ziel ist eine strukturierte, redundanzarme Datenhaltung. Die Tabelle contacts verwaltet Kontaktanfragen aus Formularen, appointments speichert Terminbuchungen, materials enthält Lehrmaterialien mit optionaler Kurszuordnung, content_entries verwaltet dynamische

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Entwurfsphase

Website-Inhalte im JSON-Format, users authentifiziert Administratoren und courses definiert die verfügbaren Kurse. Beziehungen zwischen den Tabellen werden durch Fremdschlüssel gewährleistet. Das vollständige Entity-Relationship-Modell mit Attributen und Kardinalitäten ist in Anhang A6 dargestellt.

4.5 Geschäftslogik

Die Geschäftslogik umfasst zentrale Backend-Prozesse: Authentifizierung von Administratoren, Validierung von Eingaben, Datei-Uploads sowie Verarbeitung von Kontakt- und Terminformularen. Diese Prozesse gewährleisten, dass Daten korrekt geprüft, sicher gespeichert und strukturiert an das Frontend übermittelt werden. Eine detaillierte Darstellung des Ablaufs „Inhalte speichern im CMS“ mit Authentifizierung, Validierung und Speicherung findet sich in der ereignisgesteuerten Prozesskette (EPK) in Anhang A11.

4.6 Maßnahmen zur Qualitätssicherung

Zur Sicherstellung der Funktionsfähigkeit wurde während der Entwicklung kontinuierlich getestet. Nach jedem Modul erfolgte eine Überprüfung im Browser unter realistischen Bedingungen. Getestet wurden die zentralen Funktionen: CMS-Login, Speichern und Laden von Inhalten, Datei-Uploads, Materialverwaltung sowie Kontakt- und Terminformulare. Ungültige Eingaben wurden gezielt ausgelöst, um die Fehlerbehandlung zu verifizieren. Zusätzlich wurde die Darstellung auf verschiedenen Geräten überprüft. Identifizierte Fehler wurden behoben und erneut getestet, bis die Anwendung stabil lief.

4.7 Pflichtenheft/Datenverarbeitungskonzept

Das Pflichtenheft konkretisiert die im Lastenheft beschriebenen Anforderungen und dient als Grundlage für die technische Umsetzung der Lehrwerk Webapp. Dieses ist im Anhang A5 zu finden.

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Implementierungsphase

5 Implementierungsphase

5.1 Implementierung der Datenstrukturen

Die Datenbank wurde in MariaDB basierend auf dem Entity-Relationship-Modell (Anhang A6) und dem daraus abgeleiteten relationalen Datenmodell (Anhang A7) umgesetzt. Jede Entität wurde als eigene Tabelle realisiert, Beziehungen durch Primär- und Fremdschlüssel abgebildet. Das System umfasst sechs Tabellen: courses für die Kursverwaltung, materials für Lehrmaterialien mit Kurszuordnung, users für die Admin-Authentifizierung, content_entries für dynamische Website-Inhalte, contacts für Kontaktanfragen und appointments für Terminbuchungen. Durchgängig wird der Zeichensatz utf8mb4 verwendet, um internationale Zeichensätze zu unterstützen. Die Tabellenstrukturen und Beziehungen sind im SQL-Auszug (Anhang A12) dokumentiert.

5.2 Implementierung der Benutzeroberfläche

Die Benutzeroberfläche wurde mit HTML, CSS und JavaScript umgesetzt und mit Vite modular aufgebaut. Tailwind CSS gewährleistet ein responsives Design nach dem Corporate Design der Lehrwerk GmbH. Die Anwendung ist in zwei Bereiche unterteilt: einen öffentlichen Bereich mit Informationsseiten (Start, Kursangebote, Über uns, Kontakt, FAQ), sowie einen geschützten Administrations- und Lernmaterialbereich. Interaktive Funktionen wie Navigation, Formularvalidierung und Datei-Uploads wurden mit JavaScript implementiert. Daten werden dynamisch über REST-API-Endpunkte geladen. Screenshots befinden sich in Anhang A9 und A10, die Verzeichnisstruktur in Anhang A13.

5.3 Implementierung der Geschäftslogik

Die Backend-Logik wurde mit Node.js und Express implementiert und steuert die Datenverarbeitung zwischen Frontend und Datenbank. Jede Funktion ist in einer separaten Router-Datei gekapselt. Die Routen validieren Anfragen, führen Datenbankabfragen durch und geben strukturierte Responses zurück. Die Authentifizierung erfolgt über JWT (JSON Web Tokens), während öffentliche Formulare durch Rate-Limiting vor Missbrauch geschützt sind. Datei-Uploads werden serverseitig validiert und im Verzeichnis storage/materials gespeichert. Die Backend-Struktur ist in Anhang A14 dokumentiert.

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Test- & Abnahmephase

6 Test- & Abnahmephase

In der Test- und Abnahmephase wurden Funktionsfähigkeit, Stabilität und Sicherheit überprüft. Ein Integrationstest des Kontaktformulars bestätigte die fehlerfreie Kommunikation zwischen Frontend und Backend (Anhang A18). Der Rate-Limiter wurde durch wiederholte Anfragen getestet und reagierte korrekt mit einem Sperrstatus (Anhang A17). Zusätzlich wurden manuelle Tests aller Hauptfunktionen durchgeführt: CMS-Login, Inhalte speichern/laden, Materialverwaltung und Datei-Uploads. Alle Funktionen arbeiteten fehlerfrei. Die Anwendung wurde daraufhin für den Produktivbetrieb freigegeben.

7 Einführungsphase

Nach erfolgreichem Test war die Lehrwerk Webapp produktionsreif. Das Frontend wurde mit Vite für den Produktivbetrieb optimiert, das Backend-System vollständig getestet und die Datenbankstruktur validiert. Sämtliche technischen Anforderungen sind erfüllt. Das Deployment auf dem produktiven Webserver konnte noch nicht durchgeführt werden, da die Bereitstellung des Servers durch ein paralleles Projekt verzögert wurde. Alle Komponenten sind jedoch bereit und können unmittelbar nach Verfügbarkeit des Servers deployed werden. Schulungen zur Systemnutzung erfolgen nach erfolgreichem Deployment.

8 Dokumentation

Neben dieser Projektdokumentation entstand eine Benutzerdokumentation für das CMS. Sie beschreibt die Verwaltung von Kontakten, Terminen, Materialien und Inhalten und dient als praxisnahe Anleitung für die Nutzung des Systems. Auszüge der Benutzerdokumentation sind Anhang 19 zu entnehmen.

Fazit

9 Fazit

9.1 Soll-/Ist-Vergleich

Die geplanten 80 Stunden wurden vollständig eingehalten. In der Entwurfsphase konnte durch die Nutzung bewährter Technologien (Express.js, Node.js) eine Stunde eingespart werden. Die Testphase benötigte zusätzliche Zeit, da die Upload-Funktion und der Rate-Limiter gründlich getestet wurden – dies resultierte in einer robusteren Anwendung. Das Deployment konnte aufgrund von Verzögerungen bei der Infrastrukturbereitstellung vorbereitet, aber nicht vollständig durchgeführt werden, wodurch zwei Stunden eingespart wurden. Die dadurch gewonnene Zeit wurde in die Dokumentation investiert, um eine ausführliche Benutzerdokumentation bereitzustellen. Insgesamt wurde das Projekt im geplanten Zeitrahmen von 80 Stunden abgeschlossen.

Phase	Geplant	Tatsächlich	Differenz
Analyse	6 h	6 h	
Entwurf	14 h	13 h	- 1 h
Entwicklung & Implementierung	40 h	40 h	
Test & Abnahme	7 h	9 h	+ 2 h
Deployment & Einführung	4 h	2 h	- 2 h
Dokumentation	9 h	10 h	+ 1 h
Gesamt	80 h	80 h	

Tabelle 4: Soll-/Ist-Vergleich

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Fazit

9.2 Lessons Learned

Durch das Projekt wurde deutlich, dass eine gute Architektur von Anfang an Zeit spart. Die Entscheidung, Router-Dateien zu separieren, hat sich bewährt und macht Wartung einfacher. Bei JWT wurde erkannt, dass es anfangs kompliziert wirkt, sich aber später auszahlt, besser als ständig Sessions zu verwalten. Die Zeit für UI/UX wurde unterschätzt. Mit mehr Designarbeit wäre das Frontend noch benutzerfreundlicher geworden. Dies sollte in zukünftigen Projekten früher eingeplant werden. Ein weiterer wichtiger Aspekt waren externe Abhängigkeiten wie die Serverbereitstellung, die nicht ausreichend im Risikomanagement berücksichtigt wurden. Das Projekt selbst war produktionsreif, konnte aber ohne verfügbaren Server nicht live gehen. Dies war ein wichtiger Lernpunkt für zukünftige Projektplanungen.

9.3 Ausblick

Nach der Produktivbereitstellung könnten folgende Erweiterungen umgesetzt werden: Ein integriertes Wiki würde Kursinhalte dokumentieren und Teilnehmern zusätzliche Lernressourcen bieten. Ein Analyse-Dashboard könnte Administratoren Einblicke in Teilnehmerzahlen und Zugriffe geben. Diese Erweiterungen würden die Plattform zu einem zentralen Verwaltungssystem für die Lehrwerk GmbH entwickeln.

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Literaturverzeichnis

Literaturverzeichnis

Node.js Foundation, 2024. *Node.js Documentation*. [Online] - Verfügbar unter: <https://nodejs.org/en/docs/>

Express.js Team, 2024. *Express – Node.js Web Application Framework*. [Online] - Verfügbar unter: <https://expressjs.com/>

MariaDB Foundation, 2024. *MariaDB Documentation*. [Online] - Verfügbar unter: <https://mariadb.com/kb/en/documentation/>

Tailwind Labs, 2024. *Tailwind CSS Documentation*. [Online] - Verfügbar unter: <https://tailwindcss.com/docs>

Vite, 2024. *Vite – Next Generation Frontend Tooling*. [Online] - Verfügbar unter: <https://vitejs.dev/guide/>

Mozilla Developer Network (MDN), 2024. *Web APIs & JavaScript Reference*. [Online] - Verfügbar unter: <https://developer.mozilla.org/de/docs/Web>

Npm Inc., 2024. *npm Documentation*. [Online] - Verfügbar unter: <https://docs.npmjs.com/>

GitHub, 2024. *GitHub Docs – Version Control and Collaboration*. [Online] - Verfügbar unter: <https://docs.github.com/>

W3C, 2024. *HTML and CSS Standards*. [Online] - Verfügbar unter: <https://www.w3.org/standards/webdesign/htmlcss>

OpenJS Foundation, 2023. *JavaScript Best Practices*. [Online] - Verfügbar unter: <https://openjsf.org/>

IEEE, 2022. *Guide to Software Engineering Standards*. [Online] - Verfügbar unter: <https://ieeexplore.ieee.org/>

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Eidesstattliche Erklärung

Eidesstattliche Erklärung

Ich, Tim Petersdorff, versichere hiermit, dass ich meine Dokumentation zur betrieblichen Projektarbeit mit dem Thema

Lehrwerk Webapp – Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Abgabeort, den 17.11.2025



TIM PETERSDORFF

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

Anhang

A1 Detaillierte Zeitplanung

Analysephase	6 h
1. Analyse des Ist-Zustands	1 h
2. Anforderungen mit Stakeholdern klären (Zieldefinition, Interview)	2 h
3. Pflichtenheft erstellen und abstimmen	2 h
4. Technologien auswählen (Node.js, Express, Vite, Tailwind, MariaDB)	1 h
Entwurfsphase	14 h
1. Systemarchitektur erstellen (Frontend/Backend-Trennung, APIs)	3 h
2. Datenbankentwurf	3 h
2.1. ER-Modell erstellen	2 h
2.2. Relationales Datenmodell erstellen	1 h
3. Erstellung der Diagramme (Use-Case, EPK, Sequenzdiagramm, Wireframes)	3 h
4. UI/UX-Design und Layout (Farben, Navigation, Responsivität)	3 h
5. Sicherheitskonzept entwickeln (Validierung, Rate-Limit, Passwortregeln)	2 h
Entwicklungs- & Implementierungsphase	40 h
1. Entwicklungsumgebung einrichten (Node.js, Git, DB)	2 h
2. Datenbank erstellen und befüllen	3 h
3. Backend-Implementierung (Express-Router, API-Endpoints, CRUD-Logik)	10 h
4. Frontend-Implementierung (UI, JSON-Lade- und Speicherfunktionen)	10 h

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

5. Upload- und Download-Funktionen (Materialien, Bilder) umsetzen	3 h
6. Sicherheitsfunktionen (Rate-Limit, Validierung, Sessionhandling)	3 h
7. Verbindung Frontend ↔ Backend (Fetch-API, Response-Handling)	4 h
8. Fehlerbehandlung, Logging und Optimierung	3 h
9. Feinschliff & Layout-Anpassungen (Responsives Design, UX-Verbesserungen)	2 h
Test- & Abnahmephase 7 h	
1. Funktionstests im Backend (API, Validierung)	2 h
2. Integrationstests (Formulare, Uploads, JSON-Handling)	2 h
3. Systemtest mit realen Beispieldaten	1 h
4. Abnahme anhand Pflichtenheft / Feedback Auftraggeber	2 h
Einführungsphase 4 h	
1. Produktivumgebung auf Webserver einrichten (Node-Service, DB)	2 h
2. Anwendung bereitstellen und Domain umstellen	1 h
3. Funktionstest auf Produktivsystem	1 h
Erstellen der Dokumentation 9 h	
1. Benutzerdokumentation verfassen	3 h
2. Erstellen der Projektdokumentation	6 h
Gesamt 80 h	

Tabelle 5: Detaillierte Zeitplanung

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A2 Use-Case-Diagramm

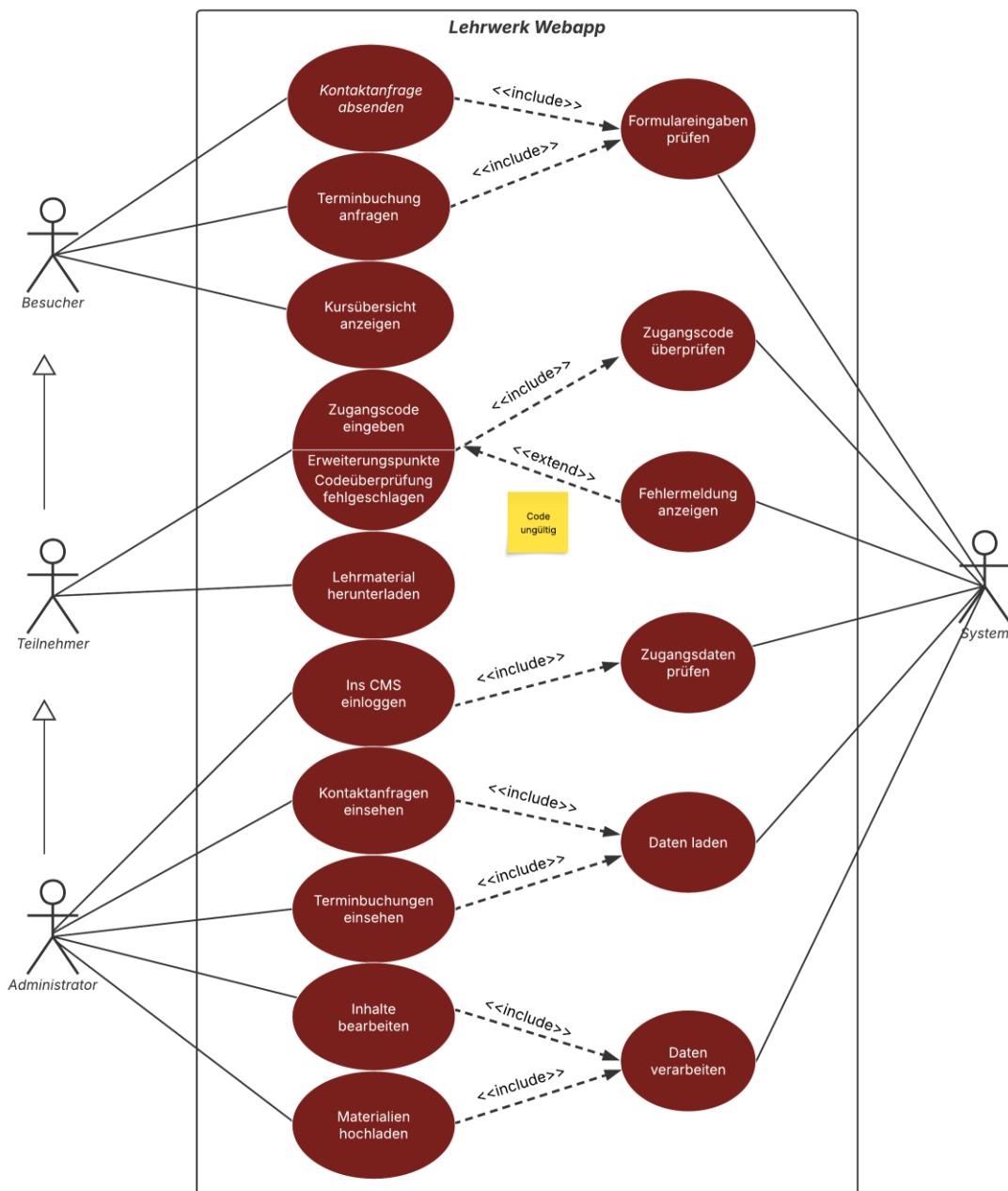


Abbildung 1: Use-Case-Diagramm

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A3 Lastenheft

1. Ziel

Die Lehrwerk Webapp soll eine zentrale Informations- und Verwaltungsplattform für den Bildungsträger bereitstellen. Sie ermöglicht Interessenten, Teilnehmern und Administratoren den Zugriff auf relevante Inhalte, Materialien und Kommunikationsfunktionen. Ziel ist eine moderne, responsive Website, deren Inhalte durch das Lehrwerk-Team selbst gepflegt werden können.

2. Funktionale Anforderungen

2.1 Öffentlicher Bereich (Homepage)

1. Die Anwendung muss grundlegende Informationen wie Unterrichtszeiten, Lernphase, Kursinhalte und Fördermöglichkeiten anzeigen.
2. Es sollen Zusatzinformationen dargestellt werden, z. B. Parkplätze, Busverbindungen, Zertifizierungen und Starttermine.
3. Telefonnummer und E-Mail-Adresse müssen anklickbar und auf mobilen Geräten direkt nutzbar sein.
4. Das Lehrwerk-Logo und die Navigation sollen beim Scrollen sichtbar bleiben.
5. Ein Kontaktformular mit Eingabevalidierung ist bereitzustellen.
6. Eine Terminbuchungsfunktion für Beratungsgespräche muss vorhanden sein.
7. Google Maps ist zur Standortanzeige einzubinden.
8. Eine Bildergalerie der Unterrichtsräume und Lernbereiche soll integriert werden.

2.2 Teilnehmerbereich

1. Teilnehmer müssen über einen Zugangscode Zugriff auf geschützte Lehrmaterialien erhalten.
2. Der Zugangscode ist serverseitig zu prüfen.
3. Bei fehlerhaftem Code wird eine Fehlermeldung angezeigt.

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

4. Bei erfolgreicher Prüfung werden Materialien (z. B. PDFs, Präsentationen) angezeigt.

2.3 Administrationsbereich (CMS)

1. Administratoren müssen sich über einen Login anmelden können.
2. Inhalte und Texte sollen direkt über das CMS bearbeitbar sein.
3. Uploads von Materialien und Bildern müssen möglich sein.
4. Kontakt- und Terminbuchungen sind im CMS einseh- und verwaltbar.
5. Starttermine und allgemeine Informationen sollen über das CMS anpassbar sein.

3. Nicht-funktionale Anforderungen

1. Responsives Design für Desktop, Tablet und Smartphone.
2. Intuitive Bedienbarkeit und klare Benutzerführung.
3. Kurze Ladezeiten und stabile Performance.
4. Datenschutzkonforme Verarbeitung von Formular- und Nutzerdaten gemäß DSGVO.
5. Plattformunabhängiger Betrieb im Browser.
6. Pflege der Inhalte ohne technische Kenntnisse möglich.

4. Abnahmekriterien

1. Alle im Lastenheft genannten Funktionen sind vollständig implementiert und funktionsfähig.
2. Die Anwendung ist responsiv und benutzerfreundlich.
3. Die Zugangscode-Prüfung funktioniert zuverlässig.
4. Inhalte sind über das CMS ohne Programmierkenntnisse änderbar.
5. Kontakt- und Terminformulare liefern korrekte Eingaben und Fehlermeldungen.

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A4 Sequenzdiagramm – Zugangscode eingeben

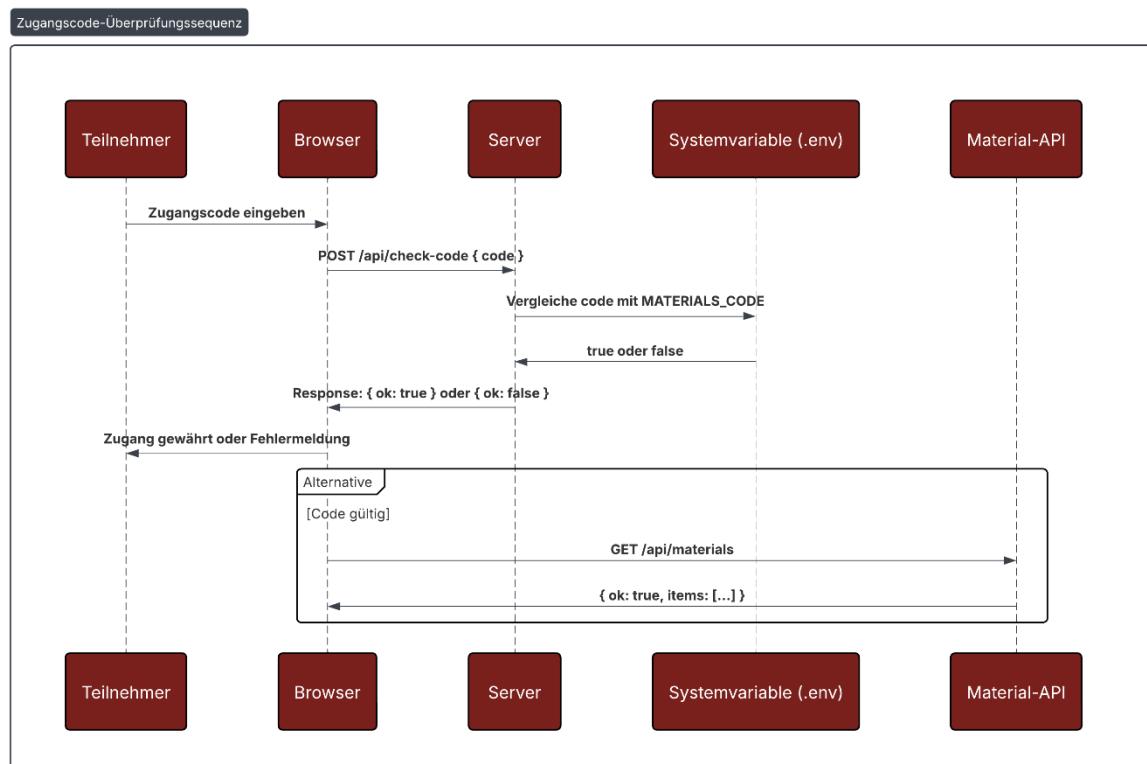


Abbildung 2: Sequenzdiagramm

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A5 Pflichtenheft

1. Zielbestimmung

Das Pflichtenheft beschreibt die technische Umsetzung der im Lastenheft definierten Anforderungen für die Lehrwerk Webapp. Es dient als verbindliche Grundlage für die Implementierung und Qualitätssicherung der Anwendung.

2. Musskriterien

2.1 CMS-Funktionalität

2.1.1 Inhaltsverwaltung

Über das Content-Management-System können Inhalte der Website wie Kursbeschreibungen, FAQs, rechtliche Hinweise und Texte der Startseite bearbeitet und gespeichert werden. Die Änderungen werden in JSON-Dateien abgelegt und stehen nach dem Speichern im Frontend automatisch zur Verfügung.

2.1.2 Authentifizierung

Der Zugriff auf das CMS ist ausschließlich für berechtigte Administratoren möglich. Die Anmeldung erfolgt über ein Login-System mit JWT-basierten Sitzungen. Nicht authentifizierte Anfragen werden vom Server blockiert, wodurch unbefugte Änderungen verhindert werden.

2.1.3 Materialverwaltung

Administratoren können Lehrmaterialien hochladen, löschen und Kursen zuordnen. Jedes Material enthält einen Titel, den Speicherpfad sowie ein Erstellungsdatum. Beim Upload wird die Datei validiert und anschließend im Storage-Bereich abgelegt.

2.1.4 Kontakt- und Terminverwaltung

Im CMS werden eingegangene Kontaktanfragen und Terminbuchungen angezeigt. Die Datensätze enthalten Name, E-Mail, Nachricht sowie Datum und Uhrzeit des Termins. Einträge können nach Bedarf gelöscht werden. Alle Daten werden in der Datenbank gespeichert und mit Zeitstempeln versehen.

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

2.2 Website- Funktionalität

2.2.1 Kursanzeige

Die Kursübersicht im öffentlichen Bereich wird dynamisch aus der Datenbank generiert. Jeder Kurs enthält Titel, Beschreibung und ggf. verknüpfte Materialien. Die Daten werden über eine REST-API vom Server abgerufen.

2.2.2 Materialzugang über Code

Teilnehmer können Materialien über einen Zugangscode abrufen. Der Server prüft den eingegebenen Code anhand einer Systemvariablen. Bei korrekter Eingabe wird der Zugriff freigegeben, andernfalls erscheint eine Fehlermeldung im Frontend.

3. Schnittstellen

3.1 Datenbank

Die Anwendung verwendet eine relationale MariaDB-Datenbank mit den Tabellen: courses, materials, content_entries, users, contacts und appointments. Primär- und Fremdschlüssel stellen die Datenintegrität sicher. Beispielsweise verweist materials.course_id auf courses.id, wodurch eine eindeutige Kurszuordnung gewährleistet ist.

3.2 API-Endpunkte

Die Kommunikation zwischen Frontend und Backend erfolgt über eine REST-API.

Wichtige Endpunkte sind:

- GET /api/courses – Kurse abrufen
- GET /api/materials – Materialien abrufen
- PUT /api/cms/content/:name – Inhalte speichern
- DELETE /api/cms/materials/:id – Material löschen

Alle Endpunkte im CMS-Bereich sind durch Authentifizierung geschützt.

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

4. Betriebsbedingungen

Die Webapp läuft auf einem Linux-Server mit Node.js und MariaDB. Das Frontend wird Vite/Tailwind kompiliert und als statische Website bereitgestellt. Uploads und herunterladbare Materialien werden im Storage-Verzeichnis abgelegt. Regelmäßige Backups der Datenbank und Dateien sichern die Verfügbarkeit des Systems.

5. Qualitätsanforderungen

Die Anwendung soll stabil, wartbar und sicher sein. Tests erfolgen während der Entwicklung manuell im Browser sowie durch gezielte API-Aufrufe mit Curl. Hierdurch werden insbesondere die Zugriffsbeschränkung, Datenvalidierung und Reaktionszeit der API geprüft. Der Quellcode ist modular aufgebaut, um zukünftige Erweiterungen zu erleichtern.

6. Abnahmebedingungen

Das System gilt als abgenommen, wenn alle Hauptfunktionen des CMS (Login, Inhaltsbearbeitung, Materialverwaltung, Kontakt- und Terminübersicht) sowie der öffentliche Materialzugriff über Zugangscode fehlerfrei funktionieren. Darüber hinaus müssen keine kritischen Fehlermeldungen im Browser oder Server-Log auftreten.

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A6 Entity-Relationship-Model

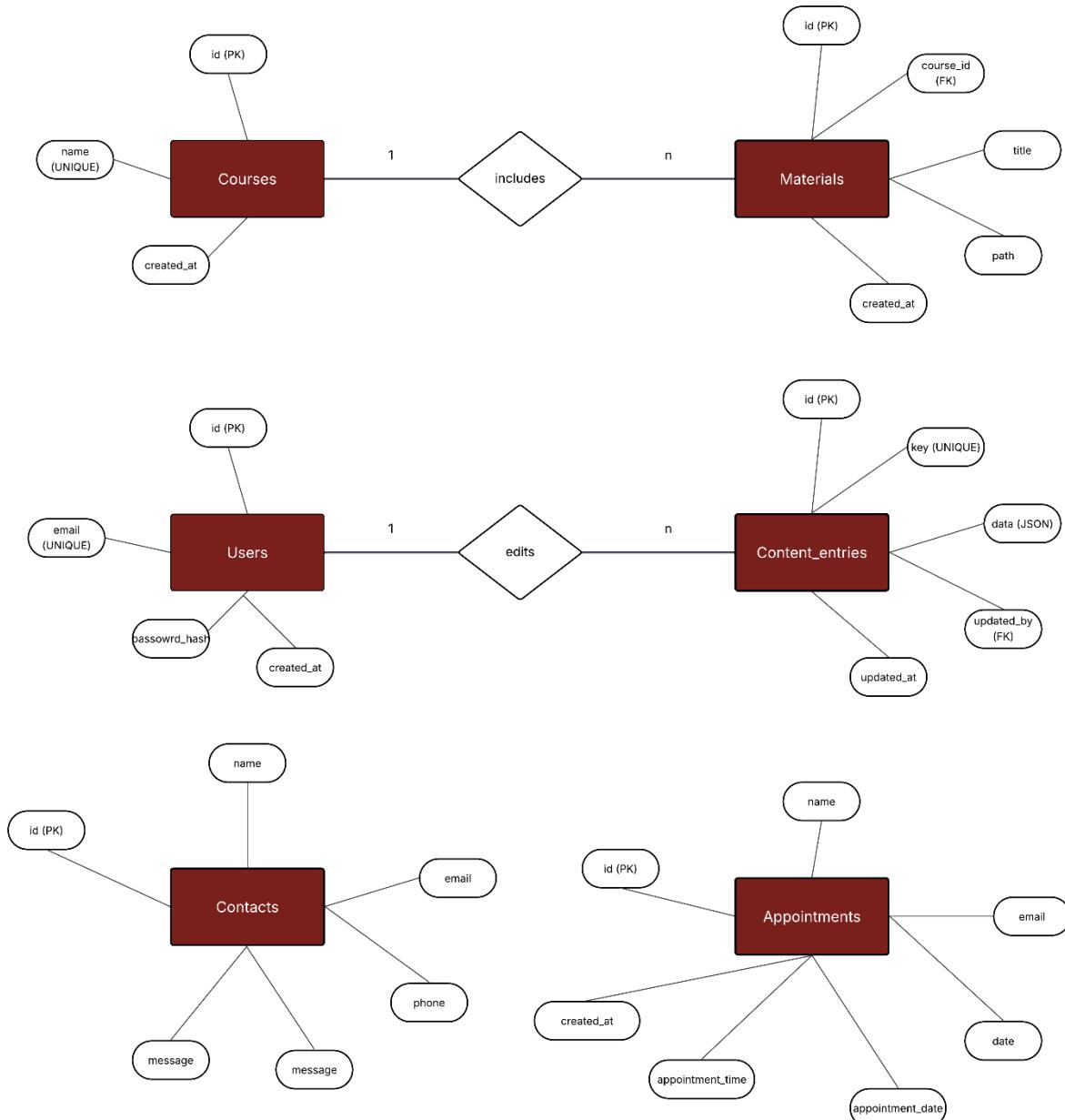


Abbildung 3: Entity-Relationship-Modell

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A7 Relationales Datenmodell

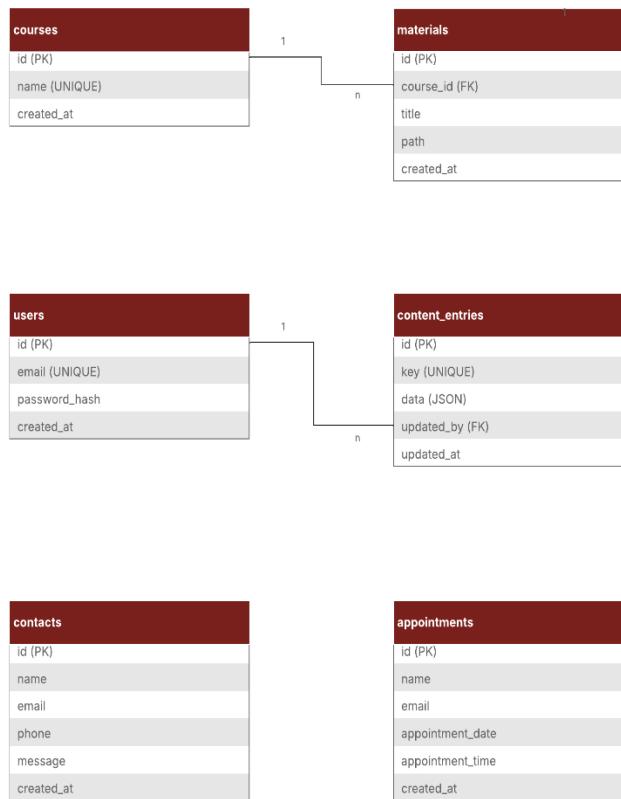


Abbildung 4: Relationales Datenmodell

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A8 Wireframes - Desktop und Mobile

DESKTOP

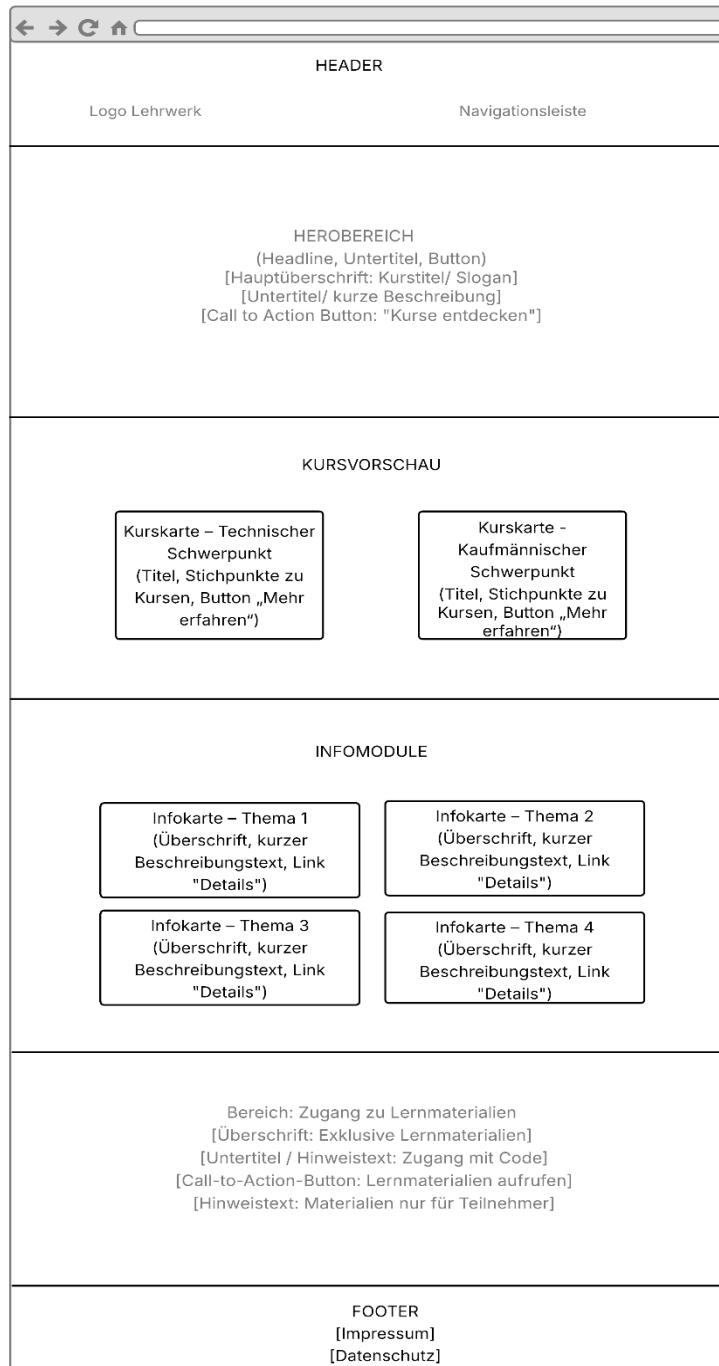


Abbildung 5: Wireframe Desktop

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

MOBILE

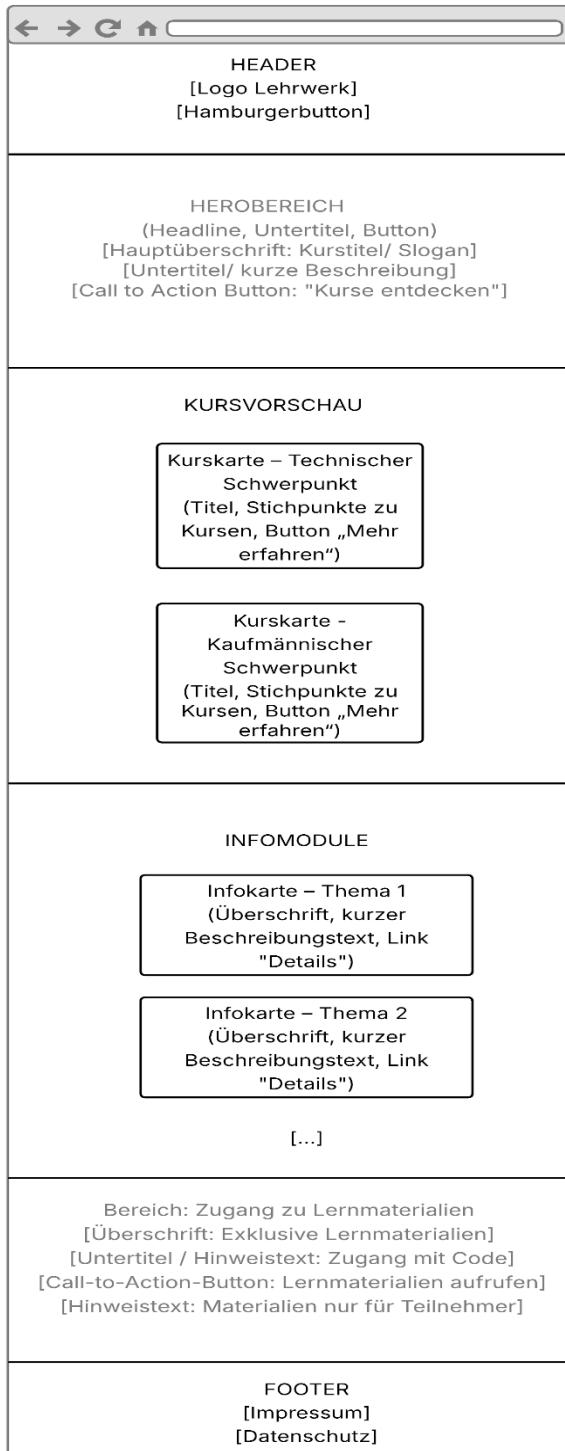


Abbildung 6: Wireframe Mobil

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A9 Screenshots Frontend

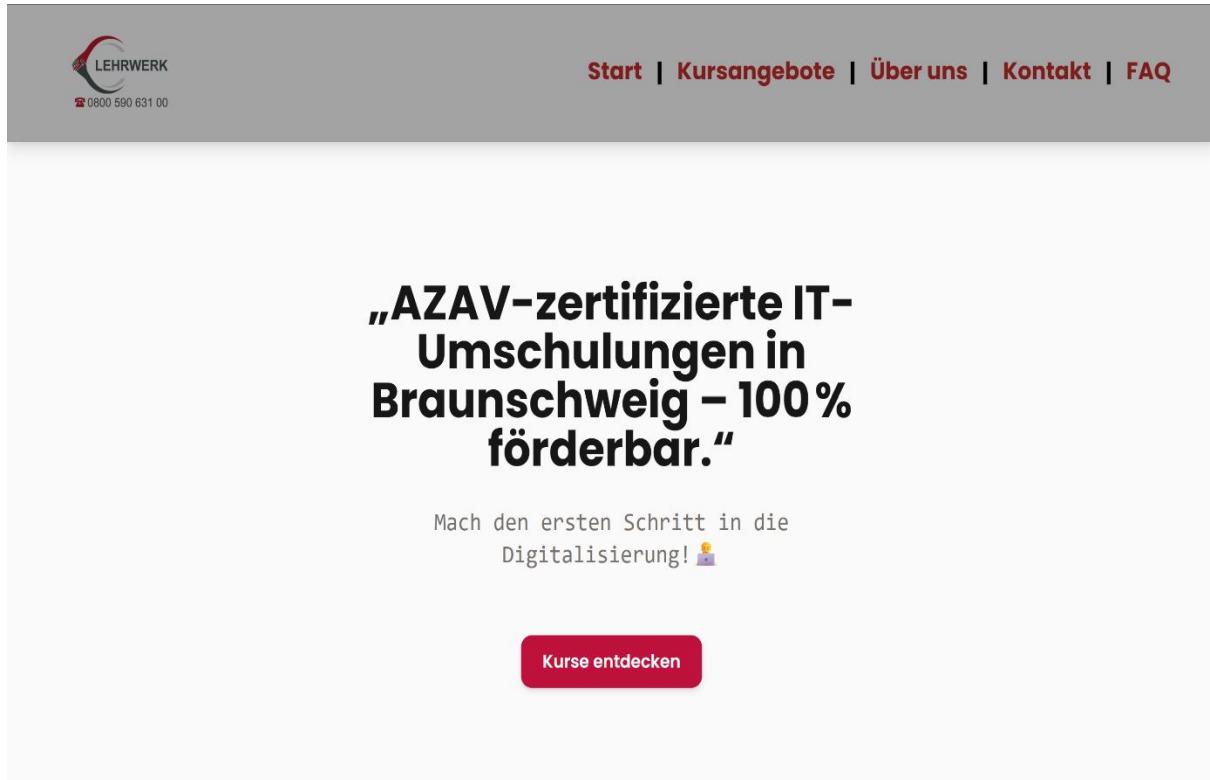


Abbildung 7: Heropage Desktop

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang



Abbildung 8: Heropage Mobil

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

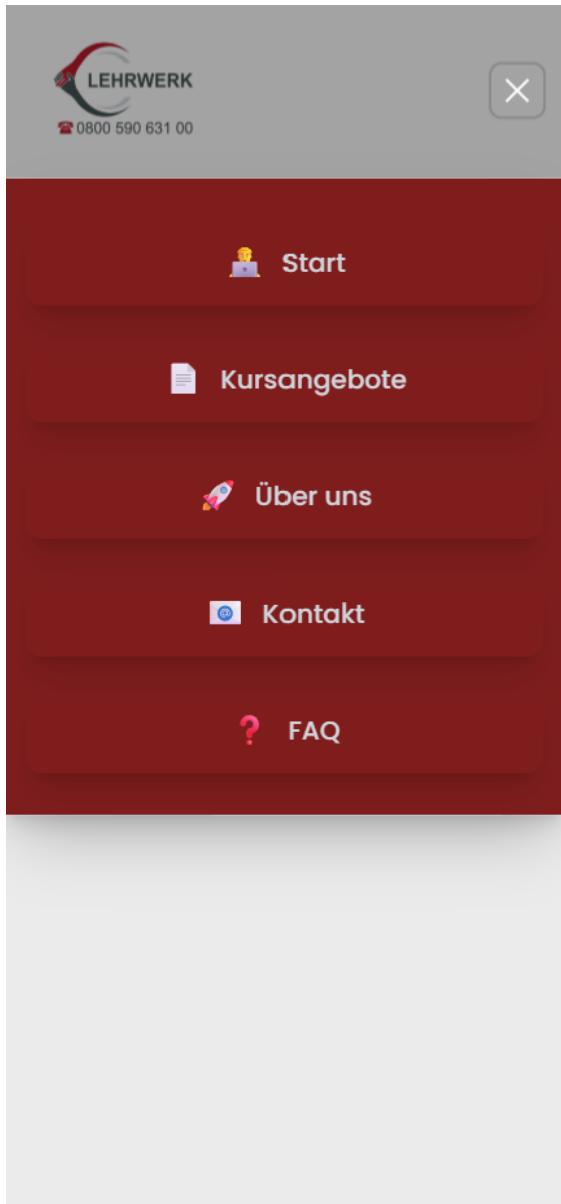
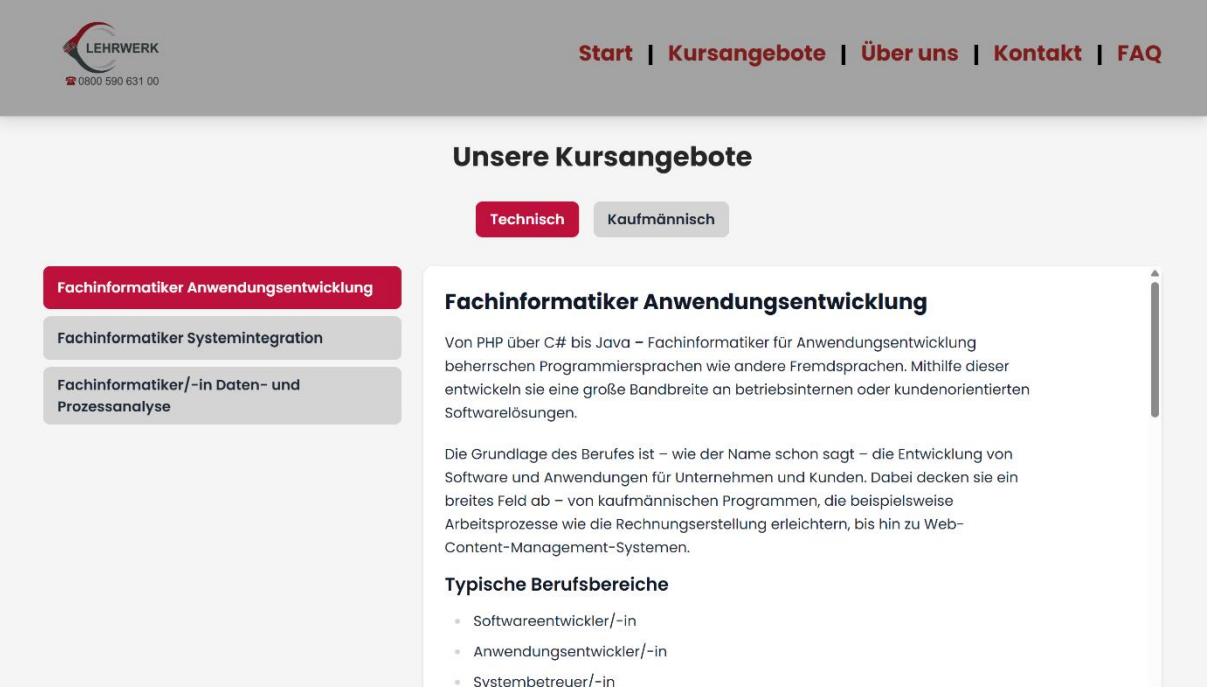


Abbildung 9: Heropage Responsive

LEHRWERK WEBAPP

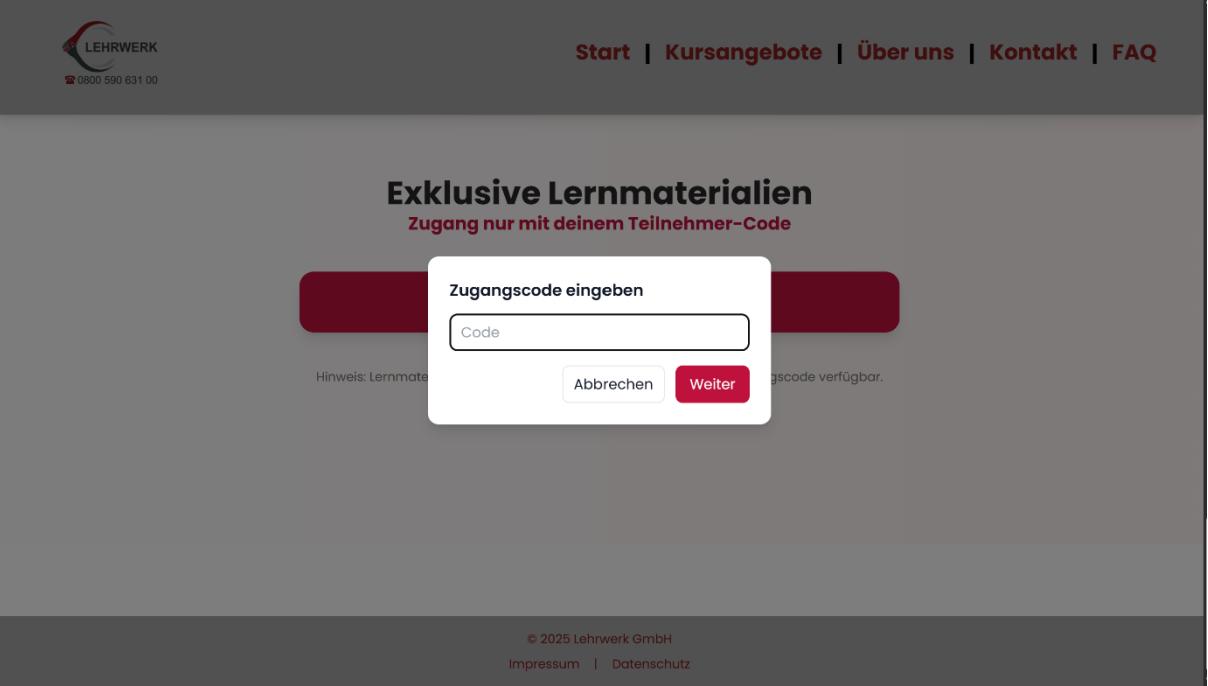
Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang



The screenshot shows the 'Unsere Kursangebote' (Our Course Offerings) section of the website. At the top, there is a navigation bar with links to 'Start', 'Kursangebote', 'Über uns', 'Kontakt', and 'FAQ'. Below the navigation, there are two buttons: 'Technisch' (Technical) and 'Kaufmännisch' (Commercial). A sidebar on the left lists course categories: 'Fachinformatiker Anwendungsentwicklung' (selected), 'Fachinformatiker Systemintegration', and 'Fachinformatiker/-in Daten- und Prozessanalyse'. The main content area is titled 'Fachinformatiker Anwendungsentwicklung' and describes the role, mentioning PHP, C#, Java, and various programming languages. It also lists typical professional areas such as Softwareentwickler/-in, Anwendungsentwickler/-in, and Systembetreuer/-in.

Abbildung 10: Kursangebote Übersicht Desktop



The screenshot shows the 'Exklusive Lernmaterialien' (Exclusive Learning Materials) page. At the top, there is a navigation bar with links to 'Start', 'Kursangebote', 'Über uns', 'Kontakt', and 'FAQ'. Below the navigation, the page title is 'Exklusive Lernmaterialien' with the subtext 'Zugang nur mit deinem Teilnehmer-Code'. A central input field is labeled 'Zugangscode eingeben' (Enter access code) with a placeholder 'Code'. Below the input field are two buttons: 'Abbrechen' (Cancel) and 'Weiter' (Continue). A small note at the bottom left says 'Hinweis: Lernmaterialien'. The bottom of the page includes a copyright notice '© 2025 Lehrwerk GmbH' and links to 'Impressum' and 'Datenschutz'.

Abbildung 11: Zugangscodeüberprüfung Desktop

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A10 Screenshots Contentmanagement System

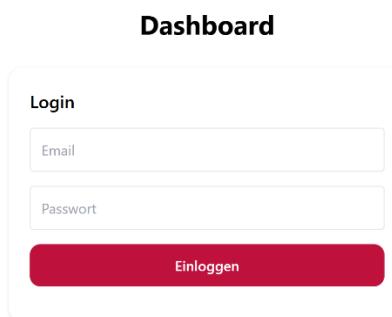


Abbildung 12: Loginmaske CMS

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

Dashboard

Kontaktanfragen					Suche: Name, Email, Telefon, Nachricht	Filtern
ID	Name / Email	Telefon	Nachricht	Erstellt	Aktion	
45	Max Mustermann max.mustermann@mustermail.de		Testnachricht	16.10.2025, 16:59:32	Löschen	

Eingegangene Termine		Suche: Name oder Email		tt.mm.jjjj	tt.mm.jjjj	Filtern
ID	Name / Email	Datum	Uhrzeit	Erstellt	Aktion	
14	Max Mustermann max.mustermann@mustermail.de	24.10.2025	12:00	16.10.2025, 17:05:12	Löschen	

Abbildung 13: Benachrichtigungen im CMS

Abbildung 14: Content bearbeiten im CMS

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

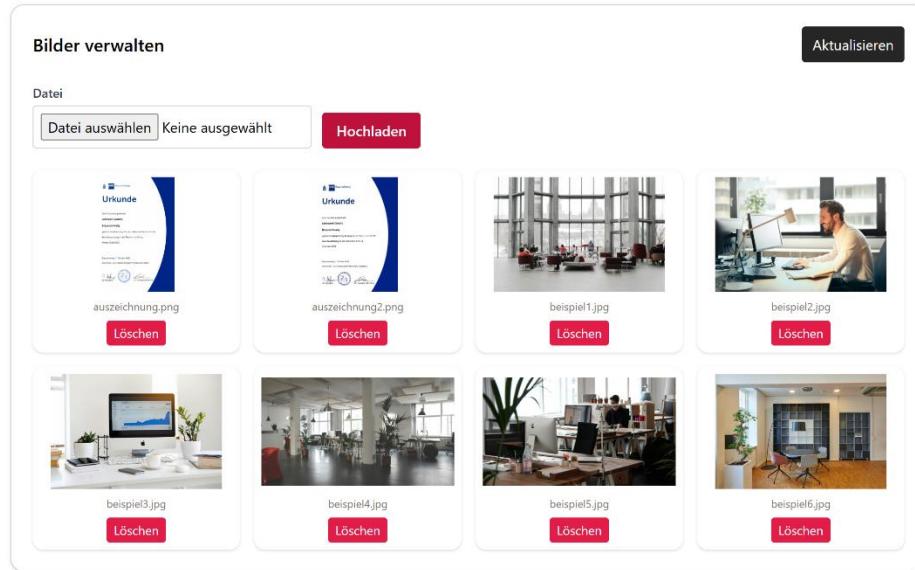


Abbildung 15: Bilder verwalten im CMS

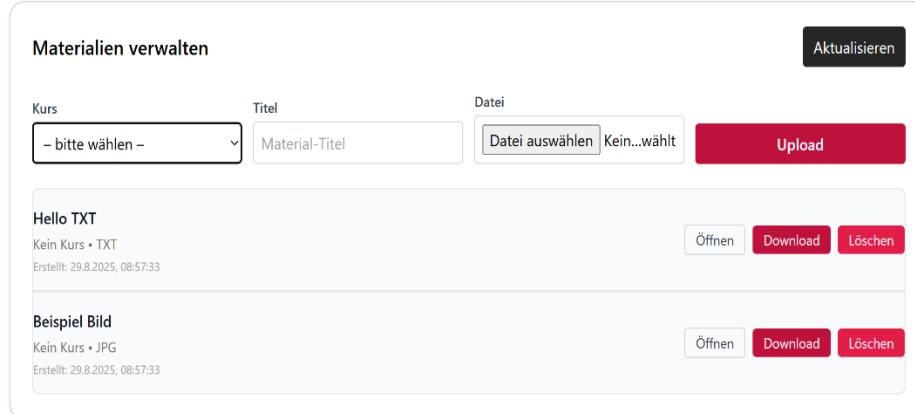


Abbildung 16: Materialien verwalten im CMS

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

Admin-Passwort ändern

Aktuelles Passwort

Neues Passwort (min. 8 Zeichen)

Neues Passwort

Neues Passwort wiederholen

Neues Passwort wiederholen

Passwort ändern

Logout

Abbildung 17: Möglichkeit zur Passwortänderung im CMS

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A11 EPK – Inhalte speichern im CMS

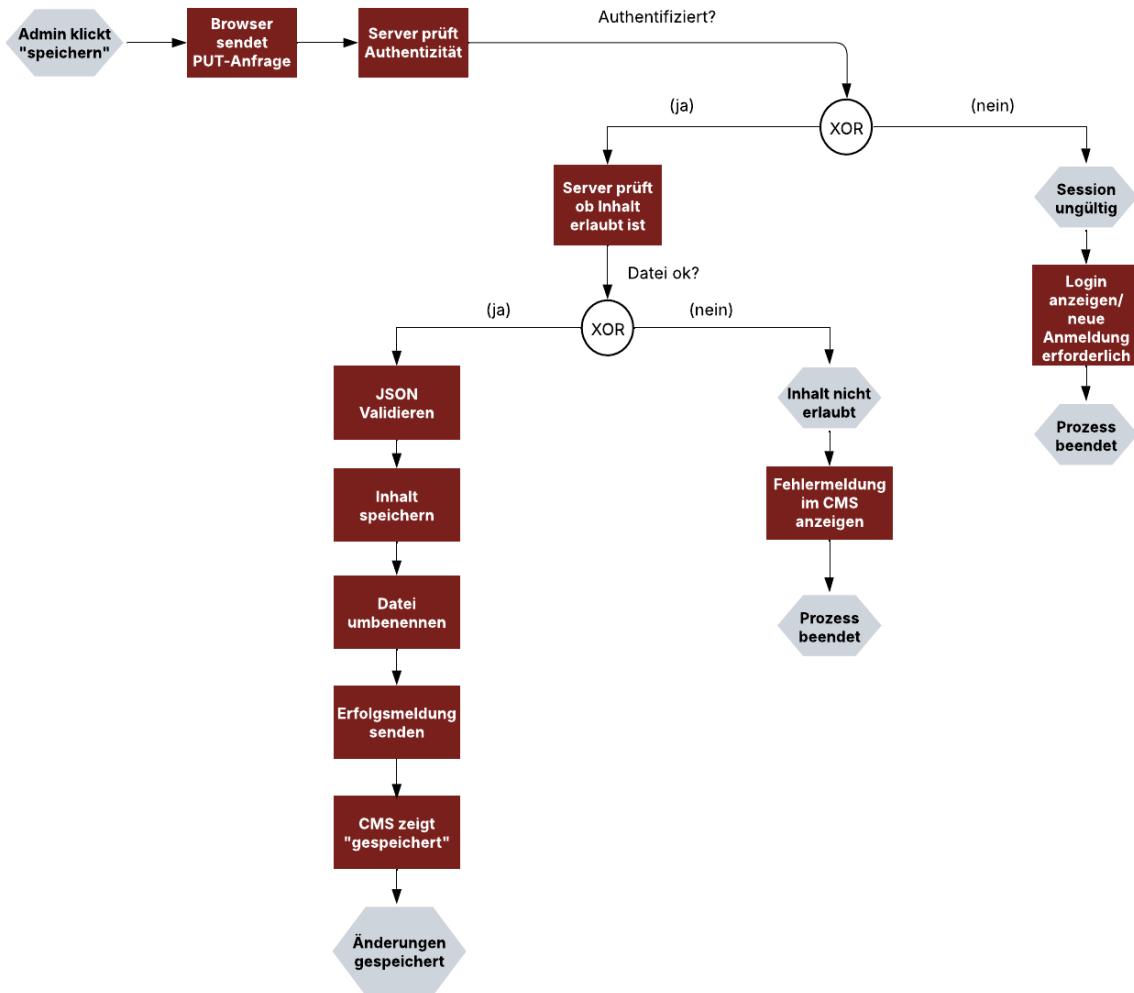


Abbildung 18: Ereignisprozesskette

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A12 SQL-Auszug Datenbankstruktur

Tabelle: courses

```
CREATE TABLE courses (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL UNIQUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Tabelle: materials

```
CREATE TABLE materials (
    id INT AUTO_INCREMENT PRIMARY KEY,
    course_id INT NULL,
    title VARCHAR(255) NOT NULL,
    path VARCHAR(500) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (course_id) REFERENCES courses(id)
        ON UPDATE CASCADE
        ON DELETE SET NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Tabelle: users

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(255) NOT NULL UNIQUE,
    password_hash VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

Tabelle: content_entries

```
CREATE TABLE content_entries (
    id INT AUTO_INCREMENT PRIMARY KEY,
    `key` VARCHAR(255) NOT NULL UNIQUE,
    data JSON NOT NULL,
    updated_by INT NULL,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (updated_by) REFERENCES users(id)
        ON UPDATE CASCADE
        ON DELETE SET NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Tabelle: contacts

```
CREATE TABLE contacts (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    phone VARCHAR(50),
    message TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Tabelle: appointments

```
CREATE TABLE appointments (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    appointment_date DATE NOT NULL,
    appointment_time TIME NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

Anhang

A13 Verzeichnisstruktur – Root/Frontend

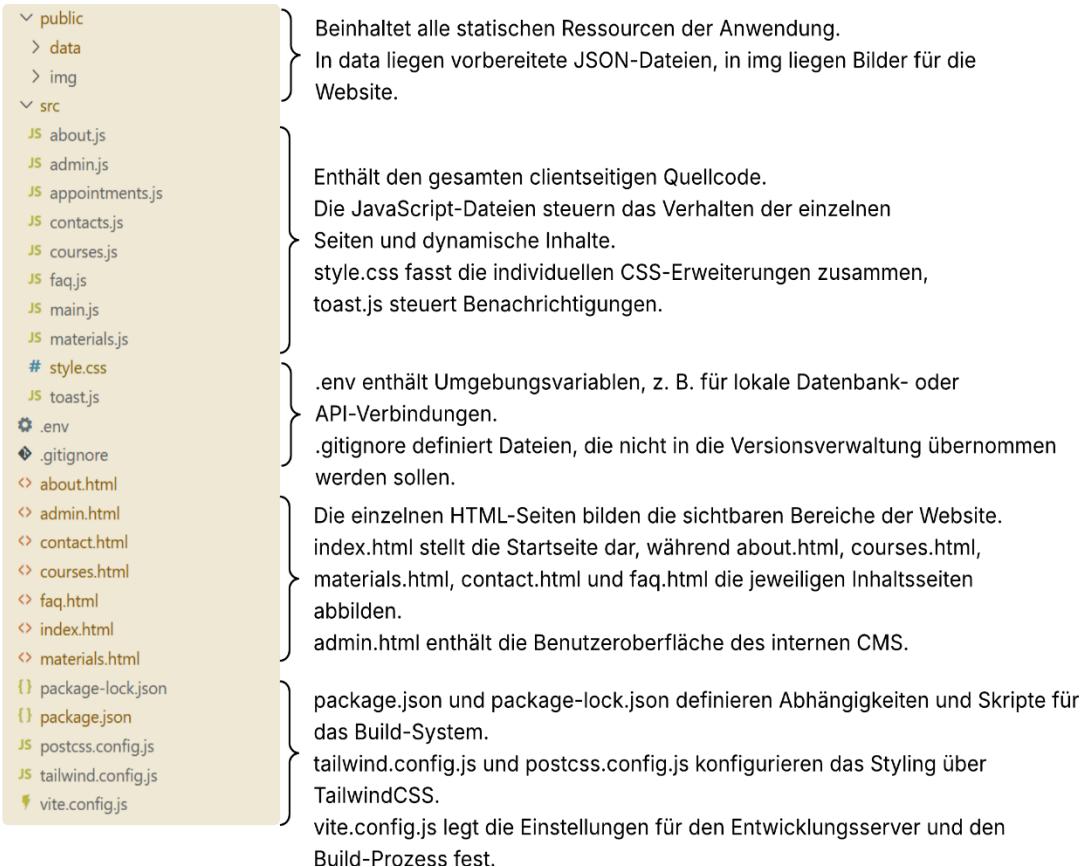


Abbildung 19: Projektstruktur Root

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A14 Verzeichnisstruktur – Backend

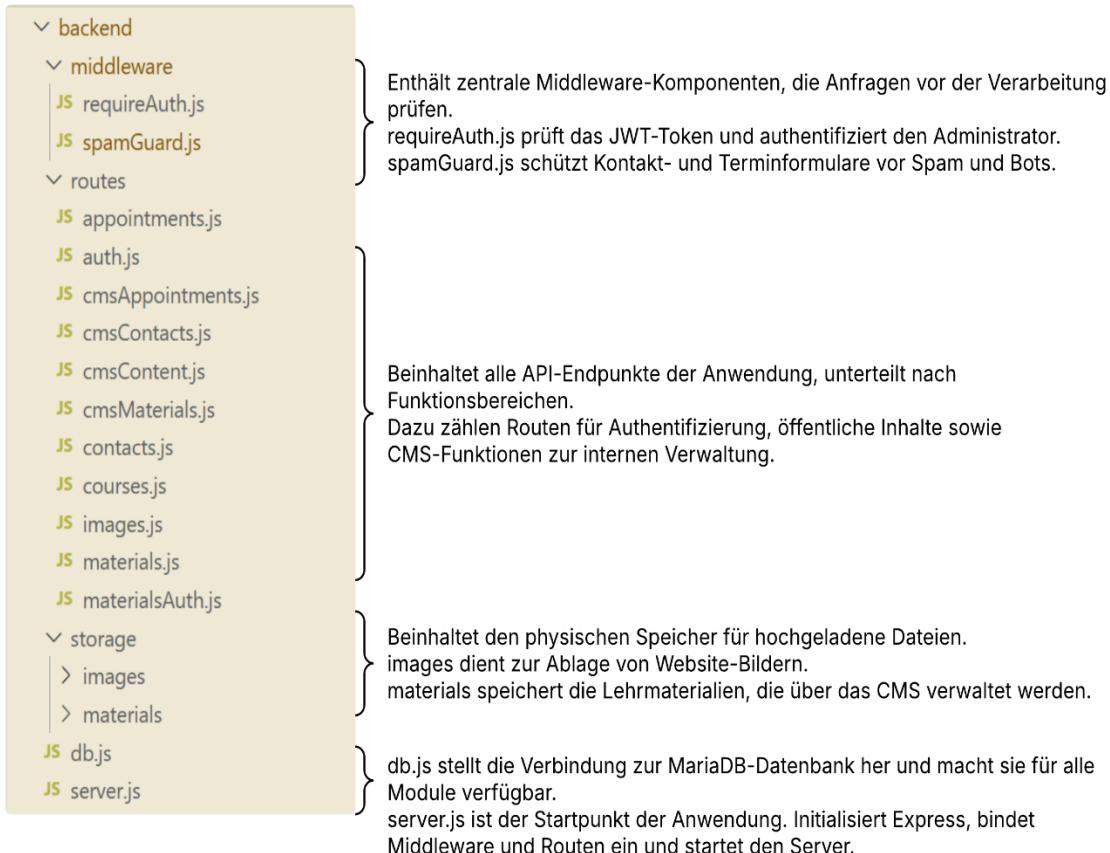


Abbildung 20: Projektstruktur Backend

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A15 Codebeispiel: JWT-Authentifizierung (requireAuth.js)

```
backend > middleware > JS requireAuth.js > ...
1  import jwt from 'jsonwebtoken';
2
3  export function requireAuth(req, res, next) {
4    const token = req.cookies?.auth;
5    if (!token) return res.status(401).json({ ok:false, message:'Not authenticated' });
6    try {
7      const payload = jwt.verify(token, process.env.JWT_SECRET);
8      req.user = payload; // { uid }
9      next();
10     } catch {
11       return res.status(401).json({ ok:false, message:'Invalid or expired session' });
12     }
13   }
14  export default requireAuth;
```

Abbildung 21: Codebeispiel Authentifizierung

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A16 Codebeispiel: Spam-Schutz (spamGuard.js)

```

25  // 3) Form-Age (Verhindert zu schnelle Formularübermittlung)
26  const now = Date.now();
27  const clientTS = Number(b.ts || 0);
28  if (!clientTS || isNaN(clientTS) || now - clientTS < 3000 || now - clientTS > 20 * 60 * 1000) {
29  | return res.status(400).json({ error: "Ungültige Formularzeit." });
30  }
31
32
33  // 4) Plausibilitätsprüfung der Eingaben
34  const fieldStr = (x) => (typeof x === "string" ? x : "");
35  const name = fieldStr(b.name);
36  const email = fieldStr(b.email);
37  const phone = fieldStr(b.phone);
38  const message = fieldStr(b.message);
39  const date = fieldStr(b.date);
40  const time = fieldStr(b.time);
41
42  // Limits
43  if (name.length > 100 || email.length > 200 || phone.length > 40 || message.length > 2000) {
44  | return res.status(400).json({ error: "Feld zu lang." });
45  }
46
47  // E-Mail grob prüfen
48  if (email && !validator.isEmail(email)) {
49  | return res.status(400).json({ error: "E-Mail ungültig." });
50  }
51
52  // Max. 2 Links im Nachrichtentext zulässig
53  const urlLike = (message.match(/https?:\/\/|www\.\./gi) || []).length;
54  if (urlLike > 2) {
55  | return res.status(400).json({ error: "Zu viele Links." });
56  }
57
58  next();
59  };
60 }

```

Abbildung 22: Codebeispiel Spamschutz

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A17 Sicherheitstest / Rate-Limit-Test

```
timmy@v45596:~/webdev/lehrwerk_app$ #!/bin/bash
URL="http://178.254.25.12:3000/api/contact"

clientTS=$(( $(date +%s%N | cut -b1-13) - 5000 ))

for i in $(seq 1 12); do
  echo "Request #$i"
  curl -s -X POST "$URL" \
    -H "Content-Type: application/json" \
    -d "{\"name\":\"Test$i\", \"email\":\"test@example.com\", \"phone\":\"0123456789\", \"message\":\"Test Nachricht\", \"hp\":\"\", \"ts\":$clientTS}" \
    | jq . 2>/dev/null || curl -s -X POST "$URL" \
    -H "Content-Type: application/json" \
    -d "{\"name\":\"Test$i\", \"email\":\"test@example.com\", \"phone\":\"0123456789\", \"message\":\"Test Nachricht\", \"hp\":\"\", \"ts\":$clientTS}"
  echo ""
  sleep 0.5
done
Request #1
{
  "success": true
}

Request #2
{
  "success": true
}

Request #3
{
  "success": true
}

Request #4
{
  "success": true
}

Request #10
{
  "success": true
}

Request #11
Too many requests, please try again later.
Request #12
Too many requests, please try again later.
timmy@v45596:~/webdev/lehrwerk_app$
```

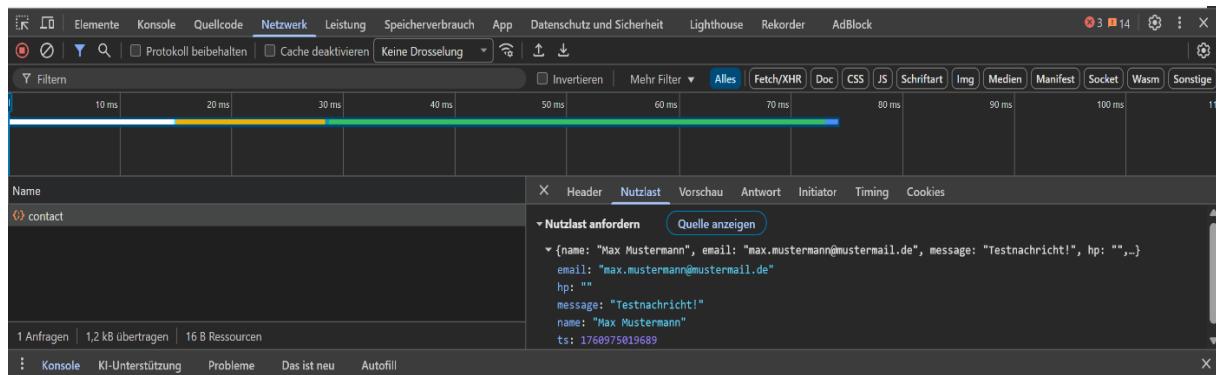
Abbildung 23: Sicherheitstest Rate-Limit

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A18 Integrationstest (Kontaktformular)

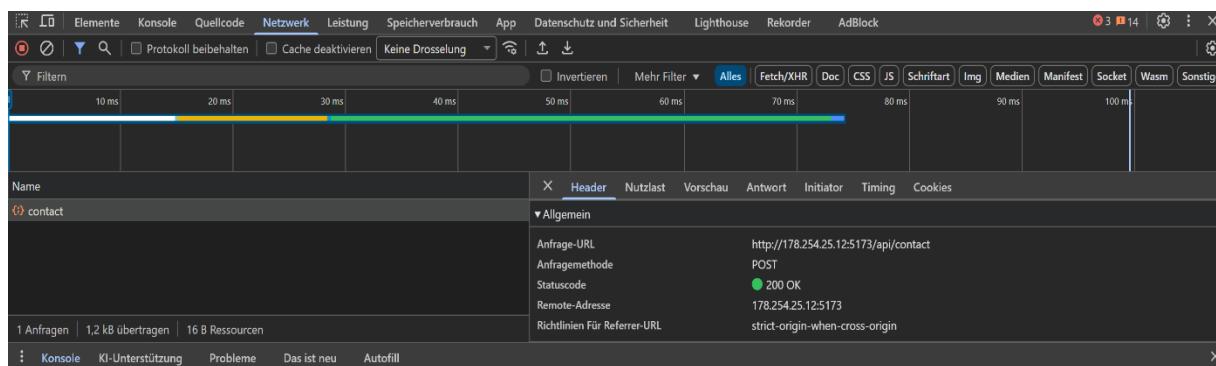


The screenshot shows the Network tab of the browser developer tools. A single request is listed under the 'contact' name. The 'Nutzlast' (Payload) tab is selected, showing the JSON data sent to the server:

```

{
  "name": "Max Mustermann",
  "email": "max.mustermann@mustermail.de",
  "message": "Testnachricht!",
  "hp": "",
  "message": "Testnachricht!",
  "name": "Max Mustermann",
  "ts": 1760975019689
}
  
```

Abbildung 24: Eingabe und Absenden der Formulardaten



The screenshot shows the Network tab of the browser developer tools. The same POST request to 'contact' is shown, but this time the 'Allgemein' (General) tab is selected. The response details are as follows:

- Anfrage-URL: <http://178.254.25.12:5173/api/contact>
- Anfragemethode: POST
- Statuscode: 200 OK
- Remote-Adresse: 178.254.25.12:5173
- Richtlinien Für Referrer-URL: strict-origin-when-cross-origin

Abbildung 25: Protokoll des erfolgreichen Datentransfers

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

A19 Benutzerdokumentation (Auszüge) – Lehrwerk Webapp CMS

1. Einleitung

Die Lehrwerk Webapp verfügt über einen internen Administrationsbereich, der es den Mitarbeitern der Lehrwerk GmbH ermöglicht, die Inhalte der Website selbstständig zu pflegen und zu verwalten. Über diesen Bereich können Kursdaten, Lernmaterialien, Bilder, Textinhalte sowie eingegangene Kontaktanfragen und Terminbuchungen bearbeitet werden.

Der Zugriff auf das Content-Management-System (CMS) ist passwortgeschützt und ausschließlich für autorisierte Benutzer vorgesehen. Nach erfolgreicher Anmeldung werden die administrativen Funktionen eingeblendet. Die Bedienung erfolgt über eine einheitliche, webbasierte Oberfläche.

3. Aufbau der Oberfläche

Nach erfolgreicher Anmeldung erscheinen alle Verwaltungsbereiche untereinander auf einer Seite. Über die vertikale Scrollnavigation können die einzelnen Module erreicht werden.

Folgende Hauptbereiche stehen zur Verfügung:

1. Kontaktanfragen
2. Eingegangene Termine
3. Content bearbeiten
4. Bilder verwalten
5. Materialien verwalten
6. Admin-Passwort ändern

Jeder Abschnitt besitzt eine eigene Funktion zur Anzeige, Filterung, Bearbeitung oder Löschung von Daten.

4. Funktionsbeschreibung der Module

4.3 Content bearbeiten

Das Modul Content bearbeiten dient der Pflege der inhaltlichen Bereiche der Website. Hierzu stehen vorbereitete JSON-Dateien zur Verfügung, die Texte, Überschriften, Teaser und Bildpfade enthalten.

LEHRWERK WEBAPP

Konzeption und Realisierung einer Full-Stack-Webanwendung zur digitalen Präsentation und Steuerung interner Abläufe

Anhang

Vorgehensweise:

1. Im Dropdown-Menü die gewünschte Datei auswählen (z. B. modals.json oder ueberuns.json).
2. Mit „Laden“ wird der aktuelle Inhalt angezeigt.
3. Änderungen können direkt im Textfeld vorgenommen werden.
4. Mit „Speichern“ werden die Anpassungen dauerhaft übernommen.

Die JSON-Struktur ist klar aufgebaut: Jeder Abschnitt enthält Schlüssel-Wert-Paare für Titel, Text, Teaser oder Icon.

Beispiel:

```
"pizzatag": {  
    "title": "Pizzatag",  
    "teaser": "Jeden letzten Freitag im Monat - für Austausch & gute  
    Laune!",  
    "content": "<p>So stärken wir das Miteinander ...</p>"  
}
```

Hinweis: In den JSON-Dateien können auch Bildpfade gepflegt werden, die auf im CMS hochgeladene Dateien verweisen.

Beispiel:

```
"src": "backend/storage/images/beispiel1.jpg"
```

7. Zusammenfassung

Das Lehrwerk CMS bietet den Administratoren der Lehrwerk GmbH eine zentrale, benutzerfreundliche Oberfläche zur Verwaltung sämtlicher Website-Inhalte. Durch die klare Struktur und die einfache Bedienung kann die Website ohne technische Kenntnisse gepflegt werden. Die Kombination aus Login-Schutz, JSON-basierten Inhalten, Upload-Funktion und Echtzeit-Speicherung sorgt für eine effiziente und sichere Verwaltung.