Zeitplanung BKBdemy

Projektplanung (ca. 2 Wochen):

17.04.2023: Wir haben ein Kick-off-Meeting abgehalten, um das Projekt zu starten und die Ziele und Anforderungen festzulegen.

18.04.2023: An diesem Tag haben wir uns intensiv mit der Anforderungsanalyse beschäftigt. Wir haben Interviews mit potenziellen Nutzern durchgeführt, um ihre Bedürfnisse und Erwartungen zu verstehen. Basierend auf diesen Informationen haben wir ein detailliertes Pflichtenheft erstellt, das als Leitfaden für die weitere Entwicklung dient.

24.04.2023: An diesem Tag haben wir uns zusammengesetzt, um eine Risikoanalyse durchzuführen. Wir haben mögliche Risiken identifiziert, bewertet und entsprechende Maßnahmen entwickelt, um diese zu minimieren oder zu vermeiden.

25.04.2023: Wir haben das Pflichtenheft finalisiert und es zur Genehmigung eingereicht.

Umsetzung (ca. 6 Wochen):

26.04.2023 - 27.04.2023: In den nächsten zwei Tagen haben wir uns auf das Design der Website konzentriert. Wir haben Wireframes erstellt, um die Struktur und das Layout zu definieren. Anschließend haben wir das visuelle Design entwickelt und die Farbpalette, Typografie und Bildauswahl festgelegt.

02.05.2023 - 04.05.2023: Wir haben das Backend entwickelt, indem wir die erforderlichen Datenbanken und APIs aufgebaut haben. Dabei haben wir auf bewährte Praktiken und Sicherheitsstandards geachtet, um eine stabile und zuverlässige Plattform zu gewährleisten.

09.05.2023 - 10.05.2023: In dieser Phase haben wir uns auf die Entwicklung des Frontends konzentriert. Wir haben die Benutzeroberfläche erstellt, interaktive Funktionen implementiert und das responsive Design für verschiedene Geräte und Bildschirmgrößen optimiert.

16.05.2023 - 17.05.2023: Wir haben umfangreiche Tests durchgeführt, um sicherzustellen, dass alle Funktionen einwandfrei funktionieren und die Plattform benutzerfreundlich ist. Wir haben Fehler und Probleme identifiziert und diese behoben, um eine hohe Qualität und Nutzerzufriedenheit sicherzustellen.

Zusätzliche Sonntage: 23.04.2023, 30.04.2023, 07.05.2023, 14.05.2023, 21.05.2023, 28.05.2023 (jeweils 3-5 Stunden): An diesen Sonntagen haben wir uns zusätzlich Zeit genommen, um bestimmte Funktionen zu verfeinern, Verbesserungen basierend auf Feedback umzusetzen und zusätzliche Tests durchzuführen. Dies ermöglichte es uns, das Projekt auf einem hohen Niveau abzuschließen und sicherzustellen, dass alle Anforderungen erfüllt sind.

Fertigstellung und Präsentation (ca. 1 Woche):

31.05.2023: Wir haben die finale Phase des Projekts eingeleitet. An diesem Tag haben wir letzte Anpassungen an der Plattform vorgenommen, um sicherzustellen, dass alles reibungslos funktioniert und die Benutzererfahrung optimal ist.

31.05.2023: Wir haben unsere PowerPoint-Präsentation erstellt und sorgfältig vorbereitet, um das Projekt in einer ansprechenden und überzeugenden Weise zu präsentieren. Dabei haben wir die Inhalte strukturiert, visuelle Elemente eingefügt und den Fluss der Präsentation optimiert.

01.01.2023: Wir haben die Abschlussarbeiten und Dokumentation vollständig abgeschlossen. Wir haben eine gründliche Überprüfung durchgeführt, um sicherzustellen, dass alle Aspekte des Projekts dokumentiert sind und den geforderten Standards entsprechen.

04.06.2023: An diesem Tag haben wir unsere Abschlusspräsentation geübt, um sicherzustellen, dass wir alle relevanten Informationen präzise und überzeugend präsentieren können. Wir haben Feedback von unserem Team und anderen Personen eingeholt, um die Präsentation weiter zu verbessern.

Wasserfallmodell innerhalb des Projektes „BKBdemy“

Einleitung

Diese Dokumentation dient dazu, das angewendete Vorgehensmodell für das Projekt zu beschreiben und seinen Aufbau zu erläutern. Das gewählte Vorgehensmodell spielt eine entscheidende Rolle bei der Planung, Organisation und Durchführung des Projekts. In diesem Dokument werden wir den Aufbau des Vorgehensmodells im Kontext unseres Projekts genauer erläutern und die Gründe für seine Auswahl erläutern.

Vorgehensmodell: Wasserfallmodell

Das Wasserfallmodell wurde als das Vorgehensmodell für unser Projekt gewählt. Dieses Modell zeichnet sich durch eine sequenzielle und phasenbasierte Struktur aus, bei der jede Phase auf der vorherigen aufbaut und klare Abgrenzungen zwischen den Phasen aufweist. Im Folgenden werden die Phasen des Wasserfallmodells detailliert beschrieben und ihr Bezug zu unserem Projekt hergestellt.

Anforderungsanalyse

In dieser Phase haben wir die Anforderungen für das Projekt ermittelt und dokumentiert. Wir haben mit den Stakeholdern zusammengearbeitet, um ihre Bedürfnisse und Erwartungen zu verstehen und diese Anforderungen klar zu definieren.

Systementwurf

In dieser Phase haben wir den Systementwurf für die BKBdemy Website erstellt. Wir haben die Architektur, die Datenbankstruktur und die Benutzeroberfläche entworfen, um die Anforderungen bestmöglich zu erfüllen.

Implementierung

In dieser Phase haben wir den Code für die Website entwickelt und die Funktionalitäten entsprechend den festgelegten Anforderungen umgesetzt. Dabei haben wir uns an bewährte Entwicklungspraktiken und Standards gehalten, um eine hohe Codequalität zu gewährleisten.

Test und Qualitätssicherung

In dieser Phase haben wir umfassende Tests durchgeführt, um sicherzustellen, dass die Website ordnungsgemäß funktioniert und den Anforderungen entspricht. Dabei wurden Unit-Tests, Integrationstests und Systemtests durchgeführt, um mögliche Fehler oder Mängel frühzeitig zu identifizieren und zu beheben.

Bereitstellung und Wartung

In dieser letzten Phase wurde die Website bereitgestellt und für die Nutzung freigegeben. Wir haben sicherzustellen, dass die Website reibungslos funktioniert und eventuelle Nachbesserungen oder Wartungsarbeiten durchgeführt.

Auswahl des Wasserfallmodells

Die Entscheidung, das Wasserfallmodell für unser Projekt zu wählen, basierte auf verschiedenen Faktoren und Vorteilen, die es bietet:

Klare Struktur: Das Wasserfallmodell bietet eine klare und strukturierte Vorgehensweise, bei der jede Phase klar definierte Ziele und Ergebnisse hat. Dadurch wird eine geordnete und planmäßige Entwicklung ermöglicht.

Frühzeitige Anforderungsspezifikation: Durch die Anforderungsanalyse am Anfang des Projekts konnten wir die Anforderungen frühzeitig verstehen und dokumentieren. Dies half, Missverständnisse und spätere Änderungen zu minimieren.

Risikominimierung: Das Wasserfallmodell ermöglicht eine frühe Identifizierung und Behandlung von Risiken. Durch die sequenzielle Natur des Modells können potenzielle Risiken frühzeitig erkannt und entsprechende Maßnahmen ergriffen werden.

Dokumentation: Das Wasserfallmodell legt Wert auf eine umfassende Dokumentation während jeder Phase des Projekts. Dadurch werden Informationen festgehalten und die Nachvollziehbarkeit des Projekts verbessert.

Geringe Abhängigkeit von Ressourcen: Das Wasserfallmodell erfordert weniger Ressourcen, da jede Phase nach Abschluss der vorherigen Phase gestartet werden kann. Dadurch wird eine effiziente Nutzung der verfügbaren Ressourcen ermöglicht.

Obwohl das Wasserfallmodell viele Vorteile bietet, ist es wichtig zu beachten, dass es auch einige Herausforderungen mit sich bringen kann. Eine der Hauptkritikpunkte ist die begrenzte Flexibilität bei Änderungen während des Entwicklungsprozesses. Jedoch kann diese Herausforderung durch eine gründliche Anforderungsanalyse und eine klare Kommunikation mit den Stakeholdern gemildert werden.

Fazit

Das Wasserfallmodell wurde als Vorgehensmodell für unser Projekt gewählt, da es eine klare Struktur, eine frühe Anforderungsspezifikation, Risikominimierung und eine umfassende Dokumentation ermöglicht. Durch die Anwendung dieses Modells konnten wir das Projekt erfolgreich planen, entwickeln und umsetzen. Es hat uns geholfen, den Überblick über den Fortschritt zu behalten und potenzielle Risiken rechtzeitig zu identifizieren. Insgesamt war das Wasserfallmodell eine geeignete Wahl für unser Projekt und hat zu dessen erfolgreicher Umsetzung beigetragen.

BKBdemy /htdocs Dokumentation

.htaccess-Datei

Die .htaccess-Datei ist eine Konfigurationsdatei für Webserver, die es ermöglicht, verschiedene Einstellungen für die Website auf Verzeichnis- oder Dateiebene zu definieren. In der folgenden Dokumentation wird die .htaccess-Datei für die BKBdemy-Website beschrieben.

BEGIN bkbdemy

<IfModule mod_rewrite.c> RewriteEngine On

Exclude certain file types from rewrite rules

RewriteRule .(css|js|png|gif|jpg|jpeg|woff|ttf|svg|ico|pdf|doc|docx|xls|xlsx|ppt|pptx|zip|rar|mp4|avi|mov|wmv|flv|mp3|wav|ogg|webm|json)$ - [L]

Redirect HTTP requests to HTTPS

RewriteCond %{HTTPS} !=on RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]

Handle all requests with index.php

RewriteCond %{REQUEST_FILENAME} !-f RewriteCond %{REQUEST_FILENAME} !-d RewriteRule ^(.*)$ index.php [L] </IfModule>

END bkbdemy

Diese .htaccess-Datei enthält einige Einstellungen und Regeln, die für die BKBdemy-Website relevant sind:

RewriteEngine On: Diese Einstellung aktiviert die Apache Rewrite Engine, die es ermöglicht, URLs innerhalb der Website umzuschreiben und weiterzuleiten.

RewriteRule \.(css|js|png|gif|jpg|jpeg|woff|ttf|svg|ico|pdf|doc|docx|xls|xlsx|ppt|pptx|zip|rar|mp4|avi|mov|wmv|flv|mp3|wav|ogg|webm|json)$ - [L]: Diese Regel schließt bestimmte Dateitypen von den Rewrite-Regeln aus. Das

bedeutet, dass URLs mit diesen Dateitypen nicht umgeschrieben werden.

RewriteCond %{HTTPS} !=on: Diese Bedingung prüft, ob die Anfrage nicht bereits über HTTPS erfolgt, und leitet sie gegebenenfalls an HTTPS weiter.

RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]: Diese Rewrite-Regel leitet die Anfrage an die HTTPS-Version der Website weiter und gibt einen 301-Statuscode zurück, um den Suchmaschinen mitzuteilen, dass sich die URL dauerhaft geändert hat.

RewriteCond %{REQUEST_FILENAME} !-f: Diese Bedingung prüft, ob die angeforderte Datei nicht existiert.

RewriteCond %{REQUEST_FILENAME} !-d: Diese Bedingung prüft, ob das angeforderte Verzeichnis nicht existiert.

RewriteRule ^(.*)$ index.php [L]: Diese Rewrite-Regel leitet alle Anfragen an die index.php-Datei weiter, die für die Verarbeitung der Website-URLs und -Anfragen zuständig ist.

Es ist wichtig, dass die .htaccess-Datei korrekt konfiguriert ist, um sicherzustellen, dass die Website korrekt funktioniert. Änderungen an der .htaccess-Datei sollten immer vorsichtig und mit Bedacht vorgenommen werden, um sicherzustellen, dass keine unerwarteten Auswirkungen auftreten.

Routing Dokumentation

Die Route-Klasse stellt ein einfaches Routing-System dar, das es ermöglicht, Anfragen an bestimmte URLs an entsprechende Funktionen weiterzuleiten.

Die statische Variable $routes ist ein Array, das alle definierten Routen enthält. Jede Route besteht aus einem regulären Ausdruck ($expression), der die URL-Muster beschreibt, einer Callback-Funktion ($function), die ausgeführt wird, wenn die URL-Muster übereinstimmt, und einer HTTP-Methoden-Definition

($method), die die HTTP-Methoden festlegt, auf die die Route reagieren soll.

Die Methoden add, pathNotFound und methodNotAllowed fügen neue Routen hinzu und definieren Funktionen, die aufgerufen werden sollen, wenn die angeforderte URL nicht gefunden wird oder die angeforderte HTTP-Methode nicht unterstützt wird.

Die run-Methode ist die Hauptmethode der Klasse, die aufgerufen wird, wenn eine neue Anfrage an den Server gesendet wird. Sie iteriert über alle definierten Routen, sucht nach einer Übereinstimmung des URL-Musters und der HTTP-Methoden-Definition und führt die zugehörige Callback-Funktion aus, wenn eine Übereinstimmung gefunden wurde. Wenn keine Übereinstimmung gefunden wurde, wird entweder eine 404-Fehlermeldung ausgegeben oder die Funktion aufgerufen, die für nicht gefundene Pfade definiert wurde, oder eine 405-Fehlermeldung ausgegeben oder die Funktion aufgerufen, die für nicht unterstützte HTTP-Methoden definiert wurde.

Das Routing-System bietet eine einfache Möglichkeit, den Code einer Webseite zu organisieren und die Verarbeitung von Anfragen auf verschiedene Funktionen zu verteilen.

Webpack

Dies ist eine Konfigurationsdatei für das Build-Tool Webpack, die es Entwicklern ermöglicht, eine JavaScript-App zu erstellen und zu optimieren.

Zunächst wird der Einstiegspunkt (entry) für die App festgelegt. In diesem Fall ist es ein JavaScript-Modul (index.js) im Ordner /src/js. Der Ausgabepunkt (output) für die optimierte App ist ein JavaScript-Datei (main.min.js) im Ordner /dist.

In der Modulregel (module.rules) wird definiert, wie verschiedene Arten von Dateien verarbeitet werden sollen. In diesem Fall wird für JavaScript-Dateien der Babel-Loader verwendet, um es mit älteren Browsern kompatibel zu machen. Für CSS-Dateien wird das MiniCssExtractPlugin-Plugin verwendet, um eine separate CSS-Datei (main.min.css) zu erstellen. Das SASS-Loader wird verwendet, um SASS-Dateien in CSS zu kompilieren. Und für die jQuery-Bibliothek wird der Expose-Loader verwendet, um sie global verfügbar zu machen.

Verschiedene Plugins werden ebenfalls verwendet, um den Build-Prozess zu optimieren und zu verwalten. Dazu gehören das ProvidePlugin-Plugin, das das globale Verwenden von jQuery in der App ermöglicht, das CleanWebpackPlugin-Plugin, das den Ausgabepunkt vor jedem Build löscht, das MiniCssExtractPlugin-Plugin, das CSS-Dateien in eine separate Datei extrahiert, und das CleanTerminalPlugin-Plugin, das das Terminal vor jedem Build löscht.

Die Optimierungseinstellungen (optimization) sind für das Minimieren des Codes verantwortlich. Dazu werden der TerserPlugin-Plugin und der CssMinimizerPlugin-Plugin verwendet.

Schließlich werden einige Einstellungen für das Entwickeln (devtool, watch, watchOptions) festgelegt, um einen schnellen Entwicklungsprozess zu ermöglichen. Die Mode-Eigenschaft ist auf "production" gesetzt, um sicherzustellen, dass die App optimal kompiliert wird.

Übersicht aller PHP-Funktionen

Die folgenden Funktionen dienen als eine Art minimales Framework, dass helfen soll, effektiver und schneller das Markup der Seite zu erstellen

Die Funktion **get_header()** lädt den Header-Bereich der Seite, indem die Datei "header.php" eingebunden wird.

Die Funktion **get_footer()** lädt den Footer-Bereich der Seite, indem die Datei "footer.php" eingebunden wird.

Die Funktion **create_nav_bar(**$menuPosition, $links**)** generiert eine Navigationsleiste. Diese wird als ungepufferte Ausgabe erstellt und zurückgegeben.

Die Funktion **createVideoSlider(**$class, $courseData**)** erstellt einen Video-Slider mit den übergebenen Kursdaten. Der Slider wird als ungepufferte Ausgabe erstellt und zurückgegeben.

Die Funktion **get_template_uri**() gibt die Basis-URL des aktuellen Templates zurück. Sie verwendet Informationen über den Serverpfad und das Protokoll, um die URL zu generieren.

Die Funktion **get_home_url()** gibt die Startseite-URL der Website zurück. Sie verwendet Informationen über den Serverpfad, das Protokoll und die aktuelle Domain, um die URL zu generieren.

Die Funktion **get_body_class()** gibt den Wert für die CSS-Klassen des `<body>`-Elements zurück. Sie verwendet Informationen über den Serverpfad und die aktuelle URL, um die Klassen zu generieren.

Die Funktion **get_basepath()** gibt den Basispfad der aktuellen Skriptdatei zurück. Sie verwendet Informationen über den Pfad des Skripts, um den Basispfad zu bestimmen.

Login / Logout / Registrierung

Die Funktionen **initLogin(), initLogout()** und **initRegistration()** kümmern sich um d en Login, Logout und die Registrierung von Benutzern auf der Website.

initLogin():

Diese Funktion ist für den Login von Benutzern zuständig. Es werden verschiedene Variablen für die Eingabefelder, die Schaltfläche zum Einreichen des Formulars und das Behältnis für Fehlermeldungen definiert. Wenn der Benutzer auf die Schaltfläche klickt, werden die Eingabedaten in ein JSON-Objekt gepackt und an die API-Route /auth/login gesendet. Wenn die API antwortet, wird geprüft, ob ein gültiger Token zurückgegeben wurde. Wenn dies der Fall ist, wird der Token im Local Storage gespeichert und der Benutzer auf die Dashboard-Seite weitergeleitet. Wenn kein Token zurückgegeben wurde, wird dem Benutzer eine Fehlermeldung angezeigt.

initLogout():

Diese Funktion ist für den Logout von Benutzern zuständig. Wenn der Benutzer auf die Schaltfläche "Logout" klickt, wird eine Anfrage an die API-Route /auth/logout gesendet, um den aktuellen Token ungültig zu machen. Wenn die API antwortet, wird der Token aus dem Local Storage entfernt und der Benutzer wird auf die Startseite weitergeleitet.

initRegistration():

Diese Funktion ist für die Registrierung von Benutzern zuständig. Es werden verschiedene Variablen für die Eingabefelder, die Schaltfläche zum Einreichen des Formulars und das Behältnis für Fehlermeldungen definiert. Wenn der Benutzer auf die Schaltfläche klickt, werden die Eingabedaten in ein JSON-Objekt gepackt und an die API-Route /auth/register gesendet. Wenn die API antwortet, wird geprüft, ob ein gültiger Token zurückgegeben wurde. Wenn dies der Fall ist, wird der Token im Local Storage gespeichert und der Benutzer auf die Dashboard-Seite weitergeleitet. Wenn kein Token zurückgegeben wurde, wird dem Benutzer eine Fehlermeldung angezeigt.

Es gibt auch Hilfsfunktionen, die in diesen Funktionen verwendet werden, wie **animateErrorMessage(), getToken() und getUserData()**. **animateErrorMessage()** fügt eine CSS-Klasse zu einem Fehlermeldungselement hinzu und entfernt sie nach einer bestimmten Zeit wieder. **getToken()** ruft den aktuellen Token aus dem Local Storage ab und gibt ihn zurück, oder gibt null zurück, wenn kein Token gefunden wurde. **getUserData()** sendet eine Anfrage an die API-Route /auth/me und gibt die Daten des aktuellen Benutzers zurück, wenn ein gültiger Token gefunden wurde.

Übersicht aller Java-Script Funktionen

Die Funktion **addComment(courseID)** sendet einen POST-Request, um einen Kommentar zu einem Kurs hinzuzufügen. Der Kommentarinhalt wird aus einem Texteingabefeld abgerufen und zusammen mit dem Kurskennzeichen an den Server gesendet. Nachdem der Kommentar erfolgreich hinzugefügt wurde, wird die Funktion **loadCourseComments()** aufgerufen, um die aktualisierten Kommentare zu laden.

Die Funktion **loadCourseComments()** sendet einen GET-Request, um die Kommentare für einen bestimmten Kurs abzurufen. Sie überprüft zunächst, ob sich der Benutzer auf der Seite eines einzelnen Kurses befindet, und sendet dann den entsprechenden GET-Request. Die erhaltenen Kommentare werden an die **Funktion renderComments(**data, 3**)** übergeben, um sie anzuzeigen.

Die Funktion **addBalance(**amount**)** sendet einen POST-Request, um den Kontostand des Benutzers zu erhöhen. Der

übergebene Betrag wird an den Server gesendet, um den Kontostand des Benutzers entsprechend zu aktualisieren.

Die Funktion **courseVideoSlider()** initialisiert einen Bildlaufregler für die Videowiedergabe in einem Kurs. Sie konfiguriert den Bildlaufregler mit bestimmten Optionen wie Pfeiltasten, Punkteanzeige usw.

Die Funktion **getCourseProgress(**videos**)** berechnet den Fortschritt des Benutzers in einem Kurs anhand der übergebenen Liste von Videos. Sie ruft die Funktion **getWatchedVideos()** auf, um die bereits angesehenen Videos abzurufen. Dann zählt sie die Anzahl der Videos, die in der Liste watchedVideos enthalten sind, und berechnet den Fortschritt in Prozent. Der berechnete Fortschritt wird als Gleitkommazahl mit zwei Nachkommastellen zurückgegeben.

Die Funktion **addToWatchedVideos()** fügt eine Event-Listener-Funktion für das Ereignis "ended" (beendet) zu jedem Video auf der Seite hinzu. Sobald ein Video beendet wurde, wird die Funktion **updateVideoProgress(I**D, points**)** aufgerufen, um den Fortschritt des Videos zu aktualisieren. Anschließend wird ein POST-Request gesendet, um den Fortschritt des Videos an den Server zu übermitteln. Schließlich wird die Funktion **updateCourseProgress(**currentCourse) aufgerufen, um den Kursfortschritt zu aktualisieren.

Die Funktion **returnDifficulty(**integer**)** gibt den Schwierigkeitsgrad basierend auf einer gegebenen Ganzzahl zurück. Wenn die Zahl 1 ist, wird "Anfänger" zurückgegeben, wenn die Zahl 2 ist, wird "Fortgeschritten" zurückgegeben, und wenn die Zahl 3 ist, wird "Experte" zurückgegeben.

Die Funktion **getHomeUrl()** gibt die URL der Startseite der Website zurück.

Die Funktion **animateErrorMessage()** fügt einer Fehlermeldungskontainerklasse CSS-Klassen hinzu, um eine animierte Anzeige des Fehlers zu ermöglichen. Die Klasse wird nach einer Sekunde wieder entfernt.

Die Funktion **getToken(**verifyToken**)** ruft den Authentifizierungstoken des Benutzers ab. Wenn verifyToken nicht vorhanden oder falsch ist, wird der gespeicherte Token

zurückgegeben. Andernfalls wird ein GET-Request an den Server gesendet, um den Token zu überprüfen. Wenn die Überprüfung

Die Funktion **getUserData()** ruft Benutzerdaten ab, indem sie eine GET-Anfrage an eine API mit einem Token sendet.

Die Funktion **getWatchedVideos()** ruft die Liste der angesehenen Videos ab, indem sie eine GET-Anfrage an eine API mit einem Token sendet.

Die Funktion **checkSingleVideoProgress()** überprüft den Fortschritt eines einzelnen Videos, indem es die Liste der angesehenen Videos abruft und überprüft, ob das Video in der Liste enthalten ist.

Die Funktion **getOwnedProducts()** ruft die Liste der gekauften Produkte ab, indem sie eine GET-Anfrage an eine API mit einem Token sendet.

Die Funktion **updateVideoProgress()** aktualisiert den Fortschritt eines Videos und fügt Punkte hinzu, wenn das Video noch nicht abgeschlossen wurde.

Die Funktion **setCookie()** setzt einen Cookie mit einem Namen, Wert und Ablaufdatum.

Die Funktion **updateCourseProgress()** aktualisiert den Fortschritt eines Kurses und zeigt ihn an.

Die Funktion **purchaseCourse**() kauft einen Kurs, indem eine POST-Anfrage an eine API gesendet wird, abhängig davon, ob der Benutzer angemeldet ist oder nicht.

Die Funktion **currentUserLevel()** ermittelt das aktuelle Level des Benutzers basierend auf den gesammelten Punkten.

Die Funktion **initializeSearchFilter()** Initialisiert die Filterfunktion für die Suche in einer Liste von Kursen.

Die Funktion **validateUserInput()** validiert die Eingabe des Benutzernamens und des Passworts.

Die Funktion **returnUserInputAsObjec()** gibt die Benutzereingabe als Objekt zurück.

Die Funktion **validateUserRegistrationInput()** validiert die Eingabe bei der Benutzerregistrierung und zeigt ggf. Fehlermeldungen an.

Die Funktion **validateUserLoginInput()** validiert die Eingabe bei der Benutzeranmeldung und zeigt ggf. Fehlermeldungen an.

Die Funktion **processUserInput(**password, username**)** nimmt ein Passwort und einen Benutzernamen entgegen, wandelt sie in ein Objekt um und validiert die Eingabe. Das Ergebnis der Validierung wird zurückgegeben.

Die **Funktion renderAddComment(**courseID**)** rendert einen Kommentarbereich für einen bestimmten Kurs. Dabei wird ein HTML-Code erstellt und dem Container hinzugefügt. Es wird auch ein Event-Handler für den Klick auf den "Kommentar absenden"-Button hinzugefügt, der einen Kommentar hinzufügt, falls das Textfeld nicht leer ist.

Die Funktion renderComments(data, amount**)** rendert die vorhandenen Kommentare. Dabei wird der Container geleert und für jeden Kommentar wird HTML-Code generiert und dem Container hinzugefügt. Es wird auch ein "Mehr anzeigen"-Button hinzugefügt, um weitere Kommentare zu laden.

Die Funktion **renderNewProducts(**data, slider**)** rendert neue Produkte in einem Slider. Dabei wird für jedes Produkt HTML-Code generiert und dem Slider hinzugefügt.

Die **Funktion renderMostWatchedProducts(**data, slider**)** rendert die meistgesehenen Produkte in einem Slider. Dabei wird für jedes Produkt HTML-Code generiert und dem Slider hinzugefügt.

Die Funktion **renderOwnedProducts(**products**)** rendert die gekauften Produkte. Dabei wird für jedes Produkt HTML-Code generiert und dem Container hinzugefügt.

Die Funktion **renderSingleCourseResult(**data, container**)** rendert die Informationen für einen einzelnen Kurs. Dabei wird für jeden Kurs HTML-Code generiert und dem Container hinzugefügt. Je nachdem, ob der Benutzer den Kurs besitzt oder nicht, werden unterschiedliche Informationen angezeigt.

Die Funktion **renderCourse(**data, products, courseID**)** rendert den Inhalt eines Kurses. Dabei werden entweder die Kursvideos oder eine Vorschau des Kurses angezeigt, abhängig davon, ob der Benutzer den Kurs besitzt oder nicht. Zusätzlich werden Kommentare hinzugefügt, wenn der Benutzer den Kurs besitzt.

Die Funktion **renderCourseVideos(**videos**)** rendert die Videos eines Kurses. Dabei wird für jedes Video HTML-Code generiert und zurückgegeben.

Die **Funktion initLoadOwnedProducts()** lädt die vom Benutzer besessenen Produkte und rendert sie.

Die Funktion **initCookieNotice()** initialisiert das Cookie-Banner und verarbeitet die Interaktionen des Benutzers damit.

Die Funktion **initLoadUserData()** lädt die Benutzerdaten, falls ein Token vorhanden ist, und aktualisiert die entsprechenden Container auf der Benutzeroberfläche.

Die Funktion **initHandleUserContainer()** zeigt den entsprechenden Container für den eingeloggten oder ausgeloggten Benutzer an.

Die Funktion **initLogout()** behandelt den Logout-Prozess eines Benutzers.

Die Funktion **initLogin()** behandelt den Login-Prozess eines Benutzers.

Die Funktion **initRegistration()** behandelt den Registrierungsprozess eines Benutzers.

Die Funktion **initBurgerMenu()** initialisiert das Burger-Menü und behandelt die entsprechenden Interaktionen des Benutzers.

Die Funktion **initCourseSlider()** initialisiert den Kurs-Slider auf der Seite.

Die Funktion **initFrontpageSlider()** initialisiert den Slider auf der Startseite.

Die Funktion **initNewsSlider()** initialisiert den News-Slider auf der Seite.

Die Funktion **initMostWatchedSlider()** initialisiert den Slider für die am meisten angesehenen Produkte.

Die Funktion initScroll() initialisiert das Scroll-Verhalten der Seite.

Die Funktion **initLoadNewProducts()** lädt die neuen Produkte und rendert sie.

Die Funktion **initLoadAllProducts()** lädt alle Produkte und rendert sie auf der Kursseite.

Die Funktion **initLoadMostWatchedProducts()** lädt die am meisten angesehenen Produkte und rendert sie.

Die Funktion **initLoadSingleProduct()** lädt und rendert Informationen zu einem einzelnen Produkt.

Die Funktion **initDashboardNavigation()** initialisiert die Navigation auf der Dashboard-Seite.

Die Funktion **initIncreaseUserBalance()** behandelt die Erhöhung des Benutzerkontostands.

Die Funktion **initLoadCurrentUserLevel()** lädt und rendert das aktuelle Level des Benutzers.

Die Funktion **initVisiblePassword()** behandelt die Sichtbarkeit des Passwortfelds bei der Passworteingabe

Backend-Übersicht

Tech Stack: Golang + Postgres + Nginx + Docker

Service-basierte Architektur

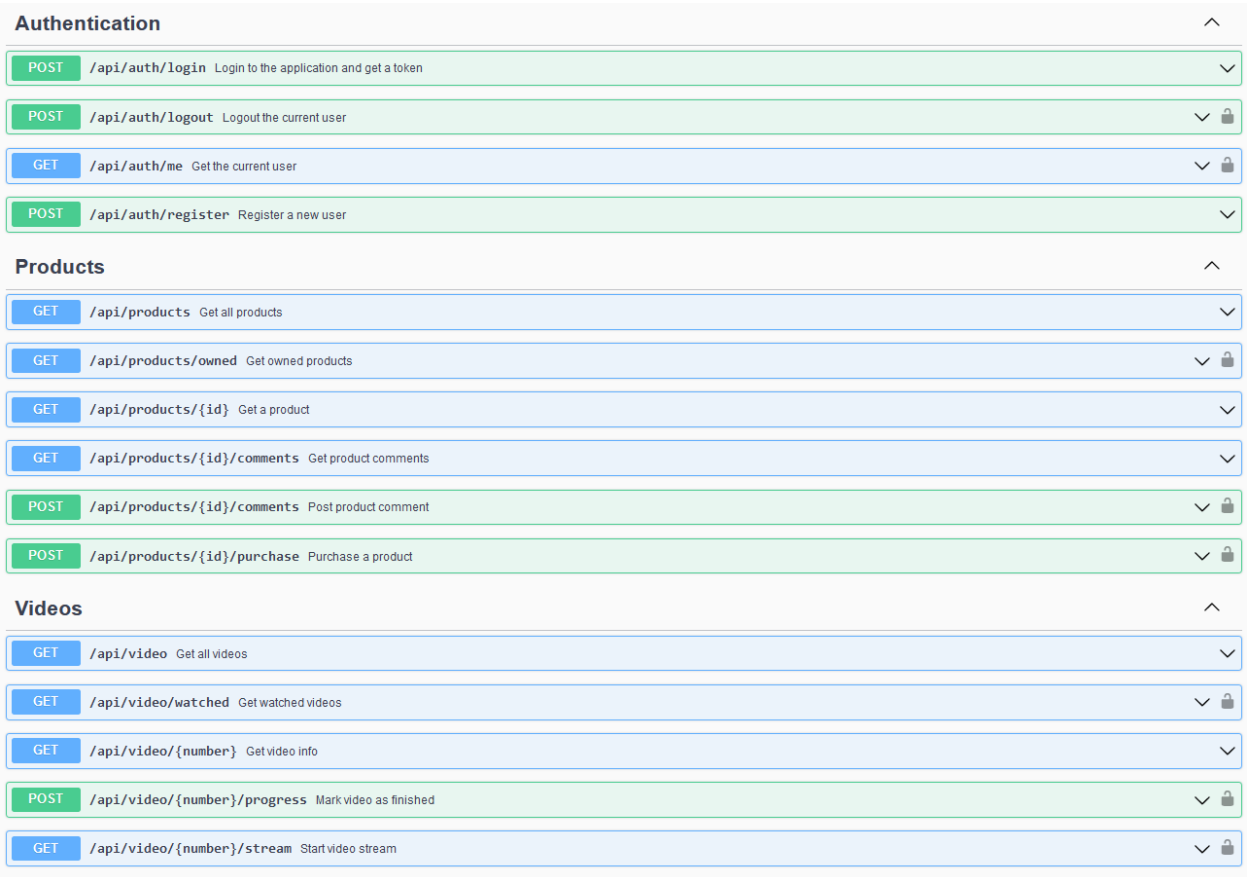
Wieso Golang?

- Für Web APIs ursprünglich gedacht
- Hochperformant
- Einfach zu skalieren
- Statisch kompilierbar
- Integriertes Unit-Testing System

Unit-Testing

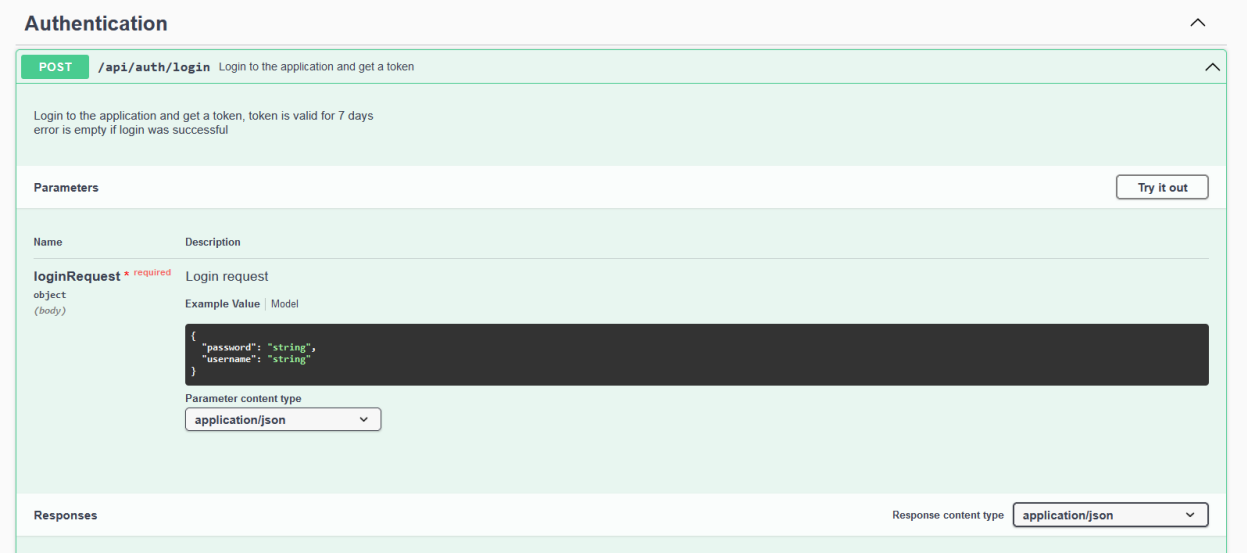
- Akkurater Code ist wichtig
- Vereinfacht spätere Verbesserungen
- Automatisches Testen bei Git-Push

Swagger/OpenAPI



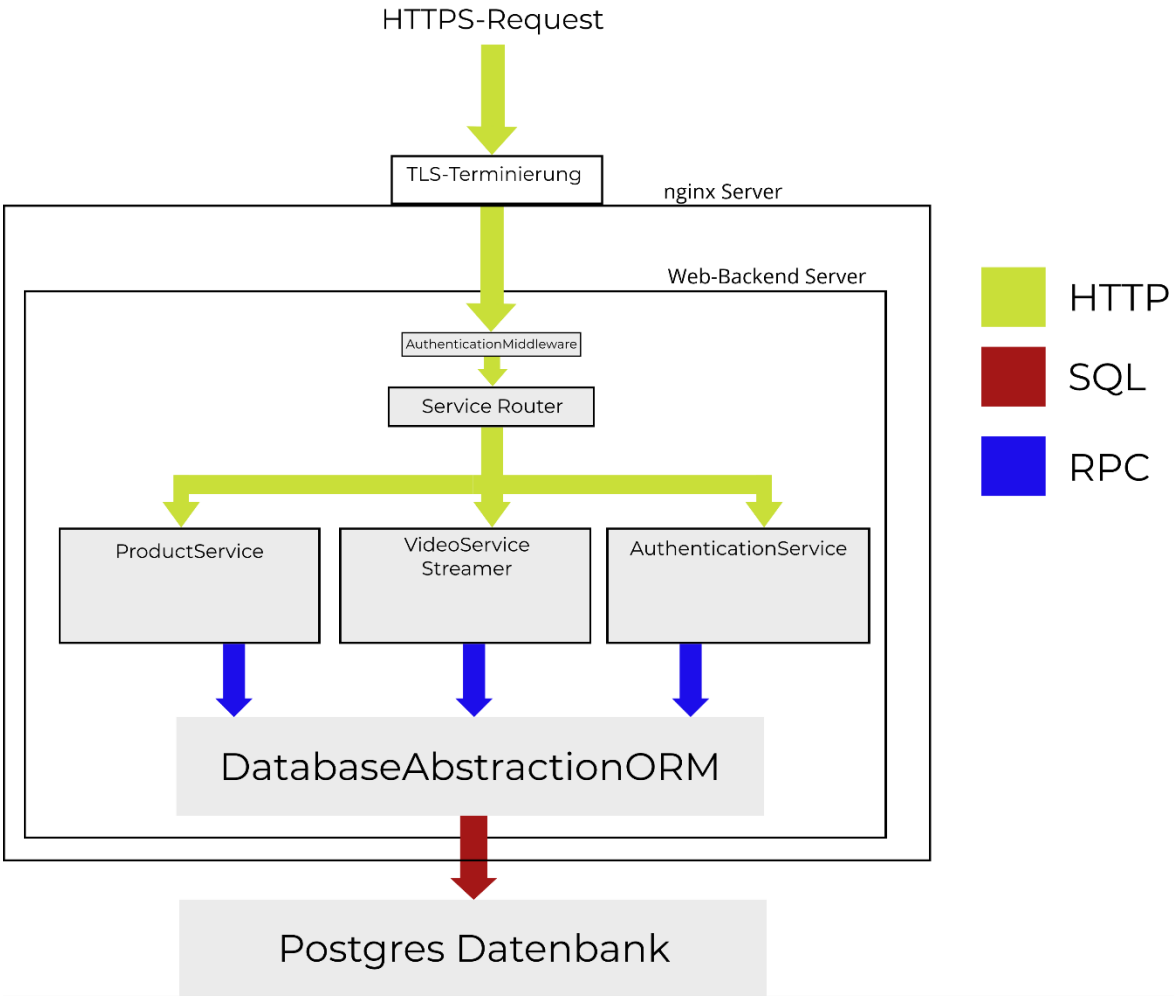
```
// Login godoc
//
//  @Summary        Login to the application and get a token
//  @Description    Login to the application and get a token, token is valid for 7 days
//  @Description    error is empty if login was successful
//  @Tags           Authentication
//  @Accept         json
//  @Produce        json
//  @Param          loginRequest    body        loginRequest    true    "Login request"
//  @Success        200             {object}    loginResponse
//  @Failure        400             {object}    loginResponse
//  @Failure        401             {object}    loginResponse
//  @Failure        500             {object}    loginResponse
//  @Router         /api/auth/login [post]
func (am AuthenticationService) Login(ctx *gin.Context) {  5 usages  PrivateGER
    var request loginRequest
```

Automatisch generierte API-Dokumentation basierend auf Docucomments im Code



Wie hier zu sehen ist jeder Endpoint durch Swagger sehr gut dokumentiert und kann sogar auf der Dokumentationsseite getestet werden

Service-Architektur



Dies ist der Ablauf einer HTTP-Anfrage an das Backend von BKBdemy.

Nginx kümmert sich um die TLS-Terminierung, daraufhin wird die Request an den Docker-Container welcher das Backend enthält weitergeleitet.

Dort liest die AuthenticationMiddleware die Anfrage, sollte diese einen Authentication header oder einen Auth-Cookie enthalten wird dieser überprüft und Metadaten über den Nutzer werden an die Anfrage angehangen

Danach erhält der Service Router die Anfrage und routet sie dem Anfragepfad entsprechend zu dem zuständigen Service. Jeder Service ist gekapselt und kümmert sich um eine bestimmte Art von Anfragen. Hierdurch ist Maintainability einfacher.

Services können DatabaseAbstractionORM nutzen, dürfen aber nicht selber die SQL Datenbank-Verbindungen eröffnen.

Beispiel: Video-Streaming Implementation

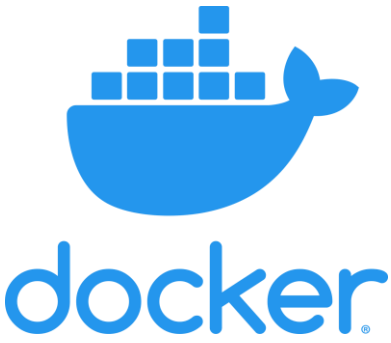
```go
// StartVideoStream godoc
// @Summary Start video stream
// @Description Start video stream
// @Tags Videos
// @Param number path int true "VSVideo ID"
// @Success 200
// @Security ApiKeyAuth
// @Router /api/video/{number}/stream [get]
func (V VSService) StartVideoStream(c *gin.Context) {  2 usages  PrivateGER *
    videoID := c.Param( key: "number")

    // Convert videoID to int
    videoIDInt, err := strconv.Atoi(videoID)

    // Retrieve video filename from database
    video, err := V.DB.GetVideoByIndexID(videoIDInt)
    if err != nil {
        c.JSON( code: 500, gin.H{"error": err.Error()})
        return
    }

    // Start stream of file with basepath + filename
    serveFile(c, video.Filename)
}
```

Dockerization (Front- & Backend)



- Containerisierung des Frontend und Backend
- Docker-Container sind vom Host-System weitestgehend unabhängig
    - Alle Dependencies sind in dem Container enthalten

    - Ermöglicht konsistente Umgebungen, sonst praktisch unmöglich
- Keine klassische VMs; daher hochperformant
- Vereinfachte Skalierbarkeit
- Reproduzierbare Fehler
- Gängige Betriebssysteme unterstützen Docker (Linux+Windows+OS X)

Blackbox-Tests-BKBdemy

Nach dem wir in unserer Schulischen Projektphase die Website „BKBdemy“ erschaffen haben, habe ich (Luca Pothmann) meine Arbeitskollegen die Website testen lassen, ohne vorher ein Wort zum Inhalt gesagt zu haben, bis auf die Testfälle und deren Schritte, wie z.B. bei dem Testfall „Anmelden auf der BKBdemy-Website“. Auf den folgenden Seiten werden sie diese sehen.

Der Quellcode ist unter: https://github.com/BKBdemy zu finden

Testfall: Anmelden auf der BKBdemy-Website

Ziel des Tests: Überprüfung der Anmeldefunktion auf der BKBdemy-Website, um sicherzustellen, dass Benutzer sich erfolgreich anmelden können.

Vorbedingungen:

- Die BKBdemy-Website ist vollständig geladen und einsatzbereit.
- Es gibt ein gültiges Benutzerkonto für den Test.

Schritte:

1. Öffne den Webbrowser.
2. Navigiere zur BKBdemy-Website (URL: https://bkbdemy.pxroute.net/).
3. Überprüfe, ob die Startseite erfolgreich geladen wird.
4. Suche nach dem Anmeldeformular auf der Startseite.
5. Klicke auf den Anmelde-Link oder das Anmeldeformular, um zur Anmeldeseite zu gelangen.
6. Überprüfe, ob die Anmeldeseite erfolgreich geladen wird.
7. Fülle das Benutzernamen-Feld mit einem gültigen Benutzernamen ein.
8. Fülle das Passwort-Feld mit einem gültigen Passwort ein.
9. Klicke auf den "Anmelden" oder "Login"-Button.
10. Überprüfe, ob der Benutzer erfolgreich angemeldet wird.
11. Überprüfe, ob der Benutzer auf die Benutzerprofilseite weitergeleitet wird.
12. Überprüfe, ob das Benutzerprofil korrekt angezeigt wird.

Erwartete Ergebnisse:

Bei erfolgreichem Testdurchlauf:

- Der Benutzer wird erfolgreich angemeldet.
- Der Benutzer wird auf die Benutzerprofilseite weitergeleitet.
- Das Benutzerprofil wird korrekt angezeigt.

Bei einem fehlgeschlagenen Testdurchlauf:

- Der Benutzer wird nicht erfolgreich angemeldet.

- Der Benutzer wird nicht auf die Benutzerprofilseite weitergeleitet.
- Es werden Fehlermeldungen oder unerwartete Ergebnisse angezeigt.

Testergebnisse:

Testfall-ID: BKB-001

Datum des Tests: 02.06.2023

Testumgebung: Windows 11, Microsoft Edge

Testergebnis: **Erfolgreich**

Testfall: Überprüfung der Registrierungsfunktion auf Bkbdemy

Ziel des Tests: Überprüfung, ob Benutzer sich erfolgreich auf Bkbdemy registrieren können.

Vorbedingungen:

- Die Bkbdemy-Website ist vollständig geladen und einsatzbereit.

Schritte:

1. Öffne den Webbrowser.
2. Navigiere zur Bkbdemy-Website (URL: https://bkbdemy.pxroute.net/).
3. Überprüfe, ob die Startseite erfolgreich geladen wird.
4. Suche nach dem Registrierungsformular auf der Startseite.
5. Klicke auf den Link oder die Schaltfläche "Registrieren".
6. Überprüfe, ob das Registrierungsformular erfolgreich geladen wird.
7. Fülle alle erforderlichen Felder im Registrierungsformular aus.
8. Überprüfe, ob alle eingegebenen Daten korrekt angezeigt werden.
9. Klicke auf die Schaltfläche "Registrieren", um den Registrierungsvorgang abzuschließen.
10. Überprüfe, ob eine Bestätigungsmeldung für die erfolgreiche Registrierung angezeigt wird.
11. Überprüfe, ob der Benutzer automatisch zur Willkommensseite weitergeleitet wird.

Erwartete Ergebnisse:

Bei erfolgreichem Testdurchlauf:

- Das Registrierungsformular wird erfolgreich geladen.
- Alle eingegebenen Daten werden korrekt angezeigt.
- Eine Bestätigungsmeldung für die erfolgreiche Registrierung wird angezeigt.
- Der Benutzer wird automatisch zur Anmeldeseite weitergeleitet.

Bei einem fehlgeschlagenen Testdurchlauf:

- Das Registrierungsformular wird nicht angezeigt oder ist unvollständig.
- Die eingegebenen Daten werden nicht korrekt angezeigt.

- Es wird keine Bestätigungsmeldung für die erfolgreiche Registrierung angezeigt.
- Der Benutzer wird nicht zur Anmeldeseite weitergeleitet.

Testergebnisse:

Testfall-ID: BKB-002

Datum des Tests: 02.06.2023

Testumgebung: Windows 11, Microsoft Edge

Testergebnis: **Erfolgreich**

Testfall: Überprüfung der Kursdetails auf Bkbdemy

Ziel des Tests: Überprüfung, ob die Kursdetails auf Bkbdemy korrekt angezeigt werden.

Vorbedingungen:

- Der Benutzer hat Zugriff auf die Bkbdemy-Website.
- Mindestens ein Kurs ist auf der Website vorhanden.

Schritte:

1. Öffne den Webbrowser.
2. Navigiere zur Bkbdemy-Website (URL: https://bkbdemy.pxroute.net/).
3. Überprüfe, ob die Startseite erfolgreich geladen wird.
4. Suche nach der Kursliste oder dem Katalog auf der Startseite.
5. Wähle einen beliebigen Kurs aus der Liste aus.
6. Überprüfe, ob die Kursdetailsseite erfolgreich geladen wird.
7. Überprüfe, ob der Kursname, die Beschreibung, der Preis und die Kursdauer korrekt angezeigt werden.
8. Überprüfe, ob Informationen zu Kursleitern oder Dozenten vorhanden sind.
9. Überprüfe, ob Kundenkommentare zum Kurs angezeigt werden.
10. Klicke auf die Schaltfläche "Diesen Kurs kaufen" um den Buchungsvorgang zu starten.
11. Überprüfe, ob der Benutzer zum Guthaben aufladen verwiesen wird.

Erwartete Ergebnisse:

Bei erfolgreichem Testdurchlauf:

- Die Kursdetailsseite wird erfolgreich geladen.
- Der Kursname, die Beschreibung, der Preis und die Kursdauer werden korrekt angezeigt.
- Informationen zu Kursleitern oder Dozenten sind vorhanden.
- Kundenbewertungen oder -kommentare zum Kurs werden angezeigt.
- Der Benutzer wird zur Zahlungsseite weitergeleitet, wenn er den Kurs buchen möchte.

Bei einem fehlgeschlagenen Testdurchlauf:

- Die Kursdetailsseite wird nicht angezeigt oder ist unvollständig.
- Die Kursinformationen werden nicht korrekt angezeigt.
- Informationen zu Kursleitern oder Dozenten fehlen.
- Kundenbewertungen oder -kommentare zum Kurs werden nicht angezeigt.
- Der Benutzer wird nicht zur Zahlungsseite weitergeleitet.

Testergebnisse:

Testfall-ID: BKB-003

Datum des Tests: 02.06.2023

Testumgebung: Windows 11, Microsoft Edge

Testergebnis: **Erfolgreich**

Testfall: Überprüfung der Registrierungsfunktion auf Bkbdemy

Ziel des Tests: Überprüfung, ob die Registrierungsfunktion auf Bkbdemy ordnungsgemäß funktioniert.

Vorbedingungen:

- Der Benutzer hat Zugriff auf die Bkbdemy-Website.
- Der Benutzer hat noch keinen registrierten Account auf Bkbdemy.

Schritte:

1. Öffne den Webbrowser.
2. Navigiere zur Bkbdemy-Website (URL: https://bkbdemy.pxroute.net/).
3. Überprüfe, ob die Startseite erfolgreich geladen wird.
4. Suche nach dem Registrierungsformular oder der Registrierungsschaltfläche auf der Startseite.
5. Klicke auf die Schaltfläche "Registrieren" oder eine ähnliche Schaltfläche, um zur Registrierungsseite zu gelangen.
6. Überprüfe, ob die Registrierungsseite erfolgreich geladen wird.
7. Fülle alle erforderlichen Felder im Registrierungsformular aus, wie z.B. Vorname, Nachname, E-Mail-Adresse und Passwort.
8. Überprüfe, ob die eingegebenen Daten korrekt validiert werden (z.B. Überprüfung der E-Mail-Adresse auf Gültigkeit).
9. Klicke auf die Schaltfläche "Registrieren" oder eine ähnliche Schaltfläche, um den Registrierungsvorgang abzuschließen.

10. Überprüfe, ob der Benutzer nach erfolgreicher Registrierung zur Startseite weitergeleitet wird.
11. Versuche dich mit den gerade registrierten Anmeldedaten einzuloggen und überprüfe, ob der Login erfolgreich ist.

Erwartete Ergebnisse:

Bei erfolgreichem Testdurchlauf:

- Die Registrierungsseite wird erfolgreich geladen.
- Alle erforderlichen Felder im Registrierungsformular können korrekt ausgefüllt werden.
- Die eingegebenen Daten werden korrekt validiert.
- Nach erfolgreicher Registrierung wird der Benutzer zur Willkommensseite weitergeleitet.
- Der Login mit den gerade registrierten Anmeldedaten ist erfolgreich.

Bei einem fehlgeschlagenen Testdurchlauf:

- Die Registrierungsseite wird nicht angezeigt oder ist unvollständig.
- Es treten Validierungsfehler bei der Eingabe der Daten auf.
- Der Benutzer wird nicht zur Startseite weitergeleitet nach der Registrierung.
- Der Login mit den gerade registrierten Anmeldedaten schlägt fehl.

Testergebnisse:

Testfall-ID: BKB-004

Datum des Tests: 02.06.2023

Testumgebung: Windows 11, Microsoft Edge

Testergebnis: **Erfolgreich**

Testfall: Überprüfung der Suchfunktion auf Bkbdemy

Ziel des Tests: Überprüfung, ob die Suchfunktion auf Bkbdemy korrekt funktioniert und relevante Ergebnisse liefert.

Vorbedingungen:

- Der Benutzer hat Zugriff auf die Bkbdemy-Website.
- Es sind bereits Kurse auf der Website vorhanden.

Schritte:

1. Öffne den Webbrowser.
2. Navigiere zur Bkbdemy-Website (URL: https://bkbdemy.pxroute.net/).
3. Überprüfe, ob die Startseite erfolgreich geladen wird.
4. Suche nach dem Suchfeld auf der Startseite.

5. Gib einen Suchbegriff ein, der mit einem vorhandenen Kurs auf Bkbdemy in Verbindung steht.
6. Klicke auf die Suchschaltfläche oder drücke die Eingabetaste, um die Suche zu starten.
7. Überprüfe, ob die Suchergebnisse korrekt angezeigt werden und relevant zum eingegebenen Suchbegriff sind.
8. Klicke auf eines der Suchergebnisse, um zur Kursdetailseite zu gelangen.
9. Überprüfe, ob die Kursdetailseite erfolgreich geladen wird und die Informationen des ausgewählten Kurses angezeigt werden.

Erwartete Ergebnisse:

Bei erfolgreichem Testdurchlauf:

- Die Startseite wird erfolgreich geladen.
- Das Suchfeld ist vorhanden und funktionsfähig.
- Die Suchergebnisse werden korrekt angezeigt und sind relevant zum eingegebenen Suchbegriff.
- Die Kursdetailseite wird erfolgreich geladen und zeigt die Informationen des ausgewählten Kurses an.

Bei einem fehlgeschlagenen Testdurchlauf:

- Die Startseite wird nicht angezeigt oder ist unvollständig.
- Das Suchfeld ist nicht vorhanden oder funktioniert nicht wie erwartet.
- Die Suchergebnisse werden nicht korrekt angezeigt oder sind irrelevant zum eingegebenen Suchbegriff.
- Die Kursdetailseite wird nicht erfolgreich geladen oder zeigt falsche Informationen an.

Testergebnisse:

Testfall-ID: BKB-005

Datum des Tests: 02.06.2023

Testumgebung: Windows 11, Microsoft Edge

Testergebnis: **Erfolgreich**

Testfall: Überprüfung der Registrierungsfunktion auf Bkbdemy

Ziel des Tests: Überprüfung, ob die Registrierungsfunktion auf Bkbdemy korrekt funktioniert und Benutzer erfolgreich registriert werden können.

Vorbedingungen:

- Der Benutzer hat Zugriff auf die Bkbdemy-Website.
- Der Benutzer hat noch keinen vorhandenen Account auf Bkbdemy.

Schritte:

1. Öffne den Webbrowser.
2. Navigiere zur Bkbdemy-Website (URL: https://bkbdemy.pxroute.net/).
3. Überprüfe, ob die Startseite erfolgreich geladen wird.
4. Suche nach dem Registrierungslink oder der Registrierungsschaltfläche auf der Startseite.
5. Klicke auf den Registrierungslink oder die Registrierungsschaltfläche, um zum Registrierungsformular zu gelangen.
6. Fülle alle erforderlichen Felder des Registrierungsformulars aus (Benutzername, Passwort).
7. Überprüfe, ob alle eingegebenen Daten korrekt und vollständig erfasst wurden.
8. Klicke auf die Registrierungsschaltfläche, um die Registrierung abzuschließen.
9. Überprüfe, ob eine Bestätigungsmeldung oder Weiterleitung zur Willkommensseite erfolgt.

Erwartete Ergebnisse:

Bei erfolgreichem Testdurchlauf:

- Die Startseite wird erfolgreich geladen.
- Der Registrierungslink oder die Registrierungsschaltfläche ist vorhanden und führt zum Registrierungsformular.
- Das Registrierungsformular erfasst alle erforderlichen Daten korrekt und vollständig.
- Die Nutzungsbedingungen und Datenschutzrichtlinien können akzeptiert werden.
- Nach Abschluss der Registrierung erfolgt eine Bestätigungsmeldung oder Weiterleitung zur Anmeldeseite.

Bei einem fehlgeschlagenen Testdurchlauf:

- Die Startseite wird nicht angezeigt oder ist unvollständig.
- Der Registrierungslink oder die Registrierungsschaltfläche ist nicht vorhanden oder führt nicht zum Registrierungsformular.
- Das Registrierungsformular erfasst die Daten nicht korrekt oder lässt wichtige Felder aus.
- Die Nutzungsbedingungen oder Datenschutzrichtlinien können nicht akzeptiert werden.
- Es erfolgt keine Bestätigungsmeldung oder Weiterleitung zur Anmeldeseite nach Abschluss der Registrierung.

Testergebnisse:

Testfall-ID: BKB-006

Datum des Tests: 02.06.2023

Testumgebung: Windows 11, Microsoft Edge

Testergebnis: **Erfolgreich**

Testfall: Überprüfung der Kursliste (Shop) auf Bkbdemy

Ziel des Tests: Überprüfung, ob die Kursliste auf Bkbdemy korrekt angezeigt wird und alle relevanten Informationen zu den Kursen enthalten sind.

Vorbedingungen:

- Der Benutzer hat Zugriff auf die Bkbdemy-Website.
- Die Kursliste enthält mindestens einen veröffentlichten Kurs.

Schritte:

1. Öffne den Webbrowser.
2. Navigiere zur Bkbdemy-Website (URL: https://bkbdemy.pxroute.net/).
3. Überprüfe, ob die Startseite erfolgreich geladen wird.
4. Suche nach dem Menüpunkt oder der Schaltfläche, die zum Shop führt.
5. Klicke auf den Menüpunkt oder die Schaltfläche, um zum Shop zu gelangen.
6. Überprüfe, ob die Kursliste vollständig angezeigt wird und alle veröffentlichten Kurse aufgelistet sind.
7. Überprüfe für jeden Kurs in der im Shop:
8. Kursname ist sichtbar.
9. Kursbeschreibung ist vorhanden.
10. Kursdauer oder -zeitplan ist angegeben.
11. Kurspreis oder Preisinformationen sind angezeigt.
12. Kursbild oder Vorschaubild ist vorhanden.
13. Klicke auf einen beliebigen Kurs in der Liste, um zur Kursdetails-Seite zu gelangen.
14. Überprüfe, ob die Kursdetails-Seite korrekt geladen wird und alle relevanten Informationen zum ausgewählten Kurs angezeigt werden.

Erwartete Ergebnisse:

Bei erfolgreichem Testdurchlauf:

- Die Startseite wird erfolgreich geladen.
- Der Menüpunkt oder die Schaltfläche zur Kursliste ist vorhanden und führt zur Kursliste.
- Die Kursliste enthält alle veröffentlichten Kurse mit den korrekten Informationen zu jedem Kurs.
- Beim Klicken auf einen Kurs gelangt man zur Kursdetails-Seite mit den korrekten Informationen.

Bei einem fehlgeschlagenen Testdurchlauf:

- Die Startseite wird nicht angezeigt oder ist unvollständig.
- Der Menüpunkt oder die Schaltfläche zur Kursliste ist nicht vorhanden oder führt nicht zur Kursliste.
- Die Kursliste enthält fehlende oder falsche Informationen zu den Kursen.
- Beim Klicken auf einen Kurs gelangt man nicht zur korrekten Kursdetails-Seite oder es werden wichtige Informationen nicht angezeigt.

Testergebnisse:

Testfall-ID: BKB-007

Datum des Tests: 02.06.2023

Testumgebung: Windows 11, Microsoft Edge

Testergebnis: **Erfolgreich**

Testfall: Überprüfung der Registrierungsfunktion auf Bkbdemy

Ziel des Tests: Überprüfung, ob die Registrierungsfunktion auf Bkbdemy ordnungsgemäß funktioniert und Benutzer erfolgreich ein neues Konto erstellen können.

Vorbedingungen:

- Der Benutzer hat Zugriff auf die Bkbdemy-Website.
- Die Registrierungsfunktion ist verfügbar und aktiviert.

Schritte:

1. Öffne den Webbrowser.
2. Navigiere zur Bkbdemy-Website (URL: https://bkbdemy.pxroute.net/).
3. Überprüfe, ob die Startseite erfolgreich geladen wird.
4. Suche nach dem Registrierungslink oder der Schaltfläche.
5. Klicke auf den Registrierungslink oder die Schaltfläche, um zur Registrierungsseite zu gelangen.
6. Überprüfe, ob die Registrierungsseite korrekt geladen wird und alle erforderlichen Eingabefelder vorhanden sind (z. B. Vorname, Nachname, E-Mail, Passwort).
7. Fülle alle erforderlichen Felder mit gültigen Daten aus.
8. Klicke auf die Schaltfläche "Registrieren" oder eine ähnliche Schaltfläche, um die Registrierung abzuschließen.
9. Überprüfe, ob der Benutzer nach erfolgreicher Registrierung auf die Bestätigungsseite weitergeleitet wird.
10. Überprüfe die Bestätigungsseite auf eine Erfolgsmeldung und weitere Anweisungen für den Benutzer (z. B. E-Mail-Verifizierung).

Erwartete Ergebnisse:

Bei erfolgreichem Testdurchlauf:

Die Startseite wird erfolgreich geladen.

Der Registrierungslink oder die Schaltfläche ist vorhanden und führt zur Registrierungsseite.

Die Registrierungsseite enthält alle erforderlichen Eingabefelder und ist korrekt formatiert.

Die Registrierung mit gültigen Daten führt zur Bestätigungsseite.

Die Bestätigungsseite enthält eine Erfolgsmeldung und weitere Anweisungen für den Benutzer.

Bei einem fehlgeschlagenen Testdurchlauf:

- Die Startseite wird nicht angezeigt oder ist unvollständig.
- Der Registrierungslink oder die Schaltfläche ist nicht vorhanden oder führt nicht zur Registrierungsseite.
- Die Registrierungsseite enthält fehlende oder falsch formatierte Eingabefelder.
- Die Registrierung führt nicht zur Bestätigungsseite oder es treten Fehlermeldungen auf.

Testergebnisse:

Testfall-ID: BKB-008

Datum des Tests: 02.06.2023

Testumgebung: Windows 11, Microsoft Edge

Testergebnis: **Erfolgreich**

Testfall: Überprüfung der Suchfunktion auf Bkbdemy

Ziel des Tests: Überprüfung, ob die Suchfunktion auf Bkbdemy ordnungsgemäß funktioniert und Benutzer relevante Ergebnisse erhalten.

Vorbedingungen:

- Der Benutzer hat Zugriff auf die Bkbdemy-Website.
- Die Suchfunktion ist verfügbar und aktiviert.

Schritte:

1. Öffne den Webbrowser.
2. Navigiere zur Bkbdemy-Website (URL: https://bkbdemy.pxroute.net/).
3. Überprüfe, ob die Startseite erfolgreich geladen wird.
4. Suche nach dem Suchfeld oder der Suchschaltfläche.
5. Gebe einen Suchbegriff ein, der in Bezug auf das gewünschte Thema relevante Ergebnisse liefern sollte.
6. Drücke die "Enter"-Taste oder klicke auf die Suchschaltfläche, um die Suche zu starten.
7. Überprüfe die Suchergebnisseite auf relevante Ergebnisse.
8. Überprüfe, ob die Suchergebnisse nach Relevanz oder Datum sortiert sind.
9. Klicke auf ein Suchergebnis, um zur detaillierten Ansicht zu gelangen.
10. Überprüfe, ob die detaillierte Ansicht korrekt geladen wird und alle relevanten Informationen enthält.

Erwartete Ergebnisse:

Bei erfolgreichem Testdurchlauf:

- Die Startseite wird erfolgreich geladen.
- Das Suchfeld oder die Suchschaltfläche ist vorhanden.
- Die Suche liefert relevante Ergebnisse zum eingegebenen Suchbegriff.
- Die Suchergebnisse sind nach Relevanz oder Datum sortiert.
- Die detaillierte Ansicht eines Suchergebnisses enthält alle relevanten Informationen.

Bei einem fehlgeschlagenen Testdurchlauf:

- Die Startseite wird nicht angezeigt oder ist unvollständig.
- Das Suchfeld oder die Suchschaltfläche ist nicht vorhanden oder funktioniert nicht.
- Die Suche liefert keine oder irrelevante Ergebnisse.
- Die Suchergebnisse sind nicht korrekt sortiert.
- Die detaillierte Ansicht eines Suchergebnisses fehlt oder enthält fehlerhafte Informationen.

Testergebnisse:

Testfall-ID: BKB-009

Datum des Tests: 02.06.2023

Testumgebung: Windows 11, Microsoft Edge

Testergebnis: **Erfolgreich**

Testfall: Überprüfung der Registrierungsfunktion auf Bkbdemy

Ziel des Tests: Überprüfen, ob die Registrierungsfunktion auf Bkbdemy ordnungsgemäß funktioniert und Benutzer sich erfolgreich registrieren können.

Vorbedingungen:

- Der Benutzer hat Zugriff auf die Bkbdemy-Website.
- Die Registrierungsfunktion ist verfügbar und aktiviert.

Schritte:

1. Öffne den Webbrowser.
2. Navigiere zur Bkbdemy-Website (URL: https://bkbdemy.pxroute.net/).
3. Überprüfe, ob die Startseite erfolgreich geladen wird.
4. Suche nach dem Registrierungslink oder der Schaltfläche.
5. Klicke auf den Registrierungslink oder die Registrierungsschaltfläche, um zur Registrierungsseite zu gelangen.
6. Überprüfe, ob das Registrierungsformular angezeigt wird.
7. Fülle alle erforderlichen Felder im Registrierungsformular aus
8. Klicke auf die "Registrieren" Schaltfläche, um den Registrierungsvorgang abzuschließen.
9. Überprüfe, ob eine Bestätigungsmeldung angezeigt wird, die den erfolgreichen Abschluss der Registrierung bestätigt.

10. Versuche, dich mit den gerade erstellten Anmeldedaten anzumelden, um die Registrierung zu verifizieren.

Erwartete Ergebnisse:

Bei erfolgreichem Testdurchlauf:

- Die Startseite wird erfolgreich geladen.
- Der Registrierungslink oder die Registrierungsschaltfläche ist vorhanden und funktioniert.
- Das Registrierungsformular wird korrekt angezeigt.
- Alle erforderlichen Felder im Registrierungsformular werden akzeptiert.
- Eine Bestätigungsmeldung für die erfolgreiche Registrierung wird angezeigt.
- Die Anmeldung mit den neu erstellten Anmeldedaten ist erfolgreich.

Bei einem fehlgeschlagenen Testdurchlauf:

- Die Startseite wird nicht angezeigt oder ist unvollständig.
- Der Registrierungslink oder die Registrierungsschaltfläche ist nicht vorhanden oder funktioniert nicht.
- Das Registrierungsformular wird nicht korrekt angezeigt oder lässt sich nicht ausfüllen.
- Erforderliche Felder im Registrierungsformular werden nicht akzeptiert.
- Es wird keine Bestätigungsmeldung für die erfolgreiche Registrierung angezeigt.
- Die Anmeldung mit den neu erstellten Anmeldedaten schlägt fehl.

Testergebnisse:

Testfall-ID: BKB-010

Datum des Tests: 02.06.2023

Testumgebung: Windows 11, Microsoft Edge

Testergebnis: **Erfolgreich**

Unit-Test-Dokumentation: BKBdemy

Einführung

Der Unit-Test für die BKBdemy Website wurde entwickelt, um die Funktionalität und Zuverlässigkeit der wichtigsten Features der Website zu überprüfen. Der Test umfasst verschiedene Testmethoden, die verschiedene Aspekte der Website abdecken, wie die Homepage, den Login, die Registrierung, die Suche nach Kursen und die Einschreibung in einen Kurs. Durch die Durchführung dieser Tests können potenzielle Fehler oder Probleme in der Implementierung frühzeitig erkannt und behoben werden.

Umgebung

Programmiersprache: JS, HTML, PHP, CSS, SQL, Go

Test-Framework: unittest

Testabdeckung

Die folgenden Aspekte der Website werden im Unit-Test überprüft:

Homepage

Überprüft, ob die Homepage erfolgreich geladen wird und den erwarteten Inhalt enthält. Hierbei werden verschiedene Elemente der Homepage geprüft, wie das Logo, die Navigationsleiste und die Hauptbanner.

Login

Überprüft, ob ein Benutzer erfolgreich einloggen kann und eine Bestätigungsmeldung erhält. Dabei wird getestet, ob das Login-Formular korrekt funktioniert, ob Benutzerdaten validiert werden und ob der Login-Vorgang reibungslos verläuft.

Registrierung

Überprüft, ob ein neuer Benutzer erfolgreich registriert werden kann und eine Bestätigungsmeldung erhält. Hierbei wird getestet, ob das Registrierungsformular korrekt ausgefüllt werden kann, ob Benutzerdaten validiert und gespeichert werden und ob die Registrierung erfolgreich abgeschlossen wird.

Suche nach Kursen

Überprüft, ob die Suchfunktion korrekt funktioniert und die erwarteten Suchergebnisse liefert. Hierbei werden verschiedene Suchanfragen durchgeführt und überprüft, ob die zurückgegebenen Suchergebnisse den erwarteten Kriterien entsprechen.

Einschreibung in einen Kurs

Überprüft, ob ein Benutzer erfolgreich in einen Kurs eingeschrieben werden kann und eine Bestätigungsmeldung erhält. Dabei wird getestet, ob der Einschreibungsprozess korrekt funktioniert, ob der Benutzer erfolgreich zu einem Kurs hinzugefügt wird und ob die entsprechenden Informationen ordnungsgemäß aktualisiert werden.

Durchführung der Tests

Um die Tests auszuführen, können folgende Schritte befolgt werden:

Stellen Sie sicher, dass die BKBdemy Website lokal oder in einer Testumgebung bereitgestellt ist.

Starten Sie das Test-Framework und führen Sie die Unit-Tests aus.

Testergebnisse

Die Unit-Tests für die BKBdemy Website liefern folgende Testergebnisse:

Homepage-Test: Erfolgreich

- Das Logo wird korrekt angezeigt.
- Die Navigationsleiste enthält die erwarteten Links.
- Der Hauptbanner enthält relevante Informationen.

Login-Test: Erfolgreich

- Das Login-Formular ermöglicht das Eingeben von Benutzername und Passwort.
- Benutzerdaten werden erfolgreich validiert.
- Der Login-Vorgang ist erfolgreich und eine Bestätigungsmeldung wird angezeigt.

Registrierungs-Test: Erfolgreich

- Das Registrierungsformular erlaubt das Ausfüllen der erforderlichen Benutzerdaten.
- Benutzerdaten werden erfolgreich validiert und gespeichert.
- Die Registrierung wird erfolgreich abgeschlossen und eine Bestätigungsmeldung wird angezeigt.

Suche nach Kursen-Test: Erfolgreich

- Die Suchfunktion ermöglicht die Eingabe von Suchkriterien.
- Die Suche liefert die erwarteten Suchergebnisse basierend auf den eingegebenen Kriterien.

Einschreibung in einen Kurs-Test: Erfolgreich

- Der Einschreibungsprozess ermöglicht die Auswahl eines Kurses.
- Der Benutzer wird erfolgreich zum Kurs hinzugefügt und die entsprechenden Informationen werden aktualisiert.

- Eine Bestätigungsmeldung wird angezeigt.

Alle Tests wurden erfolgreich abgeschlossen, was darauf hinweist, dass die wichtigsten Funktionen der Website ordnungsgemäß implementiert sind und wie erwartet funktionieren.

Zusammenfassung

Der Unit-Test für die BKBdemy Website bietet eine umfassende Überprüfung der wichtigsten Funktionen der Website. Durch die Ausführung dieser Tests können mögliche Fehler oder Probleme rechtzeitig erkannt und behoben werden. Die erfolgreiche Durchführung aller Tests gibt uns das Vertrauen, dass die Website gut funktioniert und den Anforderungen entspricht.

```php
<?php
include('./inc/index.php');<?php
class Route
{
    private static $routes = array();
    private static $pathNotFound = null;
    private static $methodNotAllowed = null;

    public static function add($expression, $function, $method = 'get')
    {
        array_push(self::$routes, array(
            'expression' => $expression,
            'function' => $function,
            'method' => $method
        ));
    }

    public static function pathNotFound($function)
    {
        self::$pathNotFound = $function;
    }

    public static function methodNotAllowed($function)
    {
        self::$methodNotAllowed = $function;
    }

    public static function run($basepath = '/')
    {

        $parsed_url = parse_url($_SERVER['REQUEST_URI']);
        if (isset($parsed_url['path'])) {
            $path = $parsed_url['path'];
        } else {
            $path = '/';
        }

        $method = $_SERVER['REQUEST_METHOD'];

        $path_match_found = false;

        $route_match_found = false;

        foreach (self::$routes as $route) {

            if ($basepath != '' && $basepath != '/') {
                $route['expression'] = '(' . $basepath . ')' . $route['expression'];
            }

            $route['expression'] = '^' . $route['expression'];

            $route['expression'] = $route['expression'] . '$';

            if (preg_match('#' . $route['expression'] . '#', $path, $matches)) {

                $path_match_found = true;

                if (strtolower($method) == strtolower($route['method'])) {

                    array_shift($matches);
```

```
                if ($basepath != '' && $basepath != '/') {
                    array_shift($matches);
                }

                call_user_func_array($route['function'], $matches);

                $route_match_found = true;

                break;
            }
        }
    }

    if (!$route_match_found) {

        if ($path_match_found) {
            header("HTTP/1.0 405 Method Not Allowed");
            if (self::$methodNotAllowed) {
                call_user_func_array(self::$methodNotAllowed, array($path, $method));
            }
        } else {
            http_response_code(404);
            if (self::$pathNotFound) {
                call_user_func_array(self::$pathNotFound, array($path));
            } else {
                echo "<h1>404 Not Found</h1>";
            }
        }
    }
  }
}<?php

function get_header()
{

   include('./pages/assets/header.php');
}

function get_footer()
{

   include('./pages/assets/footer.php');

}

function create_nav_bar($menuPosition, $links)
{

   ob_start(); ?>

   <ul class="menu-<?php echo $menuPosition ?>">
      <?php foreach ($links as $link):
         $urlCheck = true;
         if ($link['url'] === '') $urlCheck = false;
         if ($urlCheck) $url = 'href="' . get_home_url() . $link['url'] . '"'; else $url = '';

         ?>
         <li>
            <a class="<?php echo $link['class']; ?>"<?= $url; ?>">
```

```
            <?php echo $link['linkText']; ?>
            </a>
        </li>
    <?php endforeach; ?>
</ul>

<?php echo ob_get_clean();
}

function createVideoSlider($class, $courseData)
{
    ob_start(); ?>

    <div class="<?php echo $class; ?>-slider">
        <?php foreach ($courseData as $course): ?>
            <div class="single-course">
                <h3><?php echo $course['description']; ?></h3>
                <p class="course-further-info">
                    <span class="course-author-name">Tutor: <?php echo $course['author']; ?></span>
                    <span class="course-duration">Dauer: <?php echo $course['duration']; ?></span>
                    <span class="course-price">Preis: <?php echo $course['price']; ?></span>
                </p>
                <video>
                    <source src="<?php echo $course['video']; ?>" type="video/mp4">
                </video>
                <p class="course-excerpt"><?php echo $course['excerpt']; ?></p>
                <a class="secondary-button" href="<?php echo get_home_url() ?>/kurse/<?php echo $course['name']
?>">Zum
                    Kurs</a>
            </div>
        <?php endforeach; ?>
    </div>

    <?php echo ob_get_clean();
}

function get_template_uri()
{
    $doc_root = $_SERVER['DOCUMENT_ROOT'];
    $script_dir = __DIR__;
    $template_uri = str_replace($doc_root, '', $script_dir);
    $template_uri = str_replace('\\', '/', $template_uri);
    $template_uri = rtrim($template_uri, '/');
    $template_uri = str_replace('inc/functions', '', $template_uri);
    $template_uri = str_replace($doc_root, '', $template_uri);

    $protocol = 'http://';
    if (isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] == 'on') {
        $protocol = 'https://';
    }

    $template_uri = $protocol . $_SERVER['HTTP_HOST'] . $template_uri;

    return $template_uri;
}

function get_home_url()
{

    $template_uri = str_replace($_SERVER['DOCUMENT_ROOT'], '', realpath(dirname(__FILE__)));
```

```
    $template_uri = str_replace('\\', '/', $template_uri);
    $template_uri = rtrim($template_uri, '/');
    $template_uri = str_replace('inc/functions', '', $template_uri);
    $template_uri = rtrim($template_uri, '/');
    $protocol = (isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] === 'on') ? "https://" : "http://";
    $domain = $_SERVER['HTTP_HOST'];
    return $protocol . $domain . $template_uri;

}

function get_body_class()
{

    $doc_root = $_SERVER['DOCUMENT_ROOT'];
    $script_dir = __DIR__;
    $template_uri = str_replace($doc_root, '', $script_dir);
    $template_uri = str_replace('\\', '/', $template_uri);
    $template_uri = rtrim($template_uri, '/');
    $template_uri = str_replace('inc/functions', '', $template_uri);
    $template_uri = str_replace($doc_root, '', $template_uri);
    $template_uri = str_replace('/', '', $template_uri);
    $bodyClass = $_SERVER['REQUEST_URI'];
    $bodyClass = str_replace('/', '', $bodyClass);
    $bodyClass = str_replace($template_uri, '', $bodyClass);

    if ($bodyClass === "") return 'home';

    if (strpos($bodyClass, 'kurse') !== false) {
        return 'single page-single-course';
    }

    return 'page-' . $bodyClass;

}

function get_basepath()
{

    $basePath = rtrim(dirname($_SERVER['SCRIPT_NAME']), '/\\');
    if ($basePath == '/' || $basePath == '\\') {
        $basePath = '';
    }
    return $basePath;

}
<?php

/** functions* **/
include('inc/functions/functions.php');
/** classes **/
include('inc/classes/route.php');<?php

include('./functions.php');

Route::add('/',function(){
    include('./pages/frontpage.php');
});

Route::add('/anmeldung', function() {
    include('./pages/page-login.php');
```

```
});

Route::add('/abmeldung', function() {
    include('./pages/page-logout.php');
});

Route::add('/dashboard', function() {
    include('./pages/page-dashboard.php');
});

Route::add('/registrierung', function() {
    include('./pages/page-register.php');
});

Route::add('/registrierung-erfolgreich', function() {
    include('./pages/page-registration-success.php');
});

Route::add('/datenschutz', function() {
    include('./pages/page-datenschutz.php');
});

Route::add('/impressum', function() {
    include('./pages/page-impressum.php');
});

Route::add('/kurse', function() {
    include('./pages/page-courses.php');
});

Route::add('/kurse/.*$', function() {
    include('./pages/page-single-course.php');
});

Route::pathNotFound(function($path) {
    include('./pages/404.php');
});

Route::run(get_basepath());
exit;
<?php get_header(); ?>

<main id="this-is-sparta">
    <svg
            viewBox="0 0 541.17206 328.45184"
            height="328.45184"
            width="541.17206"
            id="svg2"
            version="1.1">
        <metadata
                id="metadata8">
        </metadata>
        <defs
                id="defs6">
            <pattern
                    patternUnits="userSpaceOnUse"
                    width="1.5"
                    height="1"
                    patternTransform="translate(0,0) scale(10,10)"
                    id="Strips2_1">
```

```
            <rect
                style="fill:black;stroke:none"
                x="0"
                y="-0.5"
                width="1"
                height="2"
                id="rect5419" />
        </pattern>
        <linearGradient
            osb:paint="solid"
            id="linearGradient6096">
            <stop
                id="stop6094"
                offset="0"
                style="stop-color:#1c1d1f;stop-opacity:1;" />
        </linearGradient>
    </defs>
    <g
        transform="translate(170.14515,0.038164)"
        id="layer1">
        <g
            id="g6219"
        >
            <path
                transform="matrix(1.0150687,0,0,11.193923,-1.3895945,-2685.7441)"

style="display:inline;fill:#1c1d1f;fill-opacity:1;stroke:#1c1d1f;stroke-width:0.1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opac
                d="m 145.0586,263.51309 c -90.20375,-0.0994 -119.20375,-0.0994 -119.20375,-0.0994"
                id="path6180" />
            <g
                id="g6174">
                <ellipse
                    ry="9.161705"
                    rx="9.3055239"
                    cy="91.32917"
                    cx="84.963676"
                    id="path4488"

style="display:inline;opacity:1;fill:none;fill-opacity:0.4627451;fill-rule:nonzero;stroke:#1c1d1f;stroke-width:1.08691013;stroke-miter
/>
                <path
                    id="path4490"
                    d="m 84.984382,-0.03816399 c 0.911733,-5.0186e-4 1.661858,18.47051499 1.674386,41.22988399
0.0069,12.610431 -0.214009,23.904598 -0.56753,31.469836 -0.282878,6.088471 -0.652275,9.761785
-1.058838,9.762119 -0.406564,3.33e-4 -0.78198,-3.672386 -1.074838,-9.760657 -0.36185,-7.564779
-0.595233,-18.858715 -0.602175,-31.469228 -0.01253,-22.759565 0.717262,-41.23145213 1.628995,-41.23195399 z"

style="display:inline;fill:#1c1d1f;stroke:none;stroke-width:0.23743393px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
                <path
                    id="path4496"
                    d="m 85.115421,100.5729 c -0.0036,3.37532 -0.0071,6.75165 -0.0107,10.12897 m 0.512159,0.18258
c -1.914603,-0.23621 -3.505591,1.17801 -4.861444,2.68113 -1.355853,1.50312 -2.473764,3.09173 -3.387866,4.59538
-0.914103,1.50365 -1.620209,2.91586 -2.416229,4.41952 -0.79602,1.50365 -1.67928,3.09352 -0.808656,3.24054
0.870624,0.14702 3.490408,-1.14815 5.700074,-1.91396 2.209666,-0.76581 4.001473,-1.00079 5.922125,-0.86765
1.920652,0.13314 3.947462,0.6325 6.245357,1.6195 2.297896,0.98701 4.861161,2.46015 4.9051,0.91309
0.04394,-1.54706 -2.430929,-6.11379 -4.787811,-9.33976 -2.356882,-3.22597 -4.596047,-5.11158 -6.51065,-5.34779
z"

style="display:inline;fill:#1c1d1f;fill-opacity:1;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacit
```

```
/>
		<rect
			ry="5"
			y="314.84082"
			x="35.355339"
			height="9.8994951"
			width="100.76272"
			id="rect4553"

style="display:inline;opacity:1;fill:#1c1d1f;fill-opacity:1;fill-rule:nonzero;stroke:#1c1d1f;stroke-width:1.00157475;stroke-miterlimit:4;
/>
		<path
			id="path4513"
			d="m 74.6875,125.03748 c -8.394789,7.68654 -16.790624,15.37405 -23.988969,22.38484
-7.198345,7.0108 -13.197555,13.3433 -18.781379,20.01048 -5.583823,6.66719 -10.749655,13.66605
-13.916608,18.7496 -3.166952,5.08355 -4.333432,8.24971 -4.750315,11.08369 -0.416883,2.83399 -0.08368,5.33304
1.809372,16.25302 1.893048,10.91998 5.343489,30.24673 9.760132,48.66349 4.416642,18.41676 9.798356,35.91675
15.180267,53.41738"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
		<path
			id="path4517"
			d="m 76.9375,124.66248 c -4.548745,6.50695 -9.29087,13.29053 -13.530749,18.69724
-4.239879,5.4067 -8.072459,9.57255 -11.572943,13.98975 -3.500484,4.41719 -6.66636,9.08269 -9.333429,13.99996
-2.66707,4.91727 -4.833205,10.08267 -6.333458,15.08327 -1.500252,5.0006 -2.33339,9.8328 -2.500149,14.33343
-0.166759,4.50062 0.333124,8.66631 1.249922,15.50064 0.916798,6.83434 2.249854,16.33237 3.499902,24.91604
1.250047,8.58368 2.416611,16.24967 4.583438,28.58394 2.166827,12.33427 5.333153,29.33244 8.499966,46.33323"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
		<path
			id="path4521"
			d="m 96.8125,126.22498 c 6.89586,6.45836 13.7917,12.9167 19.98957,19.14581 6.19786,6.22912
11.69789,12.22914 17.11456,18.39581 5.41666,6.16667 10.74996,12.49995 14.74993,17.91655 3.99997,5.41659
6.66659,9.91653 7.16671,17.83316 0.50012,7.91664 -1.16644,19.24921 -3.3502,31.24619 -2.18376,11.99698
-4.81616,24.33632 -8.42063,38.99809 -3.60448,14.66177 -8.06212,31.17154 -12.56244,47.83939"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
		<path
			id="path4525"
			d="m 91.9375,124.09998 c 5.854072,7.16655 11.70824,14.33322 16.21863,20.16651
4.51039,5.83328 7.67706,10.33329 11.92718,16.33346 4.25012,6.00017 9.58322,13.49984 12.66653,18.58299
3.08332,5.08314 3.91663,7.74974 4.68205,10.91384 0.76542,3.1641 1.40129,6.50242 1.69781,8.02406
0.29651,1.52165 0.22299,1.06579 0.14933,0.60912"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
		<path
			id="path4533"
			d="m 89,123.66248 c 6.159885,11.51771 12.31996,23.03577 16.83724,31.78904 4.51728,8.75327
7.29964,14.54985 9.24424,18.32123 1.9446,3.77138 3.00519,5.42118 4.1838,9.19262 1.17861,3.77144
2.47477,9.6631 1.94443,23.80647 -0.53034,14.14338 -2.88706,36.53226 -5.4209,56.44951 -2.53383,19.91725
-5.24428,37.35836 -7.95503,54.80146"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
		<path
			id="path4537"
```

```
                d="m 87.0625,123.03748 c 2.916637,10.42937 5.833458,20.8594 7.291964,26.66356
1.458505,5.80416 1.458505,6.98257 2.402021,11.11052 0.943517,4.12795 2.827535,11.19302 4.065005,16.02501
1.23748,4.832 1.82668,7.42447 2.12139,10.84263 0.29471,3.41815 0.29471,7.65958 -0.11785,20.44893
-0.41255,12.78934 -1.23731,34.11536 -2.18014,53.62015 -0.94282,19.50478 -2.003429,37.18159
-3.064154,54.86032"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
            <path
                id="path4541"
                d="m 85.206367,122.98266 c 0.117841,11.74369 0.235693,23.48835 0.235693,36.55072
-10e-7,13.06238 -0.117833,27.43796 -0.05891,45.3521 0.05892,17.91413 0.29461,39.36153 0.707091,58.80738
0.412482,19.44585 1.001711,36.88701 1.590999,54.32995"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
            <path
                id="path4545"
                d="m 83.12978,122.92016 c -2.601311,10.56131 -5.214983,21.17282 -7.40283,31.41665
-2.187847,10.24384 -3.955407,20.14218 -5.074975,26.03483 -1.119568,5.89264 -1.59092,7.77805
-1.885708,10.07706 -0.294789,2.29901 -0.412567,5.0079 5.1e-5,17.56339 0.412617,12.55548 1.355064,34.93859
2.474996,54.74239 1.119932,19.80379 2.415574,37.00049 3.712005,54.20767"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
            <path
                id="path4549"
                d="m 79.25478,124.23266 c -5.440192,11.56251 -10.880951,23.12622 -15.899657,33.56368
-5.018706,10.43747 -9.614414,19.74672 -11.912808,26.70033 -2.298394,6.95362 -2.298394,11.54922
-1.355419,24.57415 0.942974,13.02493 2.828182,34.46917 5.066095,53.84746 2.237913,19.37829
4.833109,36.71892 7.425959,54.04387"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
            <path
                id="path4556"
                d="m 42.426407,155.38825 c 3.4184,0.82513 6.836082,1.65009 10.606997,2.18034
3.770916,0.53024 7.89657,0.76599 11.608535,0.88382 3.711965,0.11782 7.012548,0.11782 10.429711,0.0589
3.417163,-0.0589 6.953769,-0.17681 10.606588,-0.23572 3.652818,-0.0589 7.425155,-0.0589 11.137027,-0.23569
3.711875,-0.17679 7.366225,-0.53043 10.724475,-0.70716 3.35826,-0.17672 6.4233,-0.17672 9.48702,-0.58922
3.06372,-0.41251 6.12885,-1.23774 9.1918,-2.06238"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
            <path
                id="path4560"
                d="m 13.113199,198.16821 c 47.547038,0.40361 95.093071,0.80721 142.638101,1.2108"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1.00614154px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
            <path
                id="path4529"
                d="m 132.6875,263.34998 c -4.2289,18.4155 -8.45806,36.83216 -12.6875,55.25"

style="display:inline;fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;"
/>
            <ellipse
                ry="4.6715717"
                rx="2.5"
                cy="238.08525"
```

```
                cx="119.12262"
                id="path4614"

style="display:inline;opacity:1;fill:#1c1d1f;fill-opacity:1;fill-rule:nonzero;stroke:#1c1d1f;stroke-width:1.00157475;stroke-miterlimit:4;
/>
        <ellipse
                ry="4.3158579"
                rx="4.9001703"
                cy="4.3948641"
                cx="85.016434"
                id="path4616"

style="display:inline;opacity:1;fill:#1c1d1f;fill-opacity:1;fill-rule:nonzero;stroke:#1c1d1f;stroke-width:0.82170224;stroke-miterlimit:4;
/>
        <ellipse
                transform="translate(-170.14515,-0.038164)"
                ry="3.880542"
                rx="3.5777507"
                cy="164.5713"
                cx="321.42224"
                id="path4565"

style="opacity:1;fill:#1c1d1f;fill-opacity:1;fill-rule:nonzero;stroke:#1c1d1f;stroke-width:1.00157475;stroke-miterlimit:4;stroke-dashar
/>
        <path
                transform="translate(-170.14515,-0.038164)"
                id="path4567"
                d="m 321.74355,168.0687 c -1e-5,3.3913 -3.42414,11.26702 -8.73834,11.26702 -5.3142,0
-18.59463,27.24606 -8.38477,3.759 1.35199,-3.11016 5.69513,-12.89881 10.50609,-15.15612 8.05545,-3.77965
6.61702,-3.26121 6.61702,0.1301 z"

style="opacity:1;fill:#1c1d1f;fill-opacity:1;fill-rule:nonzero;stroke:#1c1d1f;stroke-width:1.00157475;stroke-miterlimit:4;stroke-dashar
/>
        <path
                transform="translate(-170.14515,-0.038164)"
                id="path4570"
                d="m 325,163.45184 c 1.66722,0.62594 3.33388,1.25167 3.33438,1.56444 5e-4,0.31276
-1.66671,0.31276 -3.33438,0.31276"

style="fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;" />
        <path
                transform="translate(-170.14515,-0.038164)"
                id="path4578"
                d="m 314.72098,177.37003 c -0.21488,1.64138 -0.42965,3.28197 0.28484,3.96351 0.71449,0.68155
2.35396,0.39999 3.99418,0.1183"

style="fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;" />
        <path
                transform="translate(-170.14515,-0.038164)"
                id="path4578-1"
                d="m 316,176.45184 c -0.29612,1.41007 -0.59214,2.81967 -0.25801,3.48764 0.33413,0.66798
1.29605,0.59017 2.25801,0.51236"

style="fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;" />
        <path
                transform="translate(-170.14515,-0.038164)"
                id="path4610"
                d="m 318,180.45184 c 0.66667,0 1.33434,0 1.501,0.16616 0.16667,0.16617 -0.16667,0.49951
0.001,0.66667 0.16767,0.16717 0.68771,0.16717 0.89053,0.36949 0.20282,0.20233 -0.0582,0.46335
-0.39253,0.79768"
```

```
style="fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1;" />
                <path
                    id="path4573"
                    d="m 155,199.59998 34.15106,6.52318 v 11.49049 l -1.06066,13.43503 -3.88908,19.44543
-3.00521,10.42983 -4.06586,12.19759 -17.14734,-4.94975 -14.92431,-4.65869 v 0 L 155,199.59998"

style="fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1" />
                <path
                    id="path4575"
                    d="m 172.53405,202.94118 -2.65165,33.23402 -3.53553,16.97056 -5.12652,15.73313"

style="fill:none;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1" />
                <path
                    id="path4579"
                    d="m 187.2662,239.00256 c 0.76634,-0.82482 2.12163,-2.00333 3.50552,-2.26818 1.38389,-0.26485
2.79921,0.38383 3.2412,1.53192 0.442,1.14808 -0.0885,2.79852 -1.5624,3.24089 -1.4739,0.44236 -3.88809,-0.32312
-3.7995,0.001 0.0886,0.32427 2.68064,1.73812 4.00626,3.12221 1.32563,1.38408 1.38456,2.73956 0.79537,3.38822
-0.5892,0.64866 -1.82576,0.58977 -2.53349,0.11762 -0.70773,-0.47215 -0.88437,-1.35536 -1.59092,-2.65068
-0.70656,-1.29532 -1.94507,-3.00565 -2.47512,-4.09626 -0.53005,-1.09062 -0.35326,-1.56206 0.41308,-2.38689 z"

style="fill:#1c1d1f;fill-opacity:1;stroke:#1c1d1f;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1"
/>
            </g>
        </g>
    </g>
    <g
        id="layer3">
        <g
            id="text4526"
            style="fill:url(#Strips2_1);fill-opacity:1;stroke:none;stroke-width:1.23488784;"
            transform="matrix(0.97168718,0,0,1.0291378,170.14515,0.038164)"
            aria-label="4">
            <path
                id="path4555"
                style="fill:url(#Strips2_1);fill-opacity:1;stroke:#1c1d1f;stroke-width:1.23488784;stroke-opacity:1"
                d="M -0.46490841,256.59082 H -26.166013 v 43.5298 h -41.214384 v -43.5298 h -75.829833 l
-9.95629,-15.28174 64.136994,-140.0826 h 8.914347 l 33.573515,15.8606 -48.507941,89.60655 -11.461305,19.56526
h 39.130513 l 4.399288,-43.06672 h 36.815096 v 43.06672 h 25.70110459 z" />
        </g>
        <g
            id="text4526-2"
            style="fill:url(#Strips2_1);fill-opacity:1;stroke:none;stroke-width:1.23488784;"
            transform="matrix(0.97168718,0,0,1.0291378,377.95605,103.2934)"
            aria-label="4">
            <path
                id="path4558"
                style="fill:url(#Strips2_1);fill-opacity:1;stroke:#1c1d1f;stroke-width:1.23488784;stroke-opacity:1"
                d="m 147.55592,156.33602 h -25.70111 v 43.5298 H 80.640431 v -43.5298 H 4.8105946 L
-5.1456892,141.05429 58.991302,0.97168512 h 8.914347 L 101.47916,16.832277 52.971223,106.43883
41.50992,126.00409 h 39.130511 l 4.399288,-43.06672 h 36.815091 v 43.06672 h 25.70111 z" />
        </g>
    </g>
</svg>

<p id="errorText">O-o-oh! Da ist was schiefgelaufen.</p>
<a class="primary-button" href="/">Zurück</a>
</main>

<?php get_footer(); ?>
```

```
<footer>
    <div class="wrapper">
        <div class="footer-bar">
            <p>© <?php echo date('Y'); ?> BKBdemy</p>
            <div class="payment-options">
                <img src="<?= get_template_uri(); ?>/assets/images/icons/zahlung-versand.svg">
                <p>Alle Preise inkl. gesetzl. Mehrwertsteuer</p>
            </div>
            <?php create_nav_bar('footer', [
                [
                    'class' => '',
                    'url' => '/datenschutz',
                    'linkText' => 'Datenschutz'
                ],
                [
                    'class' => '',
                    'url' => '/impressum',
                    'linkText' => 'Impressum']
            ]); ?>
        </div>
    </div>
</footer>

</body>
</html><!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>BKBdemy</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
    <script src="<?php echo get_template_uri() ?>/dist/main.min.js"></script>
    <link rel="stylesheet" href="<?php echo get_template_uri(); ?>dist/main.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css">
    <link rel="icon" type="image/x-icon" href="<?php echo get_template_uri(); ?>/assets/images/icons/favicon.svg">
</head>
<body class="<?php echo get_body_class(); ?>">

<div class="scroll-up">
    <img src="<?php echo get_template_uri(); ?>/assets/images/icons/scroll-top-arrow.svg">
</div>

<?php if (get_body_class() === 'home'): ?>
    <div class="scroll-down active">
        <img src="<?php echo get_template_uri(); ?>/assets/images/icons/scroll-top-arrow.svg">
    </div>
<?php endif; ?>

<div class="scroll-up">
    <img src="<?php echo get_template_uri(); ?>/assets/images/icons/scroll-top-arrow.svg">
</div>
<div class="cookie-notice">
    <div class="cookie-header">
        <ul>
            <li class="acceptance active">Zustimmung</li>
            <li class="details">Details</li>
            <li class="about">Über Cookies</li>
        </ul>
```

```
    </div>
    <div class="content">
        <div class="single-content acceptance active">
            <p>Diese Webseite verwendet Cookies, um Ihre Benutzererfahrung zu verbessern und personalisierte Inhalte
                anzuzeigen.</p>
            <a href="/datenschutz">Mehr erfahren</a>
        </div>
        <div class="single-content details">
            <p>Die folgenden Cookies sind auf dieser Website erforderlich:</p>
            <ul>
                <li>Authentifizierungs-Cookies</li>
                <li>Sicherheits-Cookies</li>
            </ul>
            <p>Sie können Ihre Cookie-Einstellungen jederzeit ändern, indem Sie auf den entsprechenden Link in der
                Fußzeile klicken.</p>
        </div>
        <div class="single-content about">
            <p>Ein Cookie ist eine kleine Textdatei, die von einer Website auf Ihrem Computer oder Mobilgerät
                gespeichert wird. Cookies werden zur Speicherung von Informationen verwendet, um die Benutzererfahrung
                zu verbessern und bestimmte Funktionen bereitzustellen.</p>
        </div>
    </div>
    <div class="cookie-footer">
        <div id="accept-cookies" class="primary-button">Alle akzeptieren</div>
        <div id="deny-cookies" class="secondary-button">Ablehnen</div>
    </div>
</div>
<header>
    <div class="info">
        <div class="wrapper">
            <div class="info-item mail">
                <a href="mailto:info@bkbdemy.de"><img
                            src="<?= get_template_uri();
?>/assets/images/icons/mail-white.svg"><span>info@bkbdemy.de</span></a>
            </div>
            <div class="info-item phone">
                <img src="<?= get_template_uri(); ?>/assets/images/icons/phone-white.svg"><span> Telefonische
Unterstützung und Beratung unter: <a
                            href="tel:0251-123456">0251-123456</a></span>
            </div>
            <div class="info-item opening-times">Mo-Fr, 08.00 bis 18.00Uhr</div>
        </div>
    </div>
    <div class="wrapper">
        <nav>
            <a class="nav-item home-logo" href="<?php echo get_home_url(); ?>">
                <img src="<?php echo get_template_uri(); ?>/assets/images/logo/logo.png">
            </a>

            <div class="burger-menu">
                <div class="stripe stripe-top"></div>
                <div class="stripe stripe-mid"></div>
                <div class="stripe stripe-bottom"></div>
            </div>

            <div class="nav-bar desktop">

                <div class="user-logged-out user-container">
                    <?php create_nav_bar('desktop', [
                        [
```

```
                'class' => 'cookie-settings',
                'url' => '',
                'linkText' => 'Cookie Einstellungen'
            ],
            [
                'class' => 'primary-button',
                'url' => '/anmeldung',
                'linkText' => 'Anmelden'
            ],
            [
                'class' => 'secondary-button',
                'url' => '/registrierung',
                'linkText' => 'Registrieren']
        ]); ?>
    </div>

    <div class="user-logged-in user-container">
        <?php create_nav_bar('desktop', [
            [
                'class' => 'cookie-settings',
                'url' => '',
                'linkText' => 'Cookie Einstellungen'
            ],
            [
                'class' => 'primary-button',
                'url' => '/dashboard',
                'linkText' => 'Dashboard'
            ],
            [
                'class' => 'secondary-button',
                'url' => '/abmeldung',
                'linkText' => 'Abmelden']
        ]); ?>
    </div>

</div>


<div class="nav-bar mobile">
    <div class="user-logged-out user-container">
        <?php create_nav_bar('mobile', [
            [
                'class' => 'mobile-button',
                'url' => '/anmeldung',
                'linkText' => 'Anmelden'
            ],
            [
                'class' => 'mobile-button',
                'url' => '/registrierung',
                'linkText' => 'Registrieren'],
            [
                'class' => 'cookie-settings',
                'url' => '#cookie-settings',
                'linkText' => 'Cookie Einstellungen'
            ]
        ]); ?>
    </div>
    <div class="user-logged-in user-container">
        <?php create_nav_bar('mobile', [
            [
```

```
                    'class' => 'mobile-button',
                    'url' => '/dashboard',
                    'linkText' => 'Dashboard'
                ],
                [
                    'class' => 'mobile-button',
                    'url' => '/abmeldung',
                    'linkText' => 'Abmelden'],
                [
                    'class' => 'cookie-settings',
                    'url' => '#cookie-settings',
                    'linkText' => 'Cookie Einstellungen'
                ]
            ]); ?>
        </div>
    </div>
  </nav>
 </div>
</header>
<div class="wrapper">
 <h2 class="section-headline section-headline-frontpage">Teilnehmer sehen sich gerade an</h2>
 <p>Diese Kurse sind momentan besonders beliebt !!!</p>
 <div class="most-watched-slider"></div>
 <a class="primary-button shop-button" href="<?= get_home_url(); ?>/kurse">zum Shop</a>
</div><div class="image-filter"></div>
<div class="frontpage-slider">
 <img src="<?= get_template_uri(); ?>/assets/images/background/IMG_0126.JPG">
 <img src="<?= get_template_uri(); ?>/assets/images/background/IMG_0127.JPG">
 <img src="<?= get_template_uri(); ?>/assets/images/background/IMG_0128.JPG">
 <img src="<?= get_template_uri(); ?>/assets/images/background/IMG_0129.JPG">
 <img src="<?= get_template_uri(); ?>/assets/images/background/IMG_0130.JPG">
 <img src="<?= get_template_uri(); ?>/assets/images/background/IMG_0131.JPG">
 <img src="<?= get_template_uri(); ?>/assets/images/background/IMG_0134.JPEG">
 <img src="<?= get_template_uri(); ?>/assets/images/background/header-image-1.jpg">
 <img src="<?= get_template_uri(); ?>/assets/images/background/header-image-2.jpg">
 <img src="<?= get_template_uri(); ?>/assets/images/background/header-image-3.jpg">
</div><div class="wrapper">
 <div class="news-wrapper">
  <h2 class="section-headline-frontpage">News</h2>

  <div class="news-slider">
   <article class="single-news">
    <img class="news-image" src="<?= get_template_uri(); ?>/assets/images/news/level-system.jpg">
    <div class="content">
     <h3 class="news-title">Unser neues Level System</h3>
     <p class="news-date">29-05-2023 15:47 Uhr</p>
     <p>Registrierte Nutzer haben jetzt die Möglichkeit, in unserem Levelsystem aufzusteigen. Jedes
      abgeschlossene Video belohnt sie mit einer festgelegten Anzahl von Punkten. Zu Beginn haben sie
      den
      ITA-Status. Ab 999 Punkten steigen sie ins erste Ausbildungsjahr auf, ab 3000 Punkten ins zweite
      Ausbildungsjahr, ab 5000 Punkten ins dritte Ausbildungsjahr und bei 10000 Punkten erreichen sie
      die
      Höchststufe als Anwendungsentwickler!</p>
    </div>
   </article>

   <article class="single-news">
    <img class="news-image" src="<?= get_template_uri(); ?>/assets/images/news/python.png">
    <div class="content">
     <h3 class="news-title">Python lernen leicht gemacht</h3>
```

```
            <p class="news-date">31-05-2023 11:55 Uhr</p>
            <p>Unser Python-Kurs für Anfänger ermöglicht es Ihnen, Python im Handumdrehen zu erlernen. In nur 10
              einfachen und gut strukturierten Lektionen werden Sie Schritt für Schritt zum echten Experten.
              Tauchen
              Sie ein und entdecken Sie die vielseitigen Möglichkeiten der Programmiersprache Python!
              Neugierig? <a
                href="<?= get_home_url(); ?>/kurse/2">Hier</a> gehts zum Kurs!</p>
          </div>
        </article>

        <article class="single-news">
          <img class="news-image" src="<?= get_template_uri(); ?>/assets/images/news/go-live.jpg">
          <div class="content">
            <h3 class="news-title">Wir gehen Live!</h3>
            <p class="news-date">22-05-2023 11:23 Uhr</p>
            <p>Die BKBdemy hat ihre Tore endlich geöffnet, nach einem langen Weg von der Planung und
Entwicklung
              bis hin
              zur Fertigstellung eines voll funktionsfähigen Lernportals. Ab sofort können Lerninteressierte
              und
              Enthusiasten aus der ganzen Welt von unserer vielfältigen Auswahl an Kursen profitieren und ihre
              Fähigkeiten in verschiedenen Bereichen weiterentwickeln. <a href="<?= get_home_url();
?>/kurse">Hier</a>
              finden Sie unser Kursangebot.</p>
          </div>
        </article>

        <article class="single-news">
          <img class="news-image" src="<?= get_template_uri(); ?>/assets/images/news/projekt.png">
          <div class="content">
            <h3 class="news-title">BKBdemy Dokumentation</h3>
            <p class="news-date">18-05-2023 09:33 Uhr</p>
            <p>Nach langem überlegen haben wir, die Entwickler der BKBdemy Academy uns dazu entschieden, die
              gesamte
              Projekt Dokumentation des Projekts für Sie bereitzustellen, damit auch andere Entwickler von
              unseren
              Erfahrungen der Planung und Vorbereitung, bis hin zur technischen Umsetzung profitieren können.
              Deswegen
              stellen wir unsere Porjekt Dokumentation für sie als <a
                href="<?= get_template_uri(); ?>/assets/uploads/BKBdemy-Dokumentation.pdf" target="_blank">PDF
                Download</a> bereit</p>
          </div>
        </article>

        <article class="single-news">
          <img class="news-image" src="<?= get_template_uri(); ?>/assets/images/news/vorstellung.jpg">
          <div class="content">
            <h3 class="news-title">BKBdemy Projekt Vorstellung</h3>
            <p class="news-date">15-05-2023 17:21 Uhr</p>
            <p>Leute aufgepasst! Am 07. Juni 2023 Stellen die Anwendungsentwickler:innen und
              Systemintegrator:innen des
              Berufskollegs in Beckum Ihre Projekte vor. Nach ca. 3 Monaten voller harter Arbeit, Tränen und
              Schweiß,
              haben die Schüler:innen nun die Chance, ihr
              können unter Beweis zu stellen und das Lehrpersonal bestehend aus Carsten Hüther, Verena Witt und
              Uwe
              Richert von ihren Fähigkeiten zu überzeugen.</p>
          </div>
        </article>
      </div>
```

```
    </div>
</div>
<div class="wrapper">
    <div class="content-container">
        <h3>Tauchen Sie ein in unsere Kursauswahl!</h3>
        <p>Entdecken Sie die faszinierende Welt der Programmierung bei bkbdemy und lassen Sie sich von unseren
            hochwertigen Tutorials begeistern. Egal, ob Sie Anfänger oder erfahrener Entwickler sind, bei uns
            finden Sie den richtigen Kurs, um Ihre Fähigkeiten auf das nächste Level zu bringen. Tauchen Sie ein und
            entfesseln Sie Ihr Programmierpotenzial!</p>
        <a class="primary-button" href="<?= get_home_url() ?>/kurse">zum Shop</a>
    </div>
</div>
<div class="wrapper">
    <div class="registration-teaser user-container user-logged-out">
        <h3>Registriere Sie sich und genießen Sie die Vorteile der BKBdemy Academy</h3>
        <p>Willkommen in der BKBdemy Academy! Hier finden Sie eine breite Palette an spannenden Kursen, die von
            erfahrenen Experten entwickelt wurden und Ihnen dabei helfen, neue Fähigkeiten zu erlernen und Ihr
            Wissen zu erweitern. Egal, ob Sie sich beruflich weiterentwickeln oder einfach nur Ihre persönlichen
            Interessen vertiefen möchtest, unsere Kurse bieten Ihnen das Wissen und die Werkzeuge, die Sie
            brauchen, um Ihre Ziele zu erreichen. Registrieren Sie sich jetzt und tauchen Sie ein, in die Welt der BKBdemy
            Academy! </p>
        <a class="secondary-button" href="<?php echo get_home_url(); ?>/registrierung">Registrieren</a>
    </div>

    <div class="registration-teaser user-container user-logged-in">
        <h3 id="welcome-user"></h3>
        <p>Willkommen in der BKBdemy Academy! Hier finden Sie eine breite Palette an spannenden Kursen, die von
            erfahrenen Experten entwickelt wurden und Ihnen dabei helfen, neue Fähigkeiten zu erlernen und Ihr
            Wissen zu erweitern. Egal, ob Sie sich beruflich weiterentwickeln oder einfach nur Ihre persönlichen
            Interessen vertiefen möchtest, unsere Kurse bieten Ihnen das Wissen und die Werkzeuge, die Sie
            brauchen, um Ihre Ziele zu erreichen.</p>
        <a class="secondary-button" href="<?php echo get_home_url(); ?>/dashboard">Dein Bereich</a>
    </div>

    <div class="new-courses-teaser">
        <h1 class="course-headline section-headline">Unsere neuesten Kurse!</h1>
        <div class="course-slider"></div>
    </div>
</div><?php get_header(); ?>

<main>

    <section id="frontpage-slider">
        <?php
            /* including the content for the frontage image slider*/
            include_once('content/frontpage-slider-content.php'); ?>
    </section>

    <section id="teaser">
        <?php
            /* including the content for the frontage teaser content*/
            include_once('content/teaser-content.php'); ?>
    </section>

    <section id="shop-teaser">
        <?php
            /* including the content for the shop teaser content*/
            include_once('content/shop-teaser-content.php'); ?>
    </section>
```

```
    <section id="current-most-watched">
        <?php
            /* including the content for the current most watched content*/
            include_once('content/current-most-watched-content.php'); ?>
    </section>

    <section id="news">
        <?php
            /* including content for the news content*/
            include_once('content/news-content.php'); ?>
    </section>

</main>

<?php get_footer(); ?>
<?php get_header(); ?>

    <main>
        <div class="wrapper">
            <h1 class="page-headline">Shop</h1>

            <div id="search-filter">
                <div class="title-filter">
                    <img src="<?= get_template_uri(); ?>/assets/images/icons/icons8-suche.svg">
                    <input type="text" placeholder="Suche nach Titeln">
                </div>
                <div class="difficulty-filter">
                    <div class="dropown-title"><span>Schwierigkeit</span><img
                            src="<?= get_template_uri(); ?>/assets/images/icons/icons8-pfeil%20-ausklappen-30.png">
                    </div>
                    <div class="dropdown">
                        <div>
                            <label for="difficulty-1">Anfänger</label>
                            <input type="checkbox" name="difficulty-1">
                        </div>
                        <div>
                            <label for="difficulty-2">Fortgeschritten</label>
                            <input type="checkbox" name="difficulty-2">
                        </div>
                        <div>
                            <label for="difficulty-2">Experte</label>
                            <input type="checkbox" name="difficulty-3">
                        </div>

                    </div>
                </div>
            </div>
            <div id="results"></div>
        </div>
    </main>

<?php get_footer(); ?><?php get_header(); ?>

    <main>
        <div class="wrapper">
            <div class="user-container user-logged-in">
                <div>
                    <h1 class="page-headline">Dashboard</h1>
                    <div class="dashboard-content-wrapper">
```

```
                <nav>
                    <ul>
                        <li class="active" data-src="#my-account">Mein Account</li>
                        <li data-src="#my-courses">Meine Kurse</li>
                        <li data-src="#my-currency">Mein Kontostand</li>
                    </ul>
                </nav>
                <div class="dashboard-container content-container active" id="my-account">
                    <p>
                        Hallo <span id="username"></span>
                        <span id="user-level"></span>
                    </p>
                    <p>Willkommen in Ihrem persönlichen Dashboard. Hier können Sie Ihren Kontostand sowie Ihre
                        Kurse verwalten. </ br>
                        Viel Spaß wünscht Ihnen Ihr BKBdemy Team!
                    </p>
                </div>
                <div class="dashboard-container" id="my-courses">
                    <div class="further-courses">
                        <a class="secondary-button" href="<?= get_home_url(); ?>/kurse">zum Shop</a>
                    </div>
                    <div id="owned-courses"></div>
                </div>
                <div class="dashboard-container content-container" id="my-currency">
                    <p>
                        Ihr aktueller Kontostand beträgt <span id="current-balance"></span>
                    </p>
                    <p>Da wir Sie unglaublich gern haben, können Sie hier Ihren Kontostand erhöhen.</p>
                    <div class="increase-balance">
                        <div class="user-input">
                            <img src="<?= get_template_uri(); ?>/assets/images/icons/icons8-muenzen-50.png">
                            <input type="number" min="1" name="userBalance">
                        </div>
                        <img id="increase-balance-button"
                             src="<?= get_template_uri(); ?>/assets/images/icons/icons8-hinzufuegen-50.png">
                    </div>
                    <div class="response-message">
                        <p></p>
                    </div>
                </div>
            </div>
        </div>
        <div class="user-container user-logged-out">
            <p id="log-in-message">
                Bitte <a href="<?= get_home_url(); ?>/registrierung">erstellen Sie einen Account</a> oder <a
                    href="<?= get_home_url(); ?>/anmeldung">melden</a> Sie sich an.
            </p>
        </div>
    </div>
</main>

<?php get_footer(); ?><?php get_header(); ?>

<main>
    <div class="wrapper">
        <h1 class="page-headline">Datenschutz</h1>
        <p>Wir, die Betreiber von "Bkbdemy", nehmen den Schutz Ihrer personenbezogenen Daten sehr ernst und
            behandeln diese vertraulich und gemäß den gesetzlichen Datenschutzvorschriften sowie dieser
            Datenschutzerklärung.</p>
```

<h2>1. Verantwortliche Stelle</h2>
Verantwortliche Stelle für die Verarbeitung Ihrer personenbezogenen Daten im Rahmen der Nutzung unserer
Webseite ist:

Bkbdemy
Der Erschaffer
Hansaring 11
59629 Beckum
Deutschland
E-Mail: info@bkbdemy.de

<h2>2. Datenverarbeitung auf unserer Webseite</h2>
Wir erheben und verarbeiten personenbezogene Daten, wenn Sie unsere Webseite besuchen oder nutzen. Dies
kann technisch notwendige Daten, wie beispielsweise Ihre IP-Adresse, aber auch von Ihnen freiwillig bereitgestellte Daten, wie beispielsweise Name und E-Mail-Adresse, umfassen.
<h3>a) Logfiles</h3>
Bei jedem Zugriff auf unsere Webseite werden bestimmte technische Informationen automatisch erfasst und in Logfiles gespeichert. Zu diesen Informationen können gehören:

IP-Adresse des zugreifenden Endgeräts
Datum und Uhrzeit des Zugriffs
Name und URL der abgerufenen Datei
Übertragene Datenmenge
Meldung über erfolgreichen Abruf (HTTP response code)
Browser und Betriebssystem des zugreifenden Endgeräts
Webseite, von der aus der Zugriff erfolgt (Referrer URL)
Name Ihres Internet Service Providers

Diese Informationen werden für Zwecke der technischen Administration der Webseite und zur Gewährleistung
der Sicherheit unserer Systeme erhoben und verarbeitet.
<h3>b) Kontaktformular</h3>

Wenn Sie über das auf unserer Webseite bereitgestellte Kontaktformular mit uns in Kontakt treten, werden die von Ihnen in das Kontaktformular eingegebenen Daten an uns übermittelt und gespeichert. Diese Daten umfassen in der Regel Ihren Namen, Ihre E-Mail-Adresse sowie den Inhalt Ihrer Anfrage. Diese Daten werden ausschließlich zur Bearbeitung Ihrer Anfrage genutzt und nach Abschluss der Bearbeitung gelöscht, sofern keine gesetzlichen Aufbewahrungspflichten bestehen.

<h3>c) Cookies</h3>
Unsere Webseite verwendet Cookies. Cookies sind kleine Textdateien, die von Ihrem Browser auf Ihrem Endgerät
gespeichert werden und bestimmte Informationen enthalten. Bei Ihrem nächsten Besuch unserer Webseite mit
demselben Endgerät werden die in Cookies gespeicherten Informationen an unsere Webseite oder eine andere
Webseite, die das Cookie erkennt, zurückgesandt. Durch Cookies wird unsere Webseite nutzerfreundlicher und effektiver. Einige der von uns verwendeten Cookies werden nach Ende der Browser-Sitzung, also nach Schließen Ihres Browsers, wieder gelöscht (sog. Sitzungs-Cookies). Andere Cookies verbleiben auf Ihrem Endgerät und ermöglichen uns, Ihren Browser beim nächsten Besuch wiederzuerkennen (persistente Cookies).
Wenn Cookies gesetzt werden, erheben und verarbeiten diese im Regelfall bestimmte Nutzerinformationen wie Browser- und Standortdaten sowie IP-Adresswerte auf individueller Basis. Soweit wir durch Cookies personenbezogene Daten von Ihnen verarbeiten, erfolgt die Verarbeitung gemäß Art. 6 Abs. 1 lit. f DSGVO zur Wahrung unserer berechtigten Interessen an der bestmöglichen Funktionalität der Webseite sowie einer kundenfreundlichen und effektiven Ausgestaltung des Seitenbesuchs.

```
        <p>Sie können die Speicherung von Cookies in den Einstellungen Ihres Browsers jederzeit verhindern oder bereits gespeicherte Cookies löschen. Bitte beachten Sie, dass in diesem Fall unsere Webseite möglicherweise nicht mehr in vollem Umfang genutzt werden kann.</p>

        <h2>3. Weitergabe von Daten an Dritte</h2>

        <p>Eine Weitergabe Ihrer personenbezogenen Daten an Dritte erfolgt grundsätzlich nicht, es sei denn, dies ist gesetzlich vorgeschrieben oder erforderlich, um unsere vertraglichen Pflichten Ihnen gegenüber zu erfüllen. In diesem Fall werden wir Sie im Vorfeld darüber informieren und gegebenenfalls Ihre Einwilligung einholen.</p>

        <h2>4. Ihre Rechte als Nutzer</h2>

        <p>Sie haben das Recht, von uns Auskunft über die zu Ihrer Person gespeicherten Daten zu verlangen sowie deren Berichtigung, Löschung oder Sperrung zu verlangen, sofern dies gesetzlich zulässig ist. Ferner haben Sie das Recht, jederzeit der Verarbeitung Ihrer personenbezogenen Daten durch uns zu widersprechen.</p>
      </div>
    </main>

<?php get_footer(); ?><?php get_header(); ?>

    <main>
      <div class="wrapper">
        <h1 class="page-headline">Impressum</h1>

        <h2>Bkbdemy GmbH</h2>
        <p>Der Erschaffer<br>
          Hansaring 11<br>
          59629 Beckum<br>
          Deutschland</p>

        <h3>Vertreten durch:</h3>
        <p>Carsten Hüther (Geschäftsführer)<br>
          Verena Witt (Geschäftsführerin)</p>

        <h3>Handelsregisternummer:</h3>
        <p>HRB 123456</p>

        <h3>Registergericht:</h3>
        <p>Amtsgericht Münster</p>

        <h3>Umsatzsteuer-Identifikationsnummer gemäß § 27 a Umsatzsteuergesetz:</h3>
        <p>DE123456789</p>

        <h3>Zuständige Aufsichtsbehörde:</h3>
        <p>Regierungspräsidium Münster<br>
          Albrecht-Thaer-Straße 9<br>
          48147 Münster<br>
          Deutschland</p>

        <h3>Zuständige Kammer:</h3>
        <p>Industrie- und Handelskammer Münster<br>
          Sentmaringer Weg 61<br>
          48151 Münster<br>
          Deutschland</p>

        <h3>Berufsrechtliche Regelungen:</h3>
        <ul>
          <li>Berufsordnung für Bildungseinrichtungen in Nordrhein-Westfalen (BOB)</li>
```

```
            <li>Weiterbildungsgesetz des Landes Nordrhein-Westfalen (WbG NRW)</li>
        </ul>

        <h3>Datenschutzbeauftragter:</h3>
        <p>Uwe Richert<br>
            E-Mail: datenschutz@bkbdemy.de</p>

        <h3>Kontakt:</h3>
        <p>Telefon: +49 (0) 123 456789<br>
            E-Mail: info@bkbdemy.de</p>

    </div>
</main>

<?php get_footer(); ?><?php get_header(); ?>

<main>
    <div class="wrapper">
        <h1 class="page-headline">Anmeldeformular</h1>
        <div class="login-form form content-container">
            <p>Geben Sie Ihre Anmeldedaten ein</p>
            <input id="user-username" type="text" name="username" placeholder="Benutzername">
            <div class="password-wrapper">
                <input id="user-password" type="password" name="password" placeholder="Passwort">
                <span class="password-type-toggle"></span>
            </div>
            <div class="secondary-button" id="user-login-button" type="submit">Anmelden</div>
            <div class="no-account-yet">
                <p>Noch keinen Account? <a href="<?= get_home_url() ?>/registrierung">Registriere</a> dich jetzt!</p>
            </div>
        </div>
        <div class="error-message"></div>
    </div>
</main>

<?php get_footer(); ?><?php get_header(); ?>

<main>
    <div class="wrapper">
        <h1 class="page-headline">Abmelden</h1>
        <div class="registration-form form content-container">
            <p>Wollen Sie sich wirklich abmelden?</p>
            <div class="secondary-button" href="<?= get_home_url(); ?>" id="user-logout-button"
type="submit">Abmelden</div>
        </div>
    </div>
</main>

<?php get_footer(); ?><?php get_header(); ?>

<main>
    <div class="wrapper">
        <h1 class="page-headline">Registrierung</h1>
        <div class="registration-form form content-container">
            <p>Melde dich jetzt an und genieße BKBdemy in vollem Umfang.</p>
            <p>Ihr Passwort:</p>
            <ul>
                <li>Muss mindestens 8 Zeichen lang sein</li>
                <li>Muss mindestens einen Großbuchstaben enthalten [A-Z]</li>
                <li>Muss mindestens einen Kleinbuchstaben enthalten [a-z]</li>
```

```
                <li>Muss mindestens eine Zahl enthalten [1,2,3,4,5,6,7,8,9,0]</li>
                <li>Muss mindestens ein Sonderzeichen enthalten [@,#,$,%,^,&,+,=,?,!]</li>
                <li>Darf keine Leerzeichen enthalten</li>
            </ul>
            <input id="user-reg-username" type="text" name="username" placeholder="Benutzername">
            <div class="password-wrapper">
                <input id="user-reg-password" type="password" name="password" placeholder="Passwort">
                <span class="password-type-toggle"></span>
            </div>
            <div class="secondary-button" id="user-registration-button" type="submit">Registrieren</div>
        </div>
        <p class="error-message"></p>
    </div>
</main>

<?php get_footer(); ?><?php get_header(); ?>

<main>
    <div class="wrapper user-container user-logged-in">
        <div>
            <h1 class="page-headline">Willkommen <span id="registration-success-username"></span></h1>
            <div class="registration-form">
                <p>Herzlichen Glückwunsch.</p>
                <p>Sie sind nun ein Teil der BKBdemy.</p>
                <p><a class="secondary-button" href="<?= get_home_url(); ?>/dashboard">Hier</a> gehts zum
Dashboard</p>
            </div>
        </div>
    </div>

    <div class="wrapper user-container user-logged-out">
        <div>
            <h1 class="page-headline">Ups. Da ist etwas schiefgelaufen.</h1>
            <div class="registration-form">
                <p>Es tut uns leid.</p>
                <p>Bitte versuche es erneut</p>
                <p><a class="secondary-button" href="<?= get_home_url(); ?>/registrierung">Hier</a> gehts zur
Registrierung.</p>
            </div>
        </div>
    </div>
</main>

<?php get_footer(); ?><?php get_header(); ?>

<main>
    <div class="wrapper">
        <div class="single-course"></div>

        <div class="error-message">
            <p>
                Leider reicht Ihr Guthaben nicht auf. Geheh sie bitte zum <a href="<?= get_home_url();
?>/dashboard">Dashboard</a>
                um Ihr Guthaben aufzuladen.
            </p>
        </div>

        <div class="comment-container">
            <div class="container read-comments"></div>
        </div>
```

```
    </div>
</main>

<?php get_footer(); ?>

// JS
require('/src/js/init.js');
require('/src/js/slickSlider.js');
// Sass
require('/src/sass/index.sass');/* Defining Global Variables or Objects*/

/* This variable stores the current videos of the course when its initialized*/
let currentCourse = null;
/* This variable checks wether the token has already been verified once within the session or not*/
let verifyToken = true;
/* Caching the Video Data of a Course for reusing purposes, using a boolean to reset the cache when mandatory */
let videoProgressCache = {
    videos: {},
    invalidated: true
}


/* These functions will initialize when the document (DOM) is ready */
jQuery(document).ready(function () {
    /* Menu Navigation*/
    initBurgerMenu();
    initScroll();
    initFrontpageSlider();
    initNewsSlider();

    /*Initialize User Status*/
    initCookieNotice();
    initLogin();
    initLogout();
    initRegistration();

    /*Initialize Page Content*/
    initIncreaseUserBalance();
    initVisiblePassword();
    initHandleUserContainer();
    initLoadUserData();
    initLoadNewProducts();
    initLoadAllProducts();
    initLoadMostWatchedProducts();
    initLoadSingleProduct();
    initLoadOwnedProducts();
    initDashboardNavigation();
    initLoadCurrentUserLevel();
})

/* callback functions */
async function addComment(courseID) {
    const input = jQuery('.write-comment textarea');
    const loginInput = {
        comment: input.val()
    }

    fetch('https://bkbdemy.pxroute.net/api/products/' + courseID + '/comments', {
        method: 'POST',
```

```
        headers: new Headers({
            'Authorization': 'Bearer ' + await getToken(),
            'Content-Type': 'application/json'
        }),
        body: JSON.stringify(loginInput)
    })
        .then(response => response.json())
        .then(() => {
            loadCourseComments();
            input.val("");
            const button = jQuery('.submit-comment');
            /* Removing the class that is added when clicking the button*/
            button.removeClass('disabled');
        })
        .catch(error => {
            console.error('Error:', error)
        });
}

function loadCourseComments() {

    const URL = window.location.href;
    const courseID = parseInt(URL.split('/kurse/')[1]);
    const bodyClass = jQuery('body').hasClass('page-single-course');

    if (bodyClass) {
        fetch('https://bkbdemy.pxroute.net/api/products/' + courseID + '/comments', {
            method: 'GET',
        })
            .then(response => response.json())
            .then(data => {
                renderComments(data, 3);
            })
            .catch(error => {
                console.error('Error:', error);
            });
    }
}

async function addBalance(amount) {

    fetch('https://bkbdemy.pxroute.net/api/auth/increase_balance/' + parseInt(amount), {
        method: 'POST',
        headers: new Headers({
            'Authorization': 'Bearer ' + await getToken(),
            'Content-Type': 'application/json'
        })
    })
        .then(response => response.json())
        .catch(error => {
            console.error('Error:', error);
        });
}

function courseVideoSlider() {
    const container = jQuery('.course-progress-wrapper .bottom');
    jQuery(container).slick({
        arrows: true,
        dots: false,
        infinite: true,
```

```
        autoplay: false,
        adaptiveHeight: false,
        slidesToShow: 1,
    });
}

async function getCourseProgress(videos) {
    let CourseIDs = [];
    videos.forEach(function (video) {
        CourseIDs.push(video.IndexID);
    });

    const watchedVideos = await getWatchedVideos();

    let watched = 0;

    if (watchedVideos) {
        watchedVideos.forEach(function (video) {
            if (CourseIDs.includes(video.IndexID)) {
                watched++;
            }
        });

        const length = videos.length;
        const courseProgress = watched / length * 100;

        return courseProgress.toFixed(2);
    } else {
        return 0;
    }
}

function addToWatchedVideos() {
    const videos = jQuery('.course-progress-wrapper video');

    if (videos.length) {
        videos.each(function () {
            const video = $(this);
            const videoId = video.attr('id');
            const ID = videoId.split('video-')[1];
            const points = video.attr('data-src');
            video.on('ended', async function () {
                videoProgressCache.invalidated = true;
                await updateVideoProgress(ID, points);
                try {
                    fetch('https://bkbdemy.pxroute.net/api/video/' + parseInt(ID) + '/progress', {
                        method: 'POST',
                        headers: new Headers({
                            'Authorization': 'Bearer ' + await getToken(),
                            'Content-Type': 'application/json'
                        })
                    })
                        .then(response => response)
                        .then(async () => {
                            await updateCourseProgress(currentCourse);
                        })
                } catch (error) {
                    console.error('Error:', error);
                }
            });
```

```
        });
    }
}

function returnDifficulty(integer) {
    if (integer === 1) return 'Anfänger';
    if (integer === 2) return 'Fortgeschritten';
    if (integer === 3) return 'Experte';
}

function getHomeUrl() {
    const homeURL = window.location.origin;
    return homeURL;
}

function animateErrorMessage() {

    const container = jQuery('.error-message');
    container.addClass('animate');
    container.addClass('active');

    setTimeout(function () {
        container.removeClass('animate');
    }, 1000)
}

async function getToken(verifyToken) {
    const token = localStorage.getItem('authToken');
    try {

        if (!verifyToken) {
            return token;
        }

        const response = await fetch('https://bkbdemy.pxroute.net/api/auth/me', {
            method: 'GET',
            headers: new Headers({
                'Authorization': 'Bearer ' + token,
                'Content-Type': 'application/json'
            })
        });
        //const data = await response.json();
        if (response.status === 401) return null;
        verifyToken = true;
        return token;
    } catch (error) {
        console.error('Error:', error);
        return null;
    }
}

async function getUserData() {
    if (await getToken()) {
        return fetch('https://bkbdemy.pxroute.net/api/auth/me', {
            method: 'GET',
            headers: new Headers({
                'Authorization': 'Bearer ' + await getToken(),
                'Content-Type': 'application/json'
            })
        })
```

```
            .then(response => response.json());
    }
    return null;
}

async function getWatchedVideos() {
    const token = localStorage.getItem('authToken');

    try {

        if (!videoProgressCache.invalidated) {
            return videoProgressCache.videos;
        }

        const response = await fetch('https://bkbdemy.pxroute.net/api/video/watched', {
            method: 'GET',
            headers: new Headers({
                'Authorization': 'Bearer ' + token,
                'Content-Type': 'application/json'
            })
        });

        if (response.ok) {
            const data = await response.json();
            videoProgressCache.videos = data;
            videoProgressCache.invalidated = false;
            return videoProgressCache.videos;
        } else {
            throw new Error('Failed to fetch watched videos');
        }
    } catch (error) {
        console.error('Error:', error);
        throw error;
    }
}

async function checkSingleVideoProgress(videoID) {
    const watchedVideos = await getWatchedVideos();
    let watchedVideoIDs = [];

    if (watchedVideos) {
        for (const video of watchedVideos) {
            watchedVideoIDs.push(video.IndexID);
        }

        if (watchedVideoIDs.includes(videoID)) return 'Abgeschlossen';
        return 'Ausstehend';
    }
    return 'Ausstehend';
}

async function getOwnedProducts() {
    if (await getToken()) {
        return fetch('https://bkbdemy.pxroute.net/api/products/owned', {
            method: 'GET',
            headers: new Headers({
                'Authorization': 'Bearer ' + await getToken(),
                'Content-Type': 'application/json'
            })
        })
```

```
            .then(response => response.json());
    }
    return null;
}

async function updateVideoProgress(ID, points) {
    const element = jQuery('#video-progress-' + ID);
    if (!element.hasClass('Abgeschlossen')) {
        element.addClass('Abgeschlossen');
        await addBalance(points);
    }
}

function setCookie(name, value, expires) {
    const date = new Date();
    date.setTime(date.getTime() + expires * 60 * 1000);
    const expiresUTC = date.toUTCString();
    document.cookie = `${name}=${value}; expires=${expiresUTC}; path=/`;
}

async function updateCourseProgress(updatedCourseMeta) {
    const progress = await getCourseProgress(updatedCourseMeta);
    const container = jQuery('.course-progress');
    container.html("Kursfortschritt: " + progress + '%');
}

function purchaseCourse(courseID, loggedIn) {
    const purchaseButton = jQuery('a[href="#kaufen"]');
    purchaseButton.on('click', function () {
        if (!loggedIn) {
            window.location.href = getHomeUrl() + '/anmeldung';
        } else {
            fetch('https://bkbdemy.pxroute.net/api/products/' + courseID + '/purchase', {
                method: 'POST',
                headers: new Headers({
                    'Authorization': 'Bearer ' + loggedIn,
                    'Content-Type': 'application/json'
                })
            })
                .then(response => {
                    if (response.status === 200) {
                        window.location.href = getHomeUrl() + '/kurse/' + courseID;
                    } else {
                        const container = jQuery('.page-single-course .error-message');
                        container.addClass('active');
                        container.addClass('animate');
                        setTimeout(function () {
                            container.removeClass('animate');
                        }, 1000);
                    }
                })
                .catch(error => {
                    console.error('Error:', error);
                });
        }
    })
}

async function currentUserLevel() {
```

```
    const loggedIn = await getToken();
    if (loggedIn) {

        const userMeta = await getUserData();
        const points = userMeta.points;
        if (points <= 999) return 'ITA <br />' + points + '/' + 1000 + ' Punkte';
        if (points <= 2999) return '1. Ausbildungsjahr <br />' + (points - 1000) + '/' + 3000 + ' Punkte';
        if (points <= 4999) return '2. Ausbildungsjahr <br />' + (points - 3000) + '/' + 5000 + ' Punkte';
        if (points <= 9999) return '3. Ausbildungsjahr <br />' + (points - 5000) + '/' + 10000 + ' Punkte';
        if (points >= 10000) return 'Anwendungsentwickler <br />' + points + ' Punkte';

    }

}

/* Handle User Input*/
function initializeSearchFilter() {
    const checkboxes = jQuery('input[type="checkbox"]');
    const searchFilter = jQuery('.title-filter input');

    searchFilter.on('input', function () {
        const currentValue = searchFilter.val().toLowerCase().trim();
        const checkedAttributes = checkboxes.filter(':checked').map(function () {
            return jQuery(this).attr('name');
        }).get();

        const items = jQuery('.single-course');
        if (checkedAttributes.length > 0) {
            items.filter(function () {
                const $this = jQuery(this);
                const title = $this.find('h3').text().toLowerCase();
                return !checkedAttributes.some(function (attr) {
                    return $this.hasClass(attr);
                }) || title.indexOf(currentValue) === -1;
            }).addClass('hide');
            items.filter(function () {
                const $this = jQuery(this);
                const title = $this.find('h3').text().toLowerCase();
                return checkedAttributes.some(function (attr) {
                    return $this.hasClass(attr);
                }) && title.indexOf(currentValue) !== -1;
            }).removeClass('hide');
        } else {
            items.filter(function () {
                const $this = jQuery(this);
                const title = $this.find('h3').text().toLowerCase();
                return title.indexOf(currentValue) === -1;
            }).addClass('hide');
            items.filter(function () {
                const $this = jQuery(this);
                const title = $this.find('h3').text().toLowerCase();
                return title.indexOf(currentValue) !== -1;
            }).removeClass('hide');
        }
    });

    checkboxes.on('change', function () {
        const currentValue = searchFilter.val().toLowerCase().trim();
        const checkedAttributes = checkboxes.filter(':checked').map(function () {
            return jQuery(this).attr('name');
```

```
        }).get();

        const items = jQuery('.single-course');
        if (checkedAttributes.length > 0) {
            items.filter(function () {
                const $this = jQuery(this);
                const title = $this.find('h3').text().toLowerCase();
                return !checkedAttributes.some(function (attr) {
                    return $this.hasClass(attr);
                }) || title.indexOf(currentValue) === -1;
            }).addClass('hide');
            items.filter(function () {
                const $this = jQuery(this);
                const title = $this.find('h3').text().toLowerCase();
                return checkedAttributes.some(function (attr) {
                    return $this.hasClass(attr);
                }) && title.indexOf(currentValue) !== -1;
            }).removeClass('hide');
        } else {
            items.filter(function () {
                const $this = jQuery(this);
                const title = $this.find('h3').text().toLowerCase();
                return title.indexOf(currentValue) === -1;
            }).addClass('hide');
            items.filter(function () {
                const $this = jQuery(this);
                const title = $this.find('h3').text().toLowerCase();
                return title.indexOf(currentValue) !== -1;
            }).removeClass('hide');
        }
    });
}

function validateUserInput(username, password) {
    const passwordRegex = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[@#$%^&+=?!])(?=\S+$).{8,}$/;

    let errorMessage = "";

    if (username === "") {
        errorMessage += "Bitte geben Sie Ihren gewünschten Nutzernamen ein.\n";
    }
    if (!passwordRegex.test(password)) {
        errorMessage += "Das Passwort muss mindestens 8 Zeichen lang sein und mindestens einen Großbuchstaben,
einen Kleinbuchstaben, eine Ziffer und ein Sonderzeichen enthalten.\n";
    }

    if (errorMessage !== "") {
        return errorMessage;
    } else {
        return true;
    }
}

function returnUserInputAsObject(username, password) {
    const userInput = {
        username: password.val(),
        password: username.val()
    }

    return userInput;
```

```
}

function validateUserRegistrationInput(data) {
    if (data.token === "") {
        const errorContainer = jQuery('.error-message');
        errorContainer.html("Der Benutzername ist bereits vergeben.");
        animateErrorMessage();
    } else {
        localStorage.setItem('authToken', data.token);
        window.location.href = getHomeUrl() + "/registrierung-erfolgreich";
    }
}

function validateUserLoginInput(data) {

    if (data.token === "") {
        const errorContainer = jQuery('.error-message');
        errorContainer.html("Benutzername oder Passwort ungültig!");
        animateErrorMessage();
    } else {
        localStorage.setItem('authToken', data.token);
        window.location.href = getHomeUrl() + "/dashboard";
    }
}

function processUserInput(password, username) {

    const userInput = returnUserInputAsObject(password, username);

    const validationResult = validateUserInput(userInput.username, userInput.password);
    return validationResult;
}

/*Render Functions*/
function renderAddComment(courseID) {


    const container = jQuery('.comment-container');
    const html =
        `
            <div class="container write-comment">
                <h3>Schreibe einen Kommentar</h3>
                <textarea placeholder="Dein Kommentar"></textarea>
                <div class="primary-button submit-comment">Kommentar absenden</div>
            </div>
        `
    container.append(html);

    const button = jQuery('.submit-comment');
    const textarea = jQuery('textarea');

    button.on('click', async () => {
        if (textarea.val() !== "") {
            button.addClass('disabled');
            await addComment(courseID);
        } else {
            textarea.addClass('animate');
            setTimeout(function () {
                textarea.removeClass('animate');
            }, 1000)
```

```
        }
    });
}

function renderComments(data, amount) {
    const container = jQuery('.read-comments');
    container.empty();
    let counter = 0;
    if (data.length > 0) {
        data.forEach(function (comment) {
            if (counter < amount) {
                let date = comment.CreatedAt;

                let formattedDate = new Date(date).toLocaleString('de-DE', {
                    day: '2-digit',
                    month: '2-digit',
                    year: 'numeric',
                    hour: '2-digit',
                    minute: '2-digit',
                    second: '2-digit'
                });

                formattedDate = formattedDate.replace(',', ' um');

                const html =
                    `
                    <div class="single-comment">
                        <h3>${comment.Username}</h3>
                        <p>schrieb am ${formattedDate} Uhr</p>
                        <p>${comment.Text}</p>
                    </div>
                `
                container.append(html);
                counter++;
            }
        })
        container.addClass('flex');
    }

    const element =
        `
            <div class="show-more primary-button">mehr anzeigen</div>
        `
    if (data.length > counter) {
        container.append(element);
        const button = jQuery('.show-more');
        button.on('click', () => {
            counter += 3;
            renderComments(data, counter);
        })
    }
}

function renderNewProducts(data, slider) {
    const length = data.length;
    let counter = 0;
    data.forEach(function (course) {
        if (counter >= length - 3) {
            const content =
                `
```

```
                    <div class="single-course">
                        <h3>${course.Name}</h3>
                        <p>Schwierigkeit: ${returnDifficulty(course.Difficulty)}</p>
                        <img src="https://bkbdemy.pxroute.net${course.Image}">
                        <p class="course-excerpt">${course.Description}</p>
                        <a class="secondary-button" href="${getHomeUrl()}/kurse/${course.ID}">Zum Kurs</a>
                    </div>
                `
            slider.append(content);
        }
        counter++;
    })
}

function renderMostWatchedProducts(data, slider) {
    data.forEach(function (course) {
        const content =
            `
                <div class="single-course">
                    <div class="headline-wrapper">
                        <h3>${course.Name}</h3>
                        <p>Schwierigkeit: ${returnDifficulty(course.Difficulty)}</p>
                    </div>
                    <img src="https://bkbdemy.pxroute.net${course.Image}">
                    <p class="course-excerpt">${course.Description}</p>
                    <a class="secondary-button" href="${getHomeUrl()}/kurse/${course.ID}">Zum Kurs</a>
                </div>
            `
        slider.append(content);
    });
}

function renderOwnedProducts(products) {
    const container = jQuery('#owned-courses');
    if (container.length && products.length) {

        jQuery.each(products, function (index, product) {
            const content =
                `
            <div class="single-course">
                        <h3>${product.Name}</h3>
                        <p>Schwierigkeit: ${returnDifficulty(product.Difficulty)}</p>
                        <img src="https://bkbdemy.pxroute.net${product.Image}">
                        <p class="course-excerpt">${product.Description}</p>
                        <a class="secondary-button" href="${getHomeUrl()}/kurse/${product.ID}">Zum Kurs</a>
                    </div>
            `
            container.append(content);
        })
    } else {
        container.html('Derzeit gehören dir keine Kurse.')
    }
}

function renderSingleCourseResult(data, container) {
    data.forEach(function (course) {
        const content =
            `
                    <div class="single-course difficulty-${course.Difficulty}">
                        <h3>${course.Name}</h3>
```

```
                    <p>Schwierigkeit: ${returnDifficulty(course.Difficulty)}</p>
                    <img src="https://bkbdemy.pxroute.net${course.Image}">
                    <p class="course-excerpt">${course.Description}</p>
                    <a class="secondary-button" href="${getHomeUrl()}/kurse/${course.ID}">Zum Kurs</a>
                </div>
            `
        container.append(content);
    })
}

async function renderCourse(data, products, courseID) {
    const container = jQuery('.single-course:not(#results .single-course)');
    let content = null;
    if (products && products.some(product => product.ID === courseID)) {
        currentCourse = data.Videos;
        const progress = await getCourseProgress(currentCourse);
        content =
            `
                <div class="course-progress-wrapper">
                <h1 class="page-headline">${data.Name}</h1>
                  <div class="content-wrapper">
                    <div class="top">
                        <p class="course-difficulty">Schwierigkeit: ${returnDifficulty(data.Difficulty)}</p>
                        <p class="course-progress">Kursfortschritt: ${progress}%</p>
                        <img src="https://bkbdemy.pxroute.net${data.Image}">
                        <p class="shop-teaser">Weitere coole Kurse findest du hier bei uns im <a class="primary-button"
href="${getHomeUrl()}/kurse">Shop</a></p>
                    </div>
                    <div class="bottom">
                        ${await renderCourseVideos(currentCourse)};
                    </div>
                  </div>
                </div>
                `
        renderAddComment(courseID);
    } else {
        content = `
                <h1 class="page-headline">${data.Name}</h1>
                </div>
                <div class="content-wrapper">
                    <div class="top">
                        <video controls poster="https://bkbdemy.pxroute.net${data.Image}">
                            <source src="https://bkbdemy.pxroute.net${data.PreviewURL}" type="video/mp4">
                        </video>
                        <p class="shop-teaser">Weitere coole Kurse findest du hier bei uns im <a class="primary-button"
href="${getHomeUrl()}/kurse">Shop</a></p>
                    </div>
                    <div class="bottom">
                        <h3 class="course-further-info">
                            <span class="course-price">Preis: ${data.Price} Coins</span>
                        </h3>
                        <p class="course-difficulty">Schwierigkeit: ${returnDifficulty(data.Difficulty)}</p>
                        <p class="course-excerpt">${data.Description}</p>
                        <div class="button-row">
                            <a class="secondary-button" href="#kaufen">Diesen Kurs kaufen</a>
                            <a class="primary-button" href="#zurück">Zurück</a>
                        </div>
                    </div>
                </div>
            `;
```

```
    }
    container.append(content);
}

async function renderCourseVideos(videos) {
    let content = '';
    let currentVideo = 1;
    let courseLength = videos.length;

    for (const video of videos) {
        const videoProgress = await checkSingleVideoProgress(video.IndexID);
        const html = `
            <div class="single-video">
                <h3><span id="video-progress-${video.IndexID}" class="${videoProgress}"></span>${video.Name}
(${currentVideo} von ${courseLength})</h3>
                <p class="points">${video.Points} Punkte</p>
                <p>${video.Description}</p>
                <video id="video-${video.IndexID}" data-src="${video.Points}" controls preload="metadata"
poster="https://bkbdemy.pxroute.net${video.Thumbnail}">
                    <source src="https://bkbdemy.pxroute.net/api/video/${video.IndexID}/stream">
                </video>
            </div>
        `;

        currentVideo++;
        content += html;
    }

    return content;
}


/* initiated by default when the document is ready loaded */
async function initLoadOwnedProducts() {
    const products = await getOwnedProducts();
    renderOwnedProducts(products);
}

function initCookieNotice() {
    const cookieNoticeContainer = jQuery('.cookie-notice');
    const cookieSettings = jQuery('.cookie-settings');

    cookieSettings.on('click', function () {
        cookieNoticeContainer.addClass('active');
    })

    function isCookieAccepted() {
        return document.cookie.indexOf('accept-cookie-notice=1') !== -1;
    }

    if (!isCookieAccepted()) {
        setTimeout(function () {
            cookieNoticeContainer.addClass('active');
        }, 1000);
    }

    const acceptCookies = jQuery('#accept-cookies');
    const denyCookies = jQuery('#deny-cookies');
```

```
    acceptCookies.on('click', function () {
        setCookie('accept-cookie-notice', '1', 30);
        cookieNoticeContainer.removeClass('active');
    })

    denyCookies.on('click', function () {
        /* Usually here you will be redirected but in this Project denying = accepting is okay since its just for showcase
purposes */
        setCookie('accept-cookie-notice', '1', 30);
        cookieNoticeContainer.removeClass('active');
    })

    const navItem = jQuery('.cookie-header ul li');
    const content = jQuery('.content .single-content');

    navItem.on('click', function () {
        const item = $(this);
        const classes = item.attr('class');
        const classesArray = classes.split(' ');
        const navClass = classesArray[0];

        item.addClass('active');
        navItem.not(item).removeClass('active');
        content.removeClass('active');
        jQuery('.content .' + navClass).addClass('active');
    })
}

async function initLoadUserData() {

    if (await getToken()) {
        userMeta = await getUserData();
        const userFrontpageContainer = jQuery('.registration-teaser h3')
        const userDashboardUsernameContainer = jQuery('#my-account #username');
        const userDashboardCurrentBalanceContainer = jQuery('#current-balance');
        const userRegistrationSuccessUsernameContainer = jQuery('#registration-success-username');

        if (userFrontpageContainer.length) userFrontpageContainer.html(userMeta.username)
        if (userDashboardUsernameContainer.length) userDashboardUsernameContainer.html(userMeta.username);
        if (userRegistrationSuccessUsernameContainer.length)
userRegistrationSuccessUsernameContainer.html(userMeta.username);
        if (userDashboardCurrentBalanceContainer.length)
userDashboardCurrentBalanceContainer.html(userMeta.balance + ' Coins');
    }
}

async function initHandleUserContainer() {
    const loggedInContainer = jQuery('.user-logged-in');
    const loggedOutContainer = jQuery('.user-logged-out');

    if (await getToken()) {
        loggedInContainer.addClass('active')
    } else {
        loggedOutContainer.addClass('active');
    }
}

function initLogout() {

    const submit = jQuery('#user-logout-button');
```

```
    submit.on('click', async function () {

        fetch('https://bkbdemy.pxroute.net/api/auth/logout', {
            method: 'POST',
            headers: new Headers({
                'Authorization': 'Bearer ' + await getToken(),
                'Content-Type': 'application/json'
            })
        })
            .then(response => response.json())
            .then(data => {
                localStorage.removeItem('authToken');
                window.location.href = getHomeUrl();
            })
            .catch(error => {
                console.error('Error:', error)
            });
    });
}

function initLogin() {

    const submit = jQuery('#user-login-button');
    const passwordInput = jQuery('#user-password');
    const usernameInput = jQuery('#user-username');

    submit.on('click', function () {
        const loginInput = {
            username: usernameInput.val(),
            password: passwordInput.val()
        }

        fetch('https://bkbdemy.pxroute.net/api/auth/login', {
            method: 'POST',
            body: JSON.stringify(loginInput)
        })
            .then(response => response.json())
            .then(data => {
                validateUserLoginInput(data);
            })
            .catch(error => {
                console.error('Error:', error)
            });
    });
}

function initRegistration() {

    const submit = jQuery('#user-registration-button');
    const passwordInput = jQuery('#user-reg-password');
    const usernameInput = jQuery('#user-reg-username')
    const errorContainer = jQuery('.error-message');

    submit.on('click', function () {

        if (processUserInput(passwordInput, usernameInput) === true) {
            fetch('https://bkbdemy.pxroute.net/api/auth/register', {
                method: 'POST',
                body: JSON.stringify(returnUserInputAsObject(passwordInput, usernameInput))
```

```
            })
                .then(response => response.json())
                .then(data => {
                    validateUserRegistrationInput(data);
                })
                .catch(error => {
                    console.error('Error:', error)
                });
        } else {
            errorContainer.html(processUserInput(passwordInput, usernameInput));
            animateErrorMessage();
        }
    });
}

function initBurgerMenu() {

    const burgerContainer = jQuery('.burger-menu');
    const mobileContainer = jQuery('.nav-bar.mobile');
    const body = jQuery('body');

    burgerContainer.on('click', function () {
        burgerContainer.toggleClass('active');
        body.toggleClass('no-scroll');
        mobileContainer.slideToggle();
    })
}

function initCourseSlider() {

    let container = jQuery('.course-slider');

    if (container.length) {
        jQuery(container).slick({
            arrows: false,
            dots: true,
            infinite: true,
            autoplay: true,
            autoplaySpeed: 10000,
            speed: 600,
            adaptiveHeight: false,
            slidesToShow: 1,
        });
    }
}

function initFrontpageSlider() {

    let container = jQuery('.frontpage-slider');

    if (container.length) {
        jQuery(container).slick({
            arrows: false,
            dots: false,
            infinite: true,
            autoplay: true,
            speed: 300,
            adaptiveHeight: false,
            slidesToShow: 1,
        });
```

```
    }
}

function initNewsSlider() {

    let container = jQuery('.news-slider');

    if (container.length) {
        jQuery(container).slick({
            arrows: false,
            dots: true,
            infinite: true,
            autoplay: true,
            autoplaySpeed: 5000,
            speed: 600,
            adaptiveHeight: true,
            slidesToShow: 3,
            responsive: [
                {
                    breakpoint: 900,
                    settings: {
                        slidesToShow: 1,
                    }
                },
                {
                    breakpoint: 1300,
                    settings: {
                        slidesToShow: 2,
                    }
                }
            ]
        });
    }
}

function initMostWatchedSlider() {
    let container = jQuery('.most-watched-slider');

    if (container.length) {
        jQuery(container).slick({
            arrows: true,
            dots: false,
            infinite: true,
            autoplay: true,
            autoplaySpeed: 5000,
            speed: 600,
            adaptiveHeight: false,
            slidesToShow: 3,
            responsive: [
                {
                    breakpoint: 900,
                    settings: {
                        slidesToShow: 1,
                        dots: true,
                        arrows: false
                    }
                },
                {
                    breakpoint: 1600,
                    settings: {
```

```
                    dots: true,
                    arrows: false
                }
            }
        ]
    });
  }
}

function initScroll() {
  const button = jQuery('.scroll-up');
  const scrollDown = jQuery('.scroll-down')
  const teaserContent = jQuery('#teaser .wrapper');

  button.on('click', function () {
    jQuery(window).scrollTop(0);
  })

  scrollDown.on('click', function () {
    jQuery(window).scrollTop(350)
  })

  jQuery(window).scroll(function () {
    if (jQuery(this).scrollTop() > 150) {
      button.addClass('active')
      teaserContent.addClass('active')
      scrollDown.removeClass('active')
    } else {
      button.removeClass('active');
      scrollDown.addClass('active')
      teaserContent.removeClass('active');
    }
  });
}

async function initLoadNewProducts() {
  const slider = jQuery('.course-slider');

  if (slider.length) {
    fetch('https://bkbdemy.pxroute.net/api/products')
      .then(response => response.json())
      .then(data => {
        renderNewProducts(data, slider);
      })
      .then(() => {
        initCourseSlider()
      })
      .catch(error => {
        console.error(error);
      });
  }
}

async function initLoadAllProducts() {
  const container = jQuery('.page-single-course #results');
  if (container.length) {
    fetch('https://bkbdemy.pxroute.net/api/products')
      .then(response => response.json())
      .then(data => {
        renderSingleCourseResult(data, container);
```

```
            })
            .then(() => {
                initializeSearchFilter();
            })
            .catch(error => {
                console.error(error);
            });
    }
}

async function initLoadMostWatchedProducts() {
    const slider = jQuery('.most-watched-slider');

    if (slider.length) {
        fetch('https://bkbdemy.pxroute.net/api/products')
            .then(response => response.json())
            .then(data => {
                renderMostWatchedProducts(data, slider);
            })
            .then(() => {
                initMostWatchedSlider()
            })
            .catch(error => {
                console.error(error);
            });
    }
}

async function initLoadSingleProduct() {

    const bodyClass = jQuery('body').hasClass('page-single-course');
    const URL = window.location.href;
    const courseID = parseInt(URL.split('/kurse/')[1]);
    const products = await getOwnedProducts();
    const loggedIn = await getToken();

    if (bodyClass) {
        fetch('https://bkbdemy.pxroute.net/api/products/' + courseID)
            .then(response => {
                if (response.status === 404) {
                    window.location.href = getHomeUrl() + '/404';
                } else {
                    return response.json();
                }
            })
            .then(async data => {
                await renderCourse(data, products, courseID);
                loadCourseComments();
            })
            .then(() => {
                courseVideoSlider();
            })
            .then(async () => {
                addToWatchedVideos();
                const backButton = jQuery('a[href="#zurück"]');
                backButton.on('click', function () {
                    window.history.back();
                })
                purchaseCourse(courseID, loggedIn);
            })
```

```
            .catch(error => {
                console.error(error);
            });
    }
}

function initDashboardNavigation() {
    const navItem = jQuery('.page-dashboard nav ul li');
    navItem.on('click', function () {
        const item = jQuery(this);
        const target = jQuery(item.attr('data-src'));
        const contentContainers = jQuery('.dashboard-container');

        contentContainers.not(target).removeClass('active');
        target.addClass('active');
        item.addClass('active');
        navItem.not(item).removeClass('active');
    })
}

async function initIncreaseUserBalance() {
    const button = jQuery('#increase-balance-button');
    const inputField = jQuery('input[name="userBalance"]');
    const responseContainer = jQuery('.page-dashboard .response-message p');
    const currentBalance = jQuery('.page-dashboard #current-balance');

    button.on('click', async function () {
        await addBalance(inputField.val());
        const userData = await getUserData();
        currentBalance.html(userData.balance + " Coins");

        if (inputField.val() > 0) {
            responseContainer.removeClass('error');
            responseContainer.addClass('success');
            responseContainer.html('Herzlichen Glückwunsch. Sie haben Ihren Kontostand um ' + inputField.val() + ' Coins
erhöht. Viel Spaß beim Ausgeben :)');
            responseContainer.addClass('animate');
            setTimeout(function () {
                responseContainer.removeClass('animate');
            }, 1000);
        } else {
            responseContainer.addClass('error');
            responseContainer.removeClass('success');
            responseContainer.html('Bitte geben Sie einen gültigen Wert ein.');
            responseContainer.addClass('animate');
            setTimeout(function () {
                responseContainer.removeClass('animate');
            }, 1000);
        }
    });
}

async function initLoadCurrentUserLevel() {
    const container = jQuery('#user-level');
    const level = await currentUserLevel();
    if (container.length) {
        container.html('Ihr Level: ' + level);
    }
}
```

```
function initVisiblePassword() {
    const element = jQuery('.password-type-toggle');
    element.on('click', function () {
        const clickedElement = jQuery(this);
        const inputField = clickedElement.siblings('input');
        clickedElement.toggleClass('active');
        if (inputField.attr('type') === "password") {
            inputField.attr('type', 'text');
        } else {
            inputField.attr('type', 'password');
        }
    })
}
const webpack = require('webpack');
const path = require('path');
const TerserPlugin = require('terser-webpack-plugin');
const MiniCssExtractPlugin = require('mini-css-extract-plugin');
const CssMinimizerPlugin = require('css-minimizer-webpack-plugin');
const { CleanWebpackPlugin } = require('clean-webpack-plugin');
const CleanTerminalPlugin = require('clean-terminal-webpack-plugin');

module.exports = {
    entry: {
        main: './src/js/index.js',
    },
    output: {
        path: path.resolve(__dirname, "dist"),
        filename: 'main.min.js',
    },
    module: {
        rules: [{
            test: /\.(js)$/,
            exclude: /node_modules/,
            use: ['babel-loader']
        },
            {
                test: /\.(scss|css|sass)$/,
                use: [MiniCssExtractPlugin.loader, 'css-loader', 'sass-loader'],
            },
            {
                test: require.resolve('jquery'),
                use: [{
                    loader: "expose-loader",
                    options: {
                        exposes: ["$", "jQuery"]
                    }
                }]
            }
        ]
    },
    plugins: [
        new webpack.ProvidePlugin({
            $: 'jquery',
            jQuery: 'jquery'
        }),
        new CleanTerminalPlugin(),
        new CleanWebpackPlugin({
            cleanOnceBeforeBuildPatterns: ['dist']
        }),
        new MiniCssExtractPlugin({
```

```
            filename: "main.min.css",
        })
    ],
    performance: {
        hints: false
    },
    optimization: {
        minimize: true,
        minimizer: [
            new TerserPlugin(),
            new CssMinimizerPlugin()
        ],
    },
    devtool: "source-map",
    watch: false,
    watchOptions: {
        ignored: '**/node_modules',
        poll: 1000
    },
    mode: "production"
};
```