

Laura & Yannick

WEBAPP 2.0

Vorbereitet für
Mario Fux

Berufsfachschule Oberwallis
3930 Visp

WEBAPP 2.0

INFORMATIK PROJEKT | ME3 – MODUL 291

Die Reiseblog-Website ist eine einladende Plattform für Abenteuerer und Reiseliebhaber, die ihre Erfahrungen teilen und sich von den Erlebnissen anderer inspirieren lassen möchten. Durchstöbere visuell ansprechende Impressionen vergangener Reisen in der Bildergalerie, um neue Reiseziele zu entdecken. Erkunde Reiseorte auf einer Karte – ein einfacher Button ermöglicht es, mithilfe von DOM-Events, sich geografisch zu orientieren.

Haben Sie Fragen oder möchten Sie uns Feedback geben? Nutzen Sie unser Kontaktformular, um direkt mit uns in Verbindung zu treten. Egal, ob Sie auf einem Desktop oder Mobilgerät unterwegs sind, unsere Website passt sich an verschiedene Bildschirmgrößen an und bietet Ihnen stets eine optimale Darstellung. Navigieren Sie mühelos durch verschiedene Bereiche unserer Website dank unserer übersichtlichen Navigationsleiste. Finden Sie schnell, wonach Sie suchen, mit unserer einfachen Suchleiste. Teilen Sie Ihre Lieblingsbeiträge mit Freunden auf sozialen Medien – ein Klick auf den Teilen-Button genügt, um Ihre Reiseerlebnisse zu verbreiten. Unsere Anzeige informiert Sie über bevorstehende Reiseberichte und hält Sie auf dem Laufenden. Teilen Sie Ihre Gedanken zu unseren Blogbeiträgen mit anderen Reiseliebhabern. Nutzen Sie das unkomplizierte Kommentarformular, um sich mit der Community auszutauschen.

Github

GitHub ist eine Plattform für die Versionsverwaltung und Zusammenarbeit an Projekten. Wir verwenden GitHub für unser Informatik-Konzept, um:

- Änderungen am Projekt zu verfolgen und zu dokumentieren.
 - Gleichzeitig mit anderen zusammenzuarbeiten und Feedback auszutauschen.
 - Probleme oder Verbesserungsvorschläge zu dokumentieren und zu verfolgen.
- GitHub erleichtert die Strukturierung und Organisation unseres Projekts, während wir gleichzeitig transparent arbeiten und unsere Arbeit effizient verwalten können.

Link:

Laura & Yannick

Features

Weltuhr für verschiedene Zeitzonen:

- Inhalt: Anzeige der Uhrzeit in verschiedenen Städten.
 - Aussehen: Klare Anzeige der Uhrzeit für ausgewählte Zeitzonen.
 - Technik: JavaScript zur Anzeige der Uhrzeit in Echtzeit.
-
- Für unser Weltzeituhr-Projekt verwenden wir JavaScript, um die Uhrzeit in Echtzeit anzuzeigen. Um die verschiedenen Zeitzonen abzurufen und die Uhrzeit für jede Stadt zu aktualisieren, nutzen wir APIs, die Zeitinformationen liefern. Konkret verwenden wir wahrscheinlich die "Date" und "Intl" Objekte in JavaScript, um die Zeit für verschiedene Zeitzonen zu formatieren und anzuzeigen. Zusätzlich verwenden wir APIs wie die "Geolocation API", um die aktuellen Standortdaten des Benutzers zu erhalten, falls erforderlich, sowie APIs wie die "Time Zone API" von Google, um die entsprechenden Zeitzoneneinformationen für die ausgewählten Städte abzurufen. In Bezug auf JavaScript-Funktionen verwenden wir wahrscheinlich Timer-Funktionen wie "setInterval()" oder "requestAnimationFrame()", um die Uhrzeit kontinuierlich zu aktualisieren und eine reibungslose Benutzererfahrung zu gewährleisten. Diese Funktionen ermöglichen es uns, die Uhrzeit in Echtzeit zu aktualisieren, ohne die Seite neu laden zu müssen.

Button mit DOM-Event für Kartenansicht:

- Inhalt: Ein Button, der beim Klicken eine vordefinierte Kartenansicht öffnet.
- Aussehen: Ein klar erkennbarer Button mit der Aufschrift "Karte anzeigen".
- Technik: Verwendung von JavaScript für das DOM-Event beim Klicken des Buttons.

Navigationsleiste mit interaktiven Elementen:

- Inhalt: Eine grundlegende Navigationsleiste für Hauptbereiche der Website.
 - Aussehen: Klar sichtbare Menüpunkte, einfaches Design.
 - Technik: HTML, CSS & JavaScript für die Navigationsstruktur und das Layout.
-
- Interaktive Elemente: Um die Benutzererfahrung zu verbessern, integrieren wir interaktive Elemente wie Hover-Effekte oder Klickanimationen, die eine visuelle Rückmeldung beim Navigieren durch die Menüpunkte bieten.
 - Dropdown-Menüs: Wir verwenden JavaScript, um Dropdown-Menüs zu implementieren, die es Benutzern ermöglichen, zusätzliche Navigationsebenen anzuzeigen, wenn sie auf bestimmte Menüpunkte klicken oder über sie fahren.
 - Aktives Menü-Element: Mit JavaScript können wir das aktive Menüelement basierend auf der aktuellen Seite markieren, um Benutzern visuelles Feedback darüber zu geben, auf welcher Seite sie sich befinden.

Einfaches Kontaktformular:

- Inhalt: Ein grundlegendes Kontaktformular für Fragen oder Feedback.
 - Aussehen: Standard-Formularlayout mit Feldern für Name, E-Mail und Nachricht.
 - Technik: HTML für die Formularstruktur, CSS für das Layout.
-
- Für uns ist der E-Mail-Versand die beste Option. Wir können das Formular so einrichten, dass die eingegebenen Daten direkt per E-Mail an eine vordefinierte Adresse gesendet werden. Dies kann mit HTML und dem 'mailto:'-Attribut im Formular-Tag erreicht werden. Allerdings sollten wir beachten, dass diese Methode von den E-Mail-Clients der Benutzer abhängig ist und keine serverseitige Verarbeitung ermöglicht. Daher sollten wir die Einschränkungen dieser Methode berücksichtigen.
 - Wir können das Formular durch clientseitige Validierung (z.B. mit JavaScript) verbessern, um sicherzustellen, dass alle erforderlichen Felder ausgefüllt sind, bevor es abgesendet wird. Nach dem Absenden leiten wir den Benutzer dann auf eine 'Danke'-Seite weiter, um die Bestätigung zu erhalten, dass das Formular erfolgreich übermittelt wurde. Diese Methode bietet eine einfache Lösung, um die Benutzererfahrung zu verbessern und sicherzustellen, dass nur vollständige Formulare übermittelt werden.

Bildergalerie:

- Inhalt: Eine einfache Bildergalerie, die durch statische Bilder blättert.
 - Aussehen: Standard-Bildergalerie-Layout
 - Technik: Einfache HTML- und CSS-Struktur für die Bildergalerie.
-
- HTML-Struktur: Wir erstellen eine HTML-Struktur, die die einzelnen Bilder der Galerie enthält. Jedes Bild wird normalerweise in einem HTML-Element wie `` eingebettet.
 - CSS-Styling: Wir verwenden CSS, um das Layout und das Erscheinungsbild unserer Bildergalerie festzulegen. Dies umfasst Dinge wie die Größe der Bilder, Abstände zwischen den Bildern, den Rahmen und Hintergrund der Galerie sowie möglicherweise Übergangseffekte für das Blättern durch die Bilder.
 - JavaScript für die Interaktivität: Um die Bildergalerie interaktiv zu machen, verwenden wir JavaScript. Hierbei könnten wir beispielsweise Event-Handler wie `onclick` oder `onmouseover` verwenden, um auf Benutzerinteraktionen zu reagieren. Insbesondere könnten wir JavaScript verwenden, um zwischen den Bildern zu blättern, entweder durch Klick auf Schaltflächen oder durch Swipe-Gesten auf mobilen Geräten.
 - Bildwechsel: Um das Durchblättern der Bilder zu ermöglichen, verwenden wir JavaScript, um das aktuell angezeigte Bild zu verfolgen und beim Klicken auf die nächste bzw. vorherige Schaltfläche das entsprechende Bild anzuzeigen.
 - Optionale Funktionalitäten: Je nach Bedarf könnten wir weitere Funktionen hinzufügen, wie z.B. eine Bildergalerie mit Miniaturansichten, automatisches Diashow, Lightbox-Effekte für vergrößerte Bilder usw.

Grundlegendes Responsive Design:

- Inhalt: Eine einfache Anpassung der Website an verschiedene Bildschirmgrößen.
- Aussehen: Klare Darstellung auf Desktops und Mobilgeräten ohne aufwendige Anpassungen.
- Technik: Verwendung von Media Queries in CSS.

Teilen-Button:

- Inhalt: Ein einfacher Button zum Teilen von Blogbeiträgen.
 - Aussehen: Ein einzelner Button mit einem Social-Media-Symbol.
 - Technik: Einfache Verlinkung zu Social-Media-Plattformen in HTML.
-
- Button-Design und Symbol: Der Teilen-Button wird als stilisierter Button mit einem Symbol des entsprechenden Social-Media-Dienstes gestaltet. Zum Beispiel könnte für Facebook ein "f"-Symbol und für Twitter ein "Vogel"-Symbol verwendet werden.
 - Interaktivität mit JavaScript: Wir implementieren JavaScript, um die Interaktivität des Teilen-Buttons zu verbessern. Wenn Benutzer auf den Button klicken, wird eine Popup-Box oder ein Modal mit verschiedenen Optionen zum Teilen auf verschiedenen Plattformen angezeigt.
 - Dynamische URL-Generierung: Durch JavaScript wird die aktuelle URL des Blogbeitrags erfasst und dynamisch in die Share-Links für die verschiedenen Social-Media-Plattformen eingefügt. Auf diese Weise wird der Benutzer direkt zum Teilen des spezifischen Blogbeitrags aufgefordert, ohne die URL manuell kopieren zu müssen.
 - Zähler für geteilte Beiträge: Optional kann ein Zähler hinzugefügt werden, der die Anzahl der bisher geteilten Beiträge auf jeder Plattform anzeigt. Dies erfordert die Integration mit der API der jeweiligen Plattform und die Verwendung von JavaScript, um die Anzahl der Shares abzurufen und anzuzeigen.

Kommentarformular:

- Inhalt: Ein einfaches Formular für Benutzerkommentare.
 - Aussehen: Standard-Formularlayout mit Feldern für Name, E-Mail und Kommentar.
 - Technik: HTML für die Formularstruktur, CSS für das Layout.
-
- Fux Antwort: Siehe Formular oben. Wohin geht der Content? --> Meine Antwort: KEINE AHNUNG!!!!!!!

Anzeige kommender Reiseberichte:

- Inhalt: Eine einfache statische Anzeige von kommenden Reiseberichten.
 - Aussehen: Ein festes Element auf der Seite, das kommende Veröffentlichungen auflistet.
 - Technik: Statische Platzierung in HTML.
-
- Dynamische Datenaktualisierung: Verwendung von JavaScript, um die Liste der kommenden Reiseberichte basierend auf festgelegten Veröffentlichungsdaten zu aktualisieren, ohne dass die Seite neu geladen werden muss.
 - Interaktive Elemente: Implementierung von interaktiven Elementen wie Hover-Effekten oder Klickfunktionen, um Benutzern zusätzliche Informationen über jeden bevorstehenden Reisebericht anzuzeigen, wie z.B. eine kurze Beschreibung oder Links zu weiteren Informationen.
 - Benachrichtigungen: Optional könnten Benachrichtigungsfunktionen hinzugefügt werden, um Benutzer über bevorstehende Veröffentlichungen zu informieren, entweder durch Pop-up-Nachrichten oder Benachrichtigungsleisten auf der Website.
 - OPTIONAL Filterung und Sortierung: Integration von JavaScript-Funktionen zur Filterung und Sortierung der kommenden Reiseberichte basierend auf verschiedenen Kriterien wie Reiseziel, Veröffentlichungsdatum oder Reisetyp.

Zeitplan

Wichtig: Abgabe Ende Woche 21 | Projektpräsentation Woche 22-23

Woche 05	Webapp 2.0 Konzept
Woche 06	Webapp 2.0 Konzept
Woche 07	Ferien
Woche 08	Webapp 2.0 Konzept
Woche 09	Webapp 2.0 Konzept abgeben
Woche 10	Webapp 2.0 Websitestruktur & Navigationsleiste
Woche 11	Webapp 2.0 Bildgalerie
Woche 12	Prüfung 3
Woche 13	Webapp 2.0 DOM-Event & Kontaktformular
Woche 14	Ferien
Woche 15	Webapp 2.0 Zeitzonen Funktion
Woche 16	Webapp 2.0 Zeitzonen Funktion
Woche 17	Webapp 2.0 Responsive Design
Woche 18	Webapp 2.0 Responsive Design
Woche 19	Prüfung 4
Woche 20	Webapp 2.0 Testprotokoll
Woche 21	Abgabe Ende Woche
Woche 22-23	Projektpräsentation