Audit 3- EP

Von Zara Zaric, Melisa Tekcan

Inhaltsverzeichnis

Überarbeitung Audit- 2 S.2

Systemarchitektur S.3

Prototyp Vorschau S.7

Modellierungsbegründung S.14

Durchgeführte PoCs. S.18

Zeitplan S.22

1

Ergänzungen zu Audit 2:

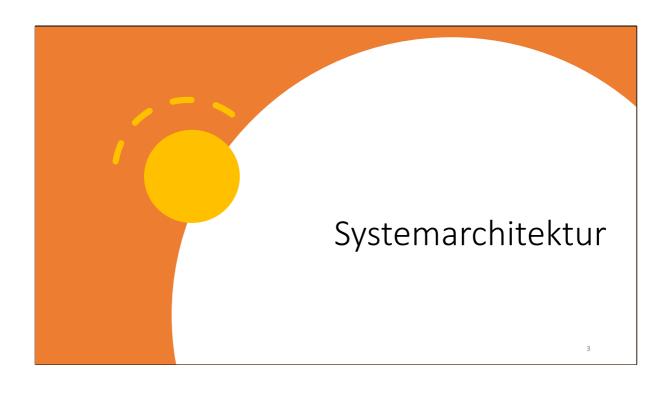
- Domänenmodell https://github.com/mtekcan/EPWS2324Zaric Tekcan/blob/main/Artefakte/Zu%20Audit3/Doma%CC%88nenmodell%20Neu.png

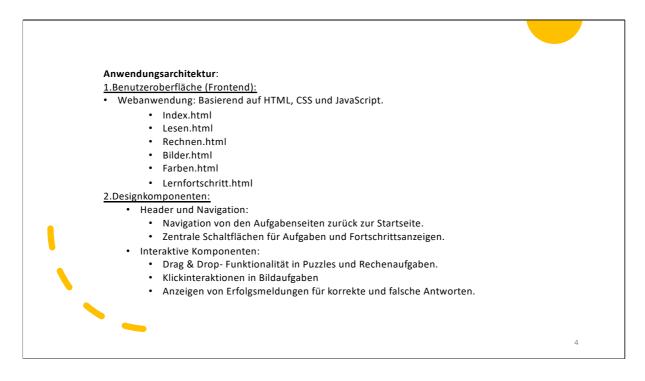
2

Hier sind die Links zum Domänenmodell und den Anforderungen.

Domänenmodell: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/Artefakte/Zu%20Audit3/Doma%CC%88nenmodell%20Neu.png

Anforderungen: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/Art efakte/Zu%20Audit3/Anforderungen%20Neu.png





Für die Entwicklung der Webanwendung haben wir mit Visual Studio Code gearbeitet. Die Einstiegsseite ist die index.html Seite. Von der Seite aus kann man durch die Anwendung Navigieren.

Wenn man dann auf die "Lesen" Seite Navigiert erscheint ein Lesepuzzle zum lösen. Auf der "Rechnen" Seite kann man Rechenaufgaben mit Drag & Drop lösen. Über die "Bilder" Seite erscheint ein Bilderquiz mit Klickaufgaben und über die "Lernfortschritt" Seite werden alle Lernfortschritte für die Aufgabenbereiche angezeigt.

Link zur

Systemarchitektur: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/ Artefakte/Zu%20Audit3/Systemarchitektur.pdf

Backend und Datenverwaltung:

1.Lokale Speicherung (LocalStorage):

- Speichert den aktuellen Fortschritt (z.B.) Anzahl richtiger/falscher Antworten).
- Fortschrittsanzeige für die verschiedenen Bereiche (Lesen, Rechnen, Bilder, Farben).

Systemkomponenten:

1.UI-Komponenten:

- Drag-and-Drop Bereich für Puzzle und Rechenaufgaben
- Bildercontainer für Bilderquiz Aufgaben
- Fortschrittsanzeige, welche aus Fortschrittsdaten gespeichert wird (richtige/falsche Antworten).

Technologiekomponenten:

1.Frontend-Technologien:

- HTML für die strukturierte Seitenbeschreibung.
- CSS für das visuelle Design und Styling.
- JavaScript für die Interaktivität und dynamische Inhalte.

2.Speicherung:

• LocalStorage() für einfache lokale Speicherung der Fortschritte

5

(Siehe Notiz aus der vorherigen Folie)

Link zur

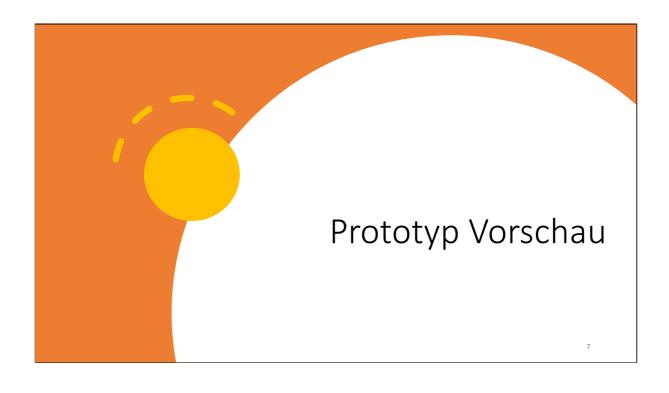
Systemarchitektur: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/ Artefakte/Zu%20Audit3/Systemarchitektur.pdf



(Erklärt sich von selbst)

Link zur

Systemarchitektur: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/Artefakte/Zu%20Audit3/Systemarchitektur.pdf





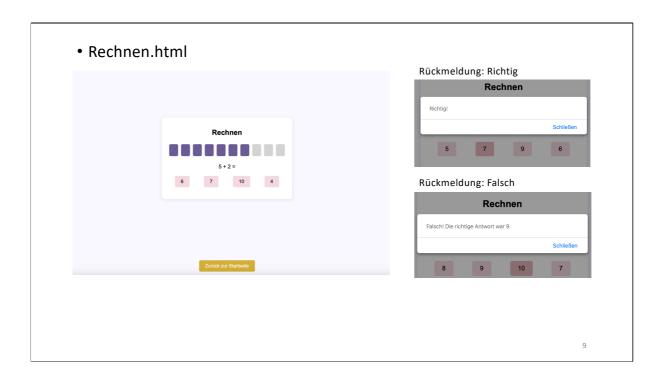
Die Startseite der Webanwendung.

Von hieraus kann der Nutzer einfach zu den Aufgabenbereichen Navigieren.

Ebenfalls kann der Nutzer seine Lernfortschritte einsehen.

Link zum

Code: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/Rapid%20Prototype/index.html



Aufgabenbereich Rechnen:

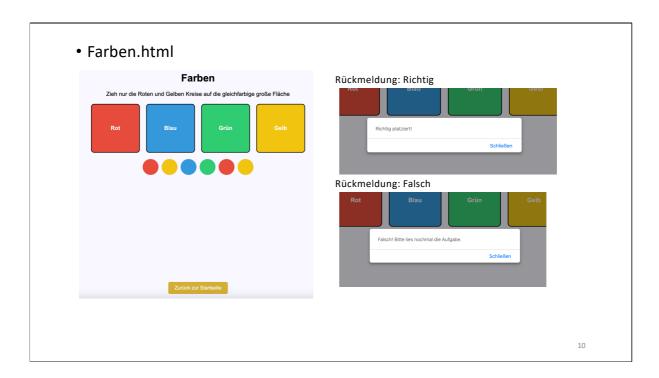
Nach dem der Nutzer seine Antwort angeklickt hat, erscheint eine Rückmeldung, ob die Antwort richtig oder falsch war.

Danach kann der Nutzer die nächste Aufgabe lösen.

Über den "Zurück zur Startseite" Button gelangt der Nutzer zurück zur Startseite.

Link zum

Code: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/Rapid%20Prototype/rechnen.html



Aufgabenbereich Farben:

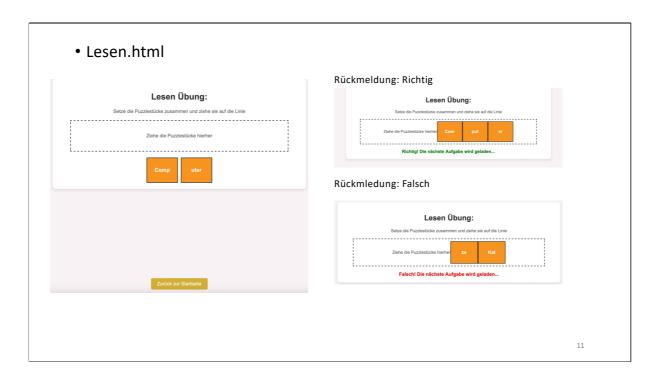
Der Nutzer kann durch Drag & Drop die Farb-kreise zu den richtigen Farb-flecken ziehen.

Auch hier wird immer eine Rückmeldung gegeben, ob die Auswahl richtig oder falsch war.

Über den "Zurück zur Startseite" Button gelangt der Nutzer zurück zur Startseite.

Link zum

Code: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/Rapid%20Prototype/farben.html



Aufgabenbereich Lesen:

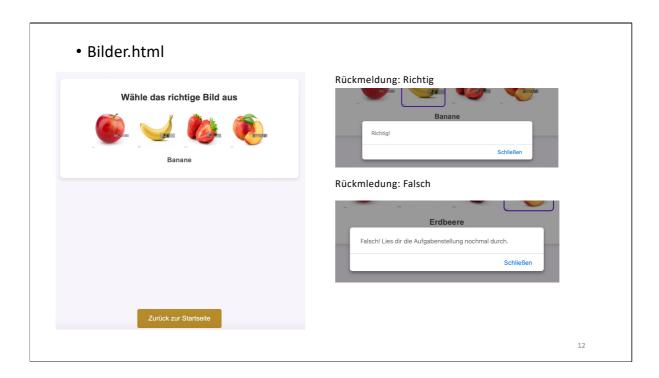
Der Nutzer kann durch Drag & Drop die Puzzlestücke in den Bildcontainer "Ziehe die Zuzzlestücke hierher" ziehen.

Auch hier wird immer eine Rückmeldung gegeben, ob das Zusammengesetze Puzzle dann richtig oder falsch ist.

Über den "Zurück zur Startseite" Button gelangt der Nutzer zurück zur Startseite.

Link zum

Code: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/Rapid%20Prototype/lesen.html



Aufgabenbereich Bilder:

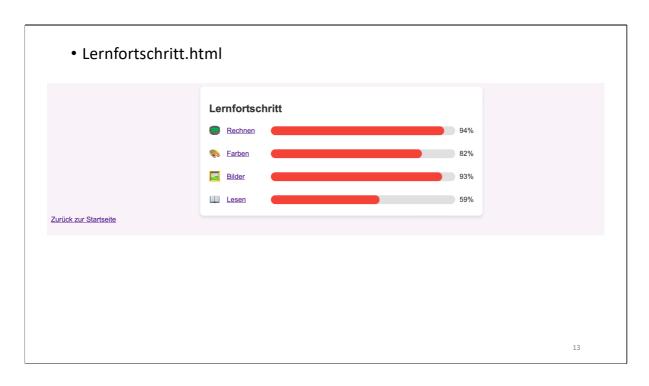
Der Nutzer kann die richtige Antwort anklicken.

Wenn eine Auswahl falsch oder richtig war, wird der Nutzer immer Benachrichtigt mit einer Rückmeldung.

Über den "Zurück zur Startseite" Button gelangt der Nutzer zurück zur Startseite.

Link zum

Code: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/Rapid%20Prototype/bilder.html



Hier kann der Nutzer seine lernfortschritte sehen.

Die Lernfortschritte werden in einem Balkendiagramm veranschaulicht und ebenfalls mit Prozentsätzen gekennzeichnet.

Um den Prozensatz zu berechnen, werden alle richtigen Antworten im Verhältnis zur Gesamtzahl der beantworteten Aufgaben in betracht genommen.

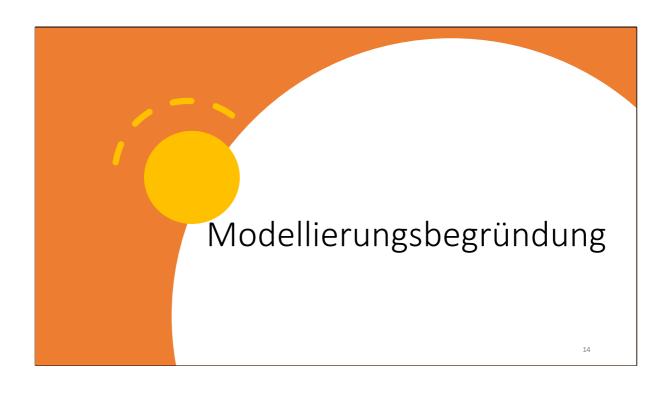
Innerhalb der Funktion wird erst einmal die Gesamtzahl der beantworteten Aufgaben berechnet, indem die Werte von richtigen und falschen Antworten addiert wird. Dann prüft die Funktion, ob die Gesamtzahl der Aufgaben gleich 0 ist. Falls keine Aufgaben beantwortet wurden, gibt die Funktion 0 zurück, um eine Division durch Null zu vermeiden.

Wenn die Gesamtzahl größer als 0 ist, berechnet die Funktion den Prozentsatz der richtigen Antworten, indem der Anteil der richtigen Antworten (Richtig / Falsch) berechnet und anschließend mit 100 multipliziert wird. Das Ergebnis wird gerundet, um einen ganzzahligen Prozentsatz zurückzugeben.

Der Nutzer kann entweder über den Button "Zurück zu Startseite" zu den Aufgabenbereichen Navigieren oder direkt mit einem klick auf die Aufgabenbereiche in der Auswertung vom Lernfortschritt, zu den Aufgabenbereichen Navigieren.

Link zum

Code: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/Rapid%20Prototype/lernfortschritt.html



Index.html

Übersichtlichkeit: Der Code ist in klaren Abschnitten wie header, main und Containern strukturiert, was ihn einfach lesbar und erweiterhar macht

Modularität: Jede Kategorie wie "Rechnen" oder "Farben" führt zu einer eigenen Seite, was die Erweiterung neuer Inhalte erleichtert.

Flexibles Design: Trennung von HTML und CSS ermöglicht einfache Anpassungen des Designs.

Motivation: Ein Smiley und der Lernfortschritt-Button fördern die Motivation und machen die Anwendung interaktiv.

Benutzerfreundlichkeit: Die Buttons sind leicht zu erkennen und sorgen für eine intuitive Bedienung, besonders für Kinder.

Rechnen.html

Modularität: Aufgaben sind in einem Array gespeichert, wodurch sie leicht erweitert werden können.

Benutzerfreundlichkeit: Kindgerechte UI mit interaktiven Elementen (farbige Blöcke, Feedback).

 $Fortschrittsspeicherung: Ergebnisse \ werden \ im \ {\it ",localStorage"} () {\it "gespeichert, um \ den \ Lernfortschritt \ anzuzeigen.}$

Dynamische Logik: Aufgaben werden nacheinander geladen, die Funktion "checkAnswer()" überprüft die Antworten und gibt sofort Feedback "Richtig" oder "Falsch".

Navigation: Einfache Rückkehr zur Startseite und automatische Weiterleitung zur Fortschrittsseite.

Logik: Funktionen wie "updateProgress()" und "loadTasks()", halten den Code modular und wartbar.

15

Link: https://github.com/mtekcan/EPWS2324Zaric Tekcan/blob/main/Artefakte/Zu% 20Audit3/Modellierungsbegru%CC%88ndung.pdf

Farben.html

allowDrop() und drag(): Diese Funktionen ermöglichen das Drag-and-Drop-Verhalten.

Drop() Funktion: Überprüft, ob der gezogene Kreis auf die richtige Zone abgelegt wird. Ist dies der Fall, wird der Fortschritt gespeichert und eine Bestätigung angezeigt. Ansonsten wird der Benutzer aufgefordert, es erneut zu versuchen.

updateProgress(): Speichert den Fortschritt des Spielers (richtig oder falsch) im "localStorage()", damit der Lernfortschritt auch nach dem Schließen des Browsers erhalten bleibt.

isGameCompleted(): Überprüft, ob alle Aufgaben korrekt ausgeführt wurden, um den Übergang zur nächsten Aufgabe zu ermöglichen und den Fortschritt sicherzustellen.

nextTask(): Lädt die nächste Aufgabe, sobald die aktuelle abgeschlossen ist, und setzt das lern Spiel zurück, um mit neuen Aufgaben fortzufahren.

Bilder.html

loadTask(): zeigt die aktuelle Aufgabe mit einer Frage und Bildern, die zum Auswählen da sind.

checkAnswer(): Funktion prüft, ob das ausgewählte Bild richtig ist und speichert den Fortschritt, um den Fortschritt sehen zu können.

localStorage(): Wird verwendet, um den Fortschritt und den aktuellen Aufgabenindex zu speichern.

 Die Bilder-URLs kommen aus einer Bilddatenbank "unsplash" um hochwertige Bilder einzubinden. Sie dienen der visuellen Darstellung der Auswahlmöglichkeiten für den Benutzer.

16

Erklärung:

- "allowDrop" verhindert die Standardaktion und lässt das Ablegen von Elementen
- "drag" setzt die ID des zu ziehenden Elements, um es später korrekt zu platzieren.

Diese Funktionen wurden verwendet, um eine interaktive und fortlaufende Lernerfahrung zu ermöglichen, die den Fortschritt speichert, das richtige Feedback gibt und die Flexibilität bietet. Die loadTask-Funktion zeigt die aktuelle Aufgabe mit einer Frage und den Auswahlbildern an. Die checkAnswer-Funktion überprüft, ob das ausgewählte Bild korrekt ist, und speichert den Fortschritt, damit dieser verfolgt werden kann. Der localStorage wird genutzt, um den Fortschritt und den aktuellen Aufgabenindex zu speichern. Die Bilder-URLs stammen aus einer Bilddatenbank "unsplash", um hochwertige Bilder als Auswahlmöglichkeiten bereitzustellen."

Link: https://github.com/mtekcan/EPWS2324Zaric Tekcan/blob/main/Artefakte/Zu% 20Audit3/Modellierungsbegru%CC%88ndung.pdf

· Lernfortschritt.html

loadProgress(): Ruft den Fortschritt aus dem "localStorage()" ab, sodass der Nutzer seinen Lernstand bei jedem Besuch sehen kann. Diese Entscheidung sorgt dafür, dass der Fortschritt zwischen den Sitzungen erhalten bleibt.

calculatePercentage(): Berechnet den Prozentsatz des Lernfortschritts basierend auf richtigen und falschen Antworten, was eine präzise Ansicht des Fortschritts ermöglicht.

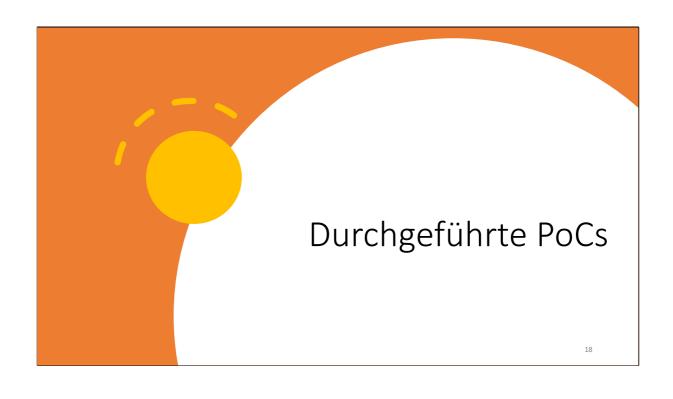
updateProgressItem(): Aktualisiert den Fortschritt jeder Kategorie (z.B. Rechnen, Farben) durch Balken und Prozentsatz. Diese modulare Struktur sorgt für einfache Wartung und Erweiterbarkeit des Codes.

localStorage(): Speichert den Lernfortschritt dauerhaft im Browser, sodass der Nutzer seine Fortschritte zwischen den Sitzungen behält.

Visuelle Darstellung durch Fortschrittsbalken: Der Fortschrittsbalken bietet eine einfache, sofort verständliche Rückmeldung zum Lernstand des Nutzers.

17

Link: https://github.com/mtekcan/EPWS2324Zaric Tekcan/blob/main/Artefakte/Zu% 20Audit3/Modellierungsbegru%CC%88ndung.pdf



1.Erfüllte PoC-Kriterien:

Interaktive Übungen:

- PoC-Kriterium -> Übungen mit Drag & Drop oder Klick-Mechaniken.
- Im Prototyp:
 - Der Prototyp enthält interaktive Elemente:
 - farben.html: Drag & Drop für Rechenaufgaben.
 - bilder.html: Bildauswahl durch Klick.
 - lesen.html: Texte und visuelle Feedbackmechanismen.
 - Feedback auf richtige und falsche Antworten wird bereitgestellt (z. B. durch Alert-Fenster im JavaScript).

19

Link zur PoC

Auflistung: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/Artefakte/https://github.com/mtekcan/Brob/main/Artefakte/https://github.com/mtekcan/Brob/main/Artefakte/ht

2.Erfüllte PoC-Kriterien:

Fortschrittsspeicherung:

- PoC-Kriterium -> Speicherung des Lernfortschritts.
- Im Prototyp:
 - Fortschritte werden mit LocalStorage gespeichert
 - Fortschrittsübersicht wird in lernfortschritt.html angezeigt.

20

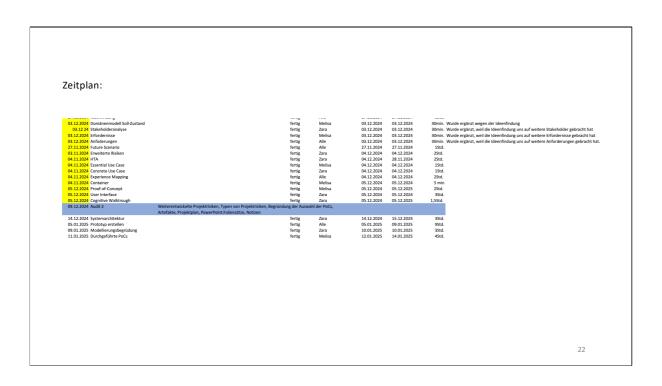
Link zur PoC Auflistung: https://github.com/mtekcan/EPWS2324Zaric Tekcan/blob/m ain/Artefakte/Zu%20Audit3/PoC.pdf

Nicht vollständig erfüllte PoC-Kriterien:

- 1. QR-Code-Authentifizierung
- 2. Feedbackmechanismen
 - o Optische Animationen für korrekt/falsch könnten durch Animationseffekte ergänzt werden.
 - z.B Nach einer richtigen Antwort könnte ein "Daumen hoch"-Symbol eingeblendet werden.
- 3. Fallback-Lösungen
 - o QR-Code-Fallback z.B. durch PIN hinzufügen.

21

Link zur PoC Auflistung: https://github.com/mtekcan/EPWS2324Zaric Tekcan/blob/m ain/Artefakte/Zu%20Audit3/PoC.pdf



Link: https://github.com/mtekcan/EPWS2324Zaric_Tekcan/blob/main/Artefakte/Projektplan.md