

Index.html :

Übersichtlichkeit: Der Code ist in klaren Abschnitten wie header, main und Containern strukturiert, was ihn einfach lesbar und erweiterbar 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).

Fortschrittspeicherung: Ergebnisse werden im „localStorage()“ 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.

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 Lernspiel 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.

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.