

## **Anwendungsarchitektur:**

### 1. Benutzeroberfläche (Frontend):

- Webanwendung: Basierend auf HTML, CSS und JavaScript.
  - Seitenstruktur:
    - index.html: Einstiegsseite der Anwendung mit Navigationsoptionen
    - lesen.html: Lesepuzzle
    - rechnen.html: Rechenaufgaben mit Drag & Drop
    - bilder.html: Bildquiz mit Klickaufgaben
    - lernfortschritt.html: Anzeigen des Fortschritts für die Aufgabenbereiche

### 2. Designkomponenten:

- Header und Navigation:
  - Navigation von den Aufgabenseiten zurück zur Startseite.
  - Zentrale Schaltflächen für Aufgaben und Fortschrittsanzeigen.
- Interaktive Komponenten:
  - Drag & Drop- Funktionalität in Puzzles und Rechenaufgaben.
  - Klickinteraktionen in Bildaufgaben
  - Anzeigen von Erfolgsmeldungen für korrekte und falsche Antworten.

## **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).

## **Systemkomponenten:**

### 1. Benutzerverwaltung:

- Benutzerinteraktionen umfassen die Auswahl von Aufgaben und deren Lösungen

### 2. UI-Komponenten:

- Drag-and-Drop Bereich für Puzzle und Rechenaufgaben
- Bildercontainer für Bildquiz 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

## **Architekturbeziehungen:**

1. Benutzer interagiert mit der Webanwendung.
2. Die Webanwendung greift auf den LocalStorage zu.
3. Drag & Drop und Klick- Aufgaben basierend auf Fronteide-Logik und Interaktivität.
4. Navigation zwischen verschiedenen Seiten erfolgt über "zurück zu Startseite" Buttons.