

Kreativ-Modul 2.0: Revolutionäre Erweiterungskonzepte für HomePilot

🌟 **Vision: Von statischen Aufgaben zum dynamischen Kreativitäts-Ökosystem**

1 NEUE THEMATISCHE KATEGORIEN

A) Zeit-Reisen & Historische Imagination

Konzept: Kinder schlüpfen in historische Rollen und erleben Geschichte kreativ

Aufgaben-Beispiele:

- "Du bist ein Kind im alten Ägypten - beschreibe deinen Tag"
- "Erfinde ein Gespräch zwischen Leonardo da Vinci und einem Smartphone"
- "Wie würde eine Steinzeit-Familie auf Pizza reagieren?"
- "Schreibe einen Brief aus der Zukunft (Jahr 2100) an dich heute"
- "Erfinde ein Museum für verschwundene Dinge (Telefonzelle, Kassette...)"

Lernziele: Historisches Bewusstsein, Perspektivwechsel, Anachronismus als Kreativitätstool

B) Unsichtbare Welten & Mikro-Makro-Kosmos

Konzept: Perspektivwechsel durch Größenordnungen

Aufgaben-Beispiele:

- "Du bist so klein wie eine Ameise - erkunde dein Zimmer"
- "Was denkt ein Wassertropfen auf seiner Reise durch den Wasserkreislauf?"
- "Erfinde eine Geschichte aus Sicht eines Atoms"
- "Du bist plötzlich 100 Meter groß - was siehst du?"
- "Zeichne die Welt, wie sie ein Marienkäfer sieht"

Lernziele: Skalierungsdenken, Naturwissenschaft spielerisch, Perspektivflexibilität

C) Emotionale Landkarten & Gefühls-Architektur

Konzept: Abstrakte Emotionen konkret visualisieren

Aufgaben-Beispiele:

- "Zeichne eine Landkarte deiner Gefühle (Freude-Insel, Wut-Vulkan...)"
- "Wenn Angst ein Gebäude wäre - wie sähe es aus?"
- "Erfinde einen Tanz für das Gefühl 'stolz'"
- "Male Musik: Wie sieht der Klang einer Gitarre aus?"
- "Baue ein Gefühls-Thermometer für deine Woche"

Lernziele: Emotionale Intelligenz, Synästhesie, Selbstreflexion durch Kreativität

D) Paradoxe Welten & Unmögliche Szenarien

Konzept: Logik aufbrechen, um kreatives Denken zu trainieren

Aufgaben-Beispiele:

- "Was wäre, wenn die Schwerkraft rückwärts funktioniert?"
- "Erfinde ein Tier, das gleichzeitig Jäger und Beute ist"
- "Wie würde eine Schule aussehen, in der Kinder die Lehrer sind?"
- "Was passiert in einer Welt, wo Lügen wahr werden?"
- "Erfinde einen Sport, der ohne Regeln funktioniert"

Lernziele: Divergentes Denken, Gedankenexperimente, Systemdenken

E) Sinn-Experimente & Wahrnehmungs-Challenges

Konzept: Bewusste Sinneswahrnehmung als Kreativitätsquelle

Aufgaben-Beispiele:

- "Erfinde 10 Geräusche, die es nicht gibt"
- "Beschreibe Farben für jemanden, der nie sehen konnte"
- "Wie schmeckt die Zahl 7?"
- "Erfinde einen Geruch für Freundschaft"
- "Zeichne Musik mit geschlossenen Augen"

Lernziele: Synästhesie, sensorische Integration, Metapherndenken

F) Erfindungs-Labor mit Constraints

Konzept: Kreativität durch Einschränkungen (SCAMPER-Methode für Kinder)

Aufgaben-Beispiele:

- "Erfinde einen Regenschirm, der NICHT vor Regen schützt - wozu dient er?"
- "Kombiniere zwei unmögliche Dinge: Toaster + Fahrrad = ?"
- "Erfinde ein Problem, für das es noch keine Lösung gibt"
- "Was würdest du aus 100 Papierclips bauen?"
- "Erfinde ein Werkzeug für ein Tier deiner Wahl"

Lernziele: Design Thinking, Ressourcenkreativität, Problemlösung

DYNAMISCHE SPIEL-MODULE

Kreativitäts-Würfel (Random Prompt Generator)

Mechanik:

- Drei Würfel-Kategorien: **WER** (Charakter) + **WO** (Ort) + **WAS** (Aufgabe)
- System generiert zufällige Kombinationen
- Beispiel: "Ein Astronaut" + "in einer Bibliothek" + "erfindet ein Lied"

Implementierung:

javascript

// Datenstruktur


```
creativeDice: {
  characters: ["Roboter", "Hexe", "Zeitreisender" ...],
  locations: ["auf dem Mond", "unter Wasser", "in einer Wolke" ...],
  actions: ["baut", "singt", "erklärt", "verwandelt" ...]
}
```

// Endpoint

POST /api/creative-tasks/dice-roll

→ Returns: random combination + optional difficulty modifier

UI-Element:

- Button " Zufalls-Challenge würfeln"
 - Animation beim Würfeln
 - "Neu würfeln" bis passende Aufgabe gefunden
-

Story-Kettensystem (Kollaborative Narrationen)

Konzept: Mehrere Schüler bauen gemeinsam eine Geschichte

Mechanik:

1. Lehrer startet Story-Kette mit Anfangssatz
2. Schüler A fügt nächsten Satz hinzu (in App oder analog)
3. Schüler B sieht nur letzten Satz, ergänzt weiter
4. Am Ende: Komplette Geschichte sichtbar

Implementierung:

```
javascript
storyChains: {
  id, title, initiatedBy, currentSegment,
  participantIds: [], segments: [
    {studentId, text, order, timestamp}
  ]
}
```

Varianten:

- **Bild-Kette:** Jeder zeichnet Teil eines gemeinsamen Bildes
- **Musik-Kette:** Jeder summt einen Teil einer Melodie
- **Gedicht-Kette:** Jeder eine Zeile

Kreativitäts-Quests (Gamification 2.0)

Konzept: Multi-Step-Challenges mit narrativem Rahmen

Beispiel-Quest: "Der verschwundene Regenbogen"

Schritt 1: Finde 7 bunte Gegenstände in deiner Umgebung

Schritt 2: Erfinde eine Geschichte, warum Farben wichtig sind

Schritt 3: Male ein Bild mit nur EINER Farbe

Schritt 4: Überzeuge jemanden, dass deine Farbe die beste ist

→ Belohnung: "Farb-Forscher" Badge

Datenmodell:

```
javascript
creativeQuests: {
  id, title, narrative, steps: [
    {order, taskType, description, completionCriteria}
  ],
  badge: {name, icon, description}
```

}

Speed-Creativity (Timed Micro-Challenges)

Konzept: Kurze 2-5 Minuten Kreativ-Sprints

Beispiele:

- "30 Sekunden: Kritzle einen Drachen"
- "2 Minuten: Liste 20 Verwendungen für einen Löffel"
- "5 Minuten: Schreibe ein 3-Zeilen-Gedicht über Schuhe"

Psychologie:

- Reduziert Perfektionismus
- Trainiert spontane Kreativität
- Niedrige Hemmschwelle

Implementierung:

- Timer-Widget in UI
 - "Quick Creative Break" Button im Dashboard
 - Optional: Tägliche 5-Min-Challenge-Benachrichtigung
-

ADAPTIVE & PERSONALISIERTE SYSTEME

KI-Aufgaben-Generator (Pattern-Based)

Konzept: System erkennt erfolgreiche Aufgabenmuster und variiert sie

Beispiel-Template:

"Erfinde [OBJEKT] aus [MATERIAL] für [ZWECK]"

→ "Erfinde einen Hut aus Papier für einen Pinguin"

→ "Erfinde ein Haus aus Musik für einen Gedanken"

Implementierung:

javascript

```
taskTemplates: {  
  pattern: "Erfinde {noun} aus {material} für {purpose}",  
  variables: {  
    noun: ["Maschine", "Tier", "Spiel"...],  
    material: ["Licht", "Träumen", "Wasser"...],
```

```
  purpose: ["Freundschaft", "Lernen", "Fliegen"...]
},
  difficulty: (abstraction_level) => score
}
```

// API

POST /api/creative-tasks/generate









{ studentId, preferredCategories, difficulty }

→ Returns: Fresh generated task

Kreativitäts-Profil (Style-Erkennung)

Konzept: System erkennt Stärken und Präferenzen

Dimensionen:

-  Visuell vs.  Verbal
-  Logisch vs.  Intuitiv
-  Fantasie vs.  Konstruktiv
-  Sozial vs.  Individuell

Output:

- "Max ist ein **Visueller Tüftler** - liebt Erfindungen und Zeichnen"
- Aufgaben-Empfehlungen basierend auf Profil
- Aber auch: "Challenge-Vorschlag" außerhalb Komfortzone

Implementierung:

javascript

```
studentCreativityProfile: {
  studentId,
  dimensions: {
    visualVerbal: -0.3, // -1 visual, +1 verbal
    logicalIntuitive: 0.6,
    fantasyPractical: 0.1,
    socialIndividual: -0.4
  },
  preferredCategories: ["Erfindungen", "MINT"],
  completedTasks: 47,
  avgEngagement: 4.2 // 1-5 scale
}
```

Wachstums-Pfade (Skill Trees)

Konzept: RPG-artige Progression durch Kreativitätsbereiche

Beispiel-Baum: "Geschichten-Meister"

Level 1: 3-Sätze-Geschichten schreiben



Level 2: Dialog erfinden



Level 3: Plot-Twist einbauen



Level 4: Charakter-Perspektive wechseln



Level 5: Eigenes Genre erfinden

Visualisierung:

- Interaktiver Skill-Tree im Dashboard
- Graue (locked) und bunte (unlocked) Knoten
- Nächste Aufgaben sichtbar

4 MULTIMEDIALE & CROSS-MODALE ERWEITERUNGEN

Audio-Aufgaben (Voice Recording)

Neue Kategorie: "Klang & Stimme"

Beispiele:

- "Erfinde eine Sprache mit 5 Wörtern"
- "Imitiere 3 Geräusche aus der Natur"
- "Erzähle eine Geschichte in verschiedenen Stimmen"
- "Erfinde einen Song mit nur Körpergeräuschen"

Technisch:

javascript

// Existing schema erweitern

creativeTasks.audioEnabled: boolean

```
studentSubmissions: {  
  taskId, studentId,  
  submissionType: 'audio' | 'text' | 'image',  
  fileUrl: string,  
  transcript?: string // Optional: Speech-to-text  
}
```

Foto-Missions (Real-World Challenges)

Konzept: Schüler dokumentieren Kreativität in ihrer Umwelt

Beispiele:

- "Fotografiere 5 Dinge, die Buchstaben bilden"
- "Finde Gesichter in Alltagsgegenständen (Pareidolie)"
- "Erstelle ein Farb-Gradient aus Fundstücken"
- "Baue ein Miniatur-Theater und fotografiere eine Szene"

Gamification:

- Foto-Sammlung wird zu Portfolio
 - Andere Schüler können "Likes" geben (wenn Lehrer aktiviert)
 - Thematische Foto-Challenges (z.B. "Herbst-Woche")
-

Story-Board Mode (Sequentielle Kreativität)

Konzept: Geschichten in Bildern erzählen (Comic-Prinzip)

Tool:

- 4-6 Panel Template
- Schüler zeichnet/schreibt in Felder
- Exportiert als PDF-Comic

Varianten:

- "Vor/Nachher"-Geschichten (2 Panels)
 - "Problem-Lösung"-Sequenzen
 - "Was-wäre-wenn" Szenarien
-

SOZIALE & KOLLABORATIVE DIMENSIONEN

Partner-Challenges (Duo-Modus)

Mechanik:

- Lehrer weist Aufgabe an zwei Schüler zu
- System zeigt beiden dieselbe Aufgabe
- Optional: Geteiltes Abgabedokument

Beispiel-Aufgaben:

- "Interviewt euch gegenseitig als Erfinder"
 - "Erfindet gemeinsam ein neues Spiel"
 - "Einer zeichnet, einer beschreibt - ohne hinzusehen"
-

Klassen-Projekte (Group Mode)

Konzept: Große kreative Projekte für ganze Klasse

Beispiele:

- **"Die große Weltreise":** Jeder Schüler gestaltet ein Land
- **"Klassenbuch der Erfindungen":** Sammlung aller Ideen
- **"Zeitkapsel":** Jeder beiträgt etwas für Zukunft

Implementierung:




javascript

```
classProjects: {  
  id, title, classId, teacherId,  
  description, deadline,  
  contributionType: 'text' | 'image' | 'audio',  
  contributions: [{studentId, content, submittedAt}]  
}
```

Kreativ-Feedback (Peer Reviews - Optional)

Konzept: Wertschätzende Rückmeldungen zwischen Schülern

Regel: "2 Sterne und 1 Wunsch"

-  "Mir gefällt deine Idee mit..."
-  "Besonders kreativ fand ich..."
-  "Ich würde gerne mehr erfahren über..."

Sicherheit:

- Lehrer muss Peer-Feedback explizit aktivieren
 - Alle Kommentare von Lehrer moderiert (Freigabe vor Sichtbarkeit)
 - Nur positive Formulierungen erlaubt
-

REFLEXIONS- & METAKOGNITIONS-TOOLS

Kreativitäts-Tagebuch

Konzept: Schüler reflektieren ihren kreativen Prozess

Prompts:

- "Was war heute besonders kreativ an meiner Lösung?"
- "Was hat mich inspiriert?"
- "Was würde ich beim nächsten Mal anders machen?"
- "Worauf bin ich stolz?"

Visualisierung:

- Tagebuch-Timeline
 - Stimmungs-Tracker (Emojis für jeden Eintrag)
 - Wortwolke aus häufigen Wörtern
-

Kreativitäts-Strategien (Toolbox)

Konzept: Kinder lernen bewusste Kreativitätstechniken

Techniken (kindgerecht erklärt):

- **SCAMPER:** "Verändere deine Idee: Was kannst du weglassen/hinzufügen?"
- **Random Input:** "Nimm ein zufälliges Wort und verbinde es mit deinem Thema"
- **Reverse Thinking:** "Was wäre das Gegenteil deiner Idee?"
- **Six Hats:** "Denke wie ein Künstler/Wissenschaftler/Träumer..."

Implementierung:

- Jede Strategie hat Tutorial-Video (oder Text)
 - Button "Kreativ-Hilfe" bei Aufgaben
 - System schlägt passende Technik vor
-

7 THEMATISCHE SPEZIAL-WOCHEN

Event-basierte Content-Drops

Konzept: Zeitlich begrenzte thematische Aufgaben-Sets

Beispiele:

"Erfinder-Woche"

- Tag 1: Erfinde ein Fahrzeug für den Ozean
- Tag 2: Verbessere eine Alltags-Erfindung
- Tag 3: Erfinde etwas, das es in 100 Jahren gibt
- Tag 4: Patent für deine Erfindung schreiben
- Tag 5: Präsentation vor imaginärer Jury

"Mythen & Legenden"

- Eigene Götter erfinden
- Moderne Heldensage schreiben
- Fabelwesen neu kombinieren

"Zukunfts-Vision"

- Schule der Zukunft
- Berufe, die es noch nicht gibt
- Klimaneutrale Erfindungen

Implementierung:

javascript

```
seasonalEvents: {  
  id, title, startDate, endDate,  
  taskIds: [...],  
  specialBadge: {name, icon},  
  isActive: boolean  
}
```

8 INKLUSION & DIFFERENZIERUNG

Mehrsprachige Aufgaben

Konzept: Aufgaben in verschiedenen Sprachstufen

- **DaZ-Level 1:** Einfache Sätze, Bildstützen

- **DaZ-Level 2:** Standarddeutsch
- **DaZ-Level 3:** Komplexe Formulierungen

Beispiel:

Level 1: "Male ein Tier. Das Tier ist bunt."

Level 2: "Erfinde ein buntes Fantasie-Tier."

Level 3: "Kreiere ein chimärisches Wesen mit unerwarteten Farbkombinationen."

Barrierefreiheit (Accessibility)

Features:

- **Vorlesefunktion:** Alle Aufgaben als Audio
- **Gebärdensprache-Videos:** Für gehörlose Kinder (optional)
- **Leichte Sprache:** Vereinfachte Versionen
- **Symbolstütze:** Icons zu abstrakten Begriffen
- **Hoher Kontrast Modus:** Für Sehschwäche

Begabtenförderung (Advanced Tracks)

Konzept: Hochkomplexe Aufgaben für schnelle Lerner

Beispiele:

- "Erfinde ein mathematisches Paradoxon"
- "Entwickle eine neue Kunstform"
- "Schreibe eine Geschichte mit unreliablen Erzähler"
- "Kritisiere eine berühmte Erfindung philosophisch"

9 ANALYTICS & PÄDAGOGISCHE INSIGHTS

Kreativitäts-Dashboard für Lehrer

Neue Metriken:

Klassenebene:

- Durchschnittliche Bearbeitungszeit pro Kategorie
- Beliebtheits-Ranking der Aufgaben
- "Kreativitäts-Diversität" (wie breit streuen Schüler?)
- Engagement-Trends über Zeit

Schülerebene:

- Kreativitätsprofil (siehe oben)
- "Flow-Indicator" (welche Aufgaben führen zu tiefem Engagement?)
- Entwicklungs-Kurve (Fortschritt über Wochen)
- Empfohlene nächste Schritte

Implementierung:

javascript

GET /api/analytics/creativity/:classId

→ Returns: {

overallEngagement: 4.1,

topCategories: [...],

studentProfiles: [...],

recommendations: [...]

}



A/B-Testing System

Konzept: Verschiedene Aufgaben-Formulierungen testen

Beispiel:

- Version A: "Male ein Fantasie-Tier"
- Version B: "Erfinde ein Tier, das es noch nie gab"

Tracking:

- Welche Version wird häufiger zugewiesen?
 - Welche führt zu längerer Bearbeitungszeit?
 - Welche erzeugt mehr Rückfragen?
-



TECHNISCHE IMPLEMENTIERUNGS-ROADMAP

Phase 1: Foundation (Weeks 1-2)

javascript

// Datenbank-Erweiterungen

- Add: creativeDice table

- Add: storyChains table

- Add: creativeQuests table

- Add: studentCreativityProfile table
- Add: studentSubmissions [table](#) (für Multimedia)
- Extend: creativeTasks mit audioEnabled, imageEnabled, difficulty

// API-Endpoints

POST /api/creative-tasks/dice-roll

POST /api/creative-tasks/generate *// Template-basiert*

GET /api/students/:id/creativity-profile

POST /api/story-chains/create

POST /api/submissions/upload *// Multipart für Dateien*

Phase 2: UI-Komponenten (Weeks 3-4)

typescript

// Neue React-Komponenten

<CreativityDiceRoller />

<SkillTreeVisualization />

<QuickChallengeTimer />

<StoryChainBuilder />

<CreativityProfileCard />

<SubmissionUploader /> *// Mit Drag&Drop*

Phase 3: Gamification (Weeks 5-6)

javascript

// Badge-System

```
badges: {
  id, name, icon, description,
  unlockCriteria: {type, threshold}
}
```

// Quest-Engine

```
questProgress: {
  studentId, questId,
  completedSteps: [...],
  currentStep: number,
  isCompleted: boolean
}
```

Phase 4: Intelligenz & Adaptivität (Weeks 7-8)

javascript






```
// Machine Learning Lite (Rule-based zunächst)
function recommendNextTask(studentId) {
  const profile = getCreativityProfile(studentId);
  const completed = getCompletedTasks(studentId);
  const unused = getUnusedTasks();

  // Algorithmus:
  // 70% aus bevorzugten Kategorien
  // 20% aus wenig genutzten (Exploration)
  // 10% Challenge außerhalb Komfortzone






  return weightedRandom(unused, profile);
}
```

PRIORISIERTE FEATURE-LISTE






Must-Have (MVP+)

1.  Kreativitäts-Würfel (random prompt generator)
2.  Story-Ketten-System (kollaborativ)
3.  Speed-Creativity Timer
4.  3 neue Themen-Kategorien (Zeit-Reisen, Unsichtbare Welten, Paradoxe Welten)
5.  Einfaches Badge-System

Should-Have (v2.0)

6.  KI-Aufgaben-Generator (template-based)
7.  Kreativitäts-Profil (Tracking)
8.  Foto-Missions
9.  Partner-Challenges
10.  Reflexions-Tagebuch

Nice-to-Have (v3.0)

11.  Kreativitäts-Quests (multi-step)
12.  Skill Trees
13.  Audio-Aufgaben
14.  Story-Board Mode
15.  Event-basierte Content-Drops

Kreativ-Modul 2.0: Lehrkraft-zentrierte Implementierung mit Interessen-Matching

 **Kern-Konzept: Von Kinderinteressen zu personalisierten A4-Arbeitsblättern**

1 INTERESSEN-PROFIL-SYSTEM

A) Interessen-Interview-Modul

Konzept: Strukturiertes Gespräch zwischen Lehrkraft und Kind, digital erfasst

Interview-Struktur (5-10 Minuten)

typescript

```
interface StudentInterestProfile {  
  studentId: number;  
  interviewDate: Date;  
  
  // Kategorie 1: Lieblingsdinge  
  favoriteAnimals: string[]; // "Delfin, Drache, Katze"  
  favoriteColors: string[]; // "Blau, Gold"  
  favoriteActivities: string[]; // "Fußball, Zeichnen, Bauen"  
  favoriteSubjects: string[]; // "Mathe, Kunst"  
  
  // Kategorie 2: Träume & Wünsche  
  dreamJob: string; // "Astronautin"  
  dreamPlace: string; // "Weltraum, Dschungel"  
  superPower: string; // "Fliegen können"  
  
  // Kategorie 3: Hobbies & Expertise  
  hobbies: string[]; // "LEGO, Tanzen, Kochen"  
  collections: string[]; // "Steine, Pokemon-Karten"  
  expertise: string[]; // "Kennt viele Dinosaurier-Namen"  
  
  // Kategorie 4: Emotionale Präferenzen  
  likesFantasy: boolean; // Mag Fantasiewelten?  
  likesRealism: boolean; // Mag echte Dinge/Fakten?
```



```

likesHumor: boolean;    // Mag lustige Sachen?
likesChallenge: boolean; // Mag knifflige Aufgaben?

// Kategorie 5: Soziale Dimension
prefersAlone: boolean;  // Arbeitet gern allein
prefersPartner: boolean; // Mag Partnerarbeit
prefersGroup: boolean;  // Mag Gruppenarbeit

// Kategorie 6: Kreative Vorlieben
expressionMode: string[]; // "Zeichnen, Schreiben, Bauen, Bewegen"
attention_span: 'short' | 'medium' | 'long';

// Metadaten
lastUpdated: Date;
completeness: number; // 0-100%
}


```













B) Interview-UI für Lehrkräfte

Bildschirm 1: Schnell-Interview (Minimal-Modus)

Interessen-Profil: Max Mustermann	
<div> <div></div> <div></div> </div>	
<div> <div>🐾</div> <div>Lieblingstiere (max. 3)</div> </div>	<div> <div></div> <div></div> </div>
<div> <div></div> <div></div> </div>	<div> <div></div> <div></div> </div>
<div> <div>🎨</div> <div>Was macht Max gerne? (Hobbies)</div> </div>	<div> <div></div> <div></div> </div>
<div> <div></div> <div></div> </div>	<div> <div></div> <div></div> </div>
<div> <div>🌟</div> <div>Traumberuf / großer Wunsch</div> </div>	<div> <div></div> <div></div> </div>
<div> <div></div> <div></div> </div>	<div> <div></div> <div></div> </div>
<div> <div>⚡</div> <div>Superkraft (wenn möglich)</div> </div>	<div> <div></div> <div></div> </div>
<div> <div></div> <div></div> </div>	<div> <div></div> <div></div> </div>
<div> <div>✓</div> <div>Speichern & Aufgaben generieren</div> </div>	

Bildschirm 2: Ausführliches Interview (Experten-Modus)

Interessen-Interview: Sophie Schmidt	
 17	Letztes Update: vor 3 Wochen

Tab: [Favoriten] [Hobbies] [Kreativ] [?]	
=== FAVORITEN ===	
Tiere: [+ Hinzufügen]	
 Einhorn	 Delfin  Drache
Farben: [Farbwähler]	
 Blau	 Gelb  Lila
Aktivitäten/Spiele:	
 Fußball	 Malen  Tanzen
=== TRÄUME & WÜNSCHE ===	
 Traumberuf: [Tierärztin_____]	
 Traumort: [Strand/Meer_____]	
 Superkraft: [Mit Tieren sprechen__]	
=== KREATIVE VORLIEBEN ===	
Drückt sich am liebsten aus durch:	
<input checked="" type="checkbox"/> Zeichnen	<input checked="" type="checkbox"/> Schreiben <input type="checkbox"/> Bauen
<input type="checkbox"/> Bewegung	<input checked="" type="checkbox"/> Erzählen
Mag eher:	
<input checked="" type="radio"/> Fantasie-Geschichten	
<input type="radio"/> Realistische Themen	
<input type="radio"/> Mix aus beidem	
[Profil speichern] [Aufgaben vorschlagen]	

C) Smarte Interview-Vorschläge

System gibt Lehrkraft Frage-Prompts basierend auf Alter:

Klasse 1-2:

- "Was ist dein Lieblings-Tier?"
- "Womit spielst du am liebsten?"
- "Was möchtest du mal werden?"

Klasse 3-4:

- "Wenn du eine Superkraft hättest, welche?"
- "Was sammelst du? Was interessiert dich?"
- "Worüber weißt du viel?"

Klasse 5-6:

- "Was würdest du gerne besser können?"
 - "Welches Thema fasziniert dich gerade?"
 - "Wenn du etwas erfinden könntest - was?"
-

2 INTELLIGENTER AUFGABEN-GENERATOR

A) Matching-Algorithmus: Interesse → Aufgabe

typescript

```
interface TaskTemplate {  
  id: string;  
  category: string;  
  basePrompt: string;    // "Erfinde eine Geschichte über {ANIMAL}"  
  variables: {  
    ANIMAL?: string;  
    HOBBY?: string;  
    COLOR?: string;  
    PLACE?: string;  
    DREAM?: string;  
    SUPERPOWER?: string;  
    OBJECT?: string;  
  };  
  requiredInterests: string[]; // ["favoriteAnimals"]  
}
```

```

optionalInterests: string[]; // ["dreamPlace", "favoriteColors"]
difficulty: 1 | 2 | 3;
estimatedMinutes: number;
outputFormat: 'text' | 'drawing' | 'mixed';
}

```

Beispiel-Templates mit Variablen:

javascript

```

const templates = [
  {
    id: "story_animal_adventure",
    category: "Geschichten",
    basePrompt: "Schreibe eine Geschichte: Dein Lieblings-Tier {ANIMAL} macht eine Reise nach {PLACE}. Was erlebt es dort?",
    requiredInterests: ["favoriteAnimals"],
    optionalInterests: ["dreamPlace"],
    difficulty: 2,
    outputFormat: "text"
  },

  {
    id: "invention_hobby",
    category: "Erfindungen",
    basePrompt: "Erfinde eine Maschine, die das {HOBBY} noch cooler macht. Zeichne sie und beschreibe, wie sie funktioniert.",
    requiredInterests: ["hobbies"],
    difficulty: 2,
    outputFormat: "mixed"
  },

  {
    id: "superpower_school",
    category: "Perspektiven",
    basePrompt: "Stell dir vor, du hast die Superkraft '{SUPERPOWER}'. Wie würde dein Schultag aussehen? Schreibe 5 Dinge auf, die anders wären.",
    requiredInterests: ["superPower"],
    difficulty: 1,
    outputFormat: "text"
  },

```

```
{
  id: "dreamjob_interview",
  category: "Soziale Kompetenzen",
  basePrompt: "Du bist schon {DREAMJOB}. Jemand interviewt dich: Wie bist du das geworden? Was machst du jeden Tag? Was liebst du an deinem Beruf?",
  requiredInterests: ["dreamJob"],
  difficulty: 2,
  outputFormat: "text"
},

{
  id: "color_world",
  category: "Kreativität & Kunst",
  basePrompt: "Stell dir eine Welt vor, in der alles {COLOR} ist. Male ein Bild von dieser Welt. Was gibt es dort? Wie fühlen sich die Menschen?",
  requiredInterests: ["favoriteColors"],
  difficulty: 1,
  outputFormat: "drawing"
},

{
  id: "collection_museum",
  category: "Kreativität & Kunst",
  basePrompt: "Du eröffnest ein Museum für {COLLECTION}. Gestalte ein Poster: Was ist ausgestellt? Warum ist deine Sammlung besonders? Wie heißt das Museum?",
  requiredInterests: ["collections"],
  difficulty: 2,
  outputFormat: "mixed"
},

{
  id: "expertise_teacher",
  category: "Soziale Kompetenzen",
  basePrompt: "Du kennst dich super gut aus mit {EXPERTISE}. Erkläre einem jüngeren Kind (1. Klasse) drei spannende Dinge darüber. Benutze einfache Worte.",
  requiredInterests: ["expertise"],
  difficulty: 3,
  outputFormat: "text"
},

{
```

```

    id: "hobby_sport_new",
    category: "Erfindungen",
    basePrompt: "Kombiniere dein Hobby '{HOBBY}' mit einem Sport. Erfinde eine neue Sportart! Wie heißt sie? Was sind die Regeln? Was braucht man dafür?",
    requiredInterests: ["hobbies"],
    difficulty: 3,
    outputFormat: "mixed"
  },

  {
    id: "animal_superpower",
    category: "MINT-Förderung",
    basePrompt: "Dein Lieblings-Tier {ANIMAL} hat eine besondere Fähigkeit. Forsche nach: Was kann es wirklich gut? Warum? Erkläre es wie ein Wissenschaftler.",
    requiredInterests: ["favoriteAnimals"],
    difficulty: 2,
    outputFormat: "text"
  },

  {
    id: "place_travel_guide",
    category: "Geschichten",
    basePrompt: "Du schreibst einen Reiseführer für {PLACE}. Was müssen Besucher unbedingt sehen? Welche Geheimtipps hast du? Male eine Landkarte dazu.",
    requiredInterests: ["dreamPlace"],
    difficulty: 2,
    outputFormat: "mixed"
  }
];

```

B) Generator-Logik

typescript

```

function generatePersonalizedTasks(
  studentProfile: StudentInterestProfile,
  count: number = 5,
  categoryFilter?: string
): PersonalizedTask[] {

```

// 1. Filter: Welche Templates passen zu den Interessen?

```

const matchingTemplates = templates.filter(template => {
  return template.requiredInterests.every(interest =>
    studentProfile[interest] && studentProfile[interest].length > 0
  );
});

// 2. Score berechnen für jedes Template
const scoredTemplates = matchingTemplates.map(template => {
  let score = 0;

  // Bonus: Optionale Interessen vorhanden
  template.optionalInterests.forEach(interest => {
    if (studentProfile[interest]?.length > 0) score += 2;
  });

  // Bonus: Passt zu Ausdrucksform
  if (studentProfile.expressionMode.includes(template.outputFormat)) {
    score += 3;
  }

  // Bonus: Passende Kategorie-Präferenz (falls vorhanden)
  if (template.category === studentProfile.favoriteCategories?.[0]) {
    score += 5;
  }

  return { template, score };
});

// 3. Top-Templates auswählen
const selectedTemplates = scoredTemplates
  .sort((a, b) => b.score - a.score)
  .slice(0, count)
  .map(item => item.template);

// 4. Variablen ersetzen mit echten Werten
return selectedTemplates.map(template => {
  let prompt = template.basePrompt;

  // Ersetze {ANIMAL} mit zufälligem Lieblings-Tier

```

```
if (prompt.includes('{ANIMAL}') && studentProfile.favoriteAnimals?.length) {  
  const animal = randomChoice(studentProfile.favoriteAnimals);  
  prompt = prompt.replace('{ANIMAL}', animal);  
}
```

```
// Ersetze {HOBBY}
```

```
if (prompt.includes('{HOBBY}') && studentProfile.hobbies?.length) {  
  const hobby = randomChoice(studentProfile.hobbies);  
  prompt = prompt.replace('{HOBBY}', hobby);  
}
```

```
// Ersetze {COLOR}
```

```
if (prompt.includes('{COLOR}') && studentProfile.favoriteColors?.length) {  
  const color = randomChoice(studentProfile.favoriteColors);  
  prompt = prompt.replace('{COLOR}', color);  
}
```

```
// Ersetze {PLACE}
```

```
if (prompt.includes('{PLACE}')) {  
  const place = studentProfile.dreamPlace || "einen magischen Ort";  
  prompt = prompt.replace('{PLACE}', place);  
}
```

```
// Ersetze {DREAMJOB}
```

```
if (prompt.includes('{DREAMJOB}')) {  
  const job = studentProfile.dreamJob || "dein Traumberuf";  
  prompt = prompt.replace('{DREAMJOB}', job);  
}
```

```
// Ersetze {SUPERPOWER}
```

```
if (prompt.includes('{SUPERPOWER}')) {  
  const power = studentProfile.superPower || "unsichtbar sein";  
  prompt = prompt.replace('{SUPERPOWER}', power);  
}
```

```
// Ersetze {COLLECTION}
```

```
if (prompt.includes('{COLLECTION}') && studentProfile.collections?.length) {  
  const collection = randomChoice(studentProfile.collections);  
  prompt = prompt.replace('{COLLECTION}', collection);  
}
```



```

}

// Ersetze {EXPERTISE}
if (prompt.includes('{EXPERTISE}') && studentProfile.expertise?.length) {
  const expertise = randomChoice(studentProfile.expertise);
  prompt = prompt.replace('{EXPERTISE}', expertise);
}

return {
  taskId: `generated_${Date.now()}_${Math.random()}`,
  title: template.category,
  description: prompt,
  category: template.category,
  difficulty: template.difficulty,
  estimatedMinutes: template.estimatedMinutes,
  personalizedFor: studentProfile.studentId,
  matchScore: 'high' // Könnte detaillierter sein
};
});
}

```

C) UI: Aufgaben-Generator im Schüler-Detail

Schüler: Emma Müller	Klasse 3b
<div>  Interessen-Profil  Bearbeiten </div>	
<div> <div>  Tiere: Katze, Einhorn, Delfin </div> <div>  Hobbies: Tanzen, Basteln, Kochen </div> <div>  Traum: Tierärztin werden </div> <div>  Superkraft: Fliegen können </div> <div>  Mag: Fantasie-Geschichten, Zeichnen </div> </div>	
<div> <div>  PERSONALISIERTE AUFGABEN </div> </div>	

 Neue Vorschläge generieren] Anzahl: [5 ▼]

★ SEHR PASSEND

✓ Geschichten: "Schreibe eine Geschichte: Deine Lieblings-Katze macht eine Reise zu einem magischen Ort. Was erlebt sie dort?"

 Text •  20 Min •  Match: 95%

[+ Zuweisen]

✓ Erfindungen: "Erfinde eine Maschine, die das Tanzen noch cooler macht. Zeichne sie und beschreibe, wie sie funktioniert."

 Gemischt •  25 Min •  Match: 92%

[+ Zuweisen]

💡 GUT PASSEND

○ Soziale Komp.: "Du bist schon Tierärztin. Jemand interviewt dich: Wie bist du das geworden? Was machst du jeden Tag?"

 Text •  15 Min •  Match: 88%

[+ Zuweisen]

○ MINT: "Deine Lieblings-Katze hat besondere Fähigkeiten. Forsche nach: Was kann sie gut? Warum? Erkläre es wie ein Wissenschaftler."

 Text •  20 Min •  Match: 85%

[+ Zuweisen]

 Alle zugewiesenen Aufgaben als PDF] (8 Aufgaben)

3 ERWEITERTES PDF-ARBEITSBLATT-SYSTEM

A) Personalisierungs-Features im PDF

typescript

```
interface WorksheetOptions {  
  includeStudentName: boolean;    // "Für Emma" oben  
  includeInterestReminder: boolean; // "Deine Lieblingstiere: ..."  
  includeMotivationalIntro: boolean; // "Das sind Aufgaben speziell für dich!"  
  layoutStyle: 'compact' | 'spacious'; // Mehr Platz zum Schreiben  
  includeDoodles: boolean;        // Kleine passende Icons (Katze, Einhorn...)  
  fontSizeAdjustment: 'small' | 'medium' | 'large';  
}
```

PDF-Layout Beispiel:

✨ KREATIVE HAUSAUFGABEN ✨	
Für Emma Müller	
Hallo Emma! Das sind Aufgaben, die extra für dich gemacht wurden – mit deinen Lieblingsdingen:	
🐱 Katzen	🦄 Einhörner 🕺 Tanzen
<hr/>	
📝 AUFGABE 1: Geschichten-Abenteuer	
Schreibe eine Geschichte: Deine Lieblings-Katze macht eine Reise zu einem magischen Ort. Was erlebt sie dort?	
<hr/>	
[Linien für Text]	

	💡 Tipp: Denk dir einen lustigen Namen für		
	die Katze aus!		
	✎ AUFGABE 2: Tanz-Erfindung		
	Erfinde eine Maschine, die das Tanzen noch		
	cooler macht. Zeichne sie hier:		
	[Leerer Raum für Zeichnung]		
	Beschreibe, wie die Maschine funktioniert:		
	[Linien]		

B) Template-Bibliothek für verschiedene Aufgabentypen

typescript

```
const pdfTemplates = {
```

```
// Story-Writing Template
```

```
story: {
```

```
  titleFormat: "✎ AUFGABE {number}: {category}",
```

```
  includePrompt: true,
```

```
  writingLines: 15,
```

```
  includeIllustrationBox: true,
```

```
illustrationBoxLabel: "Male ein Bild zu deiner Geschichte:",
tips: [
  "Denk dir gute Namen für die Figuren aus!",
  "Was passiert am Anfang? Was am Ende?",
  "Nutze spannende Wörter!"
]
},
```

// Drawing/Mixed Template

```
drawing: {
  titleFormat: "✎ AUFGABE {number}: {category}",
  includePrompt: true,
  drawingBoxSize: 'large', // 60% der Seite
  descriptionLines: 8,
  descriptionLabel: "Beschreibe deine Zeichnung:",
  tips: [
    "Denk an viele Details!",
    "Benutze bunte Farben!",
    "Was macht deine Zeichnung besonders?"
  ]
},
```

// List/Investigation Template

```
list: {
  titleFormat: "🔍 AUFGABE {number}: {category}",
  includePrompt: true,
  bulletPoints: 10,
  numberedList: true,
  tips: [
    "Versuch mindestens 5 Punkte zu finden!",
    "Sei kreativ – es gibt keine falschen Antworten!"
  ]
},
```

// Interview/Dialogue Template

```
interview: {
  titleFormat: "💬 AUFGABE {number}: {category}",
  includePrompt: true,
  questionCount: 5,
```


```



answerLines: 3,
format: "Frage-Antwort-Paare",
tips: [
  "Stell dir vor, du erzählst es einem Freund!",
  "Was würde jemand wissen wollen?"
]
}
};

```




4 BATCH-FUNKTIONEN FÜR LEHRKRÄFTE

A) Klassen-Generierung

Kreative Hausaufgaben für ganze Klasse	
Klasse: [3b ▼] 23 Schüler	
⚙️ GENERIERUNGS-OPTIONEN	
Aufgaben pro Schüler: [5 ▼]	
Nur Schüler mit Interessen-Profil:	
<input checked="" type="radio"/> Ja (18 Schüler) <input type="radio"/> Nein – generische Aufgaben für Rest (5 Schüler)	
Kategorien (optional einschränken):	
<input checked="" type="checkbox"/> Geschichten <input checked="" type="checkbox"/> Erfindungen <input type="checkbox"/> MINT	
<input checked="" type="checkbox"/> Kreativität <input type="checkbox"/> Philosophie <input checked="" type="checkbox"/> Perspektiven	
Schwierigkeitsgrad:	
<input type="radio"/> Alle gleich: [Mittel ▼] <input checked="" type="radio"/> Individuell angepasst (empfohlen)	
<hr/> <div>  AUSGABE-FORMAT </div>	

<input checked="" type="radio"/> Ein PDF pro Schüler (23 Dateien)	
<input type="radio"/> Ein großes PDF für alle (Sammeldokument)	
<input type="radio"/> ZIP-Download mit allen PDFs	
Layout: <input checked="" type="radio"/> Eine Aufgabe pro Seite	
<input type="radio"/> Kompakt (mehrere Aufgaben/Seite)	
 Vorschau für 3 zufällige Schüler]	
[ Generieren & Herunterladen]	

B) Thematische Wochen-Planung

 THEMEN-WOCHE PLANEN	
Woche vom: [20.10.2025 ] bis [26.10.2025 ]	
Klasse: [3b ▼]	
Hauptthema:	
<input checked="" type="radio"/> Herbst & Natur	
<input type="radio"/> Freundschaft	
<input type="radio"/> Erfinder & Entdecker	
<input type="radio"/> Fantasiewelten	
<input type="radio"/> Eigenes Thema: [_____]	
THEMA: Herbst & Natur	
Das System wird automatisch Aufgaben generieren, die:	
• Das Thema "Herbst" einbauen	
• ABER trotzdem individuelle Interessen berücksichtigen	

	Beispiel für Emma (Katzen-Fan):		
	"Deine Katze entdeckt den Herbstwald.		
	Was findet sie? Schreibe eine Geschichte."		
	Beispiel für Max (Fußball-Fan):		
	"Erfinde ein Herbst-		

	"Erfinde ein Herbst-Fußball-Spiel mit	
	Blättern und Kastanien. Welche Regeln?"	
Aufgaben-Verteilung:		
Montag:	[2] Aufgaben	[Leicht ▼]
Mittwoch:	[2] Aufgaben	[Mittel ▼]
Freitag:	[1] Aufgabe	[Herausfordernd ▼]
<input checked="" type="checkbox"/>	Variation innerhalb Thema (empfohlen)	
<input type="checkbox"/>	Alle Schüler bekommen identische Aufgaben	
[📊 Vorschau Aufgaben-Mix]		
[🌟 Woche generieren]		

5 INTELLIGENTE ERWEITERUNGS-FEATURES

A) Kombinations-Generator

Konzept: Mehrere Interessen eines Kindes clever kombinieren

typescript

```
function generateCombinedTask(profile: StudentInterestProfile): Task {  
  
    // Beispiel: Kind mag LEGO + Dinosaurier + Mathe  
  
    const interests = [  
        ...profile.hobbies, // ["LEGO"]  
        ...profile.favoriteAnimals, // ["Dinosaurier"]  
    ]  
}
```



```

    ...profile.favoriteSubjects // ["Mathe"]
  ];

  if (interests.length >= 2) {
    const [interest1, interest2] = randomSample(interests, 2);

    return {
      prompt: `Kombiniere ${interest1} und ${interest2}:
        Baue aus LEGO einen Dinosaurier.
        Miss ihn aus: Wie viele LEGO-Steine lang ist er?
        Wie viele Steine breit?
        Rechne aus: Wie viele Steine hast du insgesamt benutzt?`,
      category: "MINT-Förderung",
      matchScore: "ultra-high" // Weil mehrere Interessen kombiniert
    };
  }
}

```

Weitere Kombinations-Beispiele:

- **Fußball + Tiere:** "Erfinde ein Fußball-Turnier für Tiere. Welches Team gewinnt? Warum?"
- **Kochen + Farben:** "Erfinde ein Gericht, das komplett blau ist. Was sind die Zutaten?"
- **Tanzen + Weltraum:** "Du bist Astronautin und tanzt auf dem Mond. Wie bewegt man sich dort?"
- **Zeichnen + Freunde:** "Male Portraits von 3 Freunden als Superhelden. Was sind ihre Kräfte?"

B) Schwierigkeits-Anpassung basierend auf Profil

typescript

```

interface AdaptiveDifficulty {
  // System passt automatisch an:

  sentenceComplexity: 'simple' | 'medium' | 'complex';
  // Klasse 1-2: "Male einen Baum."
  // Klasse 5-6: "Visualisiere einen Baum als Symbol für..."
}

```

```
taskScope: 'narrow' | 'medium' | 'open';
```

```
// Narrow: "Schreibe 3 Sätze"
```

```
// Open: "Schreibe eine Geschichte (so lang du willst)"
```

```
abstractionLevel: number; // 1-5
```

```
// Level 1: Konkrete Dinge (Apfel, Auto)
```

```
// Level 5: Abstrakte Konzepte (Freundschaft, Zeit)
```

```
scaffolding: 'high' | 'medium' | 'low';
```

```
// High: Viele Tipps und Starthilfen
```

```
// Low: Nur Aufgabenstellung
```

```
}
```

```
function adjustDifficulty(
```

```
  template: TaskTemplate,
```

```
  student: StudentProfile
```

```
): string {
```

```
  let prompt = template.basePrompt;
```

```
  const age = calculateAge(student.birthDate);
```

```
  // Für jüngere Kinder (6-8):
```

```
  if (age <= 8) {
```

```
    // Konkretere Formulierung
```

```
    prompt = simplifyLanguage(prompt);
```

```
    // Mehr Struktur geben
```

```
    if (template.outputFormat === 'text') {
```

```
      prompt += "\n\nSchreibe mindestens 5 Sätze.";
```

```
    }
```

```
    // Scaffolding hinzufügen
```

```
    prompt += "\n\n💡 Tipp: Beginne mit 'Es war einmal...';
```

```
  }
```

```
  // Für ältere Kinder (11-12):
```

```
  if (age >= 11) {
```

```
    // Herausforderung erhöhen
```

```
    prompt = addComplexity(prompt);
```

```
// Reflexion einbauen
prompt += "\n\nDenke nach: Warum hast du dich so entschieden?";
}

return prompt;
}
```

C) Kontext-Sensitive Aufgaben

typescript

```
interface ContextualFactors {
  timeOfYear: 'spring' | 'summer' | 'autumn' | 'winter';
  currentCurriculumTopic?: string; // z.B. "Wir lernen gerade Bruchrechnen"
  upcomingEvent?: string; // z.B. "Schulfest nächste Woche"
  recentClassExperience?: string; // z.B. "Ausflug zum Zoo"
}

// Beispiel: Nach Zoo-Ausflug
const contextTemplate = {
  prompt: "Wir waren im Zoo. Dein Lieblingstier dort war {ANIMAL}."
  "Stell dir vor, du könntest einen Tag mit diesem Tier"
  "verbringen. Was würdest du fragen? Was würdet ihr machen?",
  context: "post-field-trip",
  emotionalRelevance: "high" // Weil frisches Erlebnis
};

// Beispiel: Vor Weihnachten
const seasonalTemplate = {
  prompt: "Die {ANIMAL} möchte jemandem ein Geschenk machen."
  "Sie hat aber kein Geld. Was könnte sie stattdessen schenken?"
  "(Vielleicht etwas Selbstgemachtes?)",
  context: "winter-season",
  values: ["creativity", "generosity"]
};
```


6 MONITORING & FEEDBACK-SYSTEM

A) Erfolgs-Tracking für Lehrkräfte

 AUFGABEN-ANALYTICS: Klasse 3b


Zeitraum: [Letzte 4 Wochen ▼]

PERSONALISIERTE vs. GENERISCHE AUFGABEN

 Mit Interessen-Profil: 18 Schüler (78%)

└ Ø Bearbeitungszeit: 22 Min

└ Rückmeldungen: "Hat Spaß gemacht!" (89%)

 Ohne Profil: 5 Schüler (22%)



└ Ø Bearbeitungszeit: 18 Min

└ Rückmeldungen: "War okay" (60%)

 Empfehlung: Interessen-Interview noch

durchführen bei: Tim, Sarah, Leon, Maria, Jan

BELIEBTESTE INTERESSEN-KOMBINATIONEN










1.  Tiere +  Zeichnen → 12 Schüler

2.  Sport +  Freunde → 8 Schüler


3.  Weltraum +  Forschen → 6 Schüler

4.  Kunst +  Musik → 5 Schüler

KATEGORIEN-PERFORMANCE

 Geschichten <div><div></div></div> 82% Abschluss
 Kreativität & Kunst <div><div></div></div> 88% Abschluss
 Erfindungen <div><div></div></div> 71% Abschluss
 Philosophie <div><div></div></div> 45% Abschluss
 Hinweis: Philosophie-Aufgaben evtl. zu abstrakt? → Konkretere Beispiele einbauen
<hr/>
PROFIL-QUALITÄT
 Vollständig (8+ Felder): 12 Schüler
 Basic (4-7 Felder): 6 Schüler
 Minimal (1-3 Felder): 0 Schüler
 Profile aktualisieren empfohlen] (6)

B) Schnelles Feedback-Erfassungs-Tool

 SCHNELL-FEEDBACK nach Hausaufgaben-Abgabe
Schüler: Emma Müller
Aufgabe: "Geschichte über Katze im Herbstwald"
<hr/>
Wie fand Emma die Aufgabe?
(Kurzes mündliches Gespräch)

Motivation: 😊😊😊😊😊 [Sehr motiviert]

Schwierigkeit: 🟢 Genau richtig

Dauer: [25] Minuten

☒ Hat Bezug zu Interessen erkannt

☐ Fand Aufgabe zu einfach

☐ Brauchte Hilfe

☒ Möchte mehr solche Aufgaben

Notizen (optional):

Emma hat sehr detailliert geschrieben und

sogar ein Ende mit Überraschung eingebaut!

[📁 Speichern] [→ Nächster Schüler]

Nutzen:

- System lernt, welche Aufgaben-Typen gut ankommen
- Warnt bei häufig "zu schwer" bewerteten Aufgaben
- Schlägt Anpassungen vor

7 SPEZIAL-MODULE FÜR VERSCHIEDENE SZENARIEN

A) "Ferien-Paket" Generator

🌈 FERIEN-HAUSAUFGABEN GENERIEREN

Ferien: [Herbstferien 2025 ▼]

Dauer: 2 Wochen

Philosophie:

☒ Leicht & spielerisch (Ferien-Feeling)

☐ Standard

☐ Fordernd (Vorbereitung auf neues Thema)

Aufgaben-Menge:

☐ Mini-Paket (3 Aufgaben)

☒ Standard-Paket (5 Aufgaben)

☐ Maxi-Paket (8 Aufgaben)

Besondere Ferien-Features:

☒ "Draußen-Aufgaben" (Natur beobachten, etc.)

☒ "Familien-Aufgaben" (mit Eltern/Geschwistern)

☒ "Urlaubs-Tagebuch" Option

☐ "Bücher-Empfehlungen" basierend auf Interessen

BEISPIEL-AUFGABE (Emma):

"Gehe mit deiner Familie in die Natur.

Sammele 7 verschiedene Herbstblätter.

Klebe sie auf und erfinde für jedes Blatt

eine kleine Geschichte: Wenn dieses Blatt

ein Tier wäre – welches? Warum?"

 Familien-Aktivität •  Draußen

[🌟 Paket für ganze Klasse generieren]

B) "Förder-Modus" für spezifische Kompetenzen

 GEZIELTE FÖRDERUNG

Schüler: Max Schmidt

Aktuelles Interessen-Profil: ☒ Vollständig

FÖRDER-SCHWERPUNKT WÄHLEN:

☒ Sprachliche Ausdrucksfähigkeit

☐ Selbstvertrauen / Selbstwirksamkeit

☐ Empathie / Soziale Kompetenzen

☐ Kreatives Problemlösen

☐ Konzentration / Ausdauer

☐ Feinmotorik (Schreiben/Zeichnen)

GEWÄHLT: Sprachliche Ausdrucksfähigkeit

Das System wird Aufgaben generieren, die:

• Wortschatz erweitern

• Beschreibende Sprache fördern

• Storytelling-Strukturen üben

• ABER: Verknüpft mit Max' Interessen!

Max mag: Fußball, Bauen, Autos

Beispiel-Aufgabe:

"Beschreibe dein Traum-Stadion so genau,
dass jemand es nachbauen könnte. Benutze
mindestens 10 verschiedene Eigenschafts-
wörter (groß, modern, bunt...)"

Aufgaben-Anzahl: [4 ▼]

Schwierigkeits-Steigerung: ☒ Ja (empfohlen)

[🌟 Förder-Aufgaben generieren]

C) "Projekt-Modus" für mehrteilige Aufgaben



PROJEKT-REIHE ERSTELLEN

Für: Klasse 3b (oder einzelne Schüler)

Projekt-Titel: [Mein eigenes Buch_____]

Dauer: [4] Wochen

Aufgaben pro Woche: [1-2] Aufgaben

PROJEKT-STRUKTUR (automatisch generiert):

Woche 1: PLANUNG

└ "Worum geht dein Buch? Wer sind die Figuren?"

Woche 2: INHALT ERSTELLEN

└ "Schreibe Kapitel 1: Der Anfang"

└ "Schreibe Kapitel 2: Ein Problem taucht auf"

Woche 3: AUSGESTALTEN

└ "Male Bilder zu deinem Buch"

Woche 4: ABSCHLUSS

└ "Gestalte ein Buchcover und Rückseiten-Text"

☒ An Interessen anpassen

└ Beispiel für Emma (Tiere):

"Schreibe ein Buch über eine Tierärztin"

☒ Zwischen-Feedback einplanen

☒ Abschluss-Präsentation vorbereiten

[🌟 Projekt generieren]

8 DATENBANK-SCHEMA ERWEITERUNGEN

sql

```
-- Interessen-Profile
CREATE TABLE student_interest_profiles (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  student_id INTEGER UNIQUE REFERENCES students(id),

  -- Basis-Interessen (JSON Arrays als TEXT)
  favorite_animals TEXT,      -- ["Katze", "Einhorn", "Delfin"]
  favorite_colors TEXT,      -- ["Blau", "Gold"]
  favorite_activities TEXT,   -- ["Fußball", "Zeichnen"]
  favorite_subjects TEXT,    -- ["Mathe", "Kunst"]

  -- Träume & Wünsche
  dream_job TEXT,             -- "Tierärztin"
  dream_place TEXT,           -- "Strand"
  super_power TEXT,           -- "Fliegen"

  -- Hobbies & Expertise
  hobbies TEXT,               -- ["LEGO", "Tanzen"]
  collections TEXT,           -- ["Steine", "Pokemon"]
  expertise TEXT,             -- ["Dinosaurier"]

  -- Präferenzen (BOOLEAN als 0/1)
  likes_fantasy INTEGER DEFAULT 1,
  likes_realism INTEGER DEFAULT 1,
  likes_humor INTEGER DEFAULT 1,
  likes_challenge INTEGER DEFAULT 0,

  -- Soziales
```

```

    prefers_alone INTEGER DEFAULT 0,
    prefers_partner INTEGER DEFAULT 1,
    prefers_group INTEGER DEFAULT 0,

    -- Kreative Ausdrucksformen (JSON Array)
    expression_modes TEXT,          -- ["Zeichnen", "Schreiben"]

    -- Metadaten
    attention_span TEXT DEFAULT 'medium', -- 'short'|'medium'|'long'
    interview_date TIMESTAMP,
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    completeness INTEGER DEFAULT 0,      -- 0-100

    FOREIGN KEY (student_id) REFERENCES students(id) ON DELETE CASCADE
);

-- Task-Templates für Generator
CREATE TABLE task_templates (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    title TEXT NOT NULL,
    category TEXT NOT NULL,
    base_prompt TEXT NOT NULL,

    -- Variablen (JSON Array)
    variables TEXT,          -- ["ANIMAL", "HOBBY", "COLOR"]

    -- Anforderungen (JSON Arrays)
    required_interests TEXT,  -- ["favoriteAnimals"]
    optional_interests TEXT,  -- ["dreamPlace", "favoriteColors"]

    -- Eigenschaften
    difficulty INTEGER DEFAULT 2,
    estimated_minutes INTEGER DEFAULT 20,
    output_format TEXT DEFAULT 'text', -- 'text'|'drawing'|'mixed'

    -- Scaffolding (JSON)
    tips TEXT,               -- ["Tipp 1", "Tipp 2"]
    example TEXT,            -- Optionales Beispiel

```

```

-- Metadaten
min_age INTEGER DEFAULT 6,
max_age INTEGER DEFAULT 12,
is_active INTEGER DEFAULT 1,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Generierte personalisierte Aufgaben (Log)
CREATE TABLE generated_tasks (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  student_id INTEGER REFERENCES students(id),
  template_id INTEGER REFERENCES task_templates(id),

-- Finaler Text
generated_prompt TEXT NOT NULL,

-- Verwendete Werte
used_interests TEXT,          -- JSON: {"ANIMAL": "Katze", "HOBBY":
"Tanzen"}

-- Qualität
match_score INTEGER,          -- 0-100

-- Status
was_assigned INTEGER DEFAULT 0,
assignment_id INTEGER REFERENCES student_creative_tasks(id),

generated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

FOREIGN KEY (student_id) REFERENCES students(id) ON DELETE CASCADE
);

-- Feedback & Analytics
CREATE TABLE task_feedback (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  student_id INTEGER REFERENCES students(id),
  task_assignment_id INTEGER REFERENCES student_creative_tasks(id),

-- Schnell-Feedback

```

```

motivation_level INTEGER,      -- 1-5 (Smileys)
difficulty_rating TEXT,        -- 'too_easy' | 'just_right' | 'too_hard'
completion_time INTEGER,      -- Minuten

-- Checkboxen
recognized_interest INTEGER DEFAULT 0,
found_too_easy INTEGER DEFAULT 0,
needed_help INTEGER DEFAULT 0,
wants_more INTEGER DEFAULT 0,

-- Notizen
teacher_notes TEXT,

feedback_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

FOREIGN KEY (student_id) REFERENCES students(id) ON DELETE CASCADE,
FOREIGN KEY (task_assignment_id) REFERENCES student_creative_tasks(id) ON
DELETE CASCADE
);

-- Projekt-Reihen
CREATE TABLE creative_projects (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  title TEXT NOT NULL,
  description TEXT,
  duration_weeks INTEGER DEFAULT 4,
  created_by INTEGER REFERENCES users(id),

  -- Kann für ganze Klasse oder einzelne Schüler sein
  class_id INTEGER REFERENCES classes(id),

  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

  FOREIGN KEY (created_by) REFERENCES users(id) ON DELETE SET NULL
);

CREATE TABLE project_phases (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  project_id INTEGER REFERENCES creative_projects(id),

```

```

    phase_number INTEGER NOT NULL,
    phase_title TEXT NOT NULL,      -- "Planung", "Inhalt erstellen"
    task_template_id INTEGER REFERENCES task_templates(id),

    week_number INTEGER,

    FOREIGN KEY (project_id) REFERENCES creative_projects(id) ON DELETE
    CASCADE
);

CREATE TABLE student_project_progress (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    student_id INTEGER REFERENCES students(id),
    project_id INTEGER REFERENCES creative_projects(id),

    current_phase INTEGER DEFAULT 1,
    completed_phases TEXT,          -- JSON Array [1, 2, 3]

    started_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    completed_at TIMESTAMP,

    FOREIGN KEY (student_id) REFERENCES students(id) ON DELETE CASCADE,
    FOREIGN KEY (project_id) REFERENCES creative_projects(id) ON DELETE
    CASCADE
);

```

9 API-ENDPOINTS IMPLEMENTIERUNG

typescript

```

// =====
// INTERESSEN-PROFIL MANAGEMENT
// =====

// Profil erstellen/updaten
app.post('/api/students/:id/interest-profile', async (req, res) => {
    const { studentId } = req.params;

```

```
const profileData = req.body;
```

```
// Berechne Vollständigkeit
```

```
const completeness = calculateCompleteness(profileData);
```

```
const profile = await db.insert(studentInterestProfiles).values({  
  studentId,  
  ...profileData,  
  completeness,  
  lastUpdated: new Date()  
}).onConflictDoUpdate({  
  target: studentInterestProfiles.studentId,  
  set: { ...profileData, completeness, lastUpdated: new Date() }  
});
```

```
res.json({ success: true, profile, completeness });  
});
```

```
// Profil abrufen
```

```
app.get('/api/students/:id/interest-profile', async (req, res) => {  
  const profile = await db.query.studentInterestProfiles.findFirst({  
    where: eq(studentInterestProfiles.studentId, req.params.id)  
  });
```

```
  res.json(profile || { exists: false });  
});
```

```
// =====
```

```
// AUFGABEN-GENERATOR
```

```
// =====
```

```
// Personalisierte Aufgaben generieren
```

```
app.post('/api/students/:id/generate-tasks', async (req, res) => {  
  const { studentId } = req.params;  
  const { count = 5, categoryFilter, difficultyLevel, thematicFocus } =  
  req.body;
```

```
// 1. Profil laden
```

```
const profile = await getInterestProfile(studentId);
```

```

    if (!profile || profile.completeness < 30) {
      return res.json({
        error: 'PROFILE_INCOMPLETE',
        message: 'Interessen-Profil fehlt oder unvollständig',
        genericTasks: await getGenericTasks(count, categoryFilter)
      });
    }

    // 2. Passende Templates finden
    const templates = await findMatchingTemplates(profile, {
      category: categoryFilter,
      difficulty: difficultyLevel,
      theme: thematicFocus
    });

    // 3. Aufgaben generieren
    const generatedTasks = templates.slice(0, count).map(template => {
      const task = generateTaskFromTemplate(template, profile);

      // In DB loggen
      db.insert(generatedTasks).values({
        studentId,
        templateId: template.id,
        generatedPrompt: task.description,
        usedInterests: JSON.stringify(task.usedValues),
        matchScore: task.matchScore,
        generatedAt: new Date()
      });

      return task;
    });

    res.json({
      tasks: generatedTasks,
      profileCompleteness: profile.completeness,
      matchQuality: calculateAverageMatch(generatedTasks)
    });
  });
}

```



```
// Aufgabe direkt zuweisen (aus Generator)
app.post('/api/students/:id/assign-generated-task', async (req, res) => {
  const { studentId } = req.params;
  const { generatedTaskId, teacherNotes } = req.body;

  // Generierte Aufgabe laden
  const genTask = await db.query.generatedTasks.findFirst({
    where: eq(generatedTasks.id, generatedTaskId)
  });

  // Als creative task speichern
  const assignment = await db.insert(studentCreativeTasks).values({
    studentId,
    taskId: null, // Kein vordefinierter Task
    generatedTaskId: genTask.id,
    description: genTask.generatedPrompt,
    assignedBy: req.user.id,
    notes: teacherNotes,
    createdAt: new Date()
  });

  // Generator-Log updaten
  await db.update(generatedTasks)
    .set({ wasAssigned: true, assignmentId: assignment.id })
    .where(eq(generate
```