# Plan Strategic și Roadmap de Produs V17: Proiect "Sparky's Workshop Factory"

**Versiune: 17.3** **Data Ultimei Actualizări:** 28 iunie 2025

## Partea I-XII

*(Neschimbate)*

## Partea XIII: Specificații de Produs pentru Versiunea 3.0+ (Partenerul Strategic AI)

*(Neschimbată)*

## Partea XIV: Arhitectura Backend pentru Versiunea 1.0 (MVP)

Această secțiune descrie fluxul de date și logica de procesare pe server, de la primirea cererii API până la returnarea pachetului final.

### 14.1. Endpoint-ul API Principal: /generate-workshop

* **Tehnologie:** O Google Cloud Function (Python) sau un container pe Cloud Run, expus prin API Gateway-ul nostru.
* **Metodă:** POST
* **Input:** Primește un obiect JSON care reprezintă workshop-ul, exact cum este trimis de aplicația React.  
  {  
   "title": "Atelierul Roboților Prietenoși",  
   "theme": "Roboți și Tehnologie",  
   "target\_age": "3-5 ani",  
   "components": [  
   { "id": 1, "type": "Pagină de Colorat", "subTheme": "Robot vesel" },  
   { "id": 2, "type": "Fișă de Trasare", "subTheme": "Litere R-O-B-O-T" }  
   ]  
  }
* **Output (Succes):** Un obiect JSON cu URL-ul de descărcare.  
  { "downloadUrl": "https://storage.googleapis.com/..." }
* **Output (Eroare):** Un cod de eroare (ex: 400, 500) și un mesaj JSON.  
  { "message": "A apărut o eroare la generarea ghidului." }

### 14.2. Funcția de Orchestrare: buildWorkshopPackage(workshop\_data)

Aceasta este funcția principală, "dirijorul" care se activează la primirea unei cereri pe endpoint.

**Flux de Execuție:**

1. **Validare Date:** Verifică dacă datele primite (workshop\_data) sunt complete și au formatul corect. Dacă nu, returnează o eroare 400 (Bad Request).
2. **Salvare Inițială în Baza de Date:** Creează o nouă intrare în tabelul workshops din baza de date cu starea pending (în curs de procesare). Acest lucru ne permite să urmărim toate cererile.
3. **Generarea Conținutului Scris (Apel AI):**
   * Construiește **Promptul Master** (definit în Partea VII) injectând datele din workshop\_data.
   * Apelează funcția generate\_text(prompt) din modulul ai\_services.py (pe care am proiectat-o în Faza 1).
   * Primește înapoi un obiect JSON masiv cu textul pentru ghid și marketing.
   * Salvează acest JSON în câmpul guide\_content\_json al înregistrării din baza de date.
4. **Generarea Componentelor PDF:**
   * Iterează prin lista de components din workshop\_data.
   * Pentru fiecare componentă, generează PDF-ul corespunzător.
     + **Exemplu pentru "Pagină de Colorat":**
       - Construiește un prompt de imagine: Un robot prietenos, stil carte de colorat pentru copii, contururi simple și clare.
       - Apelează funcția generate\_image(prompt, ...) din ai\_services.py.
       - Funcția generate\_image returnează URL-ul imaginii salvate pe Cloud Storage.
       - Folosește o bibliotecă Python (precum reportlab sau fpdf2) pentru a crea un fișier PDF care conține imaginea generată.
   * Salvează fiecare PDF generat temporar pe server.
5. **Generarea Ghidului PDF:**
   * Folosind aceeași bibliotecă PDF, creează Ghidul\_Atelierului.pdf.
   * Structurează documentul folosind textul extras din guide\_content\_json (introducere, instrucțiuni, etc.).
6. **Asamblarea Arhivei .ZIP:**
   * Creează o nouă arhivă .ZIP.
   * Adaugă Ghidul\_Atelierului.pdf în arhivă.
   * Creează un folder Materiale\_Printabile/ în arhivă și adaugă toate PDF-urile componentelor generate la pasul 4.
7. **Stocarea Finală și Actualizarea Bazei de Date:**
   * Încarcă arhiva .ZIP finală în bucket-ul dedicat de pe Google Cloud Storage.
   * Obține URL-ul public (sau semnat, pentru securitate) al fișierului încărcat.
   * Actualizează înregistrarea din tabelul workshops cu starea completed și salvează URL-ul final în câmpul corespunzător.
8. **Returnarea Răspunsului:**
   * Trimite înapoi către frontend un răspuns JSON de succes, care conține URL-ul final de descărcare.