

# Epic 2.16: Product Requirements Document (PRD)

## Presentation Generator for MarvelAI

*Date: February 25, 2025*

---

### 1. Overview

The Presentation Generator is an AI-powered tool within MarvelAI designed to assist educators in creating text-based presentation slides. Users input text context, specify the number of slides, and select an instructional level. The tool generates an editable outline, produces text-based slides, allows for basic editing, and supports export to multiple formats. This version prioritizes simplicity and text-based functionality, with image-based slides deferred to a later phase.

**Target Users:** Educators (e.g., teachers, professors, instructional designers).

**Objective:** Streamline presentation creation with AI-driven text content generation and basic slide editing.

---

### 2. User Flow

The tool follows a straightforward process:

1. **Input Collection:** Users provide text context, number of slides, and instructional level.
  2. **Outline Generation:** AI generates a list of slide topics based on inputs.
  3. **Outline Editing:** Users review and modify the outline (add, remove, update topics).
  4. **Slide Generation:** AI creates text-based slides from the approved outline.
  5. **Slide Editing:** Users edit existing slides or add new text-based slides.
  6. **Export:** Users export the final presentation in a chosen format (PDF, Google Slides, PPTX).
- 

### 3. Key Features

#### 3.1 Input Collection

- **Text Input:** Users enter context (e.g., subject, topic, description).

- Examples: "World War II overview," "Introduction to algebra."
- **Number of Slides:** Dropdown menu to select the number of slides (e.g., 5–20).
- **Instructional Level:** Dropdown menu to choose the educational level (e.g., Elementary, High School, University).
- **Generate Outline Button:** Triggers outline generation based on inputs.

**Exclusions:** No image uploads or language selection options.

### 3.2 Outline Generation

- **AI-Driven Outline:** AI generates a list of slide topics (e.g., "Key Inventions of the Industrial Revolution").
- **Display:** Outline is presented for user review in a numbered list format.

### 3.3 Outline Editing

- **Edit Options:** Users can:
  - Add new slide topics.
  - Remove existing topics.
  - Update topic titles.
- **Approval:** Users confirm the outline to proceed to slide generation.

### 3.4 Slide Content Generation

- **Text Content:** AI generates text (e.g., titles, bullet points) for each slide, tailored to the instructional level.
- **Templates:** AI applies simple text-based templates (e.g., title slide, bullet list).
- **Output:** Slides are rendered for editing within the platform.

### 3.5 Slide Editing

- **Core Functionality:**
  - Edit text on existing slides (e.g., modify titles, bullet points).
  - Add new text-based slides.

**Preview:** Real-time updates display changes as users edit.

- 

**Focus:** Initial development prioritizes editing existing slides and adding new slides; advanced features deferred.

### 3.6 Export Options

- **Formats:** Export options include:
    - PDF
    - Google Slides (via API)
    - PPTX
  - **Interface:** Simple dropdown or button selection for exporting the final presentation.
- 

## 4. Technical Recommendations

Teams have flexibility to choose their implementation approach, with the following suggestions based on the existing MarvelAI tech stack:

### 4.1 Frontend (marvel-platform)

- **Tech Stack:**
  - Next.js (v12.3.0) and React (v18.2.0) for UI development.
  - Redux Toolkit for state management.
  - Emotion and MUI for styling.
  - Firebase for authentication and data persistence.
- **Options:**
  - Build from scratch using React components and Firebase integration.
  - Use libraries like Slidev or Reveal.js for slide rendering.
  - Implement PDF exports with jsPDF.

### 4.2 Backend (marvel-ai-backend)

- **Tech Stack:**
  - FastAPI (Python 3.10.12) for API endpoints.
  - Google Cloud Vertex AI and Gemini 1.0 for AI content generation.

- **Options:**

- Develop custom AI logic for text generation from scratch.
- Leverage existing Google Cloud AI services for faster implementation.

#### 4.3 General Considerations

- **Modular Design:** Integrate as a tool under `/tools` (Frontend) and `app/tools` (Backend) directories.
- **Export Tools:** Use jsPDF, Google Slides API, or PptxGenJS for export functionality.

Teams can mix and match these ideas or explore alternative frameworks/libraries as needed.

---

### 5. Resources

The following resources are recommended for team reference:

- [gamma.app](#): Inspiration for design and feature benchmarking.
  - **Frontend Libraries:**
    - Next.js, React, Redux Toolkit, Emotion, MUI, Firebase SDK.
    - [Slidev](#), [Reveal.js](#), jsPDF (for slide rendering and exports).
  - **Backend Libraries:**
    - FastAPI, Python, Google Cloud SDK (Vertex AI, Gemini).
    - PptxGenJS (for PPTX exports).
  - **Documentation:**
    - MarvelAI frontend (marvel-platform) and backend (marvel-ai-backend) repos.
    - Firebase and Google Cloud service guides.
- 

### 6. Future Enhancements

After implementing the text-based functionality outlined above, the next phase will include:

- **Image Generation:** AI-powered creation of images to complement slide content.
- **Image-Based Slide Templates:** Support for visually rich templates incorporating images.

These enhancements will build on the initial text-based foundation to improve presentation engagement and visual appeal.

---