Assignment: AI-Assisted Document Authoring and Generation Platform

Objective

Design and implement a full-stack, AI-powered web application that allows authenticated users to generate, refine, and export structured business documents. Users should be guided through:

- Selecting a document type (Microsoft Word `.docx` or PowerPoint `.pptx`)
- Defining the structure and outline of the document
- Generating content using a Large Language Model (LLM) such as the Gemini API
- Iteratively refining the generated content through an interactive interface
- Exporting the final document in `.docx` or `.pptx` format

Important to Note:
The backend must be built using FastAPI or Flask, handling authentication, LLM calls, and document assembly.
The frontend must be a responsive web interface (React, Vue, or standard HTML/CSS/JS).
All user and project data must be stored in a database (e.g., Firestore, PostgreSQL, SQLite).

Evaluation Criteria

Functionality

- End-to-end flow works:
  Login → Configure → Generate → Refine → Export

All required features are fully implemented.

- AI Integration
- LLM used effectively for:
- Initial content generation
- Iterative refinement

(Optional) Outline/slide-title template generation

User Experience

- UI is clear, responsive, and intuitive.
- The refinement process should feel seamless and efficient.

Output Quality

- Exported `.docx` and `.pptx` files:

- Are well-formatted.
- Accurately reflect refined content

Code Quality

- Clean, modular, readable code
- Logical folder structure
- Good use of best practices for FastAPI/Flask and chosen frontend framework

Documentation

A comprehensive `README.md` with:

- Setup instructions
- Environment variables
- Run/deployment instructions.
- Usage examples

Submission Guidelines

Candidates must provide:

- Source Code Repository (GitHub/GitLab/etc.)
- README.md containing:
- Installation & setup steps
- Environment variable descriptions
- How to run backend & frontend
- Demo Video (5–10 minutes) showing:
- User registration & login
- Configuring a Word document
- Configuring a PowerPoint document
- Content generation
- Refinement (AI edits, like/dislike, comments)
- Exporting `.docx` and `.pptx` files
- (Optional) AI-Generated Template workflow

Functional Requirements

1. User Authentication & Project Management

- Implement secure user registration and login (Firebase Auth or JWT-based system).
- Provide a dashboard displaying all existing user projects and an option to create a new project.
- Each project must store:

- Document configuration
- Generated content
- Refinement history (AI edits, feedback, comments)

2. Document Configuration (Scaffolding)

When creating a new project, the system must:

1. Ask the user to choose a document type:
   - Microsoft Word (.docx)
   - Microsoft PowerPoint (.pptx)
2. Ask for a main topic or prompt (e.g., *"A market analysis of the EV industry in 2025"*).
   Based on the selected format:
   - If `.docx`:
     Allow users to create an outline (add, remove, reorder section headers).
   - If `.pptx`:
     Allow users to define the number of slides and specify a title/heading for each slide.

3. AI-Powered Content Generation

- After configuration, the backend must generate content section-by-section (or slide-by-slide) using an LLM.
- Each LLM call should be context-aware and specific to that section/slide.
- All generated text must be stored in the database and linked to the project.

4. Interactive Refinement Interface

After generation, users must have an editor-style interface that displays the document structure.

1. For each section/slide, provide:
   - AI Refinement Prompt:
     A textbox that sends a refinement request for only that section (e.g., "Make this more formal," "Convert to bullet points," "Shorten to 100 words").
   - Feedback Buttons:
     Like / *Dislike* to record user satisfaction.
   - Comment Box:
     For user notes, stored in the database.

All revisions, prompts, and comments must be persisted.

5. Document Export

1. Provide a Download/Export button.

   Backend should:

   - Fetch the latest refined content
   - Assemble a valid `.docx` or `.pptx` file using libraries such as `python-docx` or `python-pptx`
   - Return the file for download

Bonus Feature (Optional – Extra Points)

AI-Generated Templates

- During configuration, allow users to click “AI-Suggest Outline.”
- User provides main topic only.
- System generates:
- Section headers (for Word projects), or
- Slide titles (for PowerPoint projects)
- User may accept, edit, or discard the generated template.