PROJECT REPORT

On

AI PPT GENERATOR

Submitted in the partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

In

Computer Science & Artificial Intelligence

Batch (2022-2026)

Submitted to:

Dr.Ridhi Kapoor

Submitted by:

Amritpal Singh
12201104
B49A

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DAV UNIVERSITY

JALANDHAR-PATHANKOT NATIONAL HIGHWAY,
NH44, SARMASTPUR
PUNJAB-144012

DECLARATION

I, Amritpal Singh hereby declare that the work which is being presented in this project titled “AI PPT GENERATOR” by me, in partial fulfillment of the requirements for the award of Bachelor of Technology (B. Tech) Degree in “Computer Science & Artificial Intelligence” is an authentic record of my own work carried out under the guidance of DR. Ridhi Kapoor To the best of my knowledge, the matter embodied in this report has not been submitted to any other University/ Institute for the award of any degree or diploma.

Amritpal Singh
12201104

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project guide, Dr. Ridhi Kapoor, for their valuable guidance, constructive feedback, and continuous support throughout the duration of this project. I am also thankful to Dr. Rahul Hans and the Department of engineering for providing the necessary resources and an encouraging academic environment. My appreciation extends to all faculty members and classmates whose suggestions and cooperation helped strengthen the quality of this work. Lastly, I am deeply grateful to my family for their constant encouragement and support, which motivated me to complete this project successfully.

ABSTRACT

The AI-Based Automated Presentation Generator is an innovative web application designed to simplify and accelerate the process of creating professional presentations. The system uses Gemini AI to generate structured outlines and slide-level content from a user-provided topic, while ImageKit AI produces relevant visuals to enhance slide quality. React and Tailwind CSS provide a dynamic, real-time editing interface where users can modify content and layout instantly. Firebase Firestore stores user projects, outlines, images, and metadata, ensuring seamless access across sessions. Clerk Authentication manages secure login and subscription-based feature access. By automating content generation, slide design, and image creation, the system eliminates the repetitive and time-consuming aspects of traditional presentation building. This project demonstrates how modern AI and cloud technologies can work together to deliver an intuitive, efficient, and highly scalable solution for students, educators, professionals, and content creators.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Background of the Study

In recent years, advancements in Artificial Intelligence (AI) have transformed the way digital content is produced, optimized, and consumed. Tasks that traditionally required extensive human effort—such as writing documents, designing presentations, generating images, and preparing reports—can now be automated using intelligent systems. Among these, the automation of presentation design has become increasingly important, as presentations play a critical role in education, business communication, training, research dissemination, and corporate decision-making.

Creating a visually appealing and well-structured presentation manually is time-consuming and often requires domain knowledge, design skills, and familiarity with presentation tools such as Microsoft PowerPoint, Google Slides, or Canva. Students, professionals, and educators often struggle to convert raw ideas into well-designed slides, especially when they lack design experience or when time constraints are strict. This gap has created a strong need for an intelligent system that can automatically generate high-quality presentation slides from simple user input.

To address this challenge, modern AI tools integrate natural language processing (NLP), layout generation, image generation, and UI rendering to produce complete presentations with minimal manual effort. The availability of powerful models like Google Gemini, cloud-based storage systems like Firebase, authentication frameworks such as Clerk, and image processing services like ImageKit.io has enabled the development of intelligent and efficient slide-generation systems. By combining these technologies, a fully automated pipeline can be created that transforms a user's topic into a complete, visually appealing, and editable presentation.

This major project, titled "AI-Powered Automated Presentation Generator Using React, Firebase, Gemini AI, and ImageKit.io", focuses on building such an end-to-end system. The system converts user input into slide outlines, slide content, HTML/Tailwind-based slide layouts, and AI-generated images, offering a seamless workflow from idea to complete presentation.

1.2 Problem Statement

Traditional presentation creation involves brainstorming content, structuring ideas, designing layouts, selecting color schemes, and adding visual elements. These tasks require considerable time and design experience. Many users, particularly students and non-designers, find it challenging to produce professional-quality presentations quickly.Key issues include:

- Lack of automation in structuring slide content
- Time-consuming manual design
- Difficulty maintaining consistent formatting
- Limited accessibility to high-quality design tools

- Dependence on paid software for professional templates

1.3 Need for the Project

As digital communication becomes more prominent in both academic and professional settings, the demand for tools that improve productivity continues to grow. Presentation creation is one of the most frequently performed tasks, yet it is often repetitive and inefficient. An AI-powered presentation generator provides:

- Rapid slide generation
- Professional and consistent designs
- Editable and customizable output
- Reduced manual effort
- Accessibility for users without design skills

This project addresses these needs by combining AI-driven text generation, layout design, and automated image creation into a single integrated platform.

1.4 Scope of the Project

The scope of this project includes the development of a complete web-based system capable of:

- User authentication using Clerk
- Project creation and management
- Automatic outline generation via Gemini AI
- Slide content and layout generation using HTML and TailwindCSS
- AI-based image generation using ImageKit
- Project storage and retrieval using Firebase Firestore
- Real-time editing of generated slides

The system also supports free and paid user tiers, where free users have limited project access while paid subscribers enjoy full features. Future enhancements include export to PPT/PDF, real-time collaboration, and multilingual support.

1.5 Overview of the System

The AI-powered presentation generator functions through the following workflow:

- User inputs topic and selects slide count.
- Gemini AI generates slide titles and outlines.
- User reviews and edits the outline.
- The system generates structured slide content and layouts.
- Images relevant to the presentation are generated through ImageKit.
- A slide editor allows users to modify text, regenerate layouts, and update visuals.

- All data is stored and synced using Firebase Firestore..

1.6 Significance of the Project

This project demonstrates the integration of cutting-edge technologies including generative AI, frontend engineering, cloud databases, and real-time authentication systems. The automated presentation generator benefits students, educators, corporate professionals, and content creators by drastically reducing time and effort required to create visually compelling presentations.

Its significance lies in democratizing access to professional design tools and enabling users to generate polished presentations without prior design expertise.

CHAPTER 2
PROJECT OVERVIEW

2.1 Introduction to the Project

In the modern digital era, presentations constitute a powerful medium of communication across educational, corporate, and research domains. Whether it is a seminar, pitch deck, classroom lecture, business meeting, or project showcase, presentations play a pivotal role in delivering information concisely and visually. However, designing effective presentations requires significant time, creativity, and proficiency with tools such as Microsoft PowerPoint, Google Slides, or Canva. Many users, particularly those without design skills, find the creation of professional presentations challenging.

This project—"AI-Powered Automated Presentation Generator Using React, Firebase, Gemini AI, Clerk, and ImageKit.io"—aims to solve this challenge by automating the complete process of slide creation. The system converts a single user-provided topic into a fully structured, visually appealing, and editable set of presentation slides. With technologies such as generative AI, cloud computing, real-time databases, and modern web frameworks, the platform provides an end-to-end solution to generate professional presentations with minimal manual effort.

2.2 Project Title

AI-Powered Automated Presentation Generator Using React, Firebase, Gemini AI, Clerk, and ImageKit.io. The title captures the essence of the project by highlighting both its functionality (automated presentation generation) and the major technologies involved.

2.3 Abstract

The purpose of this project is to develop a web-based platform that automatically generates complete presentation slides using Artificial Intelligence. The system takes a user-provided topic and produces slide outlines, content, visual layouts, and relevant images. The project leverages Google Gemini for content and layout generation, Firebase for real-time database management, Clerk for authentication, and ImageKit.io for AI-based image creation and optimization.

Using a step-by-step workflow, the system generates slide outlines, transforms them into detailed slide content, produces HTML and TailwindCSS layouts, and finally delivers fully editable slides to the user. Users can modify slide text, regenerate images, rearrange content, and store multiple presentation projects in their workspace. The interface is built using React, providing a responsive and user-friendly environment.

The system includes subscription-based access control, where free users can create a limited number of projects, while paid users gain access to unlimited generation capabilities. The model demonstrates how AI can simplify creative tasks and enhance productivity in academic, professional, and corporate environments.

2.4 Problem Domain

Creating presentations manually requires expertise in content organization, visual design, and formatting. Users often face challenges such as time constraints, lack of design skills, and dependence on expensive tools. These factors create a need for automated presentation generation tools.The problem domain covers:

- Natural Language Processing
- Automated Design and Layout
- Image Generation
- User-Friendly
- Web Interface Development

2.5 Project Motivation

The motivation behind this project arises from the growing trend of AI-powered automation and the increasing demand for productivity-enhancing tools. With rising workloads, academic pressures, and corporate competitiveness, individuals and organizations seek tools that reduce time spent on routine tasks.

Key motivational factors include:

- The need for a quick and efficient method of producing structured presentations
- The desire to eliminate design-related barriers for non-designers
- The increasing adoption of AI tools to automate creative tasks
- The potential to reduce the cost of using premium design platform
- The aim to assist students and professionals in producing effective communication materials

The idea for this project also stems from observing real-world challenges: many students struggle with preparing presentations for assignments, seminars, and project reviews. Similarly, educators and corporate employees often need multiple presentations within short timeframes. This system helps bridge that gap by offering fast, intuitive, and reliable slide generation.

2.6 Key Features of the System

- AI-based outline generation
- Automatic slide content creation
- HTML/TailwindCSS slide layout generation

- AI image generation using ImageKit.io
- Interactive slide editor
- Firebase storage and sync
- Clerk authentication and subscription management
- Project workspace with unlimited edit options

2.7 Use Cases

- Students preparing academic presentations
- Teachers creating lecture slides
- Corporate professionals developing pitch decks
- Researchers summarizing findings
- Startups showcasing product ideas
- Trainers and educators designing workshop content

2.8 Expected Outcomes

The system aims to deliver a complete AI-powered presentation generation platform capable of producing visually appealing slides, generating relevant images, storing projects securely, and providing seamless user interaction. The platform will simplify the process of creating structured and professional presentations.

2.9 Summary

This chapter explained the project’s title, purpose, motivation, features, and applicability. The AI-based presentation generator integrates advanced technologies to reduce manual effort while maintaining professional quality. The next chapter focuses on the project objectives and system requirements.

CHAPTER 3

OBJECTIVES OF THE PROJECT

3.1 Introduction

Every software project begins with a clear set of objectives that define what the system aims to achieve and how it will function. The objectives serve as the guiding principles behind the design, development, and implementation of the system. In the case of the AI-Powered Automated Presentation Generator, the primary goal is to automate the creation of visually appealing, structured, and professional presentation slides using Artificial Intelligence and modern web technologies.

This chapter outlines the main objectives, functional requirements, non-functional requirements, and expected deliverables of the system. These objectives help in building a system that is reliable, user-friendly, scalable, and capable of solving real-world problems related to presentation creation.

3.2 Main Objectives of the Project

The major objectives of the AI-powered presentation generator are:

- To automate the entire presentation creation process using AI.
- To generate structured slide outlines that reflect the user’s topic.
- To automatically produce visually appealing slide layouts using HTML and Tailwind CSS.
- To integrate AI-based image generation using Image-Kit.io.
- To provide a real-time editable slide workspace.
- To store and manage user projects securely using Firebase.
- To ensure safe access control using Clerk authentication.
- To support free and premium user tiers with feature restrictions.

3.3 Functional Objectives

The project must be able to perform the following essential functions:

- Accept input from the user – The system should allow users to enter a topic and select the number of slides. This input forms the core context for AI processing.
- Generate slide outlines using AI – Gemini AI must produce a structured outline with slide titles and bullet points based on the provided topic.
- Generate slide content and layout – The system should transform outlines into fully formatted HTML/Tailwind slides, including headings, body text, and layout structure.

- Generate images for the slides – Using ImageKit.io, the system should automatically generate relevant and high-quality images that match the slide content.
- Allow real-time editing – Users must be able to edit text, regenerate slides, replace images, and update layouts instantly through a React-based editor.
- Manage user projects – The system should allow creating, saving, editing, retrieving, and deleting projects, all stored securely in Firebase.
- Support authentication and subscription logic – Clerk should handle login, identity verification, and access levels for free and paid users to enforce feature limits.

3.4 Non-Functional Objectives

Non-functional requirements define the quality attributes and overall behavior of the system. These include:

- Usability – The system interface must be easy to use, intuitive, and require no technical expertise. Navigation, editing, and project management should feel seamless.
- Performance – The system should respond quickly, especially during AI calls for outline generation, slide creation, and image generation.
- Reliability – The application must run smoothly without failures. Firebase ensures data consistency and minimal downtime.
- Scalability – The platform must support expansion, allowing more users, more projects, and higher AI usage without affecting performance.
- Security – User data, projects, and subscription details must be protected using Clerk authentication and Firebase security rules.
- Cross-platform accessibility – The system should work on different devices and browsers, ensuring accessibility on Windows, macOS, Linux, and mobile platforms.
- Maintainability – The codebase must be modular and clean so future updates, features, and bug fixes can be implemented easily.

3.5 System Requirements

3.5.1 Software Requirements

- Windows / macOS / Linux
- Browser: Chrome, Firefox, Edge
- VS Code
- React
- Firebase
- Gemini AI

- Clerk Authentication
- TailwindCSS
- Node.js

3.5.2 Hardware Requirements

- Minimum 4 GB RAM
- Intel i3 or equivalent processor
- Stable internet connection

3.6 Project Deliverables

- A fully functional web application
- User authentication and subscription system
- AI-based outline generation module
- Slide content generator
- HTML/TailwindCSS slide design generator
- AI-powered image generation integration
- Firebase-powered database storage
- User workspace/dashboard
- Editable slide interface
- Complete documentation

3.7 Summary

This chapter presented the detailed objectives, functional requirements, and non-functional requirements of the project. These criteria form the foundation upon which the system is designed and implemented. The next chapter will describe the system architecture and design methodology used to achieve these objectives.

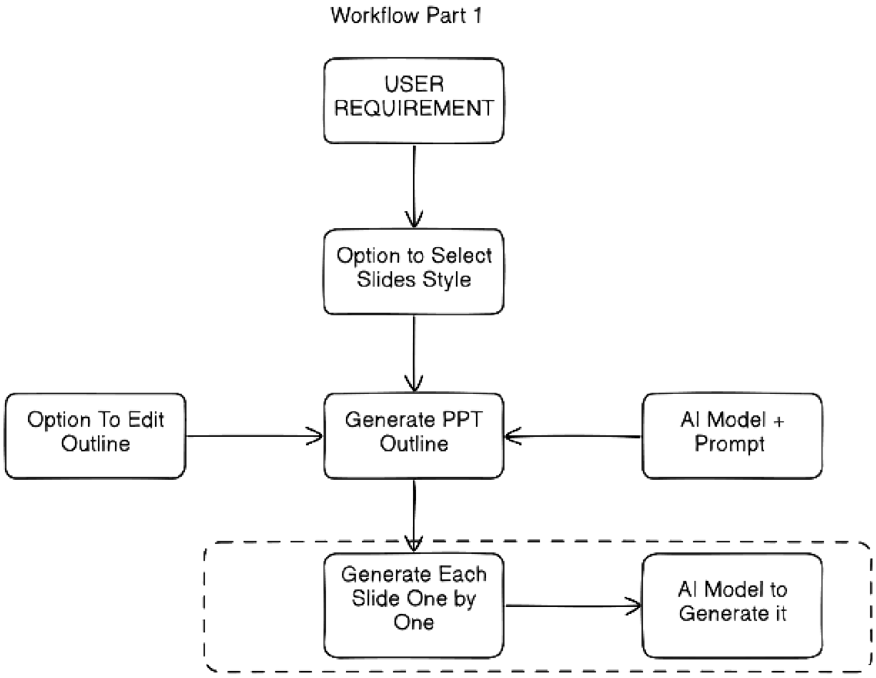
CHAPTER 4
SYSTEM DESIGN AND METHODOLOGY

4.1 Introduction

System design defines how all components of the AI-powered presentation generator interact, communicate, and work together to achieve automated outline generation, slide creation, and AI-driven enhancements. This chapter explains the architecture, design methodology, workflow, and module integration of the system.

4.2 System Architecture Overview

The system follows a client-cloud architecture where the React frontend communicates with external AI services and Firebase backend components. The architecture ensures scalability, modularity, and reliability across multiple layers of the system.



4.3 System Modules

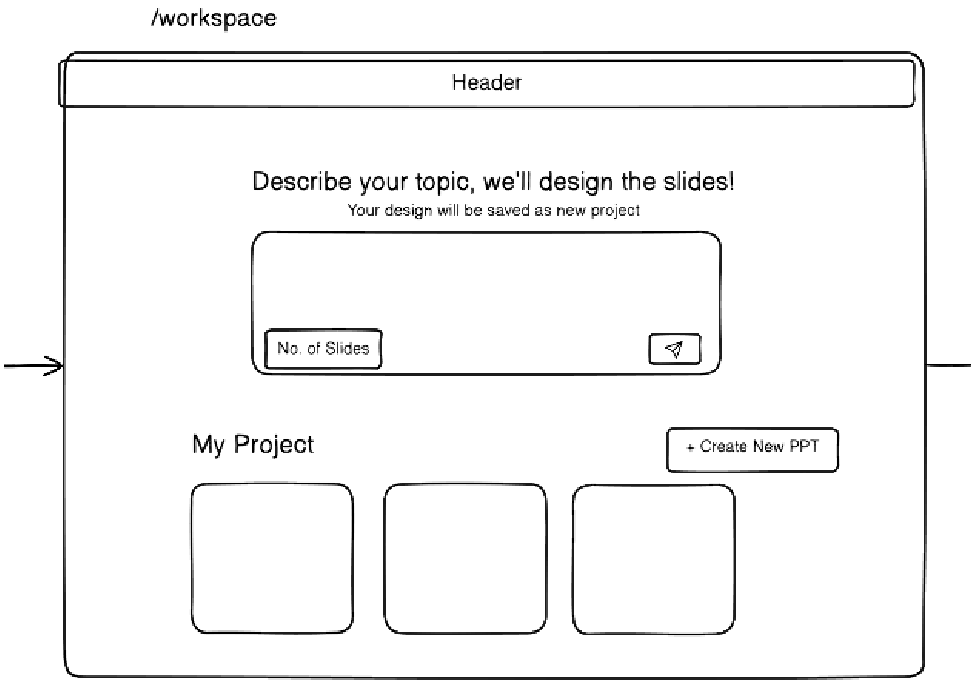
The system is divided into several modules, each responsible for a specific part of the workflow. These modules work together to provide a seamless slide generation experience.

4.3.1 Authentication & Subscription Module (Clerk)

Clerk provides a complete authentication solution that manages user identities using secure, token-based authentication. It simplifies the login and registration process through email, password, OTP, and social login options such as Google or GitHub. Clerk also maintains persistent user sessions, ensuring users stay securely logged in across page reloads. Additionally, it integrates subscription management, enabling the system to differentiate between free and premium users and enforce feature restrictions accordingly.

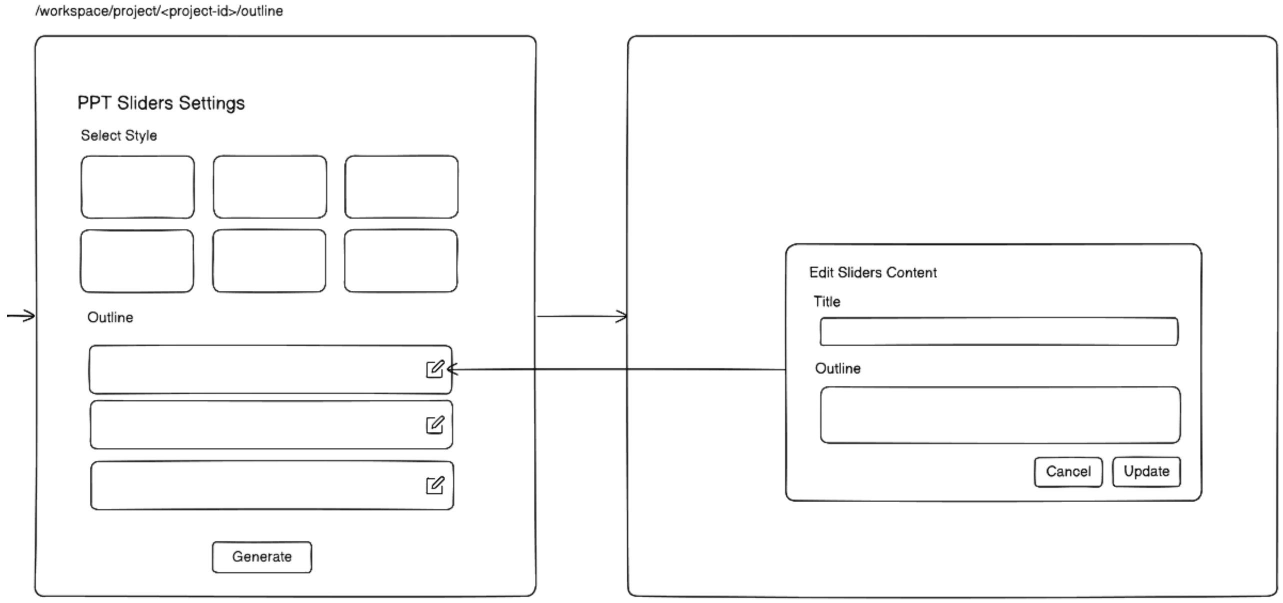
4.3.2 Workspace Module

The workspace allows users to create, manage, edit, and delete projects. It loads project metadata from Firebase and displays all user-generated presentations.


/workspace
Header
Describe your topic, we'll design the slides!
Your design will be saved as new project
No. of Slides
My Project
+ Create New PPT

4.3.3 Outline Generation Module (Gemini AI)

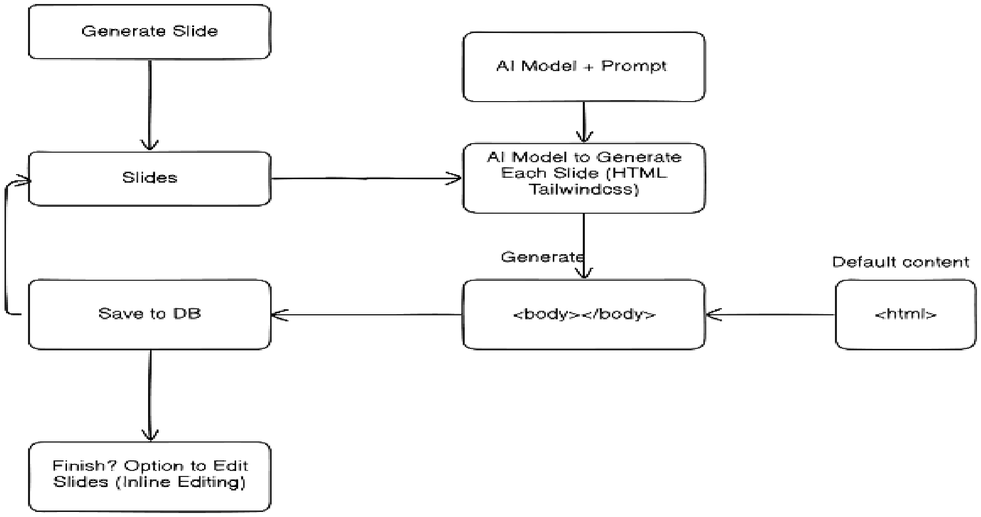
This module processes the topic entered by the user and sends it to the Gemini AI model using a carefully structured prompt. Gemini then generates a well-organized outline consisting of slide titles and key bullet points. Once the outline is received, it is stored in Firebase Firestore so that it can be accessed anytime during the project workflow. The user is then able to view, edit, reorder, or refine the outline directly in the Outline Editor movingto the slide generation stage.



.

4.3.4 Slide Content Generation Module

Gemini AI not only generates the textual content for each slide but also provides structural guidance on how the information should be visually arranged. Based on the slide outline, it expands bullet points into meaningful explanations, organizes content into well-structured sections, and proposes an appropriate layout such as two-column designs, image-text combinations, or centered title formats. These AI-generated layout suggestions are then transformed into HTML components styled with TailwindCSS, ensuring that each slide maintains a clean, modern, and responsive design. This automated conversion significantly reduces manual design work while delivering consistent and professional-quality slide templates.

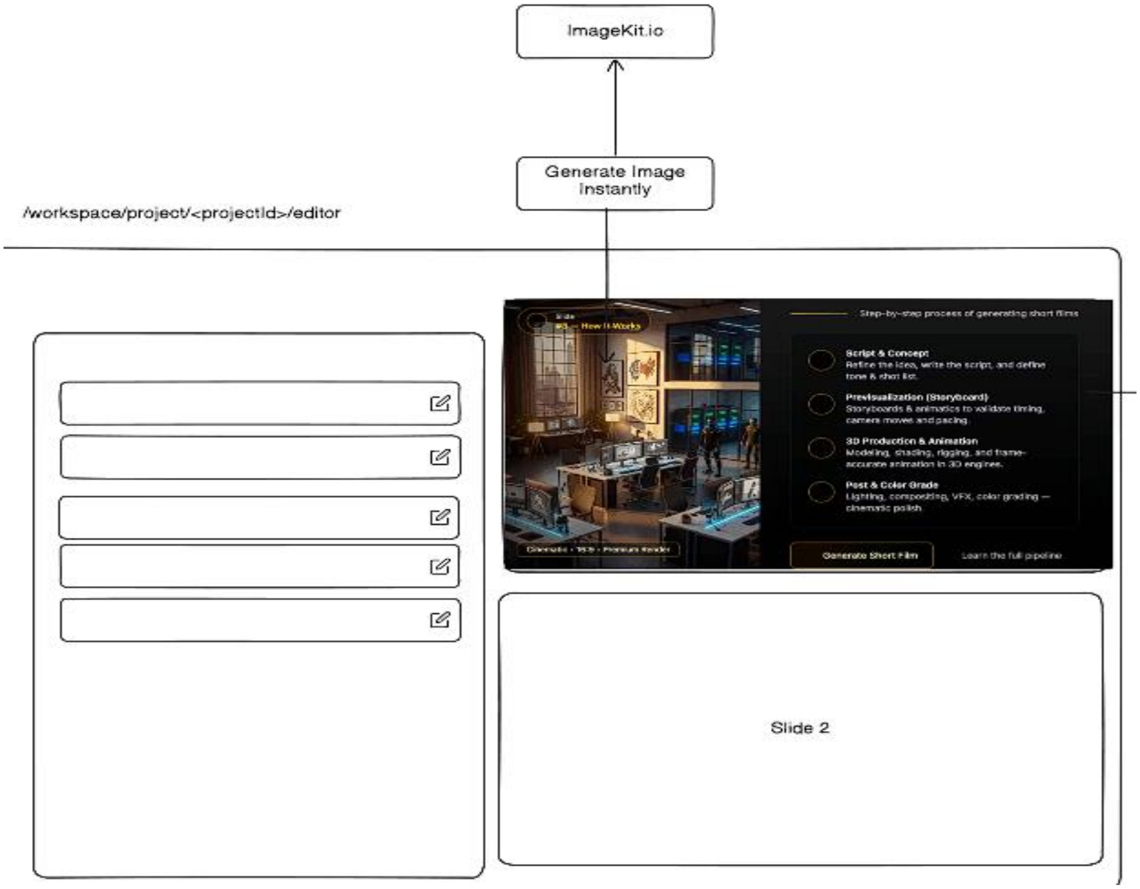

Generate Slide
AI Model + Prompt
Slides
AI Model to Generate Each Slide (HTML Tailwindcss)
Generate
Default content
Save to DB
<body></body>
<html>
Finish? Option to Edit Slides (Inline Editing)

4.3.5 Slide Rendering Module

React and Tailwind-CSS work together to dynamically generate and update slide layouts in real time. React’s component-based architecture enables instant re-rendering of slides whenever users make edits, such as changing text, modifying headings, or adjusting layout structures. Tailwind-CSS provides utility-first styling, allowing visually appealing spacing, alignment, and color adjustments without manually writing CSS. This combination ensures a smooth editing experience where updates appear instantly on the screen, eliminating the need for manual refresh or page reloads.

4.3.6 Image Generation & Enhancement Module (ImageKit.io)

ImageKit generates high-quality images based on AI-generated prompts. The returned URLs are inserted into slide templates. ImageKit also performs optimization, re-sizing, and transformations.



Step-by-step process of generating short films

Script & Concept
Refine the idea, write the script, and define tone & shot list.

Previsualization (Storyboard)
Storyboards & animatics to validate timing, camera moves and pacing.

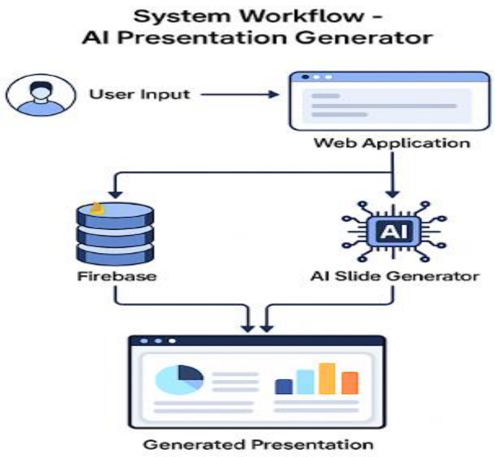
3D Production & Animation
Modeling, shading, rigging, and frame-accurate animation in 3D engines.

Post & Color Grade
Lighting, compositing, VFX, color grading — cinematic polish.

4.3.7 Database Management Module (Firebase Firestore)

Firebase Firestore acts as the primary cloud database for the AI Presentation Generator, storing all essential user data, project information, outlines, slide content, selected themes, and AI-generated assets. It is a NoSQL document-based database, meaning data is stored in collections, documents, and fields, which allows flexible schema design and highly scalable storage.

4.4 System Workflow



4.5 Design Methodology

The system adopts a combination of Agile methodology, modular code design, component-based UI architecture, and structured prompt engineering to ensure a reliable and scalable development workflow. Using the Agile approach allows the application to be built in iterative cycles, where features are tested, improved, and expanded based on continuous feedback. The modular code structure ensures that each system component—such as authentication, outline generation, slide editing, and image generation—remains independent and reusable, making maintenance significantly easier. The component-based UI architecture of React further enhances flexibility by breaking the interface into small, manageable units that render dynamically and update efficiently. Additionally, structured prompt engineering ensures that the AI consistently generates accurate outlines, coherent slide content, and optimized layout instructions. Combined, these practices result in a system that is scalable, easy to extend with new features, and capable of delivering consistent performance across all user interactions.

4.6 Summary

This chapter explained the design principles, architectural decisions, workflow, and module interactions behind the AI-powered presentation generator. Each section included placeholders where corresponding diagrams should be inserted to visually represent the system's structure and logic.

CHAPTER 5

CODING AND IMPLEMENTATION

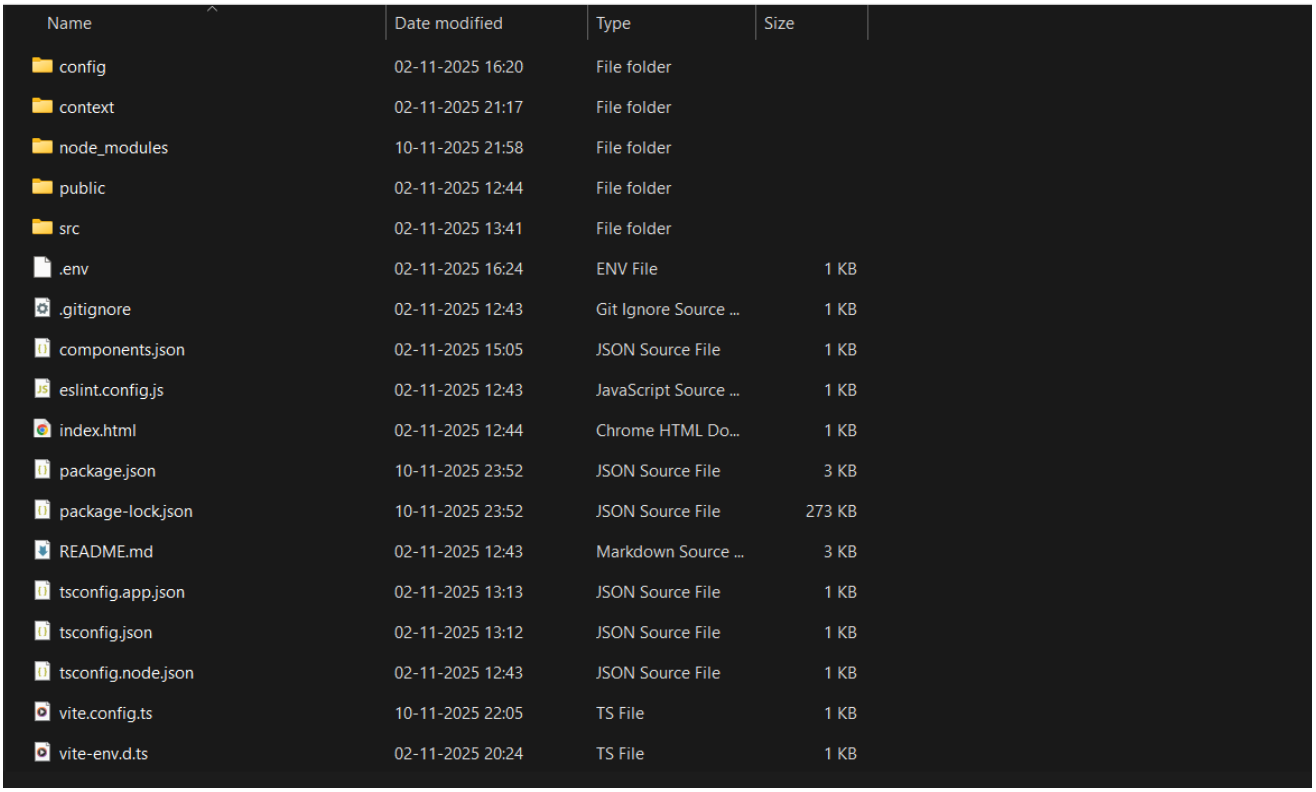
5.1 Introduction

This chapter describes the coding structure and implementation details of the AI-powered presentation generator system. It includes frontend development using React, backend utilities, AI integration through Gemini API, image generation using ImageKit, authentication using Clerk, and cloud database operations using Firebase Firestore. The goal of this chapter is to provide a clear understanding of how each part of the system is built and inter-connected.

5.2 Frontend Implementation (React + TailwindCSS)

The frontend of the system is developed using React.js, featuring reusable components, modular architecture, and real-time rendering capabilities. TailwindCSS is used to ensure clean, responsive, and modern UI styling with minimal CSS overhead.

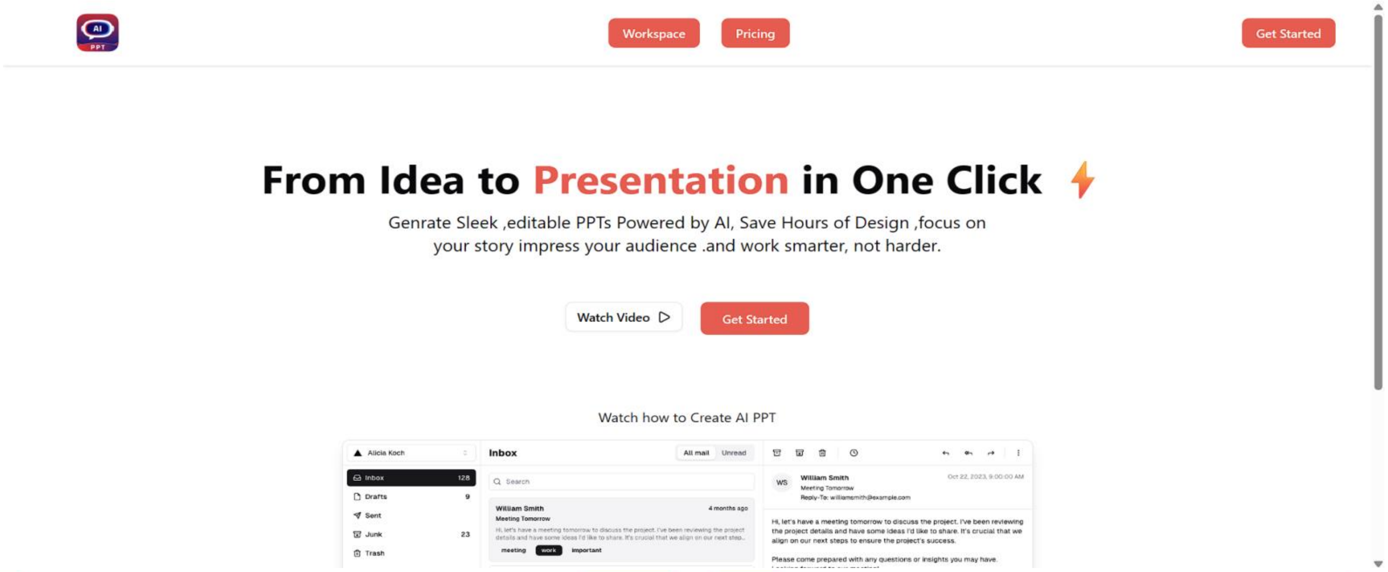
5.2.1 Project Folder Structure



Name	Date modified	Type	Size
config	02-11-2025 16:20	File folder	
context	02-11-2025 21:17	File folder	
node_modules	10-11-2025 21:58	File folder	
public	02-11-2025 12:44	File folder	
src	02-11-2025 13:41	File folder	
.env	02-11-2025 16:24	ENV File	1 KB
.gitignore	02-11-2025 12:43	Git Ignore Source ...	1 KB
components.json	02-11-2025 15:05	JSON Source File	1 KB
eslint.config.js	02-11-2025 12:43	JavaScript Source ...	1 KB
index.html	02-11-2025 12:44	Chrome HTML Do...	1 KB
package.json	10-11-2025 23:52	JSON Source File	3 KB
package-lock.json	10-11-2025 23:52	JSON Source File	273 KB
README.md	02-11-2025 12:43	Markdown Source ...	3 KB
tsconfig.app.json	02-11-2025 13:13	JSON Source File	1 KB
tsconfig.json	02-11-2025 13:12	JSON Source File	1 KB
tsconfig.node.json	02-11-2025 12:43	JSON Source File	1 KB
vite.config.ts	10-11-2025 22:05	TS File	1 KB
vite-env.d.ts	02-11-2025 20:24	TS File	1 KB

5.2.2 Key Components

Below is an overview of the key components involved in the early user flow of the system, starting from the Landing Page where the user enters their presentation topic, moving into the Workspace where all projects are displayed, and finally navigating to the Outline Settings page where the user configures the number of slides, theme preferences, and other AI generation options.

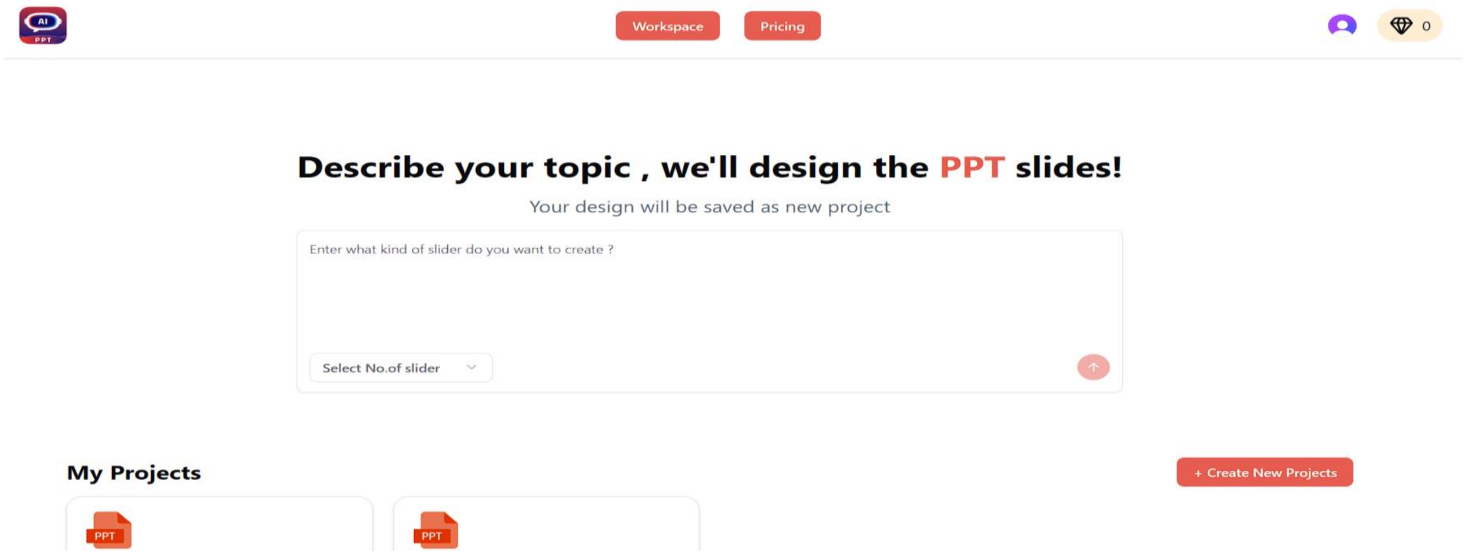

AI
PPT
Workspace
Pricing
Get Started

From Idea to Presentation in One Click

Genrate Sleek ,editable PPTs Powered by AI, Save Hours of Design ,focus on your story impress your audience .and work smarter, not harder.

Watch how to Create AI PPT

Alicia Koch
Inbox 128
Drafts 9
Sent
Junk 23
Trash
Inbox
All mail
Unread
Search
William Smith
4 months ago
Meeting Tomorrow
Hi, let's have a meeting tomorrow to discuss the project. I've been reviewing the project details and have some ideas I'd like to share. It's crucial that we align on our next step...
meeting
work
important
WS
William Smith
Meeting Tomorrow
Reply-To: williamsmith@example.com
Oct 22, 2023, 9:00:00 AM
Hi, let's have a meeting tomorrow to discuss the project. I've been reviewing the project details and have some ideas I'd like to share. It's crucial that we align on our next steps to ensure the project's success.
Please come prepared with any questions or insights you may have.

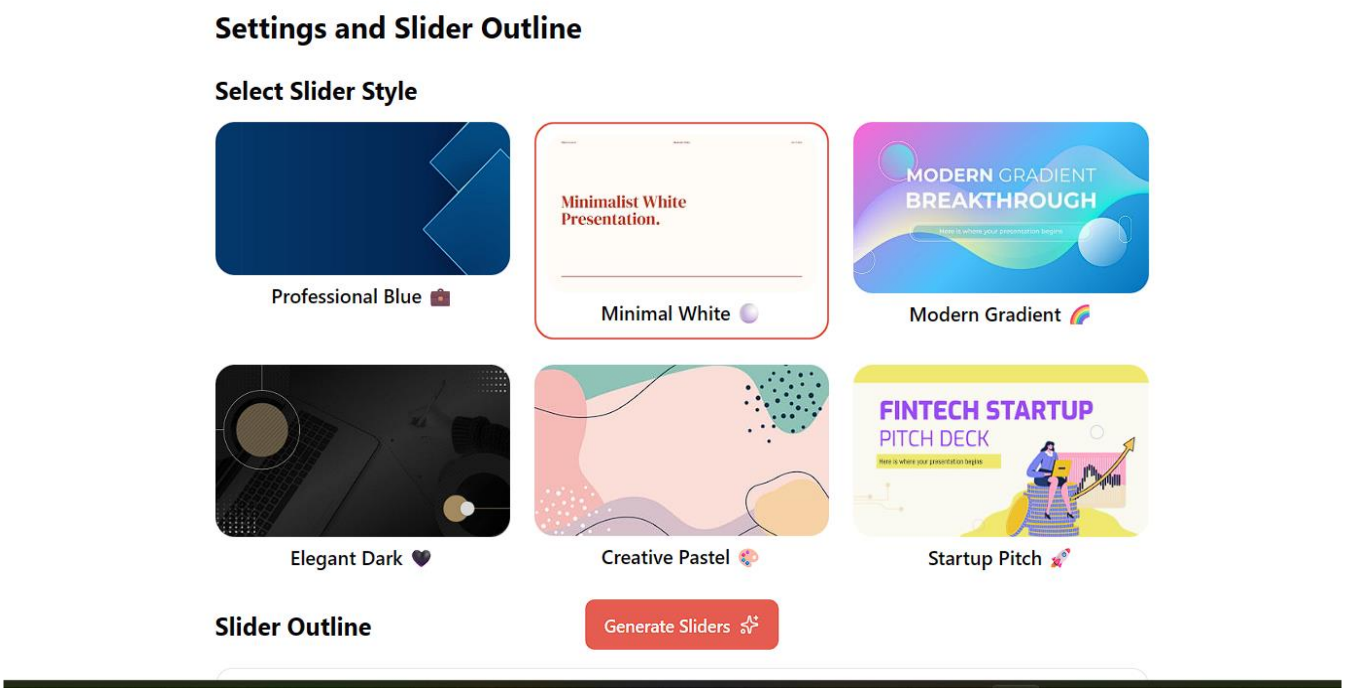


Describe your topic , we'll design the PPT slides!

Your design will be saved as new project

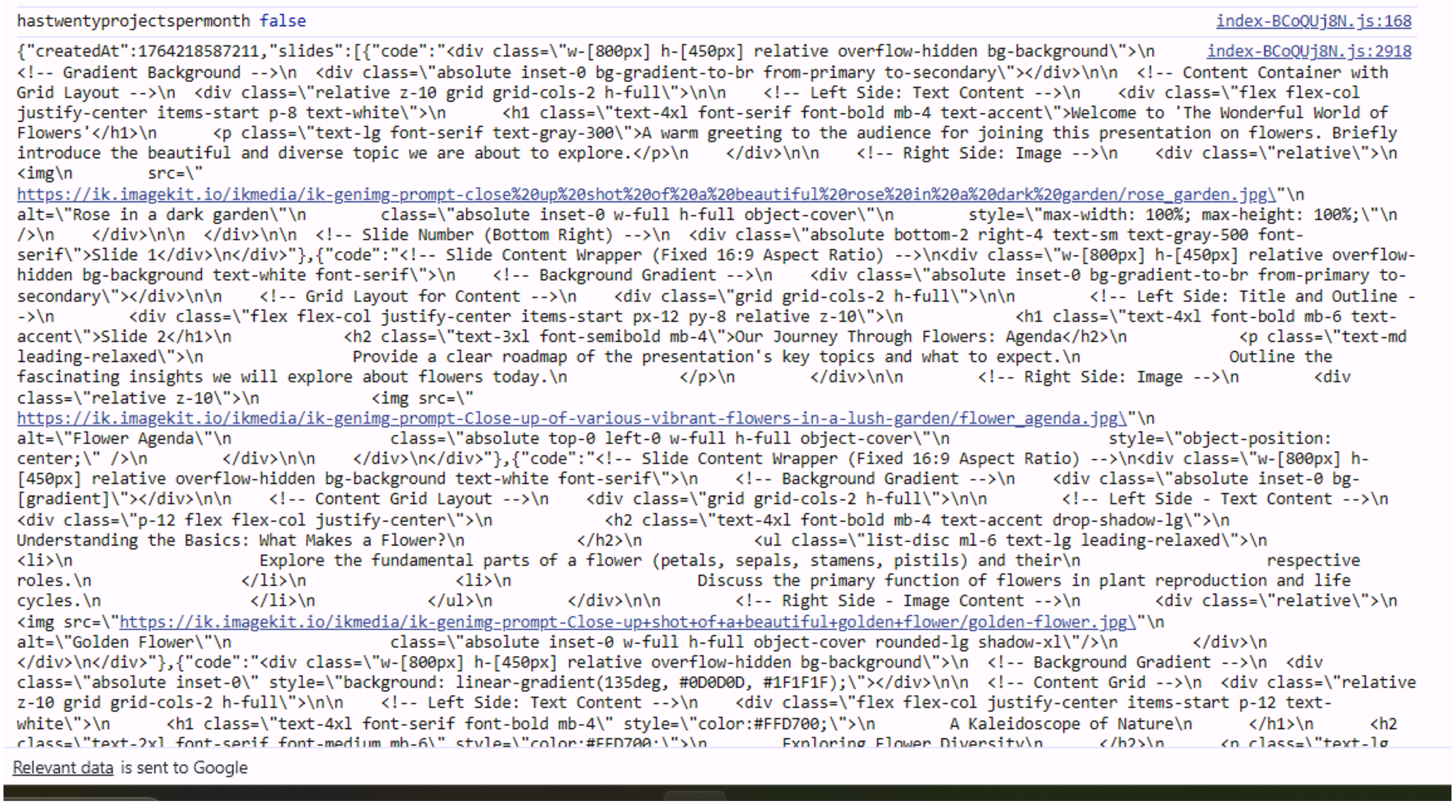
Enter what kind of slider do you want to create ?

My Projects


Settings and Slider Outline
Select Slider Style
Minimalist White Presentation.
Professional Blue 💼
Minimal White 🔮
MODERN GRADIENT BREAKTHROUGH
Here is where your presentation begins
Modern Gradient 🌈
Elegant Dark 💜
Creative Pastel 🎨
FINTECH STARTUP PITCH DECK
Here is where your presentation begins
Startup Pitch 🚀
Slider Outline
Generate Sliders

5.3 Backend Logic and Utility Functions

Backend logic consists of API calls to Gemini AI, Firestore operations, and ImageKit transformations. Utility functions manage recurring tasks such as generating prompts, saving project data, or retrieving slide content.



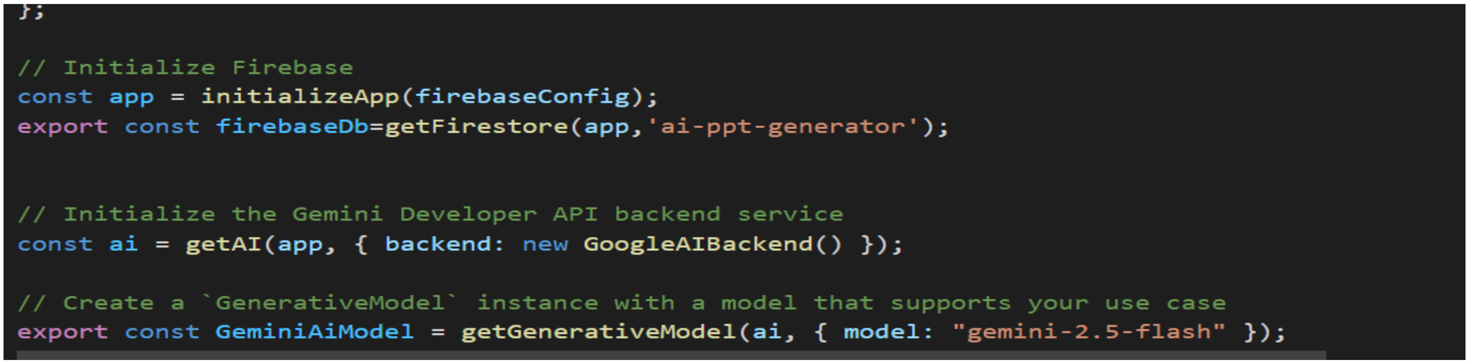
```
hastwentyprojectspermonth false                                                                          index-BCoQUj8N.js:168
{"createdAt":1764218587211,"slides":[{"code":"<div class=\"w-[800px] h-[450px] relative overflow-hidden bg-background\">\n     index-BCoQUj8N.js:2918
<!-- Gradient Background -->\n  <div class=\"absolute inset-0 bg-gradient-to-br from-primary to-secondary\"></div>\n\n  <!-- Content Container with
Grid Layout -->\n  <div class=\"relative z-10 grid grid-cols-2 h-full\">\n\n    <!-- Left Side: Text Content -->\n    <div class=\"flex flex-col
justify-center items-start p-8 text-white\">\n      <h1 class=\"text-4xl font-serif font-bold mb-4 text-accent\">Welcome to 'The Wonderful World of
Flowers'</h1>\n      <p class=\"text-lg font-serif text-gray-300\">A warm greeting to the audience for joining this presentation on flowers. Briefly
introduce the beautiful and diverse topic we are about to explore.</p>\n    </div>\n\n    <!-- Right Side: Image -->\n    <div class=\"relative\">\n
<img\n        src=\"
https://ik.imagekit.io/ikmedia/ik-genimg-prompt-close%20up%20shot%20of%20a%20beautiful%20rose%20in%20a%20dark%20garden/rose_garden.jpg\"\n
alt=\"Rose in a dark garden\"\n        class=\"absolute inset-0 w-full h-full object-cover\"\n        style=\"max-width: 100%; max-height: 100%;\"\n
/>\n    </div>\n\n  </div>\n\n  <!-- Slide Number (Bottom Right) -->\n  <div class=\"absolute bottom-2 right-4 text-sm text-gray-500 font-
serif\">Slide 1</div>\n</div>"},{"code":"<!-- Slide Content Wrapper (Fixed 16:9 Aspect Ratio) -->\n<div class=\"w-[800px] h-[450px] relative overflow-
hidden bg-background text-white font-serif\">\n    <!-- Background Gradient -->\n    <div class=\"absolute inset-0 bg-gradient-to-br from-primary to-
secondary\"></div>\n\n    <!-- Grid Layout for Content -->\n    <div class=\"grid grid-cols-2 h-full\">\n\n        <!-- Left Side: Title and Outline -
->\n        <div class=\"flex flex-col justify-center items-start px-12 py-8 relative z-10\">\n            <h1 class=\"text-4xl font-bold mb-6 text-
accent\">Slide 2</h1>\n            <h2 class=\"text-3xl font-semibold mb-4\">Our Journey Through Flowers: Agenda</h2>\n            <p class=\"text-md
leading-relaxed\">\n                Provide a clear roadmap of the presentation's key topics and what to expect.\n                Outline the
fascinating insights we will explore about flowers today.\n            </p>\n        </div>\n\n        <!-- Right Side: Image -->\n        <div
class=\"relative z-10\">\n            <img src=\"
https://ik.imagekit.io/ikmedia/ik-genimg-prompt-Close-up-of-various-vibrant-flowers-in-a-lush-garden/flower_agenda.jpg\"\n
alt=\"Flower Agenda\"\n                 class=\"absolute top-0 left-0 w-full h-full object-cover\"\n                 style=\"object-position:
center;\" />\n        </div>\n\n    </div>\n</div>"},{"code":"<!-- Slide Content Wrapper (Fixed 16:9 Aspect Ratio) -->\n<div class=\"w-[800px] h-
[450px] relative overflow-hidden bg-background text-white font-serif\">\n    <!-- Background Gradient -->\n    <div class=\"absolute inset-0 bg-
[gradient]\"></div>\n\n    <!-- Content Grid Layout -->\n    <div class=\"grid grid-cols-2 h-full\">\n\n        <!-- Left Side - Text Content -->\n
<div class=\"p-12 flex flex-col justify-center\">\n            <h2 class=\"text-4xl font-bold mb-4 text-accent drop-shadow-lg\">\n
Understanding the Basics: What Makes a Flower?\n            </h2>\n            <ul class=\"list-disc ml-6 text-lg leading-relaxed\">\n
<li>\n                    Explore the fundamental parts of a flower (petals, sepals, stamens, pistils) and their\n                    respective
roles.\n                </li>\n                <li>\n                    Discuss the primary function of flowers in plant reproduction and life
cycles.\n                </li>\n            </ul>\n        </div>\n\n        <!-- Right Side - Image Content -->\n        <div class=\"relative\">\n
<img src=\"https://ik.imagekit.io/ikmedia/ik-genimg-prompt-Close-up+shot+of+a+beautiful+golden+flower/golden-flower.jpg\"\n
alt=\"Golden Flower\"\n                 class=\"absolute inset-0 w-full h-full object-cover rounded-lg shadow-xl\"/>\n        </div>\n
</div>\n</div>"},{"code":"<div class=\"w-[800px] h-[450px] relative overflow-hidden bg-background\">\n  <!-- Background Gradient -->\n  <div
class=\"absolute inset-0\" style=\"background: linear-gradient(135deg, #0D0D0D, #1F1F1F);\"></div>\n\n  <!-- Content Grid -->\n  <div class=\"relative
z-10 grid grid-cols-2 h-full\">\n\n    <!-- Left Side: Text Content -->\n    <div class=\"flex flex-col justify-center items-start p-12 text-
white\">\n      <h1 class=\"text-4xl font-serif font-bold mb-4\" style=\"color:#FFD700;\">\n        A Kaleidoscope of Nature\n      </h1>\n      <h2
class=\"text-2xl font-serif font-medium mb-6\" style=\"color:#FFD700;\">\n        Exploring Flower Diversity\n      </h2>\n      <p class=\"text-lg
```

Relevant data is sent to Google

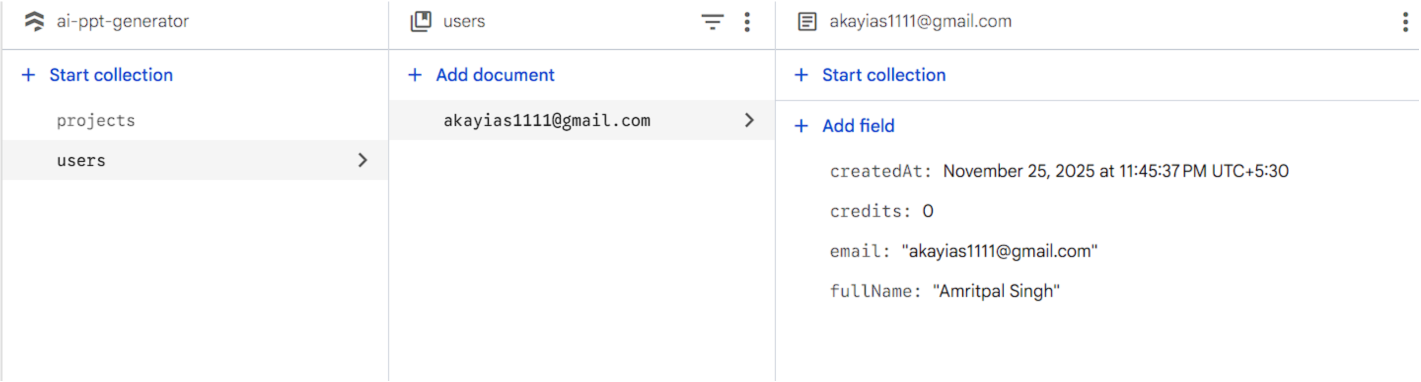
5.4 Firebase Integration

Firebase Firestore is used to securely store user accounts, project metadata, outlines, slides, and image URLs. Its real-time syncing capabilities ensure that changes made in the editor are instantly saved and reflected across the system.

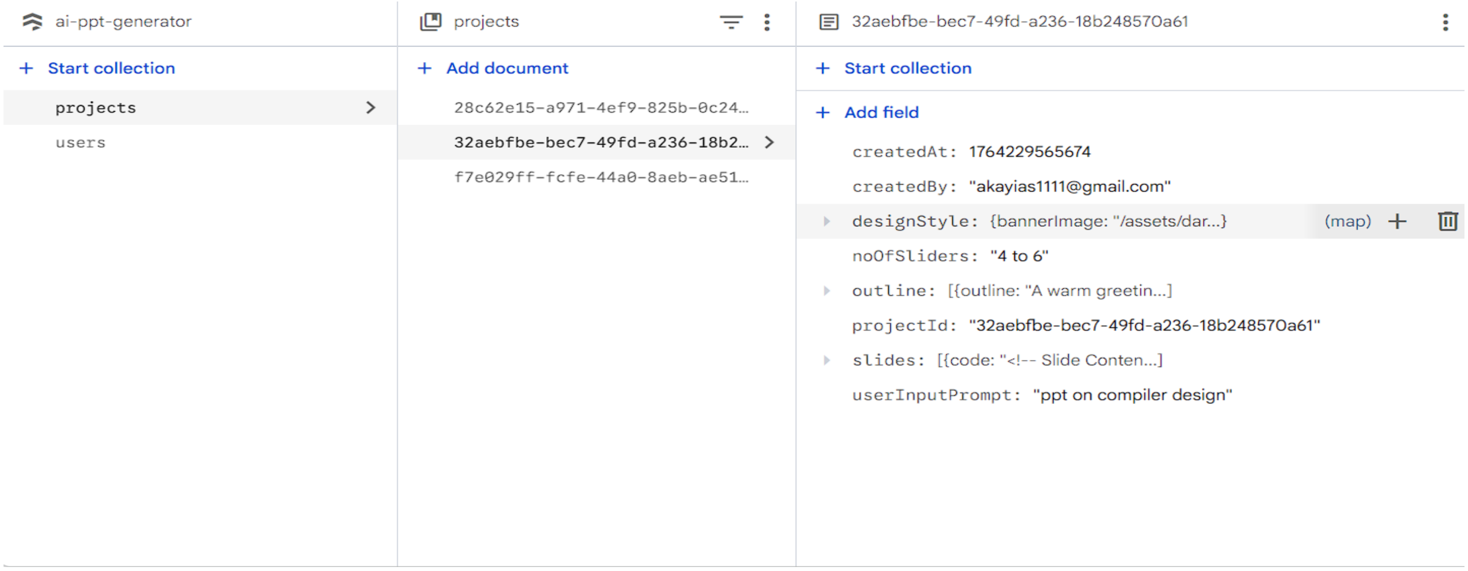
5.4.1 Firebase Configuration Code



```
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
export const firebaseDb=getFirestore(app,'ai-ppt-generator');


// Initialize the Gemini Developer API backend service
const ai = getAI(app, { backend: new GoogleAIBackend() });

// Create a `GenerativeModel` instance with a model that supports your use case
export const GeminiAiModel = getGenerativeModel(ai, { model: "gemini-2.5-flash" });
```



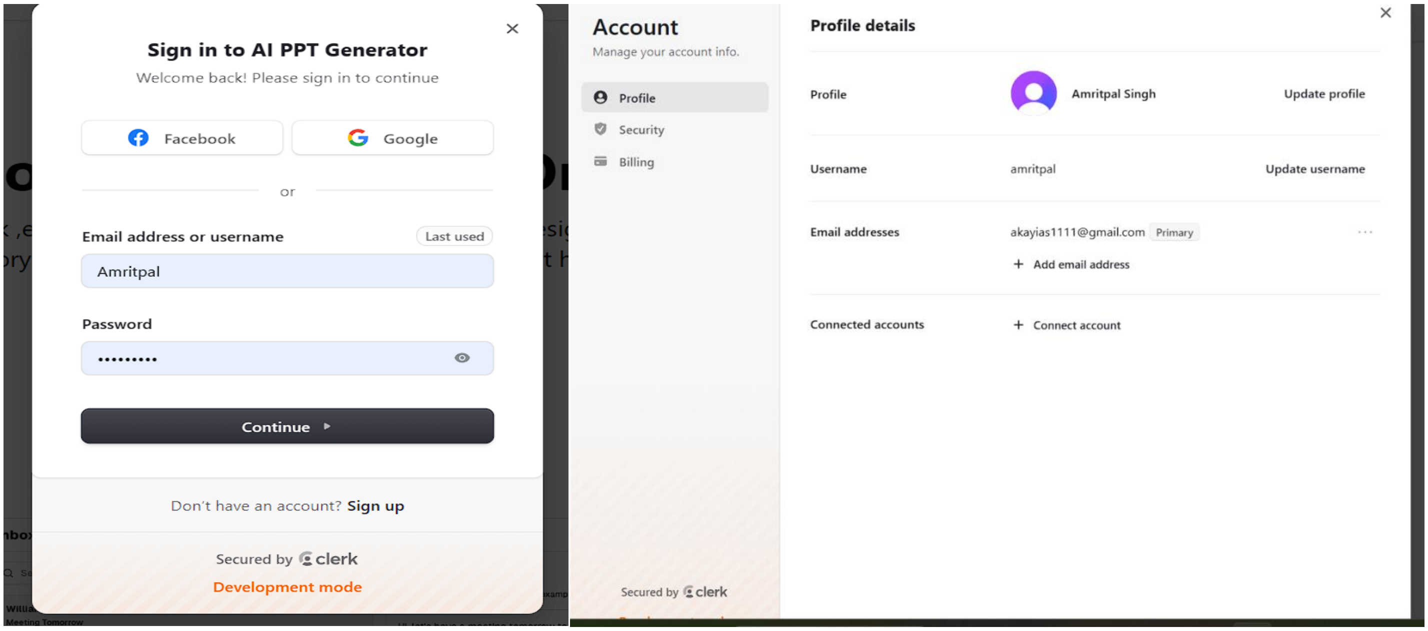
```
createdAt: November 25, 2025 at 11:45:37 PM UTC+5:30
credits: 0
email: "akayias1111@gmail.com"
fullName: "Amritpal Singh"
```


ai-ppt-generator
projects
32aebfbe-bec7-49fd-a236-18b248570a61
+ Start collection
+ Add document
+ Start collection
projects
users
28c62e15-a971-4ef9-825b-0c24…
32aebfbe-bec7-49fd-a236-18b2…
f7e029ff-fcfe-44a0-8aeb-ae51…
+ Add field
(map)

```
createdAt: 1764229565674
createdBy: "akayias1111@gmail.com"
designStyle: {bannerImage: "/assets/dar...}
noOfSliders: "4 to 6"
outline: [{outline: "A warm greetin...]
projectId: "32aebfbe-bec7-49fd-a236-18b248570a61"
slides: [{code: "<!-- Slide Conten...]
userInputPrompt: "ppt on compiler design"
```

5.5 Authentication Using Clerk

Clerk Authentication provides a complete user management system that ensures each user accessing the application is securely verified. It supports multiple authentication methods such as email/password, OTP, and third-party logins like Google or GitHub, making the login process convenient and secure. Once a user signs in, Clerk maintains their active session across pages, so they don’t need to reauthenticate repeatedly. Additionally, Clerk stores subscription details for each user, allowing the system to identify whether they belong to the freepremium tier. Based on this information, the application can enforce feature limits, restrict access to advancetools, and deliver a customized user experience.



Sign in to AI PPT Generator

Welcome back! Please sign in to continue

Facebook | Google

or

Email address or username — Last used

Amritpal

Password

•••••••••

Continue ▸

Don't have an account? **Sign up**

Secured by clerk

Development mode

Account

Manage your account info.

- Profile
- Security
- Billing

Secured by clerk

Profile details

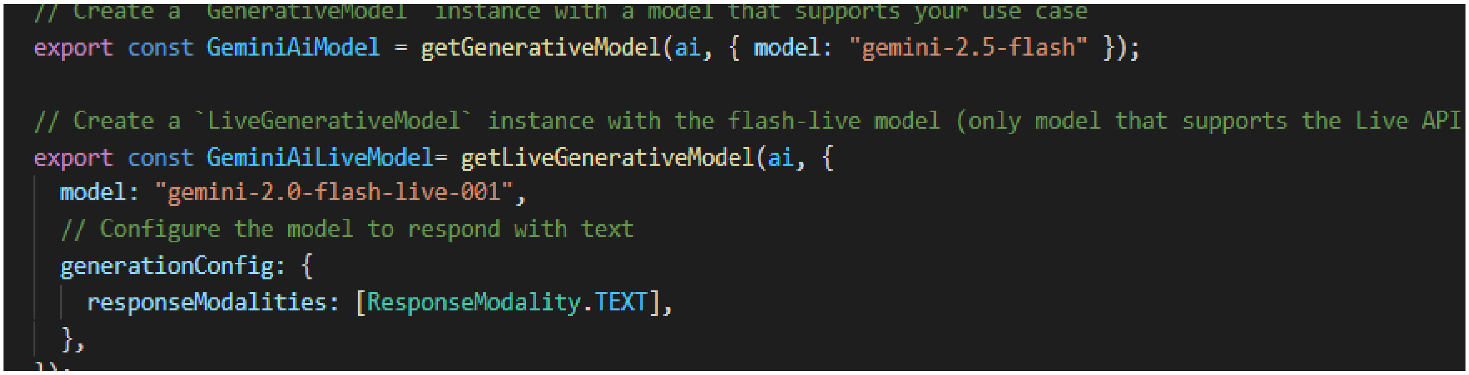
Profile — Amritpal Singh — Update profile

Username — amritpal — Update username

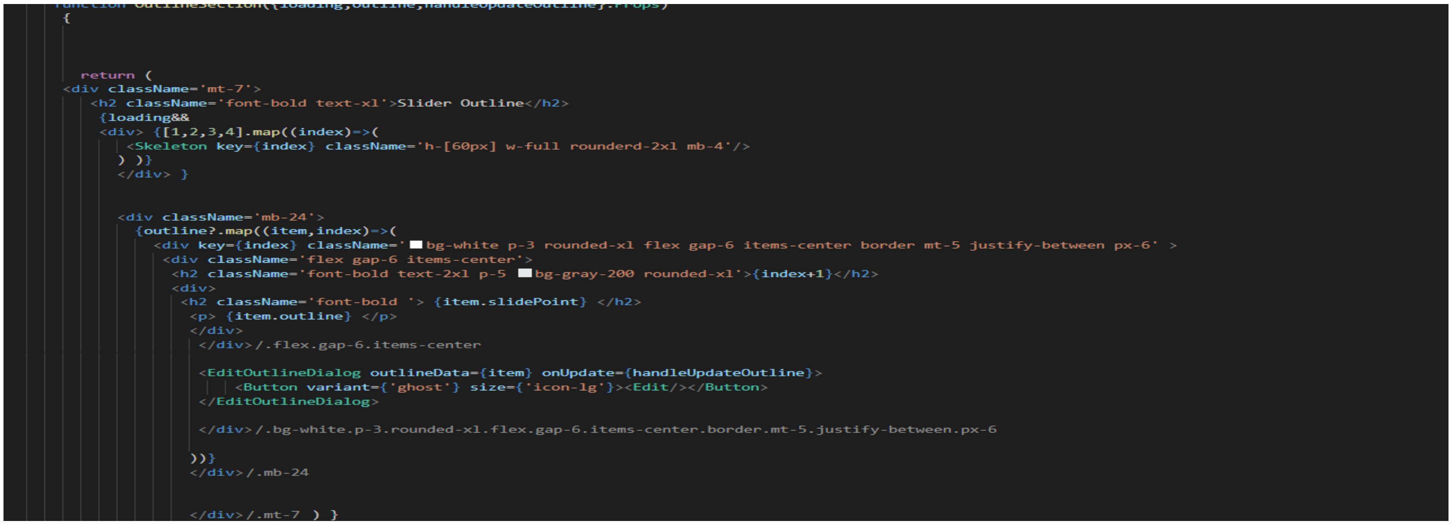
Email addresses — akayias1111@gmail.com Primary — + Add email address

Connected accounts — + Connect account

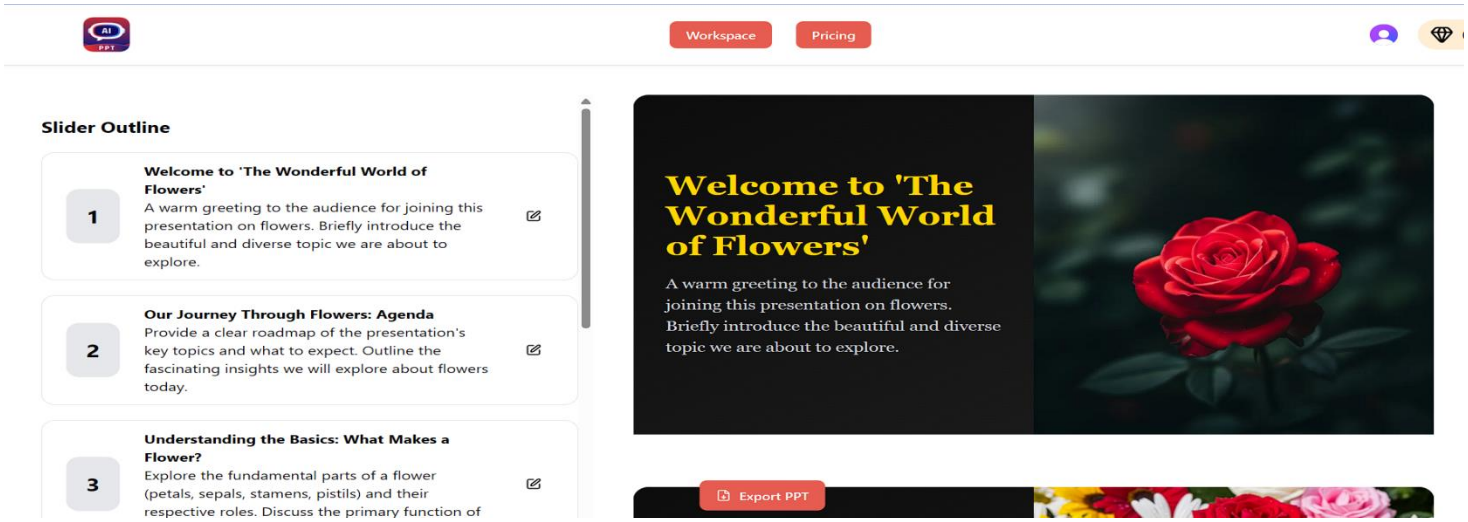
5.6 AI Integration (Gemini API + Prompt Engineering+ImageKit Integration)



```
// Create a `GenerativeModel` instance with a model that supports your use case
export const GeminiAiModel = getGenerativeModel(ai, { model: "gemini-2.5-flash" });

// Create a `LiveGenerativeModel` instance with the flash-live model (only model that supports the Live API
export const GeminiAiLiveModel= getLiveGenerativeModel(ai, {
  model: "gemini-2.0-flash-live-001",
  // Configure the model to respond with text
  generationConfig: {
    responseModalities: [ResponseModality.TEXT],
  },
});
```



```
function OutlineSection({loading,outline,handleUpdateOutline}:Props)
{


  return (
<div className='mt-7'>
   <h2 className='font-bold text-xl'>Slider Outline</h2>
    {loading&&
    <div> {[1,2,3,4].map((index)=>(
       <Skeleton key={index} className='h-[60px] w-full rounderd-2xl mb-4'/>
      ) )}
      </div> }


      <div className='mb-24'>
        {outline?.map((item,index)=>(
          <div key={index} className=' bg-white p-3 rounded-xl flex gap-6 items-center border mt-5 justify-between px-6' >
           <div className='flex gap-6 items-center'>
            <h2 className='font-bold text-2xl p-5  bg-gray-200 rounded-xl'>{index+1}</h2>
            <div>
             <h2 className='font-bold '> {item.slidePoint} </h2>
              <p> {item.outline} </p>
              </div>
               </div>/.flex.gap-6.items-center

               <EditOutlineDialog outlineData={item} onUpdate={handleUpdateOutline}>
                   <Button variant={'ghost'} size={'icon-lg'}><Edit/></Button>
               </EditOutlineDialog>

               </div>/.bg-white.p-3.rounded-xl.flex.gap-6.items-center.border.mt-5.justify-between.px-6

              ))}
              </div>/.mb-24


              </div>/.mt-7  ) }
```



Slider Outline

1 Welcome to 'The Wonderful World of Flowers'
A warm greeting to the audience for joining this presentation on flowers. Briefly introduce the beautiful and diverse topic we are about to explore.

2 Our Journey Through Flowers: Agenda
Provide a clear roadmap of the presentation's key topics and what to expect. Outline the fascinating insights we will explore about flowers today.

3 Understanding the Basics: What Makes a Flower?
Explore the fundamental parts of a flower (petals, sepals, stamens, pistils) and their respective roles. Discuss the primary function of

Welcome to 'The Wonderful World of Flowers'

A warm greeting to the audience for joining this presentation on flowers. Briefly introduce the beautiful and diverse topic we are about to explore.

5.7 Testing and Validation

Multiple types of testing were performed to ensure correct system behavior, including functional testing of each module, AI response validation, Firestore data consistency checks, and authentication flow verification.

5.8 Summary

This chapter detailed the coding and implementation aspects of the system, demonstrating how each module—from frontend React components to backend AI utilities—works together to create an automated AI presentation generator. It also included placeholders for screenshots and code snippets to illustrate the development process.

CHAPTER 6
RESULTS, CONCLUSION AND FUTURE SCOPE

6.1 Introduction

This chapter presents the outcomes of developing the AI-Powered Presentation Generator system. It summarizes the system's performance, key achievements, limitations, and future scope, offering insights into how well the final product meets its intended objectives and how it can evolve further.

6.2 Project Results and System Output

The system successfully automates the creation of presentation slides based on a user-provided topic. It generates structured outlines, meaningful slide content, visually consistent HTML/TailwindCSS layouts, and relevant AI-generated images using ImageKit. All generated data—including outlines, slides, themes, and image URLs—is stored and synced securely via Firebase Firestore. Real-time editing allows users to instantly update slide content without page reloads, showcasing smooth frontend–backend integration.

Testing demonstrated that AI responses were accurate and contextually aligned with user prompts. System responsiveness remained stable during interactions such as outline editing, slide regeneration, and saving project states.

6.3 Key Achievements

- Fully automated outline creation using Gemini AI.
- AI-powered slide text generation with structured formatting.
- Dynamic HTML + TailwindCSS slide templates for modern UI design.
- Real-time slide editor with instant Firebase synchronization.
- Secure authentication using Clerk with subscription-based access control.
- Automated image generation and transformation using ImageKit.
- A complete user workspace for managing multiple presentation projects.

6.4 System Performance Evaluation

The system was evaluated based on speed, accuracy, usability, and scalability. Gemini AI generated outlines and slide content within a reasonable time frame. Firestore ensured fast read/write operations, enabling instant updates across the UI. The application maintained consistent performance across browsers and devices, demonstrating high usability and reliability. The modular architecture allowed smooth functioning of all components, reflecting a well-engineered system.

6.5 Limitations of the System

- Requires stable internet connectivity for AI, Firestore, and ImageKit operations.
- AI-generated content may require manual refinement for accuracy or tone.
- Export to PPTX is not yet available in the current version.
- Customization options (themes, animations) are limited.
- Image quality and style depend on the capabilities of ImageKit's AI model.

6.6 Conclusion

The project successfully demonstrates how AI can automate the traditionally manual task of presentation creation. By integrating React, Firebase, Clerk, Gemini AI, and ImageKit, the system delivers an efficient and intelligent workflow that reduces user effort while increasing productivity. The completed system fulfills the primary objectives set during the analysis phase, proving the feasibility and effectiveness of AI-driven presentation generation tools.

6.7 Future Scope

- Adding export functionality for PPTX and Google Slides.
- Enabling real-time team collaboration on presentation editing.
- Introducing advanced themes, transitions, and animation effects.
- Integrating more powerful AI models for improved content quality.
- Enhancing image generation with style selection and editing tools.
- Supporting multilingual presentation generation.
- Developing a mobile application version for on-the-go usage.

6.8 Summary

This chapter summarized the key results, achievements, system evaluation, limitations, and future expansion opportunities for the AI Presentation Generator. The system provides a solid foundation for further development and demonstrates practical application of modern AI and cloud technologies to solve real-world problems.

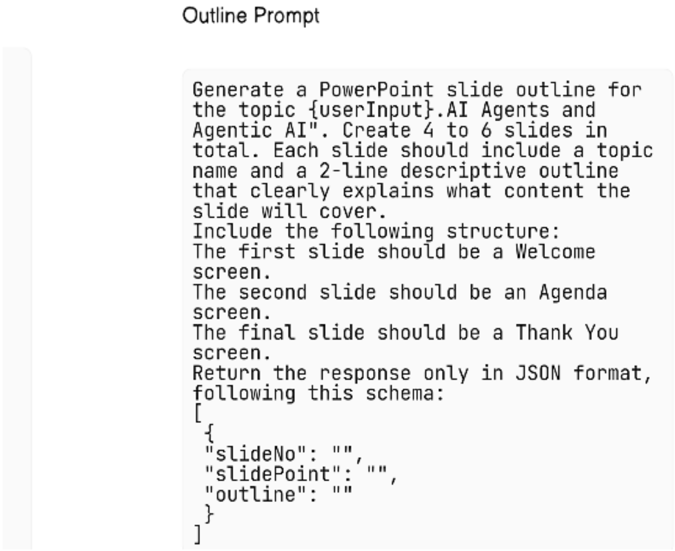
CHAPTER 7
APPENDICES

7.1 Introduction

This chapter provides additional supporting material for the project, including sample prompts, example AI outputs, diagram placeholders, and the GitHub link for full source code reference. These appendices help strengthen the clarity and completeness of the overall project documentation.

7.2 Sample AI Prompts Used

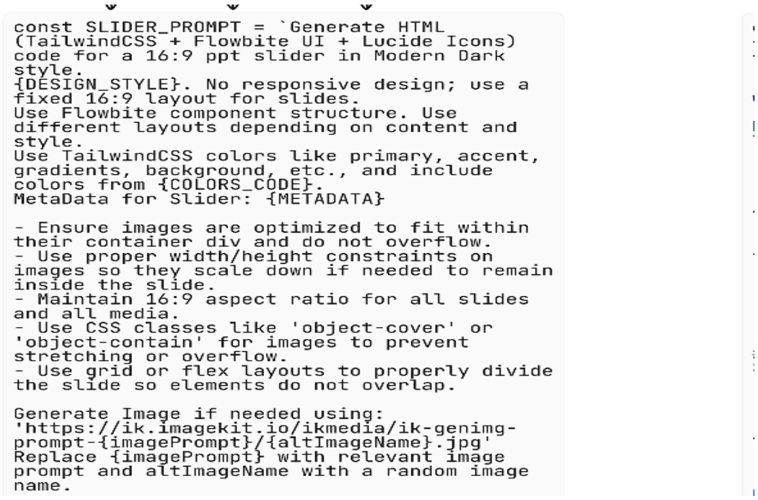
- Outline Prompt:



Outline Prompt

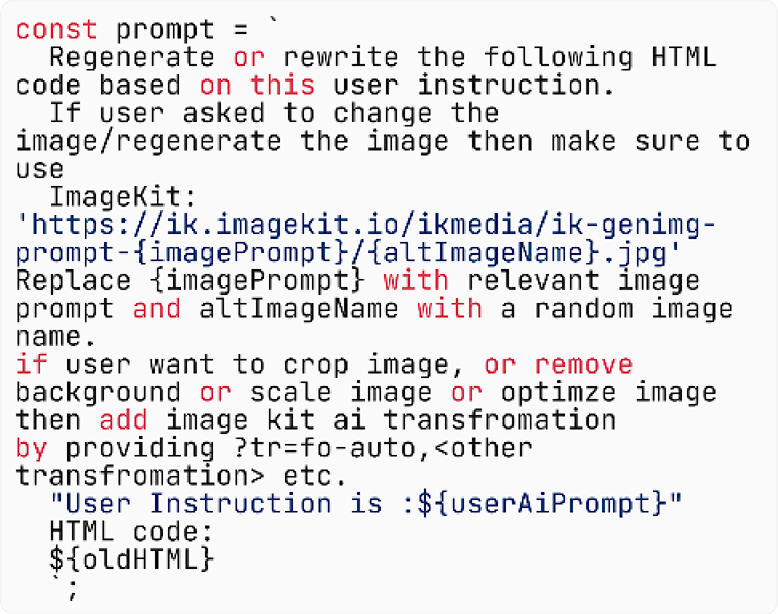
```
Generate a PowerPoint slide outline for
the topic {userInput}.AI Agents and
Agentic AI". Create 4 to 6 slides in
total. Each slide should include a topic
name and a 2-line descriptive outline
that clearly explains what content the
slide will cover.
Include the following structure:
The first slide should be a Welcome
screen.
The second slide should be an Agenda
screen.
The final slide should be a Thank You
screen.
Return the response only in JSON format,
following this schema:
[
 {
 "slideNo": "",
 "slidePoint": "",
 "outline": ""
 }
]
```

- Slide Content Prompt:



```
const SLIDER_PROMPT = `Generate HTML
(TailwindCSS + Flowbite UI + Lucide Icons)
code for a 16:9 ppt slider in Modern Dark
style.
{DESIGN_STYLE}. No responsive design; use a
fixed 16:9 layout for slides.
Use Flowbite component structure. Use
different layouts depending on content and
style.
Use TailwindCSS colors like primary, accent,
gradients, background, etc., and include
colors from {COLORS_CODE}.
MetaData for Slider: {METADATA}

- Ensure images are optimized to fit within
their container div and do not overflow.
- Use proper width/height constraints on
images so they scale down if needed to remain
inside the slide.
- Maintain 16:9 aspect ratio for all slides
and all media.
- Use CSS classes like 'object-cover' or
'object-contain' for images to prevent
stretching or overflow.
- Use grid or flex layouts to properly divide
the slide so elements do not overlap.

Generate Image if needed using:
'https://ik.imagekit.io/ikmedia/ik-genimg-
prompt-{imagePrompt}/{altImageName}.jpg'
Replace {imagePrompt} with relevant image
prompt and altImageName with a random image
name.
```

• Image regeneration Prompt:



```
const prompt = `
  Regenerate or rewrite the following HTML
code based on this user instruction.
  If user asked to change the
image/regenerate the image then make sure to
use
  ImageKit:
'https://ik.imagekit.io/ikmedia/ik-genimg-
prompt-{imagePrompt}/{altImageName}.jpg'
Replace {imagePrompt} with relevant image
prompt and altImageName with a random image
name.
if user want to crop image, or remove
background or scale image or optimze image
then add image kit ai transfromation
by providing ?tr=fo-auto,<other
transfromation> etc.
  "User Instruction is :${userAiPrompt}"
  HTML code:
  ${oldHTML}
  `;
```

7.3 Tools & Technologies Used

Vite -

https://vite.dev

React

https://react.dev

Gemini api

https://ai.google.dev/gemini-api/docs

Firebase

https://console.firebase.google.com

Clerk

https://clerk.com

Shadcn

https://ui.shadcn.com

Node js

https://nodejs.org

Typescript

https://www.typescriptlang.org

Magic ui

https://magicui.design

7.4 GitHub Repository Link

Source Code Repository:

https://github.com/amritpalsingh52/ai-ppt-generator

7.5 workflow Diagram Link

https://app.eraser.io/workspace/t91oGnP2QuYvJRWCZ4gp?origin=share

7.6 Summary

This chapter presented supplementary materials including prompts, sample outputs, essential diagrams, and the project GitHub link. These references provide additional clarity and support for understanding the project's design and implementation.