

# AI-Powered Custom Home Design Assistant - Project Documentation

## Introduction

The **AI-Powered Custom Home Design Assistant** is a tool designed to help users generate personalized home designs based on their preferences. The system utilizes **Google Generative AI (Gemini API)** to generate textual design descriptions and **Pexels API** to fetch relevant design images. The application is built using **Streamlit**, allowing users to interact with the system through a web interface.

## Project Objectives

- Automate the **home design generation process** based on user inputs.
  - Provide **text-based design descriptions** using AI models.
  - Fetch and display **design-related images** for visualization.
  - Offer an **interactive and user-friendly UI** via Streamlit.
- 

## Technology Stack

- **Python** - Primary programming language.
  - **Streamlit** - Web-based UI framework.
  - **Google Generative AI (Gemini API)** - Text generation.
  - **Pexels API** - Image fetching.
  - **Requests Library** - For API calls.
  - **Markdown** - For structured text output.
-

## Prerequisites

Ensure Python is installed on your system, then install the required libraries:

```
pip install streamlit google-generative ai requests
```

### Enabling Gemini API

1. Go to [Gemini API](#) and sign in.
2. Navigate to **API & Services Dashboard**.
3. Click **Enable APIs and Services**.
4. Search for **Gemini API**, enable it, and generate an API key.

## Setting Up Pexels API

1. Sign up at Pexels API.
2. Generate an API key for fetching home design images.

---

## Application Workflow

1. **User Input:** The user provides inputs such as **home style, size, and number of rooms** via the Streamlit UI.
2. **AI Processing:** The input is sent to **Google Generative AI (Gemini API)** to generate a detailed textual home design description.
3. **Image Fetching:** The system queries **Pexels API** to fetch an image related to the design style.
4. **Display Output:** The generated design and fetched image are displayed on the **Streamlit UI**.

---

## Implementation Details

## Building the Streamlit UI

Create an app.py file and add the following code:

```
import streamlit as st
```

```
def main():
```

```
    st.title("Custom Home Design Assistant")
```

```
    style = st.text_input("Enter home style (e.g., Modern, Rustic)")
```

```
    size = st.text_input("Enter home size in sq ft")
```

```
    rooms = st.text_input("Enter the number of rooms")
```

```
    if st.button("Generate Design"):
```

```
        if style and size and rooms:
```

```
            design = generate_design_idea(style, size, rooms)
```

```
            image_url = fetch_image_from_pexels(style)
```

```
            st.markdown(design)
```

```
            if image_url:
```

```
                st.image(image_url, caption=f"{style} Design Inspiration")
```

```
            else:
```

```
                st.warning("No image found for this style.")
```

```
        else:
```

```
            st.warning("Please fill in all fields.")
```

```
if __name__ == "__main__":
```

```
    main()
```

## Integrating Google Generative AI (Gemini API)

Create a design\_generator.py file:

```
import google.generativeai as genai
```

```
genai.configure(api_key="YOUR_GEMINI_API_KEY")
```

```
def generate_design_idea(style, size, rooms):  
    model = genai.GenerativeModel("gemini-1.5-pro")  
    context = f"Create a home design for a {style} style, {size} sq ft home with {rooms} rooms."  
    response = model.generate_content(context)  
    return response.text if response else "Design generation failed."
```

## Fetching Design Images using Pexels API

Create a `image_fetcher.py` file:

```
import requests
```

```
def fetch_image_from_pexels(query):  
    API_KEY = "YOUR_PEXELS_API_KEY"  
    url = f"https://api.pexels.com/v1/search?query={query}&per_page=1"  
    headers = {"Authorization": API_KEY}  
    response = requests.get(url, headers=headers)  
    data = response.json()  
    return data["photos"][0]["src"]["medium"] if data["photos"] else None
```

---

## Running the Application

Run the Streamlit app using:

```
python -m streamlit run app.py
```

---

## Deployment Options

### Local Deployment

- Run Streamlit on your local machine and access the app in your browser.

### Cloud Deployment (Streamlit Sharing)

1. Push your project to **GitHub**.

2. Go to Streamlit Cloud and deploy your repository.
- 

## Output Example

Once the application runs successfully, the output will include:

1. **Generated Home Design Description** (based on user inputs).
2. **Image Matching the Design Style** (fetched from Pexels API).
3. **Interactive UI for Customization.**

### Example Output:

Style: Modern

Size: 2500 sq ft

Rooms: 4

Generated Design:

The modern-style home of 2500 sq ft includes an open floor plan with large glass windows, minimalist interiors, and smart home integrations.

[Displayed Image]

---

## Conclusion

The **AI-Powered Custom Home Design Assistant** is an innovative tool that simplifies the home design process by generating **personalized design descriptions** and **matching images**. It provides an **interactive and user-friendly experience** through **Streamlit**. Future improvements may include:

- **3D visualizations** for more immersive experiences.

- **Floor plan generation** with detailed layouts.
- **Enhanced customization options** to refine generated designs.

This tool effectively bridges the gap between AI-driven home design and user creativity, making home planning more accessible and efficient. 