

ModifAI — Final Project Report

Course Code & Section: CSE299.4 **Project Group No.:** 2

Group Members:

Akram Hossain Apu Khan – 2231090642

Mehedi Hasan Dip – 2232167042

Hacibull Hashan Tosher – 2111626642

1. Introduction

ModifAI is a mobile application that helps users with interior decorating by combining cloud-hosted AI image understanding with real-time social features. Users upload room images and receive personalized design suggestions; they may also communicate with peers through one-to-one and group chats, share media (images, voice messages), and initiate voice and video calls. This report describes the motivation, architecture, implementation, evaluation, security/privacy measures, limitations, and future directions of the project. Interior design is visual and collaborative. ModifAI aims to democratize basic design recommendations by allowing users to upload room photos and receive AI-generated suggestions while enabling discussion and collaboration through integrated communication features. Over the project lifecycle, we implemented authentication, profile management, AI inference, real-time chat, media sharing, and live calls.

2. Motivation and Objectives

Motivation: Provide accessible, quick, and personalized interior design guidance for non-experts.

Objectives:

- Implement a responsive mobile UI with onboarding and profile management.
- Integrate cloud-hosted AI for text and image-based design suggestions.
- Implement one-to-one and group real-time communication (text, media, voice/video).
- Ensure secure authentication and data handling.
- Add basic personalization by learning from user interactions.

3. Related Work

Automated image-based recommendation systems typically rely on CNNs or vision transformers for feature extraction and retrieval/generative layers for suggestions. Mobile apps commonly use cloud inference to offload heavy computation. Real-time mobile media uses WebRTC or commercial SDKs for low-latency audio/video. ModifAI integrates these approaches for interior-decor suggestions and collaborative communication.

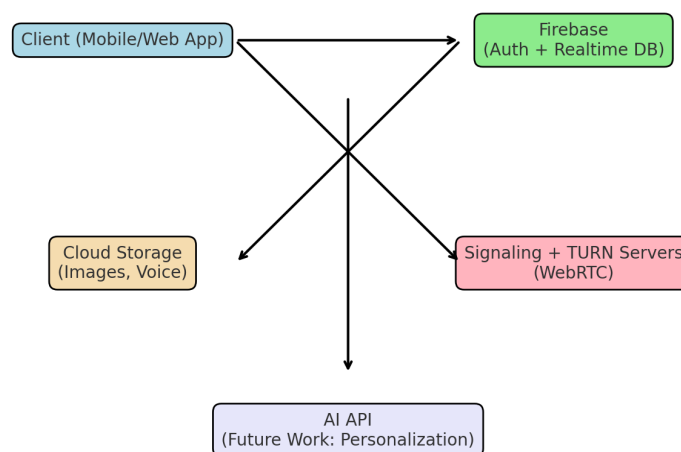
4. System Architecture

Components: Mobile client (Flutter), Firebase Authentication & Firestore, Firebase Cloud Storage, AI inference API (server-side), real-time comms (WebRTC or SDK), and a lightweight backend proxy for API calls.

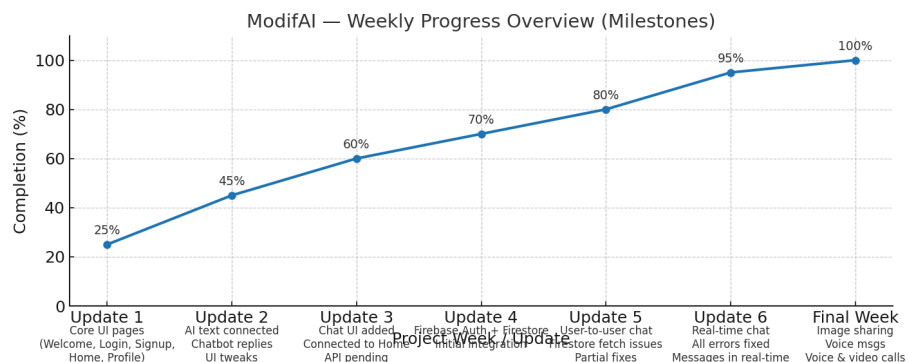
High-level flow:

1. User signs up / signs in via Firebase Authentication.
2. User uploads a room image to Cloud Storage and requests AI inference through the backend proxy.
3. AI returns structured suggestions (text, palettes, overlays) shown in the client.
4. Users interact via chat, group chat, media sharing, and voice/video calls using Firestore listeners and WebRTC signaling.

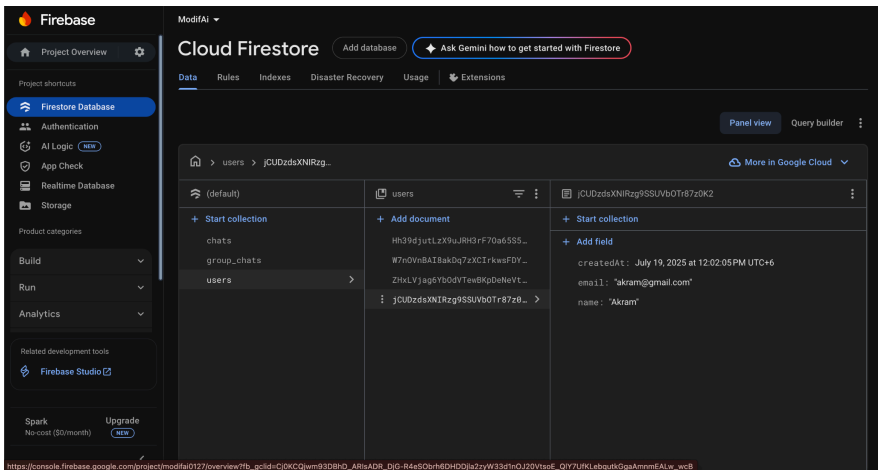
System Architecture Diagram



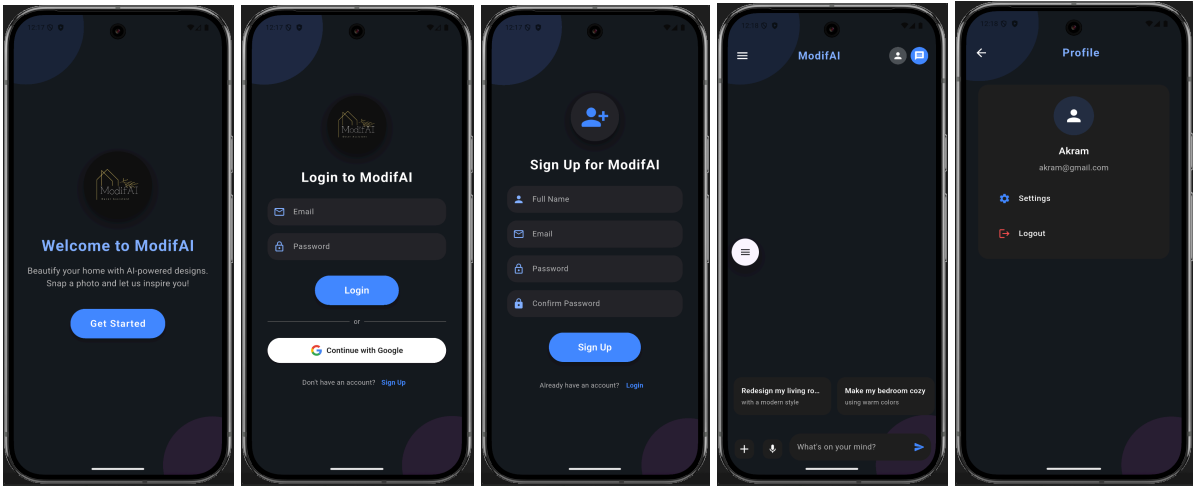
5. Feature Summary (Final Status)



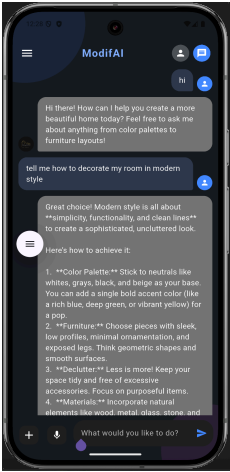
- Onboarding & authentication (Firebase) — Completed



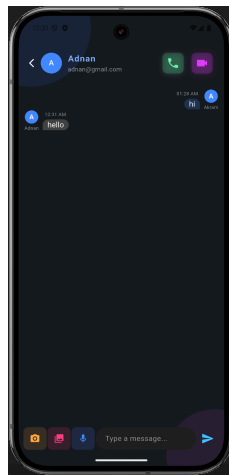
- Welcome, Login, Signup, Home, Profile — Completed



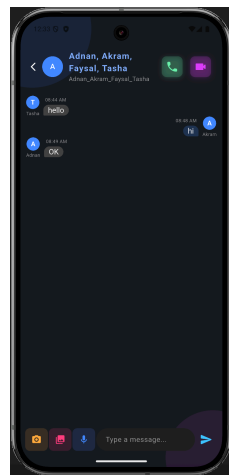
- AI integration (text & image suggestions) — Completed



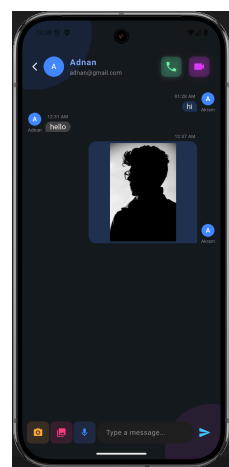
- One-to-one chat — Completed



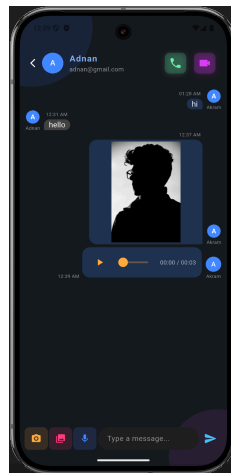
- Group chat — Completed



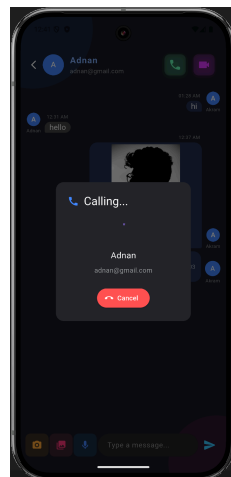
- Chat image sharing — Completed



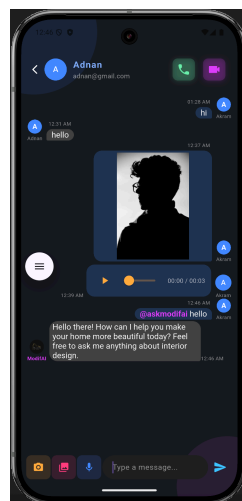
- Voice messages — Completed



- Voice call & Video call — Completed



- Personalization (learning from interactions) — Completed (basic)



6. Implementation Details

6.1 Frontend (Flutter)

- Screens: Welcome, Login, Signup, Home, Camera/Upload, AI Results, Chat List, Chat Conversation, Profile.
- State management: Provider / Riverpod (as implemented in the repo).
- Responsive layout: MediaQuery-based sizing and iterative fixes for earlier UI rendering issues.

6.2 Backend & Services

- Firebase Authentication for user identity.
- Firestore for user profiles, chat threads, messages, and suggestion history.
- Cloud Storage for images and voice files.
- AI proxy server to forward inference requests and protect API keys.
- Real-time signaling via Firestore (or dedicated signaling server) + TURN for reliable WebRTC connections.

6.3 Communication Features

- Realtime listeners provide sub-second chat updates.
- Images and voice clips uploaded to Cloud Storage and referenced in messages.
- Group chat architecture: membership-managed Firestore collection with group thread documents.
- Calls: WebRTC or SDK with ICE/STUN/TURN and robust session handling.

7. User Interface and Experience

The flow prioritizes simplicity: onboarding → profile setup → capture/upload → view AI suggestions → discuss / save designs. AI view shows a short textual suggestion, color swatches, and optional annotated overlays; chats show avatars, timestamps, read receipts, and media previews. Profile stores preferences and saved designs which feed personalization.

8. Testing and Evaluation

Functional testing: Multi-device validation for auth, uploads, AI inference, chats, media, and calls.

Performance observations:

- Image upload: 2–10 s (5–8 MB typical), network-dependent.
- AI latency: 3–12 s depending on cloud service.
- Chat latency: <1 s via Firestore listeners on stable networks.
- Call quality: good on stable Wi-Fi / 4G; TURN improves NAT traversal.

Usability checks: Team users validated AI suggestions and chat flows; personalization improved relevance based on accepted suggestions.

9. Security, Privacy, and Ethics

- API keys are stored server-side; client calls the proxy only.
- Firestore rules restrict access; Cloud Storage rules protect media.
- User images are used only for inference and retained per policy; in-app notice informs users about usage and retention.
- AI suggestions are advisory; clear disclaimers are presented.

10. Challenges and Resolutions

- UI rendering: fixed using responsive widgets and testing.
- API integration: solved with backend proxy and robust retry handling.
- Firestore queries: fixed with indexing and improved listeners.
- RTC connectivity: configured TURN and optimized signaling.
- Personalization: feedback loop implemented to adjust recommendation rankings.

11. Limitations and Future Work

- Improve recommendation explainability and provide reasoning for suggestions.
- Explore on-device lightweight models for offline suggestions.
- Formal user study to quantify benefit and usability.
- Cost and scalability planning for higher user volumes.

12. Conclusion

ModifAI has successfully integrated AI-powered design suggestions with a comprehensive communication suite. The final product supports text and image-based AI suggestions, personalization, real-time one-to-one and group chats, media sharing, voice messages, and audio/video calls. These features together create a collaborative environment for users to explore interior design ideas.

13. References (selected)

1. K. He et al., "Deep Residual Learning for Image Recognition," CVPR, 2016.
2. A. Vaswani et al., "Attention Is All You Need," NeurIPS, 2017.
3. A. Dosovitskiy et al., "Transformers for Image Recognition at Scale," arXiv, 2020.
4. Flutter Documentation. flutter.dev (accessed 2025).
5. Firebase Documentation. firebase.google.com (accessed 2025).
6. WebRTC Project. webrtc.org (accessed 2025).
7. OpenAI API Documentation. platform.openai.com/docs (accessed 2025).
8. ChatGPT (OpenAI) — tool used for development assistance (2025).