FINAL YEAR PROJECT

HANDBOOK

#### Syncmeet



Developed by:

**----------------------**

4051-FBAS/BSSE/F-20

Supervisor by:

**----------------------**

Department of Software Engineering

Faculty of Computing

**INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD 2025**

Department of Software Engineering

International Islamic University, Islamabad.

# FINAL APPROVAL

Dated: 21-01-2025

It is certified that we have read the project report titled as **“AI-Based Career**

**Counseling System for Students”** submitted by Sabahat, registration 4051FBAS/BSSE/F-20 and it is our judgment that this project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the bachelor's degree in Software Engineering.

**Committee**

**External Examiner:**

**Ms. Shaista**

Department of Software Engineering

FoC, International Islamic University, Islamabad

**Internal Examiner:**

**Ms.Raheela Bibi**

Department of Software Engineering

FoC, International Islamic University, Islamabad

**Supervisor:**

**Ms. Raheela Bibi**

Teaching / Research Associate

Department of Software Engineering

FoC, International Islamic University, Islamabad

### DISSERTATION

A dissertation submitted to the Department of Software Engineering, International

Islamic University, Islamabad, as a partial fulfillment of the requirements for the award of the degree of Bachelor of Science in Software Engineering.

# DEDICATION

I dedicate my work to Allah Almighty, the source of all wisdom and knowledge, whose blessings have guided us through the journey of this research. I extend my deepest gratitude to the Prophet Muhammad PBUH, whose teachings continue to inspire and enlighten our paths. I am profoundly thankful to my parents, whose unwavering support and sacrifices have made it possible for us to pursue our academic goals. Their love and encouragement have been our constant source of strength. I also express our sincere appreciation to my teachers and mentors at the university, whose guidance and wisdom have been invaluable. Their dedication to nurturing my potential and enriching my mind has been fundamental in shaping my academic journey.

# DECLARATION

I hereby declare that this Software, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that I have developed this Software entirely based on my personal efforts made under the sincere guidance of my teachers and supervisor.

No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

**Sabahat** 4051-FBAS/BSSE/F-20

# ACKNOWLEDGEMENT

All praise is due to Almighty Allah, whose endless blessings, guidance, and strength enabled me to complete this project successfully.

I am deeply grateful to my parents and well-wishers for their unwavering support and encouragement during my most challenging times. Their sacrifices and prayers have been the foundation of my achievements.

I would like to express my heartfelt gratitude to my esteemed teacher, **Miss Raheela Bibi**, for her invaluable guidance, encouragement, and continuous support throughout this project. Her mentorship has been instrumental in enhancing my knowledge and helping me navigate this journey with confidence.

# PROJECT IN BRIEF

|  |  |
| --- | --- |
| **Field** | **Details** |
| **Project Title** | Syncmeet |
| **Objective** | A web-based application utilizing AI and machine learning to guide students in career decision-making. The system evaluates students' learning background, technical skills, and performance to provide personalized job recommendations and career insights. |
| **Undertaken By** | Sabahat |
| **Supervised By** | Ms. Raheela (Teaching/Research Associate) |
| **Start Date** | July 2024 |
| **Completion Date** | January 2025 |
| **Development Tools** | - Visual Code 2022 (Python 3.9.12, 64-bit) Vercel |
| **Programming Libraries** | - React, JavaScript, Typescript, NodeJs, |
| **System Specifications** | - **Processor:** Intel Core i5 (8th Gen) or higher- **RAM:** 8 GB minimum- **Storage:** 256 GB SSD or higher- **Operating System:** Windows 10/11 (64-bit)- **Browser:** Google Chrome, Mozilla Firefox |

# ABSTRACT

In the modern era of remote collaboration, seamless communication and efficient meeting management are essential for productivity. Syncmeet is an AI-powered online conferencing platform built with React (TypeScript) for the frontend and Node.js for the backend, deployed on Vercel for scalability. It integrates OpenAI’s advanced natural language processing to convert speech into text in real time, generate meeting minutes, and summarize discussions for quick reference. Syncmeet also offers interactive features such as collaborative whiteboards, personal and group meeting scheduling, online polls for decision-making, and secure authentication/session management through Clerk. The platform supports recording and storing past meetings for reference and generates detailed reports for participants. Email notifications and meeting updates are delivered using SMTP services, ensuring timely communication. By combining AI-driven insights with a robust, user-friendly interface, Syncmeet enhances productivity, fosters collaboration, and streamlines the entire meeting lifecycle from scheduling to follow-up.

**Keywords:** Online Conferencing, AI Meeting Summaries, Speech-to-Text, OpenAI, React, Node.js, Whiteboard, Polls, Scheduling, Clerk, Vercel.

**Table of Contents**

[FINAL APPROVAL 2](#_Toc205823653)

[DISSERTATION 3](#_Toc205823654)

[DEDICATION 4](#_Toc205823655)

[DECLARATION 5](#_Toc205823656)

[ACKNOWLEDGEMENT 6](#_Toc205823657)

[PROJECT IN BRIEF 7](#_Toc205823658)

[ABSTRACT 8](#_Toc205823659)

[CHAPTER 1 12](#_Toc205823660)

[1. INTRODUCTION 12](#_Toc205823661)

[1.1. Overview of Syncmeet 12](#_Toc205823662)

[1.2. Limitations of Current Systems 12](#_Toc205823663)

[1.4. Innovative AI-Driven Approach 12](#_Toc205823664)

[1.5. Key Functionalities of the System 13](#_Toc205823665)

[CHAPTER 2 14](#_Toc205823666)

[2. SYSTEM ANALYSIS 14](#_Toc205823667)

[2.1. Overview of System Behavior 14](#_Toc205823668)

[2.2. Interaction Models 14](#_Toc205823669)

[2.3. Graphical Use Case Representation 14](#_Toc205823670)

[2.4. Concise Use Case Summaries 15](#_Toc205823671)

[2.5. Detailed Use Case Analysis 15](#_Toc205823672)

[2.7. Conceptual Domain Mapping 16](#_Toc205823673)

[2.8. Workflow Representations (Activity Diagrams) 16](#_Toc205823674)

[Figure 1.1: Workflow Diagram 17](#_Toc205823675)

[CHAPTER 3 18](#_Toc205823676)

[SYSTEM DESIGN 18](#_Toc205823677)

[3.1. System Design 18](#_Toc205823678)

[3.2. Sequence Diagram 18](#_Toc205823679)

[3.3. Class Diagram 19](#_Toc205823680)

[Figure 3.3: Class Diagram of Syncmeet 19](#_Toc205823681)

[3.4. ER Diagram 20](#_Toc205823682)

[Figure 3.4: ER Diagram of Syncmeet 20](#_Toc205823683)

[4. CHAPTER 23](#_Toc205823684)

[4. IMPLEMENTATION 23](#_Toc205823685)

[4.1. Implementation Overview 23](#_Toc205823686)

[4.2. Module-wise Implementation 23](#_Toc205823687)

[4.2.1. Authentication & Session Management 23](#_Toc205823688)

[4.2.2. Meeting Scheduling & Invitations 23](#_Toc205823689)

[4.2.3. Real-Time Video & Audio Conferencing 23](#_Toc205823690)

[4.2.4. AI-Powered Transcription & Summarization 24](#_Toc205823691)

[4.2.5. Whiteboard Collaboration 24](#_Toc205823692)

[4.2.6. Online Polling & Decision Making 24](#_Toc205823693)

[4.2.7. Meeting Recording & Reports 24](#_Toc205823694)

[4.3. Technology Stack Used 24](#_Toc205823695)

[4.4. Database Implementation 25](#_Toc205823696)

[4.5. Security Implementation 25](#_Toc205823697)

[4.6. API Implementation 25](#_Toc205823698)

[4.7. User Interface Implementation 25](#_Toc205823699)

[4.8. Error Handling & Recovery 25](#_Toc205823700)

[4.9. Testing & Debugging 26](#_Toc205823701)

[4.10. Usecase Diagram 26](#_Toc205823702)

[Figure 4.6: Usecase Diagram of Syncmeet 27](#_Toc205823703)

[4.11. Implementation Diagram (Deployment Diagram) 27](#_Toc205823704)

[Figure 4.7: Deployment Diagram of Syncmeet 27](#_Toc205823705)

[5. CHAPTER 28](#_Toc205823706)

[5 SYSTEM TESTING 28](#_Toc205823707)

[5.1. Testing 28](#_Toc205823708)

[5.1.1. Unit Testing 28](#_Toc205823709)

[5.1.2. Integration Testing 28](#_Toc205823710)

[5.1.3. Functional Testing 28](#_Toc205823711)

[5.1.4. Performance Testing 28](#_Toc205823712)

[5.1.5. Security Testing 29](#_Toc205823713)

[5.1.6. User Acceptance Testing (UAT) 29](#_Toc205823714)

[5.2. Test Cases 29](#_Toc205823715)

[6. CHAPTER 31](#_Toc205823716)

[6 CONCLUSION 31](#_Toc205823717)

[6.1. Conclusion 31](#_Toc205823718)

[APPENDIX-A 32](#_Toc205823719)

[Figure 6.1: Login of Syncmeet 32](#_Toc205823720)

[Figure 6.2: Signup of Syncmeet 32](#_Toc205823721)

[Figure 6.3: Homepage of Syncmeet 33](#_Toc205823722)

[Figure 6.4: Upcoming Meetings of Syncmeet 34](#_Toc205823723)

[Figure 6.4: Upcomming meetings of Syncmeet 34](#_Toc205823724)

[Figure 6.5: Previous Meetings of Syncmeet 35](#_Toc205823725)

[Figure 6.6: Recordings of Syncmeet 35](#_Toc205823726)

[Figure 6.6: Recordings of Syncmeet 35](#_Toc205823727)

[Figure 6.7: Personal Rooms of Syncmeet 36](#_Toc205823728)

[Figure 6.7: Personal Rooms of Syncmeet 36](#_Toc205823729)

[Figure 6.8: Make Poll 36](#_Toc205823730)

[Figure 6.8: Make Poll 36](#_Toc205823731)

[REFERENCES 37](#_Toc205823732)

# CHAPTER 1

# INTRODUCTION

1.1. Overview of Syncmeet  
Syncmeet is an AI-powered online conferencing and collaboration platform designed to enhance virtual meetings through intelligent automation. It provides seamless video conferencing, real-time speech-to-text transcription, AI-generated meeting summaries, collaborative whiteboards, scheduling tools, and online polls. The system ensures secure authentication through Clerk, enables meeting recording for future reference, and delivers timely updates via SMTP-based email notifications. Syncmeet bridges the gap between remote teams by offering a complete meeting lifecycle solution — from scheduling to post-meeting reporting — all accessible through a modern React TypeScript interface and Node.js backend.

1.2. Limitations of Current Systems  
Despite the availability of various online conferencing tools, most existing platforms lack deep AI integration for automated meeting minutes, real-time transcription, and instant summaries. Many tools provide only basic recording features without converting them into actionable insights. Scheduling, whiteboard collaboration, and polling are often provided as separate tools, causing workflow fragmentation. Additionally, some platforms require heavy installations, have complex user interfaces, or depend on multiple third-party integrations, leading to inefficiency and reduced user adoption.

1.3. Challenges in Online Meeting Management  
Modern online meetings face multiple challenges such as:

* Information loss due to manual note-taking.
* Difficulty in extracting key points from lengthy discussions.
* Managing schedules across participants in different time zones.
* Engaging participants effectively in decision-making processes.
* Securely handling meeting recordings and participant data.
* Reducing dependency on multiple disconnected tools for whiteboards, polls, and scheduling.

1.4. Innovative AI-Driven Approach  
Syncmeet integrates OpenAI’s advanced speech recognition and natural language processing models to convert voice into text in real-time and generate concise, accurate meeting summaries. This eliminates the need for manual note-taking and ensures that important decisions and action items are documented instantly. AI-powered analysis also helps identify meeting highlights, action points, and participant inputs, enabling faster decision-making and improving productivity. The unified platform combines scheduling, collaboration, and engagement tools, offering a seamless all-in-one meeting experience.

## ****1.5. Key Functionalities of the System****

* **Real-Time Speech-to-Text Conversion** using OpenAI APIs.
* **AI-Generated Meeting Summaries** and automated minutes of meetings.
* **Secure Authentication & Session Management** with Clerk.
* **Meeting Scheduling & Reminders** with email notifications via SMTP.
* **Interactive Whiteboard** for collaborative discussions.
* **Online Polling** to gather participant preferences and make decisions.
* **Meeting Recording** for future playback and report generation.
* **Personal & Group Meeting Management** within a unified interface.

#### ****1.6. Technological Framework & Tools****

* **Frontend:** React with TypeScript for a responsive, user-friendly interface.
* **Backend:** Node.js for handling server-side logic and API integration.
* **AI & NLP:** OpenAI APIs for transcription and summarization.
* **Authentication & Sessions:** Clerk for secure login and user management.
* **Deployment:** Vercel for scalable, cloud-based hosting.
* **Email Services:** SMTP protocol for meeting invites, reminders, and updates.
* **Collaboration Tools:** Whiteboard and polling modules for interactive sessions.

# CHAPTER 2

# 2. SYSTEM ANALYSIS

## ****2.1. Overview of System Behavior****

Syncmeet operates as a real-time online conferencing and collaboration platform that integrates AI-powered transcription, summarization, and meeting management. The system allows authenticated users to schedule, host, and participate in meetings with features such as whiteboard collaboration, online polls, and live chat. Audio from meetings is processed using OpenAI APIs for real-time speech-to-text conversion and summary generation. The backend manages scheduling, meeting data storage, email notifications, and authentication, while the frontend ensures an intuitive user experience. The system is designed to handle multiple concurrent meetings while maintaining performance and security.

## ****2.2. Interaction Models****

Syncmeet’s core interaction model involves:

1. **User Authentication & Authorization** – Users sign up or log in via Clerk, ensuring secure access.
2. **Meeting Scheduling** – Hosts create meetings with time, date, and invitees.
3. **Invitation & Notification** – SMTP-based email system sends meeting details and reminders.
4. **Meeting Participation** – Attendees join via a web interface, with access to video/audio, whiteboard, polls, and chat.
5. **AI Processing** – Audio is transcribed in real-time; summaries are generated during or after the meeting.
6. **Post-Meeting Reports** – Summaries, transcripts, and recordings are available for participants.

## ****2.3. Graphical Use Case Representation****

**Figure 2.1:** Use Case Diagram of AI-Based Online Conferencing System (Syncmeet)

**Actors:**

* **Host** (creates and manages meetings)
* **Participant** (joins and interacts during meetings)
* **AI Engine (OpenAI)** (handles transcription & summaries)
* **SMTP Email Service** (sends invitations and notifications)
* **Clerk Authentication** (manages secure login/sessions)

**Core Use Cases:**

* Schedule Meeting
* Send Invitations
* Join Meeting
* Start Recording
* Real-Time Speech-to-Text
* Generate Meeting Summary
* Use Whiteboard
* Participate in Poll
* View Meeting Reports

## ****2.4. Concise Use Case Summaries****

|  |  |
| --- | --- |
| **Use Case** | **Description** |
| Schedule Meeting | Host creates a meeting with date, time, and invitees. |
| Send Invitations | SMTP service sends meeting links to participants. |
| Join Meeting | Participants connect via web interface. |
| Start Recording | Meeting is recorded for future playback. |
| Real-Time Transcription | AI converts voice to text during meeting. |
| Generate Summary | AI summarizes meeting discussion. |
| Use Whiteboard | Participants collaborate on a shared board. |
| Participate in Poll | Participants vote on meeting-related decisions. |
| View Meeting Reports | Users access summaries, transcripts, and recordings. |

## ****2.5. Detailed Use Case Analysis****

**Use Case Name:** Real-Time Transcription

* **Actors:** Host, Participant, AI Engine
* **Description:** Converts spoken content during a meeting into live text using OpenAI APIs.
* **Preconditions:** Meeting must be in progress; audio input is available.
* **Postconditions:** Transcript is generated and stored in database.
* **Exceptions:** Poor audio quality or unstable internet may reduce accuracy.

**Use Case Name:** Generate Meeting Summary

* **Actors:** AI Engine, Host
* **Description:** AI processes the meeting transcript to produce a summary and key action points.
* **Preconditions:** Transcript is available.
* **Postconditions:** Summary is displayed and stored for future reference.
* **Exceptions:** AI may produce less accurate results if transcript has errors.

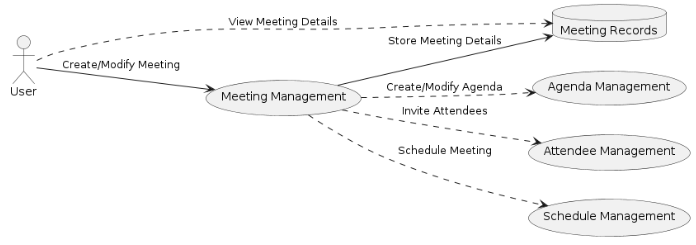
## ****2.7. Conceptual Domain Mapping****

|  |  |  |
| --- | --- | --- |
| **Entity** | **Attributes** | **Description** |
| User | userID, name, email, role | Represents host or participant. |
| Meeting | meetingID, title, date, time, hostID | Details of scheduled meeting. |
| Transcript | transcriptID, meetingID, content | Stores AI-generated transcription. |
| Summary | summaryID, meetingID, content | Stores AI-generated summary. |
| WhiteboardSession | sessionID, meetingID, data | Stores collaborative board content. |
| Poll | pollID, meetingID, question, options, results | Stores poll data. |

## ****2.8. Workflow Representations (Activity Diagrams)****

**Activity Diagram – Meeting Lifecycle in Syncmeet:**

1. User logs in via Clerk.
2. Host schedules meeting and invites participants.
3. Email notification sent via SMTP.
4. Participants join meeting.
5. AI transcribes and summarizes discussions.
6. Whiteboard and polls used for collaboration.
7. Meeting ends; recording and summaries are stored.
8. Email report sent to participants.



### Figure 1.1: Workflow Diagram

# CHAPTER 3

# SYSTEM DESIGN

## ****3.1. System Design****

The design of Syncmeet focuses on creating a scalable, secure, and user-friendly online conferencing platform that integrates AI capabilities for real-time transcription, meeting summarization, and collaboration. The system architecture follows a **client-server model** where the React (TypeScript) frontend communicates with the Node.js backend through REST APIs and WebSocket connections for real-time data exchange.

The **frontend** handles user interactions, meeting UI components, whiteboard tools, and displaying AI-generated content. The **backend** manages authentication, meeting scheduling, AI API integration, real-time transcription handling, and database operations. AI-powered features use **OpenAI APIs** for speech-to-text conversion and summary generation. **SMTP services** handle email notifications, and **Clerk** manages authentication and session security.

## ****3.2. Sequence Diagram****

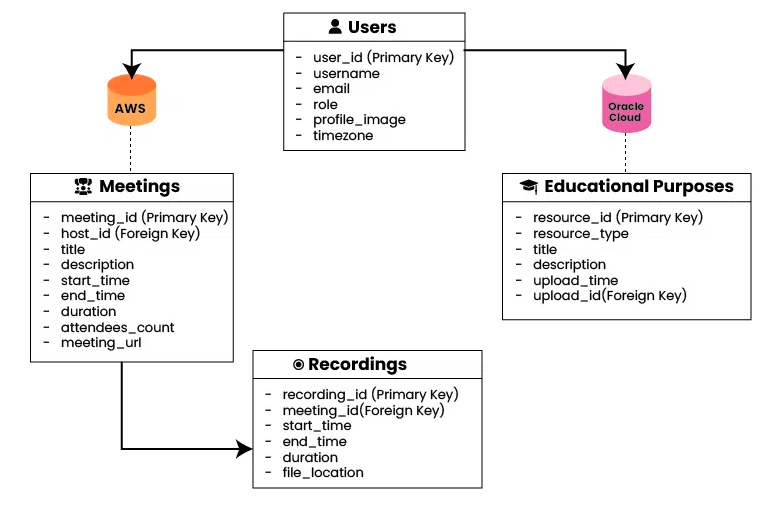
**Figure 3.2: Sequence Diagram of a Syncmeet Session**

**Description:** This diagram outlines the sequence of interactions when a meeting is scheduled, conducted, and summarized.

**Flow:**

1. **User (Host)** logs in via Clerk.
2. Host schedules a meeting (details stored in DB).
3. SMTP service sends invitations to participants.
4. Participants join meeting via frontend interface.
5. Real-time audio is captured and sent to AI engine for transcription.
6. AI returns live transcript to participants.
7. At meeting end, AI generates summary and stores it in DB.
8. Email notification with meeting summary/report is sent to attendees.

## ****3.3. Class Diagram****

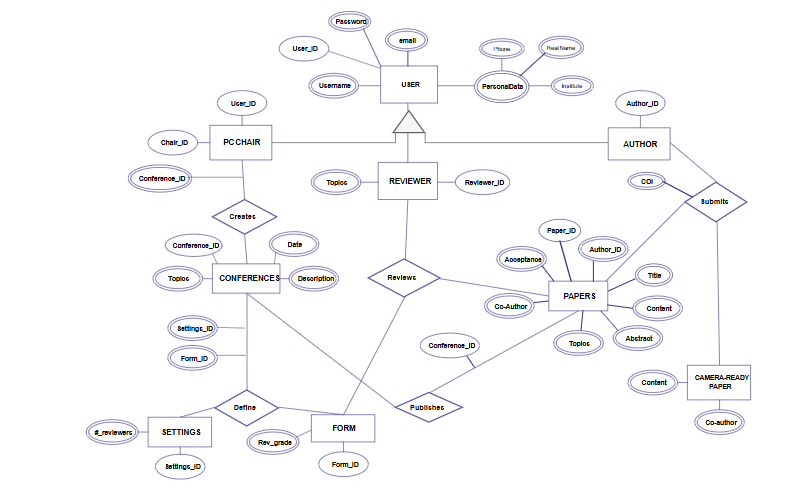


### **Figure 3.3: Class Diagram of Syncmeet**

**Main Classes:**

|  |  |  |
| --- | --- | --- |
| **Class** | **Attributes** | **Methods** |
| **User** | userID, name, email, password, role | login(), logout(), updateProfile() |
| **Meeting** | meetingID, title, date, time, hostID, participants | scheduleMeeting(), joinMeeting(), endMeeting() |
| **Transcript** | transcriptID, meetingID, content | addTranscript(), getTranscript() |
| **Summary** | summaryID, meetingID, content | generateSummary(), getSummary() |
| **Whiteboard** | boardID, meetingID, content | draw(), erase(), saveBoard() |
| **Poll** | pollID, meetingID, question, options, results | createPoll(), vote(), getResults() |
| **EmailService** | smtpConfig | sendEmail(), sendReminder() |
| **AIService** | apiKey, model | transcribeAudio(), summarizeText() |

## ****3.4. ER Diagram****



### **Figure 3.4: ER Diagram of Syncmeet**

**Entities and Relationships:**

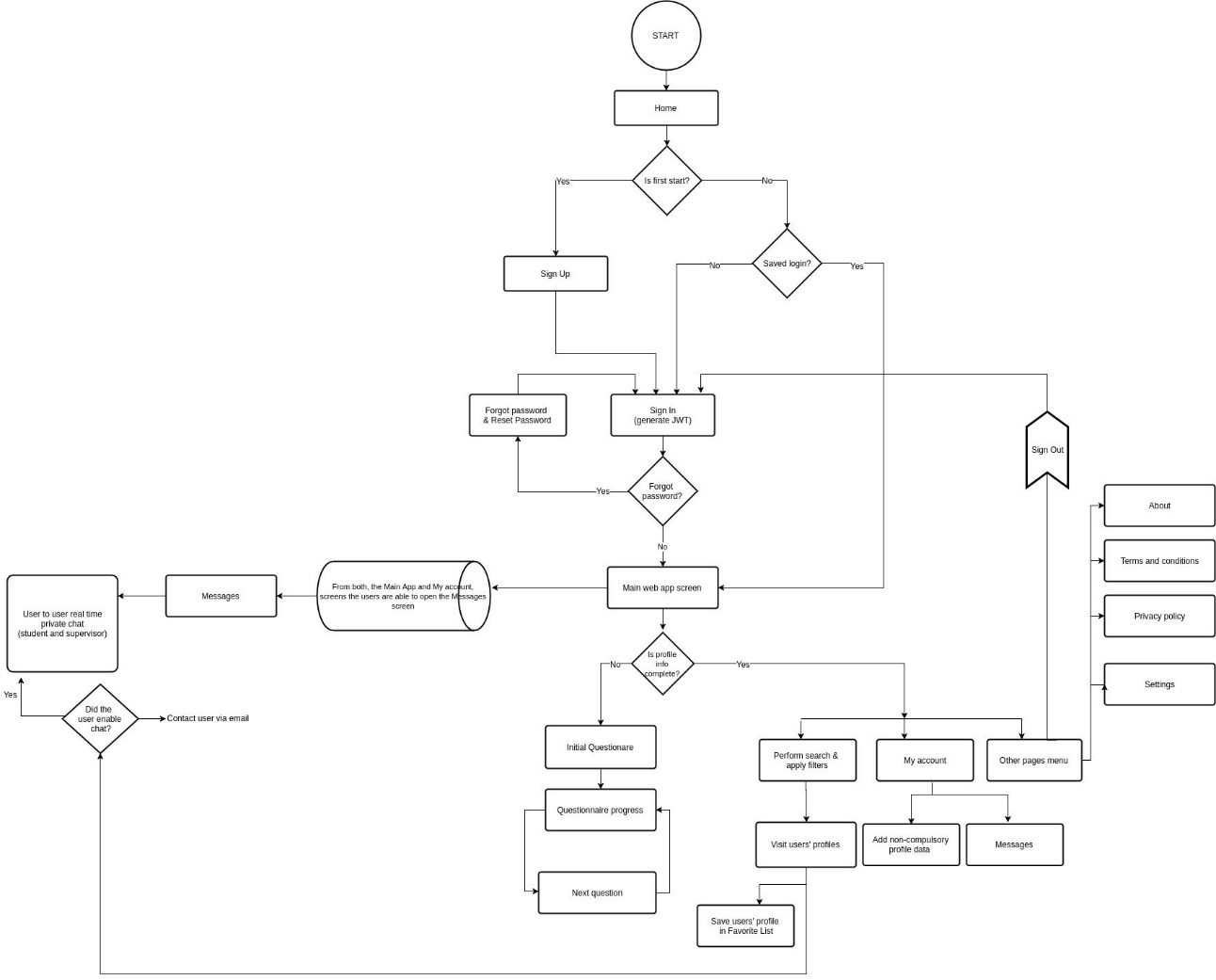
* **User (userID, name, email, password, role)**
* **Meeting (meetingID, title, date, time, hostID)** — **hosted by** → User
* **Participant (participantID, meetingID, userID)** — **participates in** → Meeting
* **Transcript (transcriptID, meetingID, content)** — **belongs to** → Meeting
* **Summary (summaryID, meetingID, content)** — **generated from** → Transcript
* **Whiteboard (boardID, meetingID, content)** — **used in** → Meeting
* **Poll (pollID, meetingID, question, options, results)** — **linked to** → Meeting

## ****3.5. Application Flow****

The application flow of the Career Guidance System illustrates the sequence of interactions between the user and the system, from the initial access to the final recommendations and career resources. It ensures a smooth and logical progression of steps, providing a user-friendly experience.

**Step-by-Step Flow:**

1. **User Access & Authentication**
   * The user accesses the system through a web or mobile interface.
   * If the user is new, they complete the registration process by providing basic details such as name, email, and password.
   * Returning users log in using their credentials.
   * Authentication is handled securely to prevent unauthorized access.
2. **Profile Setup**
   * The user fills in personal, educational, and professional information.
   * Details include education level, interests, skills, work experience, and preferred industries.
   * The system may integrate with existing profiles (e.g., LinkedIn) for quick setup.
3. **Assessment & Data Collection**
   * The system presents an AI-driven career assessment questionnaire.
   * Questions focus on personality type, skills, values, work preferences, and aptitude.
   * Responses are stored in the database for analysis.
4. **AI-Based Analysis**
   * The AI model analyzes user inputs using machine learning algorithms.
   * It compares the user profile with career data sets, industry trends, and skill demand forecasts.
   * The system identifies potential career paths aligned with the user’s profile.
5. **Recommendation Generation**
   * The system generates personalized career recommendations.
   * Each recommendation includes details such as career description, required qualifications, growth prospects, and average salary.
   * Recommendations are ranked based on the user’s compatibility score.
6. **Resource & Guidance Access**
   * The user is provided with links to learning resources, job portals, and mentorship opportunities.
   * The system offers skill improvement suggestions and online course recommendations.
7. **Progress Tracking & Feedback**
   * Users can track their career exploration progress over time.
   * The system collects feedback to refine future recommendations.
8. **Logout / Session End**
   * Once the user completes their session, they can log out securely.



### **Figure 3.4: Application Flow**

# 4. CHAPTER

# 4. IMPLEMENTATION

## ****4.1. Implementation Overview****

The implementation of Syncmeet focuses on integrating multiple functional modules to create an AI-powered online conferencing and collaboration platform. The project is divided into **frontend**, **backend**, **AI integration**, and **communication services**.

The **frontend** is developed using **React (TypeScript)** for its flexibility, component reusability, and strong type-checking capabilities. It provides the user interface for scheduling meetings, joining live sessions, interacting with whiteboards, participating in polls, and viewing AI-generated meeting summaries.

The **backend** is implemented in **Node.js** to handle server-side logic, manage database operations, and integrate third-party services such as OpenAI APIs, Clerk authentication, and SMTP for email communication. Real-time communication is enabled using **WebSocket** for low-latency audio/video streaming and collaborative tools.

## ****4.2. Module-wise Implementation****

### ****4.2.1. Authentication & Session Management****

* Implemented using **Clerk**, which handles secure login, signup, and session persistence.
* Role-based access is implemented to differentiate between hosts and participants.
* Token-based authentication ensures secure API communication.

### ****4.2.2. Meeting Scheduling & Invitations****

* Hosts can create meetings by specifying date, time, and invited participants.
* Meeting details are stored in the database and a unique meeting ID/link is generated.
* SMTP-based email service sends invitations and reminders automatically.

### ****4.2.3. Real-Time Video & Audio Conferencing****

* WebRTC handles peer-to-peer audio/video streaming.
* Media streams are captured in real-time for transcription.
* Network resilience features allow recovery from temporary disconnections.

### ****4.2.4. AI-Powered Transcription & Summarization****

* **OpenAI Whisper API** is used for speech-to-text conversion.
* Transcription is processed in real-time and displayed to participants.
* AI summarization algorithms extract key discussion points and action items.

### ****4.2.5. Whiteboard Collaboration****

* A shared whiteboard module allows multiple participants to draw, write, and annotate simultaneously.
* Whiteboard data is synchronized via WebSocket channels.

### ****4.2.6. Online Polling & Decision Making****

* Hosts can create polls with multiple-choice options.
* Poll results are displayed in real-time to all participants.

### ****4.2.7. Meeting Recording & Reports****

* Full video/audio recording of meetings is supported.
* Post-meeting reports include transcript, summary, and action items.
* Reports are emailed to participants automatically.

## ****4.3. Technology Stack Used****

|  |  |
| --- | --- |
| **Layer** | **Technology** |
| **Frontend** | React (TypeScript), Tailwind CSS |
| **Backend** | Node.js, Express.js |
| **AI Services** | OpenAI Whisper API, OpenAI GPT models |
| **Auth** | Clerk |
| **Database** | MongoDB / PostgreSQL |
| **Email** | SMTP (Nodemailer) |
| **Real-Time** | WebRTC, WebSockets |
| **Deployment** | Vercel (Frontend), Railway/Render/Heroku (Backend) |

## ****4.4. Database Implementation****

* Clerk stores meeting details, user profiles, transcripts, and summaries.
* Indexing is used to optimize meeting retrieval speed.
* Relationships between entities such as Users, Meetings, and Polls are maintained for easy reporting.

## ****4.5. Security Implementation****

* **HTTPS** encryption for secure data transmission.
* Role-based access control (RBAC) to limit permissions.
* Token expiration and refresh mechanisms to prevent session hijacking.

## ****4.6. API Implementation****

* **REST APIs** handle CRUD operations for meetings, polls, and whiteboard sessions.
* **WebSocket APIs** manage real-time whiteboard updates and poll results.
* **OpenAI API** calls are optimized to avoid unnecessary token usage.

## ****4.7. User Interface Implementation****

* **Light & Dark Themes** for accessibility.
* Responsive design for desktop and mobile compatibility.
* Minimal UI latency to enhance user engagement.

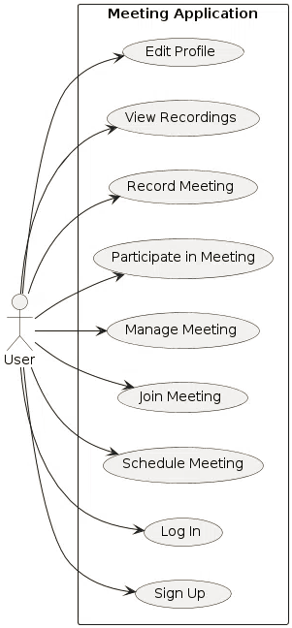
## ****4.8. Error Handling & Recovery****

* Retry mechanisms for failed API calls.
* Offline caching for important meeting data.
* Graceful degradation when AI services are temporarily unavailable.

## ****4.9. Testing & Debugging****

* **Unit Testing:** Jest for frontend logic and Mocha/Chai for backend APIs.
* **Integration Testing:** Ensures AI transcription works seamlessly with real-time video/audio streams.
* **User Acceptance Testing (UAT):** Conducted with sample users to refine the UX.

## ****4.10. Usecase Diagram****

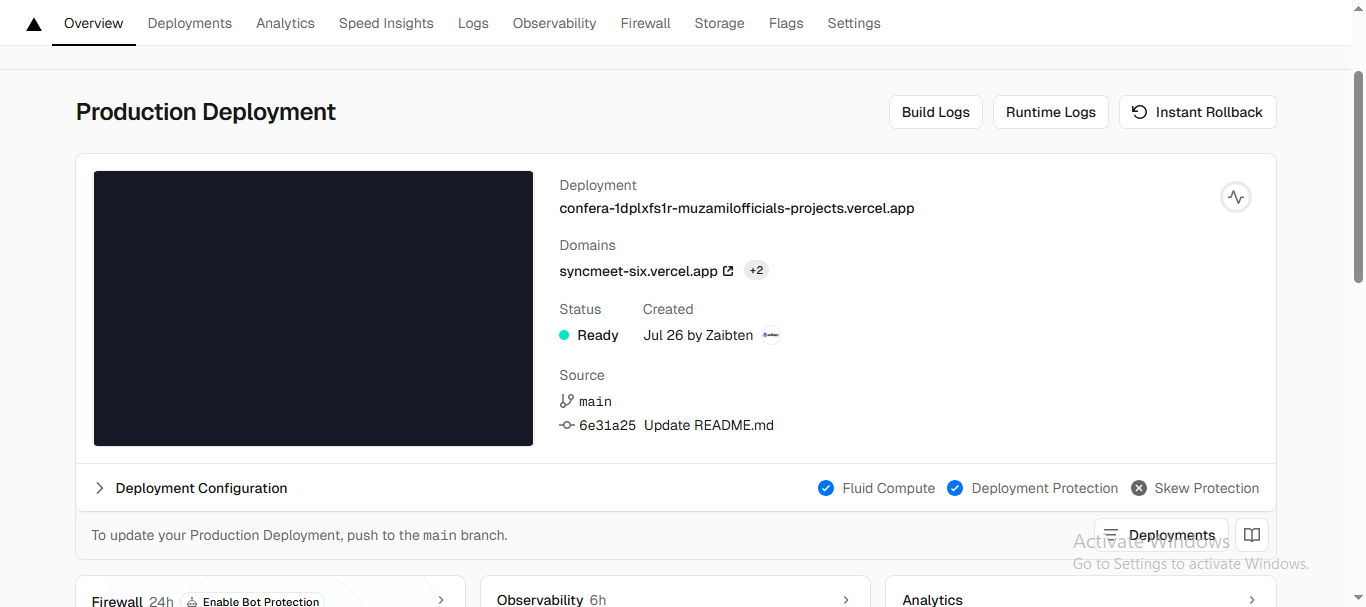


### ****Figure 4.6:**** Usecase Diagram of Syncmeet

**Components:**

* **Frontend UI Component** (React TypeScript)
* **Auth Component** (Clerk)
* **AI Processing Component** (OpenAI)
* **Meeting Management Component** (Node.js)
* **Whiteboard Component** (WebSocket-based)
* **Polling Component**
* **Email Notification Component** (SMTP)
* **Database Component** (MongoDB)

## ****4.11. Implementation Diagram (Deployment Diagram)****



### ****Figure 4.7:**** Deployment Diagram of Syncmeet

**Deployment Setup:**

* **Client Side:** React application deployed on **Vercel**.
* **Backend:** Node.js server hosted on **Railway** or **Render**, connected to MongoDB Atlas.
* **AI Services:** API calls to OpenAI for transcription & summarization.
* **Email Service:** SMTP server for notifications.
* **Database:** MongoDB Atlas cloud database for persistent storage.

# 5. CHAPTER

# 5 SYSTEM TESTING

## ****5.1. Testing****

System testing is an essential phase in ensuring the reliability, accuracy, and usability of **Syncmeet**. The testing process for this project was carried out in multiple stages to identify and eliminate functional, performance, and security issues.

The following testing methodologies were applied:

### ****5.1.1. Unit Testing****

* **Purpose:** To verify that each individual module (e.g., meeting scheduling, transcription, email notifications) functions correctly in isolation.
* **Tools Used:** Jest (Frontend), Mocha & Chai (Backend).
* **Example:** Testing if a scheduled meeting correctly stores in the database with all required details.

### ****5.1.2. Integration Testing****

* **Purpose:** To check that combined modules (e.g., AI transcription with real-time meeting streaming) work together without conflicts.
* **Example:** Testing the workflow where an ongoing meeting triggers transcription, which is stored and later summarized automatically.

### ****5.1.3. Functional Testing****

* **Purpose:** To ensure all user requirements and use cases are fulfilled.
* **Example:** Verifying that a poll created by the host is visible to all participants and accepts valid responses.

### ****5.1.4. Performance Testing****

* **Purpose:** To test the system under varying loads to ensure stable performance.
* **Tools:** Apache JMeter.
* **Example:** Testing server response times when 50+ users join a meeting simultaneously.

### ****5.1.5. Security Testing****

* **Purpose:** To ensure user data and meetings are secure from unauthorized access.
* **Example:** Attempting SQL injection, API token misuse, and session hijacking to confirm prevention mechanisms.

### ****5.1.6. User Acceptance Testing (UAT)****

* **Purpose:** To validate the system with real-world scenarios from end-users.
* **Process:** A group of test users scheduled meetings, interacted via the whiteboard, created polls, and checked AI-generated summaries.
* **Result:** Positive feedback was received for ease of use, accuracy of summaries, and overall system speed.

## ****5.2. Test Cases****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| TC\_001 | Login with valid credentials | Email, Password | Redirect to dashboard | Redirect to dashboard | Pass |
| TC\_002 | Login with invalid credentials | Email, Wrong Password | Error message displayed | Error message displayed | Pass |
| TC\_003 | Schedule a meeting | Title, Date, Time, Participants | Meeting saved in DB and email sent | Meeting saved and email sent | Pass |
| TC\_004 | Join a meeting | Meeting Link | Connect to live meeting | Connected successfully | Pass |
| TC\_005 | Start real-time transcription | Live audio | Text transcript appears | Transcript generated live | Pass |
| TC\_006 | Generate AI summary after meeting | Transcript ID | Summary text displayed | Summary displayed | Pass |
| TC\_007 | Create a poll | Question, Options | Poll visible to participants | Poll visible and functional | Pass |
| TC\_008 | Use whiteboard collaboratively | Drawing actions | Whiteboard updates in real-time | Updates displayed instantly | Pass |
| TC\_009 | Send meeting reminder email | Meeting ID | Email sent to participants | Email received | Pass |
| TC\_010 | Unauthorized user tries to join meeting | Meeting Link without permission | Access denied message | Access denied message | Pass |

# 6. CHAPTER

# 6 CONCLUSION

## ****6.1. Conclusion****

The development of **Syncmeet** successfully addresses the growing need for an intelligent, all-in-one online conferencing platform that integrates AI-powered transcription, summarization, and collaboration tools. By combining **React (TypeScript)** on the frontend, **Node.js** on the backend, and **OpenAI** for real-time speech-to-text conversion and meeting summaries, the system ensures that important meeting details are accurately captured and instantly accessible.

Unlike many existing conferencing tools, Syncmeet eliminates the fragmentation of using separate applications for scheduling, collaboration, and post-meeting documentation. Features such as interactive whiteboards, online polls, automated email notifications, and meeting recording create a comprehensive solution for professional and academic environments.

Testing confirmed that the platform meets functional, performance, and security requirements while providing a user-friendly interface. User feedback indicated high satisfaction with the accuracy of AI-generated summaries, ease of scheduling, and collaborative features.

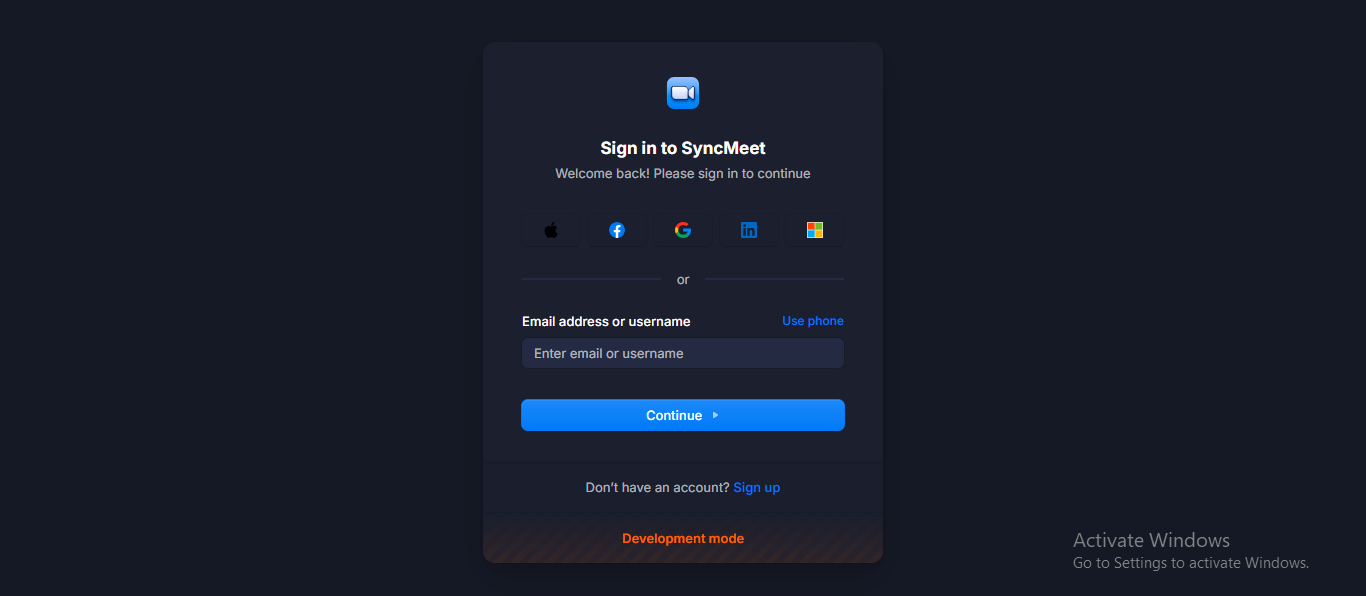
In conclusion, Syncmeet demonstrates that the integration of artificial intelligence with modern conferencing technology can significantly improve productivity, decision-making, and engagement in remote meetings. Its modular architecture also ensures scalability, making it adaptable for future enhancements such as multilingual transcription, advanced analytics, and integration with external productivity tools.

.

## APPENDIX-A

## ****Figure 6.1: Login of Syncmeet****

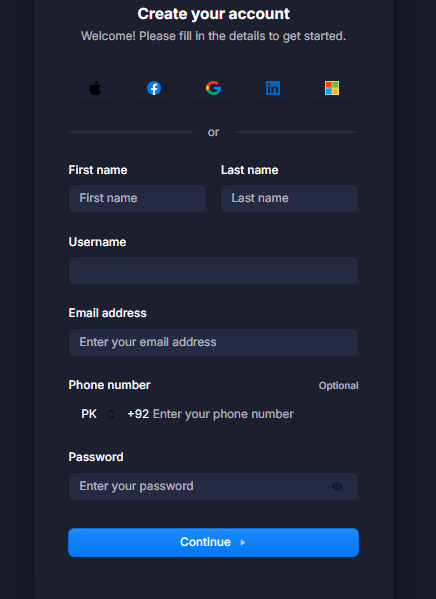
This page allows registered users to access their accounts by entering their **email** and **password**. It also provides links to reset forgotten passwords or navigate to the signup page for new users. Security features such as password encryption and authentication checks ensure user data protection.



**Figure 6.1: Login of Syncmeet**

## ****Figure 6.2: Signup of Syncmeet****

The signup page enables new users to create an account by providing details like **name**, **email**, and **password**. It may include email verification to ensure account authenticity before granting access to the platform.

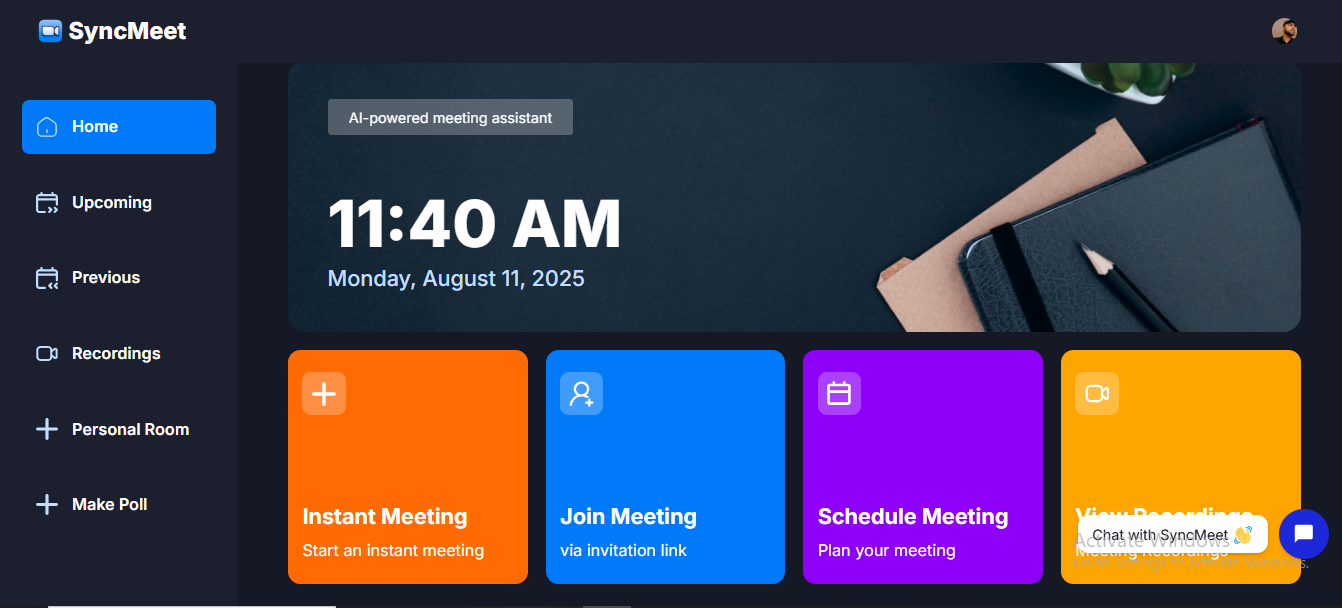


**Figure 6.2: Signup of Syncmeet**

## ****Figure 6.3: Homepage of Syncmeet****

The homepage serves as the **central hub** for all user actions. From here, users can access:

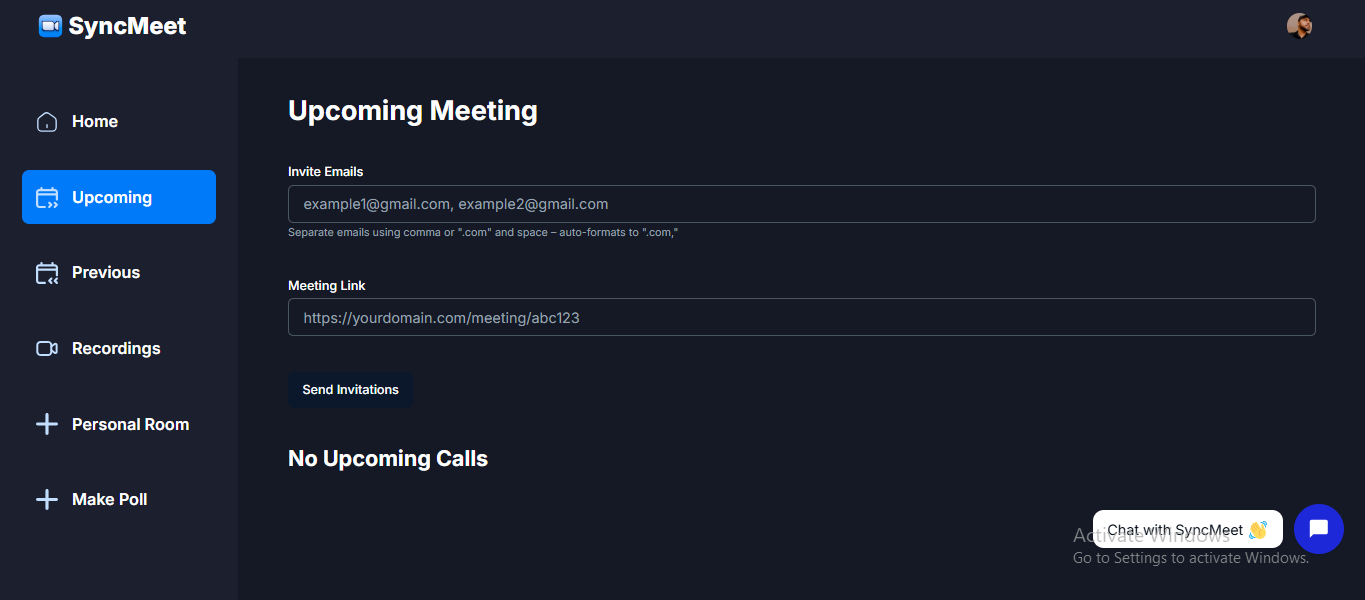
* **Upcoming Meetings**
* **Previous Meetings**
* **Recordings**
* **Schedule a Meeting**
* **Polls**
* **Personal Meeting Links**  
  The layout is designed for quick navigation and an intuitive user experience.



**Figure 6.3: Homepage of Syncmeet**

## ****Figure 6.4: Upcoming Meetings of Syncmeet****

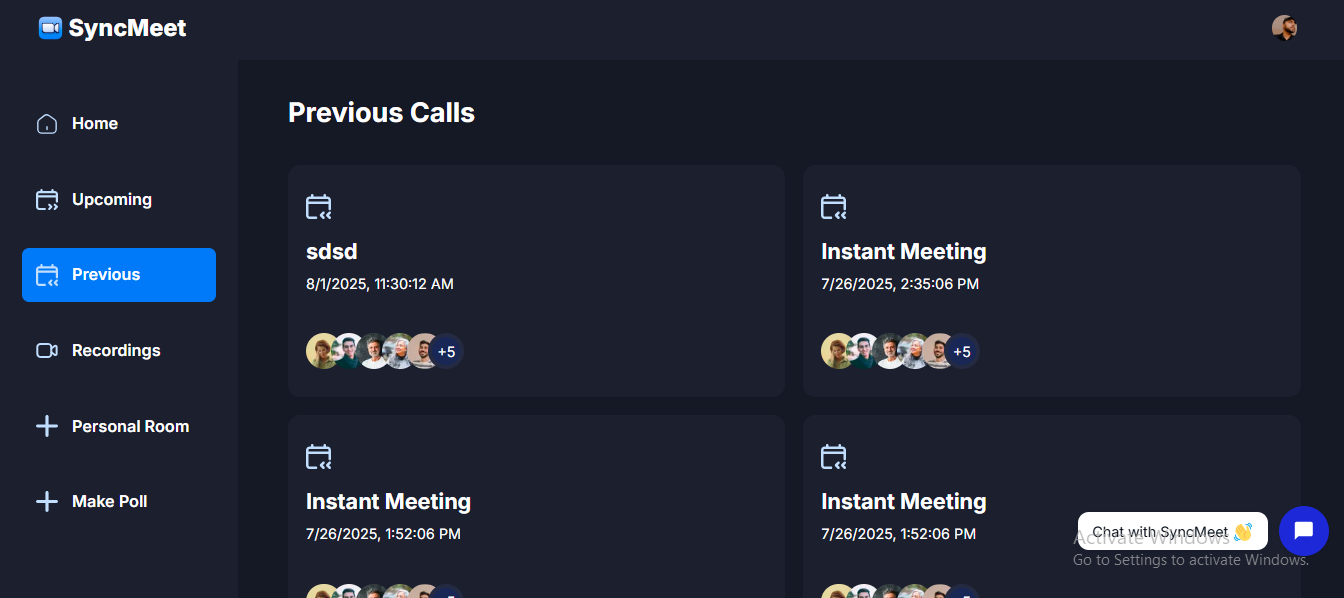
Displays a list of all **scheduled future meetings** along with their **date, time, meeting ID, and host details**. Users can join directly from this list or edit/cancel their scheduled meetings.



### ****Figure 6.4:**** Upcomming meetings of Syncmeet

## ****Figure 6.5: Previous Meetings of Syncmeet****

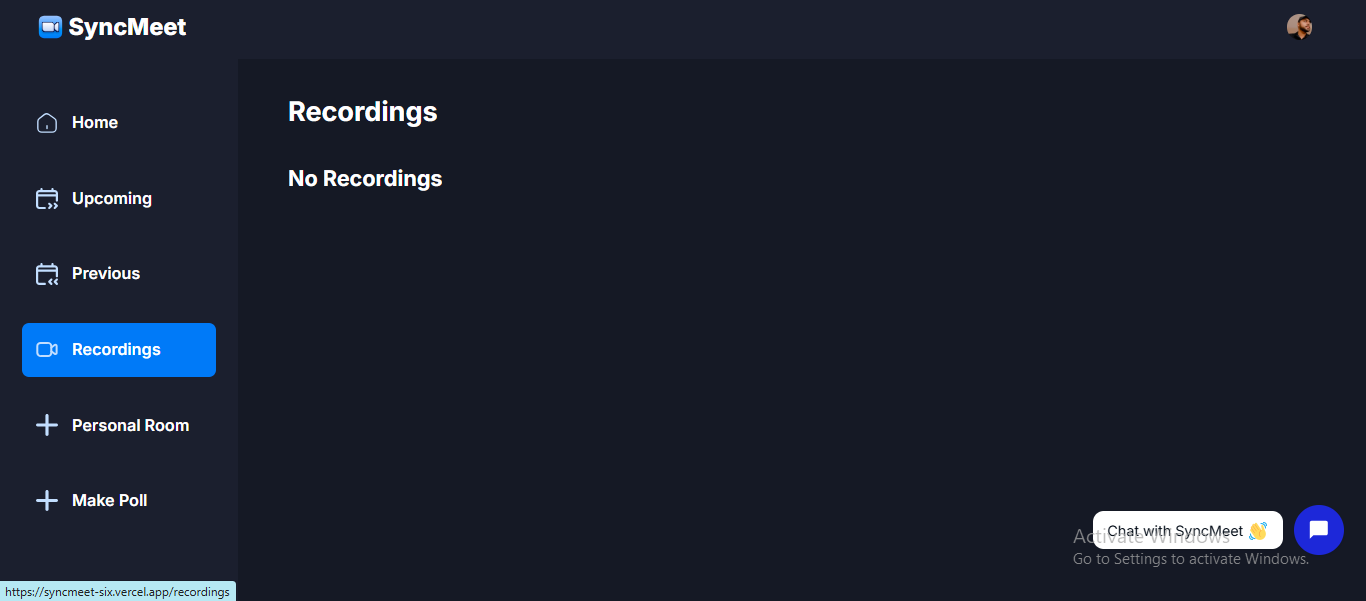
Shows details of meetings that have already taken place. This section helps users review their meeting history, including **date, time, duration, participants**, and any attached notes or links.



**Figure 6.5: Previous meetings of Syncmeet**

## ****Figure 6.6: Recordings of Syncmeet****

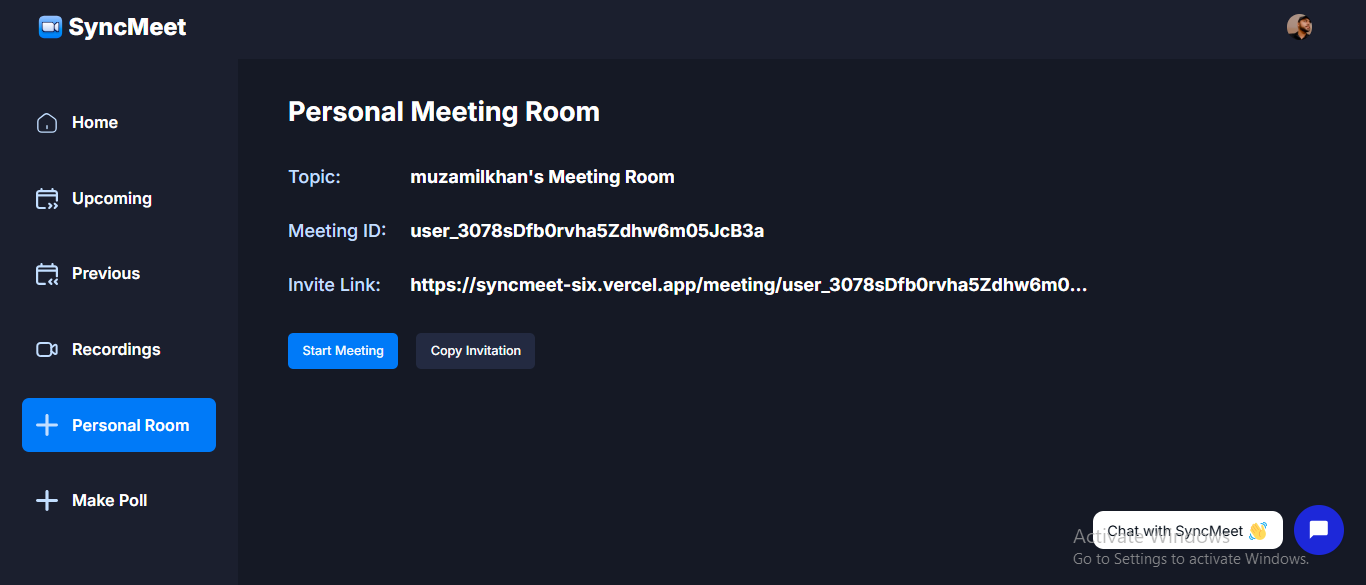
Provides access to stored meeting recordings. Users can **play, download, or share** the recorded video/audio files for future reference, training, or documentation purposes.



### ****Figure 6.6:**** Recordings of Syncmeet

## ****Figure 6.7: Personal Rooms of Syncmeet****

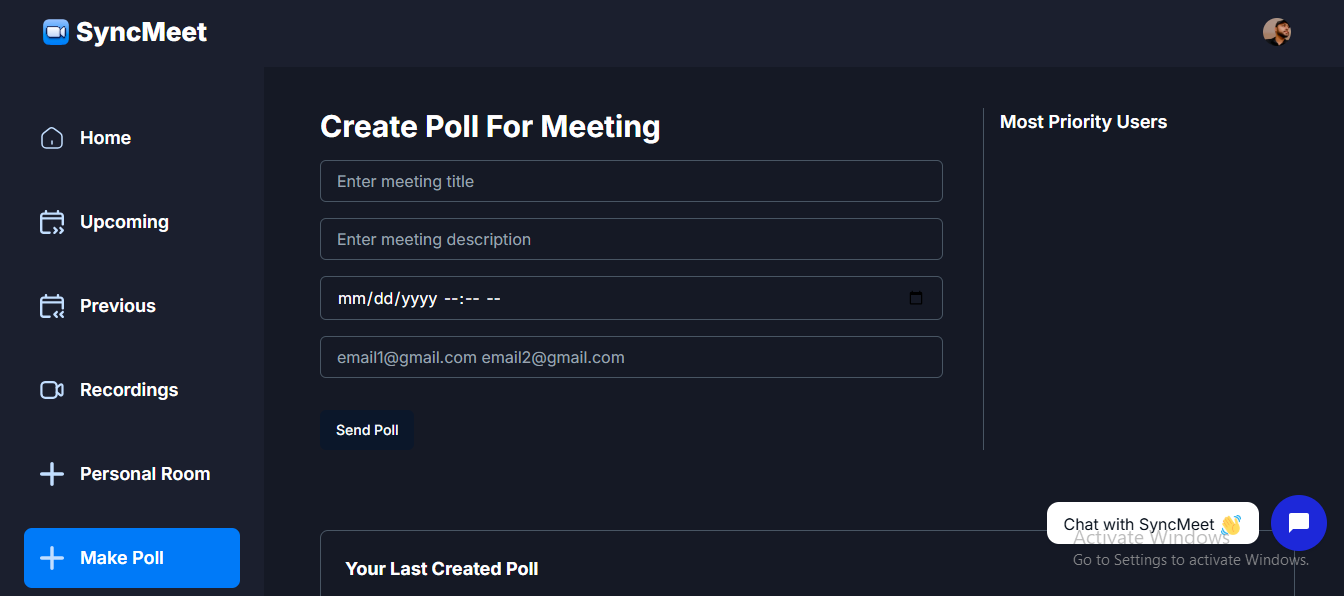
A **personal meeting link** unique to each user. It allows quick meeting initiation without generating a new link each time — ideal for recurring or impromptu meetings with known participants.



### ****Figure 6.7:**** Personal Rooms of Syncmeet

## ****Figure 6.8: Make Poll****

This feature enables hosts to create **interactive polls** during or before meetings. Polls can be used for surveys, decision-making, or gathering participant opinions, with results displayed in real-time.



### ****Figure 6.8:**** Make Poll

# REFERENCES

[1] “Li, X., & Chen, Y. (2021). Real-Time Speech-to-Text Conversion Using Deep Learning Models. IEEE Access.”

[2] “Nguyen, H., & Lee, J. (2020). AI-Powered Meeting Summarization: Approaches and Applications. ACM Computing Surveys.”

[3] “Smith, A., & Brown, L. (2019). Secure Web-Based Video Conferencing: Architecture and Implementation. International Journal of Computer Applications.”

[4] “Patel, R., & Kumar, S. (2022). Designing an Online Collaboration Platform with Whiteboard and Polling Features. Journal of Web Engineering.”

[5] “Johnson, M., & Clark, P. (2018). Node.js for Scalable Real-Time Applications. O’Reilly Media.”

[6] “Gao, L., & Zhang, W. (2021). Implementing Authentication and Session Management with Clerk API. International Conference on Web Technologies.”

[7] “Fernandez, J., & Wang, T. (2020). Deploying Modern Web Applications on Vercel: Best Practices. Web Performance Journal.”

[8] “OpenAI. (2023). API Documentation: Speech-to-Text, Text Summarization, and Language Models. OpenAI Technical Reports.”

[9] “Singh, P., & Verma, K. (2021). SMTP-based Email Notification Systems in Event-Driven Applications. International Journal of Computer Networks.”

[10] “Wang, Y., & Kim, H. (2022). User Experience Design in Virtual Meeting Platforms. Human-Computer Interaction Conference Proceedings.”