## Title:SwiftChatPro



### Submitted By

|  |  |
| --- | --- |
| Hasnain Yousaf | 21-Arid-3416 |
| Muhammad Hamza | 21-Arid-3423 |
| Amir Imtiaz | 21-Arid-3409 |

### Submitted to

Mr.Awais

**Gujrat Institute of Management Sciences**

**PMAS-Arid Agriculture University, Rawalpindi**

Write a Project Evolution Criteria

### 1.Case Study

SwiftChat Pro is an all-in-one communication and collaboration platform designed for businesses and organizations to streamline team interactions, project management, and real-time communication. Many existing tools lack integration, forcing teams to switch between multiple applications for messaging, video conferencing, and task management, leading to inefficiencies and communication gaps.

Modern businesses face challenges due to fragmented communication tools, lack of AI-driven automation, and inefficient workflows. Teams often struggle with managing conversations, project tasks, and meetings across separate platforms, leading to time loss and reduced productivity.

SwiftChatZone Pro provides a seamless, integrated experience by offering:

Real-Time Messaging – One-to-one and group chat using Socket.IO for instant communication.

Audio & Video Calls – WebRTC-powered voice and video calls.

Zoom Integartion – Integrate the Zoom for Create meetings for Video Conference .

Task Management – Kanban board, task assignments, and deadline tracking within the chat interface.

AI-Powered Features – Chatbot for automated responses and conversation summarization.

File Sharing – Secure document, image, and multimedia sharing within conversations.

The system is built using the MEAN Stack (MongoDB, Express.js, Angular, Node.js) with WebRTC for calls and Socket.IO for real-time messaging. AI-driven features enhance automation and user engagement, while JWT ensures secure authentication and Zoom Integartion.

By consolidating messaging, calling, and task management into a single application, SwiftChatZone Pro enhances productivity, improves workflow efficiency, and reduces reliance on multiple disconnected tools. Businesses can experience faster communication, enhanced collaboration, and better project tracking, leading to overall operational improvements.

### Core Modules

**1 User Authentication & Profile Management**

* 1. Secure user authentication using JWT.
  2. User profile view Profile of Partcpant and hidden profile privacy

**2 Real-Time Messaging**

1. One-to-one & group chats powered by Socket.IO.
2. Typing indicators, read receipts, and online/offline status ,Deleteforme,Deleteforeveryone,EditMessage.
3. AI-powered chatbot for auto-responses and summarization.

**3 Audio & Video Calling**

1.WebRTC-based real-time audio/video calls.

**4 Task Management System**

1. Kanban board for task tracking (To-Do, In Progress, Completed).

2. Task creation, assignment.

3. Deadline reminders .

**5 Zoom Integration**

1. Create Meetings and send link in chat .

**6 AI-Powered Features**

1. Automated response.

2. Auto-Summarization of long chats.

**7 File Sharing & Document Management**

1. Upload & share images, PDFs, and documents.

**8 Notifications & Alerts**

1. Notifications for new messages, and tasks.

2. Email & in-app alerts for pending actions.

* 1. Smart reminders for deadlines and unread messages.

### ****3. Non-Functional Modules****

**Scalability**

* The system is designed to handle a growing number of users and messages efficiently.
* Supports cloud-based deployment for better scalability.

**Security & Authentication**

* JWT-based secure authentication.
* End-to-end encryption for messages and file sharing.

**Performance & Speed**

* Optimized database queries for faster response times.
* Real-time updates with minimal latency using Socket.IO and WebRTC.

**Reliability & Availability**

* Redundant database and server backups to prevent data loss.
* 99.9% uptime guarantee with load balancing mechanisms.

**User Experience & Accessibility**

* Responsive UI for seamless experience across devices (desktop & mobile).
* Intuitive navigation with a clean, user-friendly interface.

**Logging & Monitoring**

* Server-side logging for tracking system performance and debugging.
* AI-driven monitoring to detect anomalies and potential system failures.

**Integration Support**

* APIs for third-party integrations (e.g., Zoom,).
* Webhooks for external service connectivity.

### ****4. System Interaction****

**User Authentication & Authorization**

* The system verifies login credentials using JWT authentication.
* Users’ session data is stored securely in the backend.
* Role-based access control is implemented to restrict unauthorized actions.

**Database Interaction**

* MongoDB stores user profiles, messages, tasks, and call logs.
* CRUD operations allow users to send, receive, edit, and delete messages.
* Task assignments, deadlines, and project details are managed in the database.

**Real-Time Messaging & Communication**

* Messages are exchanged using Socket.IO for real-time updates.
* WebRTC handles peer-to-peer video/audio calling.
* AI chatbots process user queries and generate automated responses.

**File Upload & Sharing**

* Uploaded files (images, PDFs, documents) are stored securely.
* The system ensures efficient retrieval and sharing of files within chat conversations.

**AI-Powered Features**

* AI modules analyze chat history for summarization and smart suggestions.
* Automated responses are generated based on predefined patterns and ML models.

**Third-Party API Integration**

* Zoom API is used for video conferencing and meeting scheduling.
* Email notifications are sent via an external SMTP service (e.g., Nodemailer).

**Notification & Alert Mechanism**

* notifications for users about unread messages and task updates.
* Email notifications are sent f and pending tasks.

### ****5. User Interaction****

**User Registration & Login**

* Users sign up and log in securely with email and password.
* Upon login, users can set up their profile and privacy settings.

**Messaging & Communication**

* Users can send messages in one-to-one or group chats.
* Features like read receipts, typing indicators, and online/offline status enhance the chat experience.
* Messages can be edited, deleted (for self or everyone), and pinned.

**Audio & Video Calling**

* Users can initiate calls directly from chat.

**Task Management**

* Users can create, assign, and track tasks on a Kanban board.
* Task deadlines can be adjusted.

**File Sharing**

* Users can upload, preview, and share files within chat conversations.
* Files can be categorized for easy retrieval.

**AI Assistance**

* Users can interact with AI chatbots for quick replies and information retrieval.
* AI-generated chat summaries help users review long conversations quickly.

**Notification System**

* Users receive in-app, email, and notifications for new messages, mentions, and task updates.
* Smart reminders ensure users don’t miss deadlines .

### ****Predictable Patterns****

**1. User Authentication & Session Management**

**Pattern:** Users log in once and stay authenticated until they log out or their session expires.  
**System Response:** JWT tokens manage authentication, and expired sessions require re-login.

**2. Real-Time Messaging Flow**

**Pattern:** A message is sent → Delivered → Read → Acknowledged (with read receipts).  
**System Response:**

* Messages update in real-time using **Socket.IO**.
* "Typing..." indicators appear when a user is typing.
* Read receipts confirm when a message is viewed.

**3. Audio & Video Call Flow**

**Pattern:** A user initiates a call → The recipient receives a notification → Call is answered or declined.  
**System Response:**

* WebRTC establishes a connection.
* If the call is not answered within a set time, it is marked as **missed**.

**4. Task Management Workflow**

**Pattern:** A task is created → Assigned to a user → Updated (in progress, completed) → Marked as done.  
**System Response:**

* Notifications are sent when a task is assigned or updated.

**5. AI-Powered Chatbot Interaction**

**Pattern:** A user asks a question → AI processes the request → The chatbot responds.  
**System Response:**

* If the chatbot doesn’t understand, it suggests related responses.
* AI auto-summarization provides key points of a long conversation.

**6. File Upload & Sharing**

**Pattern:** A user uploads a file → File is processed and stored → The file link is shared in chat.  
**System Response:**

* Large files are compressed before uploading.
* Users receive confirmation once a file is successfully uploaded.

**7. Notification & Alert Mechanism**

**Pattern:** A message or task update occurs → The user receives an alert (in-app, email).

### ****Success Criteria****

**Functional Success**

* Secure **user authentication** (JWT) & profile management.
* **Real-time messaging** with Socket.IO (read receipts, typing indicators, AI chatbot).
* **Audio/video calls** via WebRTC & **Zoom integration** for meetings.
* **Task management** with Kanban board, assignments, and deadline tracking.
* **File sharing** (images, PDFs, docs) & secure storage.
* **AI-powered features** (chatbot, auto-summarization).
* **Notifications & alerts** for messages, tasks, and deadlines.

**Performance Success**

* **Fast response time** (low-latency messaging & calls).
* **Scalable system** handling multiple users.
* **Efficient database & WebSocket management**.

**User Experience Success**

* **Intuitive UI** for easy navigation.
* **Cross-platform compatibility** (desktop & mobile).
* **Smooth onboarding** and minimal bugs.

**Security & Data Protection**

* **Encrypted communication & secure file sharing**.

**Business & Market Success**

* **Seamless Zoom integration** for enhanced collaboration.

### ****Evaluation Criteria for SwiftChatPro****

**Functionality** – Features like authentication, messaging, calling, task management, AI chatbot, and file sharing work as expected.

**Performance** – Fast response time, real-time updates, and efficient database handling with minimal latency.

**User Experience** – Intuitive UI, easy navigation, smooth onboarding, and cross-platform accessibility.

**Security** – secure file sharing.

**Scalability & Reliability** – Handles multiple concurrent users .

**Business Impact** – , seamless third-party integrations

### ****Progress & Development Criteria for SwiftChatPro****

**Phase 1: Planning & Design**

* Define **system architecture** (MEAN stack, WebRTC, Socket.IO).
* Finalize **UI/UX design** and feature requirements.

**Phase 2: Core Development**

* Implement **user authentication & profile management**.
* Develop **real-time messaging** (Socket.IO, AI chatbot).
* Integrate **audio/video calls** (WebRTC, Zoom API).
* Build **task management system** (Kanban board, deadlines).
* Enable **file sharing & document management**.

**Phase 3: AI & Enhancements**

* Add **auto-summarization** & **AI chatbot** for automation.
* Implement **notifications & alerts** (in-app, email).

**Phase 4: Testing & Optimization**

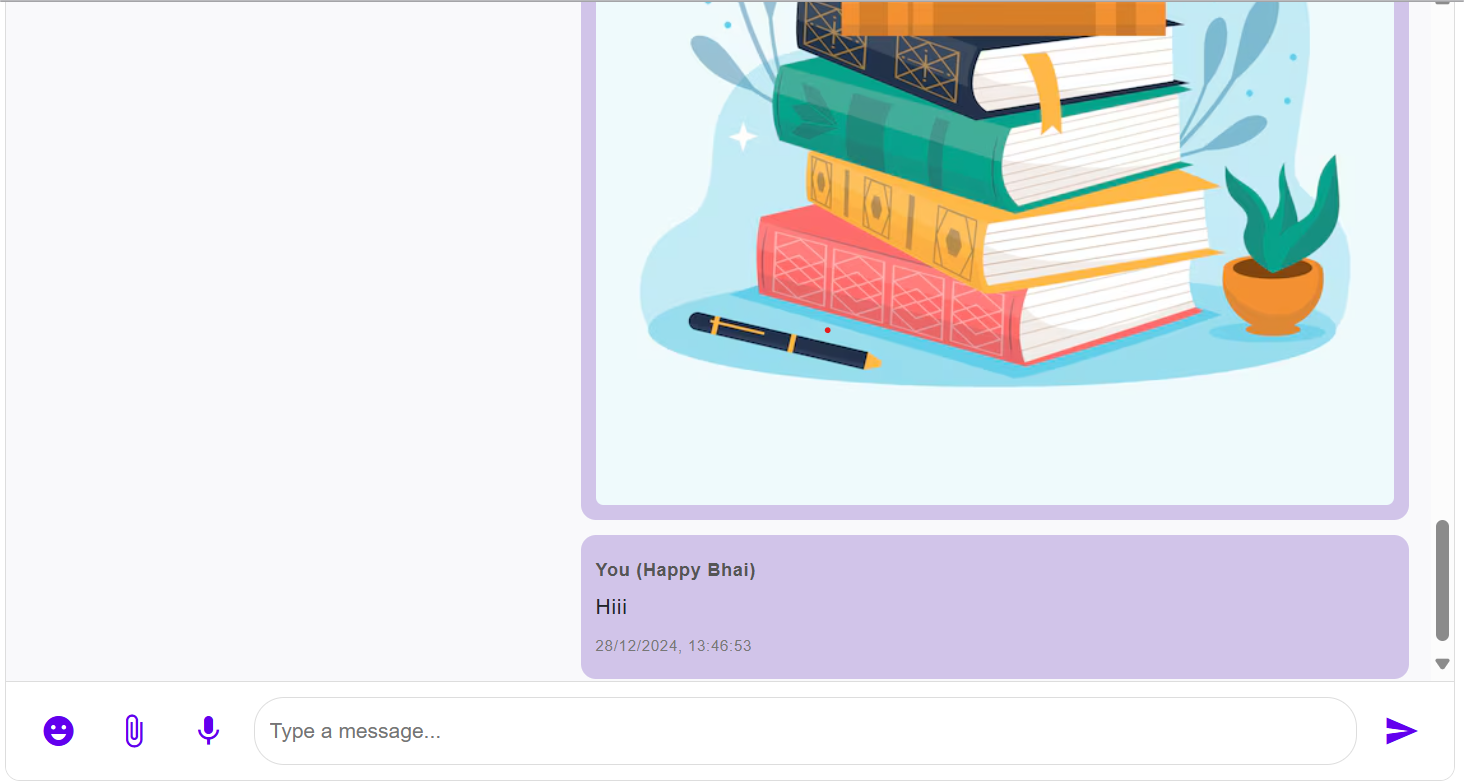
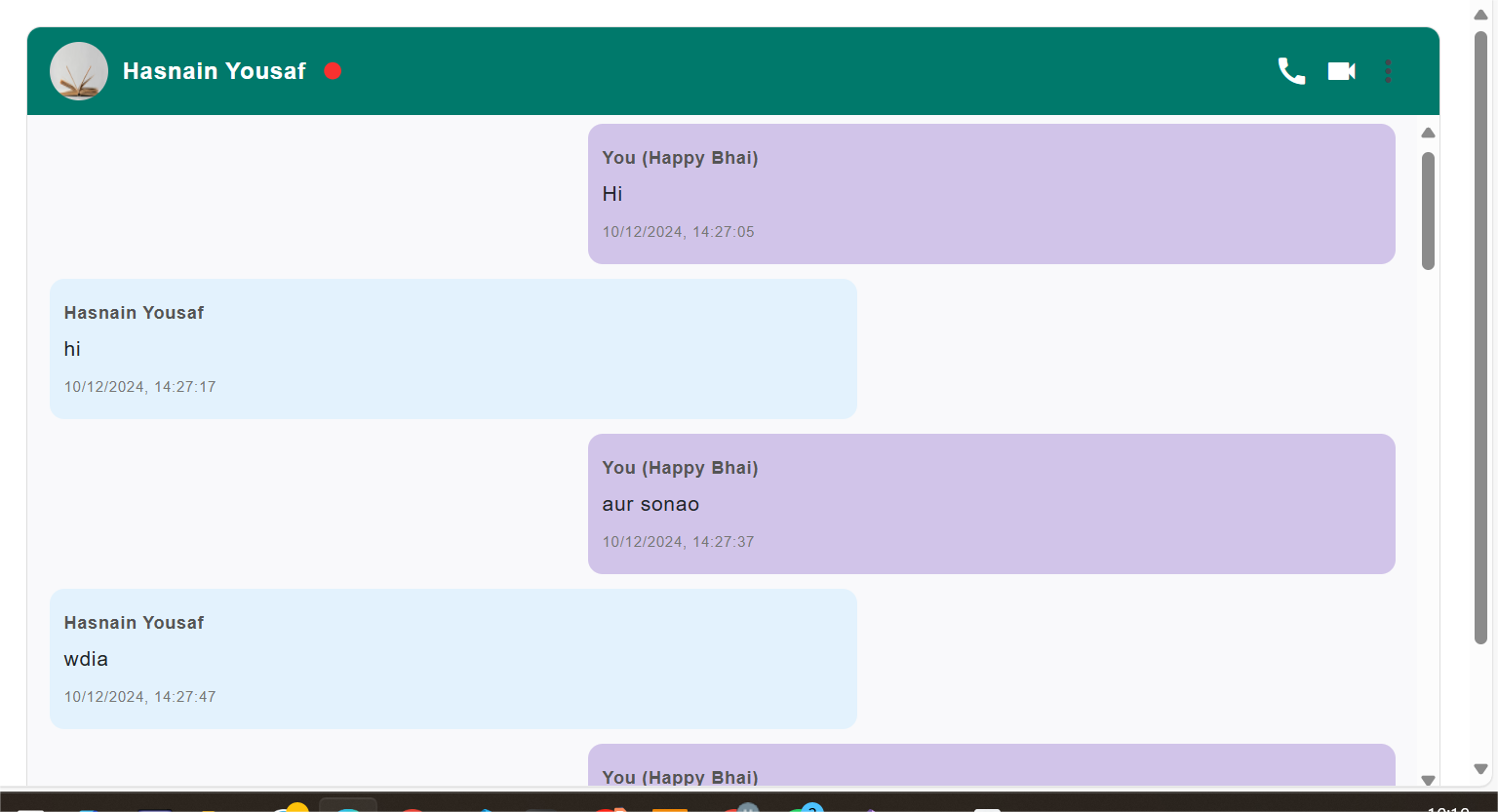
* Conduct **unit, integration, and performance tests**.
* Fix bugs and optimize **speed, scalability, and security**.

✅ **Phase 5: Deployment & User Feedback**

* Deploy on **cloud servers** (MongoDB Atlas, AWS, or Firebase).
* Collect **user feedback**, improve features, and **ensure smooth operation**.

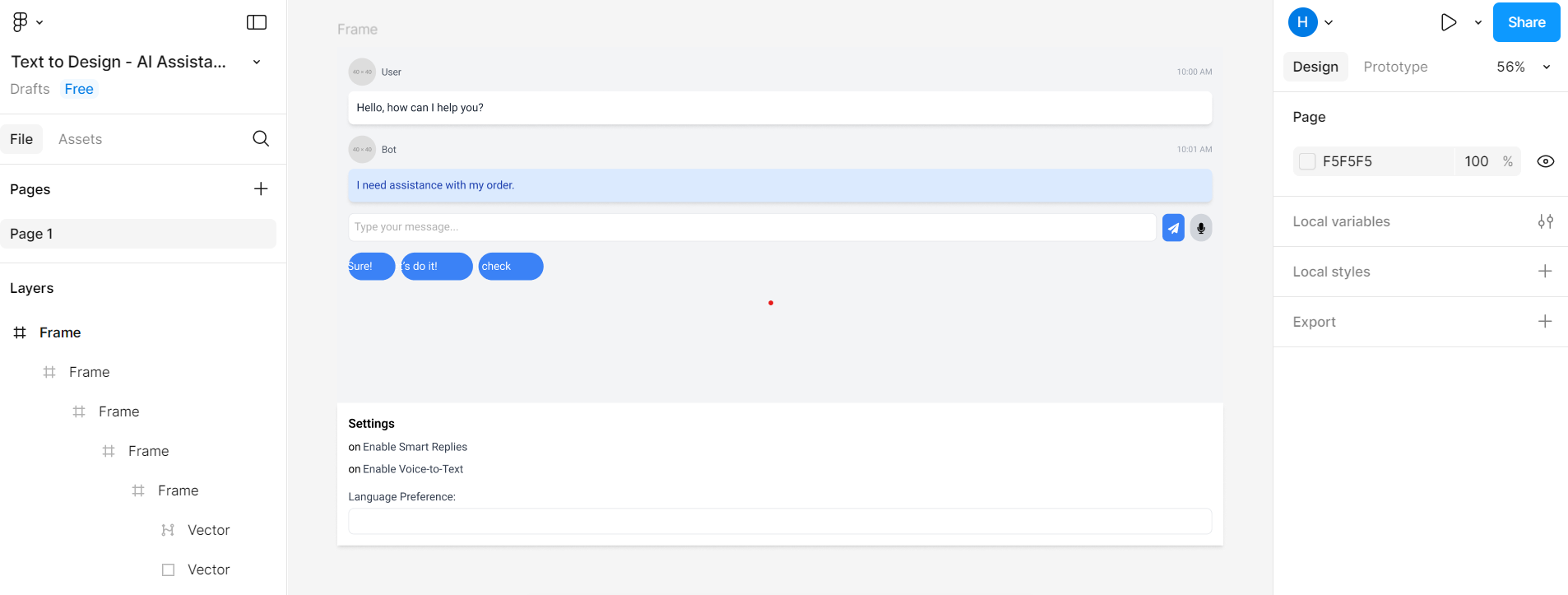
### ****Blueprint****

**Basic Version (Initial Development)**

****

**Advanced Version (Future Enhancements)**

**Voice-to-Text & Smart Replies** – AI-generated smart responses and voice message transcription.

****