

**Android Application Development**

**CHATCONNECT**

**A Realtime Chat and Communication**

**TEAM ID-NM2024TMID04923**

**SEMESTER – V**

**B.TECH -INFORMATION TECHNOLOGY**

**ACADEMIC YEAR-2024-2025**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**V.S.B. COLLEGE OF ENGINEERING TECHNICAL CAMPUS**

**COIMBATORE-6420109**

## **TABLE OF CONTENTS**

CHAPTER NO	TITLES	PAGE NO
I	<u>DESCRIPTION</u>	1
II	REQUIREMENTS	2
III	APP FUNCTIONALITIES	3
IV	PROGRAMS	5
V	OUTPUTS	7

# **CHATCONNECT- A REAL-TIME CHAT AND COMMUNICATION APP**

## **DESCRIPTION:**

CHATCONNECT is an all-in-one real-time chat and communication app designed to streamline how individuals, teams, and communities interact in today's fast-paced digital world. Offering seamless messaging, voice, and video calling, CHATCONNECT ensures smooth communication across all devices, including iOS, Android, Windows, macOS, and web platforms. Users can easily send instant text messages, share multimedia (images, videos, and files), and collaborate on projects with features like shared calendars, real-time document editing, and task management tools. The app supports secure, encrypted communication with end-to-end encryption for both messages and calls, ensuring user privacy is always protected. For added convenience, users can enjoy customizable notifications, send voice notes with automatic transcriptions, and sync messages across devices. With a focus on user-friendly design, CHATCONNECT offers both dark and light modes and intuitive navigation, making it accessible to individuals and teams of all sizes. Whether you're staying connected with friends and family, collaborating with coworkers, or managing a community, CHATCONNECT brings everything you need for personal, professional, and collaborative communication into one secure and efficient platform. The app follows a freemium model, offering essential features for free, with premium options available for businesses and power users seeking additional storage, advanced security, and enhanced collaboration tools.

## **REQUIREMENTS:**

### **1)Hardware Requirements:**

**Servers:** Scalable cloud-based servers (e.g., AWS, Google Cloud, or Microsoft Azure) to handle message processing, real-time communication, file storage, and user data.

**Storage:** High-capacity, redundant storage solutions (e.g., AWS S3, Google Cloud Storage) to store user data, media files, and backups securely.

**Networking:** Low-latency, high-bandwidth networking infrastructure to ensure fast real-time messaging and call quality, especially for voice and video communications.

**Backup & Disaster Recovery:** Hardware systems for regular data backups and quick recovery in case of service interruptions or data loss.

## **2)Software Requirements:**

**Mobile Platforms:** iOS (version 12.0 or higher) and Android (version 6.0 or higher) with SDKs like Swift (for iOS) and Kotlin/Java (for Android).

**Web Platform:** A responsive web app built with HTML5, CSS3, JavaScript (React.js, Vue.js), and **WebSocket for real-time communication.**

**Backend:** Cloud-based server architecture using technologies such as **Node.js, Python (Django/Flask), or Java with microservices architecture for scalability.**

**Database:** **NoSQL (MongoDB) or relational (PostgreSQL/MySQL) databases for efficient data storage and retrieval.**

**Security:** End-to-end encryption libraries (e.g., OpenSSL, WebRTC) for secure messaging, along with authentication systems (OAuth 2.0, JWT) and 2FA (TwoFactor Authentication)

## **APP FUNCTIONALITIES:**

CHATCONNECT provides a range of functionalities designed to facilitate seamless real-time communication and collaboration across personal, professional, and community settings. Key functionalities include:

### **1.Real-Time Messaging and Media Sharing :**

Instant Text Messaging : Send one-on-one or group messages with real-time delivery, read receipts, and typing indicators.

Multimedia Sharing : Share images, videos, voice notes, and documents directly within chats. Inline previews for easy viewing of media files.

Message Search & History : Quickly search through chat history to find important messages, media, or files.

## **2. Voice and Video Communication :**

Voice & Video Calls : Make high-quality voice and video calls, including one-on-one and group calls, with minimal latency and excellent audio/video quality.

Screen Sharing : Share your screen during video calls for presentations, demos, or collaborative work.

Call Recording : Option to record voice and video calls for business or personal use (with user consent).

## **3. Collaboration and Productivity Tools :**

Task Management : Create, assign, and track tasks within team chats. Integrate to-do lists and reminders for better task management.

Shared Calendars & Events : Schedule and manage events with group calendars, syncing across all participants to ensure everyone stays informed.

Real-Time Document Collaboration : Collaborate on documents, spreadsheets, and presentations in real-time with integrated tools like Google Docs, Microsoft Office 365, or built-in editing tools.

## **4. Security and Privacy Features :**

End-to-End Encryption : Ensure secure communication with end-to-end encryption for messages, voice calls, and video calls.

Two-Factor Authentication : Strengthen account security with 2FA, protecting users from unauthorized access.

Self-Destructing Messages : Send messages that disappear after being read, enhancing privacy for sensitive communications.

## **PROGRAM:**

```
package com.example.kekala
```

```
import android.annotation.SuppressLint import
import android.os.Bundle import
import androidx.activity.ComponentActivity import
import androidx.activity.compose.setContent import
import androidx.compose.foundation.layout.* import
import androidx.compose.material3.* import
import androidx.compose.runtime.Composable import
import androidx.compose.ui.Alignment import
import androidx.compose.ui.Modifier import
import androidx.compose.ui.res.painterResource import
import androidx.compose.ui.unit.dp import
import androidx.navigation.NavHostController import
import androidx.navigation.compose.NavHost import
import androidx.navigation.compose.composable import
import androidx.compose.material3.Text import
import androidx.compose.foundation.layout.Column import
import androidx.compose.foundation.lazy.items
import androidx.compose.foundation.lazy.LazyColumn
```

```
import androidx.navigation.compose.rememberNavController import
import com.example.kekala.ui.theme.KekalaTheme
```

```
class MainActivity : ComponentActivity() {    override
fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)    setContent {
KekalaTheme {
        AppNavigation()
    }
}
}
```

```
}
```

```
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
```

```
@Composable
```

```
fun AppNavigation() {    val navController: NavHostController =  
rememberNavController()
```

```
    Scaffold {
```

```
        NavHost(  
            navController  
        ) {
```

```
            startDestination = "login" // Start screen is Login
```

```
        } {
```

```
            composable("login") { LoginScreen(navController) }
```

```
            composable("signup") { SignUpScreen(navController) }
```

```
        }
```

```
    }
```

```
}
```

## OUTPUT:







