COLLEGE CODE : 9111

COLLEGE NAME: SRM MADURAI COLLEGE FOR ENGINEERING AND TECHNOLOGY

DEPARTMENT: B.E COMPUTER SCIENCE AND ENGINEERING

STUDENT NM-ID: **29693809323E58D85804E77C620A3B33**
E84A706084E253582B34A98D05CA3B22
5783594E2D19B3FA18F47F4C526F12C7
1069BEA60F6B24E85854A583833EB1C6
5EAED41CE1AB1D89AC442AA0A8A642F7

ROLL NO: 911123104021
911123104003
911123104049
911123104035
911123104042

DATE:

Completed the project named as Phase_II_

TECHNOLOGY PROJECT NAME : CHAT APPLICATION UI

SUBMITTED BY,
NAME : C.AKSHAYA
H.JESIMA YOHAANA
S.SHREYA MERCY
D.PRIYA DHARSHINI
M.K.ROSHINI

CHAT APPLICATION UI – PHASE II

1. Tech Stack Selection

Frontend:

- **Framework:** React Native (for cross-platform mobile development)
- **UI Library:** React Native Paper or NativeBase
- **State Management:** Redux or Context API
- **Real-Time Communication:** Socket.IO or Firebase Realtime Database

Backend:

- **Server:** Node.js with Express
- **Real-Time Communication:** Socket.IO or Firebase Cloud Functions
- **Database:** MongoDB or Firebase Firestore
- **Authentication:** Firebase Authentication or JWT with Passport.js

DevOps:

- **Version Control:** Git with GitHub
- **CI/CD:** GitHub Actions or CircleCI
- **Deployment:** Heroku, Firebase Hosting, or AWS Amplify

2. UI Structure / API Schema Design

UI Components:

- **Login/Signup Screens:** Forms for user authentication
- **Chat List Screen:** Displays list of conversations
- **Chat Screen:** Real-time messaging interface
- **Settings Screen:** User profile and app settings

API Endpoints:

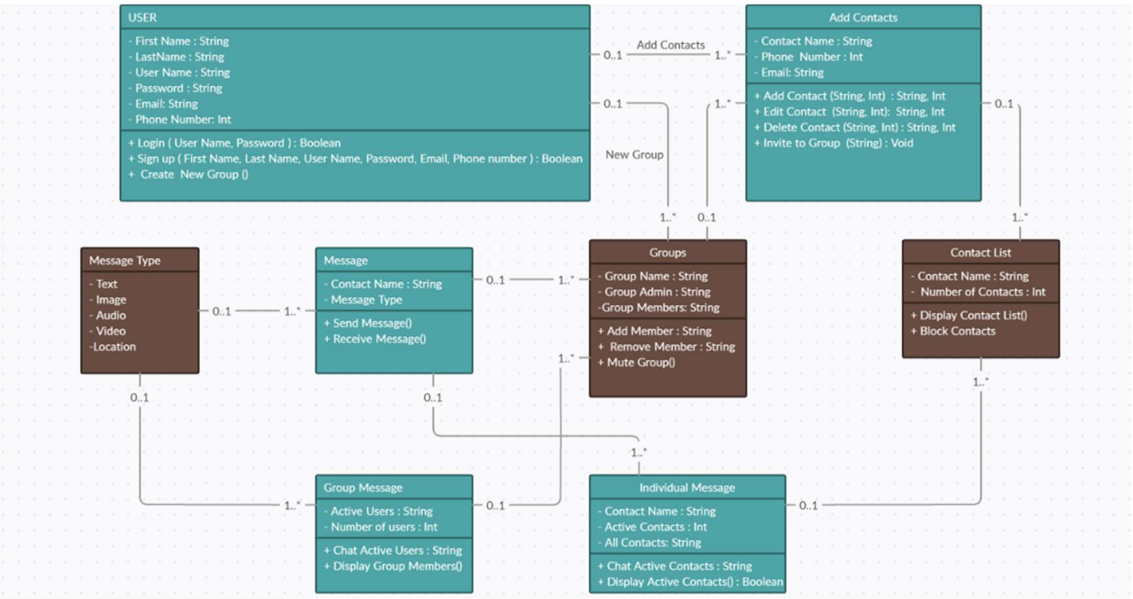
- POST /api/auth/register: Register a new user
- POST /api/auth/login: Authenticate a user

- GET /api/messages: Retrieve messages for a conversation
- POST /api/messages: Send a new message
- PUT /api/users/:id: Update user profile

3. Data Handling Approach

- **Real-Time Messaging:** Utilize WebSockets (Socket.IO) or Firebase Realtime Database for instant message delivery.
- **Message Storage:** Store messages in a NoSQL database (MongoDB or Firebase Firestore) for scalability.
- **User Authentication:** Implement secure authentication using Firebase Authentication or JWT with Passport.js.
- **Data Synchronization:** Ensure data consistency across devices using Firebase's built-in synchronization features.

4. Component / Module Diagram



5. Basic Flow Diagram

1. **User Registration/Login:**
    - User enters credentials.

- System authenticates user.
- User is redirected to the chat list screen.

2. **Chat Interaction:**
   - User selects a conversation.
   - Messages are fetched from the database.
   - User sends a message.
   - Message is stored in the database and broadcasted in real-time to the recipient.
3. **User Settings:**
   - User accesses settings.
   - User updates profile information.
   - Changes are saved to the database.