9530

St. MOTHER THERESA ENGINEERING COLLEGE

COMPUTER SCIENCE ENGINEERING

NM-ID: 80D1A28C2DB0E7EACF842232341EF1B6

REG NO: 953023104026

DATE:22-09-2025

Completed the project named as

Phase 3

FRONT END TECHNOLOGY

CHAT APPLICATION UI

SUBMITTED BY,

M. EBENEZER JENOVA

9791255005

Phase 3 – MVP Implementation

Project Setup:

We will create the ReactJS project using modern tooling such as create-react-app or Vite. We will configure ESLint/Prettier for code quality, install required UI libraries (TailwindCSS, shadcn), and establish a clear folder structure for components, services, and assets. Environment variables and initial configuration files will also be set up to make later integration smoother.

Core Features Implementation:

We will implement the three core UI modules:

- **Sidebar**: lists contacts/conversations, includes search and status indicators.
- **Chat Window**: displays message history, typing indicators, timestamps, and supports scroll behaviour.
- **Message Input**: allows the user to compose and send messages, attach files (optional), and use emojis.

We will use React hooks (useState, useEffect, useContext) to manage state and share data between these components. We will also create reusable components (buttons, cards) for consistency.

Data Storage (Local State / Database):

- In this phase we will store chat messages, user information, and UI states in React’s local state and Context API for fast development.
- We will also define interfaces and mock data to simulate database responses, so that actual backend integration in later phases becomes seamless..

Testing Core Features:

Each module will be tested individually and together (integration testing). We will check for:

- Correct rendering of contact lists and messages
- Smooth sending and receiving of new messages
- UI responsiveness across devices
- Error handling and state reset
- We will also use basic unit tests (Jest/React Testing Library) for critical functions.

Version Control (GitHub):

We will host the project in a GitHub repository, using a main branch for stable code and feature branches for development. We will create a clear commit message convention, add a README with setup instructions, and enable GitHub Issues/Projects to track tasks. This ensures transparency and easy collaboration.