

# Chat Application - Technical Report

## Project Overview

A full-stack real-time chat application built using React + Vite and Node.js + Express. It features real-time messaging, secure user authentication, and a modern UI with dynamic theme customization.

## Technology Stack

### Backend Technologies:

- Node.js with ES6 modules
- Express.js
- MongoDB with Mongoose ODM
- JWT Authentication and bcryptjs for password hashing
- Socket.io for real-time communication
- Cloudinary for image storage
- Middleware: CORS, cookie-parser, body-parser
- Development: Nodemon

### Frontend Technologies:

- React 19 with Vite
- Zustand for global state
- React Router DOM v7
- Tailwind CSS + DaisyUI (32 themes)
- Axios with interceptors
- Socket.io-client
- Lucide React icons
- React Hot Toast for notifications
- Browser Image Compression

## Architecture Overview

### Backend Structure:

## Chat Application - Technical Report

- controllers/: Business logic
- models/: Mongoose schemas
- routes/: API definitions
- middleware/: Auth & validation
- lib/: Utility files and configs
- index.js: Main server file

### Frontend Structure:

- components/: UI components
- pages/: Route pages
- store/: Zustand state store
- lib/: Utility helpers
- skeleton/: Loading placeholders
- constants/: App-wide constants

## Authentication System

- Email verification, password hashing with bcryptjs (10 salt rounds)
- JWT-based auth with 7-day token expiry
- HTTP-only cookies for sessions
- Protected routes via middleware
- Profile picture upload with Cloudinary
- Input validation, CORS, and cookie security measures

## Messaging System

- Real-time messaging via Socket.io
- Supports text and images (uploaded to Cloudinary)
- Online/offline indicators and status updates
- Persistent message history in MongoDB
- Chat management including delete history

## Chat Application - Technical Report

- Message schema includes sender, receiver, text, image, timestamps

### User Management

- User model with email, fullName, hashed password, profilePic, timestamps
- Search and filter contacts
- Online user tracking in real-time
- Exclude current user from contact list

### UI/UX Features

- 32 built-in DaisyUI themes with live preview
- Responsive mobile-first design with Tailwind
- Sidebar, chat container, input field, navbar, skeleton loaders
- Mobile support: collapsible sidebar, touch-optimized UI
- Toast notifications for interactions

### API Endpoints

/api/auth:

- POST /signup
- POST /login
- POST /logout
- PUT /update-profile
- GET /check

/api/messages:

- GET /users
- GET /:id
- POST /send/:id

## Chat Application - Technical Report

- DELETE /:id

### Database Design

Users Collection:

- Stores auth and profile data

Messages Collection:

- Stores message content and metadata
- Indexes for efficient querying by userId and timestamp

### Deployment Features

- Static serving of frontend via Express
- Environment-based configuration
- CORS settings for environments
- Optimized builds: Vite for frontend, production-ready Node backend
- Error handling for robustness

### Key Features Summary

- Real-time messaging
- User authentication & authorization
- Profile picture upload
- Persistent message history
- Online status indicators
- Image sharing
- Chat deletion & search
- 32 theme UI, responsive design, notifications

## Chat Application - Technical Report

### Security Considerations

- bcryptjs password hashing
- JWT token-based auth with secure cookies
- File type and size validation for uploads
- Secure image handling and CORS configuration
- HTTPS in production, secure DB connection

### Performance Optimizations

Frontend:

- Vite optimization
- Image compression
- Lazy loading and optimistic updates
- Zustand for efficient state handling

Backend:

- Query optimization
- Socket management
- Cloudinary optimizations
- Centralized error logging