COLLEGE CODE: 9623

COLLEGE NAME: Amrita College of Engineering and Technology

DEPARTMENT: Artificial Intelligence and Data Science

STUDENT NM-ID: autad0008

ROLL NO: 962323243008

DATE : 22-09-2025

Completed the project named as Phase 3

TECHNOLOGY PROJECT NAME : Chat Application UI

SUBMITTED BY,

NAME : Anushiya R

MOBILE NO: 9342305214

Chat Application UI

MVP Implementation

A step-by-step guide for building your first chat app

Getting Started with MVP

What is MVP?

Minimum Viable Product - core features only

Fast deployment, user feedback, iterative improvement

Why Chat Apps?

Real-time communication essential

Scalable foundation for future features

Chat Application UI

Technology Stack & Setup

Frontend Options

React, Flutter, or Swift

Choose based on your target platform

Backend Services

Node.js/Express or Firebase

Firestore , MongoDB, or SQLite

Development Environment

VS Code or Android Studio

GitHub for version control

Initialize Project

create -react -app or framework command

Setup Folder Structure

/components, /assets, /services, /   api

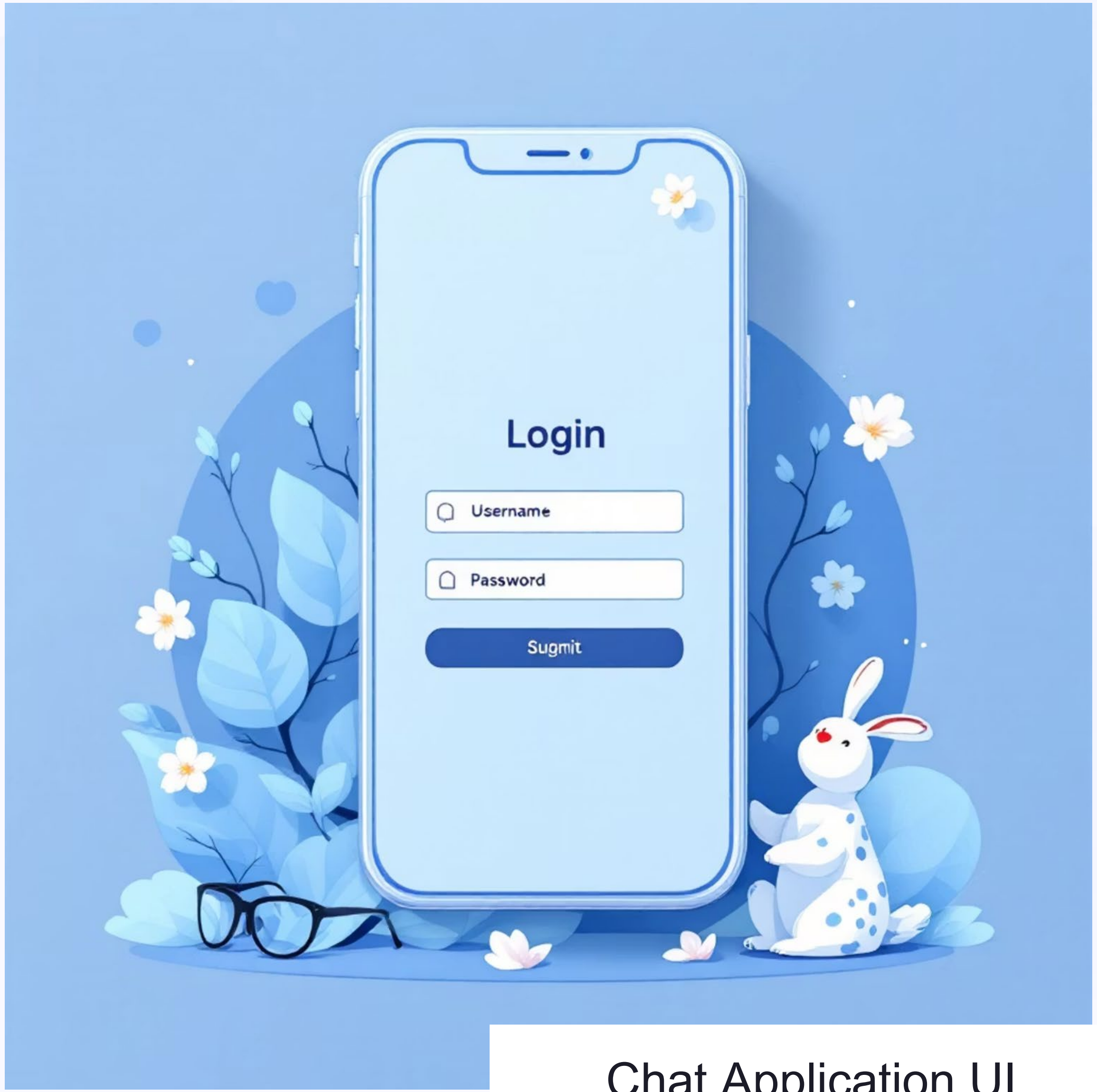
Core Features: Authentication & UI

User Authentication

- Email, social media, or anonymous login
- Firebase Auth or JWT tokens
- Secure session management

UI Design Principles

- Clean, intuitive interface
- Message bubbles & input field
- Mobile -first responsive design



Chat Application UI

Real-Time Messaging & Storage

Real-Time Delivery

WebSockets or Firebase Realtime Database

Instant message delivery

Read receipts (optional)

Message Persistence

Local storage for offline access

Firebase or MongoDB backend

Message history sync

Notifications & Group Features

Wen you don"d mest cane you cand youc un'lt eesly and for yut in fee you thing you want is out that notificaion your ple for worle.

Push Notifications

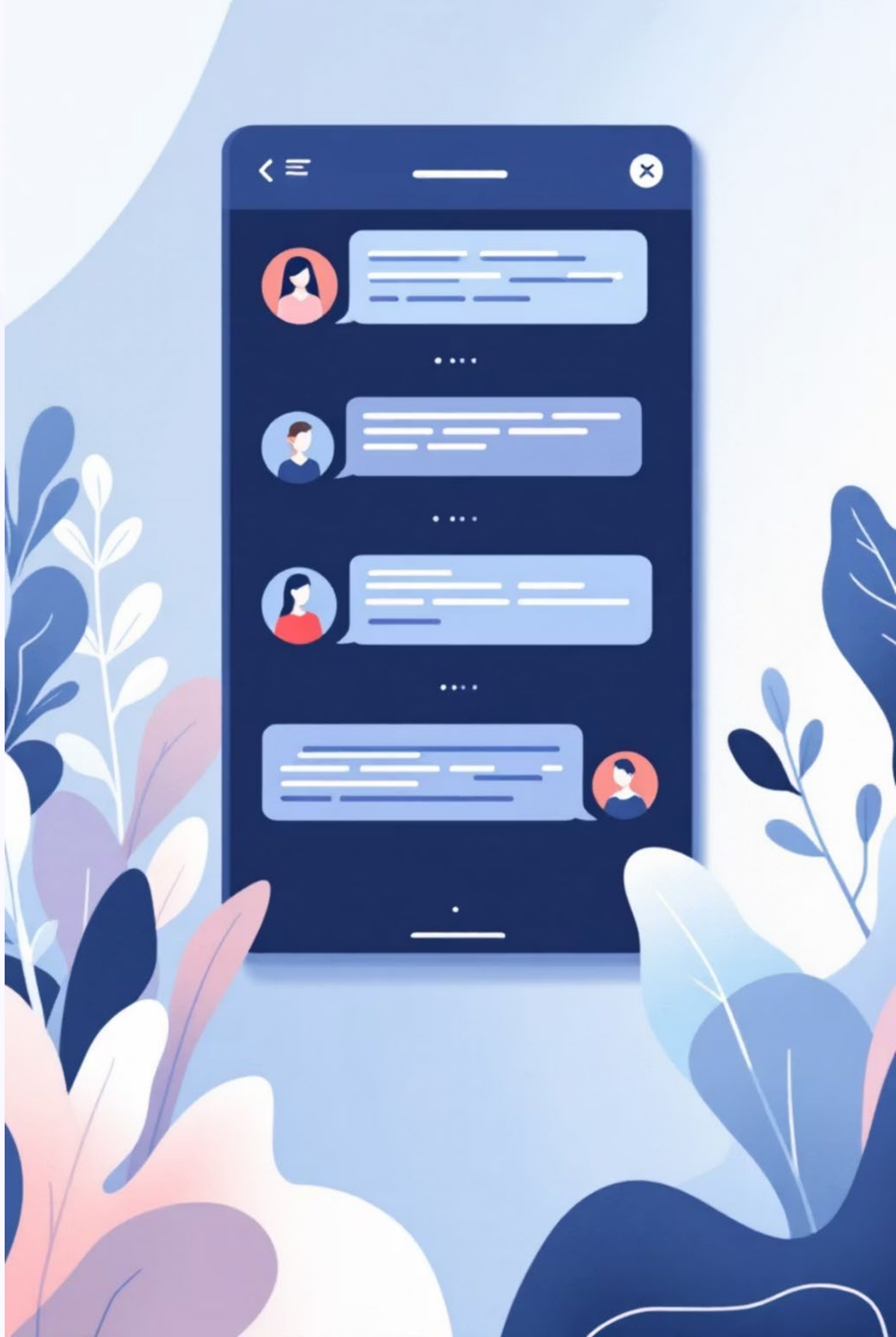
Background message alerts

Firebase Cloud Messaging (FCM)

Chat Rooms & Groups

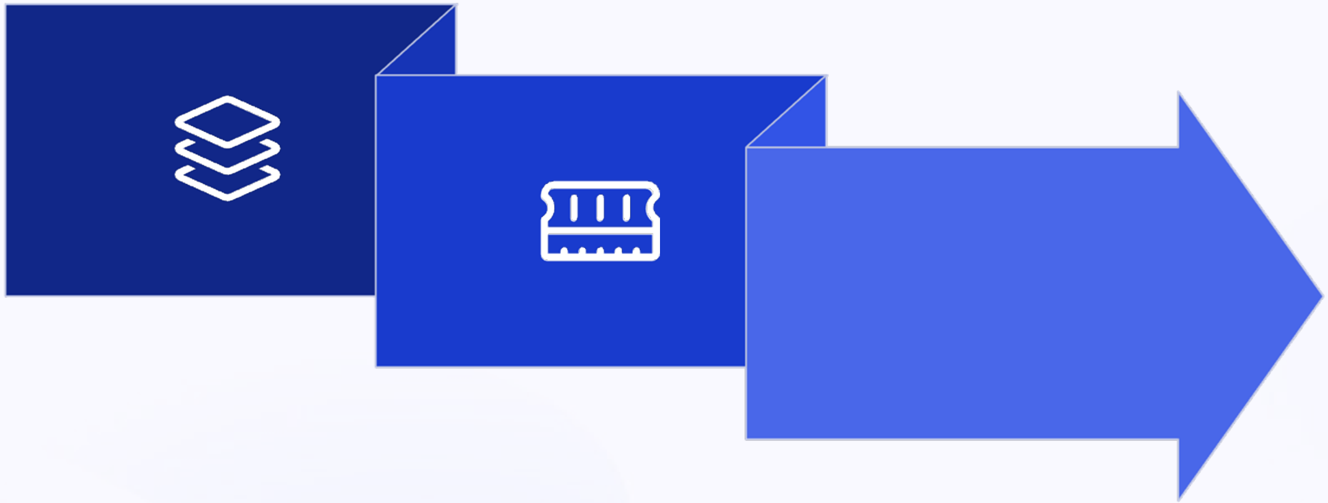
Create and manage group chats

Member roles and permissions



Data Management

Strategy



Context API / Redux

Firebase / MongoDB

Session Memory

Chat Application UI

Testing Your Chat App

1 Unit Testing

Jest for individual components

Test Message, ChatBubble, Login forms

Core logic: send/receive, authentication

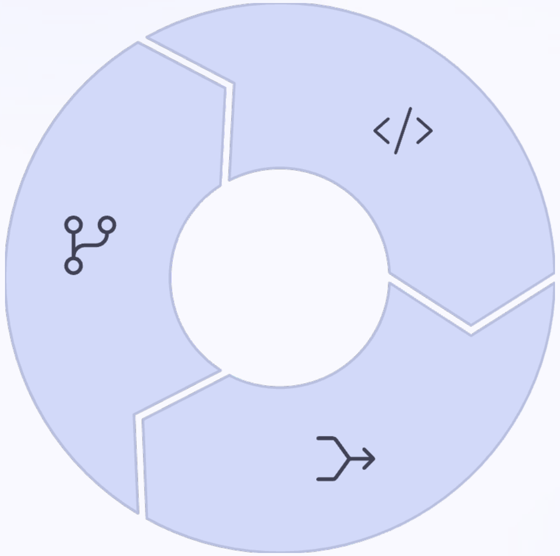
2 End-to-End Testing

Cypress for complete user flows

Login → send messages → notifications

Multi-device simulation

Version Control with GitHub



Master/Main Branch

Stable, production - ready code

Feature Branches

feature/chat - ui, feature/auth

Pull Requests

Code reviews and merging

Pro Tip: Commit often with meaningful messages like "Implemented message sending functionality"

You're Ready to Build!

5

Core Features

Authentication, messaging, storage, notifications, groups

3

Testing Phases

Unit tests, integration tests, end-to-end validation

1

MVP Goal

Functional chat app ready for user feedback

Start building, test thoroughly, iterate based on feedback