KLE Society's
BACHELOR OF COMPUTER APPLICATIONS
P C Jabin Science College, Autonomous, Hubballi

SYNOPSIS
on
FULL STACK DEVELOPMENT PROJECT 2025-26

Semester : V

Title of the Project:

ClarifyAI – AI-Powered Campus Assistant Chatbot

Introduction:

Students waste hours navigating bureaucracy. Simple questions—library hours, registration deadlines, cafeteria locations—shouldn't require hunting down administrators or sifting through outdated PDFs. Campus information gets buried in emails, lost in announcements, forgotten in orientation packets.
We built ClarifyAI to fix this.

Key Features:

I. AI-Powered Responses
Groq AI with Llama 3.1-8B delivers intelligent, campus-specific answers through real-time streaming responses. No generic information—it knows your campus inside out.

II. Secure Authentication
Google OAuth and email/password login with role-based access control using Supabase Auth and JWT tokens.

III. Admin Dashboard
Clean interface for managing FAQs and announcements. Changes go live immediately—no deployment delays.

IV. Chat History
Every conversation gets saved. Students can reference previous interactions, and you get insights into common queries.

V. Responsive Design
Built with Tailwind CSS and shadcn/ui. Works flawlessly across phones, tablets, and desktops.

Modules:

1. User Authentication Module

Handles registration, login, and session management through Supabase Auth. Supports Google OAuth and email/password authentication with role-based access control using JWT tokens. Middleware verifies authorization on every request to keep admin functions secure.

2. AI Chat Module

The core engine powering natural conversations. Retrieves relevant FAQs and announcements from the database, feeds them to Groq AI as context, and delivers streaming responses in real-time. Maintains conversation history automatically so the AI remembers previous interactions.

3. Admin Management Module

Provides CRUD operations for FAQs, announcements, and campus content with admin-only access. Content updates reflect immediately in chatbot responses. Includes analytics to track popular queries and usage patterns.

4. Frontend (User Interface) Module

Built with React 18, TypeScript, and Vite for fast development. Features a modern chat interface with streaming messages and typing indicators, plus an admin dashboard with tables and forms. Fully responsive design using Tailwind CSS works across all devices.

5. Backend (API Server) Module
FastAPI handles all server-side logic with async/await for performance. Authentication middleware verifies tokens, Groq AI integration manages inference requests, and optimized database queries stay fast as data grows.

6. Database Module
Supabase provides PostgreSQL's reliability with modern tools. Stores users, FAQs, announcements, and chat history with proper relational structure. Row Level Security policies protect data, and real-time subscriptions update the UI immediately when data changes.

Hardware Requirements:

I. Processor
Intel Core i3 or equivalent. An i5 makes development smoother but isn't strictly necessary.

II. RAM
4GB minimum to run everything. 8GB recommended if you're running other applications simultaneously.

III. Storage
5GB free space for dependencies, build artifacts, and local data.

IV. Monitor
Any modern display works. 15.6" or larger recommended for comfortable development.

V. Internet Connection
Stable connection required for API calls to Groq and Supabase. Mobile hotspot works if bandwidth is decent.

Software Requirements:

Frontend Technologies

i. React.js

UI library for building interactive interfaces. We chose it because the ecosystem is mature and hiring React developers is straightforward.

ii. TypeScript
Adds type safety to JavaScript. Catches bugs at compile time instead of runtime. Makes refactoring less terrifying.

iii. Vite
Build tool that's actually fast. Hot module replacement works reliably. Development server starts in milliseconds, not minutes.

iv. Tailwind CSS
Utility-first CSS framework. No fighting with CSS specificity. No wondering which class names are already taken.

v. shadcn/ui
Pre-built accessible components. Saves weeks of building dropdowns, modals, and form inputs from scratch.

Backend Technologies

i. Python 3.8+
Backend language with excellent AI library support. FastAPI requires modern Python features, so don't try running this on Python 2.

ii. FastAPI
Modern framework for building APIs quickly. Automatic API documentation. Type hints catch errors before deployment.

iii. Uvicorn
ASGI server that runs FastAPI applications. Handles async requests efficiently. Simple to configure and deploy.

iv. Groq AI
Ultra-fast inference platform. Their Llama 3.1-8B model delivers sub-second response times. Traditional LLM hosting can't match this speed.

Database & Authentication

i. Supabase
Backend-as-a-Service built on PostgreSQL. Handles authentication, database, and storage. Open source, so you're not locked into proprietary systems.

ii. PostgreSQL
Relational database that's been battle-tested for decades. ACID compliance, proper constraints, performant queries.

Development Tools

i. Visual Studio Code
Code editor with excellent extension support. TypeScript and Python tooling works well out of the box.

ii. npm
Package manager for frontend dependencies. Despite its flaws, it's the standard and it works.

iii. pip
Python package manager. Simple, reliable, doesn't try to be clever.

iv. Web Browser (Chrome/Edge)
For testing and debugging. Chrome DevTools are still the best in class.

Team Member Details

1. Reg No : 223058 –Vinay G B
2. Reg No : 223013 – Deepak M