# CSCI 3240.21 Computer Networks –

# Project Proposal

**Team Members**

- Thomas Mason - Frontend Development (Android Studio)
- Matus Kudlac - Backend Development & Database
- Will Anan - Backend Development & Database

---

## Project Title

*MediGuide* — Mobile Health Catalog & AI Symptom Assistant

---

## Project Scope

*MediGuide* is a mobile application designed for Android devices (via Android Studio and emulator) that helps users explore a database of medicines, supplements, and remedies. The app supports two modes:

1. **Customer/User End**
    - ○ Browse catalog of medicines and supplements.
    - ○ View product descriptions, recommended usage, and stock availability.
    - ○ Input symptoms or health concerns into an AI assistant.
    - ○ Receive AI-driven recommendations for relevant products in the catalog.
    - ○ Simulated payment system to “purchase” products.

2. **Provider/Admin End**
    - Manage catalog items (add, edit, remove medicines/supplements).
    - Track inventory and user purchase history.

**Key Features mapped to requirements:**

- **Payment Simulation (20 pts):** Users can “checkout” items with a mock payment gateway (simulated credit card flow).
- **Database Integration (20 pts):** Medicine catalog, user accounts, and purchase history stored/retrieved from backend database.
- **AI Agent (20 pts):** An AI companion integrated into the app that interprets user symptom descriptions and suggests catalog items that could help.
- **Customer/Provider Interfaces:** Two different roles — customer-facing catalog, provider/admin management portal.

---

# Design and Implementation Plan

- **Frontend (Thomas Mason):**
    - Develop Android mobile application (Android Studio, Java/Kotlin).
    - Build user-friendly UI for browsing catalog, searching products, and interacting with AI assistant.
    - Implement checkout flow with simulated payment.
- **Backend (Matus Kudlac & Will Anan):**
    - Build RESTful backend (Node.js/Express or Flask) with endpoints for authentication, catalog, orders, and AI queries.

  - Database (Firebase Firestore or PostgreSQL) to store:
    - Users: {id, name, email, role, createdAt}
    - Catalog: {id, name, description, category, stock, price}
    - Orders: {id, userId, items, total, status, paymentInfo}
    - AI logs: {id, userId, query, recommendation, timestamp}
  - Handle payment simulation endpoint (/payments/process).
  - Integrate AI assistant via API (e.g., OpenAI Chat API).
- **AI Integration:**
  - User inputs symptoms → backend processes request → sends query to AI model.
  - AI recommends relevant catalog items by matching symptoms to available medicines/supplements.
  - Example: User: *“I have a sore throat and headache”* → AI: *“You may consider Throat Lozenges (in stock) and Ibuprofen (in stock).”*

---

## Division of Work

- **Thomas Mason**
  - Android app development (frontend).
  - UI/UX: Catalog browsing, AI assistant chat interface, checkout page.
  - Integration of frontend with backend APIs.
- **Will Anan**
  - Backend API design and implementation.

  - Database schema & management (catalog, users, orders).
  - Payment simulation system.

- **Matus Kudlac**
  - Backend API implementation and testing.
  - AI assistant integration and query processing.
  - Provider/Admin dashboard logic.

## Evaluation Plan & Risks

- **Risks:** API key management, ensuring AI responses map correctly to catalog entries, and keeping demo stable offline.
- **Mitigation:** Use a controlled subset of catalog items for testing, cache AI responses, and mock payment for reliability.
- **Success Measure:** User can input a symptom → AI recommends catalog item → user adds to cart → simulated payment processes → provider/admin sees order in system.

**Submitted by:**

Thomas Mason, Matus Kudlac, Will Anan
CSCI 3240.21 — Computer Networks
Proposal Deadline: 9/26/2025