# CNCT Backend Design Doc

## Intro

CNCT is short for connect. It's a web app where FIU students can easily create or join plans like volleyball, running, and study groups. The purpose is to promote real-world connections among students by offering an easy-to-use digital space combining plan scheduling, messaging, and visibility on who's attending.

The backend team's role is to ensure all data interactions, user authentication, and storage operations are secure, organized, and scalable. This document is a rough outline of the backend system architecture, planned development stages, and integrations with the frontend. It's meant to be a unified roadmap allowing the backend team to stay aligned and consistent across all stages of development.

## Current Status

At this point, frontend team is moving onto coding the landing, home, settings, and profile pages after wireframing and planning. The backend team has completed their initial research and testing phase which included:

- Exploring supabase features for database and authentification
- Research on image storage and moderation
- Testing prototypes and mock datasets
- Planning database schema

The backend team will now be entering the next phase with a focus on establishing the foundation of our system. This includes finalizing/creating database tables, RLS policies, defined middleware, and architecture.

## Architecture

### Primary Architecture: 3-tier

CNCT will adopt a 3-tier architecture, the three layers will be:

- Presentation Layer – React frontend that presents info to user and send requests to backend
- Logic Layer – Express server that processes requests, applies validations, handles authentifications, and communications with data layer
- Data Layer – Supabase that stores and retrieves all data like user details, events, messages, etc.

### Secondary Architecture: Repository

The Repository Pattern serves as an additional abstraction between the backend logic and the database. Instead of writing raw Supabase queries directly inside route handlers, the backend will use repository files to perform all data operations. This allows us to:

- Keeps route code clean and readable.
- Allows easy migration if the team changes the database system later.
- Improves testing and debugging.
- Encourages consistent database logic and naming conventions.

## Middleware

Middleware acts as a checkpoint between the client request and backend routes. Every request passes through a series of functions that can verify, modify, or reject it before it reaches the route logic.

### Planned Middleware for CNCT

- cors() – Allows the frontend to communicate with the backend.
- express.json() – Parses incoming JSON data.
- morgan() – Logs HTTP requests for debugging.
- verifyUser (custom) – Checks the validity of the user's Supabase JWT token.
- errorHandler (custom) – Catches and formats server errors into clean responses.

### Potential Future Middleware

- uploadHandler for integrating the upcoming image upload API.
- Basic rate limiting or spam control for chat routes.

This setup keeps the backend modular and secure while remaining lightweight enough for the expected user base (roughly 100–200 users).

## Frontend Integration

The frontend communicates with the backend through REST API endpoints hosted on Express. All responses will use standardized JSON structures to maintain predictability. Frontend fetches will use fetch() or axios calls to retrieve and send data. Authentication tokens will be included in request headers to verify identity.

## Features Overview

The backend supports the following core features:

- User Authentication: FIU email-restricted sign up/login using Auth.js and Supabase.
- Plan Creation & Joining: Users can create, view, and join activity plans.
- Messaging: Each plan includes a chat thread for participants.

- Profile Management: Each user has a profile page with editable bio and picture.
- Image Uploads: Handles storage and future moderation of images.
- "Who's Going" List: Displays attendees for each plan to encourage engagement.

Each feature ties directly into a Supabase table and a corresponding API route.

## Development Phases & Timeline

### Phase 1: Foundation (COMPLETED)

Focus: Setup, research, and early Supabase structure.
Status: Complete
Deliverables:

- Supabase instance created and connected
- Initial tables created
- Image storage API researched
- Basic repo setup for backend

### Phase 2: Core Infrastructure (In Progress: Oct 28 – Nov 10)

Focus: Database + Authentication
Tasks:

- Finalize all Supabase tables & ERD
- Write and test RLS policies
- Integrate Auth.js with Supabase JWTs
- Enforce FIU email restriction
- Implement verifyUser middleware
- Begin repository layer setup
  Dependencies: None (can continue even if frontend still working on UI).
  Frontend Coordination: Confirm which pages will use authenticated access first (e.g. Home, Plans).

### Phase 3: API Routes + Image Upload (Nov 10 – Nov 20)

Focus: Express API development & storage integration
Tasks:

- Build main endpoints (/api/plans, /api/users, /api/messages, /api/join)
- Implement repository pattern for cleaner data access
- Add upload middleware for images
- Connect Supabase Storage for image saving
- Return sample JSON for frontend testing
  Parallel Work: Frontend continues building forms and UI that will later call these routes.
  Goal: Backend functional before full frontend integration begins.

**Phase 4: Integration, Testing & Finalization (Nov 20 – Dec 1)**

Focus: Connecting backend with frontend and final polishing
Teams: All backend members (joint effort)
Tasks:

- Test endpoint connections with frontend pages
- Validate the authentication flow end-to-end
- Run mock user tests and fix bugs
- Document all API routes and data structures for frontend reference
- Prepare presentation/demo version with sample users and data
  Deliverables:
- Working CNCT web app with user auth, plan creation, chat, and image upload
- Final architecture and README documentation complete by Dec 1