**🚀 Mivton Project - Phase 2.1 Context Prompt**

**I am working on MIVTON - a futuristic multilingual chat platform. You are my technical developer. We have completed ALL of Phase 1 (Foundation & Core Setup) and now need to implement Phase 2.1 - Dashboard Framework.**

**🎯 Project Overview**

* **Name**: Mivton
* **Purpose**: Friends-only multilingual text chat with real-time OpenAI translation
* **Target**: Gen Z users with futuristic design
* **Status**: Phase 1 ✅ COMPLETE, Phase 2.1 🔄 IN PROGRESS

**✅ PHASE 1 - COMPLETE FOUNDATION**

All Phase 1 objectives have been successfully achieved:

**Phase 1.1 - Project Initialization ✅ COMPLETE**

* **Landing Page**: Live at https://mivton-production.up.railway.app
* **Domain**: mivton.com (connected and operational)
* **Infrastructure**: Express.js + Socket.IO + Railway deployment
* **Design**: Futuristic Gen Z aesthetic with glassmorphism effects

**Phase 1.2 - Database Setup ✅ COMPLETE**

* **Database**: PostgreSQL fully operational on Railway
* **Tables**: users, session, waitlist, user\_stats
* **Features**: Waitlist system, database health checks, connection pooling
* **API Endpoints**: /health, /api/status, /api/waitlist, /api/waitlist/stats

**Phase 1.3 - Authentication System ✅ COMPLETE**

* **User Registration**: Complete with email validation and bcrypt hashing
* **User Login**: Secure session-based authentication
* **Futuristic UI**: login.html, register.html, dashboard.html (basic)
* **Email Integration**: Welcome emails via Hostinger SMTP
* **Security**: Rate limiting, CSP headers, input validation
* **API Endpoints**: /api/auth/\* (register, login, logout, me, status, check-username, check-email)

**🔄 CURRENT PHASE 2.1 - Dashboard Framework (Day 1)**

**Goal**: Create main dashboard with modern UI components and navigation **Priority**: IMPLEMENT THIS PHASE

**Phase 2.1 Tasks:**

* [ x] Create main dashboard HTML structure
* [ x] Implement sidebar navigation with glassmorphism
* [x ] Add responsive mobile navigation
* [ x] Create user profile section
* [ x] Implement dark theme with neon accents
* [ x] Test: Dashboard loads and navigation works

**Required Dashboard Features:**

1. **Main Dashboard Structure**
   * Modern layout with sidebar + main content area
   * Responsive design for mobile/tablet/desktop
   * Header with user info and notifications
   * Navigation between different sections
2. **Sidebar Navigation**
   * Friends Tab 🫂 (with count)
   * Requests Tab 📨 (with count badges)
   * Find Tab 🔍 (user search)
   * Blocked Tab 🚫 (blocked users)
   * Settings/Profile section
3. **User Profile Section**
   * Display current user info
   * Language preferences
   * Online status indicator
   * Quick settings access
4. **Modern UI Components**
   * Glassmorphism effects
   * Hover animations
   * Loading states
   * Notification badges
   * Modal components

**🗄️ Database Schema (Ready for Dashboard)**

-- Users table ready for dashboard integration

users (

id, username, email, password\_hash, full\_name,

gender, native\_language, is\_verified, is\_admin,

admin\_level, is\_blocked, status, last\_login,

created\_at, updated\_at

);

-- Session table for authentication

session (sid, sess, expire);

-- Waitlist table for analytics

waitlist (id, email, referrer, user\_agent, ip\_address, created\_at, notified);

**🔧 Technical Infrastructure (Production Ready)**

**Railway Environment Variables ✅**

NODE\_ENV=production

JWT\_SECRET=mivton-super-secret-jwt-key-2025-production

OPENAI\_API\_KEY=sk-proj-ssWG4RYWzRSkC6N5wSrwR-ajRzmcGMLG2agNfBO1IRpplc7a8LzwmHrKNeShj4J2gm8ynWDu\_2T3BlbkFJob8\_1Ny3bs5vVdEKNo48BEsXZOB4kGzgtQ-O0-JAssePAFHU7CFAi-cLPc-xYop2z362vHtd4A

SMTP\_HOST=smtp.hostinger.com

SMTP\_PORT=465

SMTP\_SECURE=true

SMTP\_USER=info@mivton.com

SMTP\_PASS=Bacau@2012

APP\_URL=https://mivton.com

FRONTEND\_URL=https://mivton.com

DATABASE\_URL=[Railway PostgreSQL URL]

**Current File Structure ✅**

mivton/

├── server.js # Main Express server ✅

├── package.json # Dependencies ✅

├── .env # Environment variables ✅

├── public/

│ ├── index.html # Landing page ✅

│ ├── login.html # Login page ✅

│ ├── register.html # Register page ✅

│ ├── dashboard.html # Basic dashboard ✅ (NEEDS ENHANCEMENT)

│ ├── css/

│ │ ├── style.css # Landing page styles ✅

│ │ └── auth.css # Auth page styles ✅

│ └── js/

│ ├── app.js # Landing page logic ✅

│ ├── auth.js # Auth page logic ✅

│ └── validation.js # Validation logic ✅

├── database/ # Database layer ✅

│ ├── connection.js # PostgreSQL connection ✅

│ ├── schema.sql # Core schema ✅

│ └── init.js # DB initialization ✅

├── middleware/ # Middleware layer ✅

│ ├── database.js # DB middleware ✅

│ ├── auth.js # Auth middleware ✅

│ └── validation.js # Validation middleware ✅

├── utils/ # Utility layer ✅

│ ├── database.js # DB utilities ✅

│ ├── auth.js # Auth utilities ✅

│ ├── email.js # Email system ✅

│ ├── validation.js # Validation helpers ✅

│ └── waitlist.js # Waitlist functions ✅

├── routes/ # API routes ✅

│ ├── auth.js # Authentication routes ✅

│ └── users.js # User management routes ✅

└── socket/ # Socket.IO handlers (ready for Phase 4)

**🎨 Design System (Established)**

Primary: #6366f1 (Electric Blue)

Secondary: #8b5cf6 (Vibrant Purple)

Accent: #06b6d4 (Cyan)

Success: #10b981 (Green)

Warning: #f59e0b (Amber)

Error: #ef4444 (Red)

Background: #0f172a (Dark Navy)

Surface: #1e293b (Slate)

Text: #f1f5f9 (Light)

**💻 Technology Stack**

* **Backend**: Node.js + Express.js ✅
* **Database**: PostgreSQL (Railway) ✅
* **Authentication**: Session-based with bcrypt ✅
* **Email**: Hostinger SMTP ✅
* **Real-time**: Socket.IO ✅ (ready for Phase 4)
* **Translation**: OpenAI GPT-4 (ready for Phase 5)
* **Frontend**: HTML5 + CSS3 + Vanilla JS ✅
* **Deployment**: Railway CLI (no GitHub) ✅

**📋 Phase 2.1 Specific Requirements**

**Files to Create/Enhance in Phase 2.1:**

public/

├── dashboard.html # 🔄 ENHANCE - Complete dashboard redesign

├── css/

│ ├── dashboard.css # 🆕 NEW - Dashboard-specific styles

│ └── components.css # 🆕 NEW - Reusable UI components

└── js/

├── dashboard.js # 🆕 NEW - Dashboard functionality

└── components.js # 🆕 NEW - UI component logic

routes/

└── dashboard.js # 🆕 NEW - Dashboard API routes

utils/

└── dashboard.js # 🆕 NEW - Dashboard utilities

**New API Endpoints for Phase 2.1:**

* GET /api/dashboard/stats - User dashboard statistics
* GET /api/dashboard/recent-activity - Recent user activity
* PUT /api/dashboard/profile - Update user profile
* PUT /api/dashboard/settings - Update user settings

**Dashboard Navigation Structure:**

1. **Header Section**
   * User avatar and name
   * Online status indicator
   * Notifications bell
   * Settings dropdown
2. **Sidebar Navigation**
   * Friends (count badge)
   * Requests (count badge)
   * Find Users
   * Blocked Users
   * Settings
3. **Main Content Area**
   * Dynamic content based on selected tab
   * Chat area preparation
   * User profile display

**🔮 Phase 2.2 Preview (Modern UI Components)**

**After Phase 2.1, we'll implement:**

* Custom buttons with hover animations
* Card components with glass effects
* Loading animations and micro-interactions
* Icon library integration
* Notification toast system
* Modal components with blur backgrounds

This context helps prepare dashboard framework for advanced UI components.

**👥 Our Roles - IMPORTANT**

* **You (Claude)**: Create ALL files directly in the mivton directory structure. I CANNOT copy, modify, or create files manually.
* **Me**: Only run railway up to deploy the changes you create.

**🚨 CRITICAL INSTRUCTIONS FOR CLAUDE**

1. **CREATE FILES DIRECTLY**: Use the file creation tools to create all files in the correct mivton directory structure
2. **NO COPY-PASTE INSTRUCTIONS**: I cannot copy/paste code - you must create the actual files
3. **COMPLETE FILE CREATION**: Create entire files, not just code snippets
4. **DIRECTORY STRUCTURE**: Maintain proper folder organization
5. **ALL MODIFICATIONS**: Any fixes or updates must be done by creating/updating files directly

**🚀 What I Need You To Do**

I need you to implement Phase 2.1 - Dashboard Framework by:

1. **Enhancing dashboard.html** - Complete redesign with modern navigation
2. **Creating dashboard.css** - Futuristic styling with glassmorphism
3. **Building dashboard.js** - Interactive functionality and navigation
4. **Adding dashboard routes** - API endpoints for dashboard data
5. **Creating UI components** - Reusable components for future phases
6. **Testing integration** - Ensure all dashboard features work with existing auth

Please create all the necessary files directly in the mivton directory structure and provide deployment instructions.

**🎯 Success Criteria**

* Modern dashboard with sidebar navigation working
* Responsive design for all devices
* User profile section displaying current user data
* Navigation between different sections
* Glassmorphism effects and Gen Z aesthetic
* All dashboard features working on production after deployment

**END OF PROMPT - Ready for Phase 2.1 Dashboard Framework Implementation**