

HIIT Gym Manager

Rome Was Built In A Day

Vansh Sood (u23534402)

Denis Woolley (u23528860)

Jared Hurlimann (u23543932)

Jason Mayo (u23587572)

Amadeus Fidos (u22526162)

Technical Installation Manual

Version 1.0
August 20, 2025

Contents

1	Introduction	2
2	Prerequisites	2
2.1	System Requirements	2
2.2	Required Software	2
2.3	Mobile Development Requirements	2
3	Web Application Installation	3
4	Mobile Application Installation	4
5	Environment Configuration	5
5.1	Web Application Variables	5
5.2	Mobile Application Variables	5
6	Running the Applications	5
6.1	Web Application	5
6.2	Mobile Application	5
7	Troubleshooting	6
7.1	Common Issues	6
8	Development Workflow	7
8.1	Code Quality	7
8.2	Git Hooks	7
8.3	Testing	7
9	Deployment	8
9.1	Web Application Deployment	8
9.2	Mobile Application Deployment	8
10	Additional Resources	8

1 Introduction

The Gym Manager system is a cross-platform fitness and gym membership management solution. It includes:

A REST API for core business logic, hosted on Vercel.

A web interface built using React, also hosted on Vercel.

A mobile application built with Expo (React Native), which can be run as a PWA or installed as an Android APK.

A PostgreSQL database and live subscriptions hosted via Supabase.

This manual provides instructions to set up the development environment, clone the project, install dependencies, and deploy all components.

2 Prerequisites

2.1 System Requirements

- **Operating System:** macOS 10.15+, Windows 10+, or Ubuntu 18.04+
- **Node.js:** Version 18.17.0 or higher (LTS recommended)
- **npm:** Version 9.0.0 or higher
- **Git:** Version 2.30.0 or higher

2.2 Required Software

- **Code Editor:** VS Code (recommended), Sublime Text, etc.
- **Terminal:** iTerm2 (macOS), Windows Terminal, etc.
- **Browser:** Chrome, Firefox, Safari, or Edge (latest)

2.3 Mobile Development Requirements

- **Expo CLI**
- **iOS Simulator:** Xcode (macOS only)
- **Android Emulator:** Android Studio
- **Physical Device:** iOS or Android

3 Web Application Installation

Step 1: Clone the Repository

```
git clone <repository-url>  
cd Gym-Manager  
cd apps/web
```

Step 2: Install Dependencies

```
npm ci  
# Alternative  
npm install
```

Step 3: Environment Configuration

Create a `.env.local` file:

```
touch .env.local
```

Add variables:

```
NEXT_PUBLIC_API_URL=http://localhost:3001  
NEXTAUTH_SECRET=your-secret-key-here  
NEXTAUTH_URL=http://localhost:3000  
NEXT_PUBLIC_SUPABASE_URL=your-supabase-url  
NEXT_PUBLIC_SUPABASE_ANON_KEY=your-supabase-anon-key
```

Step 4: Verify Installation

```
npm list --depth=0  
npx tsc --noEmit
```

4 Mobile Application Installation

Step 1: Navigate to Mobile Directory

```
cd apps/mobile
```

Step 2: Install Expo CLI

```
npm install -g @expo/cli  
expo --version
```

Step 3: Install Dependencies

```
npm install  
npx expo install expo-dev-client
```

Step 4: Environment Configuration

```
{  
  "expo": {  
    "extra": {  
      "EXPO_PUBLIC_SUPABASE_URL": "your-supabase-url",  
      "EXPO_PUBLIC_SUPABASE_ANON_KEY": "your-supabase-anon-key",  
      "EXPO_PUBLIC_API_URL": "your-api-url"  
    }  
  }  
}
```

Step 5: Install iOS Dependencies (macOS only)

```
cd ios  
pod install  
cd ..
```

5 Environment Configuration

5.1 Web Application Variables

Variable	Description	Required	Default
NEXT_PUBLIC_API_URL	Backend API URL	Yes	localhost:3001
NEXTAUTH_SECRET	NextAuth secret key for session encryption	Yes	-
NEXTAUTH_URL	NextAuth callback URL for authentication	Yes	localhost:3000
NEXT_PUBLIC_SUPABASE_URL	Supabase project URL for database connection	No	-
NEXT_PUBLIC_SUPABASE_ANON_KEY	Supabase anonymous key for public access	No	-

5.2 Mobile Application Variables

Variable	Description	Required	Default
EXPO_PUBLIC_SUPABASE_URL	Supabase project URL for mobile app	Yes	-
EXPO_PUBLIC_SUPABASE_ANON_KEY	Supabase anonymous key for mobile access	Yes	-
EXPO_PUBLIC_API_URL	Backend API URL for mobile requests	Yes	-

6 Running the Applications

6.1 Web Application

```
cd apps/web
npm run dev      # Development mode
npm run build    # Build for production
npm run start    # Run production build
```

6.2 Mobile Application

```
cd apps/mobile
npm start          # Start Expo development server
npm run android    # Run on Android emulator
npm run ios        # Run on iOS simulator
npm run web        # Run web version
```

7 Troubleshooting

7.1 Common Issues

Port already in use:

```
# Kill process using port 3000
lsof -ti:3000 | xargs kill -9
```

Dependency issues:

```
# Clean npm cache and reinstall
npm cache clean --force
rm -rf node_modules package-lock.json
npm install
```

Expo CLI issues:

```
# Update Expo CLI
npm install -g @expo/cli@latest
expo doctor
```

iOS build issues (macOS):

```
# Clean iOS build
cd ios
rm -rf build
pod deintegrate
pod install
cd ..
```

8 Development Workflow

8.1 Code Quality

```
# Linting
npm run lint
npm run lint:fix

# Formatting
npm run format
npm run format:fix

# Type checking
npm run type-check
```

8.2 Git Hooks

```
# Setup pre-commit hooks
npm run prepare

# Standard commit flow
git add .
git commit -m "feat: your commit message"
git push origin main
```

8.3 Testing

```
# Run tests
npm test
npm run test:coverage

# Integration testing
npm run test:integration
```


9 Deployment

9.1 Web Application Deployment

```
# Build for production  
npm run build  
  
# Deploy to Vercel  
npm install -g vercel  
vercel --prod
```

9.2 Mobile Application Deployment

```
# Build for production  
expo build:android  
expo build:ios  
  
# Submit to stores  
expo submit --platform=android  
expo submit --platform=ios
```

10 Additional Resources

- [Next.js Documentation](#) - Complete guide for Next.js framework
- [React Native Documentation](#) - Official React Native docs
- [Expo Documentation](#) - Comprehensive Expo development guide
- [Supabase Documentation](#) - Database and authentication setup
- [Tailwind CSS Documentation](#) - Styling framework guide