Norvia Backoffice

Step-by-step setup and local development guide | .NET 10 + Azure SQL + AI Coding Agent

Step 1: Set Up the Local Machine for .NET Development

Install the required development tools on your Windows or macOS computer, including the .NET 10 SDK and Visual Studio Code. Verify the installation and complete any prerequisites. Follow the Local Machine Setup document.

Step 2: Set Up Azure Portal, Azure SQL Server, and Database

Configure your Azure account and create or select the Azure SQL logical server and development database. Make sure SQL authentication is configured and your local IP address is permitted by the database firewall. Follow the Azure Setup document.

Step 3: Create the Required Database Tables

Connect to the Azure SQL development database and execute the SQL scripts provided for the Norvia Labs client requirements. Confirm that all required tables, roles, and initial data have been created successfully. Follow the Database Schema document.

Step 4: Configure an AI Coding Tool in VS Code

Open Visual Studio Code and set up an AI coding assistant, such as Codex or Claude Code. Complete its installation, sign-in, and any required permissions before starting development.

Step 5: Create the Local Project Folder

Create a new folder on your computer named NorviaBackoffice. This folder will contain the complete .NET application source code.

Step 6: Open the Project Folder and Coding Agent

In VS Code, select File > Open Folder and open NorviaBackoffice. Launch your configured AI coding agent in VS Code.

Step 7: Generate the .NET Project Using the Prompt

Provide the coding agent with the Norvia Backoffice Phase 1 development prompt. Ask it to create the .NET 10 MVC project in the currently opened folder, including the initial UI, authentication, and project structure. Follow the Project Development Prompt document.

Step 8: Configure Local Credentials

Fill in the required configuration values for Azure SQL (server, database, SQL username and password), Gmail SMTP (email address and App Password), and any application settings. Use .NET User Secrets or local environment variables for actual credentials; keep only placeholders in committed configuration files. Do not share or commit passwords.

Step 9: Run the Project Locally

Ask the AI coding agent to restore dependencies, build the project, resolve any compilation errors, and run the application locally. Open the localhost URL displayed in the terminal and verify that the login page loads and connects to the development database.

Suggested prompt for Step 9 (copy into Codex or Claude Code)

"Please run the NorviaBackoffice .NET 10 project locally. First verify the .NET SDK and required configuration, then execute dotnet restore, dotnet build, and dotnet run (or dotnet watch run). Fix any build or startup errors without modifying the existing Azure SQL schema. Tell me the localhost URL and explain how to stop the application."

Completion check: The application starts on localhost, displays the login page, and can access the configured Azure SQL development database.