### **Prerequisites**

Before you begin, ensure you have the following installed on your machine:

* Node.js
* npm or yarn
* .NET SDK
* SQL Server Management Studio (SSMS)
* A code editor like Visual Studio Code or Visual Studio

### **Step 1: Set Up the Database**

1. **Import the Schema to SSMS:**
   * Open SSMS and connect to your SQL Server instance.
   * In the frontend folder of your project, locate the schema folder.
   * Right-click on your database, and select Tasks -> Import Data...
   * Follow the wizard to import the schema files into your database.
2. **Switch Connection Strings:**
   * In your frontend project, find the configuration file (usually config.js or .env.local) and update the connection string to point to your local or production database.
   * Similarly, in your backend project, locate the appsettings.json file and update the connection string under the ConnectionStrings section.

### **Step 2: Set Up the Frontend (Next.js)**

1. **Navigate to the Frontend Directory:**

cd path/to/your/frontend

1. **Install Dependencies:**

npm install

or if you are using yarn:

yarn install

1. **Run the Development Server:**

npm run dev-to run Frontend But you have to be in folder C:\Users\HR\Desktop\outofservice2\frontend\pages

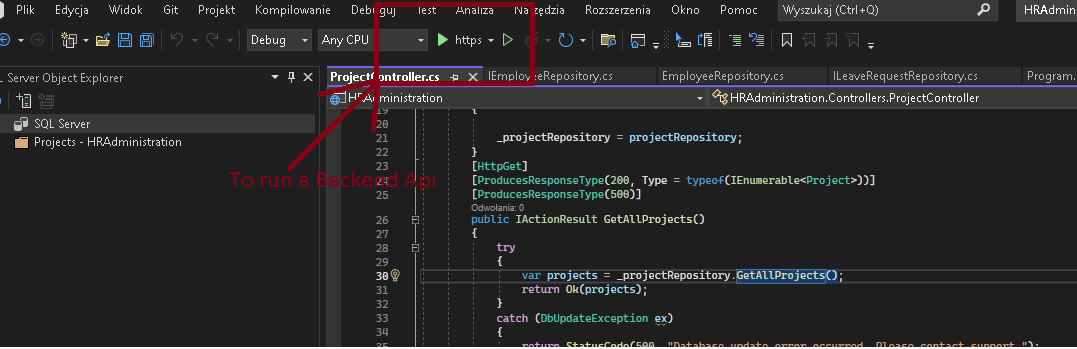
or if you are using yarn:

yarn dev

* + Your Next.js application should now be running at <http://localhost:3000>.

### **Step 3: Set Up the Backend (Swagger UI with C#)**

1. **Navigate to the Backend Directory: --You have to open it THROUGH Visual Studio 2022**



1. **Restore .NET Dependencies:**

dotnet restore

1. **Update Database Connection String:**
   * Open appsettings.json and update the connection string to point to your SQL Server database.
2. **Run the Application:**

dotnet run

* + Your backend should now be running, and the Swagger UI should be accessible at https://localhost:7091/swagger/index.html.

### **Step 4: Testing the Setup**

1. **Access the Frontend:**
   * Open your browser and go to <http://localhost:3000>.
2. **Access the Backend Swagger UI:**
   * Open another browser tab and go to <http://localhost:5000/swagger>.
3. **Verify Database Connection:**
   * Ensure that both the frontend and backend can communicate with the database by performing CRUD operations through the UI and Swagger endpoints.

### **Summary of Files and Commands**

* **Frontend Directory:**
  + Install dependencies: npm install or yarn install
  + Run the development server: npm run dev or yarn dev
  + Configuration file: .env.local or config.js
* **Backend Directory:**
  + Install dependencies: dotnet restore
  + Run the application: dotnet run
  + Configuration file: appsettings.json

env

DATABASE\_URL=your\_database\_connection\_string  
API\_URL=http://localhost:7091

#### **appsettings.json (Backend)**

{  
 "ConnectionStrings": {  
 "DefaultConnection": "Server=your\_server;Database=your\_database;User Id=your\_user;Password=your\_password;"  
 },  
 "Logging": {  
 "LogLevel": {  
 "Default": "Information",  
 "Microsoft": "Warning",  
 "Microsoft.Hosting.Lifetime": "Information"  
 }  
 },  
 "AllowedHosts": "\*"  
}