

TECHNICAL TEST

SR DEVELOPER FULLSTACK

**Project Overview:**

A large real estate company requires the creation of an API to fetch information about properties stored in a database, and a web page to display this information. The goal is to create a full-stack application using .NET, MongoDB, C, and ReactJS or Next.js. The project will involve building a set of services to manage property data and create a user interface to display it.

**Technologies Required:**

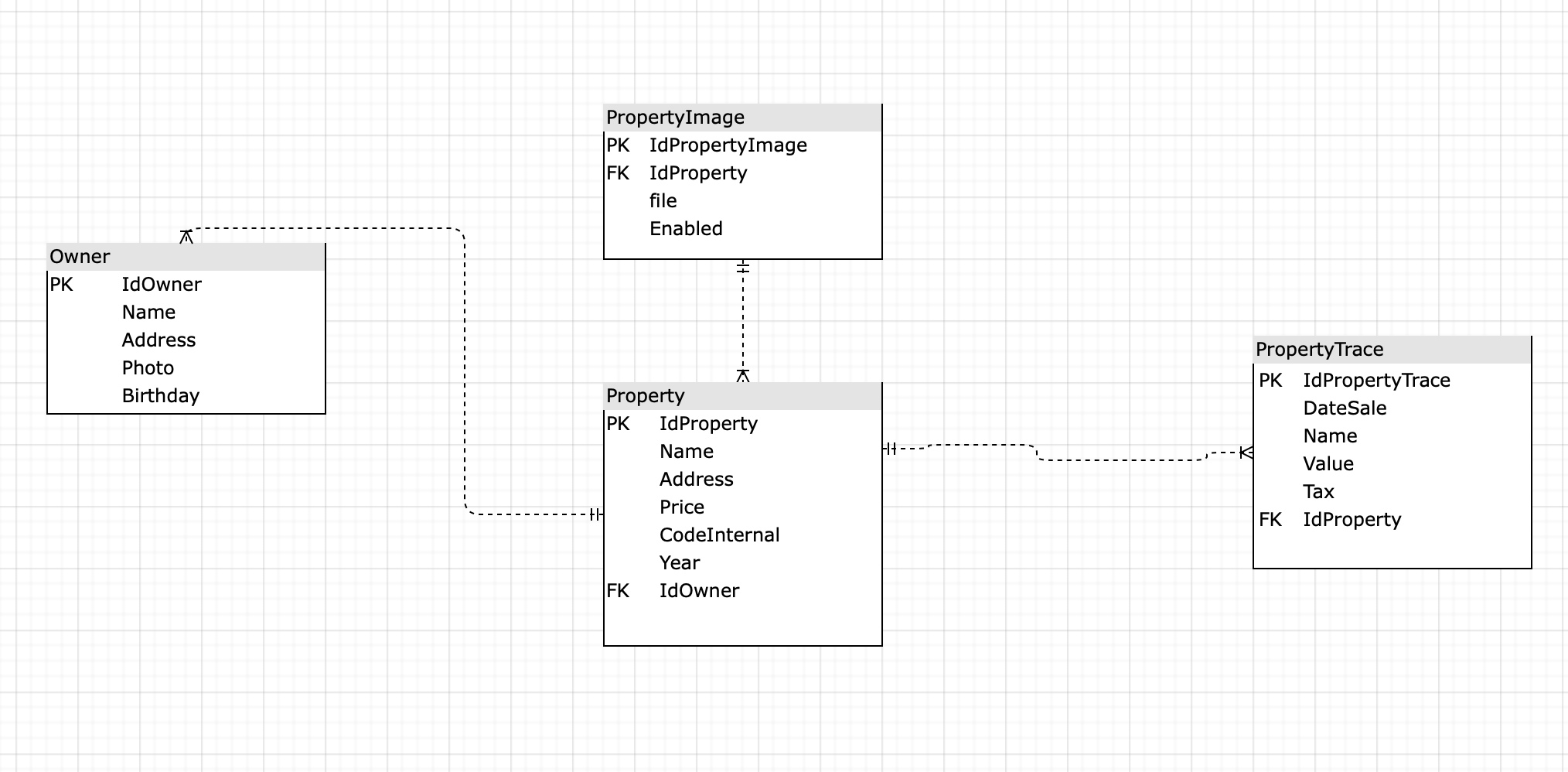
* Backend: .NET 8 or 9, C
* Database: MongoDB
* Frontend: ReactJS or Next.js
* Testing: NUnit for unit testing

**Task Breakdown:**

1. **Backend (API) Development:**
   * Create a C API using .NET 8 or 9 to retrieve property data from the MongoDB database.
   * Implement ﬁlters in the API for retrieving a list of properties based on parameters like name, address and range price.
   * Deﬁne Dtos ﬁelds such as:
     + IdOwner, Name, Address Property, Price Preoprty and just one image.
2. **Frontend (Web Page) Development:**
   * Create a web page using ReactJS or Next.js to display the property data.
   * The page should include:
   * A list of properties, fetched from the API.
     + Filters for searching properties (name, address and range price).
     + Option to view more details about individual properties.
   * Ensure the frontend is responsive, providing a seamless user experience across different devices.
3. **Evaluation Criteria:**
   * Backend and Frontend Architecture: Implement a clean and efﬁcient architecture for both the backend API and the frontend web page.
   * Code Structure: Organize code in a modular and maintainable way.
   * Documentation: Provide clear and concise documentation for both the API and frontend code.
   * Best Practices: Follow best practices in both backend and frontend development, including:
     + Clean architecture
     + Proper error handling
     + Optimized queries to database (if required)
   * Performance: Ensure both the API and frontend are optimized for performance, especially for handling large datasets or multiple ﬁlters.
   * Unit Testing: Write unit tests for both the backend API and the frontend components using NUnit or other appropriate testing frameworks.
   * Clean Code: Ensure that the code is readable, maintainable, and follows established coding conventions.

**By completing this project, you'll demonstrate your ability to build a full-stack application with a clean architecture, good performance, and proper testing and documentation.**

**This version organizes the requirements in a more structured way and clariffies the tasks and evaluation criteria.**

****

**Send solution with project in zip ffile or upload from github or other similar site. Attach database backup. if is necessary specify steps for run project when download.**

**Please give access to the following emails:**

**\***[**hr@millionluxury.com**](mailto:hr@millionluxury.com)

**\***[**amorau@millionluxury.com**](mailto:amorau@millionluxury.com)