**Note Taking API**

This is a RESTful API built with .NET 6 and C# for taking and storing notes. The API is connected to a SQL Server database and allows users to retrieve, create, update and delete notes.

**Features**

* Create new notes with a title, description and creation date.
* Retrieve all notes in the database.
* Retrieve a single note by its id.
* Update a note by its id.
* Delete a note by its id.

**Prerequisites**

Before you begin, make sure you have the following installed:

* .NET 6 SDK
* Visual Studio Code or Visual Studio
* SQL Server

**Getting Started**

1. Clone the repository to your local machine:

bashCopy code

git clone https://github.com/yourusername/note-taking-api.git

1. Open the solution file in Visual Studio Code or Visual Studio.
2. Update the connection string in the appsettings.json file to match your SQL Server connection information.

4. Run the API project by pressing F5 or by clicking the Run button.

5. The API will be hosted at <https://localhost:7111/api/notes>. You can test the API using a tool like Postman or Swagger.

**API Endpoints**

The API has the following endpoints:

* **GET /api/notes**: Retrieve all notes.
* **GET /api/notes/{id}**: Retrieve a single note by its id.
* **POST /api/notes**: Create a new note.
* **PUT /api/notes/{id}**: Update a note by its id.
* **DELETE /api/notes/{id}**: Delete a note by its id.

**Running the Tests**

The project includes xUnit unit tests for testing the API. To run the tests, follow these steps:

1. Open the solution file in Visual Studio Code or Visual Studio.
2. Navigate to the NoteTakingAPI.Tests project.
3. Right-click on the project and select Run Tests.
4. The tests will run and display the results in the Test Explorer window.

**Built With**

* [.NET 6](https://dotnet.microsoft.com/download/dotnet-core/6.0)
* [C#](https://docs.microsoft.com/en-us/dotnet/csharp/)
* [ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-6.0)
* [Entity Framework Core](https://docs.microsoft.com/en-us/ef/core/)
* [SQL Server](https://www.microsoft.com/en-us/sql-server/sql-server-downloads)

**Additional functionalities that can be added to the back-end API:**

1. Authentication and Authorization - Implement OAuth2 or JWT tokens for secure access to the API, and add roles and permissions to control who can perform certain actions on the notes.
2. Pagination - Add support for pagination to the API so that it can return a limited number of notes per request, improving performance and reducing the amount of data transferred.
3. Searching - Add a search functionality to the API so that users can search for notes by title, description, or creation date.
4. Sorting - Allow users to sort the list of notes by title, description, or creation date in ascending or descending order.
5. Versioning - Implement versioning for the API to allow for backwards compatibility and easier updates.
6. Logging - Add logging to the API to keep track of requests, responses, and errors.
7. Backup and Recovery - Implement a backup and recovery solution for the database, to ensure data is safe in the case of a disaster.
8. Unit and Integration Testing - Add unit and integration tests to the API to ensure that it works as expected, and to prevent regressions.
9. Performance Optimization - Optimize the performance of the API by adding caching, indexing, and optimizing queries.
10. Documentation - Add detailed documentation to the API, including examples of how to use it, endpoints, parameters, and response formats.