**Full-Stack Developer Take-Home Assessment**

**Project Overview**

As part of our selection process, we invite you to complete a take-home project that encompasses both front-end and back-end development skills. This project is designed to assess your ability to create a simple, functional web application using C#, JavaScript, and HTML5.

**Objective:**

Your task is to develop a Simple Note-Taking Application that allows users to create, read, update, and delete notes. This application should consist of a front-end built with JavaScript and HTML5, and a back-end developed with C#.

**Requirements**

**Front-End**

Design a responsive user interface for note operations.

Include elements for note creation, display, editing, and deletion.

**Use JavaScript for dynamic content updates, including:**

Adding new notes.

Editing and deleting existing notes.

Displaying all notes from the back-end.

**Styling**:

Ensure the application is visually appealing and user-friendly.

**Back-End**

**API:**

Develop a RESTful API using C# (ASP.NET Core recommended) to handle CRUD operations for notes.

Implement endpoints for creating, retrieving, updating, and deleting notes.

**Data Storage:**

Use file storage or a lightweight database (e.g., SQLite) for data persistence.

**Error Handling:**

Incorporate basic error handling to manage exceptions and provide meaningful error responses.

**Optional Challenges:**

Implement user authentication.

Add functionality to tag notes and filter them by tags.

Add functionality to share notes between different users.

**Submission Guidelines**

Provide the complete source code for both the front-end and back-end components of the project.

Include a README file with:

Instructions on setting up and running the application.

A list of any dependencies that need to be installed.

Use Git for version control. Hosting the project on GitHub or a similar platform is encouraged for easy access and review.

**Assessment Criteria**

Functionality: The application should function as specified.

Code Quality: Your code should be clean, well-organized, and follow best practices.

Design and Usability: The application should be reactive and have a user-friendly interface.