Technical Screening Coding Exercise

Goal:

This is an exercise to simulate our daily work: a ticket is assigned to you, and you are expected to complete the task timely with a high-quality solution. Through this coding exercise, you can partially demonstrate the technical skills required in the job description.

Expectation:

1. Before coding, please estimate how many hours you may need for this task.
2. Track your time during the implementation and report the actual hours you spent.
3. Create a Git repository and commit your code there.
4. Create a `README.md` file and document how to run your app.
5. Complete the task independently. Note that you are working on this coding exercise in the same time frame as other applicants, please don't share this file through internet.
6. It is OK if you are not able to complete all features, but please submit your results by the due date. Also, please review the "Task Submission Checklist" before responding to us.

Task Description:

Assume your team is working on developing a web app that is used by university employees. In the real world, your team will have a set of documents before implementation. However, for this coding exercise, you don't need to consider peripheral requirements because you are not going to implement a full-fledged application. Please only focus on basic functionalities.

Your task is to implement a feature which allows an employee to submit a receipt for reimbursement. To demonstrate the feature, you need to build a minimal app, which includes a front-end web interface and a back-end API service.

The web page has a form that allows an employee to submit required information for a purchase. The employee is required to fill in the date, the amount, and a description of the purchase, and upload a file that contains a receipt. A successful form submission will send information to the backend.

The API service should accept those requests and correctly parse the data. Take the data and persist it in a database of your choosing (SQLite, SQL Server Express, MySQL, and PostgreSQL are some examples).

If you think some reasonable business rules would help your implementation, or some coding practice would benefit your solution, please feel free to put those into action. If so, please document your assumptions, highlight of your practice, and so on, and report to us.

For this coding exercise, you are allowed to use your favorite programming languages, frameworks, and tools although the latest versions of .NET and Angular are preferred.

Task Submission Checklist:

1. The link to your Git repository.
    a. This repository must be visible to us, and it's easier if you set it public. We will review the code and the commit history.
    b. The repository must have a `README.md` file which documents how to run your app.
2. The numbers of the estimated hours and the actual hours. You can optionally write a short paragraph with details and explanations.
3. Reasons for choosing your tech stack for this task. You can write a short paragraph to justify your choice among other alternatives.
4. Comments (Optional). You can describe the following:
    a. The assumptions you made during implementation.
    b. The problems you came across, and how you solved them.
    c. The highlights in your code or coding practice that you want us to notice. d. Anything else.