

Technical assessment

The story:

Part one:

Creating sample web API for storing documents with some metadata, and accept request to get documents itself or information about it (metadata).

Part two

Create sample web interface for uploading document(s) or display available documents in table, each row has a link to display document information(s), editing and deleting.

Preferred technologies:

Backend: C# .Net (ASP.Net core Framework)

Database: use any server database, we prefer (SQL Server)

Frontend: use any well-known frontend framework, (we prefer Angular framework).

Source control: Git (GitHub).

Architecture: MVC structure.

Video: discussing your task is plus.

Documentation: Microsoft word.

Part One API Functions:

Develop the samples functions:

- **Insert** document.
- **Update** document (information).
- **Get document** information.
- **Download** document (Optional).
- **Update** document files (Optional).
- **Delete** document (delete full record).

Database:

The database tables

Table: user: contains user main information

- ID (Guide)
- Name

- email
- Address
- password

Table: documents: contains documents main information

- ID (auto id incrementable)
- Name
- Date
- Created date
- Priority
- Due_date

Table: Priorities: contains all possible priorities

- ID
- Name
- Add these priorities (Low, High, Critical)
- File_path -- the path of files

Note:

1- can use a user (model) to link with the user who uploaded the document and register or login.

2-each document contains at least one file (one to many relation)

3- can add additional priorities to models if you need this

Part two: The Front end:

The frontend contains 4 pages:

- **Page (1)** Upload document with information.
Create a page to upload a document (upload is a multiframe upload) and input fields for providing document information, create a folder in your same project directory name it uploads for uploaded documents.
- **Page (2)** displays all available documents information in a tabular data each row as link to *open document* information, link to *edit information*, and link to *delete document*
- **Page (3)** display document information from selected document in page (2).
- **Page (4):** Editing document information from selected document in page (2).
- **Page (5):** login page as well as register
- • Updated document files (add new files, remove current files) Optional
- • Updated document files (add new files, remove current files) Optional
- Download selected document is optional.
- Upload progress for files (Optional)

Assessment Grading

You assessment will be evaluated as below:

- Application Architecture & Code style/quality & Documentation of Code.
- User Interface
- Error Handling
- Interpretation of specification
- Tests
- Time of implementation

Feel free to contact our software developer team at wael.tolba@streams.com.eg