) memorable.

Technical Assessment

Back End Developer

Overview

Thank you for applying to Memorable! The purpose of this test is to assess your technical skills on tasks that Memorable developers face every day.

We are currently developing tools based on AI models to assess the cognitive impact of videos and images. Our main product is offered through a dashboard where users can upload their assets and analyze memorability and saliency metrics. This simple technical assessment is meant to measure your backend abilities.

The Problem: We have two types of assets, which can be either videos or images. These assets have the following metadata: ID, type, filename and extension (jpg, png, mp4, etc). Additionally, each one of these assets can have multiple scores, one for each Score Type (Type1, Type2, Type3). This score can be a number between 0 and 100. To share these metrics with the rest of the organization, we need to be able to interact with this database and know the average score for each asset type (video, image) for a specific Score Type.

The Task:

We propose to develop a few endpoints that interact with a mock database containing these assets. You are free to implement this mock database in any way you want in your tests. The endpoints are as follows:

A. **An endpoint to create assets.** The endpoint should create a new row in an assets table.

B. **An endpoint to add scores to assets.** The endpoint receives an asset ID, and three values between 0 and 100 corresponding to our three Score Types (Type1, Type2, Type3). It then adds these three scores to the database, in the corresponding asset ID row.

C. **An endpoint to get the data for an asset.** The endpoint receives an asset id and returns the asset data, including all of the asset's scores.

D. **An endpoint to get the average score for a specific Score Type and a specific asset type.** The endpoint receives an asset type (image, video), and a score type (Type1, etc). The endpoint should return the average score for that asset type and score type.

Example of the database (feel free to implement a different version):

Assets table						
Asset ID	type	filename	extension	Score Type1	Score Type2	Score Type3
001	video	file1	mp4	34	55	12
002	image	file2	png	24	23	55
003	image	file3	png	53	28	34

Stack:

- The code should be written in Typescript
- You can use whatever database you want, including sqlite.
- It does not need to be deployed anywhere, but we should be able to run it locally.
- Ideally the endpoints should use GraphQL.

Additional comments: we would love to see **tests** for all your endpoints that we can easily run to check the functionality of your code.

Deadline: the task is due on Wednesday 23 rd at 23.59pm EST. During the week, we will be at your disposal in case you have any questions!

Questions: if you have questions, please email matias.feld@tekal.ai and gabriel.vaquer@tekal.ai with the email subject “Tech Test Questions - Back End Developer”.