Full-Stack take home task

Task Outline

We would like you to build an image generation playground consisting of

- A react/typescript frontend
- A backend service(s) for generating images

Backend

Instructions
Create an API for generating images of animals using AI using **Typescript**.

Deploy it using the platform of your choice.

The API has the following endpoints. You can use any image gen provider.

POST `/generations`

- Generates n images of a user-selected animal
- The endpoint **must not** perform the image generation synchronously
  - Store a job record in a DB (SQLIte / Postgres)
  - Trigger a worker process that
    - picks up pending jobs
    - calls the image provider
    - updates job status and saves a reference to the output images

- Request body:
  - numImages - the number of images to be generated

    - animal - the animal image to generate. This will be driven by a dropdown in the frontend

- Response body:
    - jobId
    - status: “pending” (other statuses - “completed”, “error”)

GET `/generations`

- Returns a list of jobs with their status and any outputs
- The outputs should contain the url of the imaged
- The jobs should be ordered by creation time
- The endpoint should support offset based pagination

Other notes

Include instructions to run the service locally

Add at least a few tests (unit or simple integration)

Handle error cases gracefully (invalid input, missing job, and ideally provider failures)

Design your code in a way that would be easy to extend (e.g. swapping AI providers)

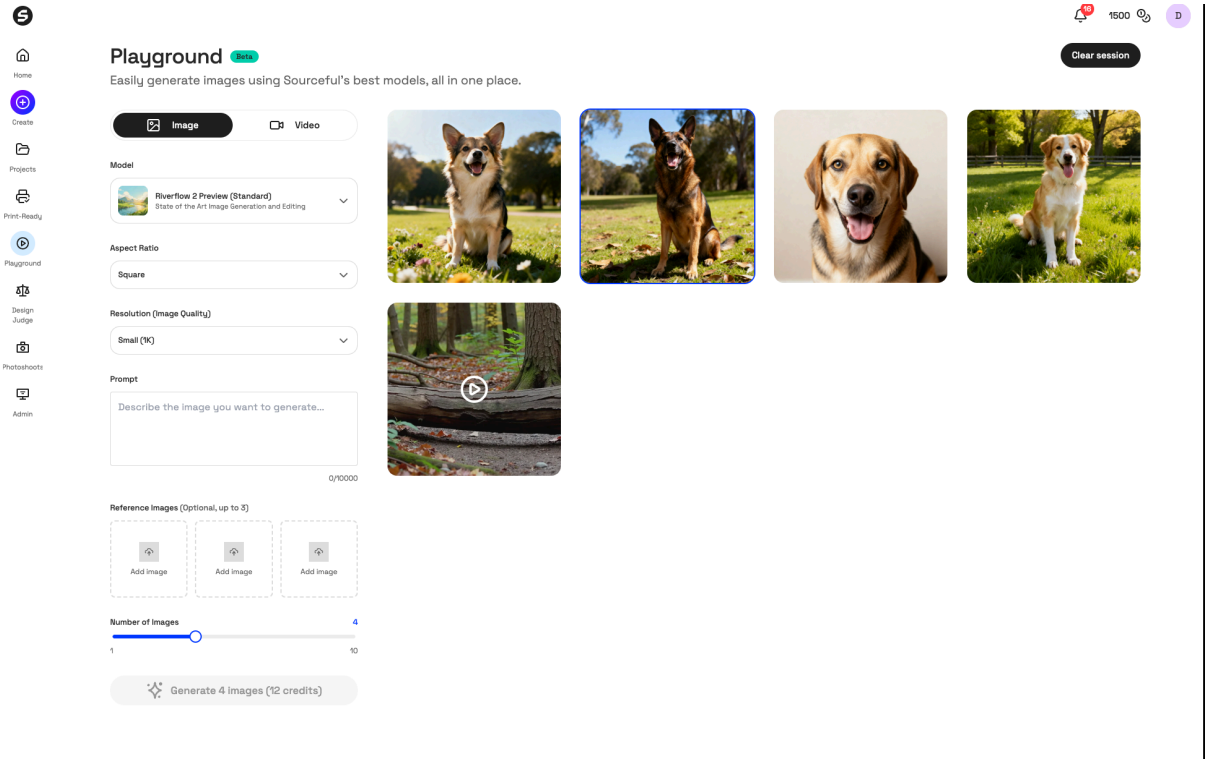
Frontend

Implement a frontend for triggering an image generation.

Tech stack:

- React
- Typescript

You can use https://www.sourceful.com/dashboard/playground for an idea of the layout.



A user should be able to select an animal, select the number of images and click ‘generate’.

When the image is enqueued a placeholder tile should be rendered. This should persist across page refreshes - you can assume all users see the same set of images.

All finished jobs should show in the image grid.
The image grid should support infinite scrolling.

Bonus points

- Style the frontend to match the Sourceful branding