Here's all my info on Toptal:

Hi all -

One year ago, I was invited to join Toptal.

https://www.toptal.com/

This is considered a much more exclusive and prestigious freelancing network than Upwork. They have strict (but still kind of arbitrary) selection criteria, and somehow I got lucky enough to get through the first several rounds of interviewing to the project phase.

Unfortunately, I had a full time job back then, so I didn't allocate enough time to the project and they refuse to consider it unless it is feature complete.

Now, after one-year freeze, I've been invited to reapply.

My plan: you will complete this project for me. I've included two of their sample projects below.

If I am able to become a member of Toptal based on this project, I will be able to get higher quality contracts than if I were just forwarding you work from Upwork. This will help bring in more work, improve your skills, and make us all money.

I am full-stack Ruby on Rails developer. I'm not sure if you guys are familiar with the framework, but I use React/Rails exclusively. I'm not the best React developer, but React/Rails is a very popular framework for startups, and if you all became proficient in it, we could work together more closely because our skills would line up. (I get offered Rails/React work all the time but don't have time)

Anyway - check out the projects below. I'll also include a half-completed version of the application that I did myself. I prefer deploying react apps to S3 static website and the back-end can go on Heroku, but it's whatever.

The new trend is to use GraphQL instead of REST API with Rails, that's what I use professionally now.

This is a good example of how to use graphql/rails/react: https://github.com/MatthieuSegret/graphql-rails-blog

Here was my attempt at the restaurant application

https://github.com/mdonagh/restaurant-react-rails

Let me know if you have any questions

Here are two examples of their evaluation assignments:

Write an application for the input of calories

User must be able to create an account and log in. (If a mobile application, this means that more users can use the app from the same phone).

When logged in, a user can see a list of his meals, also he should be able to add, edit and delete meals. (user enters calories manually, no auto calculations!)

Implement at least three roles with different permission levels: a regular user would only be able to CRUD on their owned records, a user manager would be able to CRUD users, and an admin would be able to CRUD all records and users.

Each entry has a date, time, text, and num of calories.

Filter by dates from-to, time from-to (e.g. how much calories have I had for lunch each day in the last month if lunch is between 12 and 15h).

User setting – Expected number of calories per day.

When meals are displayed, they go green if the total for that day is less than expected number of calories per day, otherwise they go red.

REST API. Make it possible to perform all user actions via the API, including authentication (If a mobile application and you don’t know how to create your own backend you can use Firebase.com or similar services to create the API).

In any case, you should be able to explain how a REST API works and demonstrate that by creating functional tests that use the REST Layer directly. Please be prepared to use REST clients like Postman, cURL, etc. for this purpose.

If it’s a web application, it must be a single-page application. All actions need to be done client side using AJAX, refreshing the page is not acceptable. (If a mobile application, disregard this).

Functional UI/UX design is needed. You are not required to create a unique design, however, do follow best practices to make the project as functional as possible.

New users need to verify their account by email. Users should not be able to log in until this verification is complete.

Additionally, provide an option for the user to log in using at least two social media providers (you can pick from Google, Facebook, Twitter, Github, or similar).

When a user fails to log in three times in a row, his or her account should be blocked automatically, and only admins and managers should be able to unblock it.

An admin should be able to invite someone to the application by typing an email address in an input field; the system should then send an invitation message automatically, prompting the user to complete the registration.

Users have to be able to upload and change their profile picture. If they log in using a social media pull the image from their account they used to log in.

If the number of calories is not provided, the API should connect to a Calories API provider (for example https://www.nutritionix.com) and try to get the number of calories for the entered meal.

Write unit and e2e tests.

Write an application to Review Restaurants

User must be able to create an account and log in. (If a mobile application, this means that more users can use the app from the same phone).

Implement 3 roles with different permission levels

\* Regular User: Can rate and leave a comment for a restaurant

\* Owner: Can create restaurants and reply comments about owned restaurants

\* Admin: Can edit/delete all users, restaurants, comments, and reviews

Reviews should have:

\* A 5 star based rate

\* Date of the visit

\* Comment

When a Regular User logs in he will see a Restaurant List ordered by Rate Average

When an Owner logs in he will see a Restaurant List only the ones owned by him, and the reviews pending to reply

Owners can reply the review once

Restaurants detailed view should have:

\* The overall average rating

\* The highest rated review

\* The lowest rated review

\* Last reviews with rate, comment, and reply

Restaurant List can be filtered by Rating

REST API. Make it possible to perform all user actions via the API, including authentication (If a mobile application and you don’t know how to create your own backend you can use Firebase.com or similar services to create the API).

In any case, you should be able to explain how a REST API works and demonstrate that by creating functional tests that use the REST Layer directly. Please be prepared to use REST clients like Postman, cURL, etc. for this purpose.

If it’s a web application, it must be a single-page application. All actions need to be done client side using AJAX, refreshing the page is not acceptable. (If a mobile application, disregard this).

Functional UI/UX design is needed. You are not required to create a unique design, however, do follow best practices to make the project as functional as possible.

Bonus: unit and e2e tests.

Please use this private repository to version-control your code: