

Back-End/Full-Stack Exam

Senior/Staff/Lead Engineers who can prove their current/previous base salary is over NT\$1.5 million per year can choose to opt out of taking the technical exam. Freelance and founder salaries do NOT count as a previous salary reference. We will make an offer based on resume, working experience, education, personality assessment, and job references. Failure to accurately report previous salary will result in automatic rejection of application. If you'd like to pursue the expedited option, please schedule the final round [here](#).

Introduction

1. All instructions are on here.
 - a. This exam is to test whether a candidate can read and follow written instructions.
 - b. Exams that fail to follow precisely submission instructions will not receive offers, because it reflects the candidate's inability to follow written instructions, which is extremely important in an all remote team such as ours.**
 - c. No further questions may be asked.
2. Candidates must follow submission instructions.
 - a. Candidates must submit a link to a live, working app that is accessible from the internet.
 - b. Candidates must submit a link to a live API documentation/testing page (e.g. Swagger)
3. Candidate's exam result.
 - a. Our senior engineers will review the candidate's code quality, and make a hiring decision. It will take 3-4 weeks for us to review.
 - b. Candidates will get the result regardless of the engineer's decision.
 - c. Candidates can email us about any questions or competing exploding offers at hr@avancevl.com.

What We Look For

This exam is designed to test if you can:

1. Follow simple, written, English instructions.
2. Build a simple sign up/sign in app that exists in all apps.
3. Find and use third party APIs to save time.
4. Write clean and well documented code.
5. Work with ambiguity in instructions to build an app that has a good user experience.

Exam Grade and Starting Salary

Each portion of the exam is worth a certain maximum number of points if features are functional. You do not have to attempt all features. We will test your demo app and add up the points. The

more points in the total points tally, the higher your starting salary. The number of points will directly determine your starting salary.

Submission Instructions

1. Please complete the exam within 3 weeks.
 - a. Due date is approximate.
 - b. You can take more time if you need to, or if you have a busy schedule and can only start the exam at a later date.
2. Please commit all your code to GitHub. Our engineers will read your code.
3. Please complete the following form to submit your exam and schedule final round of interview:

<https://2bs9m2ujxlo.typeform.com/to/EMuA9dnZ>

Requirements

Create a simple app where users can sign up and sign in from a landing page into a simple dashboard. The landing page can be blank with only two separate links to “**Sign Up**” and “**Sign In**”. The simple dashboard can only be accessed after the user signs up or signs in.

Back-End Frameworks and Language

You may use the following tech stacks:

1. Language
 - a. Preferred language:
 - i. Typescript, Node.js
 - b. Accepted language:
 - i. JavaScript
 - c. Not accepted languages
 - i. Ruby on Rails
 - ii. Java, C#, C++
 - iii. Golang
2. Database
 - a. Relational databases, SQL
 - b. Non-relational databases not accepted.

Suggested Third-Party APIs

- Auth0 for authentication
- SendGrid for email
- Retool for dashboard

Coding Styles

Please follow [Google Coding Style](#) or Airbnb Style for your chosen language. We will review your code to check your ability to conform to a written style guide.

API Design

You must design an API for all features you implement below. You must use Swagger, Postman, or a similar tool to (1) document and (2) allow us to test the API.

Hosting

You must host the live app either yourself or a free app hosting service like Heroku, Firebase, Vercel, etc. We will not accept exam submissions that are not live.

Sign Up [20 points]

The Sign Up page should allow users 3 options to create an account:

- (1) email and user defined password
- (2) Google OAuth
- (3) Facebook OAuth

You can use any third party tool, library, or API for this. In fact, to save time, it is highly recommended that you use a third party auth API. Create your own guest or trial accounts with third party tools. We only need a live demo app to test for 1 or 2 weeks.

User Defined Password [20 points]

The user defined password must be entered twice and match. In addition, the password must be validated by the following conditions.

- contains at least one lower character
- contains at least one upper character
- contains at least one digit character
- contains at least one special character
- contains at least 8 characters

Email Verification [40 points]

For the user defined password, after the validated password is entered, your back-end app must send an verification email to the email address provided. The email must contain a link, that if the user clicks the link, their email will be verified in the back-end and the user will be directed to a simple dashboard in a new browser.

Only accounts that have email verified can access the simple dashboard. Users that have not verified email will only see a button or link that says "Resend Email Verification", and clicking on this link will resend the same email verification.

Only accounts created via email and password must be verified with email verification. Facebook and Google OAuth sign up accounts do not need to send email verification, and can immediately access the simple dashboard.

Like with authentication, you can use any third party email sending tool, library, or API. Create your own guest or trial accounts. We only need a live demo app to test for 1 or 2 weeks.

Login [10 points]

We will allow users to login through the 3 methods that users can sign up with:

- (1) email and user defined password
- (2) Google OAuth
- (3) Facebook OAuth

You can use any third party tool, library, or API for this feature.

User Profile [20 points]

The user profile will display the user's email and name (from Google or Facebook OAuth). In addition, the user can reset their name. Everytime the user goes to user profile, the user should see the name they have chosen.

Reset Password [30 points]

In the simple dashboard, add the ability to reset password. The password must meet the same criterias as defined previously. In addition, the user must enter 3 text input boxes:

1. Old password
2. New password
3. Re-enter new password

Cookies and Logout [50 points]

Store cookies in the browser so that next time a logged in user returns to your site, the user will be automatically logged in. Add a logout feature in the user profile so that cookies can be cleared.

User Database Dashboard [50 points]

Create a dashboard with a list of all users that have signed up to your app. For each user, show the following information:

1. Timestamp of user sign up.
2. Number of times logged in.
3. Timestamp of the last user session. For users with cookies, session and login may be different, since the user may not need to log in to start a new session.

User Statistics [30 points]

At the top of the user database dashboard, show the following statistics:

1. Total number of users who have signed up.
2. Total number of users with active sessions today.
3. Average number of active session users in the last 7 days rolling.

Attention to Detail, Bug Finding [5 points per bug]

We look for candidates with extreme attention to detail. One test of that is your ability to find bugs, in both your own and other people's products. Please find at least 5 bugs in our live app:

app.earnaha.com

1. For each bug you find, report the bug using this Typeform:
<https://2bs9m2ujxlo.typeform.com/to/mQTiDV5n>

2. Record number of bugs found in the Exam Submission Form:
<https://2bs9m2ujxlo.typeform.com/to/EMuA9dnZ>