

ALLERGAN DATA LABS APPLICANT EXERCISE

OVERVIEW

Implement a Registration (and Login) page defined by the below specification.

EVALUATION

Your submission should:

- Demonstrate your mastery of application architecture, extensibility and performance.
- Satisfy the specification (below). If some behavior is unspecified, draw upon your experience to fill in the blanks.
- Be authored with JavaScript, Supersets of JavaScript such as TypeScript and Flow are OK. Please no Coffee Script.

We will run your submission in the latest version of Chrome.

3RD PARTY CODE

Use your discretion. We're trying to assess your architecture expertise, not the framework author's. Please be clear about what code is and isn't yours.

SUBMISSION

Feel free to submit your code via private/public GitHub, GitLab, or Zip archive. We are aware that interviewing is a private conversation, we respect your privacy and will accept a submission based on your comfort level

SPECIFICATION

FRONTEND

Feel free to use any frontend framework or starter pack e.g. Expo, Create React App, Next.js, codesandbox.io. It just needs to be ran locally.

REGISTRATION PAGE

Build a Registration Page with the following fields

- First Name: string
- Last Name: string
- Username: string
- Email Address: string
- Password: string

On submit: The form should send an XHR Request to the backend for processing.

On Success: Show a message that user was created successfully

On Error: Show the user why there was an error

LOGIN PAGE (BONUS)

Build a Login Page with the following fields

- Username
- Password

On Submit: the form should send an XHR Request to the backend for processing.

On Success: the form should store the appropriate information to know that the user is logged in. Use any storage medium (cookies, local storage, etc.) you would like, but please be able to talk through why it was chosen.

On Error: Show an error message

BACKEND ENDPOINTS

Base URL: Will be provided separately.

Create User:

POST: /user

In order to create a user, POST json in the body with all the required fields.

HTTP Status Codes

- `200` - success
- `400` - validation error
- `418` - username is already taken or unknown error

On Success, the body of the response will be in JSON. *On Error* the body of the response will be in plain text. Please plan accordingly.

```
const regObject = {
  username: "",
  password: "",
  firstName: "",
  lastName: "",
  mobilePhone: ""
}

fetch(`${baseUrl}/dev/user`, {
  method: "POST",
  cache: "no-cache",
  body: JSON.stringify(regObject)
})
```

Login:

GET: /user/{username}/{password}

HTTP Status Codes

- `200` - success
- `400` - No user found with that username and/or password

On Success, the body of the response will be in JSON. *On Error* the body of the response will be in plain text. Please plan accordingly.

Debugging URLs

These endpoints are provided for debugging purposes. Feel free to create and delete any amount of users, the database will be destroyed after the final code review.

List All Users

GET: /dev/user

Delete Users

DELETE: /user/{username}

For your convenience we have provided a design, but feel free to make it your own.

Allē



Create your new account

Enter your email and password

First Name

Last Name

Email Address

Password

CREATE ACCOUNT

Send promos and offers via SMS



BONUS FEATURES

- Create a Login Page
- Host the web application online
- Add Code Quality tools and scripts
 - Tests (Jest, Mocha, Chai, etc),
 - Linter (ESLint, TSLint),
 - Code-Formatters (Prettier)