

Back-End Dev Practical Test

Thank you for your interest in joining the HulkApps Team! Please **choose one** of the below small projects for your practical test. You are not obligated to do it entirely but it is a huge plus in your overall evaluation if you submit a complete project.

Some tips to help you stand out:

- Try to keep the file structure clean and easy to follow.
- If you are familiar with VueJs, AngularJs, ReactJs, Typescript, etc showcase it.
- Utilize a Design Framework or Component Library such as Bootstrap (4 or 5), Tailwind, Bulma, Vuetify, etc
- Share your notes on anything you want to point out.
- Make sure you use proper validation
- Make sure you follow proper guidelines and code standards
- Make sure to use all known features from the selected framework (Laravel, Express, Rails, etc..)
- Try to beautify the design pages by using the Component Library.
- Lastly, if the test feels too long and you don't have the time, it's ok. We can understand. However, finishing it completely, also shows us your commitment. Either way, we'll take whatever you can do.

Choose only **ONE** of the following for your test. All of the assets are under the Specs tab for your convenience. The Design does not contain a mobile view but should be presentable on a desktop view.

Option 1:PDF Upload Test

Languages / Frameworks Accepted:

PHP / Laravel

Test Details:

You need to create a module in which the following points must be covered.

- Project Detail:
 - <https://www.figma.com/file/O4lDnEkJUnuj8338PbNMWB34/React-JS-Test?node-id=0%3A1>
 - You need to create a page like above.
 - Your code should be able to upload pdf files
 - On the left side, all uploaded files should be listed.
 - Upon clicking on a file, pdf file data should be displayed on the right side.

Option 2:Appointment Management System

Languages / Frameworks Accepted:

PHP / Laravel / Ruby On Rails / Nodejs

Test Details:

You need to create a module in which the following points must be covered.

- User type: Admin, Doctor, registration page for patients
- Modules
 - Admin
 - Doctor
 - Patients
 - Appointment

Functionality:

- Patient
 - Login and check their appointment
 - Guest users can register the appointment using their email id
- Doctor
 - Login and check their appointment
 - Update appointment status based on their action
- Admin
 - Can manage Doctor, Patients and appointments
- Appointment
 - Admin or patient can create appointment
 - One patient or admin can create one appointment at a time(use a time slot one hour ex, 1:00 PM, 2:00 PM etc)

Expected Deliverables

- Share a link to your project repository (github). It must be public repository
 - Please use english while writing readme.md
- You can host the code or you can include deployment instructions in your Readme.md file, especially if you are using an advanced tech stack.
- The Readme file must have
 - Tech stack used, including specific libraries / versions.
 - Project setup guidelines
 - A quick paragraph with how you approached the project, what you liked, what you didn't like, and where you faced issues.
 - Estimated time to complete your test.
 - What is pending by your side?
- Make sure to include sample data (SQL file works) to test in an isolated environment.

We look forward to seeing your submission.

Thank you!!