4 hour fullstack test project

Description

In this project you are to build out a simple project using Laravel and the front end of your choice. This assessment has three components:

- Application Construction: You are given a set of requirements. Build an MVP.
- Client Communication: After the MVP, the client has plans for the next phase of the project. They have some questions for you.
- Documentation: Let's be sure the onboarding process to this project for future developers is seamless.

Application Construction

The system is a simple online shopping system which has Users, Products, and Orders. Our client "Johansen Cumberbund" is having you build a 'Phase 1' MVP.

Database Relationships

- Users have many Orders.
- Orders have one Product only.
- Products can be ordered many times.

Users

Users are either 'admins' or 'customers'. For Phase 1 only customers will be created.

User Attributes:

- ID
- Email
- Name
- Favorite Food
- (other attributes, if deemed appropriate)

Products

Products are things people buy. Customers indicate to the client that they wish to own one of the products by creating orders.

- ID
- Description
- Price
- (other attributes, if deemed appropriate)

Orders

Customers should be able to make orders given a list of products. They should see a history of their past orders. They cannot edit or delete an order after it has been created.

Order Attributes:

- ID
- Notes
- (other attributes, if deemed appropriate)

Database

Should be either a MySQL, or SQLite database.
All tables in the database must be created programatically.
The user table should be seeded with two (2) customers.

The products table should be seeded with two (2) products.

Client Communication

After the MVP is completed, the client expects the following during ‘Phase 2’ of the project. None of this needs to be built as part of this assessment. It is only a description of future plans.

- The creation of admin users, which can see lists of products and users.
- The ability for admins to create, edit, and delete products.

- The ability for admins to create, edit, and delete users.
- The ability for admins to see all orders, and the total amount each customer has spent.
- The ability for orders to have discounts. Customers are trusted to give themselves discounts.
- The ability for admins to edit the discount after the order has been created.
- The ability to cancel orders.

The client has the following questions for you:

1. What is the state of the MVP? Is it usable? Bug free? If not, how much more time do you need for Phase 1?
2. How do we maintain a full and accurate history of orders given changing products, discounts, and users?
3. A ‘shopping cart’, where users can view their selections before purchasing, was not included as part of the Phase 1 requirements. Do you feel it should be a part of the MVP, or did you include it? Why or why not? There is no ‘right’ answer.
4. Please deliver an estimate for Phase 2. How many hours would it take? On what date can you deliver?
5. In Phase 3, the client would like to allow customers to pay within the application. Do you have any recommendations for the client?

Please respond in a simple text file.

Documentation

Because FluxBucket has many projects, the ability to be brought up to speed on any particular project is important. If we handed the results of this assessment to an intern, would they be able to install the necessary packages and setup an environment to begin where you left off?

How do you install the necessary packages?

How do you create and seed database tables?

How does the first user log in?

How do you run the tests if tests are provided?

What is the purpose of the application?

Notes

Details

- You may install any packages you would like to use
- Tests are not a requirement, but encouraged
- Users should authenticate with passwords
- None of the following is necessary: email verification, email sending of any kind, ‘forgot password’ links, the ability to change a password.
- This app is invite-only. Registration should be disabled. During Phase 1 all users will be created with database seeders. A UI for the creation of users is not necessary. Storing passwords in your written documentation is permissible for this assessment.
- Phase 2 is not expected to be completed either now or in any future assessment. If you choose to work on Phase 2 during the 4 hours, do so in a separate branch. The ‘Client Communication’ and ‘Documentation’ sections are more important though.

Assessment

- Readability, use of methods / classes, code organization
- Detail orientation: Were all instructions followed?
- Client Communication and Documentation: How clear is your writing?

Completion

When you are done, please push your application up to either GitHub, GitLab, or, BitBucket.

Then notify Joshua via email (josh@fluxbucket.com) with a link to the repository. Please include any documentation which is not in the readme of the project. Please include the ‘Client Communication’ as a txt file (either attached to the email or included in the repository.)

Josh is available for any questions you may have regarding requirements, or anything else via email. Please feel free to reach out at any time.

Thanks, and good luck!
Josh