Excelitas Technologies

Drupal Development Basic Test

Notes:

1. Do this test on your own home, anytime, as long as this is complied to in a given time-frame or deadline.
2. Feel free to use google search to search for solutions. Feel free to copy codes from your searches. No copying of codes from other candidates; it would be easy to spot one.
3. It doesn't matter if you did not fully finished all tasks. Tasks 1, 2.1, and 5 are the only ones required. In case you are not familiar with git, please let us know.
4. The basis for hiring would not be the pass or fail, but the attitude in dealing these tasks.
5. For questions and clarifications, please contact us.
6. If you need more time, just let us know. Don't get intimidated.

Task 1: Setup.

1. Fork git@github.com:Excelitas-Technologies/etctest.git
   or https://github.com/Excelitas-Technologies/etctest.git

   Note: The repository is empty.

   Then, clone to your local machine, your forked repository.

2. On your local machine, create a branch test/<first-name>
   Example: test/john-mark

3. Install Drupal 8. Ensure that the username and password are:
   User: admin
   Pswd: drupaladm1n

4. Ensure that the Drupal instance on your local machine is running the latest version.
5. Commit your changes. You may do commits several times or one time, it depends on you. Best practices is optional but appreciated.

Task 2.1: Site building.

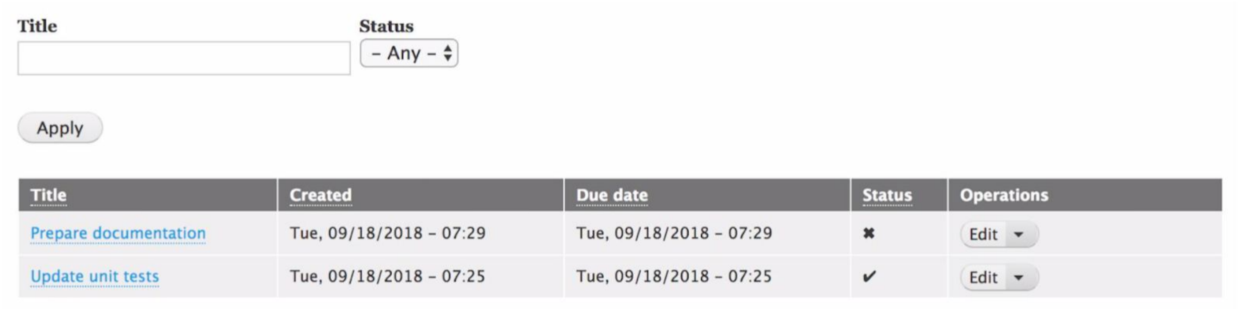
1. Create a new Content Type called *Task*.
2. Ensure that the *Task* CT has the following fields:
   a. Title: Text
   b. Due date: Date
   c. Status: Boolean

Task 2.2: Site building (optional, but highly encouraged).

1. Create a View page that displays all tasks. Of course this means, you have to add contents into *Task* CT. The view's format must be a table with the following columns:

    a. Title
    b. Created
    c. Due date
    d. Status
    e. Operations links
2. The view must have the following filters:
    a. Title: A test field for searching tasks by title.
    b. Status: A dropdown for filtering pending or completed tasks.
3. All the view's columns must be sortable, except for the operations links.
4. Export all configuration and commit your changes.

Please refer to the image below for reference. Please note that the theme or style doesn't matter for this task.



Title	Created	Due date	Status	Operations
Prepare documentation	Tue, 09/18/2018 - 07:29	Tue, 09/18/2018 - 07:29	✖	Edit
Update unit tests	Tue, 09/18/2018 - 07:25	Tue, 09/18/2018 - 07:25	✔	Edit

Task 3: Backend Development (optional, but encouraged).

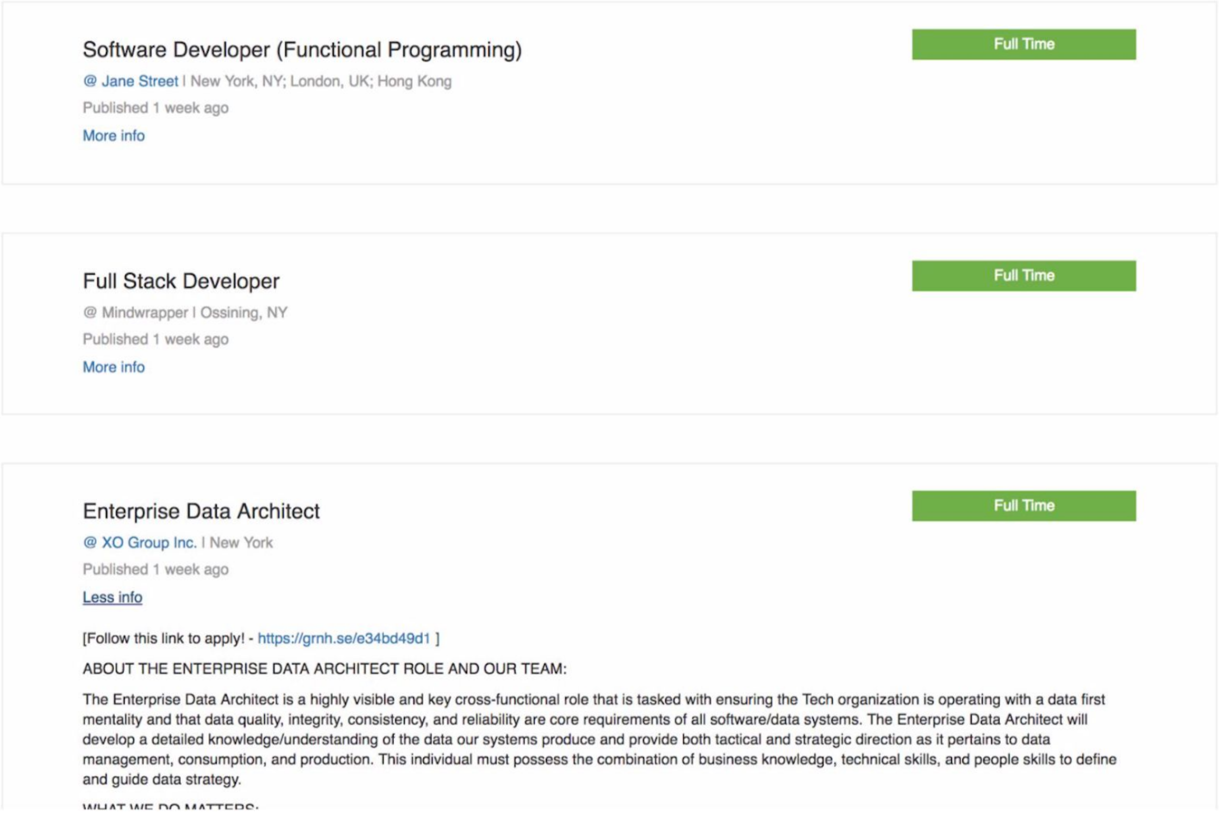
1. Create a custom module called *etc_jobs*.
2. Make a job listing page with the following details:
    a. Page path should be: */jobs*
    b. The job listing is coming from a public JSON file: https://jobs.github.com/positions.json
    c. Make sure the job listing contains the following info:
        i. Title
        ii. Company
        iii. Location
        iv. Type
        v. Created at
        vi. Description
3. Set the Job listing page as homepage.
4. Export your configuration and commit your changes.

Task 4: Frontend and theming (optional).

Note: Do this only if you finished Task 3.

1. Create a sub-theme based from *classy*. Name it to *etc_main*.

2. Style the job listing page similar to the image below. The style doesn't have to be perfect but it should be as close as possible.
3. The *description* should be hidden from the start, not via css. When the user clicks on "More info", the description should expand. And when the user clicks on "Less info", the description should collapse.
4. Commit your changes.



Software Developer (Functional Programming)

@ Jane Street | New York, NY; London, UK; Hong Kong

Published 1 week ago

More info

Full Stack Developer

@ Mindwrapper | Ossining, NY

Published 1 week ago

More info

Enterprise Data Architect

@ XO Group Inc. | New York

Published 1 week ago

Less info

[Follow this link to apply! - https://grnh.se/e34bd49d1]

ABOUT THE ENTERPRISE DATA ARCHITECT ROLE AND OUR TEAM:

The Enterprise Data Architect is a highly visible and key cross-functional role that is tasked with ensuring the Tech organization is operating with a data first mentality and that data quality, integrity, consistency, and reliability are core requirements of all software/data systems. The Enterprise Data Architect will develop a detailed knowledge/understanding of the data our systems produce and provide both tactical and strategic direction as it pertains to data management, consumption, and production. This individual must possess the combination of business knowledge, technical skills, and people skills to define and guide data strategy.

WHAT WE DO MATTERS:

Task 5: Push.

1. Make sure all changes in code, must be committed and all changes in the site administration shall be exported into code. Best practices is encouraged.
2. Create a Pull Request (PR) to the repository mentioned in Task 1.1.
3. Dump your database and send the compressed version to us. We will get back to you as soon as we have reviewed your test.