Boost up Your Certification Score

Acquia
Drupal-Site-Builder
Acquia Certified Drupal Site Builder Exam for Drupal 10, 11

Exams Boost
Boost Up Your Career

For More Information – Visit link below:

https://www.examsboost.com/

Product Version

✓ Up to Date products, reliable and verified.
✓ Questions and Answers in PDF Format.

Visit us at: https://www.examsboost.com/test/drupal-site-builder

Latest Version: 6.0

Question: 1

You have installed a contributed module called "Sample Module" that looks like it will be a great fit for the business case you are trying to solve. However, upon closer examination, it looks like the module only supplies a drush command; it does not have an admin interface.
As a site builder with no command-line experience, this will not work for you! You need a web user interface to use this module.
How should you request a web UI in the module's issue queue?

A. Create a "Bug Report" issue for the module with the subject line, "Create web UI for Sample Module."
B. Create a "Plan" issue for the module with the subject line, "Create web UI for Sample Module."
C. Create a "Feature Request" issue for the module with the subject line, "Create web UI for Sample Module."
D. Create a "Feature Request" issue for the module with the subject line, "URGENT: Module Broken!!! Needs UI."

Answer: C

Explanation:
Drupal's issue queue guidelines clearly distinguish between different issue types. A Bug Report is used when existing functionality is broken or not working as intended. In this case, the module is functioning correctly—it simply does not provide a UI—so option A is incorrect.
A Feature Request is the correct issue type when asking for new functionality or enhancements to an existing module. Requesting a web-based UI for a module that currently only provides Drush commands is a classic example of a feature enhancement, making option C correct.
Option B ("Plan") is typically used for outlining larger initiatives or coordinated development efforts, not for requesting a specific feature. Option D is incorrect because Drupal issue queue best practices emphasize clear, descriptive, and professional issue titles , avoiding urgency or misleading wording like "Module Broken!!!" when it is not actually broken.
Drupal.org documentation encourages contributors to categorize issues properly and write clear, concise summaries so maintainers and the community can effectively review and respond. Therefore, submitting a Feature Request with a clear subject line is the correct and recommended approach.

Question: 2

You have a website which is using a custom theme made specifically for that website. The client complains that content creators see an admin theme called "Claro" while creating or editing content. They want to see the content edit page in their own custom theme, but keep the other admin section in the Claro theme.
How can you achieve this?

A. On Appearance page, unselect the checkbox "See the administration theme when editing or creating content"
B. On the Appearance page set admin theme as default theme
C. Revoke the permission of viewing the custom theme for content creator role.
D. Give the Content Creator role permission to "View Admin Theme"

Answer: A

Explanation:
Drupal core provides a specific setting for whether node add/edit forms should use the administration theme. On the Appearance page, this is the checkbox labeled “Use the administration theme when editing or creating content” . When that setting is enabled, content creation and editing routes are treated as admin-theme routes; when it is disabled, those forms use the site’s default front-end theme instead, while the rest of the administration area can still use the admin theme such as Claro. This behavior is implemented in core through the node.settings configuration value use_admin_theme and the node route subscriber that marks node operation routes as admin routes only when that setting is turned on.
That makes option A correct. Option B would not solve the requirement, because setting the admin theme as the default theme would affect the whole site’s front-end presentation. Options C and D are incorrect because Drupal does not use a role permission like “View Admin Theme” to control this behavior for node edit pages. The documented control for this requirement is the Appearance-page checkbox tied to node.settings.use_admin_theme .

Question: 3

You have enabled website feedback contact form to allow users to submit feedback. You would like to redirect users to a different page after submission.
How will you accomplish this?

A. Use an add-on module like Webform, since Core doesn't offer this feature.
B. Set "Redirect URL" on Site information page under configuration.
C. Allow users to add their own Redirect URL in user profile pages.
D. Add the desired destination page to the "Redirect path" field in contact form settings.

Answer: D

Explanation:
Drupal core’s Contact module allows administrators to configure multiple contact forms, each with its own settings. One of the configurable options is the ability to define a redirect path after form submission. This is done within the specific contact form’s configuration, where you can specify the destination page users should be taken to after submitting the form.
Option D is correct because Drupal provides a “Redirect path” setting directly in the contact form configuration , making it possible to send users to a custom page such as a thank-you or confirmation page after submission.
Option A is incorrect because Drupal core already supports this functionality without needing a contributed module like Webform. Option B is incorrect because Site information settings do not control

form submission redirects. Option C is not relevant, as redirect behavior is not controlled through user profiles.
Thus, the correct Drupal-native solution is to configure the redirect path within the contact form settings, making D the correct answer.

Question: 4

You are building a recipe site in which users submit their favorite recipes as nodes, tagged with common ingredients from a carefully curated taxonomy vocabulary. You've been asked to create a page on which site visitors can select the ingredients they have on hand and view all the site's recipes that use those ingredients. You've already created a view listing all Recipe nodes.
How can you modify the view to support filtering by ingredient?

A. For each ingredient, create a separate view display with a contextual filter to limit results by that ingredient.
B. Include a "Combine fields" filter that searches all fields for the ingredient name(s) the user enters.
C. Create a filter using the Recipe content type's "Ingredients" field. Expose the filter to users, making sure it allows multiple selections.
D. Add a menu block listing all ingredients. Each item in the menu block should link to a subdirectory URL associated with the listed ingredient.

Answer: C

Explanation:
Drupal's recommended way to let visitors narrow a Views listing by a field value is to add that field as a filter criterion and then expose the filter to site visitors . The core Views documentation explains that filters can be exposed so users can interact with the data shown in the view, instead of hard-coding separate displays or relying on free-text searching.
This question specifically says the recipes are tagged with a curated taxonomy vocabulary of ingredients. Drupal's User Guide uses Ingredients as an example of a field that should be a taxonomy term reference , because the allowed values come from a managed list that can grow over time. That means the correct filter is the Recipe content type's Ingredients taxonomy-reference field in the View. Since visitors should be able to choose the ingredients they have on hand, the exposed filter should allow multiple selections.
The other options are not the standard Drupal site-building solution. Separate displays per ingredient do not scale, a combined-text search is not a structured taxonomy filter, and a menu block is navigation rather than a proper Views filter.

Question: 5

You need to add a "Star performer" block into your company's website. The block should show the name and picture of a selected star performer employee and should be editable in the normal block layout interface. All the employees are users of the website.
How do you implement the block?

A. Install the user_blocks module from Drupal.org, select the fields you need and place block via block layout.

B. Create a new Users View with user name field and user picture and an exposed filter on uid. Add a block display to the view and place the block in the block layout.
C. Add a custom block and select the type "user account information", activate the user name and picture fields.
D. Create a user view mode with user name and picture. Add a user reference field to one of your block types and let the user reference field use the new user view mode.

Answer: D

Explanation:
Drupal 10 and Drupal 11 recommend using entity references and view modes to display structured entity data (like users) within other entities such as blocks. The requirement is to display a selected user (employee) with their name and picture and allow editors to choose which user appears in the block through the block editing UI.
Option D follows Drupal best practices. First, you create a custom view mode for users (e.g., “Star Performer”) and configure it to display only the necessary fields such as user name and profile picture. Then, you create a custom block type with a user reference field , allowing editors to select a specific user. The display of that referenced user can be configured to use the custom view mode, ensuring consistent formatting.
Option B is less suitable because it relies on Views filtering rather than allowing editors to directly select a specific user. Option A depends on a contributed module unnecessarily. Option C is incorrect because Drupal does not provide a default block type for selecting arbitrary user entities in that way.
Thus, using a user reference field with a dedicated view mode is the correct and scalable Drupal solution.

Question: 6

The UX team has suggested that “Social share” buttons currently visible in the sidebar region should be moved to the footer. The Social share buttons are implemented in a custom block.
How should you make the requested changes?

A. Using the "Basic page" content type's "Manage display" interface, drag the existing block from the Sidebar region to the Footer region.
B. From the block layout admin page, drag the existing block from the Sidebar region to the Footer region.
C. From the "Appearance" admin page, reconfigure the site's theme to place the block in the Footer region instead of the Sidebar.
D. For each content type, use the Layout Builder module's "manage layout" feature to reassign the block from the Sidebar region to the Footer region.

Answer: B

Explanation:
In Drupal 10 and Drupal 11, blocks are placed and managed through the Block Layout system available at Structure → Block layout . Each theme defines regions such as Sidebar, Footer, Header, etc., and administrators can assign blocks to these regions.

Since the “Social share” buttons are implemented as a custom block , the correct approach is to reposition the block using the Block Layout admin interface . This interface allows site builders to move blocks between regions either by dragging or by editing the block’s region setting. This matches Drupal’s documented site-building practice for managing block placement.
Option A is incorrect because Manage display is used for configuring how fields are displayed on entities, not for block placement. Option C is incorrect because the Appearance page controls themes, not individual block placement. Option D is unnecessary because Layout Builder is used for per-content or per-entity layouts, not for globally placed blocks like a social share block.
Therefore, the simplest and correct solution is to move the block via the Block Layout page, making option B correct.

Question: 7

You have created a custom block “Festival Offer” and placed this block in the header section of your site. Your site has a “Premium customer” user role. You need to show this block only on the front page and only to users who have a “Premium customer” user role.
Which two visibility settings you need to make to the “Festival Offer” block to achieve these requirements?

A. Select the user role "Premium customer" from the Roles tab.
B. Add < front > path in the Pages tab and select "Show for the listed pages".
C. Remove < front > path in the Pages tab & select "Hide for the listed pages".
D. Select the user role "Authenticated user" from the Roles tab.

Answer: A, B

Explanation:
Drupal 10 and Drupal 11 let you control block display through visibility conditions on the block configuration form. Drupal’s block management documentation explains that a block’s visibility can depend on page-specific visibility settings and also on user role conditions. For page visibility, Drupal specifically notes that the front page is an exception and must be entered as < front > . To show a block only on the homepage, you add < front > in the Pages visibility settings and choose “Show for the listed pages.”
To limit the block to a specific audience, you use the Roles visibility condition and select the exact role that should see the block. In this case, that is Premium customer , not the broader Authenticated user role, because selecting Authenticated user would show the block to all logged-in users who have that role, not just premium customers. Drupal core’s block form API also confirms that block visibility is built through configurable visibility conditions in the block UI.
So the correct combination is:
A for the role restriction, and B for the front-page-only restriction.

Question: 8

The site you are managing has a "Comments approval" process. You have moderators with appropriate permissions to review and approve comments. The moderators want to see all the unapproved

comments sorted by the node on which they are posted, but they can't do this on the Comments approval page.
What is the simplest way to solve this problem?

A. Have the content editor review comments by visiting the node instead of using the Comments page.
B. Edit the Comments View, and update the table settings to make the Entity ID field sortable.
C. Create a View of unapproved comments and set the sort order to "Entity ID."
D. Create an Export view, so the content editor can review comments in a spreadsheet.

Answer: B

Explanation:
In Drupal 10 and Drupal 11, the administrative Comments listing is provided by the core comment View. In core's default views.view.comment configuration, the comments administration display uses the table style, includes the entity_id field labeled "Posted in" , and that column is present in the table configuration. However, core sets that column's table option to sortable: false , while other columns such as Subject, Author, and Updated are sortable. That is why moderators cannot sort the approval page by the content item the comment belongs to, even though the field is already there.
Because the existing approval page already uses a View, the simplest solution is to edit that View and enable sorting for the Entity ID / Posted in column in the table settings. Creating a brand-new View would work, but it is not the simplest option because it duplicates an existing administrative display. The other options do not address the actual issue with the current moderation screen. Drupal's content administration documentation also confirms that comments are managed from the Comments administration page, making improvement of that interface the most appropriate site-building solution.

Question: 9

Recent log entries of your site show that there are too many failed login attempts for the site super admin user (user 1).
How can you better secure the Drupal super admin user (user 1) from outside attacks?

A. Only allow members of the Drupal community in good standing to be user 1.
B. Cancel the user account for user 1.
C. Remove the "Administrator" role from user 1, which makes them an anonymous user.
D. Do not give this user an obvious name like "admin" or "administrator" which are too easy to guess.

Answer: D

Explanation:
Drupal's official security guidance for the super user account (user 1) specifically recommends that administrators do not name the first account "admin" or something obvious . The reason is straightforward: obvious usernames make brute-force and credential-guessing attacks easier, because attackers already know half of the login credentials they need to target. Drupal's administration and security documentation also recommends limiting direct use of user 1 and protecting it carefully because it is a uniquely powerful account.

That makes option D the correct answer. The other options do not match Drupal's documented best practices. There is no Drupal security rule about "members of the Drupal community in good standing" being allowed to use user 1. Canceling the user 1 account is not the recommended solution for this situation, and removing the Administrator role from user 1 would not make the account anonymous; in Drupal, user 1 is a special account with superuser behavior that is handled separately from ordinary role assignment. Drupal's own guidance emphasizes safer naming and tighter handling of the account, not these alternative actions.

Question: 10

A page has been added for a new product, and the marketing team wants you to add it to the main navigation menu, as a child of the "Products" page. During a promotion period, the team also wants you to add a link to the new page as a child of the "What's New" page.
What is the best way to add both links to the main navigation menu?

A. Edit the new page, add a link in Menu Settings, and select "Products" as the Parent item. In the Main navigation menu, add a new link to the page and select "What's New" as the parent item. When the promotion is over, delete the second link from the menu.
B. Edit the new page, add a link in Menu Settings, and select "Products" as the Parent item. Clone the new page, and set the menu link parent to the "What's New" page. When the promotion is over, delete the cloned page.
C. In the Main navigation menu, add two links to the new page, one with "Products" as the parent page and the other with "What's New" as the parent page. Set the second link to expire on the date the promotion ends.
D. Edit the new page and create two links in the Menu Settings section, one with "Products" as the parent page and the other with "What's New" as the parent page. When the promotion is over, delete the second menu link.

Answer: A

Explanation:
In Drupal 10 and Drupal 11, each content item (node) can only have one menu link created through its Menu settings . Drupal documentation explains that the Menu settings section allows you to place a node in a menu, but only as a single menu item . Therefore, you cannot create two menu links for the same node directly from the node edit form, which makes option D incorrect.
The correct approach is to create one menu link via the node's Menu settings (placing it under "Products"), and then manually add an additional menu link in the Menu administration (Structure → Menus → Main navigation) that points to the same node but is placed under "What's New." This matches option A . After the promotion, the temporary menu link can be removed.
Option B is incorrect because duplicating content creates unnecessary redundancy and breaks content management best practices. Option C is incorrect because Drupal core does not provide a built-in "menu link expiration" feature.
Thus, Drupal's recommended method is to create one menu link via the node and another manually via the menu UI, making A the correct answer.

Thank You for Trying Our Product

For More Information – Visit link below:

https://www.examsboost.com/

15 USD Discount Coupon Code:

G74JA8UF

FEATURES

- ✓ 90 Days Free Updates
- ✓ Money Back Pass Guarantee
- ✓ Instant Download or Email Attachment
- ✓ 24/7 Live Chat Support
- ✓ PDF file could be used at any Platform
- ✓ 50,000 Happy Customer