

## **Resources MODULE – TEST CASE REPORT**

Sr. No.	Test Case ID	Test Name	Type	Purpose
1	RES-M-001	test_str_returns_title	Unit (Model)	Verify that __str__ of Resources model returns the resource title.
2	RES-M-002	test_default_ordering_featured_then_rating_then_created_at	Unit (Model)	Ensure default model ordering shows featured items first, then by rating and recency.
3	RES-M-003	test_defaults_for_views_and_rating	Unit (Model)	Confirm that views defaults to 0 and rating to 0.0 when not provided.
4	RES-V-001	test_resource_list_status_and_template	Integration (View)	Check that /resources/ loads successfully and uses resources/resources.html with correct context.
5	RES-V-002	test_search_filters_by_title_description_category	Integration (View)	Validate that the search parameter filters resources by title/description/category.
6	RES-V-003	test_filter_by_type	Integration (View)	Verify filtering by type query parameter.
7	RES-V-004	test_filter_by_category	Integration (View)	Verify filtering by category query parameter.
8	RES-V-005	test_filter_by_difficulty	Integration (View)	Ensure difficulty filter works and that any is treated as “no filter”.
9	RES-V-006	test_sort_by_popular_orders_by_views_then_rating	Integration (View)	Ensure sort=popular orders resources by views (and then rating).

10	RES-V-007	test_sort_by_title_orders_alphabetically	Integration (View)	Ensure sort=title returns resources in alphabetical order of title.
11	RES-V-008	test_pagination_limits_items_per_page	Integration (View)	Verify that pagination is applied with max 9 resources per page and total count is correct.

## Resources Module – Testing Summary

- **Total tests:** 11
- **Type split:** 3 unit tests (model), 8 integration tests (view).
- **Coverage focus:**
  - Model behaviour: defaults, ordering, `__str__`.
  - View behaviour: search keyword, filters (type/category/difficulty), sorting (recommended, popular, title), pagination, and template rendering.
- **Result:** After running `python manage.py test resources`, all **11 tests should pass**, showing that **resource discovery and filtering logic is working as intended**.

The screenshot shows a VS Code interface with a terminal window open. The terminal output shows the command `(.venv) C:\Users\Himani\OneDrive\Desktop\Mind_Ease_V>python manage.py test resources` being executed. The output indicates that 11 tests were found and passed successfully, with a system check identifying no issues. The terminal also shows the command `python manage.py test resources` being run again, resulting in the same successful outcome. The VS Code Explorer on the left shows the project structure, including the `resources` module and its tests.

```
(.venv) C:\Users\Himani\OneDrive\Desktop\Mind_Ease_V>python manage.py test resources
Found 11 test(s).
Creating test database for alias 'default'...
C:\Users\Himani\OneDrive\Desktop\Mind_Ease_V\.venv\Lib\site-packages\razorpay\client.py:4: UserWarning: pkg_resources is deprecated as an API.
See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for removal as early as 2025-11-30. Refrain fr
om using this package or pin to Setuptools<81.
  import pkg_resources
System check identified no issues (0 silenced).
.....
Ran 11 tests in 0.305s

OK
Destroying test database for alias 'default'...

(.venv) C:\Users\Himani\OneDrive\Desktop\Mind_Ease_V>
```