QA Testing Document — Board Module (Multi-Project)

Product:

Bharathi's Canvas – Project & Task Board

Scope:

Covers testing of the Board Module (Projects, Columns, Tasks/Stories, and Feature Integration) across multiple projects and users.

Version:

v1.0 | Date: October 2025 | Prepared by: *Sr. QA Engineer*

🔍 1. Sanity Testing — Multi-User Validation

Test ID	Scenario	Steps	Expected Result	Network Verification	DB Verification
SAN-01	Verify User A login and Board access	1. Login with User A credentials. 2. Navigate to Board.	Dashboard should show User A's projects only.	`GET /api/v1/projects` returns list scoped to User A's `user_id`.	`projects` table → `user_id = User A UUID`.
SAN-02	Verify User B login and isolation	1. Logout User A. 2. Login as User B. 3. Check Board view.	User B sees only their projects. No shared data from User A.	`GET /api/v1/projects` response count differs from User A.	Confirm DB: `project.user_id = User B` rows only.
SAN-03	Verify user session token	Refresh page or open new tab after login.	User remains logged in until JWT expires.	`Authorization` header present on all requests.	n/a

🧩 2. Project Management

Test ID	Scenario	Steps	Expected Result	Network Verification	DB Verification
PROJ-01	Create new Project	1. Click “Add Project”. 2. Enter name “Kanban Alpha”. 3. Save.	New project appears in sidebar.	`POST /api/v1/projects` → returns `201` with correct payload.	`projects` table contains new row with matching `name` and `user_id`.
PROJ-02	Delete existing Project	1. Hover project. 2. Click 🗑 delete. 3. Confirm modal.	Project removed from sidebar.	`DELETE /api/v1/projects/{id}` → returns `204`.	DB row removed.
PROJ-03	Verify multiple projects visible	1. Create two more projects. 2. Reload board.	Sidebar lists all projects under logged-in user.	`GET /api/v1/projects` returns list of 3 items.	DB count = 3 for same `user_id`.

🧱 3. Column Operations (To Do, In Progress, Validation, Done)

Test ID	Scenario	Steps	Expected Result	Network Verification	DB Verification
COL-01	Add new column	Click “Add Column”. Name it `QA Review`.	New column appears instantly.	`POST /api/v1/columns` → `201 Created`.	`columns` table → new entry with `project_id`.
COL-02	Delete column	Click 🗑 on “Validation”. Confirm modal.	Column disappears.	`DELETE /api/v1/columns/{id}` → `204`.	Row removed from `columns` table.
COL-03	Drag-drop task between columns	Move a card from “To Do” → “In Progress”.	Task card moves successfully.	`PATCH /api/v1/tasks/{id}` → payload `{"status":"in_progress"}`.	`tasks.status` updated.

🧩 4. Task (User Story) Management

Test ID	Scenario	Steps	Expected Result	Network Verification	DB Verification
TASK-01	Create new Story	Click “Create Story”. Fill all required fields.	Task card appears under “To Do”.	`POST /api/v1/tasks` → 201 response with correct payload.	Verify task in DB with `project_id` and `user_id`.
TASK-02	Edit Story details	Click existing card → “Edit Task”. Change title or assignee.	Updated values visible on card.	`PATCH /api/v1/tasks/{id}` returns updated object.	DB reflects updated columns (`title`, `assignee`).
TASK-03	Link Story to Feature	In TaskForm, choose a Feature and save.	Task now visible under that Feature’s story list.	`PATCH /api/v1/tasks/{id}` includes `"feature_id":"<uuid>"`.	`tasks.feature_id` column populated.
TASK-04	Delete Story	Click delete on task card.	Story disappears from board.	`DELETE /api/v1/tasks/{id}` → `204`.	Row removed from `tasks` table.
TASK-05	Validation of empty fields	Leave “Title” blank and submit.	Validation message “Title is required”.	Request blocked — no network call made.	No DB insert.
TASK-06	Verify story status badges	Create tasks in all columns.	Cards display colored badges for each status.	n/a	DB status values: `to_do`, `in_progress`, `validation`, `done`.

5. Feature ↔ Story Integration

Test ID	Scenario	Steps	Expected Result	Network Verification	DB Verification
FEA-01	Create new Feature	Go to Features tab → “New Feature”. Fill name, story, notes.	Feature row visible in Feature Dashboard.	`POST /api/v1/features` → 201 response.	`features` table updated.
FEA-02	Attach Story to Feature	Create Task with feature selected.	Story appears inside the	`PATCH /api/v1/tasks/{id}` with correct `feature_id`.	DB shows `tasks.feature_id`

Test ID	Scenario	Steps	Expected Result	Network Verification	DB Verification
			feature accordion.		matches feature ID.
FEA-03	Fetch all stories under Feature	Expand FeatureRow → “User Stories”.	Stories displayed with correct codes and status badges.	`GET /api/v1/features/feature/{id}` → list of stories.	DB join query returns same count.
FEA-04	Delete Feature	Click 🗑 on feature. Confirm modal.	Feature and its association removed.	`DELETE /api/v1/features/{id}` → 204.	Verify no orphan `tasks.feature_id` references.

🌐 6. API Integration & Network Validation

Test ID	Scenario	Steps	Expected Result
API-01	Verify all requests include Authorization	Open DevTools → Network tab → Inspect API requests.	Every API call includes `Authorization: Bearer <token>`.
API-02	Validate payload fields	Inspect `POST /tasks` request body.	Fields: `title`, `description`, `assignee`, `project_id`, `feature_id` (if linked).
API-03	Verify correct HTTP responses	Create, update, and delete entities.	201 → create, 200 → update, 204 → delete.
API-04	Verify feature fetch integration	Inspect `GET /features/feature/<id>`.	Returns list of tasks with correct mapping.

🗄️ 7. Database Integration Tests (PostgreSQL)

Test ID	Scenario	SQL Query	Expected Result
DB-01	Verify Task creation	`SELECT * FROM tasks WHERE title='story 1';`	Row exists with proper `project_id`, `user_id`, and `feature_id`.
DB-02	Verify Project count per user	`SELECT COUNT(*) FROM projects WHERE user_id='<uuid>';`	Matches number of projects in UI.
DB-0	Verify Feature	`SELECT * FROM tasks WHERE`	Returns rows that appear under

Test ID	Scenario	SQL Query	Expected Result
3	linkage	`feature_id IS NOT NULL;`	Features UI.
DB-04	Verify deletion cascade	Delete project → check tasks.	Tasks for deleted project should be gone.

8. UI & Responsive Testing

Test ID	Scenario	Steps	Expected Result
UI-01	Check responsiveness	Resize browser (mobile → desktop).	Layout adjusts, modals centered, cards stacked properly.
UI-02	Verify modal accessibility	Open TaskForm and FeatureForm.	`Tab` navigation works, focus stays inside modal.
UI-03	Verify dark/light mode (if available)	Toggle browser theme.	Text contrast and card shadows remain readable.

⚠ 9. Negative / Edge Testing

Test ID	Scenario	Steps	Expected Result
NEG-01	Invalid token	Clear localStorage token and refresh.	Redirect to login page or `401 Unauthorized`.
NEG-02	Feature delete with linked stories	Delete a feature having tasks.	System either restricts or gracefully unlinks tasks.
NEG-03	Duplicate project name	Try to add same project twice.	Error or validation shown.

10. Regression Checklist

- Project CRUD tested across users.
- Board columns drag/drop persisted.
- Feature and Task association consistent.
- API responses verified with 2 active users.
- DB reflects same structure as UI.
- No orphan records or 500 server errors.
- UI responsive and modals accessible.