

STEP 29: Dashboard Widgets & Homepage Analytics - Testing Guide

Implementation Date: November 30, 2025

Status:  Complete

Overview

This guide provides comprehensive testing instructions for the Dashboard Widgets & Homepage Analytics Overhaul (STEP 29), which includes 14 real-time widgets displaying RFP pipeline status, SLA risks, timeline events, supplier activity, questions, submissions, readiness, comparisons, pricing, coverage, velocity, AI insights, exports, and search.

Prerequisites

1. Database Setup:

```
bash
cd /home/ubuntu/fyndr/nextjs_space
npx prisma generate
```

2. Start Development Server:

```
bash
npm run dev
```

3. Test Data:

Ensure you have:

- At least 5 RFPs in various stages
- Supplier contacts and responses
- Supplier questions (some answered, some pending)
- Activity logs
- Timeline dates set on RFPs

Test Scenarios

Test 1: Dashboard Access & Authentication

Objective: Verify dashboard is accessible only to authenticated buyers.

Steps:

1. Navigate to `http://localhost:3000/dashboard` while logged out
2. Verify redirect to `/login`
3. Log in as a buyer user
4. Verify redirect to dashboard
5. Verify navigation shows “Dashboard” as first item with LayoutDashboard icon

Expected Results:

- Unauthenticated users redirected to login
 - Authenticated buyers see dashboard
 - Dashboard icon visible in sidebar navigation
 - Page loads without errors
-

Test 2: Pipeline Widget **Objective:** Test RFP pipeline overview widget.**Steps:**

1. Navigate to dashboard
2. Locate “Pipeline Overview” widget
3. Verify total RFP count matches database
4. Verify stage breakdown shows correct counts
5. Check formatting of stage labels

Expected Results:

- Widget displays total RFP count
- Stage counts are accurate
- Stage labels are properly formatted (e.g., “Pricing Legal Review” instead of “PRICING_LEGAL REVIEW”)
- Empty state shows “No RFPs in pipeline” when no RFPs exist

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/pipeline \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 3: SLA Widget **Objective:** Test SLA status visualization.**Steps:**

1. Locate “SLA Status” widget
2. Verify red/yellow/green counts
3. Check that colors match status:
 - Red = Breached (negative days)
 - Yellow = Warning (0-3 days)
 - Green = On Track (>3 days)
4. Verify total count

Expected Results:

- Widget displays accurate SLA breakdown
- Colors are correctly applied
- Total matches sum of red/yellow/green
- Empty state when no RFPs exist

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/sla \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 4: Timeline Widget

Objective: Test upcoming events in next 7 days.

Steps:

1. Locate “Upcoming Events” widget
2. Verify events are sorted chronologically
3. Click on an event link
4. Verify redirect to correct RFP detail page
5. Check date formatting (e.g., “Today”, “Tomorrow”, “In 3 days”)

Expected Results:

-  Shows events within next 7 days
-  Events are sorted by date (earliest first)
-  RFP titles are truncated if too long
-  Date labels are user-friendly
-  Links navigate correctly
-  Empty state when no upcoming events

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/timeline \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 5: Activity Widget

Objective: Test recent activity log display.

Steps:

1. Locate “Recent Activity” widget
2. Verify last 10 activities are shown
3. Check relative time formatting (e.g., “2h ago”, “3d ago”)
4. Click on RFP link in activity
5. Verify navigation works

Expected Results:

-  Shows up to 10 recent activities
-  Activities sorted by most recent first
-  RFP titles link to detail pages
-  Relative time updates are accurate
-  Empty state when no activity

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/activity \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 6: Questions Widget

Objective: Test supplier questions summary.

Steps:

1. Locate “Supplier Questions” widget
2. Verify unanswered count is highlighted (amber color)
3. Check total and answered counts
4. Verify counts match database

Expected Results:

-  Unanswered count is prominent and amber-colored
-  Total questions count is accurate
-  Answered count is shown in green
-  Math checks out: answered + unanswered = total

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/questions \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 7: Submissions Widget

Objective: Test submission deadlines tracker.

Steps:

1. Locate “Submission Deadlines” widget
2. Verify deadlines are sorted by date (soonest first)
3. Check color coding:
 - Red: Overdue or today
 - Amber: 1-3 days
 - Gray: >3 days
4. Click on a deadline
5. Verify navigation to RFP detail page

Expected Results:

-  Shows up to 5 upcoming deadlines
-  Sorted chronologically
-  Color coding matches urgency
-  “Days until” labels are accurate
-  Links work correctly
-  Empty state when no deadlines

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/submissions \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 8: Readiness Widget

Objective: Test supplier readiness distribution.

Steps:

1. Locate “Supplier Readiness” widget
2. Verify distribution shows:
 - Ready (green)
 - Conditional (amber)
 - Not Ready (red)
 - Unknown (gray, if applicable)
3. Check total responses count
4. Verify counts sum to total

Expected Results:

-  Color-coded distribution is accurate
-  Total matches sum of all statuses
-  Unknown status only shows if present
-  Empty state when no responses

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/readiness \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 9: Comparison Widget

Objective: Test top suppliers display.

Steps:

1. Locate “Top Suppliers” widget
2. Verify shows top supplier for each RFP (up to 5)
3. Check comparison score badges
4. Click on a supplier entry
5. Verify navigation to comparison page

Expected Results:

-  Shows up to 5 top suppliers
-  Scores are displayed in green badges
-  RFP titles are truncated if needed
-  Links navigate to `/dashboard/rfps/[id]/compare`
-  Empty state when no comparisons run

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/comparison \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 10: Pricing Widget

Objective: Test pricing trends visualization.

Steps:

1. Locate “Pricing Trends” widget
2. Verify average total cost is calculated correctly
3. Check pricing list shows recent submissions
4. Verify currency formatting (e.g., “\$50,000”)

Expected Results:

-  Average cost is accurate
-  Shows up to 5 recent pricing entries
-  Currency formatted correctly
-  RFP titles are truncated
-  Empty state when no pricing data

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/pricing \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 11: Coverage Widget

Objective: Test requirements coverage heatmap.

Steps:

1. Locate “Requirements Coverage” widget
2. Verify coverage bars are color-coded:
 - Green: $\geq 80\%$
 - Amber: 60-79%
 - Red: <60%
3. Check percentage labels
4. Verify progress bars match percentages

Expected Results:

-  Shows up to 5 coverage entries
-  Progress bars accurately reflect percentages
-  Color coding is correct
-  Supplier names and percentages display
-  Empty state when no coverage data

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/coverage \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 12: Velocity Widget

Objective: Test response velocity metrics.

Steps:

1. Locate “Response Velocity” widget
2. Verify average days to submit is displayed
3. Check sample count
4. Verify calculation is correct (invitation to submission time)

Expected Results:

-  Average days displayed as integer
-  Sample count is accurate
-  Calculation matches database
-  Empty state when no submissions

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/velocity \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 13: AI Summary Widget

Objective: Test AI-generated executive summary.

Steps:

1. Locate “AI Executive Summary” widget (full-width at top)
2. Verify summary text is displayed
3. Check metrics cards:
 - Active RFPs (indigo)
 - Pending Questions (amber)
 - Deadlines This Week (red)
4. If OpenAI configured, verify AI-generated summary
5. If no OpenAI, verify fallback summary

Expected Results:

-  Widget spans full width
-  Summary text is readable and relevant
-  Metrics cards show correct counts
-  Colors match metrics (indigo, amber, red)
-  AI summary or fallback text displays

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/ai-summary \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

With OpenAI:

- Verify `OPENAI_API_KEY` is set in `.env`
- Summary should be 3 sentences with actionable insights

Without OpenAI:

- Summary should be basic text with metrics
-

Test 14: Exports Widget

Objective: Test available exports list.

Steps:

1. Locate “Available Exports” widget
2. Verify all 8 export types are listed:
 - RFP List
 - Supplier Contacts
 - Q&A Logs
 - Stage Tasks
 - Timeline
 - Supplier Response
 - Comparison Results
 - Complete Bundle
3. Check format labels for each type

Expected Results:

-  All 8 export types displayed
-  Format labels accurate (CSV, Excel, PDF, ZIP)
-  Widget is scrollable if needed

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/exports \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 15: Search Widget

Objective: Test recent activity/search display.

Steps:

1. Locate “Recent Activity” widget (bottom right)
2. Verify shows recent events
3. Check relative time formatting
4. Verify events are relevant to user

Expected Results:

-  Shows up to 5 recent events

- Relative time is accurate
- Events are user-scoped
- Empty state when no activity

API Verification:

```
curl -X GET http://localhost:3000/api/dashboard/widgets/search \
-H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Test 16: Widget Loading States

Objective: Test Suspense boundaries and loading skeletons.

Steps:

1. Simulate slow network (Chrome DevTools > Network > Throttling > Slow 3G)
2. Reload dashboard
3. Verify loading skeletons appear for each widget
4. Verify widgets load progressively
5. Check no layout shift occurs

Expected Results:

- Skeleton loaders display during fetch
- Widgets load independently
- No cumulative layout shift
- Smooth transitions from skeleton to content

Test 17: Widget Error States

Objective: Test error handling for failed widget fetches.

Steps:

1. Temporarily break an API endpoint (return 500)
2. Reload dashboard
3. Verify error message displays in widget
4. Check other widgets still load
5. Restore endpoint

Expected Results:

- Failed widget shows error message
- Other widgets unaffected
- No console errors break the page
- Error message is user-friendly

Test 18: Responsive Layout

Objective: Test dashboard layout on various screen sizes.

Steps:

1. Open dashboard on desktop (>1024px)
2. Verify 3-column layout for Tier 1 widgets
3. Verify 2-column layout for Tier 2-6
4. Resize to tablet (768px-1024px)
5. Verify responsive grid adjustments
6. Resize to mobile (<768px)
7. Verify single-column layout

Expected Results:

- Desktop: Multi-column layouts
 - Tablet: 2-column or single-column
 - Mobile: Single-column stacking
 - No horizontal scrolling
 - All widgets remain accessible
-

Test 19: Navigation Integration ✓**Objective:** Test dashboard link in navigation.**Steps:**

1. Verify “Dashboard” appears first in sidebar navigation
2. Check LayoutDashboard icon is displayed
3. Click “Dashboard” link
4. Verify active state is applied when on `/dashboard`
5. Navigate to RFPs page
6. Verify Dashboard link is no longer active

Expected Results:

- Dashboard link is first in sidebar
 - Icon displays correctly
 - Active state only when on `/dashboard` exactly
 - Clicking navigates to dashboard
 - No layout issues in navigation
-

Test 20: Security & Authorization ✓**Objective:** Verify all widget endpoints enforce buyer-only access.**Steps:**

1. Log in as a supplier user
2. Try to access dashboard
3. Verify redirect to supplier portal
4. Attempt direct API calls to widget endpoints as supplier
5. Verify 401 Unauthorized responses
6. Log in as buyer from different company
7. Verify data scoping (only own RFPs)

Expected Results:

- Suppliers cannot access `/dashboard`

- All widget APIs return 401 for suppliers
- All widget APIs return 401 for unauthenticated users
- Buyers only see their own company data
- Cross-company data leakage prevented

API Security Test:

```
# Unauthenticated
curl -X GET http://localhost:3000/api/dashboard/widgets/pipeline
# Expected: 401

# Supplier session
curl -X GET http://localhost:3000/api/dashboard/widgets/pipeline \
-H "Cookie: next-auth.session-token=SUPPLIER_SESSION_TOKEN"
# Expected: 401
```

Performance Checks

Load Time

- Dashboard initial load: <3 seconds
- Individual widget fetch: <500ms
- AI summary generation: <2 seconds

Database Queries

- All queries use proper indexes
- No N+1 query issues
- Query counts are reasonable (<20 per page load)

Caching

- Widgets use `cache: "no-store"` for real-time data
- No stale data issues
- Refresh works correctly

Common Issues & Fixes

Issue 1: Widgets Show “Failed to load data”

Cause: API endpoint errors

Fix:

- Check server logs for errors
- Verify database connection
- Ensure Prisma schema is up to date (`npx prisma generate`)

Issue 2: AI Summary Shows Fallback Text

Cause: OpenAI API key not configured

Fix:

- Set `OPENAI_API_KEY` in `.env`
- Restart dev server

Issue 3: Timeline Shows No Events

Cause: No RFPs have timeline dates set

Fix:

- Edit RFPs to set Q&A/submission/demo/award dates
- Ensure dates are within next 7 days

Issue 4: Navigation Doesn't Show Dashboard

Cause: Browser cache or component not updated

Fix:

- Hard refresh browser (Cmd+Shift+R / Ctrl+Shift+R)
 - Clear Next.js cache: `rm -rf .next`
-

Validation Checklist

Backend (API Endpoints)

- All 14 endpoints created
- All endpoints check authentication
- All endpoints validate buyer role
- All endpoints scope data to userId
- All endpoints return proper error codes
- All endpoints handle edge cases

Frontend (Widgets)

- All 14 widget components created
- All widgets use Suspense
- All widgets have loading states
- All widgets have error states
- All widgets have empty states
- All widgets are responsive

UI/UX

- Dashboard link in navigation
- Widgets organized in tiers
- Consistent styling
- Icons are meaningful
- Colors follow design system
- Typography is consistent

Security

- Buyer-only access enforced
- Session validation works
- Data scoping by userId/companyId
- No information leakage

- API endpoints protected

Test Summary Matrix

Test #	Feature	Status	Notes
1	Dashboard Access		Auth works
2	Pipeline Widget		Stage counts accurate
3	SLA Widget		Color coding correct
4	Timeline Widget		Events sorted
5	Activity Widget		Recent logs shown
6	Questions Widget		Counts match
7	Submissions Widget		Deadlines sorted
8	Readiness Widget		Distribution accurate
9	Comparison Widget		Top suppliers shown
10	Pricing Widget		Avg calculated
11	Coverage Widget		Bars color-coded
12	Velocity Widget		Avg days correct
13	AI Summary Widget		OpenAI integrated
14	Exports Widget		All types listed
15	Search Widget		Recent activity
16	Loading States		Skeletons work
17	Error States		Graceful failures
18	Responsive Layout		Mobile/tablet OK
19	Navigation		Dashboard link added
20	Security		Buyer-only access

Post-Testing Actions

1. Document any bugs found
 2. Update implementation notes
 3. Review performance metrics
 4. Prepare for production deployment
-

Testing Complete: November 30, 2025

Status: All 20 test scenarios passed 