

# Command Palette Component

---

## Overview

---

The Command Palette is a keyboard-driven modal search interface that provides quick access to RFPs, Companies, and Suppliers throughout the dashboard application.

## Features

---

### ✓ Keyboard Shortcuts

- **Open:** `⌘K` (Mac) or `Ctrl+K` (Windows/Linux)
- **Close:** `Escape` key or click outside the modal
- **Navigate:** `↑` and `↓` arrow keys
- **Select:** `Enter` key

### ✓ Search Functionality

- **Debounced Search:** 300ms delay to prevent excessive API calls
- **Minimum Query Length:** 2 characters required to start searching
- **Real-time Results:** Instant feedback as you type
- **API Endpoint:** Uses `/api/search` (same as global search)

### ✓ Result Categories

#### 1. RFPs

- Icon: `FileText` (lucide-react)
- Displays: Title, Status badge, Description
- Status colors: Draft (yellow), Published (blue), Completed (green)
- Navigate to: `/dashboard/rfps/[id]`

#### 2. Companies

- Icon: `Building2` (lucide-react)
- Displays: Company name
- Navigate to: `/dashboard/companies/[id]`

#### 3. Suppliers

- Icon: `Users` (lucide-react)
- Displays: Supplier name
- Navigate to: `/dashboard/suppliers/[id]`

### ✓ UI/UX Features

- **Modal Overlay:** Semi-transparent backdrop (`bg-black/50`)
- **Centered Layout:** 600px width (`max-w-2xl`)
- **Auto-focus:** Search input automatically focused when opened
- **Body Scroll Lock:** Prevents scrolling when modal is open
- **Keyboard Hints:** Visual guide at the bottom showing available shortcuts
- **Empty States:**
  - Initial: "Search for RFPs, Companies, or Suppliers"

- No Results: “No results found for [query]”
- **Loading Indicator:** Animated spinner during search
- **Smooth Animations:** Fade-in/slide-in effects using Tailwind

## File Structure

```

/home/ubuntu/fyndr/nextjs_space/
├── app/
│   ├── dashboard/
│   │   ├── command-palette.tsx      # Main component
│   │   ├── dashboard-layout.tsx     # Integration point
│   │   └── global-search.tsx        # Existing search (similar)
│   └── api/
│       └── search/
│           └── route.ts              # Search API endpoint

```

## Component Implementation

### Props Interface

```

interface CommandPaletteProps {
  isOpen: boolean;    // Controls modal visibility
  onClose: () => void; // Callback to close modal
}

```

## Key Implementation Details

### 1. State Management

```

typescript
const [query, setQuery] = useState('');
const [results, setResults] = useState<SearchResults>({...});
const [isLoading, setIsLoading] = useState(false);
const [selectedIndex, setSelectedIndex] = useState(-1);

```

### 2. Keyboard Event Listeners

- Global listener in `dashboard-layout.tsx` for `%K` / `Ctrl+K`
- Local listeners in component for navigation and escape

### 3. Auto-focus Behavior

```

typescript
useEffect(() => {
  if (isOpen && inputRef.current) {
    inputRef.current.focus();
  }
}, [isOpen]);

```

### 4. Body Scroll Prevention

```

typescript
useEffect(() => {
  if (isOpen) {
    document.body.style.overflow = 'hidden';
  } else {

```

```
document.body.style.overflow = 'unset';
}
}, [isOpen]));
```

## Testing Results

### ✓ All Tests Passed

#### 1. Keyboard Shortcuts

- [x] `Ctrl+K` opens the modal
- [x] `Escape` closes the modal
- [x] Click outside closes the modal
- [x] Browser default for `Ctrl+K` prevented

#### 2. Search Functionality

- [x] Search query "test" returns 2 RFPs
- [x] Search query "default" returns 1 Company and 1 Supplier
- [x] Debouncing works (300ms delay)
- [x] Loading spinner appears during search
- [x] Empty state shown when no query entered
- [x] No results message shown for empty results

#### 3. Keyboard Navigation

- [x] Down arrow moves selection down
- [x] Up arrow moves selection up
- [x] Enter navigates to selected item
- [x] Visual highlight on selected item (bg-indigo-50)

#### 4. Navigation & Routing

- [x] RFP navigation to `/dashboard/rfps/[id]`
- [x] Company navigation to `/dashboard/companies/[id]`
- [x] Supplier navigation to `/dashboard/suppliers/[id]`
- [x] Modal closes after navigation
- [x] Search cleared after navigation

#### 5. UI/UX

- [x] Modal centered on screen
- [x] Backdrop overlay visible
- [x] Auto-focus on search input
- [x] Keyboard hints displayed
- [x] Status badges color-coded
- [x] Icons displayed correctly
- [x] Smooth animations

## Integration Guide

The Command Palette is automatically integrated into the dashboard layout and requires no additional setup. It's available globally across all dashboard pages.

### Usage

1. Press `⌘K` (Mac) or `Ctrl+K` (Windows/Linux) anywhere in the dashboard

2. Type at least 2 characters to search
3. Use arrow keys to navigate results
4. Press `Enter` to navigate to selected item
5. Press `Escape` or click outside to close

## Styling

---

The component uses Tailwind CSS and matches the existing dashboard design:

- **Primary Color:** Indigo (indigo-600, indigo-50)
- **Text Colors:** Gray scale (gray-400 to gray-900)
- **Hover States:** indigo-50 background
- **Shadows:** shadow-xl for modal elevation
- **Borders:** gray-200 for subtle separation

## Performance Considerations

---

1. **Debounced Search:** Prevents excessive API calls
2. **Minimal Re-renders:** Uses refs for input and debounce timer
3. **Conditional Rendering:** Modal only renders when open
4. **Optimized Queries:** Search API returns max 10 results per category

## Future Enhancements (Optional)

---

- Recent searches history
- Search result preview on hover
- Fuzzy search matching
- Search filters (by status, date, etc.)
- Keyboard shortcut customization
- Command actions (e.g., "Create new RFP")

## Dependencies

---

- **React:** `useState`, `useEffect`, `useRef` hooks
- **Next.js:** `useRouter` for navigation
- **lucide-react:** Icons (Search, FileText, Building2, Users, Loader2)
- **Tailwind CSS:** Styling and animations

## Browser Compatibility

---

- Modern browsers with ES6+ support
- Chrome, Firefox, Safari, Edge (latest versions)
- Keyboard shortcuts work on Mac, Windows, and Linux

## Accessibility Notes

---

- Keyboard navigable
- Focus management handled properly

- ARIA labels could be added for screen readers (future enhancement)
  - High contrast color scheme for readability
- 

**Created:** November 28, 2025

**Status:**  Fully Implemented and Tested

**Location:** `/home/ubuntu/fyndr/nextjs_space/app/dashboard/command-palette.tsx`