```
import { useEffect, useState } from 'react';Ð
import { useParams, Link, useNavigate } from 'react-router-dom';Ð
import api from '../api/axios';Ð
import { Ð
  Folder, FileText, Plus, Loader2, ArrowLeft, CheckCircle, Ð
  MessageSquare, Trash2, UploadCloud, Home, Edit2 Ð
} from 'lucide-react';Ð
import toast, { Toaster } from 'react-hot-toast';Ð
import UploadModal from '../components/UploadModal';Ð
Ð
export default function Dashboard() {Ð
  const { folderId } = useParams();Ð
  const navigate = useNavigate();Ð
Ð
  // DataÐ
  const [data, setData] = useState({ folders: [], files: [], currentFolder: null });Ð
  const [loading, setLoading] = useState(true);Ð
  Ð
  // StateÐ
  const [creating, setCreating] = useState(false);Ð
  const [newFolderName, setNewFolderName] = useState("");Ð
  const [isUploadOpen, setIsUploadOpen] = useState(false);Ð
  Ð
  // ADVANCED SELECTION STATEÐ
  const [selectedItems, setSelectedItems] = useState([]); Ð
  const [lastSelectedId, setLastSelectedId] = useState(null); Ð
  Ð
  // RENAME STATEÐ
  const [editingId, setEditingId] = useState(null);Ð
  const [renameValue, setRenameValue] = useState("");Ð
Ð
  // --- 1. Fetch Data ---Ð
  useEffect(() => {Ð
    const fetchContents = async () => {Ð
      setLoading(true);Ð
      try {Ð
        const endpoint = `/folders/${folderId || 'root'}`;Ð
        const res = await api.get(endpoint);Ð
        Ð
        // Ø=Þáþ  SAFETY CHECK: Ensure arrays exist to prevent crashesÐ
        setData({Ð
            folders: res.data.folders || [],Ð
            files: res.data.files || [],Ð
            currentFolder: res.data.folder || nullÐ
        });Ð
        Ð
        setSelectedItems([]); Ð
        setLastSelectedId(null);Ð
      } catch (err) {Ð
        console.error(err);Ð
        toast.error("Failed to load contents");Ð
      } finally {Ð
        setLoading(false);Ð
      }Ð
    };Ð
    fetchContents();Ð
  }, [folderId]);Ð
Ð
  // Combine for Uniform List (Folders first)Ð
  // Ø=Þáþ  SAFETY CHECK: Use ( || []) to prevent "map of undefined" crashÐ
  const allItems = [Ð
    ...(data.folders || []).map(f => ({ ...f, type: 'folder' })),Ð
    ...(data.files || []).map(f => ({ ...f, type: 'file' }))Ð
  ];Ð
Ð
  // --- 2. Windows-Style Selection Logic ---Ð
  const handleItemClick = (e, item) => {Ð
    if (editingId === item._id) return;Ð
Ð
    const isSelected = selectedItems.includes(item._id);Ð
Ð
    // CTRL Key: Toggle SelectionÐ
    if (e.ctrlKey || e.metaKey) {Ð
      if (isSelected) {Ð
        setSelectedItems(prev => prev.filter(id => id !== item._id));Ð
      } else {Ð
        setSelectedItems(prev => [...prev, item._id]);Ð
        setLastSelectedId(item._id);Ð
      }Ð
    }Ð
    // SHIFT Key: Range SelectionÐ
    else if (e.shiftKey && lastSelectedId) {Ð
      const currentIndex = allItems.findIndex(i => i._id === item._id);Ð
      const lastIndex = allItems.findIndex(i => i._id === lastSelectedId);Ð
      Ð
      const start = Math.min(currentIndex, lastIndex);Ð
      const end = Math.max(currentIndex, lastIndex);Ð
      Ð
      const rangeIds = allItems.slice(start, end + 1).map(i => i._id);Ð
      setSelectedItems(prev => [...new Set([...prev, ...rangeIds])]);Ð
    }Ð
    // NORMAL Click: Select SingleÐ
    else {Đ
      setSelectedItems([item._id]);Đ
      setLastSelectedId(item._id);Đ
    }Đ
  };Đ
Đ
  // --- 3. Rename Logic ---Đ
  const handleNameClick = (e, item) => {Đ
    e.stopPropagation();Đ
    if (selectedItems.includes(item._id) && selectedItems.length === 1 && !e.ctrlKey && !
e.shiftKey) {Đ
       setTimeout(() => {Đ
         setEditingId(item._id);Đ
         setRenameValue(item.name || item.fileName);Đ
       }, 200);Đ
    } else {Đ
       handleItemClick(e, item);Đ
    }Đ
  };Đ
Đ
  const submitRename = async () => {Đ
    if (!renameValue.trim() || !editingId) {Đ
      setEditingId(null);Đ
      return;Đ
    }Đ
Đ
    const item = allItems.find(i => i._id === editingId);Đ
    const oldName = item.type === 'folder' ? item.name : item.fileName;Đ
    if (oldName === renameValue) {Đ
        setEditingId(null);Đ
        return;Đ
    }Đ
Đ
    try {Đ
        // '  API CALL RESTOREDĐ
        const endpoint = item.type === 'folder' ? `/folders/${item._id}` : `/files/${item._id}`;Đ
        const payload = item.type === 'folder' ? { name: renameValue } : { fileName:
renameValue };Đ
        Đ
        await api.put(endpoint, payload); Đ
        Đ
        // UI UpdateĐ
        if (item.type === 'folder') {Đ
            setData(prev => ({Đ
                ...prev,Đ
                folders: prev.folders.map(f => f._id === editingId ? { ...f, name:
```

```
renameValue } : f)Ð
            }));Ð
        } else {Ð
            setData(prev => ({Ð
                ...prev,Ð
                files: prev.files.map(f => f._id === editingId ? { ...f, fileName: renameValue } : f)Ð
            }));Ð
        }Ð
        toast.success("Renamed");Ð
    } catch (err) {Ð
        console.error(err);Ð
        toast.error("Rename failed");Ð
    } finally {Ð
        setEditingId(null);Ð
    }Ð
  };Ð
Ð
  const handleKeyDown = (e) => {Ð
    if (e.key === 'Enter') submitRename();Ð
    if (e.key === 'Escape') setEditingId(null);Ð
  };Ð
Ð
  // --- 4. Navigation (Double Click) ---Ð
  const handleDoubleClick = (item) => {Ð
    if (editingId) return; Ð
Ð
    if (item.type === 'folder') {Ð
        navigate(`/folder/${item._id}`);Ð
    } else {Ð
        // '  FIX: Go to File Viewer (Not Chat)Ð
        navigate(`/files/${item._id}`);Ð
    }Ð
  };Ð
Ð
  // --- 5. Actions ---Ð
  const handleDelete = async () => {Ð
    if (!window.confirm(`Delete ${selectedItems.length} items?`)) return;Ð
    try {Ð
        for (const id of selectedItems) {Ð
            const item = allItems.find(i => i._id === id);Ð
            // Handle both endpointsÐ
            const endpoint = item.type === 'folder' ? `/folders/${id}` : `/files/${id}`;Ð
            await api.delete(endpoint);Ð
        }Ð
        Ð
        setData(prev => ({Ð
          ...prev,Đ
          folders: prev.folders.filter(f => !selectedItems.includes(f._id)),Đ
          files: prev.files.filter(f => !selectedItems.includes(f._id))Đ
        }));Đ
        setSelectedItems([]);Đ
        toast.success("Deleted");Đ
    } catch (e) { console.error(e); toast.error("Delete failed"); }Đ
  };Đ
  Đ
  const handleStartChat = () => {Đ
      const fileIds = selectedItems.filter(id => {Đ
          const item = allItems.find(i => i._id === id);Đ
          return item && item.type === 'file';Đ
      });Đ
      if (fileIds.length === 0) return toast.error("Select files to chat");Đ
      // '  Pass files to chat correctlyĐ
      navigate('/chat', { state: { contextFiles: fileIds } });Đ
  };Đ
Đ
  const handleCreateFolder = async (e) => {Đ
    e.preventDefault();Đ
    if (!newFolderName.trim()) return;Đ
    try {Đ
      await api.post('/folders', { name: newFolderName, parentId: folderId || null });Đ
      setNewFolderName("");Đ
      setCreating(false);Đ
      // RefreshĐ
      const endpoint = `/folders/${folderId || 'root'}`;Đ
      const res = await api.get(endpoint);Đ
      setData({Đ
        folders: res.data.folders || [],Đ
        files: res.data.files || [],Đ
        currentFolder: res.data.folder || nullĐ
      });Đ
    } catch (err) {Đ
      console.error(err);Đ
      toast.error("Failed to create folder");Đ
    }Đ
  };Đ
Đ
  const handleUploadComplete = async () => {Đ
    try {Đ
      const endpoint = `/folders/${folderId || 'root'}`;Đ
      const res = await api.get(endpoint);Đ
      setData({Đ
        folders: res.data.folders || [],Đ
        files: res.data.files || [],Đ
        currentFolder: res.data.folder || nullĐ
      });Đ
    } catch (err) {Đ
      console.error(err);Đ
      toast.error("Failed to refresh after upload");Đ
    }Đ
  };Đ
Đ
  // Improved NavigationĐ
  const goUp = () => {Đ
    // '  FIX 3: Correct parent routingĐ
    if (data.currentFolder?.parentId) {Đ
        navigate(`/folder/${data.currentFolder.parentId}`);Đ
    } else {Đ
        navigate('/dashboard');Đ
    }Đ
  };Đ
Đ
  if (loading) return <div className="flex justify-center items-center h-full"><Loader2 
className="animate-spin text-blue-600" /></div>;Đ
Đ
  return (Đ
    <div Đ
      className="p-6 md:p-10 h-full overflow-y-auto select-none" Đ
      onClick={() => {Đ
        if (!editingId) setSelectedItems([]); Đ
      }}Đ
    >Đ
      <Toaster position="bottom-center" />Đ
      Đ
      {/* HEADER */}Đ
      <div className="flex flex-col md:flex-row justify-between items-start md:items-
center gap-4 mb-8" onClick={(e) => e.stopPropagation()}>Đ
        <div className="flex items-center gap-3 text-2xl font-bold text-gray-800">Đ
          {folderId ? (Đ
            <button onClick={goUp} className="p-2 hover:bg-gray-200 rounded-full 
transition text-gray-600">Đ
              <ArrowLeft className="w-6 h-6" />Đ
            </button>Đ
          ) : (Đ
            <div className="p-2 bg-blue-50 rounded-lg">Đ
               <Home className="w-6 h-6 text-blue-600" />Đ
            </div>Đ
          )}Đ
          <span className="truncate max-w-[200px] md:max-w-md">Đ
```

```
           {data.currentFolder ? data.currentFolder.name : "My Drive"}Đ
          </span>Đ
        </div>Đ
Đ
        <div className="flex gap-3">Đ
          <button onClick={() => setCreating(!creating)} className="flex items-center
gap-2 px-4 py-2 bg-white border border-gray-200 hover:bg-gray-50 rounded-xl text-sm
font-semibold transition shadow-sm">Đ
            <Plus className="w-4 h-4" /> New FolderĐ
          </button>Đ
          <button onClick={() => setIsUploadOpen(true)} className="flex items-center
gap-2 px-4 py-2 bg-blue-600 text-white hover:bg-blue-700 rounded-xl text-sm font-
semibold transition shadow-md shadow-blue-200">Đ
            <UploadCloud className="w-4 h-4" /> UploadĐ
          </button>Đ
        </div>Đ
      </div>Đ
Đ
      {/* CREATE FOLDER INPUT */}Đ
      {creating && (Đ
        <div className="mb-8" onClick={(e) => e.stopPropagation()}>Đ
          <form onSubmit={handleCreateFolder} className="flex gap-2 items-center bg-
white p-2 rounded-xl shadow-sm border border-blue-100 w-fit">Đ
            <div className="w-10 h-10 bg-blue-50 rounded-lg flex items-center justify-
center">Đ
              <Folder className="w-5 h-5 text-blue-600" />Đ
            </div>Đ
            <input autoFocus type="text" placeholder="Folder Name" className="outline-
none text-sm font-medium bg-transparent w-48 px-2" value={newFolderName}
onChange={(e) => setNewFolderName(e.target.value)} />Đ
            <button type="submit" className="bg-blue-600 text-white px-4 py-2 rounded-
lg text-sm font-bold hover:bg-blue-700 transition">Create</button>Đ
          </form>Đ
        </div>Đ
      )}Đ
Đ
      {/* UNIFIED GRID */}Đ
      <div className="grid grid-cols-2 md:grid-cols-4 lg:grid-cols-6 xl:grid-cols-7 gap-4
pb-32">Đ
          {allItems.length === 0 && !loading && (Đ
            <div className="col-span-full flex flex-col items-center justify-center py-20 text-
gray-400">Đ
              <Folder className="w-12 h-12 mb-2 opacity-20" />Đ
              <p>This folder is empty</p>Đ
            </div>Đ
          )}Đ
```

```
Đ
         {allItems.map(item => {Đ
           const active = selectedItems.includes(item._id);Đ
           const isEditing = editingId === item._id;Đ
           Đ
           return (Đ
             <div Đ
               key={item._id}Đ
               onClick={(e) => { e.stopPropagation(); handleItemClick(e, item); }}Đ
               onDoubleClick={(e) => { e.stopPropagation(); handleDoubleClick(item); }}Đ
               className={`Đ
                 relative group flex flex-col items-center justify-center p-4 rounded-xl cursor-
pointer transition-all duration-100 borderĐ
                 ${active Đ
                   ? 'bg-blue-100 border-blue-500 shadow-sm z-10' Đ
                   : 'bg-white border-transparent hover:bg-gray-50 hover:border-gray-200'Đ
                 }Đ
               `}Đ
             >Đ
               {/* Icon */}Đ
               <div className={`w-14 h-14 rounded-xl flex items-center justify-center mb-3
transition-colors Đ
                 ${active ? 'bg-blue-200' : 'bg-gray-100 group-hover:bg-white group-
hover:shadow-sm'}`}>Đ
                 {item.type === 'folder' ? (Đ
                   <Folder className={`w-7 h-7 ${active ? 'text-blue-700' : 'text-blue-500'}`} />Đ
                 ) : (Đ
                   <FileText className={`w-7 h-7 ${active ? 'text-blue-700' : 'text-gray-500'}`} /
>Đ
                 )}Đ
               </div>Đ
               Đ
               {/* Name / Rename Input */}Đ
               {isEditing ? (Đ
                   <input Đ
                       autoFocusĐ
                       type="text"Đ
                       className="w-full text-center text-sm font-medium bg-white border
border-blue-500 rounded px-1 py-0.5 outline-none shadow-lg z-50"Đ
                       value={renameValue}Đ
                       onChange={(e) => setRenameValue(e.target.value)}Đ
                       onBlur={submitRename}Đ
                       onKeyDown={handleKeyDown}Đ
                       onClick={(e) => e.stopPropagation()} Đ
                   />Đ
               ) : (Đ
```

```
          <span Đ
            onClick={(e) => handleNameClick(e, item)}Đ
            className={`text-sm font-medium truncate w-full text-center px-1
roundedĐ
              ${active ? 'text-blue-900' : 'text-gray-700'}Đ
            `}Đ
          >Đ
           {item.name || item.fileName}Đ
          </span>Đ
        )}Đ
Đ
        {/* Size */}Đ
        {!isEditing && item.type === 'file' && (Đ
          <span className="text-[10px] text-gray-400 mt-1 uppercase tracking-wide
font-medium">Đ
           {item.size ? `${(item.size / 1024).toFixed(0)} KB` : 'PDF'} Đ
          </span>Đ
        )}Đ
       </div>Đ
      );Đ
     })}Đ
   </div>Đ
Đ
   {/* FLOATING ACTION BAR */}Đ
   {selectedItems.length > 0 && (Đ
    <div onClick={(e) => e.stopPropagation()} className="fixed bottom-6 left-1/2
transform -translate-x-1/2 bg-gray-900 text-white px-3 py-2 rounded-2xl shadow-2xl flex
items-center gap-3 z-50 animate-in fade-in slide-in-from-bottom-4 border border-
gray-700">Đ
     <div className="px-3 py-1 bg-gray-800 rounded-lg">Đ
       <span className="font-bold text-xs">{selectedItems.length} selected</span>Đ
     </div>Đ
     Đ
     <button onClick={handleStartChat} className="flex items-center gap-2 px-3
py-2 hover:bg-gray-700 rounded-xl transition text-sm font-medium text-blue-300
hover:text-blue-200">Đ
       <MessageSquare className="w-4 h-4" /> StudyĐ
     </button>Đ
Đ
     <div className="w-px h-5 bg-gray-700"></div>Đ
Đ
     <button onClick={() => setEditingId(selectedItems[0])} className="flex items-
center gap-2 px-3 py-2 hover:bg-gray-700 rounded-xl transition text-sm font-medium
text-gray-300 hover:text-white" title="Rename (F2)">Đ
       <Edit2 className="w-4 h-4" />Đ
     </button>Đ
```

```
Đ
          <button onClick={handleDelete} className="flex items-center gap-2 px-3 py-2 
hover:bg-red-900/50 rounded-xl transition text-sm font-medium text-red-400 hover:text-
red-300">Đ
            <Trash2 className="w-4 h-4" />Đ
          </button>Đ
        </div>Đ
      )}Đ
      <UploadModal isOpen={isUploadOpen} onClose={() => setIsUploadOpen(false)} 
folderId={folderId} onUploadComplete={handleUploadComplete}/>Đ
    </div>Đ
  );Đ
}
```