

# Admin Features Implementation Guide

## ☑ COMPLETED

### Backend (100% Done)

- ☒ Database tables created (departments, department\_codes)
- ☒ All API endpoints implemented
- ☒ Password reset with bcrypt
- ☒ Requisition rerouting with audit logging
- ☒ Server running on http://localhost:3001

### Frontend State & Handlers (100% Done)

- ☒ All state variables added
- ☒ All handler functions implemented
- ☒ API client functions added
- ☒ Tab buttons added (Users, Vendors, Departments, Dept Codes, Reroute Reqs)

## ⚠ REMAINING: UI Components

You need to add the UI components for each new tab. Add these AFTER the vendors tab section (around line 2200+ in frontend/app.js):

### 1. Add Password Reset Button to Users Tab

Find the user table row (around line 2100) and add a "Reset Password" button next to Edit/Delete:

```
// In the users table, add this button in the actions column:
React.createElement('button', {
  onClick: () => {
    setResetUserId(user.id);
    setShowPasswordReset(true);
  },
  className: "text-yellow-600 hover:text-yellow-800 text-sm ml-2"
}, "Reset Password")

// Add this modal BEFORE the users table closes:
showPasswordReset && React.createElement('div', { className: "fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50" },
  React.createElement('div', { className: "bg-white rounded-lg p-6 max-w-md w-full" },
    React.createElement('h3', { className: "text-lg font-bold mb-4" }, "Reset User Password"),
    React.createElement('input', {
      type: "password",
      placeholder: "New Password (min 6 characters)",
      value: newPassword,
      onChange: (e) => setNewPassword(e.target.value),
      className: "w-full px-3 py-2 border rounded-lg mb-4"
    }),
    React.createElement('div', { className: "flex gap-2" },
      React.createElement('button', {
        onClick: handleResetPassword,
        className: "flex-1 px-4 py-2 bg-blue-600 text-white rounded-lg hover:bg-blue-700"
      }, "Reset Password"),
      React.createElement('button', {
        onClick: () => {
          setShowPasswordReset(false);
          setResetUserId(null);
          setNewPassword('');
        },
        className: "px-4 py-2 bg-gray-300 text-gray-700 rounded-lg hover:bg-gray-400"
      }, "Cancel")
    )
  )
)
```

### 2. Departments Tab UI

Add after the vendors tab section:

```
activeTab === 'departments' && React.createElement('div', { className: "bg-white rounded-lg shadow-sm border p-6" },
  React.createElement('div', { className: "flex items-center justify-between mb-4" },
    React.createElement('h3', { className: "text-xl font-bold text-gray-800" }, "Department Management"),
    React.createElement('button', {
      onClick: () => {
        setShowDepartmentForm(true);
        setEditingDepartment(null);
        setDepartmentForm({ name: '', code: '', description: '', is_active: 1 });
      },
      className: "px-4 py-2 bg-blue-600 text-white rounded-lg hover:bg-blue-700"
    }, "+ Add Department")
  ),

  // Department Form
  showDepartmentForm && React.createElement('div', { className: "mb-6 p-4 bg-gray-50 rounded-lg" },
    React.createElement('h4', { className: "font-semibold mb-4" }, editingDepartment ? 'Edit Department' : 'New Department'),
```

```

React.createElement('div', { className: "grid grid-cols-1 md:grid-cols-2 gap-4" },
  React.createElement('input', {
    type: "text",
    placeholder: "Department Name",
    value: departmentForm.name,
    onChange: (e) => setDepartmentForm({ ...departmentForm, name: e.target.value }),
    className: "px-3 py-2 border rounded-lg"
  }),
  React.createElement('input', {
    type: "text",
    placeholder: "Department Code",
    value: departmentForm.code,
    onChange: (e) => setDepartmentForm({ ...departmentForm, code: e.target.value }),
    className: "px-3 py-2 border rounded-lg"
  }),
  React.createElement('input', {
    type: "text",
    placeholder: "Description",
    value: departmentForm.description,
    onChange: (e) => setDepartmentForm({ ...departmentForm, description: e.target.value }),
    className: "px-3 py-2 border rounded-lg md:col-span-2"
  }),
  React.createElement('label', { className: "flex items-center gap-2" },
    React.createElement('input', {
      type: "checkbox",
      checked: departmentForm.is_active === 1,
      onChange: (e) => setDepartmentForm({ ...departmentForm, is_active: e.target.checked ? 1 : 0 }),
      className: "w-4 h-4"
    }),
    React.createElement('span', null, "Active")
  )
),
React.createElement('div', { className: "flex gap-2 mt-4" },
  React.createElement('button', {
    onClick: handleSaveDepartment,
    className: "px-4 py-2 bg-blue-600 text-white rounded-lg hover:bg-blue-700"
  }, "Save Department"),
  React.createElement('button', {
    onClick: () => {
      setShowDepartmentForm(false);
      setEditingDepartment(null);
    },
    className: "px-4 py-2 bg-gray-300 text-gray-700 rounded-lg hover:bg-gray-400"
  }, "Cancel")
)
),

// Departments Table
React.createElement('table', { className: "w-full" },
  React.createElement('thead', { className: "bg-gray-50" },
    React.createElement('tr', null,
      React.createElement('th', { className: "px-4 py-2 text-left" }, "Name"),
      React.createElement('th', { className: "px-4 py-2 text-left" }, "Code"),
      React.createElement('th', { className: "px-4 py-2 text-left" }, "Description"),
      React.createElement('th', { className: "px-4 py-2 text-left" }, "Status"),
      React.createElement('th', { className: "px-4 py-2 text-left" }, "Actions")
    )
  ),
  React.createElement('tbody', null,
    departments.map(dept =>
      React.createElement('tr', { key: dept.id, className: "border-b hover:bg-gray-50" },
        React.createElement('td', { className: "px-4 py-2" }, dept.name),
        React.createElement('td', { className: "px-4 py-2" }, dept.code),
        React.createElement('td', { className: "px-4 py-2" }, dept.description || '-'),
        React.createElement('td', { className: "px-4 py-2" },
          React.createElement('span', {
            className: `px-2 py-1 rounded text-xs ${dept.is_active ? 'bg-green-100 text-green-800' : 'bg-red-100 text-red-800'}`
          }, dept.is_active ? 'Active' : 'Inactive')
        ),
        React.createElement('td', { className: "px-4 py-2" },
          React.createElement('button', {
            onClick: () => {
              setEditingDepartment(dept);
              setDepartmentForm(dept);
              setShowDepartmentForm(true);
            },
            className: "text-blue-600 hover:text-blue-800 text-sm mr-2"
          }, "Edit"),
          React.createElement('button', {
            onClick: () => handleDeleteDepartment(dept.id),
            className: "text-red-600 hover:text-red-800 text-sm"
          }, "Delete")
        )
      )
    )
  )
)
)
)

```

```
)  
},
```

### 3. Department Codes Tab UI

```
activeTab === 'codes' && React.createElement('div', { className: "bg-white rounded-lg shadow-sm border p-6" },  
  React.createElement('div', { className: "flex items-center justify-between mb-4" },  
    React.createElement('h3', { className: "text-xl font-bold text-gray-800" }, "Department Codes Management"),  
    React.createElement('button', {  
      onClick: () => {  
        setShowCodeForm(true);  
        setEditingCode(null);  
        setCodeForm({ department_id: '', code: '', description: '', is_active: 1 });  
      },  
      className: "px-4 py-2 bg-blue-600 text-white rounded-lg hover:bg-blue-700"  
    }, "+ Add Code")  
  ),  
  
  // Code Form  
  showCodeForm && React.createElement('div', { className: "mb-6 p-4 bg-gray-50 rounded-lg" },  
    React.createElement('h4', { className: "font-semibold mb-4" }, editingCode ? 'Edit Code' : 'New Code'),  
    React.createElement('div', { className: "grid grid-cols-1 md:grid-cols-2 gap-4" },  
      React.createElement('select', {  
        value: codeForm.department_id,  
        onChange: (e) => setCodeForm({ ...codeForm, department_id: e.target.value }),  
        className: "px-3 py-2 border rounded-lg"  
      },  
        React.createElement('option', { value: "" }, "Select Department"),  
        departments.map(dept =>  
          React.createElement('option', { key: dept.id, value: dept.id }, dept.name)  
        )  
      ),  
      React.createElement('input', {  
        type: "text",  
        placeholder: "Code",  
        value: codeForm.code,  
        onChange: (e) => setCodeForm({ ...codeForm, code: e.target.value }),  
        className: "px-3 py-2 border rounded-lg"  
      },  
      React.createElement('input', {  
        type: "text",  
        placeholder: "Description",  
        value: codeForm.description,  
        onChange: (e) => setCodeForm({ ...codeForm, description: e.target.value }),  
        className: "px-3 py-2 border rounded-lg md:col-span-2"  
      },  
      React.createElement('label', { className: "flex items-center gap-2" },  
        React.createElement('input', {  
          type: "checkbox",  
          checked: codeForm.is_active === 1,  
          onChange: (e) => setCodeForm({ ...codeForm, is_active: e.target.checked ? 1 : 0 }),  
          className: "w-4 h-4"  
        },  
        React.createElement('span', null, "Active")  
      )  
    ),  
    React.createElement('div', { className: "flex gap-2 mt-4" },  
      React.createElement('button', {  
        onClick: handleSaveCode,  
        className: "px-4 py-2 bg-blue-600 text-white rounded-lg hover:bg-blue-700"  
      }, "Save Code"),  
      React.createElement('button', {  
        onClick: () => {  
          setShowCodeForm(false);  
          setEditingCode(null);  
        },  
        className: "px-4 py-2 bg-gray-300 text-gray-700 rounded-lg hover:bg-gray-400"  
      }, "Cancel")  
    )  
  ),  
  
  // Codes Table  
  React.createElement('table', { className: "w-full" },  
    React.createElement('thead', { className: "bg-gray-50" },  
      React.createElement('tr', null,  
        React.createElement('th', { className: "px-4 py-2 text-left" }, "Code"),  
        React.createElement('th', { className: "px-4 py-2 text-left" }, "Department"),  
        React.createElement('th', { className: "px-4 py-2 text-left" }, "Description"),  
        React.createElement('th', { className: "px-4 py-2 text-left" }, "Status"),  
        React.createElement('th', { className: "px-4 py-2 text-left" }, "Actions")  
      )  
    ),  
    React.createElement('tbody', null,  
      departmentCodes.map(code =>  
        React.createElement('tr', { key: code.id, className: "border-b hover:bg-gray-50" },  
          React.createElement('td', { className: "px-4 py-2 font-semibold" }, code.code),
```

```

    React.createElement('td', { className: "px-4 py-2" }, code.department_name || '-'),
    React.createElement('td', { className: "px-4 py-2" }, code.description || '-'),
    React.createElement('td', { className: "px-4 py-2" },
      React.createElement('span', {
        className: `px-2 py-1 rounded text-xs ${code.is_active ? 'bg-green-100 text-green-800' : 'bg-red-100 text-red-800'}`
      }, code.is_active ? 'Active' : 'Inactive')
    ),
    React.createElement('td', { className: "px-4 py-2" },
      React.createElement('button', {
        onClick: () => {
          setEditingCode(code);
          setCodeForm(code);
          setShowCodeForm(true);
        },
        className: "text-blue-600 hover:text-blue-800 text-sm mr-2"
      }, "Edit"),
      React.createElement('button', {
        onClick: () => handleDeleteCode(code.id),
        className: "text-red-600 hover:text-red-800 text-sm"
      }, "Delete")
    )
  )
)
)
),
),

```

#### 4. Reroute Requisitions Tab UI

```

activeTab === 'reroute' && React.createElement('div', { className: "bg-white rounded-lg shadow-sm border p-6" },
  React.createElement('h3', { className: "text-xl font-bold text-gray-800 mb-4" }, "Reroute Requisitions"),

  // Reroute Modal
  showRerouteModal && React.createElement('div', { className: "fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50" },
    React.createElement('div', { className: "bg-white rounded-lg p-6 max-w-md w-full" },
      React.createElement('h3', { className: "text-lg font-bold mb-4" }, "Reroute Requisition"),
      React.createElement('select', {
        value: rerouteForm.to_user_id,
        onChange: (e) => setRerouteForm({ ...rerouteForm, to_user_id: e.target.value }),
        className: "w-full px-3 py-2 border rounded-lg mb-3"
      }),
      React.createElement('option', { value: "" }, "Select User"),
      rerouteUsers.map(user =>
        React.createElement('option', { key: user.id, value: user.id }, `${user.full_name} (${user.role})`)
      ),
      React.createElement('textarea', {
        placeholder: "Reason for rerouting (required)",
        value: rerouteForm.reason,
        onChange: (e) => setRerouteForm({ ...rerouteForm, reason: e.target.value }),
        className: "w-full px-3 py-2 border rounded-lg mb-3",
        rows: 3
      }),
      React.createElement('select', {
        value: rerouteForm.new_status,
        onChange: (e) => setRerouteForm({ ...rerouteForm, new_status: e.target.value }),
        className: "w-full px-3 py-2 border rounded-lg mb-4"
      }),
      React.createElement('option', { value: "" }, "Keep Current Status"),
      React.createElement('option', { value: "pending_hod" }, "Pending HOD"),
      React.createElement('option', { value: "pending_procurement" }, "Pending Procurement"),
      React.createElement('option', { value: "pending_finance" }, "Pending Finance"),
      React.createElement('option', { value: "pending_md" }, "Pending MD")
    ),
    React.createElement('div', { className: "flex gap-2" },
      React.createElement('button', {
        onClick: handleReroute,
        className: "flex-1 px-4 py-2 bg-blue-600 text-white rounded-lg hover:bg-blue-700"
      }, "Reroute"),
      React.createElement('button', {
        onClick: () => {
          setShowRerouteModal(false);
          setRerouteReqId(null);
          setRerouteForm({ to_user_id: '', reason: '', new_status: '' });
        },
        className: "px-4 py-2 bg-gray-300 text-gray-700 rounded-lg hover:bg-gray-400"
      }, "Cancel")
    )
  ),
),

// Requisitions List
React.createElement('table', { className: "w-full" },
  React.createElement('thead', { className: "bg-gray-50" },
    React.createElement('tr', null,
      React.createElement('th', { className: "px-4 py-2 text-left" }, "Req #"),
      React.createElement('th', { className: "px-4 py-2 text-left" }, "Title"),
      React.createElement('th', { className: "px-4 py-2 text-left" }, "Status"),
      React.createElement('th', { className: "px-4 py-2 text-left" }, "Created By"),
      React.createElement('th', { className: "px-4 py-2 text-left" }, "Actions")
    )
  ),
  React.createElement('tbody', null,
    data.requisitions.map(req =>
      React.createElement('tr', { key: req.id, className: "border-b hover:bg-gray-50" },
        React.createElement('td', { className: "px-4 py-2" }, req.req_number || req.id),
        React.createElement('td', { className: "px-4 py-2" }, req.title || req.description),
        React.createElement('td', { className: "px-4 py-2" }, req.status),
        React.createElement('td', { className: "px-4 py-2" }, req.created_by_name),
        React.createElement('td', { className: "px-4 py-2" },
          React.createElement('button', {
            onClick: () => handleOpenReroute(req.id),
            className: "px-3 py-1 bg-yellow-100 text-yellow-800 rounded hover:bg-yellow-200 text-sm"
          }, "Reroute")
        )
      )
    )
  )
),

```

## Installation

The backend is ready. To add the UI components:

1. Open `frontend/app.js`
2. Find line ~2200 (after the vendors tab section)
3. Add the UI components above in order
4. Save and refresh your browser

## Testing

1. Login as admin (admin / admin123)
2. Click on Admin Panel
3. You should see 5 tabs: Users, Vendors, Departments, Dept Codes, Reroute Reqs
4. Test each feature:
  - Add/edit/delete departments
  - Add/edit/delete department codes
  - Reset user passwords from Users tab
  - Reroute requisitions from Reroute tab

All backend APIs are working and ready!