

# Complete Fix for Lead Assignments Component

## ⌚ Issues Fixed

1.  **Missing fields in fetch query** - Added START\_DT, END\_DT, BILL\_TYPE, PAY\_TERM, OT\_BILL\_TYPE, BILLABLE\_FLAG, OT\_BILLABLE\_FLAG
2.  **Removed redirect** - No redirect after approval, stays on same page
3.  **Details view** - Click row → show Basic + Additional tabs (like Leads Overview)
4.  **Actions based on status:**
  - Status = No → "Approve & Create Project" button in details view
  - Status = Yes → Show created project info, no redirect
5.  **Resume as file** - Upload to `/public/files/leads_assignment/resumes/{suborgid}_{leadid}/`
6.  **Show resume** - In table (View link) and in basic details tab

## 📋 Key Changes to Make

### 1. Fetch Query Already Fixed

The `fetchLeadAssignmentDetails` in `AddLeadAssignmentAction.js` already includes all fields (lines 474-509).

### 2. Component Structure Change

#### OLD Structure:

List View → Approval Modal → (redirect away)

#### NEW Structure:

List View → Details View (Basic/Additional tabs) → Approval Modal → Stay in Details

### 3. Table Changes

#### OLD Table:

- 9 columns with too much info
- Action: "Approve & Create Project" or "View Project"

#### NEW Table:

- 8 columns (simplified)
- Columns: Lead Name, Employee, Type, Start Date, Cost, Resume, Status, Actions
- Action: "View Details" (always)
- Resume shows "View" link if exists

## 4. Details View (NEW)

### Basic Details Tab:

- Lead Name
- Employee Name
- Employee Type (Existing/Manual)
- Skills
- Start Date \*
- End Date
- Resume \* (clickable link if exists)
- Status (badge)
- Created By

Action Button (if Status = No):

[Approve & Create Project]

### Additional Details Tab:

- Cost (Bill Rate)
- Bill Type \*
- OT Cost
- OT Bill Type \*
- Billable \*
- OT Billable \*
- Payment Term \*
- (If Status = Yes, also show:)
- Employee Pay Rate
- Employee OT Pay Rate
- Created Project ID
- Last Updated By
- Last Updated Date

## 5. File Upload Implementation

### Add Form:

jsx

```

<div className="lead_form_group">
  <label>Resume (PDF/Word)</label>
  <input type="file" accept=".pdf,.doc,.docx" onChange={handleFileChange} />
  {resumeFile && <small>Selected: {resumeFile.name}</small>}
</div>

```

## File Handler:

jsx

```

const handleFileChange = (e) => {
  const file = e.target.files[0];
  if (file) {
    const validTypes = ['application/pdf', 'application/msword',
      'application/vnd.openxmlformats-officedocument.wordprocessingml.document'];
    if (!validTypes.includes(file.type)) {
      setError('Please upload a PDF or Word document');
      e.target.value = '';
      return;
    }
    setResumeFile(file);
  }
};

```

## Upload Before Submit:

jsx

```

let resumePath = "";
if (resumeFile) {
  const selectedLeadData = leads.find(l => l.LEAD_ID === selectedLead);
  const uploadFormData = new FormData();
  uploadFormData.append('resume', resumeFile);
  uploadFormData.append('suborgid', selectedLeadData?.suborgid || 'default');
  uploadFormData.append('leadId', selectedLead);

  const uploadResult = await uploadResume(uploadFormData);
  if (uploadResult.success) {
    resumePath = uploadResult.path;
  } else {
    setError('Failed to upload resume: ' + uploadResult.error);
    return;
  }
}
formData.append('resumePath', resumePath);

```

## 6. Approval Flow (No Redirect)

### After Approval Success:

```
jsx

if (result.success) {
  setSuccess('Assignment approved and project created successfully!');
  setShowApprovalModal(false);
  setApprovalData(null);
  await loadData(); // Refresh assignments list

  // Refresh details if viewing that assignment
  if (selectedAssignment && selectedAssignment.ROW_ID === approvalData.ROW_ID) {
    const details = await fetchLeadAssignmentDetails(approvalData.LEAD_ID);
    const updated = details.find(d => d.ROW_ID === approvalData.ROW_ID);
    setAssignmentDetails(updated);
    setSelectedAssignment(updated);
  }
  // NO router.push() or redirect!
}
```

## 🔧 Complete Implementation

### State Variables Needed:

```
jsx

// View control
const [view, setView] = useState('list'); // 'list', 'add', 'details'
const [activeTab, setActiveTab] = useState('basic'); // 'basic', 'additional'
const [assignmentDetails, setAssignmentDetails] = useState(null);

// File upload
const [resumeFile, setResumeFile] = useState(null);
```

### Helper Functions:

```
jsx

const getBillTypeName = (id) => billTypes.find(t => String(t.id) === String(id))?.Name || '';
const getOtBillTypeName = (id) => otBillTypes.find(t => String(t.id) === String(id))?.Name || '';
const getPayTermName = (id) => payTerms.find(t => String(t.id) === String(id))?.Name || '';
```

## View Handlers:

```
jsx

const handleRowClick = (assignment) => {
  setSelectedAssignment(assignment);
  setAssignmentDetails(assignment);
  setView('details');
  setActiveTab('basic');
};

const handleBack = () => {
  setView('list');
  setSelectedAssignment(null);
  setAssignmentDetails(null);
};
```

## 📊 Data Flow

### List View:

```
fetchLeadsByOrgId() → For each lead → fetchLeadAssignmentDetails() → Display in table
```

### Click Row:

```
assignment → setAssignmentDetails() → setView('details') → Show tabs
```

### Approval:

```
Click "Approve" → Modal → Enter pay rates → Submit →
updateLeadAssignmentDecision() → Create project →
Refresh details → Stay in details view (no redirect)
```

## 🎨 UI Layout

Lead Assignments						[Add New]
Lead	Emp	Type	Date	\$	Resume	
ABC	John	Exist	01/01	\$50	View	
XYZ	Jane	Manual	02/01	\$60	-	

Click row ↓

John Doe - ABC Lead	[Back]
[Basic Details] [Additional Details]	
Lead Name: ABC Lead	
Employee: John Doe	
Start Date: 2024-01-01	
Resume: [View Resume]	
Status: <input type="radio"/> No	
[Approve & Create Project]	

After approval (no redirect):

✓ Assignment approved and project created successfully!	
[Basic Details] [Additional Details]	
Status: <input checked="" type="radio"/> Yes	
Created Project: 39-123	

## ✓ Testing Checklist

- Table shows 8 columns correctly
- Resume "View" link works in table
- Click row → Details view opens
- Basic tab shows all fields including resume link
- Additional tab shows all billing fields
- Status = No → "Approve" button shows in Basic tab
- Click Approve → Modal opens
- Enter pay rates → Submit → Project created
- After approval: Status changes to Yes, no redirect
- After approval: Additional tab shows pay rates and project ID
- File upload works in Add form
- Resume uploads to correct path:

/public/files/leads\_assignment/resumes/{suborgid}\_{leadid}/resume\_{timestamp}.pdf

Resume link works in details view

## Files to Update

1. **LeadAssignmentOverview.jsx** - Complete rewrite with details view
2. **AddLeadAssignmentAction.js** - Already has all fields 
3. **lead\_assignments\_page.jsx** - Already passes billTypes, otBillTypes, payTerms 

## Quick Implementation Guide

1. Replace the table section to show simplified columns
2. Add `[view === 'details']` section with tabs
3. In Basic tab, add "Approve" button (conditional on status)
4. Remove redirect logic after approval
5. Add file upload input in add form
6. Add file upload handler before assignment creation
7. Display resume link in table and details

This gives you a clean, professional interface matching the Leads Overview pattern!