# 1. Payments vs Invoices

### ✅ Changes Implemented:
- Payments are now properly linked to invoices
- Invoice status updates automatically upon payment creation
- Payment amount auto-fills when selecting an invoice
- Payment receipt print functionality added

### 📍 Testing Locations (4 Dashboards):

#### *A. Admin Dashboard*
*Location:* Admin Dashboard → Clients → [Select Client] → Payments Tab

*Testing Steps:*
1. Log in to Admin Dashboard
2. Click "Clients" in left sidebar
3. Open any client profile
4. Navigate to "Payments" tab
5. Click "Add Payment" button
6. *Verify:*
   - ✅ All invoices appear in invoice dropdown
   - ✅ "Amount" field auto-fills with invoice due amount upon selection
   - ✅ Complete payment details (Date, Payment Method, Notes)
   - ✅ Click "Save Payment"
   - ✅ New payment appears in payment list
   - ✅ Invoice status updates correctly (Partially Paid / Fully Paid)

  - ✅ Invoice list refreshes with updated status

*Payment Receipt Print:*
1. Click "Print Receipt" button next to any payment
2. *Verify:*
  - ✅ Print window opens successfully
  - ✅ Receipt displays payment details (Amount, Date, Invoice No.)

#### *B. Client Dashboard*
*Location:* Client Dashboard → Invoices → [Select Invoice] → Add Payment

*Testing Steps:*
1. Log in to Client Dashboard
2. Click "Invoices" in left sidebar
3. Open any invoice
4. *Verify:*
  - ✅ "Add Payment" button is visible
  - ✅ Invoice number is pre-filled
  - ✅ "Amount Due" displays correctly
5. Click "Add Payment"
6. In payment modal:
  - ✅ Invoice number is pre-selected
  - ✅ Amount auto-fills with due amount
7. Complete payment details and click "Save"
8. *Verify:*
  - ✅ New payment appears in Payment History
  - ✅ Invoice status updates
  - ✅ Balance Due reflects updated amount

*Alternative Location:* Client Dashboard → Payments → Add Payment
*Testing Steps:*
1. Open "Payments" page in Client Dashboard
2. Click "Add Payment"
3. *Verify:*
   - ✅ All invoices appear in dropdown
   - ✅ Amount auto-fills upon invoice selection
4. Save payment
5. *Verify:*
   - ✅ Payment appears in payment list

#### *C. Admin Dashboard - Invoice Detail*
*Location:* Admin Dashboard → Invoices → [Select Invoice] → Payments Button

*Testing Steps:*
1. Open "Invoices" in Admin Dashboard
2. Select any invoice
3. Click "Payments" button
4. *Verify:*
   - ✅ Payment history displays correctly
   - ✅ "Add Payment" button is available
5. Add payment and verify invoice status updates

---

# 2. Task Comments System

### ✅ Changes Implemented:
- User name and timestamp display for all comments

### 📍 Testing Locations (4 Dashboards):

#### *A. Admin Dashboard*
*Location:* Admin Dashboard → Tasks → [Select Task] → Comments Section
*Testing Steps:*
1. Open "Tasks" → Select any task
2. Scroll to Comments section
3. *Verify:*
   - ✅ Existing comments display (if any)
   - ✅ Comment input box is available
4. Type comment → Click "Add Comment"/"Post"
5. *Verify:*
   - ✅ Comment appears immediately
   - ✅ User name and timestamp display correctly
   - ✅ Comment persists after page refresh

#### *B. Client Dashboard*
*Location:* Client Dashboard → Tasks → [Select Task] → Comments
*Testing Steps:*
1. Open "Tasks" → Select task
2. Add comment
3. *Verify:*
   - ✅ Comment saves successfully
   - ✅ Comment appears in comment list

#### *C. Employee Dashboard*
*Location:* Employee Dashboard → My Tasks → [Select Task] → Comments

*Testing Steps:*
1. Open "My Tasks" → Select task
2. Add comment
3. *Verify:*
   - ✅ Comment saves successfully
   - ✅ Comment displays correctly with metadata

#### *D. Project Detail (Admin)*
*Location:* Admin Dashboard → Projects → [Select Project] → Tasks Tab → [Select Task] → Comments
*Testing Steps:*
1. Open project → Navigate to Tasks tab → Open task
2. Add comment and verify display

---

# 3. Client Statement

#### *A. Admin Dashboard*
*Location:* Admin Dashboard → Clients → [Select Client] → Statement Tab
*Testing Steps:*
1. Open "Clients" → Select client with invoices/payments
2. Click "Statement" tab
3. *Verify:*
   - ✅ Statement data loads correctly
   - ✅ Invoices list shows (Invoice No., Date, Amount, Status)
   - ✅ Payments list shows (Payment Date, Amount, Invoice No.)
   - ✅ Running balance calculates accurately

   - ✅ Summary displays: Opening Balance, Total Invoiced, Total Paid, Balance Due
4. Apply date range filter (Start Date, End Date)
5. *Verify:*
   - ✅ Filtered data displays correctly
6. Click "Print Statement"
7. *Verify:*
   - ✅ Print window opens
   - ✅ Statement prints with proper formatting

---

# 4. Contract Module

### ✅ Changes Implemented:
- Contract templates system
- Rich Text Editor integration
- Template-based contract creation
- Dynamic field replacement (Client Name, Project Name, etc.)
- Template switching functionality
- PDF generation (conceptual)
- Email sending capability

### 📍 Testing Location:

#### *A. Admin Dashboard*
*Location:* Admin Dashboard → Contracts → Add/Edit Contract
*Testing Steps:*
1. Open "Contracts" → Click "Add Contract"

2. *Verify:*
   - ✅ Contract form loads
   - ✅ "Select Template" dropdown available
3. Select template
4. *Verify:*
   - ✅ Template content loads automatically
   - ✅ Rich Text Editor displays content
   - ✅ Dynamic fields populate correctly
5. Complete contract:
   - Select Client/Project
   - Add contract items (Description, Quantity, Rate, Amount)
   - Edit terms via Rich Text Editor
6. Click "Save Contract"
7. *Verify:*
   - ✅ Contract saves successfully
   - ✅ Appears in contract list

*Change Template:*
1. Edit existing contract → Click "Change Template"
2. *Verify:*
   - ✅ Template selection modal opens
   - ✅ Templates list displays with search functionality
3. Select new template
4. *Verify:*
   - ✅ Warning appears for unsaved changes
   - ✅ Content updates to new template

*Contract Actions:*
- ✅ "Send Email": Triggers email send
- ✅ "Download PDF": Downloads PDF (if implemented)
- ✅ "Print": Opens print window

---

# 5. Client Notes

### 📍 Testing Location:

#### *A. Admin Dashboard*
*Location:* Admin Dashboard → Clients → [Select Client] → Notes Tab
*Testing Steps:*
1. Open client profile → Click "Notes" tab
2. *Verify:*
   - ✅ Existing notes display (if any)
   - ✅ "Add Note" button available
3. Click "Add Note"
4. In modal:
   - ✅ Title field present
   - ✅ Content field (Rich Text Editor) available
   - ✅ File attachment option visible (if implemented)
5. Save note
6. *Verify:*
   - ✅ Note appears immediately
   - ✅ User name and timestamp display
   - ✅ Note persists after refresh

*Edit/Delete Note:*
1. Click "Edit" on existing note → Verify edits save
2. Click "Delete" → Verify note removes permanently

---

# 6. Comment Boxes (All Modules)

### ✅ Changes Implemented:
- Project comments fixed (using notes API)
- Task comments fixed (using task_comments table)
- Client comments fixed (using notes API)
- Enhanced validation and error handling

### 📍 Testing Locations:

#### *A. Project Comments (Admin Dashboard)*
*Location:* Admin Dashboard → Projects → [Select Project] → Comments Tab
*Testing Steps:*
1. Open project → Click "Comments" tab
2. Add comment → Click "Add Comment"
3. *Verify:*
   - ✅ Comment saves successfully
   - ✅ Comment appears in list
   - ✅ No "Failed to connect" errors

#### *B. Task Comments*
*(Covered in Section 2)*

#### *C. Client Comments*
*(Covered in Section 5 - Notes)*

#### *D. Discussion Comments*
*Note:* Discussion comments not yet implemented (separate system). Test only if available.

---

# 7. Roles & Permissions

### ✅ Changes Implemented:
- Complete roles & permissions system
- Dynamic module management
- Granular permissions (View, Add, Edit, Delete)
- Role assignment to users
- Permission-checking middleware
- Default roles auto-creation

### 📍 Testing Location:

#### *A. Admin Dashboard*
*Location:* Admin Dashboard → System & Settings → Roles & Permissions
*Testing Steps:*
1. Navigate to Roles & Permissions
2. *Verify:*
   - ✅ Roles list displays
   - ✅ Default roles visible (ADMIN, EMPLOYEE, CLIENT, MANAGER, HR)
3. Click "Add Role" → Create "Test Role"
4. *Verify:*
   - ✅ Role creates successfully
   - ✅ Appears in roles list

*Set Permissions:*
1. Click "Permissions" for any role

2. *Verify:*
   - ✅ Permission modal opens
   - ✅ Modules list with checkboxes (View, Add, Edit, Delete) displays
3. Configure permissions → Click "Save Permissions"
4. *Verify:*
   - ✅ Permissions save successfully
   - ✅ Modal closes

---

# 8. Staff Login & Role-Based Access

### ✅ Changes Implemented:
- Employee dashboard now role-based
- Permission checks on all actions
- Dynamic menu items based on permissions
- Module visibility controlled by permissions

### 📍 Testing Location:

#### *A. Employee Dashboard*
*Location:* Employee Dashboard (Login as Employee)
*Testing Steps:*
1. Log in with Employee account
2. *Verify:*
   - ✅ Dashboard loads
   - ✅ Stats cards show only permitted modules
3. Check left sidebar
4. *Verify:*
   - ✅ Only permitted menu items visible

  - ✅ Restricted items hidden

*Page-Specific Verification:*
- **My Tasks:** "Add Task"/"Edit" buttons visible only with respective permissions
- **My Projects:** Page accessible only with `canView('myProjects')`
- **Time Tracking:** "Add Time Entry" visible only with `canAdd('timeTracking')`
- **Attendance:** "Check In/Out" visible only with `canAdd('attendance')`
- **Leave Requests:** "Add Leave"/"Edit" visible per permissions

*Permission Test:*
1. Assign Employee "View Only" permissions for Tasks
2. Log in as Employee
3. *Verify:*
   - ✅ Tasks page accessible
   - ✅ "Add Task" and "Edit" buttons hidden

---

# 9. Activities System

### ✅ Changes Implemented:
- Activities table created
- Activities API implemented
- Project Detail page shows real activities (no mock data)

### 📍 Testing Location:

#### *A. Admin Dashboard*
*Location:* Admin Dashboard → Projects → [Select Project] → Activities Tab
*Testing Steps:*
1. Open project → Click "Activities" tab
2. *Verify:*
   - ✅ Activities list loads
   - ✅ Real activities display (no mock data)
   - ✅ Shows User, Action, Description, Timestamp
3. Perform action in project (e.g., add task)
4. *Verify:*
   - ✅ Activity logs automatically
   - ✅ List refreshes with new activity

*Note:* Activity logging requires manual implementation in controllers via `activityController.logActivity`.

---

# 10. Task Labels System

### ✅ Changes Implemented:
- Labels load from database (no hardcoded values)
- Labels persist to database on add/delete
- Default labels auto-create on first load

### 📍 Testing Location:

#### *A. Admin Dashboard*
*Location:* Admin Dashboard → Tasks → Manage Labels

*Testing Steps:*
1. Open Tasks page → Click "Manage Labels"
2. *Verify:*
   - ✅ Labels load from database
   - ✅ Default labels visible (Bug, Design, Enhancement, Feedback)
3. *Add Label:*
   - Enter name (e.g., "Urgent") + select color → Click "Add"
   - ✅ Label saves to database and appears immediately
4. *Delete Label:*
   - Click delete icon → ✅ Label removes from DB and UI
5. *Use in Tasks:*
   - Assign label when creating/editing task
   - ✅ Label saves with task and displays on task detail page