Proposed comprehensive commenting and approval system design:

1. User Roles Hierarchy:
   Department Level:
   - Prof. Baraka Maiseli (Head of Department)

   Lab Supervision Level:
   - Dr. Nassor Ally (Lab Supervisor)
   - Mr. James Micheal (Lab Supervisor)

   Technical Level:
   - Mr. Adrian (Lab Technician)

   Management Level:
   - You (Lab Chairperson)

2. Commenting System Features:

   A. **Section-Level Comments**:
   - Each section (Launch Strategy, Requirements, Specifications, etc.) should have:
     - Comment thread
     - Approval status
     - Review history
     - Required approvers list

   B. **Project-Level Comments**:
   - General comments section for the entire project
   - Overall project approval status
   - Project milestones review

3. Comment Types:
   a) Review Comments:
   - Technical review
   - Administrative review
   - Resource allocation review
   - Safety compliance review

   b) Approval States:
   - Pending Review
   - Under Review
   - Changes Requested
   - Approved
   - Rejected

c) Comment Priority:
- Critical (Blocking)
- Important
- Suggestion
- Note

4. Workflow:

1. Initial Submission:
   - Section marked as "Pending Review"
   - Notification sent to relevant reviewers

2. Review Process:
   - Reviewers can add comments
   - Tag specific team members
   - Request changes
   - Approve their part

3. Resolution:
   - Changes implemented
   - Reviewers notified
   - Re-review if needed

4. Final Approval:
   - All required approvers must sign off
   - Hierarchical approval (e.g., technician → supervisors → department head)

5. Comment Features:
- Rich text formatting
- File attachments
- @mentions
- Comment threads
- Edit history
- Resolution status
- Due dates for requested changes
- Email notifications

6. Dashboard Views:

a) Personal Dashboard:
- Items pending your review
- Your recent comments
- Action items assigned to you

b) Project Dashboard:

- Overall approval status
- Pending reviews
- Recent activity
- Blocked items

c) Department Dashboard:

- Cross-project status
- Department-level approvals
- Resource allocation

7. Integration Points:

- Project Timeline: Mark review milestones
- Activity Log: Track review/approval history
- Notification System: In-app alerts
- Document Management: Link comments to specific documents
- Version Control: Track changes made after reviews

Implementation idea:

1. The basic commenting structure
2. The approval workflow
3. The notification system
4. The dashboard views

Comment management integrating task’s review column

1. Task Review Flow

Doing -> Review -> Done

When task moves to Review:

- Automatically creates a review request
- Notifies relevant reviewers based on task type
- Blocks movement to Done until approved

2. Review Process Integration

Task in Review Column:

```
├── Review Status Badge
│   ├── 🟡 Pending Review
│   ├── 🔵 Under Review
│   ├── 🟠 Changes Requested
│   └── 🟢 Approved
│
├── Reviewers Section
│   ├── Required Reviewers
│   │   ├── ○ Prof. Baraka Maiseli (if needed)
│   │   ├── ○ Dr. Nassor Ally
│   │   ├── ○ Mr. James Micheal
│   │   ├── ○ Mr. Adrian
│   │   └── ○ You (Lab Chairperson)
│   │
│   └── Review Status
│       ├── Approved by: [Names]
│       ├── Pending: [Names]
│       └── Changes Requested by: [Names]
│
└── Comments Thread
    ├── Technical Review Comments
    ├── Required Changes
    └── Approval Comments
```

3. Automation Rules
   When task enters Review:
   - Assign required reviewers based on task type
   - Create review checklist
   - Send notifications

   When all approvals received:
   - Allow movement to Done column
   - Update task status
   - Log completion with review references

   If changes requested:
   - Move back to Doing
   - Tag original assignee
   - Create subtasks for requested changes

4. Review Requirements Matrix

Task Type	Required Reviewers
Technical Implementation	→ Lab Technician + Supervisors
Resource Allocation	→ Lab Chairperson + Department Head
Safety Protocols	→ All Levels
Documentation	→ Supervisors
General Tasks	→ Direct Supervisor

Implementation idea

1. Add the review system structure to tasks

2. Create the review UI components
3. Implement the automation rules
4. Set up the notification system

The key benefit of this integration is that it:

- Enforces proper review processes
- Maintains clear accountability
- Provides transparency in the review process
- Automates the workflow
- Creates a documented approval trail