

PROJECT MANAGEMENT TOOL

```
from django.db import models
from django.contrib.auth.models import User

class Project(models.Model):
    name = models.CharField(max_length=255)
    description = models.TextField(blank=True)
    owner = models.ForeignKey(User, related_name='owned_projects', on_delete=models.CASCADE)
    members = models.ManyToManyField(User, related_name='projects')
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.name

class Task(models.Model):
    STATUS_CHOICES = [
        ('TODO', 'To Do'),
        ('IN_PROGRESS', 'In Progress'),
        ('DONE', 'Done'),
    ]
    project = models.ForeignKey(Project, related_name='tasks', on_delete=models.CASCADE)
    title = models.CharField(max_length=255)
    description = models.TextField(blank=True)
    assigned_to = models.ForeignKey(User, related_name='assigned_tasks', on_delete=models.SET_NULL,
null=True, blank=True)
    status = models.CharField(max_length=20, choices=STATUS_CHOICES, default='TODO')
    due_date = models.DateTimeField(null=True, blank=True)
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f"{self.project.name} - {self.title}"

class Comment(models.Model):
    task = models.ForeignKey(Task, related_name='comments', on_delete=models.CASCADE)
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    content = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)

class Meta:
    ordering = ['created_at']

    def __str__(self):
        return f"Comment by {self.user.username} on {self.task.title}"
```

Task Details & Commenting Component :

```
// TaskDetailsModal.jsx (Conceptual React Component)
import React, { useState, useEffect } from 'react';
import axios from 'axios'; // For API calls

function TaskDetailsModal({ taskId, onClose }) {
  const [task, setTask] = useState(null);
  const [comments, setComments] = useState([]);
  const [newComment, setNewComment] = useState('');

  // Fetch Task and Comments on load
  useEffect(() => {
    axios.get(`/api/tasks/${taskId}`)
      .then(res => setTask(res.data));
    axios.get(`/api/tasks/${taskId}/comments/`)
      .then(res => setComments(res.data));
  }, [taskId]);

  const handleCommentSubmit = (e) => {
    e.preventDefault();
    axios.post(`/api/tasks/${taskId}/comments/`, { content: newComment })
      .then(res => {
        setComments([...comments, res.data]); // Update comments list
        setNewComment('');
      });
  };

  if (!task) return <div>Loading...</div>;

  return (
    <div className="modal">
      <h2>{task.title}</h2>
      <p>Assigned To: **{task.assigned_to?.username || 'Unassigned'}**</p>
      {/* ... dropdown for assigning user and changing status ... */}

      <h3>Comments</h3>
      <div className="comment-list">
        {comments.map(c => (
          <p key={c.id}>
            **{c.user.username}**: {c.content}
        ))
      
```

```
</p>
    ))
</div>

<form onSubmit={handleCommentSubmit}>
  <textarea
    value={newComment}
    onChange={(e) => setNewComment(e.target.value)}
    placeholder="Add a comment...">
  />
  <button type="submit">Post Comment</button>
</form>
<button onClick={onClose}>Close</button>
</div>
);
}
```