|  |  |
| --- | --- |
| **1. Task Management / To-Do App API** | **Status** |
|  |  |
| Custom user auth (JWT) | Pending |
| CRUD for tasks with status, priority, deadlines | partial pending |
| Assign tasks to users | partial pending |
| Filter by due date/status | Pending |
| Soft delete (optional) | Pending |
|  |  |
| Want to also work on below points for Task Management / To-Do App API |  |
|  |  |
| ✅ README.md with endpoint docs | Pending |
| ✅ Deployed backend (Render, Railway, etc.) | Pending |
| ✅ GitHub repo link | Pending |
| ✅ Bonus: Postman collection or Swagger docs | Pending |

That’s a perfect approach! ✅ You're doing exactly what a focused, high-quality freelancer would do — **completing one solid project end-to-end** before starting another.

Your checklist looks great and well-organized. Here’s a suggested **next-step order** for finishing your Task Management API efficiently:

**🔁 Recommended Completion Flow**

1. **JWT Custom Auth (User creation & login)**  
   → Finish the base so you can test the rest with real users.
2. **Complete CRUD for Tasks**  
   → Ensure all fields (status, priority, deadline) are functional.
3. **Assign Tasks to Users**  
   → Add assigned\_to field and permission check (only assign to existing users).
4. **Filter by Due Date / Status**  
   → Use DRF filters or custom query params like ?status=pending.
5. **(Optional) Soft Delete**  
   → Add a is\_deleted=True/False flag instead of permanently deleting.

**🎁 Final Touches (Freelancer-Ready Polish)**

1. ✅ README.md with endpoint docs  
   → Can use Markdown with examples of each endpoint (GET/POST etc.)
2. ✅ Deploy to Render or Railway  
   → Free & simple deployment. I can help you step-by-step when ready.
3. ✅ Upload project to GitHub  
   → Include README, .env.example, and setup steps.
4. ✅ Postman collection or Swagger docs  
   → Swagger: drf-yasg  
   → Postman: Export collection and share it in GitHub repo.