# Software Developer Take-Home Assignment

## 🌟 Objective:

Develop a feature-rich application that demonstrates your proficiency in software development, problem-solving, and understanding of real-world scenarios.

## 📂 Assignment Overview:

You're tasked with building a Task Management System tailored for a small team. The system should allow users to create, assign, track, and manage tasks efficiently.

## 🧹 Core Requirements:

✅ User Authentication:

* Secure user registration and login functionalities.
* Use industry-standard practices for password storage and session management.

✅ Task Management:

* Create tasks with attributes: title, description, due date, priority, and status.
* Implement full CRUD (Create, Read, Update, Delete) operations for tasks.

✅ Team Collaboration:

* Allow users to assign tasks to other registered users.
* Implement a notification system to alert users when a task is assigned.

✅ Dashboard:

* Tasks assigned to them
* Tasks they created
* Overdue tasks

✅ Search and Filter:

* Implement search by task title or description.
* Filtering based on status, priority, and due date.

## ⚙️ Technical Specifications:

* Frontend: Next.js
* Backend: Node.js with nestjs or Express
* Database: MongoDB or PostgreSQL
* Version Control: Host your code on a public Git repository with clear commit messages.

## 🧪 Evaluation Criteria:

* Code Quality (readability, documentation)
* Functionality (meeting the core requirements)
* Security (auth, validation)
* User Experience (UI/UX, responsiveness)
* Problem-Solving (handling edge cases)
* Commit History (logical and meaningful)

## ✉️ Submission Guidelines:

* Deadline: Submit within 7 days from receiving the assignment.
* **For submitting the response, Kindly fill out the following google form :-** [**https://forms.gle/QnLMWKDLeAYhzWGz5**](https://forms.gle/QnLMWKDLeAYhzWGz5)
* Deployment: Deploy your app (Vercel, Netlify, Render, Railway, etc.).
* Important: We will NOT run your code locally — deployment is mandatory.
* Repository: Share the public GitHub/GitLab repo link.
* Documentation: Include a README with:
* Setup instructions
* Approach explanation
* Any assumptions/trade-offs made

## 🌟 Optional (But Highly Impressive) Advanced Tasks:

* Role-Based Access Control (RBAC): Implement Admin, Manager, and Regular User roles with different permissions.
* Real-Time Notifications: Use WebSockets (Socket.io) to instantly notify users when a task is assigned or updated.
* Recurring Tasks: Add functionality for tasks that repeat daily, weekly, or monthly.
* Audit Logging: Maintain a log of all user actions (task created, assigned, deleted, updated) for transparency.
* Offline Support (PWA): Make your app a Progressive Web App (PWA) so it works offline and syncs when back online.
* Unit and Integration Tests: Add Jest or Mocha tests covering important functionalities and business logic.
* Analytics Dashboard: Track metrics like number of completed tasks per user, overdue trends, task completion rates.
* Customizable Notifications: Allow users to set their notification preferences (email, in-app alerts, mute certain updates).

## ⚡ Quick Note:

We encourage using AI tools for development. We would love to know in detail how you used AI to complete this assignment. We would be looking for original thinking, clear coding decisions, and your problem-solving approach. Be ready to discuss your implementation in a follow-up interview.