**Estimation for Basic Task Management App (Backend with Simple UI):**

If you’re focusing on a **backend with Node.js and MySQL**, plus a **basic frontend (HTML/CSS/JavaScript)**, here’s a reasonable estimate.

**1. Initial Setup (Node.js, MySQL, Project Setup): 1 - 2 days**

* Installing and configuring Node.js, Express, MySQL, and any necessary packages.
* Setting up the database schema for users and tasks.
* Creating a basic project structure (API routes, database connections).

**2. Basic CRUD Functionality (API): 2 - 3 days**

* Implementing CRUD operations for tasks: creating, reading, updating, and deleting tasks.
* Implementing basic user registration and login.
* Using bcrypt for password hashing and possibly JWT for authentication.
* Testing API routes with Postman or cURL.

**3. UI Implementation (Basic Frontend): 2 - 3 days**

* Designing simple HTML forms to add tasks, view tasks, and update tasks.
* Connecting the front-end to the back-end API using JavaScript (Fetch API or Axios).
* Testing the frontend interactions with the backend.

**4. User Authentication (Optional but recommended): 1 - 2 days**

* Implementing login, session handling, and user-specific task management.
* Securing task-related API endpoints so users can only access their own tasks.

**5. Final Touches and Testing: 2 - 3 days**

* Testing for bugs, edge cases, and security vulnerabilities.
* Adding error handling (e.g., handling missing fields, invalid credentials).
* Implementing a responsive design (if you’re going beyond basic HTML/CSS).

**6. Documentation and GitHub Setup: 1 day**

* Writing a detailed README.md file explaining the project setup and features.
* Setting up a GitHub repository and uploading the project code.
* (Optional) Recording demo videos or screenshots for your LinkedIn portfolio.

**Total Time Estimate for Basic Version: 8 - 13 days**

**Breakdown for More Advanced Features (Optional):**

If you want to extend the functionality beyond the basics, such as adding the following:

**1. Advanced Front-End (Using React or Vue): 3 - 5 extra days**

* Set up React or Vue, structure the front-end as components, and handle state management.
* Connect the front-end to the Node.js backend using Axios or Fetch.
* Handle routing on the front-end for different pages (task list, create task, etc.).

**2. Real-time Functionality (Optional with Socket.IO): 2 - 3 extra days**

* Implement real-time task updates using WebSockets with **Socket.IO** (e.g., if multiple users are collaborating).

**3. Deployment: 1 - 2 extra days**

* Deploy the backend to **Heroku**, **DigitalOcean**, or **AWS**.
* Deploy the database to **Heroku PostgreSQL**, **AWS RDS**, or another managed MySQL service.
* Deploy the front-end (if applicable) to **Netlify** or **Vercel**.