

# Fullstack Take-Home Project: React + Node + TypeScript + SQL + Next.js

Take-Home Project: Task Manager App (Fullstack TypeScript)

Goal:

Build a task management web app with user authentication, protected routes, and a SQL database.

The app should allow users to register, log in, and manage their personal tasks.

Features to Build:

Frontend (Next.js + React + TypeScript):

## 1. Auth Pages

- Register page
- Login page
- Use JWT stored in cookies or local storage

## 2. Task Dashboard

- Protected route that shows the user's tasks
- Create, edit, delete tasks
- Mark tasks as completed
- Filter by completion status

## 3. Error & Loading States

- Show errors on login/registration failure
- Display loading indicators on API calls

Backend (Node.js + Express + TypeScript):

## 1. Auth Routes

- POST /api/register - creates user with hashed password (bcrypt)
- POST /api/login - validates credentials, returns JWT
- Middleware to verify JWT and protect routes

## 2. Task Routes

- GET /api/tasks - fetch all tasks for logged-in user
- POST /api/tasks - create new task
- PUT /api/tasks/:id - update task
- DELETE /api/tasks/:id - delete task

Database (SQL):

- Use PostgreSQL (or MySQL)
- Tables:
  - users (id, name, email, password\_hash)
  - tasks (id, user\_id, title, description, completed, created\_at)

Tech Stack:

- Frontend: Next.js, React, TypeScript
- Backend: Node.js, Express, TypeScript
- Auth: JWT + bcrypt
- Database: PostgreSQL (or MySQL) via Prisma or TypeORM

What We're Looking For:

- Strong TypeScript usage on both client & server

- Clear project structure and organization
- Secure auth implementation
- RESTful routing and SQL query logic
- Clean, reusable React components

#### Bonus Points:

- Use Prisma ORM
- Responsive design
- Deploy using Vercel (frontend) and Render or Railway (backend)

#### Submission:

1. GitHub repo with setup instructions
2. Live demo (optional)
3. Notes on design choices or tradeoffs