Todo App Project Summary

What Was Asked

- Todo Application: Add, view, update, and delete todos.
- Authentication: Signup and Login functionality.
- Route Protection: Restrict access to `/` if not logged in.

What We Completed

- Backend (Node.js + Express + PostgreSQL):
  * Setup PostgreSQL with a `todos` table.
  * Routes: GET, POST, PUT, DELETE for todos.
  * Auth: Signup and Login endpoints.
  * PostgreSQL SSL toggle setup.

- Frontend (React + Axios + React Router):
  * Components: Login, Signup, Main (Todos).
  * Routing: Protected with `PrivateRoute`.
  * LocalStorage: Used for session tracking.
  * Axios: Used for API calls.

Extra Features Added

- Protected Routes with `PrivateRoute`.
- Toast Notifications using `react-toastify`.
- SSL config dynamically adjusted for PostgreSQL.
- Redirects post login/signup.
- Modular backend and frontend structure.

Features Not Implemented (Due to Time Constraints)

- Token-Based Auth (JWT).
- User-specific todos (currently shared).
- Logout functionality.

Todo App Project Summary

- Polished UI or styling upgrades.
- Todo filtering, tagging, or completion tracking.