

Todo Calendar App Tasks

Backend

- Add DELETE /todos/:id endpoint to server.js
- Update POST /todos in server.js to accept startTime and endTime
- Update POST /todos to accept category
- Install json-server package
- Replace server.js logic with json-server middleware
- Create initial db.json

Frontend Logic

- Update api.js with updateTodo and deleteTodo functions
- Update App.jsx to handle toggle complete and delete actions
- Add visual indicators for completed/deleted todos in the list
- (Optional) Add interaction to CalendarView items
- Update TodoForm.jsx to include start and end time inputs
- Update CalendarView.jsx to display times
- Update App.jsx/Sidebar to display times in the list
- Update TodoForm to include Category selector
- Update App.jsx to pass category data and render with colors
- Update CalendarView to color-code events by category

UI Overhaul

- Global Styles: Reset, Typography (Inter/Roboto), Variables
- Layout: Modern Card-based layout for Sidebar and Calendar
- Components: Styled Buttons, Inputs with focus states
- Calendar: Aesthetic grid, headers, and event "pills"
- Todo List: Clean list items with subtle hover effects and transitions
- Add category styles (colors) to App.css

Drag & Drop Feature

- Add updateDate handler to App.jsx
- Implement Drag & Drop logic in CalendarView.jsx
- Add visual feedback for drag targets in App.css

Click-to-Add Feature

- Add prefillDate state to App.jsx
- Update TodoForm to accept and observe prefillDate prop
- Update CalendarView to invoke onDateClick

Modal Form Feature

- Create Modal wrapper component logic (in App.jsx or new file)
- Style the Modal overlay and box in App.css
- Refactor App.jsx: Move TodoForm inside Modal
- Update App.jsx: Show Modal on date click, close on submit/cancel

Edit Todo Features

- Update App.jsx: Add editingTodo state and handleEdit function
- Update TodoForm.jsx: key off todoToEdit prop to pre-fill fields
- Update App.jsx: Modify handleModalSubmit to update if editing
- Update CalendarView.jsx: Change click handler to trigger edit
- Verify editing a todo updates it correctly

Refactoring & Polish

- Create generic Modal.jsx component
- Refactor App.jsx to use Modal component
- Move TodoForm inline styles to App.css
- Verify functionality remains unchanged

Verification

- Verify adding a todo persists
- Verify completing a todo updates state
- Verify deleting a todo removes it
- Verify creating a todo with time saves correctly
- Verify UI responsiveness and visual consistency
- Verify categories are saved and colorful
- Verify data persists after server restart
- Verify dragging a todo changes its date
- Verify clicking a date fills the form
- Verify form opens in modal on date click