### Soal Essay

1. Jelaskan perbedaan antara "bind()", "call()", dan "apply()" dalam konteks pengaturan nilai "this" dalam fungsi.

bind():

Metode bind() digunakan untuk mengikat nilai this pada sebuah fungsi dan mengembalikan fungsi baru yang memiliki this yang sudah terikat. Metode ini tidak langsung memanggil fungsi, tetapi mengembalikan fungsi yang siap dipanggil dengan this yang sudah ditentukan.

call():

Metode call() digunakan untuk memanggil fungsi dengan menentukan nilai this secara langsung. Selain itu, metode ini juga memungkinkan kita untuk memberikan argumen-argumen fungsi secara terpisah setelah nilai this.

apply():  
Metode apply() mirip dengan call(), tetapi argumen-argumen fungsi diberikan dalam bentuk array.

1. Apa itu "memoization" dalam konteks pemrograman JavaScript? Jelaskan bagaimana memoization dapat meningkatkan performa fungsi.

"Memoization" dalam konteks pemrograman JavaScript adalah teknik yang digunakan untuk menyimpan hasil komputasi dari suatu fungsi berdasarkan argumen yang diberikan. Dengan menyimpan hasil tersebut, kita dapat menghindari pengulangan komputasi yang mahal secara berulang-ulang saat fungsi dipanggil dengan argumen yang sama. Memoization dapat meningkatkan performa fungsi dengan mengurangi waktu eksekusi dan kompleksitas yang diperlukan.

1. Jelaskan perbedaan antara "synchronous" dan "asynchronous" dalam konteks pemrograman JavaScript.

Synchronous:

Kode yang dieksekusi secara sinkronous akan menjalankan pernyataan-pernyataan secara berurutan satu per satu, menunggu setiap pernyataan selesai sebelum melanjutkan ke pernyataan berikutnya. Ini berarti bahwa eksekusi kode terjadi dalam urutan yang ditulis, dan program akan berhenti dan menunggu operasi yang membutuhkan waktu lama (misalnya I/O, jaringan) selesai sebelum melanjutkan eksekusi selanjutnya. Ini adalah perilaku default dalam JavaScript.

Asynchronous:

Kode yang dieksekusi secara asynchronous tidak akan berhenti dan menunggu operasi yang membutuhkan waktu lama selesai. Sebaliknya, program akan melanjutkan eksekusi ke pernyataan berikutnya tanpa harus menunggu operasi tersebut selesai. Hasil dari operasi yang membutuhkan waktu lama akan di-handle melalui callback, Promise, atau menggunakan async/await. Ini memungkinkan kode lain untuk dieksekusi selama operasi yang membutuhkan waktu lama berjalan

1. Apa itu "generator" dalam JavaScript? Jelaskan bagaimana cara kerja generator.

"Generator" dalam JavaScript adalah fungsi yang dapat dihentikan dan dilanjutkan kembali (disuspen) saat dieksekusi. Saat fungsi generator dipanggil, itu tidak akan menjalankan kode secara penuh, melainkan menghasilkan objek yang dapat menghasilkan nilai dalam urutan yang dapat dikendalikan. Setiap kali generator dihentikan, nilai yang dihasilkan dapat diambil, dan generator dapat dilanjutkan lagi untuk menghasilkan nilai berikutnya. Generators menggunakan sintaksis khusus dengan kata kunci function\* (function asterisk).

1. Jelaskan perbedaan antara "Promise" dan "async/await" dalam JavaScript.

Promise:

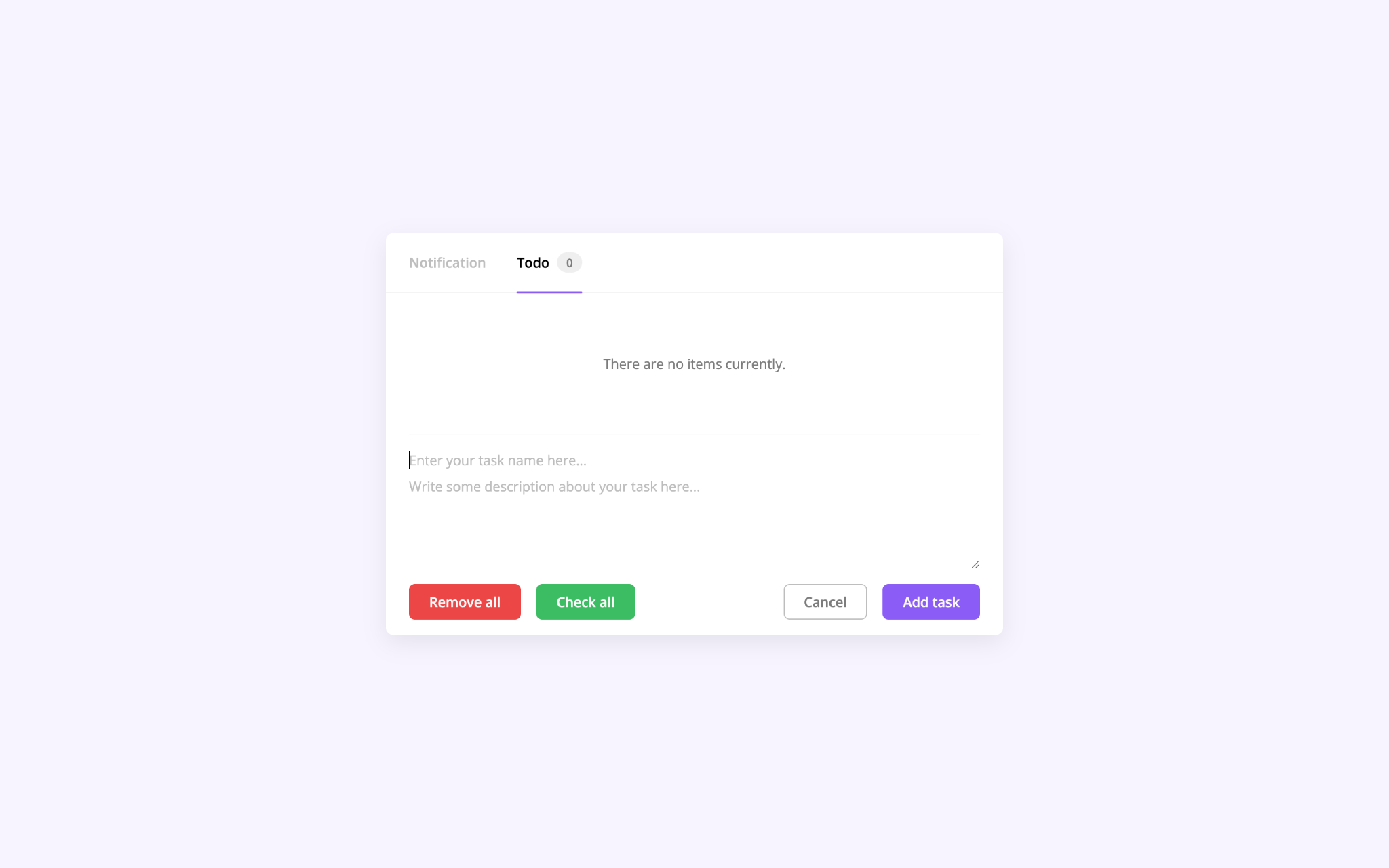
Promise adalah konstruksi JavaScript yang digunakan untuk mengelola operasi asinkron. Ini mengatasi masalah callback hell (penyusunan callback yang kompleks) dengan menggunakan metode then() untuk menangani hasil sukses dan metode catch() untuk menangani kesalahan. Promise dapat berada dalam tiga keadaan: pending (tunda), fulfilled (berhasil), atau rejected (gagal). Promise juga dapat digabungkan menggunakan Promise.all() atau Promise.race().

Async/await:

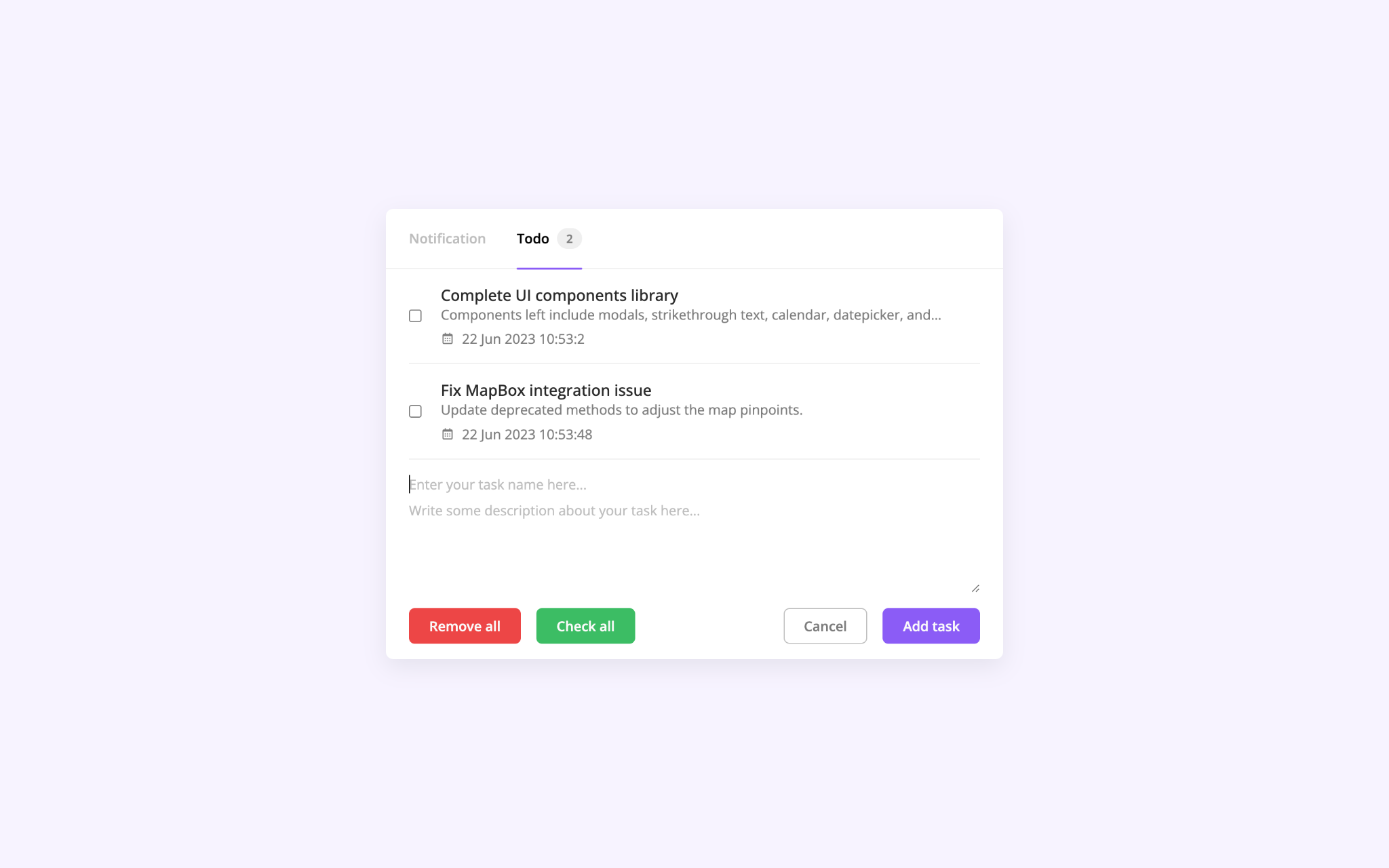
Async/await adalah fitur yang ditambahkan pada ES2017 (ES8) yang berfungsi untuk mengelola aliran kode asinkron dengan menggunakan sintaksis yang lebih mirip dengan kode synchronous. Keyword async digunakan untuk mendeklarasikan fungsi asynchronous, dan await digunakan dalam fungsi async untuk menunggu hasil Promise.

### Soal Studi Kasus

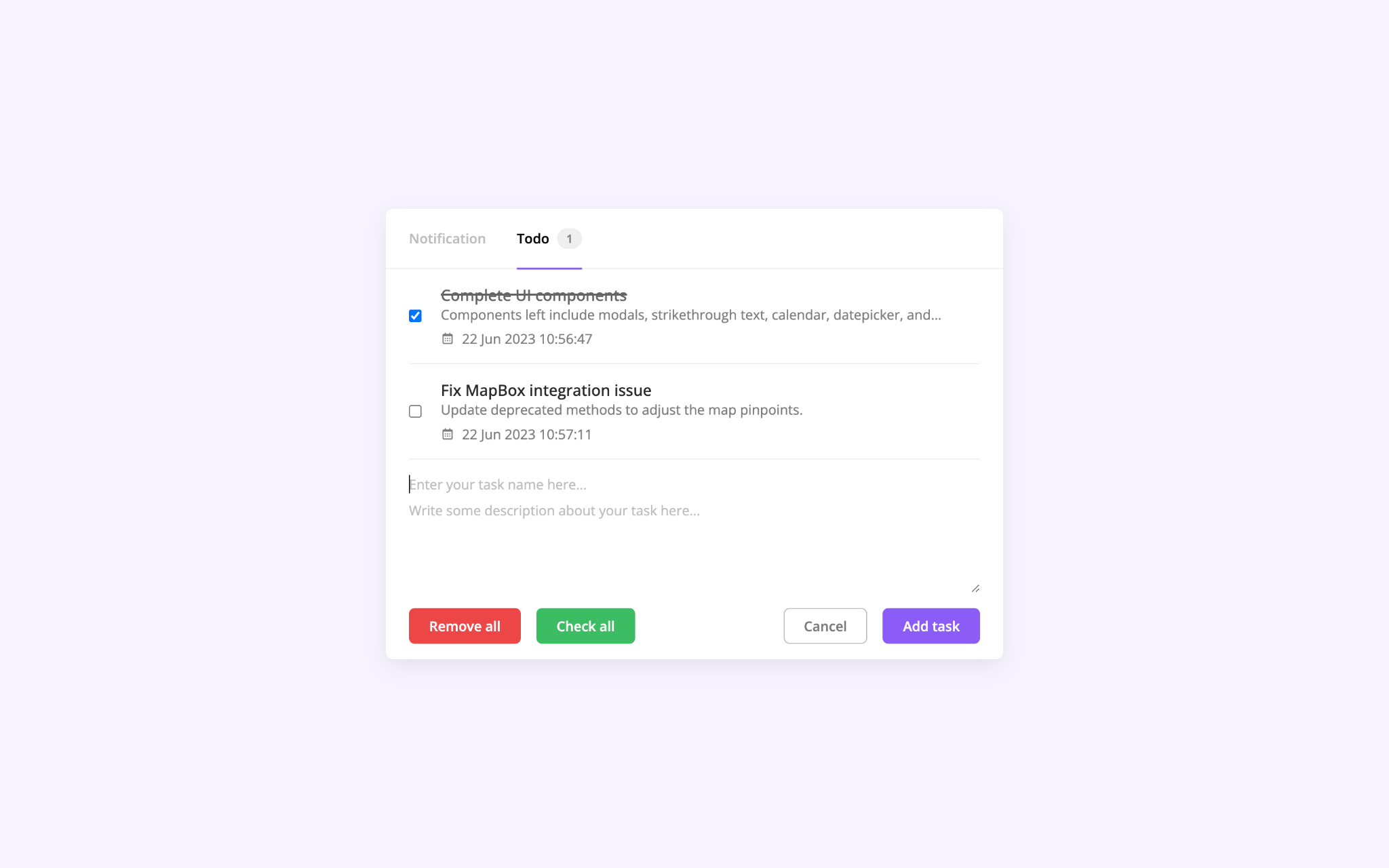
1. Buatlah sebuah form input yang memungkinkan karyawan untuk memasukkan tugas baru.



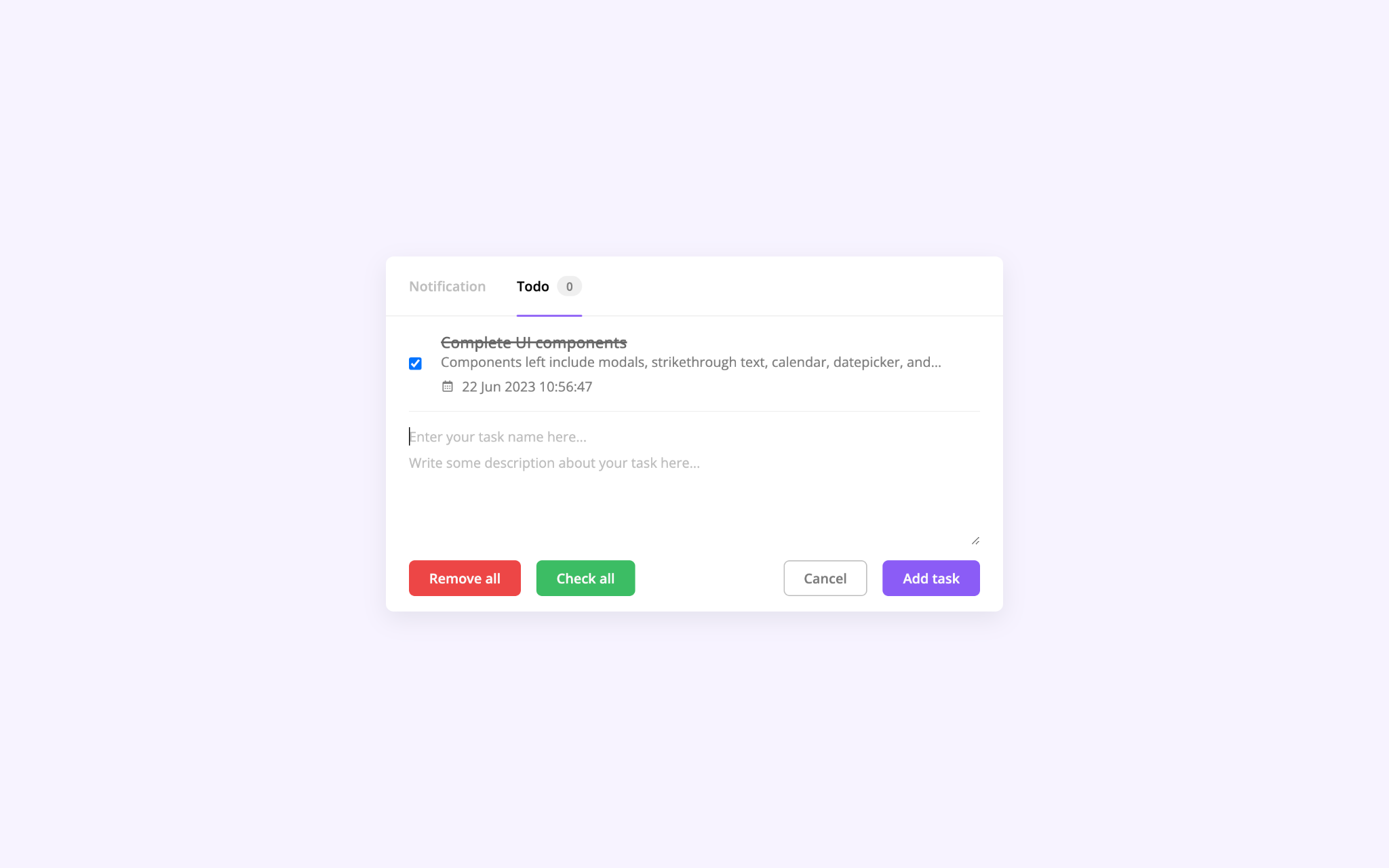
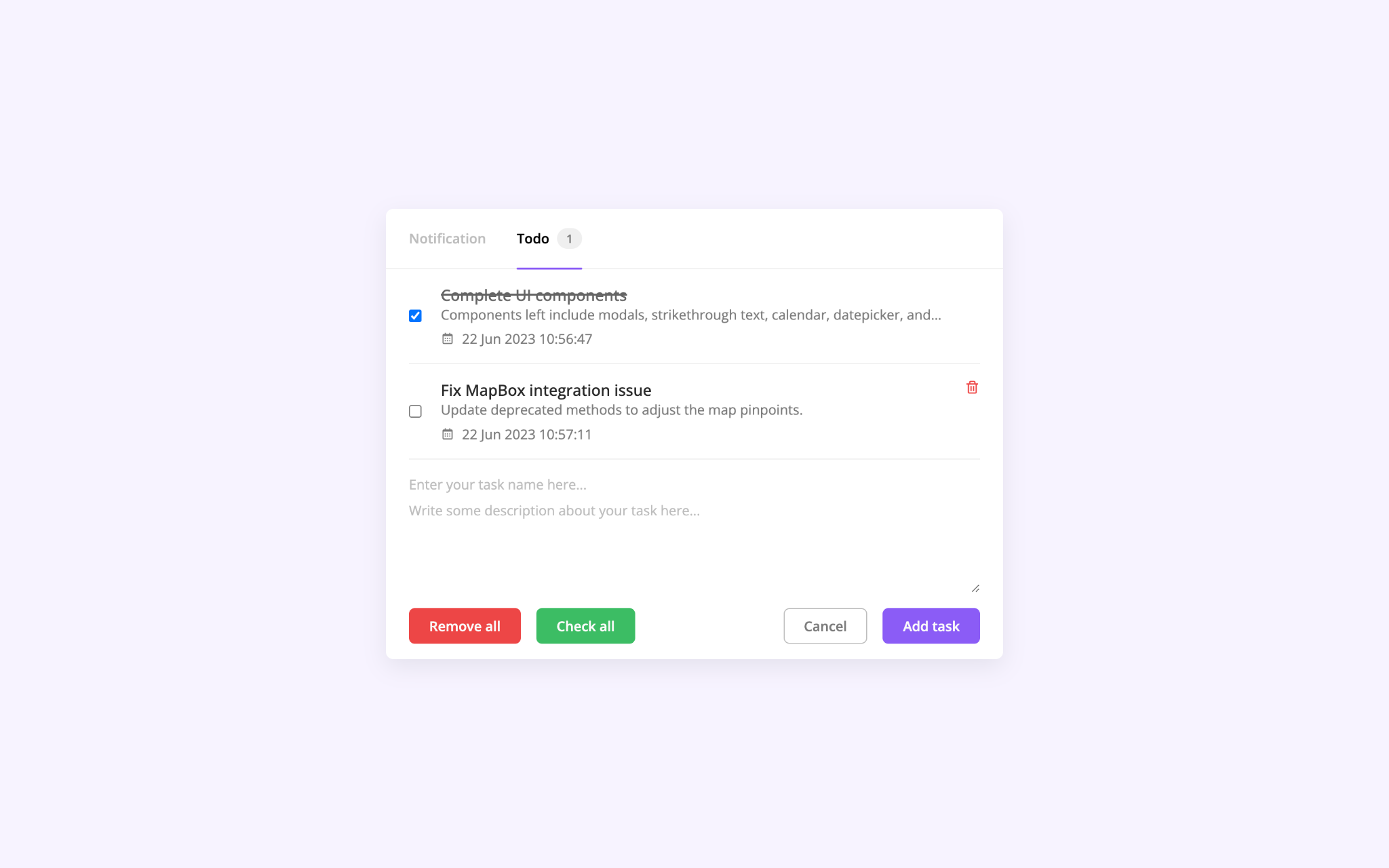
1. Tampilkan daftar tugas yang sudah ditambahkan dengan menggunakan elemen HTML yang sesuai.



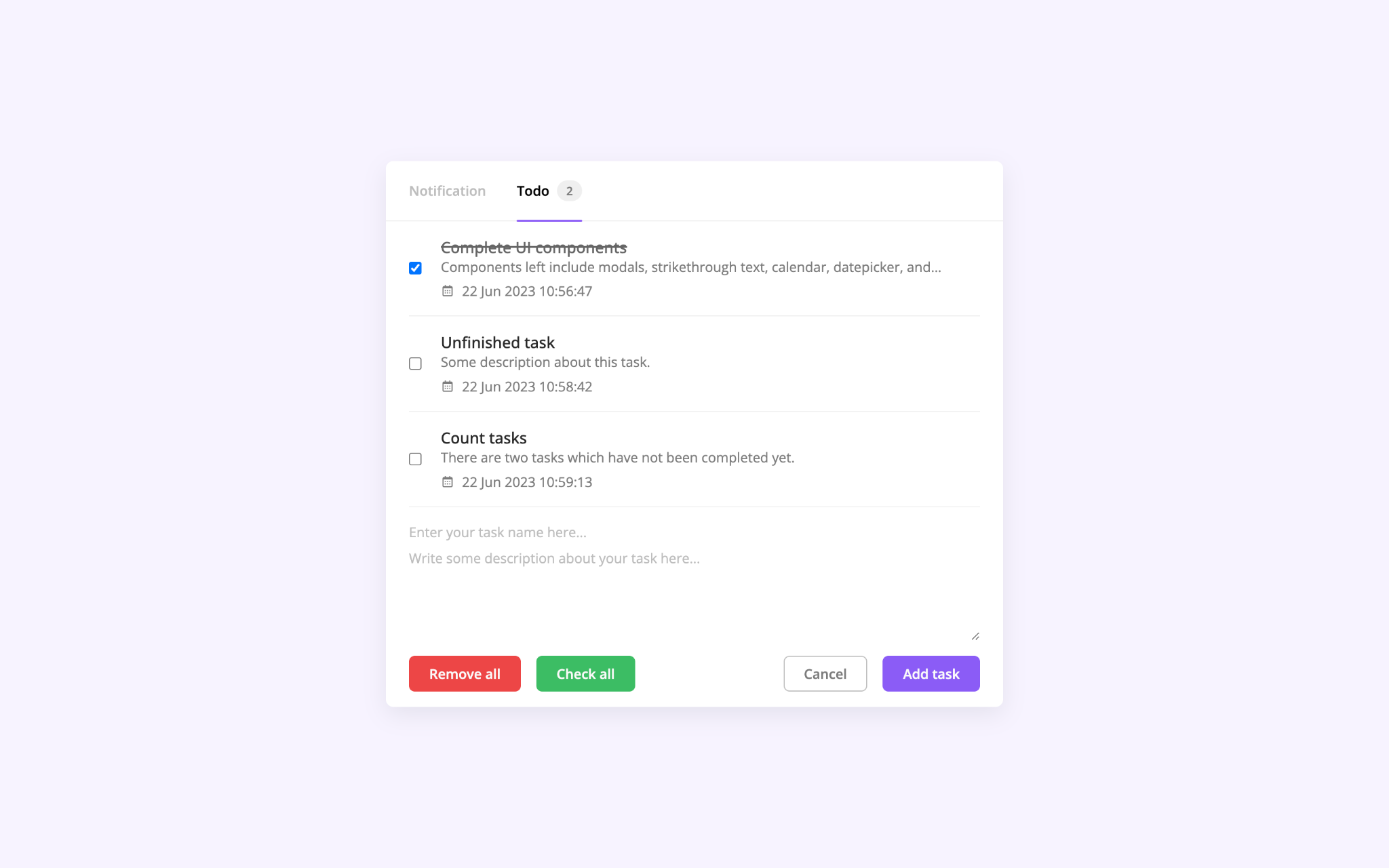
1. Berikan opsi untuk menandai tugas sebagai selesai atau belum selesai.



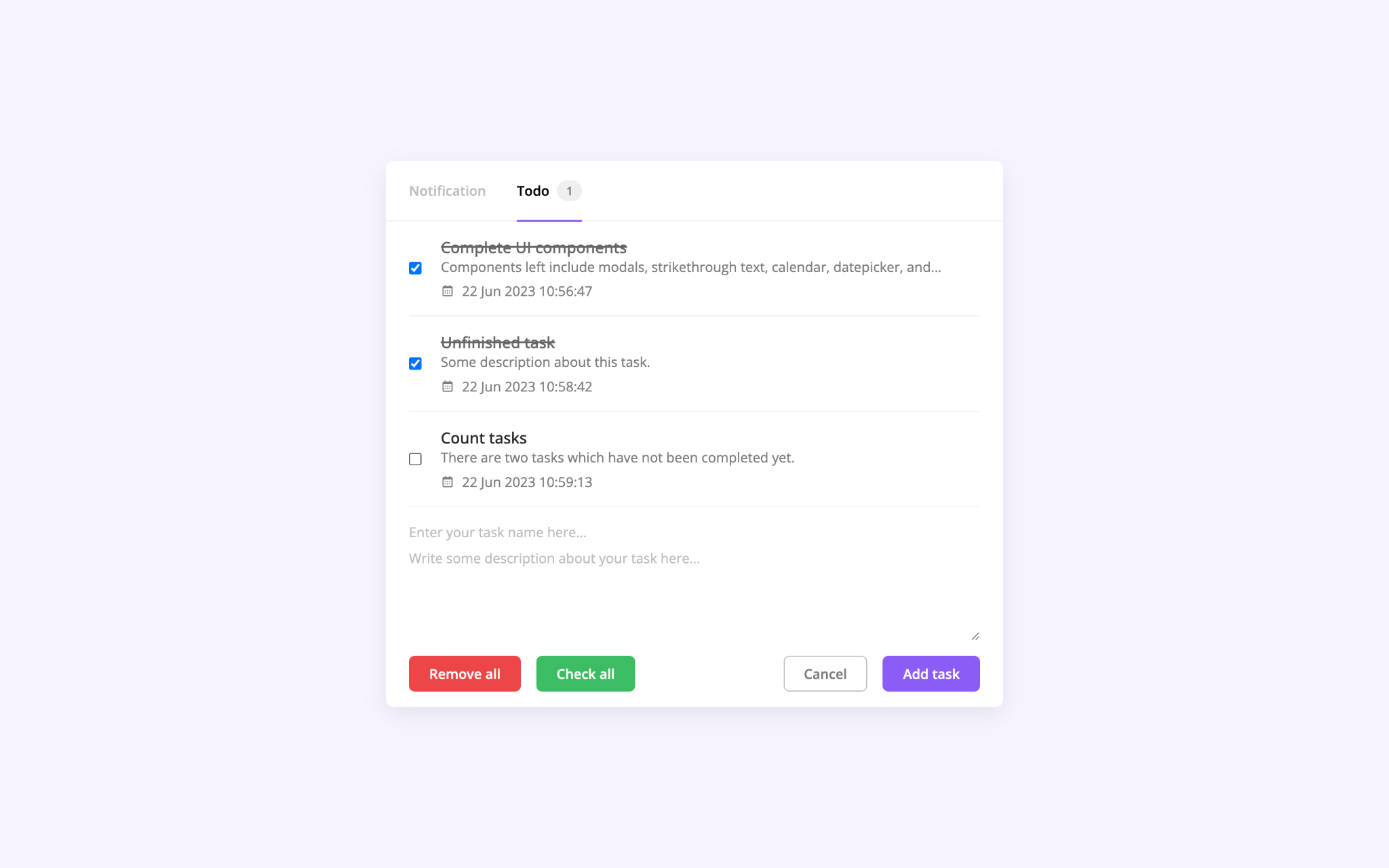
1. Tambahkan fitur penghapusan tugas dari daftar.

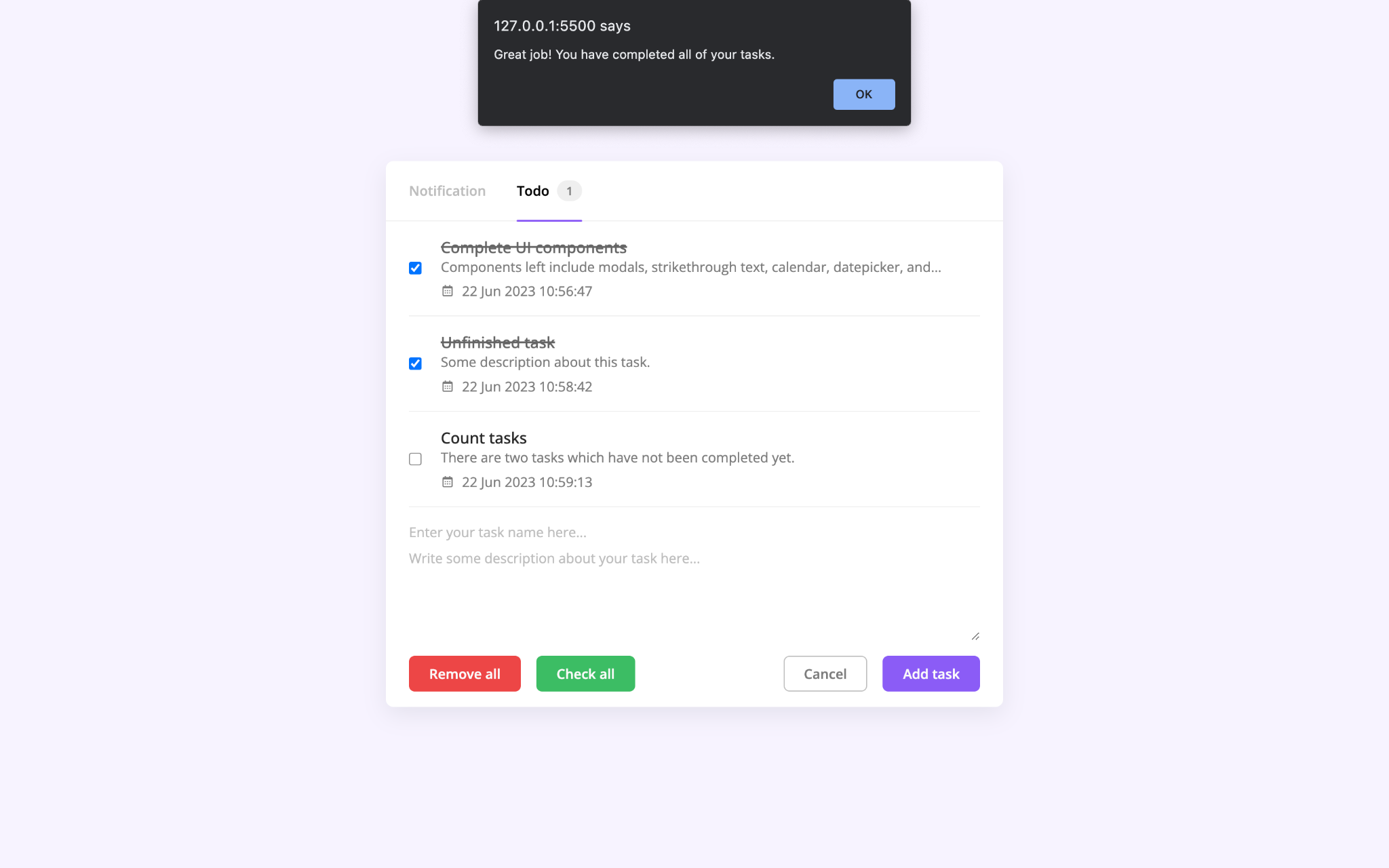


1. Tampilkan jumlah total tugas yang belum selesai.



1. Tampilkan pesan atau notifikasi jika karyawan telah menyelesaikan semua tugas.





1. Buatlah tombol untuk menghapus semua tugas sekaligus.

