

# Secret Santa Tracker

## Obiettivo

Progettare e sviluppare un'applicazione web dinamica basata su API e che utilizzi il database per la persistenza dei dati.

## Argomenti da ripassare

- Concetto di Rest API
- Operazioni CRUD sulle risorse
- Database relazionali
- Entità e relazioni
- Form e validazione dati

## Consegna

Nome repo:

- **secret-santa-wishlist-frontend**
- **secret-santa-wishlist-backend**

Immagina un'applicazione pensata per organizzare il Secret Santa tra amici, colleghi o familiari: ogni partecipante può creare la propria wishlist natalizia, aggiungendo regali con nome, immagine, link, prezzo e priorità.

Gli altri partecipanti, accedendo tramite un link segreto e univoco, possono consultare la lista e prenotare un regalo da acquistare, senza che il proprietario venga a sapere chi lo ha scelto.

In questo modo si evitano doppioni e si preserva l'effetto sorpresa tipico del Secret Santa!

</aside>

## Requisiti

### 1. Wishlist pubblica e gestione regali

- L'utente può creare una wishlist personale e aggiungere, modificare, eliminare e visualizzare i regali finché la lista non viene pubblicata.
- Ogni regalo ha: nome, link, prezzo, priorità (1-5), note opzionali.

### 2. Pubblicazione e generazione link segreto

- Solo dopo la pubblicazione, la wishlist diventa pubblica e viene generato un link segreto e univoco (es. UUID o token randomico) da condividere.
- Prima della pubblicazione, la wishlist è modificabile liberamente dall'utente.

### 3. Accesso tramite link pubblico

- Chi riceve il link può visualizzare la wishlist in sola lettura e prenotare/acquistare un regalo.
- Il proprietario della wishlist non può vedere chi ha prenotato o acquistato i regali.

### 4. Gestione prenotazioni

- I regali presenti nella wishlist pubblicata possono essere prenotati da chiunque abbia il link segreto.
- Al momento della prenotazione, il visitatore può aggiungere un messaggio pubblico che sarà visibile nell'interfaccia della lista (ad esempio "Babbo Natale è passato di qui!").
- Il proprietario della wishlist non potrà vedere l'identità di chi ha prenotato il regalo, ma solo il messaggio associato.

### 1. Salvataggio bozza wishlist

- Consenti all'utente di salvare la wishlist in fase di compilazione come bozza nel browser tramite localStorage.
- In questo modo, la lista potrà essere recuperata e modificata anche dopo un aggiornamento o la chiusura accidentale della pagina, fino alla pubblicazione definitiva.

## UX/UI


L'interfaccia è progettata mobile first, ottimizzata per l'uso e la condivisione da smartphone e tablet, ma adattabile anche a desktop.


- **Homepage:** schermata iniziale dell'applicazione
  - visualizza gli oggetti della wishlist attualmente in modifica
  - permette di accedere al form per aggiungere un nuovo regalo
  - include un pulsante per pubblicare la wishlist
- **Pagina di aggiunta regalo:**
  - consente l'inserimento di un nuovo regalo nella wishlist
  - campi obbligatori: nome, immagine, link, prezzo, priorità
- **Pagina della wishlist pubblica:**
  - mostra l'elenco dei regali disponibili nella lista
  - ogni regalo può essere prenotato tramite apposito pulsante
  - opzionalmente, chi prenota può lasciare un messaggio (ad esempio tramite modale o campo a comparsa)
  - è consigliato richiedere una conferma all'utente prima di completare la prenotazione

## Opzioni di consegna


1. Solo Frontend + API Rest in Express
2. Frontend + Backoffice + Api REST
  - Sviluppa anche una semplice area utente per gestire e visualizzare le proprie wishlist

## Bonus

 **Bonus 1 - Liste salvate:** Crea una pagina dedicata dove l'utente può vedere sia le proprie wishlist pubblicate\*\* che quelle di altri che ha salvato come preferite (ad esempio tramite un pulsante "Salva tra i preferiti" presente nella pagina pubblica di una wishlist). Da questa pagina l'utente può accedere rapidamente alle wishlist preferite, rimuoverle dai preferiti o visualizzarle.

 **Bonus 2 - Condivisione:** Implementa una funzionalità che permetta all'utente di condividere facilmente la propria wishlist pubblicata tramite la [Web Share API](#) del browser. Aggiungi un pulsante "Condividi" nella pagina della wishlist pubblica che, se il browser lo supporta, apre il

pannello di condivisione nativo (es. per inviare il link via WhatsApp, Telegram, email, ecc.). In caso di browser non supportato, mostra un messaggio o copia il link negli appunti.

 **Bonus 3 - Suggerimento regalo:** Durante la compilazione della wishlist, aggiungi un pulsante “Suggerisci un regalo” che apre una modale con un regalo selezionato casualmente tra tutti quelli presenti nel database delle wishlist. L’utente può scegliere se ignorare il suggerimento oppure aggiungerlo direttamente alla propria lista.

### **Note Importanti**

- Progetta lo schema del database e le API prima di iniziare
- Gestisci la validazione sia lato client che server
- Implementa gestione degli errori appropriata e casi limite
- Includi il file [README.md](#) in cui:
  - Descrivi la struttura del tuo progetto
  - Spiega le scelte implementative
  - Documenta eventuali bonus realizzati

## Reference e Documentazione

- [Web Share API – MDN](#)