

Magazzino Fermi

Gestionale componenti per laboratori scolastici

Alessandro Veliko · Daniel Starcevic
Committente: Matia Florini

Una web app unica per consultare componenti, scorte, esperienze e acquisti.

Browser + mobile

Python Tornado

SQLite

Docker

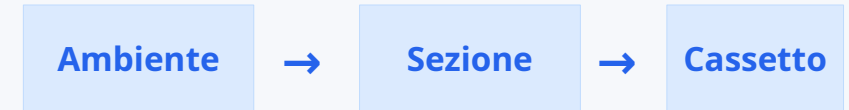


Richiesta del committente

Da magazzino diffuso a sistema consultabile in modo rapido

Il committente ha richiesto un gestionale per minuteria, componentistica e reagenti dell'ITIS E. Fermi di Modena. Il problema principale è che il materiale non si trova in un unico punto: può essere in armadi, scaffali, cassette o ambienti diversi. Il software deve quindi rendere immediata la domanda: "che cosa abbiamo, dove si trova e cosa manca?".

L'obiettivo pratico è semplificare la progettazione delle attività di laboratorio e la verifica degli acquisti necessari. Per questo il progetto è stato pensato come applicazione web responsive, accessibile sia da browser desktop sia da dispositivi mobili.



La posizione fisica è descritta con il sistema ASC: Ambiente, Sezione e Cassetto. In questo modo anche un magazzino "diffuso" può essere rappresentato come un archivio ordinato e ricercabile.

Obiettivi del progetto

Requisiti trasformati in funzioni verificabili durante la demo

Abbiamo organizzato il lavoro partendo dai bisogni reali del laboratorio: catalogare i componenti, controllare le quantità, collegare i materiali alle esperienze e differenziare le operazioni in base al ruolo dell'utente. In questo modo il progetto non è solo una pagina di consultazione, ma un piccolo sistema gestionale.

Durante il collaudo si possono verificare gli obiettivi principali: ricerca dei componenti, visualizzazione della posizione ASC, gestione delle scorte, creazione delle esperienze e generazione della lista acquisti.

Catalogo

componenti classificati e cercabili

Scorte

quantità, soglia minima e acquisti

Esperienze

materiali richiesti e PDF collegato

Ruoli

permessi diversi per ogni utente

Log

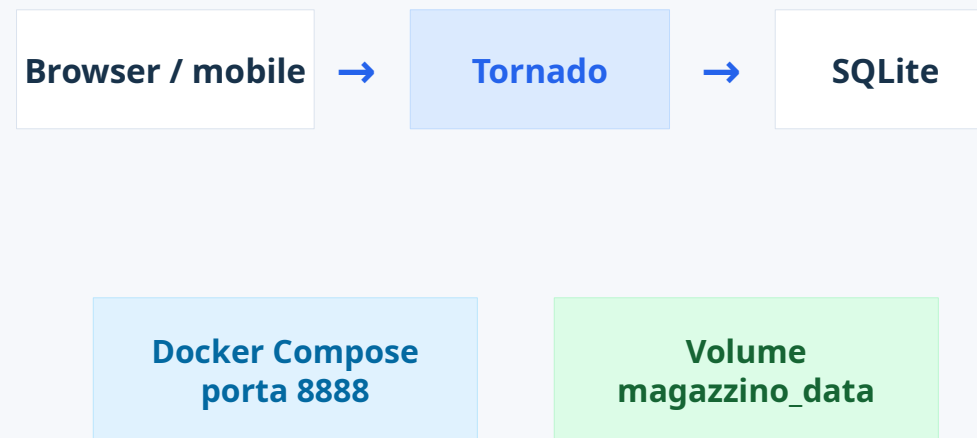
tracciabilità delle operazioni

Tecnologie utilizzate

Una web app leggera, semplice da avviare e adatta al contesto scolastico

Il backend è scritto in Python usando Tornado, un framework web che gestisce le richieste HTTP e genera le pagine dell'applicazione. Il frontend è composto da HTML, CSS e JavaScript: HTML per la struttura, CSS per l'interfaccia responsive e JavaScript per rendere più dinamica la gestione delle esperienze.

Come database è stato scelto SQLite, perché per un progetto locale/scolastico permette di salvare dati reali senza configurare un server database separato. Docker e Docker Compose rendono l'avvio più ripetibile e mantengono il database persistente tramite volume.



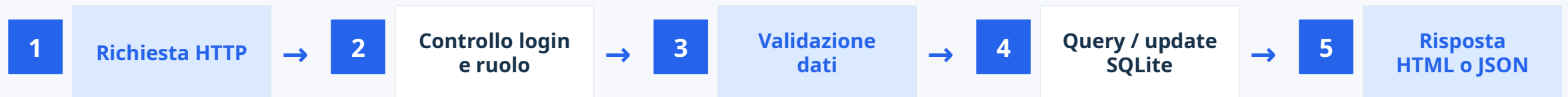
Lo stack è volutamente essenziale: meno componenti da configurare significa più semplicità nel collaudo e nella manutenzione.

Flusso dei dati



Ogni operazione passa da richiesta, controllo, database e risposta

Il flusso dati è sempre lo stesso: l'utente compie un'azione dall'interfaccia, il browser invia una richiesta HTTP al server Tornado, il server controlla sessione e permessi, valida i dati ricevuti e poi legge o aggiorna SQLite. Infine viene restituita una pagina HTML oppure una risposta JSON.



Esempio: nel carico/scarico il tecnico seleziona il componente e inserisce una quantità. Il server verifica che l'utente abbia il ruolo corretto, impedisce scarichi negativi, aggiorna la quantità e salva anche il movimento e il log attività.

Lo stesso schema vale per login, ricerca componenti, creazione delle esperienze e generazione della lista acquisti. Cambia l'operazione, ma il percorso dei dati rimane controllato e tracciato.

Database SQLite

Le tabelle rappresentano utenti, componenti, locazioni, esperienze e attività

Il database non contiene solo un elenco di componenti: collega tra loro utenti, ruoli, posizioni fisiche, tag di caratteristiche, esperienze di laboratorio e movimenti di magazzino. Questa struttura permette ricerche più precise e rende possibile ricostruire cosa è stato fatto nel sistema.

La locazione è divisa in laboratori, armadi e cassetti, così il sistema ASC diventa parte del modello dati. I componenti possono inoltre essere collegati a più tag e a più esperienze, perché lo stesso materiale può servire in attività diverse.



utenti

ruolo · approvato
password_hash

componenti

classificazione 4 livelli
quantità · scorta

tag

caratteristiche

ASC

laboratori → armadi →
cassetti

esperienze

pdf_url + materiali

movimenti / log

tracciabilità

Ruoli e sicurezza

Permessi differenziati e controllati lato server

La gestione degli accessi serve a evitare modifiche non autorizzate al magazzino. L'amministratore ha il controllo completo, il tecnico gestisce componenti e approvazioni operative, mentre il professore consulta il catalogo e prepara le esperienze.

La sicurezza non dipende solo dai pulsanti mostrati nell'interfaccia: ogni richiesta viene verificata dal server. Prima delle operazioni sensibili viene controllata la sessione dell'utente e il ruolo necessario per eseguire quell'azione.



Amministratore

gestione completa
utenti e log

Tecnico

componenti
approva professori

Professore

consulta catalogo
crea esperienze

PBKDF2
hash + salt

cookie sicuro
+ controlli ruolo

Registrazione e invio credenziali

Login, registrazione e invio credenziali via Brevo

All'apertura del sito l'utente può scegliere se accedere o registrarsi. In registrazione inserisce email, username e ruolo richiesto, scegliendo tra tecnico e professore. Il ruolo di amministratore è riservato all'indirizzo autorizzato `daniel.starcevic@fermi.mo.it`.

Il server genera una password casuale, prepara l'email con le credenziali e la invia tramite Brevo dal mittente `lavorosshvs@gmail.com`. La API key resta nella configurazione del server: non viene mostrata nel browser e non viene richiesta agli utenti.

Solo se Brevo conferma l'invio, l'account viene salvato nel database. Nei login successivi l'utente entra con username o email e password ricevuta; il sistema verifica le credenziali e controlla che l'account sia approvato prima di concedere l'accesso.

Funzionamento dell'interfaccia

Dashboard, ricerche e consultazione rapida delle informazioni utili

Dopo il login l'utente entra in una dashboard che riassume lo stato del magazzino: numero di componenti, locazioni, esperienze, tag e voci sotto scorta. Questa schermata serve come punto di partenza per capire subito se ci sono criticità.

La ricerca è stata pensata per rispondere a domande reali: posso cercare un componente per nome, filtrarlo per classificazione, per tag, per esperienza collegata o visualizzare soltanto ciò che è sotto scorta. I risultati mostrano quantità, posizione ASC, tag e datasheet.

nome

classificazione

tag

esperienza

sotto scorta

Output mostrato: quantità disponibile, unità di misura, posizione ASC, scorta minima, tag assegnati e collegamento al datasheet.

Gestione componenti e magazzino

Ogni componente ha identità, posizione e stato di disponibilità



L'inserimento di un componente richiede informazioni complete: nome, descrizione, classificazione a quattro livelli, datasheet, quantità, unità di misura, scorta minima, locazione ASC e almeno dieci tag di caratteristiche. Questo obbliga a creare schede utili, non dati incompleti.

Admin e tecnici possono registrare movimenti di carico e scarico. Ogni movimento aggiorna la quantità del componente e viene salvato nel database, così è possibile controllare le modifiche nel tempo. Il sistema impedisce uno scarico maggiore della quantità disponibile.

Identità
nome · descrizione · datasheet

Classificazione
famiglia · tipo · ambito · categoria

Locazione
Ambiente · Sezione · Cassetto

Stato
quantità · scorta minima · sotto scorta

carico +



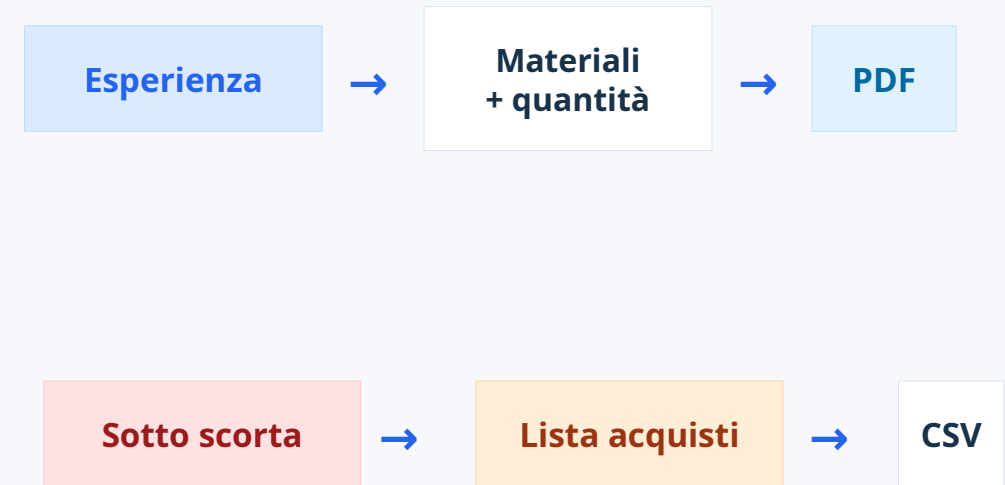
quantità aggiornata

Esperienze e lista acquisti

Dai materiali di laboratorio alla verifica automatica delle scorte

Le esperienze di laboratorio permettono di raggruppare i componenti necessari per una certa attività. Ogni esperienza ha nome, descrizione, materiali richiesti e quantità. È stata aggiunta anche la possibilità di collegare un PDF con la spiegazione dell'esperienza, così la scheda diventa più completa.

La lista acquisti nasce dal confronto tra quantità disponibile e quantità minima di sicurezza. In questo modo il sistema aiuta a capire quali materiali devono essere reintegrati. L'esportazione CSV consente poi di ottenere un file riutilizzabile fuori dall'applicazione.



Modifiche successive: link PDF e cancellazione multipla delle esperienze.

Avvio con Docker e collaudo

Ambiente ripetibile e test semplici da verificare



Docker è stato aggiunto per rendere l'avvio più controllato: il Dockerfile costruisce l'immagine Python dell'applicazione, mentre compose.yaml espone la porta 8888, crea la rete e collega un volume per conservare il database SQLite anche quando il container viene fermato o ricreato.

Il collaudo non consiste solo nel vedere che la pagina si apre: bisogna provare login, ricerca, inserimento o modifica dei componenti, carico/scarico, creazione esperienza con PDF, lista acquisti, esportazione CSV e verifica dei log.

1 `docker compose up --build`

2 `localhost:8888`

3 `login demo`

4 `test funzioni`

5 `controllo log`

Demo finale e conclusione

Scaletta per rimanere nei tempi e chiudere sui requisiti soddisfatti

La demo deve essere breve e guidata: prima si accede con un utente demo, poi si mostra una ricerca componente, la posizione ASC, un movimento di carico/scarico, la creazione o consultazione di un'esperienza con PDF, la lista acquisti e infine i log.

In conclusione, il progetto risponde alla richiesta perché centralizza componenti, utenti, scorte, esperienze e acquisti in una sola applicazione web responsive. La parte più importante da evidenziare è che ogni funzione ha un'utilità pratica nel lavoro quotidiano del laboratorio.

1

Login

2

Ricerca

3

ASC + scorte

4

Esperienza + PDF

5

CSV + log

Domande