

Report Progetto Fladale

Leonardo Brunetti — Flavia Feliciani — Davide Nardoni

brunetti.2046981@studenti.uniroma1.it

feliciani.2055134@studenti.uniroma1.it

nardoni.2060903@studenti.uniroma1.it

Backend Node.js: la logica server-side è gestita da `server.js` con Express. Il file `api.js` definisce rotte REST (GET e POST) per piante, tags, utenti (inclusi profilo, login, registrazione, attivazione e password dimenticata) e preferiti. Le connessioni al DB MariaDB sono gestite tramite `db.js`. Il file `fladale.sql` contiene la definizione delle tabelle relazionali (piante, utenti, tag). Il server espone API asincrone e gestisce la comunicazione con il frontend.

Home: la pagina è principalmente un carosello di video gestito con `hp_scripts.js` e immagini associati ognuno ad una pagina del sito, la top-section come in tutte le altre pagine è dinamica, dipende da l'accesso o no dell'utente, che vede. Accedi per `accesso_index.html` o l'icona del profilo per `profilo_index.html`.

Login: la pagina è il frontend della rotta POST che controlla le credenziali inserite dall'utente, sia che coincidano con quelle nel database sia che il profilo sia stato attivato nel momento della registrazione con il token di `attivazione_index.html`, se l'utente non è ancora registrato può reindirizzarsi a `registrazione_index.html`.

Password dimenticata: la pagina prende in input l'email dell'utente e invia una email che tramite token, fa accedere direttamente alla pagina `profilo_index.html`.

Profilo: la pagina riceve con una rotta GET le informazioni dell'utente e le inserisce in un form che non ha abilitata la modifica, ma con dei bottoni ogni campo può essere modificato, confermando le modifiche, si attiva la POST al database `utente`, tutte le pagine del sito hanno la top-section dinamica che mostra questa pagina o `accesso_index.html`, l'avvenuto accesso viene controllato attraverso i cookie.

Registrazione: la pagina prende le informazioni dell'utente in un form, ne controlla la formattazione e con una rotta POST salva il nuovo utente, con un token poi invia un email di attivazione all'utente, nel caso l'utente sia già registrato e abbia fatto l'attivazione può reindirizzarsi a `accesso_index.html`.

Attivazione: la pagina che si apre dall'email che l'utente riceve quando attiva l'account per la prima volta, il file `attivazione_scripts.js` gestisce la creazione del token individuale e il controllo della validità di esso, se l'operazione va a buon fine l'utente attende 5 secondi e viene reindirizzato a `accesso_index.html`.

Scopri: la pagina mostra tutte le piante nel database `piante`, ottenute in modo dinamico con una rotta GET, che attraverso i `tags` può essere modificata, in più per ogni pianta c'è un logo per inserirla con una POST nel database `preferiti`, ogni pianta se selezionata reindirizza alla propria pagina `dettaglio_index.html`.

Dettaglio: la pagina è completamente dinamica e si basa su un formato generico per tutte le pagine, che poi viene modificato con le caratteristiche di ogni pianta, ottenute con la rotta GET dal database, la sezione principale contiene la foto della pianta e le informazioni della pianta per alcuni campi in particolare, la seconda parte contiene una breve descrizione e una foto del luogo di origine che vengono ottenuti in maniera dinamica dal file `descrizioni.js`.

Consigli: la pagina è statica se non per la lista di piante per ogni camera, che viene gestita con una rotta GET per il database delle piante con il tag di ogni stanza, in fondo alla pagina c'è il reindirizzamento a `consigli_form.html`.

Form: la pagina contiene un form a tendina con i campi utilizzati per la descrizione della pianta nella pagina `dettaglio_index.html`, ma possono anche non essere selezionati, compilandolo l'utente riceverà un consiglio personalizzato di una pianta nello specifico, verrà quindi reindirizzato alla `dettaglio_index.html` di questa pianta.

Giardino: la pagina si suddivide nella parte superiore, che attraverso una rotta GET accede al database `preferiti` dell'utente e gestisce con una POST la rimozione e anche la modifica della data dell'ultimo annaffiamento, la parte inferiore è dedicata ad un calendario con gli eventi dell'annaffiamento, creati dinamicamente, affianco al calendario c'è una sezione dedicata al salvataggio dei prossimi eventi per ogni pianta nel calendario google dell'utente.

Storia: la pagina è statica e ha solo due paragrafi che spiegano l'idea del sito e l'importanza delle piante per l'ambiente e per le persone, in fondo c'è un footer con i nostri nomi.