

Prova tecnica Angular + Node.js

Descrizione

È richiesta la realizzazione di un pannello di amministrazione per gestire il backoffice di un determinato software immaginario.

Il front-end deve essere realizzato in Angular, prestando attenzione alla qualità estetica del prodotto. Non è richiesto l'utilizzo di una specifica libreria di componenti, la scelta è lasciata al candidato purché venga rispettato uno stile grafico moderno e adeguato al prodotto.

La web app dovrà seguire il paradigma mobile-first, per essere quindi responsive ed utilizzabile sia da smartphone che da desktop.

Il back-end deve essere realizzato in Node.js, utilizzando framework come Express o Nest.js, seguendo un'architettura REST. Gli endpoint dovranno utilizzare i metodi corretti e tutti quelli che necessitano di autenticazione dovranno verificarla tramite un middleware.

Il database è lasciato a libera scelta del candidato, che può valutare quale soluzione ritiene più adatta a soddisfare le esigenze del prodotto. Un modello relazionale è preferibile, ma è possibile procedere anche utilizzando database documentali o non relazionali.

Se necessario, il candidato fornirà il DDL del database per consentirne la verifica di corretta modellazione.

Ogni informazione necessaria alla preparazione del sistema per consentire un corretto avvio della prova in fase di verifica sarà posta dal candidato all'interno del file README.md.

In tutte le richieste successivamente descritte saranno sempre considerati impliciti i seguenti punti:

- i campi di un form necessitano delle corrette validazioni;
- le casistiche di errore vanno sempre gestite;
- l'utente deve sempre essere a conoscenza dell'esito di un'operazione che eseguito, sia esso positivo o negativo;
- le operazioni distruttive dovranno essere sempre protette da un popup di conferma.

Si prega di notare che la seguente consegna rappresenta i requisiti minimi della richiesta: una volta completati i punti richiesti, il candidato è libero di aggiungere funzionalità se lo desidera, purché inerenti con il resto dell'applicativo. Tutti i punti etichettati come **Nice to have** sono da considerarsi come suggerimenti da implementare una volta terminati i punti minimi.

La navigazione tra le pagine potrà essere implementata tramite navbar laterale o superiore, in ogni caso dovrà essere data evidenza della pagina attualmente attiva e dovrà essere presente una call to action per effettuare il logout.

Autenticazione

L'applicativo dovrà presentarsi all'utente con un form di login (email e password). A meno che l'accesso non venga effettuato correttamente, l'utente non potrà accedere a nessun'altra pagina.

Il login effettuato imposterà un cookie così da ricordare l'utente anche in caso di refresh della pagina.



Se l'utente non è in possesso di credenziali di accesso potrà, tramite apposito link, accedere ad un modulo di registrazione, nel quale sarà richiesto l'indirizzo email, la password, e la relativa conferma. A seguito di compilazione corretta, sarà reindirizzato al modulo di login per effettuare l'accesso con le credenziali appena inserite.

Nice to have: impostare un sistema di valutazione della sicurezza della password scelta.

Dashboard

A seguito del login l'utente atterrà sulla dashboard dove visualizzerà un messaggio di benvenuto oltre che dati e grafici riguardanti le altre sezioni del pannello di amministrazione. La scelta di quali dati mostrare viene lasciata al candidato, che dovrà considerare di mostrare almeno un'informazione per ogni sezione. A titolo esemplificativo, le informazioni relative alla sezione "Gestione utenti" possono essere il numero totale di utenti ed un grafico che mostra il rapporto tra gli utenti attivi e quelli non attivi.

La dashboard potrà essere organizzata in griglia o in "masonry" se visualizzata da desktop, viceversa sarà una lista verticale se visualizzata da mobile.

Nice to have: implementare la possibilità per l'utente di riorganizzare le posizioni dei widget con un sistema a trascinamento.

Sezione "Gestione utenti"

In questa sezione verrà visualizzata la lista di utenti registrati al pannello di amministrazione, in formato tabellare, che mostra per ognuno l'indirizzo email, la data di registrazione e quella di ultimo accesso, oltre che il suo stato ("Attivo" o "Non attivo").

Per ogni utente sarà presente una call to action per cambiarne lo stato, a seguito di popup di conferma. Un utente in stato "Non attivo" non potrà effettuare l'accesso.

Non sarà possibile modificare lo stato di sé stessi.

Sarà presente una call to action per consentire la registrazione di un nuovo utente, che tramite un popup utilizzerà lo stesso modulo usato precedentemente per la registrazione.

Nice to have: implementare la possibilità di registrare un utente che dovrà obbligatoriamente cambiare la password nel momento del primo login.

Sezione "Articoli"

In questa sezione l'utente potrà visualizzare e pubblicare degli articoli che saranno ipoteticamente visualizzati sul sito principale (inesistente ai fini della prova).

Il listato degli articoli dovrà mostrarli in ordine decrescente di data creazione e al di sopra dovrà essere presente il modulo di inserimento di un nuovo articolo.

Un articolo sarà composto da un titolo e da un corpo, entrambi obbligatori, oltre che da un eventuale link.

La scrittura del corpo dovrà essere effettuata con un editor WYSIWYG che dovrà supportare solo funzionalità di formattazione di base (grassetto e corsivo, allineamento a sinistra e al centro, due dimensioni del testo, divisione in paragrafi). Un articolo dovrà anche contenere un flag che ne determina lo stato di pubblicazione.

Nice to have: implementare un sistema di generazione automatica di un articolo che, dato in input un URL, ne legga i tag Open Graph e costruisca autonomamente titolo e contenuto.

Sezione "Note"

In questa sezione l'utente potrà inserire delle note a testo libero.

Ogni nota sarà composta da un titolo (facoltativo) e da una descrizione che ne compone il corpo. Potrà inoltre esserle assegnato un colore, prelevato da una palette predefinita, che ne cambierà il colore di sfondo.

Il modulo di creazione e di modifica di una nota dovrà essere il medesimo componente.

Le note potranno anche essere eliminate.

Nice to have: implementare un sistema di condivisione delle note, così che più utenti possano avere accesso alla medesima nota.

Consegna

1. Consegnare il codice sorgente del progetto realizzato.
2. Porre eventuali note su file README.md.
3. Non porre eventuali directory contenenti librerie di terze parti (es: vendor, var, node_modules).
4. Eseguire un push di quanto realizzato sul repository GIT disponibile all'URL [https://git.exacloud.it/prove-di-ammissione/angular-**nome-candidato**.git](https://git.exacloud.it/prove-di-ammissione/angular-nome-candidato.git), dove **nome-candidato** è una stringa composta da nome e cognome del candidato, con "-" come carattere separatore (esempio: "paolo dall'aria" => <https://git.exacloud.it/prove-di-ammissione/angular-paolo-dall-aria.git>).
5. Le credenziali di accesso al server GIT verranno consegnate in separata mail.
6. Il candidato è invitato ad utilizzare GIT in modo quanto più corretto, evitando di effettuare un unico commit alla fine dei lavori, ma documentando ogni feature implementata in un commit separato.