

Diario di lavoro

Luogo	Trevano
Data	10.02.2020

Lavori svolti

Oggi ho continuato con il collegamento della pagina di login con quella successiva e sono riuscito a portarlo a termine. Per spostarsi tra le pagine avrei dovuto usare il comando `window.location = "{ url('route/pagina') }"` ma questo non consente di passare un header aggiuntivo e essendo che io utilizzo i JWT dovevo passare anche l'Authorization con il token. Ho quindi modificato leggermente il middleware così che ora accetta sia i token passati tramite header che quelli tramite URL:

[http://localhost/admin?token=\[token\]](http://localhost/admin?token=[token])

Questo mi consente così di utilizzare il metodo citato prima passando comunque il token, questa è la modifica che ho apportato al middleware:

```
//Prendo il token dagli header della richiesta
$token = $request->header('Authorization');

//Verifico che sia stato impostato
if(!$token){
    //Verifico se è nell'URL
    $token = $request->get('token');
}else{
    //Rimuovo la stringa 'Bearer' iniziale
    $token = explode(" ", $token)[1];
}

//Se il token non è settato ritorno l'errore
if(!$token) {
    return response()->json(['error' => 'Token non impostato'], 401);
}
...
```

Dopodiché ho sviluppato parte del frontend della pagina di amministrazione collegandola al backend così che ora quando ci si collega si possono visualizzare tutte le motivazioni e tutti gli utenti nel sistema in due tabelle distinte. Le tabelle utilizzano l'estensione di Material Design Bootstrap chiamata "Datatables" che consente di aggiungere delle operazioni sui dati della tabella come la ricerca e la suddivisione in più pagine. Per prendere i dati in JSON e trasformarli in una tabella ho scritto una funzione che esegue questa operazione e si trova nel file

`codice/ValutazioneLPI/public/resources/js/JSONToHTML.js.`

Infine ho sviluppato, solo nel backend, i metodi di eliminazione e modifica di un utente o di una motivazione e li ho testati con postman. I metodi di eliminazione sono piuttosto semplici e consentono, tramite un id passato, di trovare un utente e se esiste eliminarlo.

```
public function deleteUser($id, Request $request){
    //Verifico che l'utente sia un admin
    if($request->all()['id_role'] == 2){
        //Se è amministratore cerco l'utente e lo elimino
    }
}
```

```
User::findOrFail($id)->delete();  
return response('Deleted Successfully', 200);  
}else{  
    //Se non lo è ritorno l'errore  
    return response()->json(['Unauthorized' => 'Non hai i permessi  
                                necessari per accedere'], 401);  
}  
}
```

Mentre quelle di modifica sono più complesse dal momento che prima comprendono pure la validazione dei dati, per la modifica dell'utente ho deciso che l'email non sarà modificabile mentre la password potrà farlo l'utente dalla pagina apposita. Tutti gli altri campi saranno modificabili dall'admin.

Problemi riscontrati e soluzioni adottate

Oggi non ho avuto problemi.

Punto della situazione rispetto alla pianificazione

Sono in orario.

Programma di massima per la prossima giornata di lavoro

La prossima volta voglio continuare con la pagina di amministrazione collegando i metodi di modifica e eliminazione al frontend e se ho tempo vorrei creare quelli di aggiunta.