



## **Travail pratique 2**

### **Rapport**

**Par :**

**Charles Ricciardi (537 181 593)**

**Dans le cadre des cours GLO-3202 – Sécurité des applications Web**

**Université Laval**

# Sommaire :

<b>Implémentation de l'authentification:</b>	<b>3</b>
Inscription.....	3
Connexion.....	4
Déconnexion.....	4
Routes.....	4
<b>Implémentation de l'autorisation:</b>	<b>5</b>
Pages publiques :	5
Pages privées :	7

# Implémentation de l'authentification:

Afin de permettre à l'utilisateur d'accéder à ses pages privées telles que sa page de dessin et sa page de profil, il est nécessaire d'implémenter une authentification.

## Inscription

L'utilisateur peut accéder à la page d'inscription depuis la page d'accueil. Sur cette page l'utilisateur doit renseigner 4 informations afin de créer son compte: son nom d'utilisateur, son adresse e-mail, son numéro de téléphone ainsi que son mot de passe. Lorsque l'utilisateur a renseigné ses informations, il peut cliquer sur un bouton afin de confirmer son inscription. Une vérification est effectuée sur les champs renseignés afin de s'assurer que les informations telles que son e-mail ou son mot de passe sont valides et que son e-mail n'est pas déjà utilisé par un autre compte.

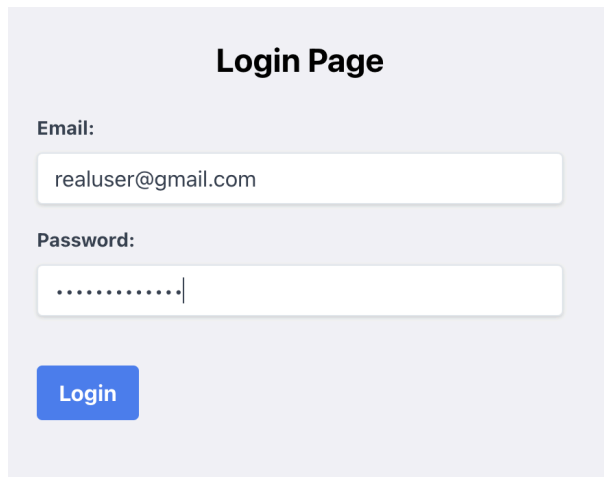
The image displays three sequential screenshots of a web registration form titled "Registration Page". Each form contains four input fields: "Username:" (containing "myUsername"), "Email:" (containing "realuser@gmail.com"), "Phone Number:" (containing "0767063130"), and "Password:" (containing masked characters "....."). A blue "Register" button is positioned at the bottom of each form.

- Left Screenshot:** The form is in its initial state with all fields filled and no error messages.
- Middle Screenshot:** An error message "Invalid email address." is displayed in red text above the Username field, indicating a validation failure.
- Right Screenshot:** An error message "Invalid phone number. Must be 10 digits." is displayed in red text above the Username field, indicating a validation failure.

Lorsque le bouton "register" est cliqué, une requête POST est envoyée à l'api sur la route /register, les informations sont vérifiées à nouveau et si elles sont valides et que l'e-mail n'est pas déjà utilisé l'inscription est validée et l'utilisateur est inséré dans la base de données. Le mot de passe est stocké sous forme de hash réalisé avec la bibliothèque bcrypt afin de garantir la confidentialité de cette information. Si ces informations ne sont pas valides, un message d'erreur est renvoyé.

## Connexion

Une fois son inscription réalisée avec succès, l'utilisateur est redirigé sur la page de connexion. Sur celle-ci il doit entrer son e-mail ainsi que son mot de passe.

A screenshot of a login page titled "Login Page". It features two input fields: "Email:" with the value "realuser@gmail.com" and "Password:" with a masked password ".....". Below the fields is a blue "Login" button.

**Login Page**

Email:  
realuser@gmail.com

Password:  
.....

Login

Une fois le formulaire envoyé les informations sont vérifiées et si un utilisateur correspond à l'e-mail entrée et que son mot de passe correspond avec le hash enregistré pour l'utilisateur, un cookie d'authentification contenant son ID est stocké sur son navigateur.

## Déconnexion

L'utilisateur peut se déconnecter à l'aide d'un bouton sur la page de dessin. Sinon, le cookie expire au bout de 24 heures et l'utilisateur ne sera plus authentifié.

## Routes

### **POST /register :**

Cette route attend un body contenant un nom d'utilisateur, un mot de passe, une adresse email et un numéro de téléphone. Lorsque cette route est utilisée, le backend vérifie que l'email n'est pas déjà utilisé (auquel cas il renvoie un message d'erreur), puis il attribue un id unique à l'utilisateur, réalise un hash avec le mot de passe en utilisant la librairie bcrypt et sauvegarde ses données dans une base de données MongoDB.

### **POST /login :**

Cette route attend un body contenant un email et un mot de passe, l'email est cherché dans la collection users de la base de données et si une correspondance est trouvée le mot de passe est vérifié avec le hash sauvegardé. Si la vérification est un succès, un cookie d'authentification est envoyé à l'utilisateur contenant son id.

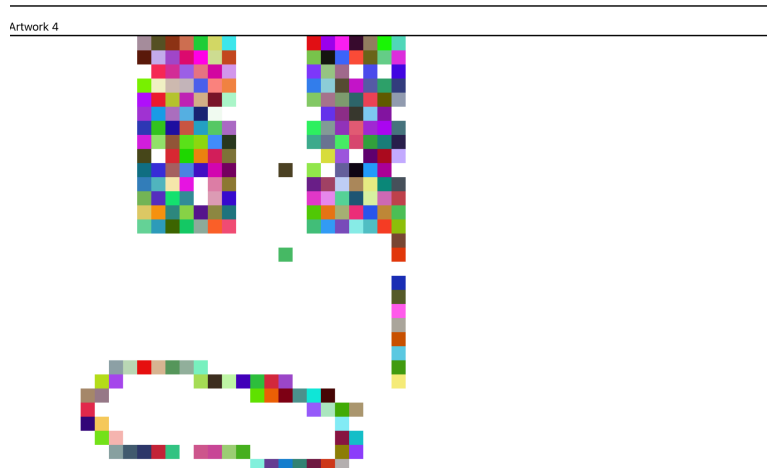
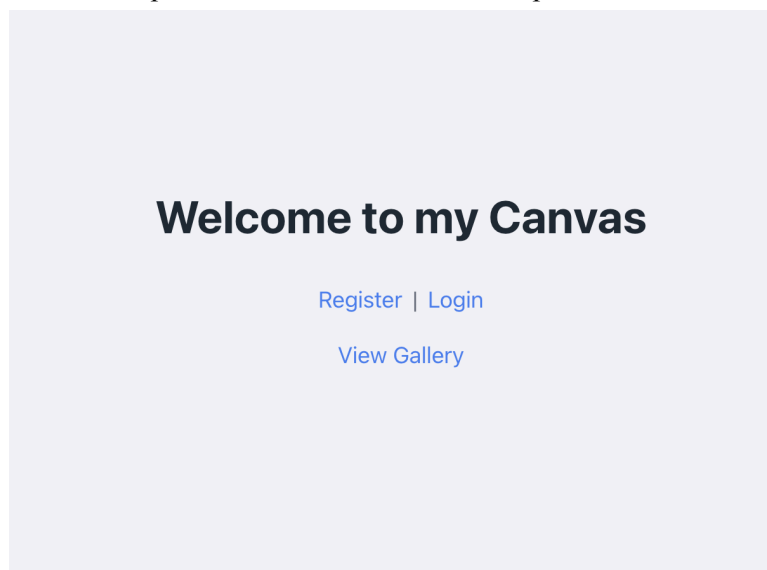
### **GET /logout :**

Lorsque cette route est appelée, elle efface le localStorage de l'utilisateur ainsi que son cookie d'authentification.

# Implémentation de l'autorisation:

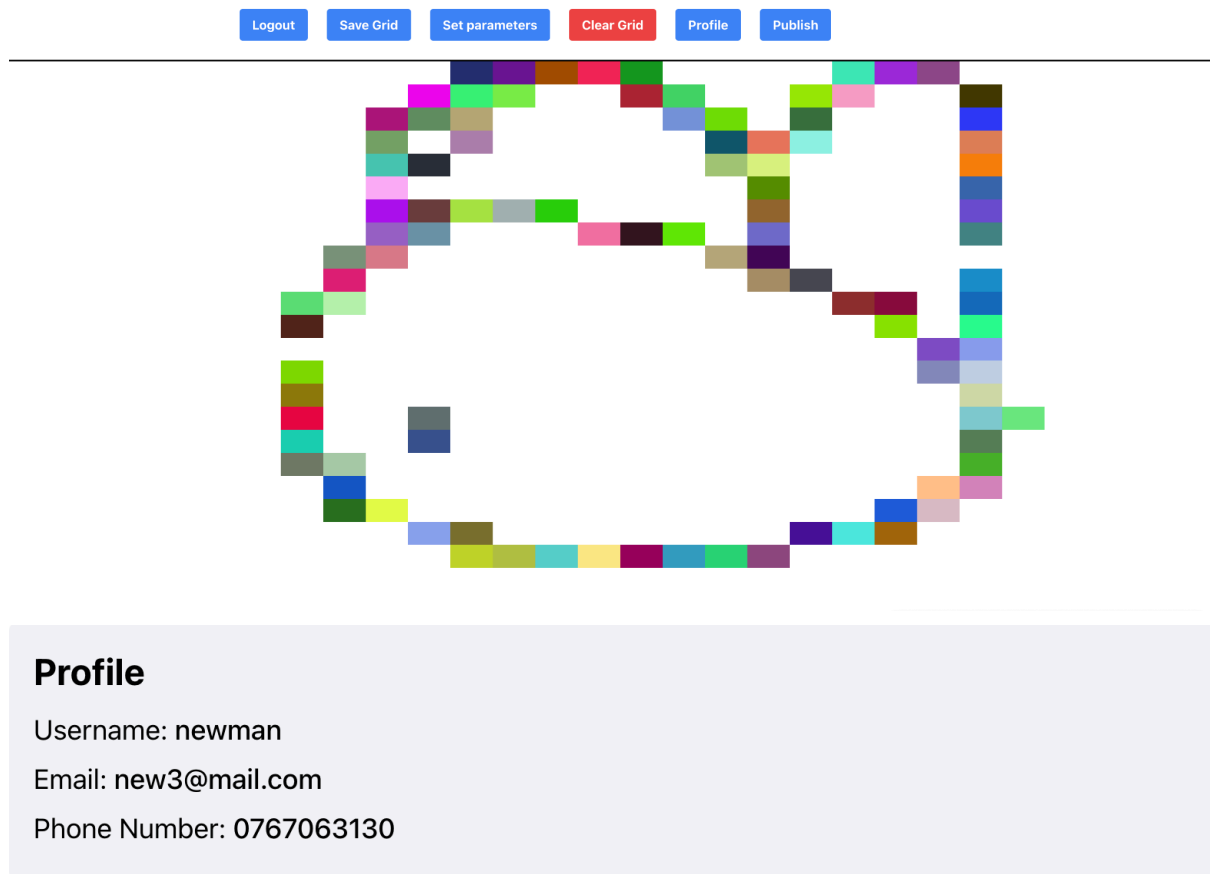
## Pages publiques :

L'application comprend plusieurs pages publiques accessibles aux utilisateurs non-authentifiés. Ces pages sont la page d'inscription et de connexion vues auparavant, la page d'accueil et la galerie où il est possible de visionner les canvas publiés.



## Pages privées :

Il existe également des pages privées accessibles uniquement aux utilisateurs authentifiés. Ces pages sont la page de dessin et la page de profil.



Lorsque l'utilisateur arrive sur ces pages, son cookie d'authentification est vérifié et s'il n'est pas présent, il est redirigé sur la page de connexion. Les routes permettant de récupérer les données telles que le canvas en cours ainsi que les informations de profil vérifient que l'utilisateur possède un cookie d'authentification dans sa requête et si ce n'est pas le cas ou que le cookie n'est pas valide un message d'erreur est renvoyé.