

République du Sénégal



UN PEUPLE – UN BUT – UNE FOI

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR DE LA RECHERCHE ET DE L'INNOVATION

UNIVERSITE GASTON BERGER DE SAINT-LOUIS



U.F.R DES SCIENCES APPLIQUEES ET TECHNOLOGIE
SECTION D'INFORMATIQUE

Complément à la programmation Web

Master I GDIL

Sujet TP : Application messagerie

Réalisé par :

- Fatou DIA : dia.fatou@ugb.edu.sn
- Ndèye Aïssatou GASSAMA : gassama.ndeye-aissatou@ugb.edu.sn
- Saly SAMAKE : samake.saly@ugb.edu.sn

2020-2021

Table des matières

Introduction	3
1 Structure de la base de données	3
1.1 Table "Utilisateur"	3
1.2 Table "Message"	3
2 Fonctionnalités et fenêtres de l'application	4
2.1 Accueil	4
2.2 Inscription	5
2.3 Connexion	6
2.4 Page utilisateur connecté - Discussion de groupe	7
2.5 Page utilisateur connecté - Discussion privée	8
3 Manuel d'installation	8
4 Manuel d'utilisation	8
5 Outils et technologies utilisés	10
Problèmes rencontrés	11
Répartition des tâches	11
Conclusion	11

Table des figures

1	Table "Utilisateur"	3
2	Table "Message"	3
3	Page d'accueil	4
4	Formulaire d'inscription	5
5	Formulaire de connexion	6
6	Discussion de groupe	7
7	Discussion privée	8
8	Liste utilisateurs	9
9	Recherche utilisateur	9
10	Envoie message	10
11	Supprimer message	10

Introduction

Dans le cadre de notre TP de Complément à la Programmation Web il nous a été demandé de développer une application de Messagerie Instantanée. Cette application permettra aux utilisateurs inscrits de pouvoir se connecter, de discuter dans un groupe de chat ou en privée. Ainsi via l'application l'utilisateur aura une vue globale sur ses discussions, pourra rechercher des utilisateurs avec qui il n'a jamais discuter, il pourra aussi supprimer des messages privés.

1 Structure de la base de données

Pour gérer les données de l'application nous avons créé les tables suivantes :

1.1 Table "Utilisateur"

#	Nom	Type	Interclassement
<input type="checkbox"/> 1	id 🔑	int(11)	
<input type="checkbox"/> 2	prenom	varchar(32)	latin1_swedish_ci
<input type="checkbox"/> 3	nom	varchar(32)	latin1_swedish_ci
<input type="checkbox"/> 4	email	varchar(32)	latin1_swedish_ci
<input type="checkbox"/> 5	pass	varchar(20)	latin1_swedish_ci

FIGURE 1 – Table "Utilisateur"

Cette table contient les données personnelles de l'utilisateur (nom, prénom, email, mot de passe) utiles au fonctionnement de l'application.

1.2 Table "Message"

#	Nom	Type	Interclassement
<input type="checkbox"/> 1	id 🔑	int(11)	
<input type="checkbox"/> 2	destinataire	varchar(32)	latin1_swedish_ci
<input type="checkbox"/> 3	origine	varchar(32)	latin1_swedish_ci
<input type="checkbox"/> 4	contenu	text	latin1_swedish_ci
<input type="checkbox"/> 5	date	datetime	

FIGURE 2 – Table "Message"

Cette table contient les messages entre utilisateurs de la plateforme ainsi que les messages du groupe.

2 Fonctionnalités et fenêtres de l'application

Dans cette partie nous allons présenter les principales fenêtres/fonctionnalités de notre application

2.1 Accueil

Cette fenêtre montre la page d'accueil pour les utilisateurs non connectés

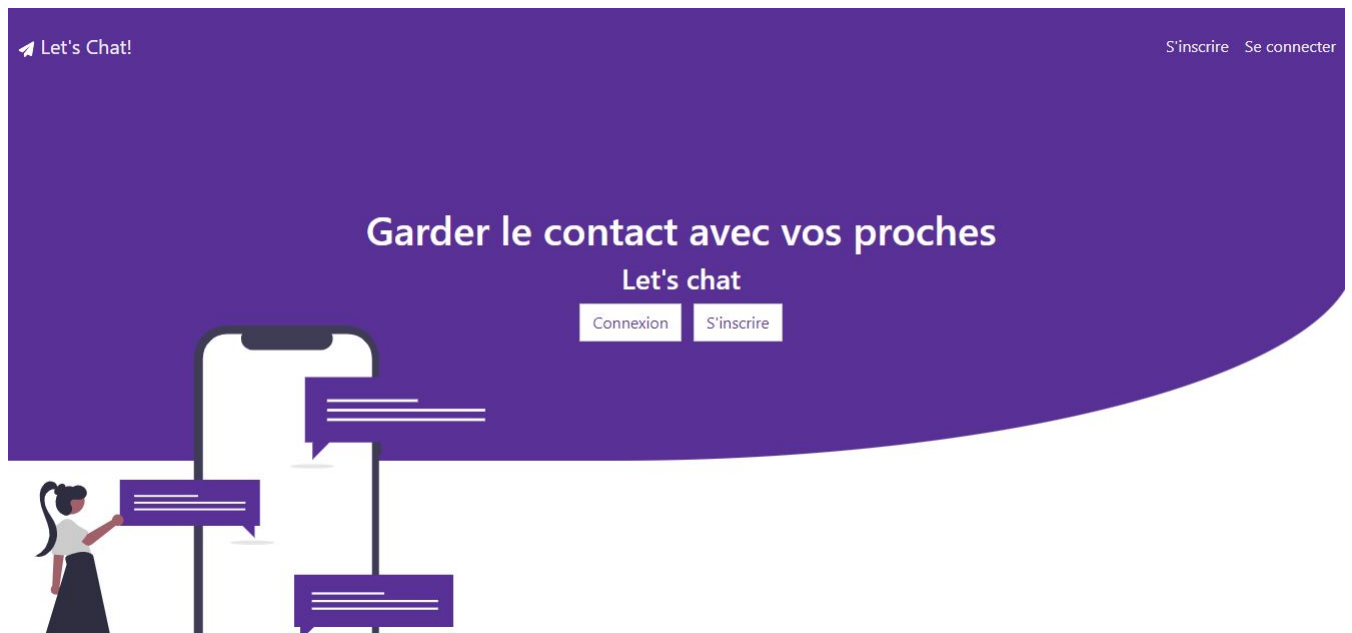
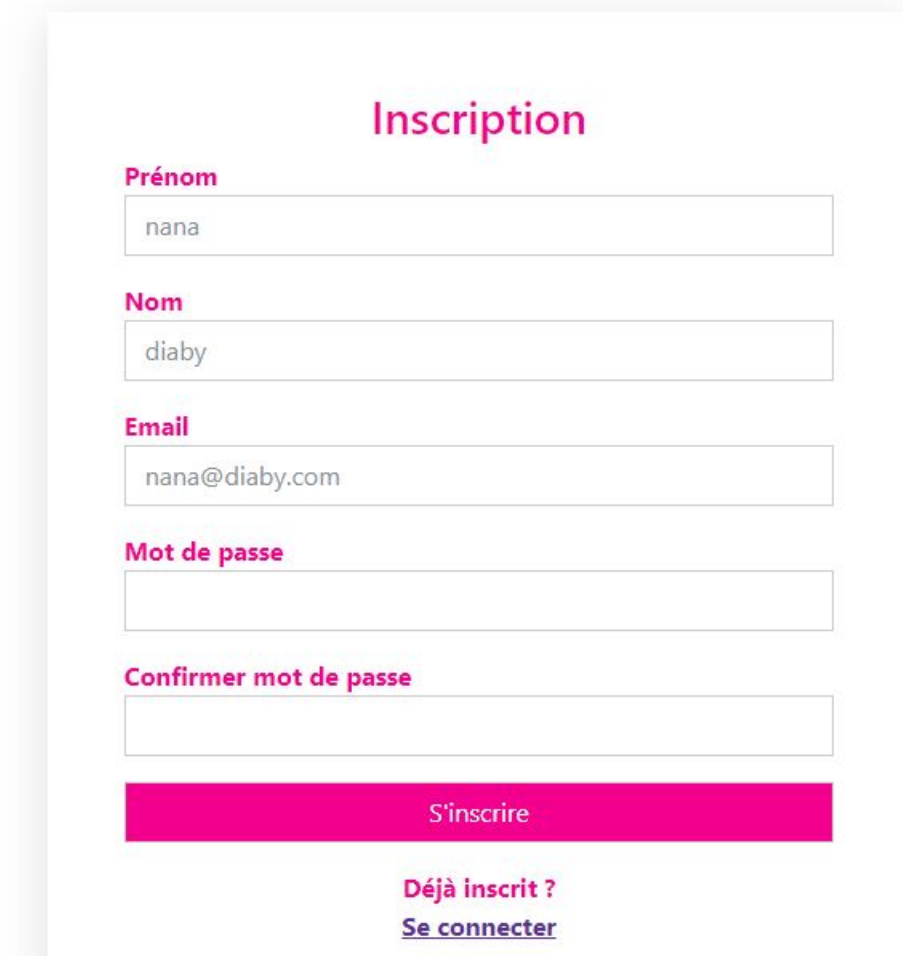


FIGURE 3 – Page d'accueil

2.2 Inscription



The image shows a registration form titled "Inscription" in a bold, pink font. Below the title, there are five input fields, each with a label in bold pink text: "Prénom" (containing "nana"), "Nom" (containing "diaby"), "Email" (containing "nana@diaby.com"), "Mot de passe", and "Confirmer mot de passe". Below these fields is a large pink button labeled "S'inscrire". At the bottom, there are two links: "Déjà inscrit ?" in pink and "Se connecter" in blue with a red underline.

Inscription

Prénom
nana

Nom
diaby

Email
nana@diaby.com

Mot de passe

Confirmer mot de passe

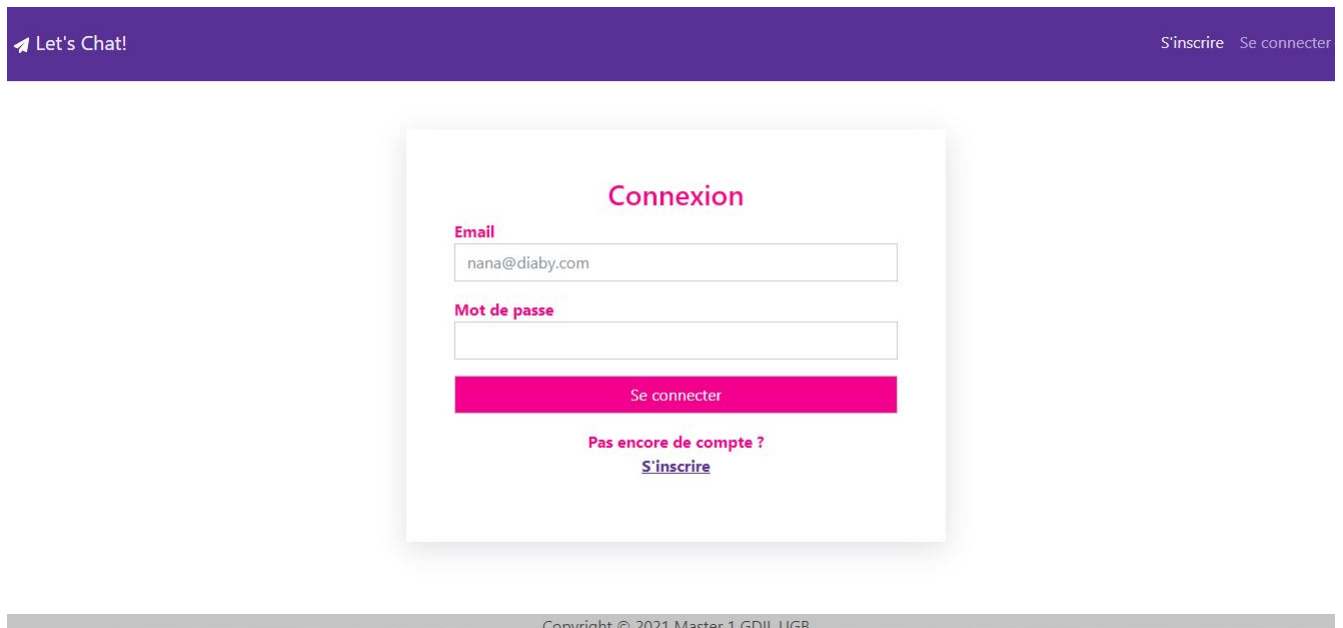
S'inscrire

Déjà inscrit ?
[Se connecter](#)

FIGURE 4 – Formulaire d'inscription

Cette fenêtre montre le formulaire d'inscription sur la plateforme. L'utilisateur doit saisir son nom, prénom, email et mot de passe pour pouvoir s'inscrire. Si les informations fournies sont conformes il sera inscrit sur la plateforme puis redirigé sur la page principale.

2.3 Connexion



The screenshot shows a web interface with a purple header bar. On the left of the header is a chat icon and the text "Let's Chat!". On the right are the links "S'inscrire" and "Se connecter". In the center is a white login form titled "Connexion" in pink. The form contains two input fields: "Email" with the value "nana@diaby.com" and "Mot de passe" which is empty. Below these fields is a pink button labeled "Se connecter". At the bottom of the form, it says "Pas encore de compte ?" followed by a blue link "S'inscrire". A grey footer bar at the bottom contains the text "Copyright © 2021 Master 1 GDII UGB".

FIGURE 5 – Formulaire de connexion

Cette fenêtre montre le formulaire de connexion sur la plateforme. L'utilisateur doit saisir son identifiant (mail) ainsi que son mot de passe pour se connecter. Si les informations fournies sont correctes il sera redirigé sur la page principale.

2.4 Page utilisateur connecté - Discussion de groupe

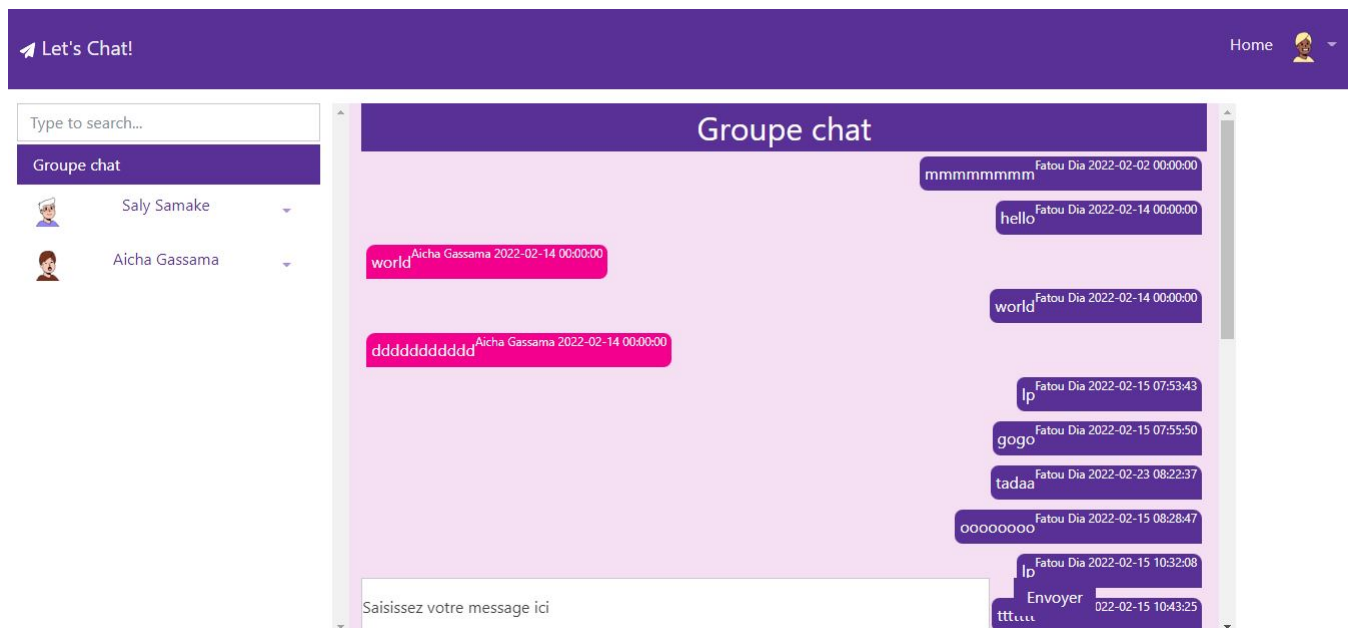


FIGURE 6 – Discussion de groupe

Cette fenêtre montre la page d'accueil de l'utilisateur. Ce dernier pourra consulter ses messages de groupe ou ses messages privés. Sur la barre gauche il peut voir la liste de ses contacts ou rechercher les utilisateurs inscrit sur la plateforme par leur email. Il pourra également envoyer des messages et se déconnecter. Sur cette fenêtre on peut voir la discussion de groupe entre les différents utilisateurs. Sur chaque message on peut voir le nom de l'utilisateur qui l'a envoyé ainsi que la date d'envoi.

2.5 Page utilisateur connecté - Discussion privée



FIGURE 7 – Discussion privée

Sur cette fenêtre on peut voir un exemple de discussion privée. L'utilisateur a la possibilité de supprimer un message privé.

3 Manuel d'installation

Pour installer l'application il faut d'abord avoir un serveur local (XAMPP ou WAMP) et être connecté à internet (pour l'API DiceBear). Ensuite :

- Placer le dossier "app-messagerie" dans le dossier "xampp/htdocs" si vous utilisez XAMPP ou "wamp/www" si vous utilisez WAMP
- Créer une base de données nommée "messagerie" dans MySQL et importer (à partir de l'interface de phpMyAdmin) le fichier "messagerie.sql"
- Démarrer les serveurs Apache et MySQL

4 Manuel d'utilisation

- Ouvrir un navigateur ou 2 (différents) pour tester la fonctionnalité discussion instantanée, de préférence Chrome, Edge ou Firefox (L'API fetch ne fonctionne pas sur Internet Explorer)
- sur la barre d'adresse saisir `http://localhost/app-messagerie/frontend/index.html`
- Une fois sur la page d'accueil choisir le bouton "connexion" pour se connecter ou "s'inscrire" pour créer un compte. Vous pouvez vous logger avec un des comptes déjà créé (voir capture ci après). On ne peut pas se logger en tant que "groupe"

id	prenom	nom	email	pass
1	Groupe			
2	Fatou	Dia	fatou@dia.com	password
3	Aicha	Gassama	aicha@gassama.com	password
4	Saly	Samake	saly@samake.com	password
5	Modou	modou	modou@modou.com	password
6	nana	diaby	nana@diaby.com	password

FIGURE 8 – Liste utilisateurs

- Une fois connecté on peut rechercher les utilisateurs par leur mail avec le formulaire de recherche sur la barre gauche (voir capture ci-dessous). Ensuite choisir l'utilisateur qu'on cherche parmi la liste déroulante et taper la touche "Enter" du clavier. Un bouton va apparaître avec le nom de la personne trouvée. On peut alors commencer à discuter avec ce dernier.

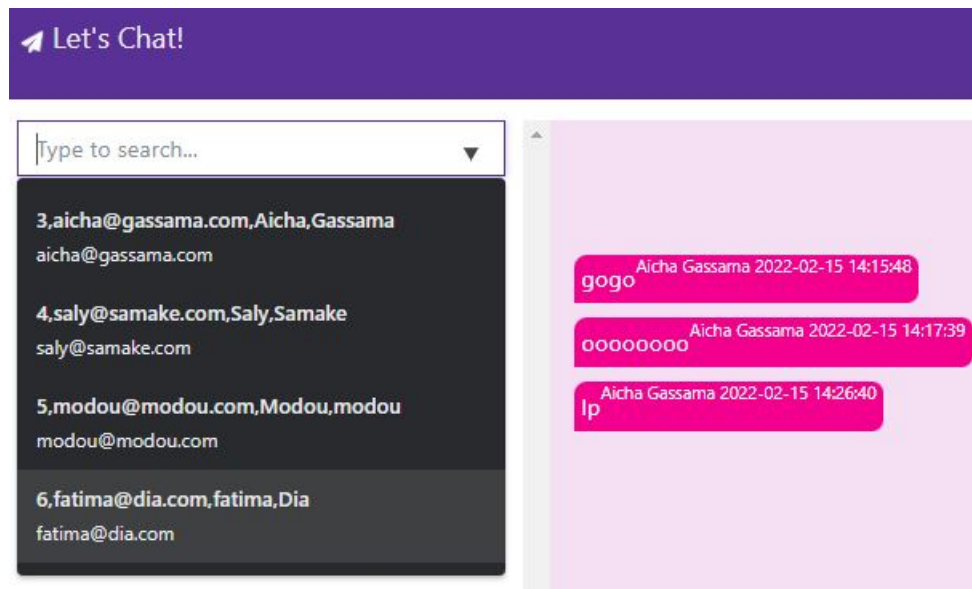


FIGURE 9 – Recherche utilisateur

- Pour envoyer un message choisir le bouton portant le nom de l'utilisateur avec qui on veut discuter (ou bien choisir le groupe) et aller sur le champ du formulaire d'envoi (voir capture)



FIGURE 10 – Envoie message

- Pour supprimer un message privée (on ne peut supprimer un message de groupe) cliquer sur le bouton déroulant du message et cliquer sur supprimer

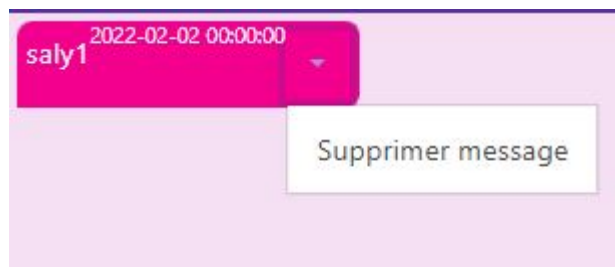


FIGURE 11 – Supprimer message

5 Outils et technologies utilisés

Pour réaliser ce projet nous avons utilisé :

- Langages de programmation : HTML, CSS, JavaScript et PHP simple.
- Framework : Bootstrap pour le style
- APIs :
 - API Fetch pour faire des requêtes en asynchrone
 - API DOM pour créer et supprimer dynamiquement des éléments dans le document HTML
 - API DiceBear (<https://avatars.dicebear.com>) pour générer automatiquement des avatars ou images de profile pour rendre l'interface plus conviviale
- Base de données : MySQL
- Autres : XMLHttpRequest pour faire des requêtes avec le "back-end", PhpMyAdmin

Problèmes rencontrés

La principale difficulté rencontrée était de faire des requêtes de type POST avec l'API fetch. C'est pour cette raison qu'on a utilisé l'objet XMLHttpRequest pour faire des requêtes POST.

Répartition des tâches

- Fatou DIA : back-end
- Ndèye Aïssatou GASSAMA : front-end, back-end
- Saly SAMAKE : back-end

Conclusion

Pour finir nous pouvons dire que ce projet nous a permis d'approfondir les connaissances acquises durant le cours de Complément en Programmation Web. En effet il nous a permis d'utiliser Ajax, d'approfondir nos connaissances en JavaScript et de mieux nous familiariser avec PHP.