



# First Term Project - Messaging Application

*MohammedAmineAghlaOuaziz*

## P1. Messaging Web application

### Introduction

The project consists of creating a web application that allows users to exchange messages. It is similar to the usual internal messaging systems in companies or a webmail system.

### Requirements Specification

The Messaging Application is designed to provide a secure and efficient platform for users to communicate. Accessible through a username-password combination stored in a database, the application enables users to send messages to specific recipients and view their inbox, displaying a concise overview of messages. Full message content can be viewed individually, and users can respond as needed. Additionally, an outbox feature organizes sent messages, detailing the recipient's interaction status. Connection data to the database and mail server is securely stored in files. Various extensions, including encrypted passwords, self-registration with email activation, password recovery, and support for multiple recipients and image attachments, enhance the application's functionality. These extensions cater to user experience, security, and administrative capabilities, contributing to a versatile and user-friendly communication platform.

| We used TailwindCSS to style the application and give it a nice look for a better user experience

### Main functionalities

- **A1: Access:** Users log in using a secure username-password combination stored in a database.
- **A2: Sending:** Application users can send messages to others, with the recipient specified by username.
- **A3: Inbox:** Users view a summarized list of messages, distinguishing between read and unread messages. The inbox displays subjects or the initial content.
- **A4: Messages:** Users can view individual messages, read full content, and respond accordingly.
- **A5: Outbox:** Users access a sorted table of sent messages, indicating the recipient, date sent, and allowing access to each message. Interaction status with the recipient is also visible.
- **A6: Connection Data:** Database and mail server connection data is securely stored in files.

### Extensions

- **E1. Presentation:** Public presentation includes a comprehensive overview, development process description, and a live demo.
- **E2. Encrypted User Password:** User passwords are stored securely using encryption `password_hash()`.
- **E3. Self-Registration:** Realistic user registration includes email activation.
- **E4. Password Recovery:** Users can realistically retrieve or generate a new password.
- **E5. Messages to Multiple Recipients:** Users can enter multiple recipients in the message submission form.
- **E6. User Avatar:** Users upload and update avatars, impacting their appearance in messages and profiles.
- **E7. User Profile:** Users enter and modify personal data; others can access this information.
- **E8. Administration Zone:** An administrator-exclusive zone facilitates user management tasks.
- **E9. Attached Files:** Messages support file attachments.
- **E10. Images:** Images can be inserted directly into messages, with full viewability.

## Implementation Details

To bring the outlined features to life, the project has been crafted with attention to the following implementation details

**Access Control:** Leveraging session-based authentication, the system rigorously verifies user credentials before granting access.

**Message Management:** The platform offers a user-friendly interface for composing, sending, and managing messages. Messages marked as private are restricted from the view of other users.

**User Profiles:** Profiles act as a public representation for users, displaying their messages, profile picture, and bio to fellow community members.

**Search Functionality:** Users have the ability to search for other users and messages across the platform, enhancing overall accessibility.

**Secure Configuration:** Configuration files serve as secure repositories for storing sensitive information such as database connections and mail server settings, preventing exposure in the public codebase.

**Password Encryption:** Prior to storage in the database, user passwords undergo secure hashing through the `password_hash()` function, ensuring an additional layer of protection.

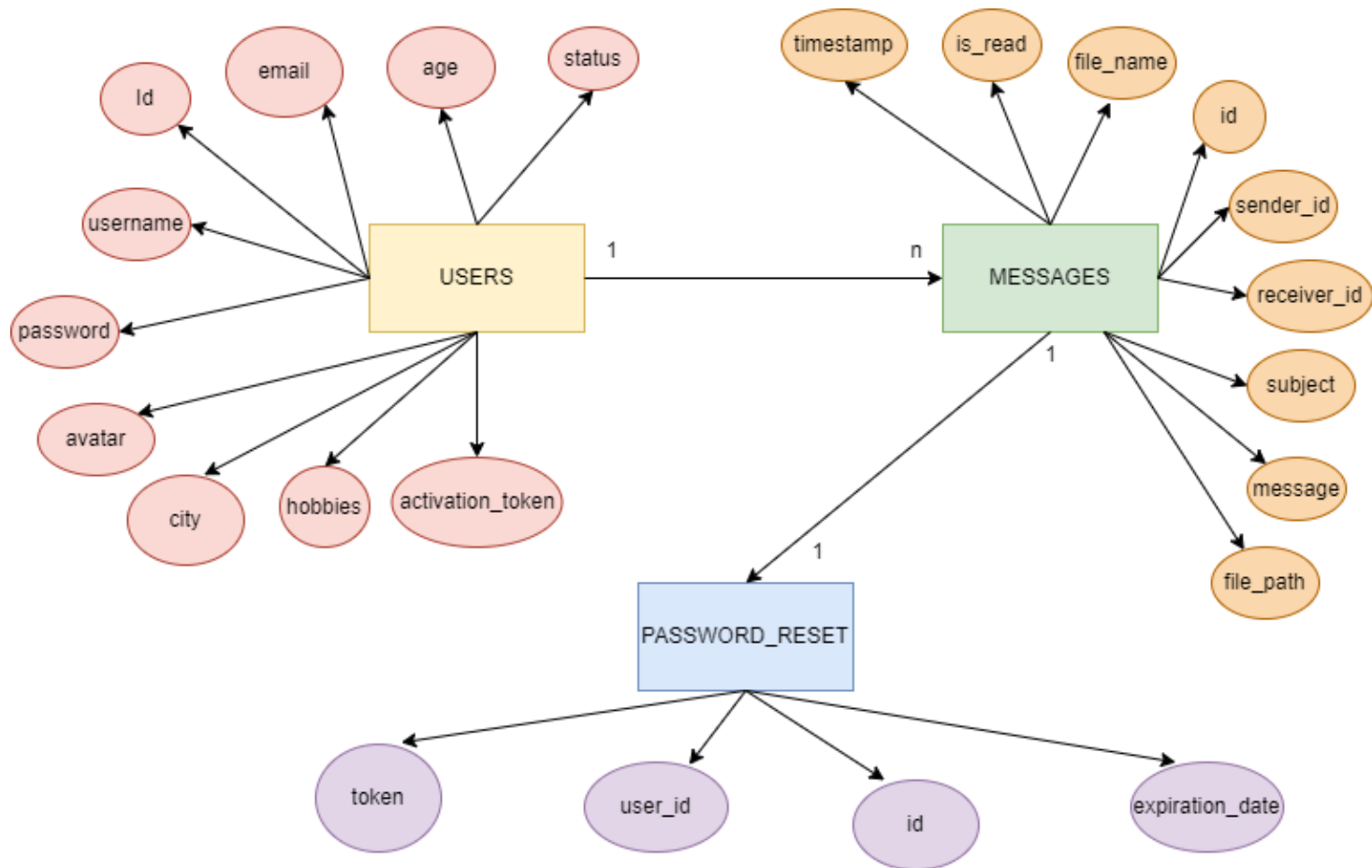
**Self-registration and Account Activation:** New users seamlessly register on the platform, with an email verification link sent to activate their accounts, enhancing security.

**Password Recovery:** In situations of forgotten passwords, a robust recovery process empowers users to set new passwords via a secure link sent to their registered email.

**User Interactions:** A dynamic commenting and rating system fosters a vibrant sense of community and interaction among platform users.

**User Experience:** Beyond functionality, the platform prioritizes an appealing visual experience, achieved through thoughtful styling with TailwindCSS.

## E/R SCHEME



Relationships :  
Users to Messages: One-to-Many (Each user can send/receive multiple messages, but each message belongs to one user)  
Messages to Password\_Reset: One-to-One (Each message has one password reset entry, and each password reset entry is associated with one message)

## Logical Model

The logical model for this messaging application comprises three primary entities: `Users`, `Messages`, and `Password_Reset`. Each entity is defined by specific attributes, capturing essential information about users, messages, and password reset entries. The relationships between these entities establish the connections necessary for the application to function cohesively.

### 1. Users:

- Attributes:
  - `id: INT` (Primary Key, Auto-Increment, Not Null)
  - `username: VARCHAR(50)` (Unique, Not Null)
  - `password: VARCHAR(255)` (Not Null)
  - `email: VARCHAR(100)` (Unique, Not Null)
  - `avatar: VARCHAR(255)` (Nullable)
  - `age: INT` (Nullable)
  - `city: VARCHAR(255)` (Nullable)
  - `hobbies: VARCHAR(255)` (Nullable)
  - `activation_token: VARCHAR(64)` (Nullable)
  - `status: ENUM('inactive', 'active')` (Default 'inactive', Not Null)

### 2. Messages:

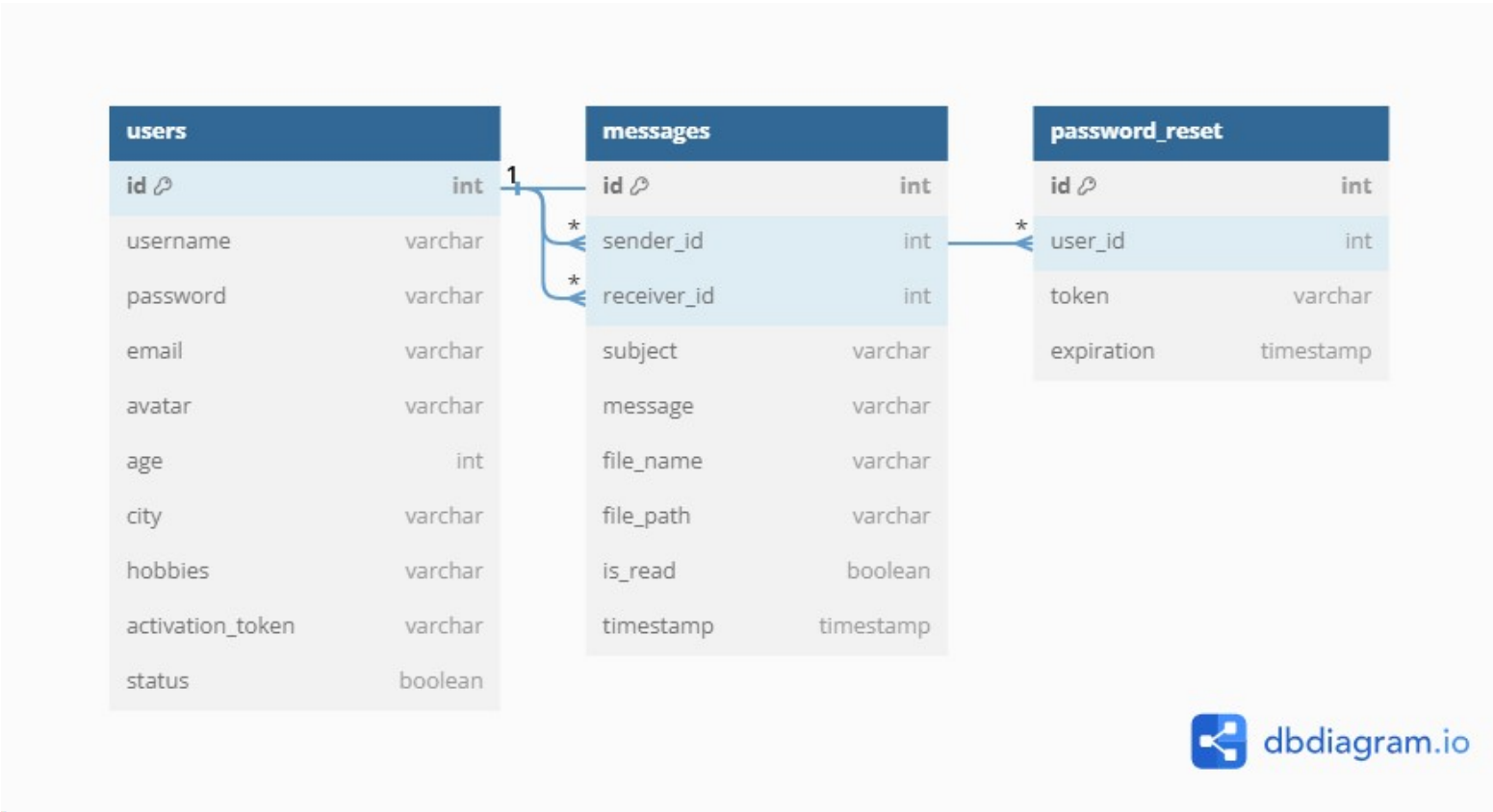
- Attributes:
  - `id: INT` (Primary Key, Auto-Increment, Not Null)
  - `sender_id: INT` (Foreign Key referencing Users, Not Null)
  - `receiver_id: INT` (Foreign Key referencing Users, Not Null)
  - `subject: VARCHAR(255)` (Not Null)
  - `message: TEXT` (Not Null)
  - `file_name: VARCHAR(255)` (Nullable)
  - `file_path: VARCHAR(255)` (Nullable)
  - `is_read: BOOLEAN` (Not Null)

- `timestamp: TIMESTAMP` (Not Null)

### 3. Password\_Reset:

- **Attributes:**
  - `id: INT` (Primary Key, Auto-Increment, Not Null)
  - `user_id: INT` (Foreign Key referencing Users, Not Null)
  - `token: VARCHAR(255)` (Not Null)
  - `expiration: TIMESTAMP` (Not Null)

## Database Diagram



This Database diagram is generated using [dbdiagram.io](https://dbdiagram.io) an online website similar to sql developer

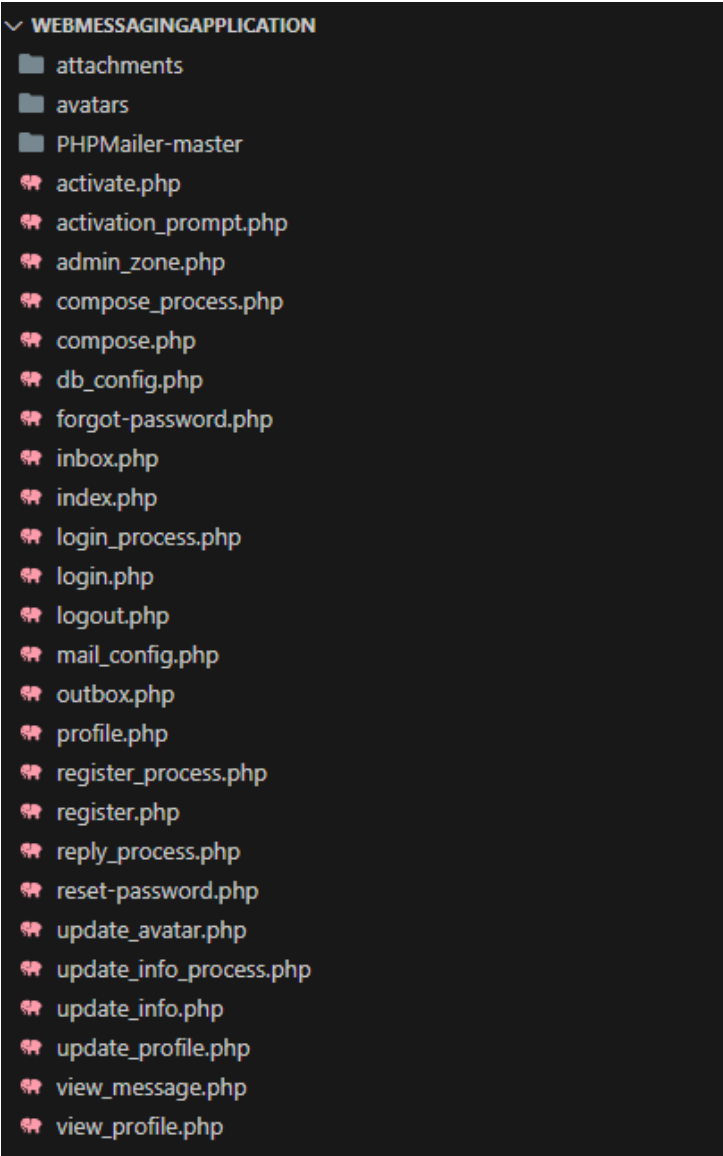
## Screens Map

1. **Welcome Page ( `index.php` ):**
  - Welcoming title and image
  - Links to:
    - Login ( `login.php` )
    - Register ( `register.php` )
    - Forgot Password ( `forgot-password.php` )
2. **Login Page ( `login.php` ):**
  - Login form
  - Link to Forgot Password ( `forgot-password.php` )
  - Link to Register ( `register.php` )
3. **Registration Page ( `register.php` ):**
  - Registration form
  - Link to Login ( `login.php` )

- 4. **Forgot Password Page ( forgot-password.php ):**
  - Password recovery form
  - Link to Login ( login.php )
- 5. **Activation Page ( activate.php ):**
  - Activation status message
- 6. **Dashboard / Inbox ( inbox.php ):**
  - List/table of received messages (subject or beginning)
  - Links to view individual messages
  - Compose message button
  - Link to Outbox ( outbox.php )
  - Link to Profile ( profile.php )
  - Logout option
- 7. **Outbox ( outbox.php ):**
  - List/table of sent messages
  - Links to view individual sent messages
- 8. **View Message ( view\_message.php ):**
  - Display individual message details
  - Reply form
- 9. **Compose Message ( compose.php ):**
  - Message composition form
  - Option to attach files/images
- 10. **Profile Page ( profile.php ):**
  - Display user profile information
  - Option to update avatar
  - Option to update profile information
  - Link to Inbox ( inbox.php )
  - Link to Outbox ( outbox.php )
  - Logout option
- 11. **Admin Zone ( admin\_zone.php ):**
  - Specific tasks for administrators
  - (Tasks to be defined based on admin functionalities)
- 12. **Password Reset Page ( reset-password.php ):**
  - Password reset form
- 13. **Update Avatar Page ( update\_avatar.php ):**
  - Form to update user avatar
- 14. **Update Profile Information Page ( update\_info.php ):**
  - Form to update user profile information
- 15. **View Profile ( view\_profile.php ):**
  - Display profile information of other users
- 16. **Activation Prompt ( activation\_prompt.php ):**
  - Prompt for activation status

## Files Summary Table



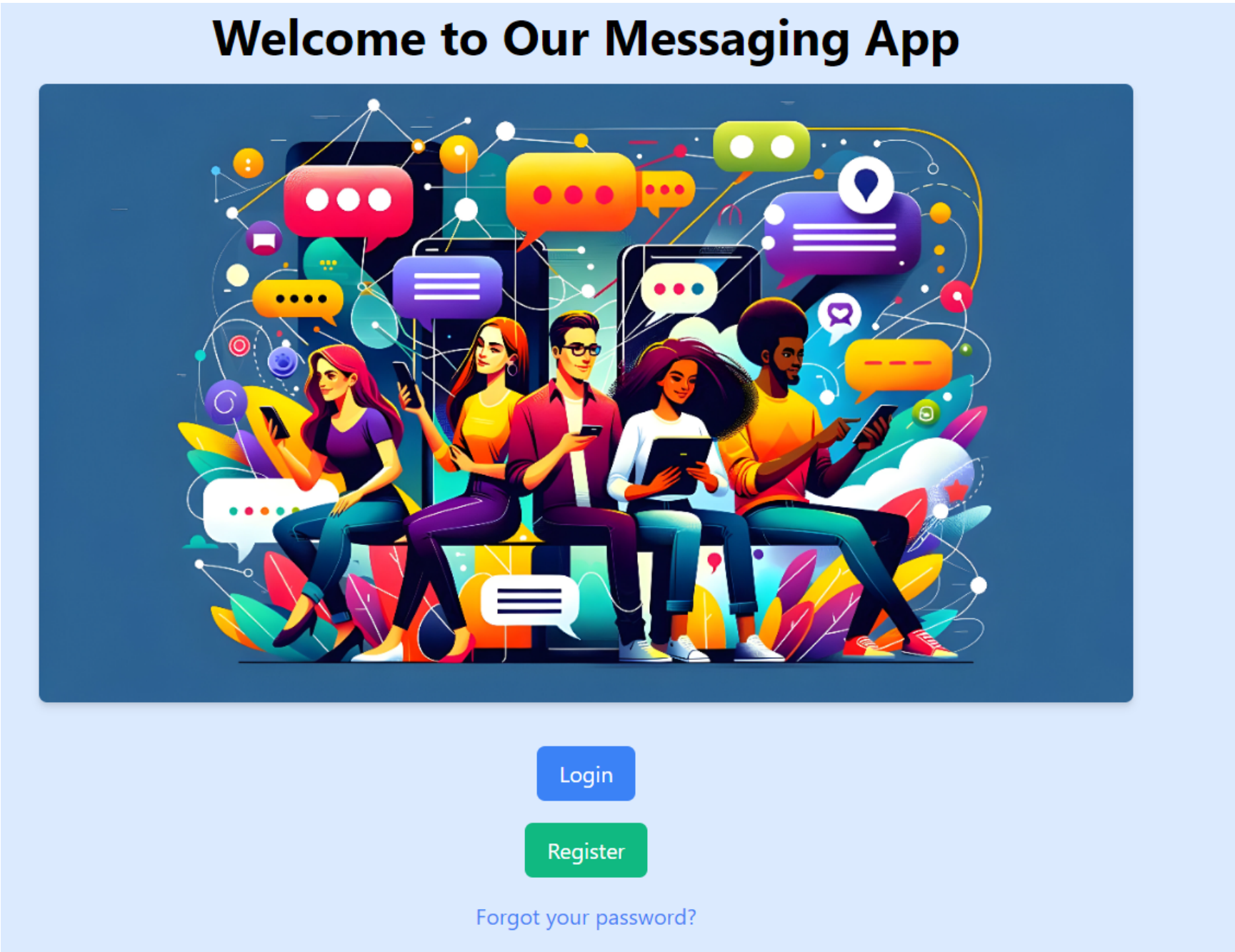


## Quick overview of each file

Path	
	Directory to store files attached to messages
attachments folder	Directory to store files attached to messages
avatars	Directory to store user avatars
PhpMailer-master	PHPMailer library for sending emails
activate.php	Handles user account activation
activation_prompt.php	Displays a prompt after successful activation
admin_zone.php	Admin zone with tasks for administrators
compose_process.php	Processes message composition
compose.php	User interface for composing messages
db_config.php	Configuration file for connecting to the database
forgot_password.php	Initiates the password recovery process
inbox.php	Displays received messages
index.php	Main landing page or home page
login_process.php	Handles user login process
login.php	User login page
logout.php	Logs out the user
outbox.php	Displays sent messages
profile.php	Displays user profile
register_process.php	Handles user registration process
register.php	User registration page
reply_process.php	Processes message replies
reset-password.php	Allows users to reset their password
update_avatar.php	Handles updating user avatars
update_info_process.php	Processes updates to user information
update_info.php	User interface for updating information
update_profile.php	User interface for updating the user profile

Path	Directory to store files attached to messages
<a href="#">view_message.php</a>	Displays details of a single message
<a href="#">view_profile.php</a>	Displays the user's profile

User Guide



This is the main page `index.php` or the welcoming page, from here the user can register or login, and if he forgot the password he can click the link ' `forgot your password?` '.

# Register

Email:

Username:

Password:

Register



This is the register.php page, here the user chooses a username, he writes his e-mail and password.

# Registration Successful

Please check your email for the activation link.



Once the registration is completed, the user is asked to check his e-mail for an activation link, he can't log in without activating his account first.



# Activate Your Account

Inbox x



**Your Name** <aminefootball4me@gmail.com>

to me ▾

Hello bran,

Click the following link to activate your account:

<http://localhost/php/activate.php?token=d2a78c87a5ab5366d6553c89294209e4>

↩ Reply

➦ Forward



This is how the activation link looks like in the user's e-mail.

Account activated successfully. You can now log in.

[Login](#)



When clicking the activation link, it redirects him to this page where he can click the login link.

# Login

Username:

Password:

Login

[Forgot your password?](#)

Not registered yet? [Register here.](#)



This is the login.php page where the user can login with his username and his chosen password.

# Password Recovery

Email:

Send Password Reset Email



and he will be asked to introduce his e-mail an he will receive a reset password link. There is also `the forgot your password?` link , in case the user forgot his password, he can click it and he will receive a `reset password` link.

# Password Recovery

Password reset link sent to your email.



When the reset is sent, the user will get this page.

## Reset Password

New Password:

Update Password



Here the user is asked to introduce a new password and click update password.

## Reset Password

Password updated successfully.



After introducing the new password, the new is informed that the password is updated successfully.

## User Profile

Status: Active

No avatar available.

Hello, bran!

Update Avatar:

Seleccionar archivo

 Ninguno archivo selec.

Upload Avatar

Hello, bran!

Age:

City:

Hobbies:

Update Information

Logout

[Go to Inbox](#)

[Send a Message](#)

[Go to Outbox](#)



now after loging-in, the user can access the user\_profile.php!

# User Profile

Status: Active



Hello, bran!

Update Avatar:

Seleccionar archivo

 Ninguno archivo selec.

Upload Avatar

Hello, bran!

Age:

City:

Hobbies:

Update Information

Logout

[Go to Inbox](#)

[Send a Message](#)

[Go to Outbox](#)



The user can add an avatar to his profile, or he can update his current one.



## Update Information

Age:

City:

Hobbies:

## Update Information

[Back to Profile](#)

Here the user can add or update personal informations about himself like city of residence, age and hobbies, also the other users can see his personal informations.

# Compose Message

Recipients (comma-separated):

Subject:

Message:

Attachment:

Seleccionar archivo

Ninguno archivo selec.

Send



This is the compose\_message.php page, the user can send a message to one or many recipients at the same time. also he can attach files or images to his message.

## Outbox

Recipient	Subject	Date Sent	Read Status	View Message
mike	mike	2023-11-21 22:30:13	Unread	<a href="#">View Message</a>



After succefully sending the message, the user get redirected to his outbox.php page, showing the message he had sent with the subject, the date sent, the read status and a [view\\_message](#) link.

## Message Details

**Subject:** mike  
**From:** bran  
**Message:** mike  
**Attachment:**



## Reply to Message

Your Reply:

Send Reply



This is what appear after clicking the view\_message link, as you can see, when sending an image it appears directly with the message sent, and below the use can write his reply message and send it.

## Inbox

From: [bran](#) | Subject: mike | Status: Unread | Time: 2023-11-21 22:30:13

[View Message](#)

[Go to Administration](#)



This is the inbox.php page where the user can see the messages sent to him, with the subject, status, and the time the message was sent.

## Welcome, mike (Admin)!

Administration features:

Username	Status	Action
mike	Active	<a href="#">Block</a>   <a href="#">Unblock</a>
sam	Active	<a href="#">Block</a>   <a href="#">Unblock</a>
bran	Active	<a href="#">Block</a>   <a href="#">Unblock</a>



and This is the admin page, in this case Mike the admin has the possibility to block and unblock users, if some user is blocked, he can't send nore receive messages.

🌟 Thank You for Reading! 🌟