Functionality Guide

+ DB Description and Task Breakdown

Abdulraof Abdu, Abdirahman Ali, Matthew Augi

-  -

Table of Contents

[Functionality Guide 1](#_Toc174308759)

[Introduction 1](#_Toc174308760)

[Overview of the Application 1](#_Toc174308761)

[JavaScript Files 1](#_Toc174308762)

[Front end 4](#_Toc174308763)

[Server files 7](#_Toc174308764)

[Database Description 10](#_Toc174308765)

[Task Breakdown 12](#_Toc174308766)

# Functionality Guide

## Introduction

This is the functionality guide for the chat application. This paper provides an overview of how it’s various components work together to provide a seamless chat experience. We'll cover the purpose and functionality of each PHP and JavaScript file used in the application.

## Overview of the Application

The chat website consists of several pages and scripts that work together to enable user registration, login, chatting, and user management. The application includes:

1. **User Registration and Login**: Allows users to sign up and log in.
2. **User Profile and Chat Interface**: Enables users to view profiles and send messages.
3. **User Management**: Provides functionalities for viewing and deleting users.
4. **Real-Time Messaging**: Facilitates real-time communication between users.

### JavaScript Files

**1. chat.js**

**Description**

Handles real-time communication between users by managing the sending and receiving of messages.

**Features**

* **Message Sending**:
  + Prevents the default form submission behavior.
  + Uses AJAX to send message data to php/insert-chat.php when the send button is clicked.
  + Clears the input field and scrolls to the bottom of the chat box after sending a message.
* **Message Retrieval**:
  + Periodically sends AJAX requests to php/get-chat.php every 500 milliseconds to fetch new messages.
  + Updates the chat box with new messages and scrolls to the bottom if the chat box is not active.
* **Chat Box Interaction**:
  + Adds an "active" class to the chat box when hovered over, and removes it when the mouse leaves.

**2. login.js**

**Description**

Manages the user login process, including validation and AJAX communication with the server.

**Features**

* **Form Submission**:
  + Prevents the default form submission behavior.
  + Validates the email and password fields for correct format and non-emptiness.
* **Login Validation**:
  + Alerts the user if the email is invalid or if the password is missing.
* **AJAX Login Request**:
  + Sends login credentials to php/login.php using AJAX.
  + Redirects the user to users.php upon successful login.
  + Displays error messages if login fails.

**3. pass-show-hide.js**

**Description**

Allows users to toggle the visibility of their password in the login and signup forms.

**Features**

* **Password Visibility Toggle**:
  + Changes the password field type between "password" and "text" when the toggle button is clicked.
  + Updates the appearance of the toggle button based on the visibility state.

**4. signup.js**

**Description**

Handles the user registration process, including form validation and AJAX communication.

**Features**

* **Form Submission**:
  + Prevents the default form submission behavior.
* **Form Validation**:
  + Validates first name, last name, email, password, and image fields for correctness and completeness.
  + Displays error messages next to invalid fields.
* **AJAX Signup Request**:
  + Sends the registration data to php/signup.php using AJAX.
  + Redirects the user to users.php upon successful signup.
  + Displays error messages if signup fails.

**5. users.js**

**Description**

Manages user-related functionalities, including searching for users and managing user accounts.

**Features**

* **User Search**:
  + Toggles the visibility of the search bar and updates its focus when the search button is clicked.
  + Sends AJAX requests to php/search.php to retrieve search results based on user input.
* **User List Update**:
  + Periodically refreshes the user list every 500 milliseconds using AJAX requests to php/users.php.
* **User Deletion**:
  + Handles user deletions by sending AJAX requests to php/delete.php when a delete button is clicked.
  + Confirms the deletion action and refreshes the user list upon successful deletion.

### Front end

**1. chat.php**

**Description**

Manages the chat interface for users, displaying the selected user's details and handling message input.

**Features**

* **User Authentication**:
  + Checks if the user is logged in; redirects to login.php if not authenticated.
* **User Information**:
  + Retrieves and displays the selected user's information (e.g., name, status, profile image) based on the user\_id parameter.
* **Chat Interface**:
  + Displays the chat area with the option to send messages.
  + Includes a form for typing messages, hidden fields to track user IDs, and a submit button.
* **Script Integration**:
  + Includes chat.js for handling real-time message sending and retrieval.

**2. delete.php**

**Description**

Handles the deletion of a user from the database.

**Features**

* **User Deletion**:
  + Receives a POST request with a user\_id to delete the corresponding user from the database.
  + Redirects to users.php upon successful deletion or displays an error message if deletion fails.

**3. header.php**

**Description**

Provides the common header structure for all pages.

**Features**

* **HTML Head Section**:
  + Includes meta tags for character set, viewport, and compatibility.
  + Links to the main CSS file and Font Awesome for icons.

**4. index.php**

**Description**

Handles user registration.

**Features**

* **Registration Form**:
  + Collects user details such as first name, last name, email, password, and profile image.
  + Uses enctype="multipart/form-data" to handle file uploads.
* **Error Display**:
  + Provides a section to display validation errors.
* **Script Integration**:
  + Includes pass-show-hide.js for password visibility toggling and signup.js for form validation and AJAX submission.

**5. login.php**

**Description**

Manages user login.

**Features**

* **Login Form**:
  + Collects user email and password.
  + Uses pass-show-hide.js to toggle password visibility.
* **Error Display**:
  + Provides a section to display login errors.
* **Script Integration**:
  + Includes pass-show-hide.js for password visibility and login.js for form handling and AJAX submission.

**6. users.php**

**Description**

Displays a list of users and manages user interactions.

**Features**

* **User Authentication**:
  + Checks if the user is logged in; redirects to login.php if not authenticated.
* **User Information**:
  + Displays the logged-in user's profile details.
* **User Search**:
  + Provides a search bar to filter users.
* **User List**:
  + Displays a list of users, dynamically populated by AJAX.
* **Script Integration**:
  + Includes users.js for managing user interactions, search, and list updates.

### Server files

**1. add\_members\_form.php**

* **Purpose: This file displays a form to select users to add to a specific group.**
* **Functionality:**
  + **Fetches and displays a list of all users with checkboxes.**
  + **Users can select multiple users to add to a group.**
  + **The group\_id is passed as a query parameter to identify which group the users will be added to.**
  + **Submits the form data to add-members.php.**

**2. add-members.php**

* **Purpose: Handles the addition of selected users to a group.**
* **Functionality:**
  + **Receives the group\_id and an array of user\_ids from the form submission.**
  + **Iterates through the selected user IDs and inserts each into the group\_members table.**
  + **Redirects to groups.php after successfully adding members.**

**3. config.php**

* **Purpose: Establishes a connection to the database.**
* **Functionality:**
  + **Contains database connection settings.**
  + **Connects to the MySQL database and checks the connection status.**

**4. data.php**

* **Purpose: Generates HTML output for displaying messages and user statuses.**
* **Functionality:**
  + **Fetches the most recent message between users.**
  + **Formats messages and displays user status (online/offline).**
  + **Includes a delete button for each user.**

**5. delete-user.php**

* **Purpose: Deletes a user from the database.**
* **Functionality:**
  + **Receives a user\_id and deletes the corresponding user from the users table.**
  + **Provides feedback on whether the deletion was successful.**

**6. get-chat.php**

* **Purpose: Retrieves chat messages between two users.**
* **Functionality:**
  + **Fetches and displays all messages exchanged between the current user and another user.**
  + **Formats messages based on the sender (outgoing or incoming).**

**7. insert-chat.php**

* **Purpose: Inserts a new chat message into the database.**
* **Functionality:**
  + **Receives the message content and the incoming\_id (recipient) from a form submission.**
  + **Inserts the message into the messages table if the message is not empty.**

**8. login.php**

* **Purpose: Handles user authentication.**
* **Functionality:**
  + **Validates user credentials (email and password) against the database.**
  + **Updates the user's status to "Active now" upon successful login.**
  + **Starts a session for the logged-in user.**

**9. logout.php**

* **Purpose: Logs out the current user.**
* **Functionality:**
  + **Updates the user’s status to "Offline now".**
  + **Destroys the user session and redirects to the login page.**

**10. search.php**

* **Purpose: Searches for users based on a search term.**
* **Functionality:**
  + **Receives a search term and queries the users table for matches.**
  + **Excludes the current user from the search results.**
  + **Displays search results or a message if no users are found.**

**11. signup.php**

* **Purpose: Handles user registration.**
* **Functionality:**
  + **Receives user information and an uploaded image.**
  + **Validates and processes the registration data.**
  + **Checks for duplicate email addresses.**
  + **Uploads the user's image and inserts the new user into the users table.**
  + **Starts a session for the newly registered user.**

**12. users.php**

* **Purpose: Displays a list of users excluding the current user.**
* **Functionality:**
  + **Fetches and displays users from the users table.**
  + **Includes user details and a link to chat with each user.**
  + **Uses data.php for generating the output.**

# Database Description

**1. Introduction**

The website uses MySQL database to support a chat application. This section describes the database schema, tables, relationships, and any special considerations relevant to the implementation of the chat system.

**2. Database Overview**

**Database Name:** chat

This database contains tables to manage users and messages within the chat application. The structure supports user authentication, message exchange, and user management functionalities.

**3. Tables and Schema**

**3.1. users Table**

* **Description:** Stores user information including unique ID, name, email, password, image, status, and creation timestamp.

**Columns:**

* + user\_id (INT, Primary Key, Auto Increment): Unique identifier for each user.
  + unique\_id (INT, Unique Key): Unique identifier used to link messages to users.
  + fname (VARCHAR(255)): User's first name.
  + lname (VARCHAR(255)): User's last name.
  + email (VARCHAR(255)): User's email address.
  + password (VARCHAR(255)): User's hashed password.
  + img (VARCHAR(255)): Path to user's profile image.
  + status (VARCHAR(50)): Current status of the user (e.g., 'Online', 'Offline now').
  + created\_at (TIMESTAMP): Timestamp when the user was created.

**Indexes:**

* + Primary Key on user\_id.
  + Unique Key on unique\_id.

**3.2. messages Table**

* **Description:** Stores chat messages between users.

**Columns:**

* + msg\_id (INT, Primary Key, Auto Increment): Unique identifier for each message.
  + incoming\_msg\_id (INT, Foreign Key): ID of the user receiving the message, references users.unique\_id.
  + outgoing\_msg\_id (INT, Foreign Key): ID of the user sending the message, references users.unique\_id.
  + msg (TEXT): The message content.
  + created\_at (TIMESTAMP): Timestamp when the message was created.

**Indexes:**

* + Primary Key on msg\_id.
  + Indexes on incoming\_msg\_id and outgoing\_msg\_id.

**Constraints:**

* + Foreign Key constraint on incoming\_msg\_id references users(unique\_id).
  + Foreign Key constraint on outgoing\_msg\_id references users(unique\_id).

**4. Relationships**

* **User to Message:**
  + One-to-Many relationship from users to messages.
  + A user can send multiple messages and receive multiple messages.
  + This relationship is implemented through the incoming\_msg\_id and outgoing\_msg\_id foreign keys in the messages table.

**5. Special Coding Considerations**

* **Data Integrity:**
  + Foreign key constraints are used to ensure that messages are only linked to existing users.
  + ON DELETE CASCADE is used to remove messages if a user is deleted.
* **Indexes:**
  + Primary keys and indexes are created to enhance query performance, particularly for searching and sorting operations.
* **Auto-Increment:**
  + Auto-increment is used for primary keys to automatically generate unique IDs for new records.

**6. SQL Scripts**

* **DDL (Data Definition Language) Scripts:**
  + schema.sql: Defines the structure of tables and constraints.
* **DML (Data Manipulation Language) Scripts:**
  + insert-data.sql: Contains sample data for testing.
* **Triggers and Procedures:**
  + Not applicable in this dump, but should be documented if used in the future.

# Task Breakdown

**1. Project Setup and Configuration**

* **Tasks:**
  + Set up the project repository.
  + Configure the development environment.
  + Set up the database schema and connection configuration.
* **Assigned to:** Abdirahman Ali

**2. User Authentication**

* **Tasks:**
  + Create and implement user registration (signup) functionality.
  + Implement user login and logout functionality.
  + Secure password handling and session management.
* **Assigned to:** Matthew Augi

**3. Chat Functionality**

* **Tasks:**
  + Develop the chat interface (chat.php) and ensure real-time messaging.
  + Implement chat history management.
  + Handle sending and receiving messages via AJAX.
* **Assigned to:** Abdul Abdu

**4. User Management**

* **Tasks:**
  + Create user management functionalities (view users, delete users).
  + Implement user search functionality.
  + Design user profile pages.
* **Assigned to:** Abdirahman Ali

**5. Form and Input Handling**

* **Tasks:**
  + Develop forms for user signup and chat message input.
  + Handle form validation and error messaging.
  + Manage file uploads (e.g., user profile images).
* **Assigned to:** Matthew Augi

**6. UI/UX Design**

* **Tasks:**
  + Design and implement the website's layout and styling (CSS).
  + Ensure responsiveness and cross-browser compatibility.
  + Create a user-friendly interface for all pages.
* **Assigned to:** Abdul Abdu

**7. JavaScript and AJAX Integration**

* **Tasks:**
  + Implement JavaScript for interactive elements (e.g., show/hide password, real-time chat updates).
  + Integrate AJAX for dynamic content loading (e.g., user list, chat messages).
* **Assigned to:** Abdirahman Ali

**8. Error Handling and Debugging**

* **Tasks:**
  + Test and debug the entire application.
  + Implement error handling and logging mechanisms.
  + Ensure all functionalities work seamlessly.
* **Assigned to:** Matthew Augi

**9. Documentation and Final Touches**

* **Tasks:**
  + Create user guides and technical documentation.
  + Review and finalize code.
  + Prepare the project for deployment.
* **Assigned to:** Abdul Abdu

Editing:

| **File** | **And who it was Assigned To** |
| --- | --- |

|  |  |
| --- | --- |
| chat.php | Abdul Abdu |

|  |  |
| --- | --- |
| delete.php | Matthew Augi |

|  |  |
| --- | --- |
| header.php | Abdirahman Ali |

|  |  |
| --- | --- |
| index.php | Matthew Augi |

|  |  |
| --- | --- |
| login.php | Abdul Abdu |

|  |  |
| --- | --- |
| users.php | Abdirahman Ali |

|  |  |
| --- | --- |
| add\_members\_form.php | Matthew Augi |

|  |  |
| --- | --- |
| add-members.php | Abdul Abdu |

|  |  |
| --- | --- |
| config.php | Abdirahman Ali |

|  |  |
| --- | --- |
| data.php | Abdul Abdu |

|  |  |
| --- | --- |
| delete-user.php | Matthew Augi |

|  |  |
| --- | --- |
| get-chat.php | Abdul Abdu |

|  |  |
| --- | --- |
| insert-chat.php | Abdirahman Ali |

|  |  |
| --- | --- |
| logout.php | Matthew Augi |

|  |  |
| --- | --- |
| search.php | Abdul Abdu |

|  |  |
| --- | --- |
| signup.php | Abdirahman Ali |

|  |  |
| --- | --- |
| users.js | Abdul Abdu |

|  |  |
| --- | --- |
| chat.js | Matthew Augi |

|  |  |
| --- | --- |
| pass-show-hide.js | Abdirahman Ali |

|  |  |
| --- | --- |
| signup.js | Abdul Abdu |

|  |  |
| --- | --- |
| login.js | Matthew Augi |