**Functionality Guide**

This task management web application allows users to manage tasks through an intuitive interface. The application includes features such as user registration, login, task creation, task filtering, and task deletion.

**Features**

1. **User Registration**:
   * Users can create an account by providing a username and password.
   * Registration form includes client-side validation.
2. **User Login**:
   * Registered users can log in using their username and password.
   * Upon successful login, users are redirected to the tasks page.
3. **Task Management**:
   * Users can add tasks with a title, description, priority, and due date.
   * Tasks can be filtered based on priority and due date.
   * Users can remove tasks.
   * Tasks are displayed in a list format with real-time updates.
4. **Responsive Design**:
   * The application is designed to be responsive, ensuring usability on various devices.

**Description of the Database**

The database for this application is MySQL, which consists of two main tables: users and tasks.

**Users Table**

* **Table Name**: users
* **Columns**:
  + id (INT, Primary Key, Auto Increment): Unique identifier for each user.
  + username (VARCHAR(255), Unique, Not Null): Username of the user.
  + password (VARCHAR(255), Not Null): Password of the user (hashed).

**Tasks Table**

* **Table Name**: tasks
* **Columns**:
  + id (INT, Primary Key, Auto Increment): Unique identifier for each task.
  + user\_id (INT, Foreign Key): Identifier linking the task to a user.
  + title (VARCHAR(255), Not Null): Title of the task.
  + description (TEXT): Description of the task.
  + priority (ENUM('Low', 'Medium', 'High'), Not Null): Priority of the task.
  + due\_date (DATE, Not Null): Due date of the task.
  + status (ENUM('Pending', 'Completed'), Default 'Pending'): Status of the task.

**Special Coding Considerations**

1. **Security**:
   * Passwords are hashed using PHP’s password\_hash function before storing them in the database.
   * SQL queries are prepared statements to prevent SQL injection.
   * User input is validated both on the client-side (JavaScript) and server-side (PHP).
2. **Session Management**:
   * PHP sessions are used to manage user login states.
   * Users must be logged in to view and manage tasks.
3. **Error Handling**:
   * Proper error messages and validations are implemented for user registration and login.
   * Task management actions provide feedback to the user (e.g., task added, task removed).
4. **Responsive Design**:
   * CSS media queries ensure the application is usable on different screen sizes.
   * Flexbox is used for layout adjustments.