**Task management web project**

**Functionality guide**

**Backend Database Functionality**

**Task Management Operations**: The backend supports the following basic CRUD (Create, Read, Update, Delete) operations for tasks.

**Add Task**: Users can add new tasks with details including title, description, priority, due date, and status.

**Display Task**: Users can view their tasks. Tasks are displayed in a list.

**Remove Task**: Users have the option to delete tasks.

**Search and filtering**

**Task Search**: Users can search for tasks using keywords or phrases.

**Filtering** **Options**: Users can filter tasks based on:

**Priority**: (different layers of priority can be selected)

**Due Date**: Specific dates or date ranges (represented by numbers 1-9).

**Status**: Completed, or not(represented by numbers 0 and 1).

**User Registration Form**

**Account** **Creation**: new users can create an account.

**Form** **Fields**: Include fields username, first name, last name, email, password, and confirm password.

**Submission**: A submit button to send form data to the server.

**HTML Structure**

**Semantic** **HTML**: Use HTML5 semantic elements to structure the task details and listings, ensuring accessibility and SEO-friendly markup.

**Form Validation**

If the user typed wrong information such as email without correct format, an error message will appear reminding the user to enter correct input.

If the second input of the password does not match the first input, an error message will appear showing that the password does not match.

**Dynamic Behavior**

**Task** **Prioritization**: Users can set or change the priority of tasks.

Completion Status: Users can mark tasks as complete, in-progress, or pending.

**Responsiveness**

The user interface can adapt to different screen sizes and devices, and is fully usable on mobile devices.

**Tables and relationships**

A screenshot of a computer

Description automatically generated

* **Relationship**

Users - Tasks (One-to-Many):

Each user (in the users table) can be associated with multiple tasks.

The user\_id in the tasks table is a foreign key referencing the user\_id primary key in the users table.The foreign key constraint (tasks\_ibfk\_1) ensures that the user\_id in the tasks table must correspond to an existing user\_id in the users table. The user\_id in the tasks table represents the user to whom a task is assigned.The relationships allow for linking tasks to the users who created them or are responsible for their completion. The tasks\_ibfk\_1 constraint ensures that a task cannot be assigned to a non-existing user.