

# **PERSONAL TASK MANAGEMENT**

## **A Mini Project Report**

**Submitted by**

**DIVYA DHARSHINI S**

**23CDR044**

**HEMAPRIYA V S**

**23CDR058**

**22CDT41 - DATABASE MANAGEMENT SYSTEMS**



**DEPARTMENT OF COMPUTER SCIENCE AND DESIGN  
KONGU ENGINEERING COLLEGE**

**(Autonomous)**

**PERUNDURAI ERODE-638060**

**MAY 2025**

## **Problem Statement:**

Personal Task Manager with Categories: Enables users to manage their personal tasks with categorization and due dates. (RDBMS: Tasks, Categories; Backend: Task creation/editing, categorization, due date management).

## **Github Link: <https://github.com/Divya0804Dharshini/Personal-Task-Management>**

<https://github.com/HE-MA/Personal-Task-Management>

## **Modules and Description:**

### **User Registration & Login:**

Secure signup and login system to manage personal task data.

### **Task Management:**

Add new tasks with a title, description, due date, priority (Low, Medium, High), and category.

Edit and update existing tasks.

Delete tasks when completed or no longer needed.

### **Category :**

Tasks can be sorted or filtered by default categories: *Personal, Work, College*.

### **User Dashboard:**

Personalized view of all tasks with options to sort by date, priority, or category.

### **Responsive Design:**

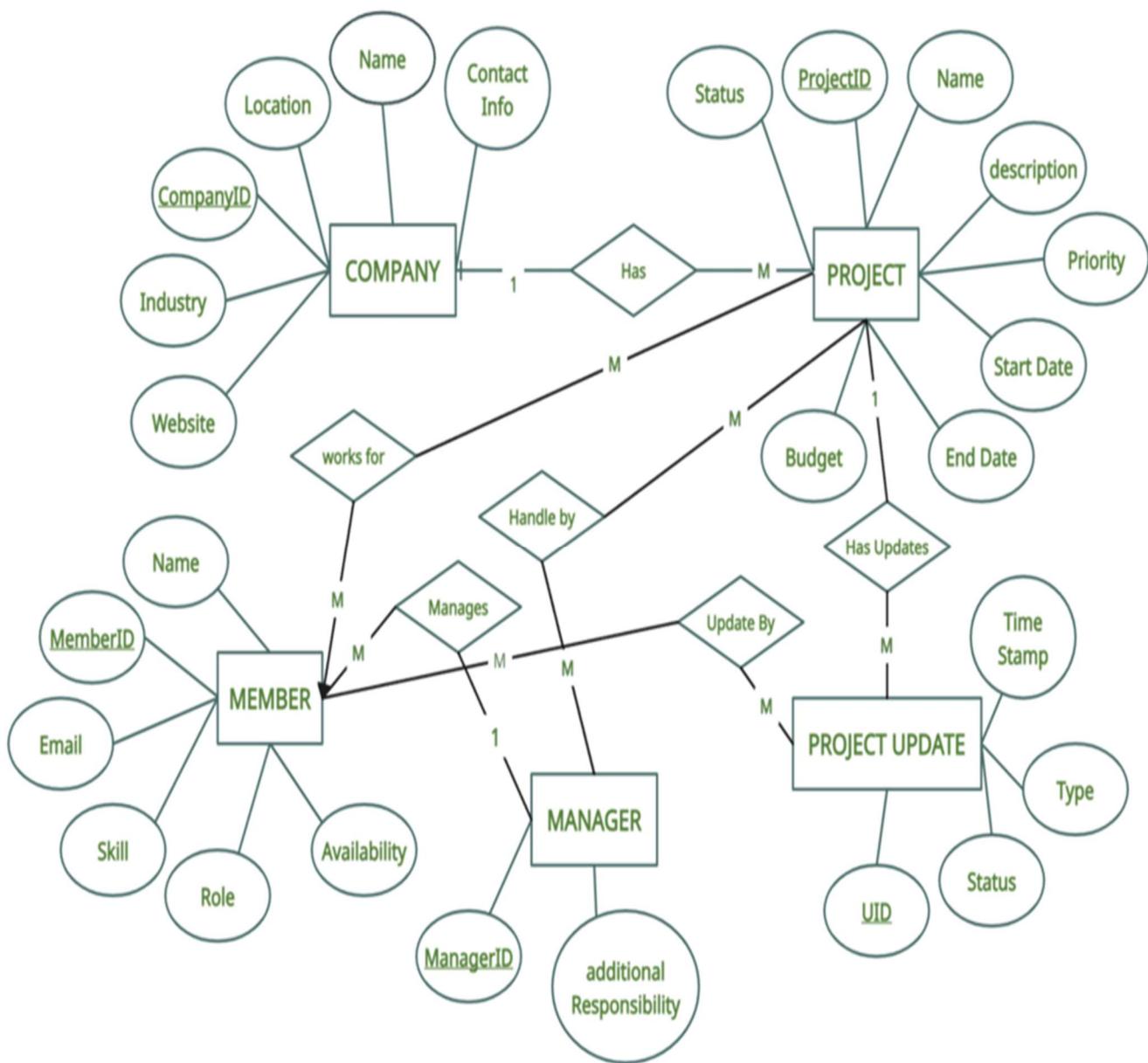
Fully responsive UI for desktop and mobile browsers.

## ER Diagram:

### Relationship      Type      Description

Users → Tasks      One-to-Many      A user can have many tasks

Categories → Tasks      One-to-Many      A category can be assigned to many tasks



## **Programming Tools:**

### **1. Frontend Tools (User Interface)**

These tools define the layout, styling, and user interaction for your app.

#### **HTML (HyperText Markup Language)**

- **Purpose:** Structure of the web pages (e.g., login form, task list).
- **Usage Examples:**
  - Forms for login, signup, and task input (<form>)
  - Page layout using <div>, <header>, <footer>, etc.

#### **CSS (Cascading Style Sheets)**

- **Purpose:** Visual styling (fonts, colors, spacing, layout).
- **Usage Examples:**
  - Style for header/footer navigation
  - Task cards with different colors for priority levels
  - Responsive design with flexbox or grid

### **2. Backend Tools (Server-side Logic)**

This layer processes user data, connects to the database, and serves responses.

#### **PHP (Hypertext Preprocessor)**

- **Purpose:** Server-side scripting to handle logic and database operations.
- **Usage Examples:**
  - login.php and signup.php to authenticate users
  - handle\_add\_task.php to save tasks

- delete\_task.php, edit\_task.php, etc.

## XAMPP

- **Purpose:** Local web server stack that includes:
  - **Apache:** Web server to run your PHP files
  - **MySQL:** Relational database server
  - **phpMyAdmin:** GUI for managing MySQL
- **Usage:** Run your entire web app locally and manage your database.

### 3. Database Tools

This layer stores and manages your persistent data (tasks, users, etc.).

#### MySQL

- **Purpose:** Store and manage structured data in tables.
- **Your Tables:**
  - users: Stores user credentials
  - categories: Stores task categories (e.g., Personal, Work, College)
  - tasks: Stores task details (title, due date, priority, category, user ID)

#### Database Design:

Table	Action	Rows	Type	Collation	Size	Overhead
categories	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 Kib	-
tasks	★ Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	48.0 Kib	-
users	★ Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	32.0 Kib	-
3 tables	Sum			21	InnoDB utf8mb4_general_ci 96.0 Kib	0 B
	Check all	With selected:	v			

## Categories table:

	<input type="checkbox"/>		Edit		Copy		Delete	<b>id</b>	<b>name</b>
	<input type="checkbox"/>		Edit		Copy		Delete	1	Personal
	<input type="checkbox"/>		Edit		Copy		Delete	2	Work
	<input type="checkbox"/>		Edit		Copy		Delete	3	College

Check all      With selected: Edit

## Tasks table:

	<input type="checkbox"/>		Edit		Copy		Delete	<b>id</b>	<b>title</b>	<b>due_date</b>	<b>priority</b>	<b>category_id</b>	<b>user_id</b>
	<input type="checkbox"/>		Edit		Copy		Delete	2	dbms project	2025-05-03	High	2	NULL
	<input type="checkbox"/>		Edit		Copy		Delete	3	cooking	2025-05-22	High	2	NULL
	<input type="checkbox"/>		Edit		Copy		Delete	4	walking	2025-05-09	High	1	2
	<input type="checkbox"/>		Edit		Copy		Delete	5	cooking	2025-05-10	Medium	1	2
	<input type="checkbox"/>		Edit		Copy		Delete	6	df	2025-05-10	Medium	1	1

## Users table:

	<input type="button" value="T"/>	<input type="button" value="id"/>	username	password
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1 divya@gmail.com \$2y\$10\$wQV1P4pnO4sKVH6WFEoAZuO3CAk/L9bLPKrumh04sAW...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2 divya \$2y\$10\$do91oNXXlsoXPSeDj7l20kKZbCUm7A4oet22iyBmPi...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3 hema \$2y\$10\$RZSsd/yPQv.eYAGQc1LVYeXbXfv35sVPlf4TM66h/R/...

## SQL Query:

```
DROP TABLE IF EXISTS tasks;
DROP TABLE IF EXISTS categories;
DROP TABLE IF EXISTS users;

-- Recreate all tables
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL
);

CREATE TABLE categories (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) NOT NULL
);

CREATE TABLE tasks (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    due_date DATE,
    priority ENUM('Low', 'Medium', 'High') DEFAULT 'Medium',
    category_id INT,
    user_id INT,
    FOREIGN KEY (category_id) REFERENCES categories(id) ON DELETE SET
NULL,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);

-- Insert default categories
INSERT INTO categories (name) VALUES ('Personal'), ('Work'), ('College');
```

## Coding:

### Add\_task.php:

```
<?php
session_start();
include
'auth.php';
include 'db.php';
include
'header.php';

$categories = $conn->query("SELECT id, name FROM categories");
?>

<main>
<h2>Add New Task</h2>
<form method="POST" action="handle_add_task.php">
<input type="text" name="title" placeholder="Title" required>
<label>Priority:</label>
<select name="priority" required>
<option value="Low">Low</option>
<option value="Medium" selected>Medium</option>
<option value="High">High</option>
</select>
<label>Category:</label>
<select name="category_id" required>
<?php while($cat = $categories->fetch_assoc()): ?>
<option value=<?= $cat['id'] ?><?= htmlspecialchars($cat['name']) ?></option>
<?php endwhile; ?>
```

```
</select>

<input type="date" name="due_date" required>

<button type="submit">Add Task</button>

</form>

</main>
```

## **Auth.php:**

```
<?php

if

(!isset($_SESSION['user_ id'])) { header("Location: login.php"); exit;

}

?
```

## **Db.php:**

```
<?php

$host = "localhost";

$user = "root";

$pass = "";

$dbname = "task_manager";

$conn = new mysqli($host, $user, $pass, $dbname); if ($conn->connect_error) {

die("Connection failed: " . $conn->connect_error);

}

?>
```

## **Delete\_task.php:**

```
<?php
```

```
session_start();

include 'auth.php';

include 'db.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    $task_id = intval($_POST['task_id']);

    $user_id = $_SESSION['user_id'];
    $stmt = $conn->prepare("DELETE FROM tasks WHERE id = ? AND user_id = ?");

    $stmt->bind_param("ii", $task_id, $user_id);

    $stmt->execute();

}

header("Location:

index.php"); exit;

?>
```

## Edit\_task.php:

```
<?php

session_start();

include 'auth.php';

include 'db.php';

$task_id = intval($_GET['id']);

$user_id = $_SESSION['user_id'];

$stmt = $conn->prepare("SELECT title, due_date, priority, category_id FROM tasks
WHERE id = ? AND user_id = ?");

$stmt->bind_param("ii", $task_id, $user_id);

$stmt->execute();

$stmt->bind_result($title, $due_date, $priority,
$category_id); if (!$stmt->fetch()) {

echo "Task not

found!"; exit;

} $stmt->close();
```

```

$categories = $conn->query("SELECT id, name FROM
categories"); include 'header.php';
?>
<main>
<h2>Edit Task</h2>
<form method="POST">
<input type="text" name="title" value="<?= htmlspecialchars($title) ?>" required>
<label>Priority:</label>
<select name="priority" required>
<option value="Low" <?= $priority == 'Low' ? 'selected' : " ?>>Low</option>
<option value="Medium" <?= $priority == 'Medium' ? 'selected' : "
?>>Medium</option>
<option value="High" <?= $priority == 'High' ? 'selected' : " ?>>High</option>
</select>
<label>Category:</label>
<select name="category_id" required>
<?php while($cat = $categories->fetch_assoc()): ?>
<option value="<?= $cat['id'] ?>" <?= $cat['id'] == $category_id ?
'selected' : " ?>><?= htmlspecialchars($cat['name']) ?></option>
<?php endwhile; ?>
</select>
<input type="date" name="due_date" value="<?= $due_date ?>" required>
<button type="submit">Update Task</button>
</form>
</main>
<?php
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $new_title = trim($_POST['title']);
    $new_priority = $_POST['priority'];
    $new_category_id = intval($_POST['category_id']);
}

```

```

$new_due_date = $_POST['due_date'];

$stmt = $conn->prepare("UPDATE tasks SET title=?, due_date=?, priority=?,
category_id=? WHERE id=? AND user_id=?");

$stmt->bind_param("sssi", $new_title, $new_due_date, $new_priority,
$new_category_id,
$task_id, $user_id);

$stmt->execute();

header("Location:
index.php"); exit;

}

?>

```

## **Handle\_add\_task.php:**

```

<?php

session_start();

include 'auth.php';

include 'db.php';

$title = trim($_POST['title']);

$priority = $_POST['priority'];

$category_id = intval($_POST['category_id']);

$due_date = $_POST['due_date'];

$user_id = $_SESSION['user_id'];

$stmt = $conn->prepare("INSERT INTO tasks (title, due_date, priority, category_id,
user_id) VALUES (?, ?, ?, ?, ?)");

$stmt->bind_param("sssi", $title, $due_date, $priority, $category_id, $user_id);

$stmt->execute();

header("Location:
index.php"); exit;

?>

```

## **Header.php:**

```
<?php
```

```

session_start(); // It's generally good practice to start the session at the top
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Personal Task Manager</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<header>
    <h1>Personal Task Manager</h1>
    <nav>
        <a href="index.php">Home</a>
        <?php if (isset($_SESSION['user_id'])): ?>
            <a href="add_task.php">Add Task</a>
            <a href="logout.php">Logout</a>
        <?php else: ?>
            <a href="login.php">Login</a>
            <a href="signup.php">Sign Up</a>
        <?php endif; ?>
    </nav>
</header>

```

## **Index.php:**

```

<?php
session_start();
include
'auth.php';

```

```

include 'db.php';

include

'header.php';

$user_id = $_SESSION['user_id'];

$sql = "SELECT tasks.*, categories.name AS category_name

        FROM tasks LEFT JOIN categories ON

        tasks.category_id = categories.id

        WHERE tasks.user_id = ? ORDER BY due_date";

$stmt = $conn->prepare($sql);

$stmt->bind_param("i", $user_id);

$stmt->execute();

$result = $stmt->get_result();

?>

<main>

<h2>Your Tasks</h2>

<ul class="task-list">
    <?php while($row = $result->fetch_assoc()): ?>
        <li>
            <strong><?= htmlspecialchars($row['title']) ?><br> Priority: <?= $row['priority'] ?><br>
            Category: <?= htmlspecialchars($row['category_name']) ?><br> Due:
            <?= $row['due_date'] ?><br>
            <a href="edit_task.php?id=<?= $row['id'] ?>">Edit</a>
            <form method="POST" action="delete_task.php" style="display:inline;">
                <input type="hidden" name="task_id" value="<?= $row['id'] ?>">
                <button type="submit" onclick="return confirm('Delete this task?')">Delete</button>
        </li>
    <?php endwhile; ?>
</ul>

```

```
</form>
</li>
<?php endwhile; ?>
</ul>
</main>

Login.php:
<?php
include
'db.php';
session_start()
;
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $username = trim($_POST['username']);
    $password = $_POST['password'];
    $stmt = $conn->prepare("SELECT id, password FROM users WHERE username =
    ?");
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $stmt->bind_result($user_id, $hash);
    if ($stmt->fetch() && password_verify($password, $hash)) {
        $_SESSION['user_id'] = $user_id;
        header("Location: index.php");
    } else {
        echo "Invalid login!";
    }
}
include 'header.php';
?>
<main>
```

```
<h2>Login</h2>
<form method="POST">
    <input type="text" name="username" placeholder="Username" required>
    <input type="password" name="password" placeholder="Password" required>
    <button type="submit">Login</button>
</form>
</main>
```

### **Logout.php:**

```
<?php
session_start();
session_destroy();
header("Location:
login.php"); exit;
?>
```

### **Signup.php:**

```
<?php
include 'db.php';
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $username = trim($_POST['username']);
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT);
    $stmt = $conn->prepare("INSERT INTO users (username, password) VALUES (?, ?)");
    $stmt->bind_param("ss", $username, $password); if ($stmt->execute()) {
        header("Location: login.php");
    } else {
        echo "Error: Username may already exist.";
    }
}
```

```
include 'header.php';

?>

<main>

    <h2>Sign Up</h2>

    <form method="POST">

        <input type="text" name="username" placeholder="Username" required>
        <input type="password" name="password" placeholder="Password" required>
        <button type="submit">Register</button>
    </form>
</main>
```

### Style.css:

```
body {

    font-family: 'Segoe UI', Tahoma, Geneva, Verdana,
    sans-serif; background-color: #f0f6ff;
    margin: 0;
    padding: 0;
}

header {

    background-color: #007bff;
    color: white;
    padding: 20px
    0; text-align:
    center;
}

header h1

{
    margin:
    0;
```

```
        font-size: 28px;
    }

nav {
    margin-top: 10px;
}

nav a {
    color: white;
    text-decoration: none;
    margin: 0 15px;
    font-weight: bold;
    transition: color 0.3s;
}

nav a:hover {
    color: #dfefff;
}

main {
    padding: 20px;
    max-width: 800px;
    margin: auto;
    background-color: white;
    margin-top: 20px;
    border-radius: 8px;
    box-shadow: 0 2px 8px rgba(0, 123, 255, 0.2);
}

form {
    display: flex;
    flex-direction: column;
    gap: 12px;
```

```
 }input[type="text"],  
 input[type="password"  
"],  
 input[type="date"],  
 select {  
 padding: 10px;  
 border: 1px solid  
 #ccc;  
 border-radius: 5px;  
 }button {  
 padding:  
 10px;  
 background-color: #007bff;  
 color: white;  
 border: none;  
 border-radius:  
 5px; font-  
 weight: bold;  
 cursor: pointer;  
 transition: background-color 0.3s;  
 }button:hover {  
 background-color: #0056b3;  
 }.task-list {  
 list-style:  
 none;  
 padding: 0;  
 }.task-list li {  
 background-color:
```

```
#e9f3ff; border:  
1px solid #b3d7ff;  
border-radius: 6px;  
margin-bottom:  
10px; padding:  
15px;  
}.task-list li  
form {  
display:  
inline;  
}.task-list a {  
margin-right:  
10px; text-  
decoration:  
none; color:  
#007bff;  
font-weight: bold;  
}.task-list a:hover {  
text-decoration: underline;  
}
```

## Output:

Personal Task Manager

Home Login Sign Up

Login

Username

Password

Login

Personal Task Manager

Home Login Sign Up

Login

Username

Password

Please fill out this field.

Login

# Personal Task Manager

[Home](#)   [Login](#)   [Sign Up](#)

## Sign Up

Username

Password

Register

# Personal Task Manager

[Home](#)   [Add Task](#)   [Logout](#)

## Add New Task

Title

Priority:

Medium

Category:

Personal

dd-mm-yyyy



Add Task

# Personal Task Manager

[Home](#) [Add Task](#) [Logout](#)

## Your Tasks

df  
Priority: Medium  
Category: Personal  
Due: 2025-05-10

[Edit](#)

Delete

May, 2025 ▾

↑ ↓

Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	<b>13</b>	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

[Clear](#)

[Today](#)

dd - mm - yyyy



[Add Task](#)

## **Reference:**

1. J. K. Author, Database Management Systems, 7th ed. Boston, MA, USA: McGraw-Hill, 2020.
2. PHP: A beginner's guide
3. Learning Web Design