

National School of Sciences
Lainchaur, Kathmandu



TO DO LIST

Submitted by:

TEAM A

ANKIT KHATRI KC (719)

PRATIK SHRESTHA (708)

SUMIKA MANANDHAR (699)

KRISH GURUNG (704)

ABIK GURUNG (783)

Submitted to:

Department of Computer Science

National School of Sciences

ACKNOWLEDGEMENT

It is our good fortune to get very cooperative teacher Mr. Santosh Bhat. During the work in progress, his support, helpfulness and constant encouragement kept us motivated in research work. He provided invaluable interest, guidance during the course of the work. We have not only learned but also got important suggestions regarding scientific writing and other related matters. We are very much grateful to him.

We would like to thank all the faculty members who have provided us encouragement and suggestions during the course of work and special thanks to lab assistants.

TEAM A

CERTIFICATE OF APPROVAL

It is certified that TEAM A has carried out the project work entitled “TO DO LIST”.

This project is the result of their endeavors and research. It is finalized under our guidance and supervision in the academic year 2023-2024.

.....

Head of Department

Er. Sabin Raj Bagale

Department of Computer Science
National School of Sciences

Lainchaur, Kathmandu

.....

Supervisor

Mr. Santosh Bhat

Department of Computer Science
National School of Sciences

Lainchaur, Kathmandu

.....

External, NEB

TABLE OF CONTENT

<u>Topics</u>	<u>Page No.</u>
Introduction	1
Home page	2
Login page	3
Creating a new task	6
Read operation	8
Update operation	11
Delete operation	14
Additional Functions	15-18
• Completed operation	15
• Pending operation	17
Conclusion	19
Bibliography	20

INTRODUCTION

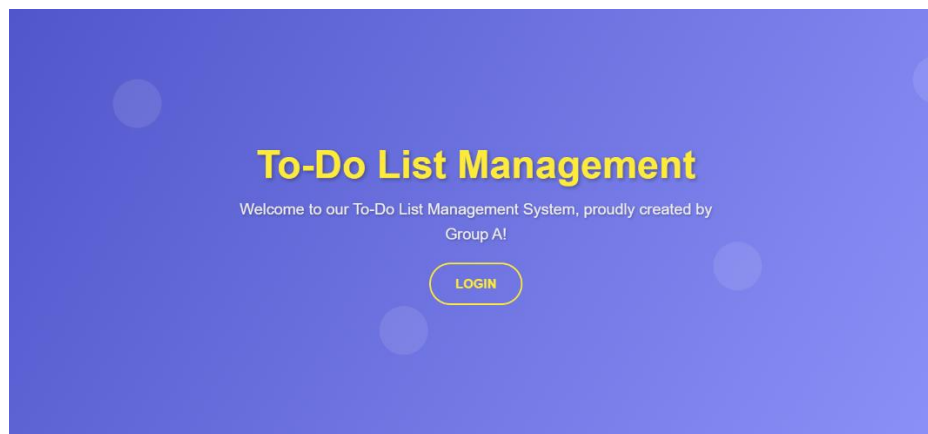
This project is designed to demonstrate the implementation of **CRUD** (Create, Read, Update, Delete) operations in a web or software application. The project serves as a basic framework for handling data management in a structured and efficient manner.

The To-Do List Project is a simple yet essential application that helps users manage their tasks and daily activities. It allows users to create and organize a list of tasks, each with details such as a title, description, and deadline. The goal is to provide a straightforward platform where users can track what they need to do, prioritize their activities, and stay productive. The project typically features a clean and user-friendly interface, making it easy to add new tasks, edit existing ones, and mark tasks as completed.

The To-Do List Project implements CRUD operations to manage tasks effectively. Users can **Create** tasks by adding new ones with details, **Read** tasks by viewing the entire list or filtered tasks, **Update** tasks by modifying their status, deadlines, or details, and **Delete** tasks when they are completed or no longer needed. These operations provide a smooth user experience for organizing, tracking, and managing tasks, enhancing productivity and task management.

HOME PAGE

The Home Page, acting as the dashboard of the To-Do List Project, provides users with an organized overview of their tasks. It features sections for All Tasks, where users can view both pending and completed tasks, and a New Task button for adding new entries. The Pending Tasks section displays tasks still in progress, while Completed Tasks showcases finished tasks. Additionally, users can securely log out through the Logout option. This dashboard design ensures a streamlined, efficient task management experience.

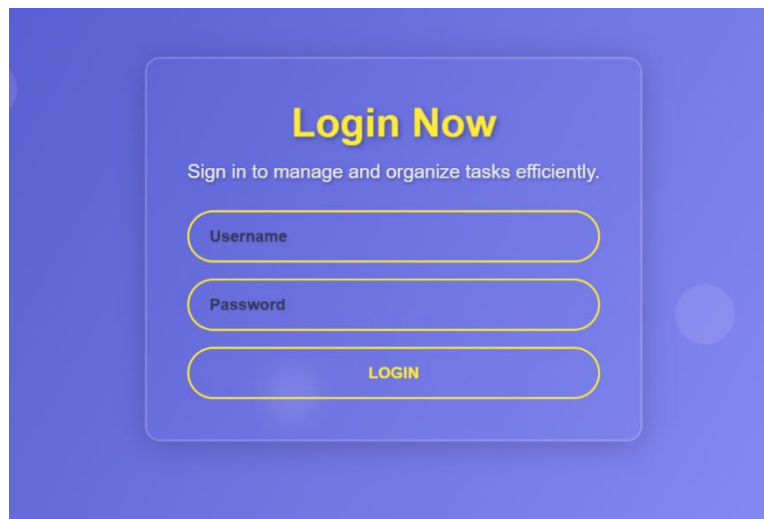


Code used:

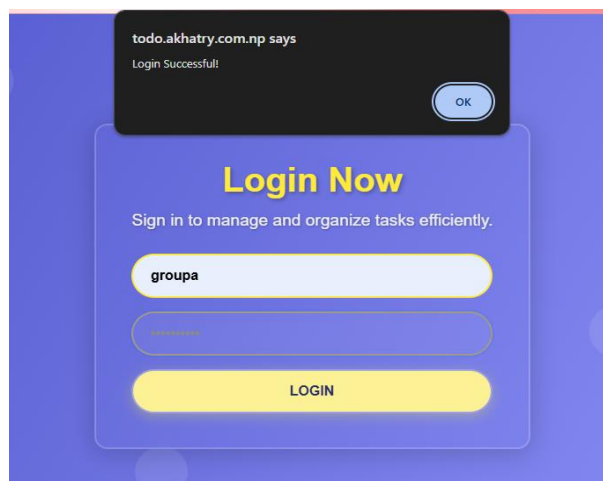
```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>To-Do List Management</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="style/index.css">
</head>
<body>
  <div class="hero">
    <div class="hero-content">
      <h1>To-Do List Management</h1>
      <p>Welcome to our To-Do List Management System, proudly created by Group A!</p>
      <div class="login-btn">
        <a href="login.php" class="btn btn-outline-light">Login</a>
      </div>
    </div>
  </div>
  <div class="bg-animation">
    <span></span>
    <span></span>
    <span></span>
    <span></span>
  </div>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"></script></body></html>
```

LOGIN PAGE

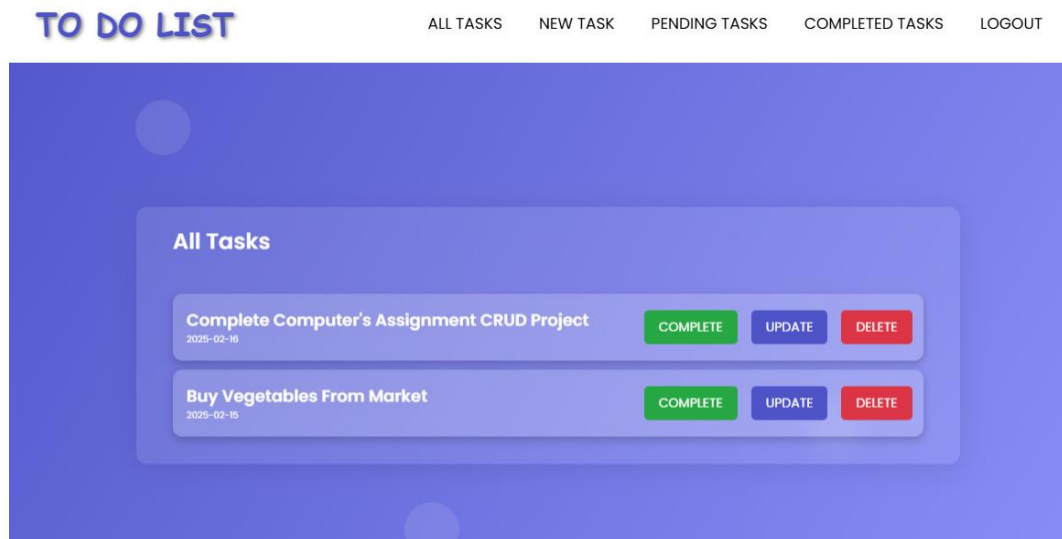
The Login Page of the To-Do List Project provides a simple and secure authentication process, allowing users to log in using their username and password. Once the correct credentials are entered, users gain access to their personalized task dashboard. The page is designed to ensure secure login and efficient navigation, providing a seamless experience for managing tasks. This straightforward login interface ensures that only authenticated users can access their task lists, enhancing security while maintaining simplicity.

A login form titled "Login Now" in yellow text. Below the title is the subtitle "Sign in to manage and organize tasks efficiently." in white. There are two input fields: "Username" and "Password", both with yellow borders. Below these fields is a yellow "LOGIN" button. The entire form is centered on a blue background with subtle circular patterns.

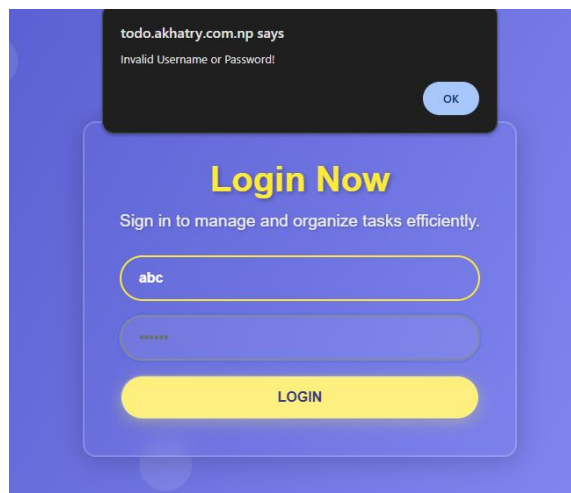
- **For valid inputs:**

The same login form as above, but with a success message overlay. A black toast notification at the top left says "todo.akhatry.com.np says" and "Login Successful!". An "OK" button is in the top right of the toast. The "Username" field now contains the text "groupa". The "LOGIN" button is highlighted in yellow.

When valid details are entered on the Login Page; correct username and password that match the existing account, the system authenticates the user. Upon successful login, the user is granted access and immediately redirected to the main task dashboard page. This transition is seamless, and the user can start managing their tasks right away.



- **For invalid inputs**



When invalid details are provided, such as incorrect username or password, or if any fields are left empty, the system prevents access and displays an error message. The user remains on the login page, and the error message prompts them to re-enter the correct credentials. This ensures that only authorized users can access the main page, maintaining security and preventing unauthorized access.

Code used:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login Now</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="style/index.css">
```



```

</head>
<body>
  <div class="login">
    <div class="login-content">
      <h1>Login Now</h1>
      <p>Sign in to manage and organize tasks efficiently.</p>

      <form class="mt-4" id="loginForm">
        <div class="mb-3">
          <input
            type="text"
            class="form-control form-control-lg"
            placeholder="Username"
            id="username"
            required>
        </div>
        <div class="mb-3">
          <input
            type="password"
            class="form-control form-control-lg"
            placeholder="Password"
            id="password"
            required>
        </div>
        <button type="submit" class="btn btn-outline-light w-100">Login</button>
      </form>
    </div>
  </div>

  <div class="bg-animation">
    <span></span>
    <span></span>
    <span></span>
    <span></span>
  </div>

  <script>
    document.getElementById("loginForm").addEventListener("submit", function(event) {
      event.preventDefault();

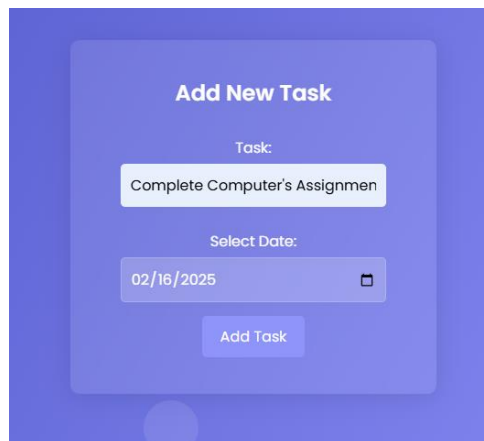
      const username = document.getElementById("username").value;
      const password = document.getElementById("password").value;
      if (username === "groupa" && password === "groupa@123") {
        alert("Login Successful!");
        window.location.href = "dashboard.php";
      } else {
        alert("Invalid Username or Password!");
      }
    });
  </script>









  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

CREATING A NEW TASK

Creating a new task in the To-Do List Project is a simple process. When users click on the New Task button, a form appears where they can enter the task title and select a due date. After submitting the form, the new task is added to the All-Tasks section on the Home Page, which displays a complete list of tasks. This allows users to view and manage all their tasks, including the newly added ones, in one central location.



		id	task	date	created_at	status
<input type="checkbox"/>	 Edit  Copy  Delete	35	Complete computer's assignment	2025-02-16	2025-02-14 21:48:46	0
	<input type="checkbox"/> Check all	With selected:		 Edit  Copy  Delete  Export		
<input type="checkbox"/> Show all	Number of rows:	25	Filter rows:	<input type="text" value="Search this table"/>		

Code used:

```
<?php
session_start();

if (isset($_GET['logout'])) {
    session_destroy();
    header("Location: login.php");
    exit();
}
include('connection.php');
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $task = $_POST['task'];
    $date = $_POST['date'];
    $sql = "INSERT INTO tasks (task, date) VALUES ('$task', '$date')";
    if ($conn->query($sql) === TRUE) {
        echo "<script>alert('Task added successfully!');</script>";
    } else {
        echo "<script>alert('Error adding task: " . $conn->error . "');</script>";
    }
}
$conn->close();
?>
```

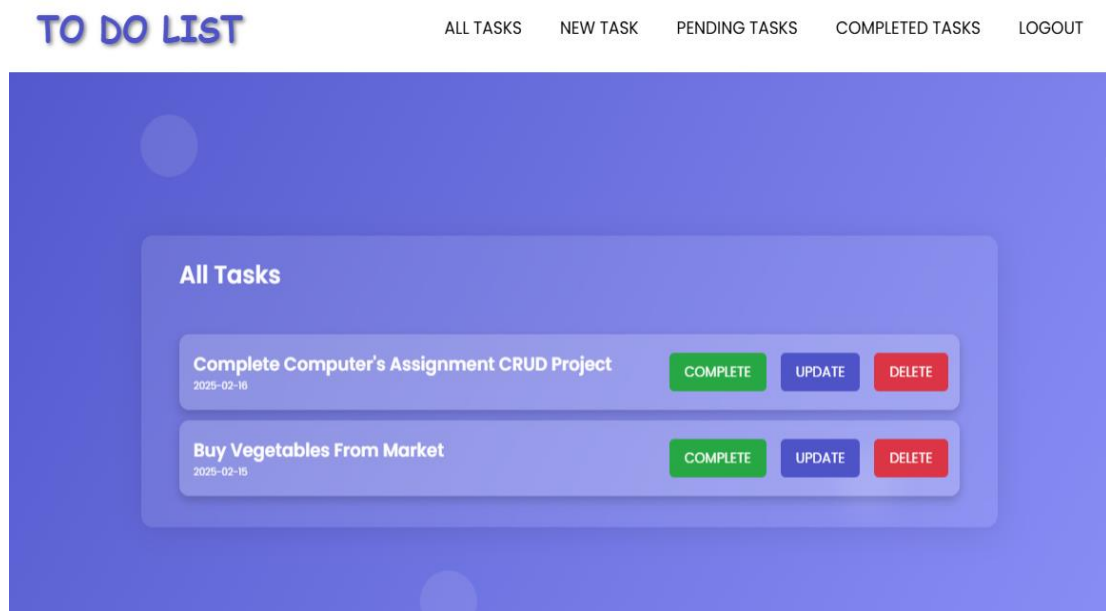
```

<html lang="en">
<head>
<title>New Task</title>
<link rel="stylesheet" href="style/dash.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-
icons.css">
</head>
<body>
  <header class="header">
    <a class="logo">TO DO LIST</a>
    <div id="menu-btn" class="fas fa-bars"></div>
    <nav class="navbar">
      <ul>
        <li><a href="dashboard.php">ALL TASKS</a></li>
        <li><a href="newtask.php">NEW TASK</a></li>
        <li><a href="pendingtasks.php">PENDING TASKS</a></li>
        <li><a href="completedtasks.php">COMPLETED TASKS</a></li>
        <li><a href="dashboard.php?logout=true">LOGOUT</a></li>
      </ul>
    </nav>
  </header>
  <div class="bg-animation">
    <span></span>
    <span></span>
    <span></span>
    <span></span>
  </div>
  <div class="task-form">
    <h2>Add New Task</h2>
    <form action="" method="POST">
      <label for="task">Task:</label>
      <input type="text" id="task" name="task" required>
      <label for="date">Select Date:</label>
      <input type="date" id="date" name="date" required>
      <button type="submit">Add Task</button>
    </form>
  </div>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
  alpha3/dist/js/bootstrap.bundle.min.js"></script>
  <script>
    let navbar = document.querySelector('.header .navbar');
    if (navbar) {
      document.querySelector('#menu-btn').onclick = () => {
        navbar.classList.toggle('active');
      }
      window.onscroll = () => {
        navbar.classList.remove('active');
      }
    } else {
      console.error("Navbar element not found.");
    }
  </script></body></html>

```

READ OPERATION

The All-Tasks page serves as the central hub for managing tasks within the To-Do List Project. Upon successful login, users are directed to this page, which displays a comprehensive list of all active tasks. Each task is clearly presented with its title and relevant details, allowing for quick scanning and prioritization. This page provides an intuitive interface for users to interact with their tasks, offering options to mark tasks as complete, update existing task information, or delete tasks that are no longer needed. The All-Tasks page is designed for efficient task management, providing a clear overview of pending activities and facilitating seamless updates to ensure productivity. This streamlined view of all tasks empowers users to stay organized and focused on their objectives.



		id	task	date	created_at	status
<input type="checkbox"/>	Edit Copy Delete	35	Complete computer's assignment CRUD Project	2025-02-16	2025-02-14 21:48:46	1
<input type="checkbox"/>	Edit Copy Delete	36	Buy Vegetables from market	2025-02-15	2025-02-14 21:49:34	0

☐ Check all With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

☐ Show all Number of rows: 25 Filter rows: Sort by key: None

Code used:

```
<?php
session_start();
if (isset($_GET['logout'])) {
    session_destroy();
    header("Location: login.php");
    exit();
}
include('connection.php');
if (isset($_GET['complete'])) {
    $task_id = $_GET['complete'];
    $conn->query("UPDATE tasks SET status = 1 WHERE id = $task_id");
    header("Location: dashboard.php");
    exit();
}
$sql = "SELECT * FROM tasks ORDER BY date DESC";
$result = $conn->query($sql);
if (isset($_GET['delete'])) {
    $task_id = $_GET['delete'];
    $conn->query("DELETE FROM tasks WHERE id = $task_id");
    header("Location: dashboard.php");
    exit();
}
$conn->close();
?>
```

```
<html lang="en">
<head>
<title>Dashboard</title>
<link rel="stylesheet" href="style/dash.css">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome/5.15.4/css/all.min.css">
</head>
<body>
<header class="header">
    <a class="logo">TO DO LIST</a>
    <div id="menu-btn" class="fas fa-bars"></div>
    <nav class="navbar">
        <ul>
            <li><a href="dashboard.php">ALL TASKS</a></li>
            <li><a href="newtask.php">NEW TASK</a></li>
            <li><a href="pendingtasks.php">PENDING TASKS</a></li>
            <li><a href="completedtasks.php">COMPLETED TASKS</a></li>
            <li><a href="dashboard.php?logout=true">LOGOUT</a></li>
        </ul>
    </nav>
</header>

<div class="task-list">
    <h2>All Tasks</h2>
    <ul class="task-items">
        <?php while ($row = $result->fetch_assoc()) { ?>
            <li class="task-item">
                <div class="task-info">
                    <h3><?php echo $row['task']; ?></h3>
                    <p><?php echo $row['date']; ?></p>
                </div>

                <div class="task-actions">
```

```

        <a href="dashboard.php?complete=<?php echo $row['id']; ?>" class="btn-
complete">Complete</a>
        <a href="updatetask.php?id=<?php echo $row['id']; ?>" class="btn-update">Update</a>
        <a href="dashboard.php?delete=<?php echo $row['id']; ?>" class="btn-delete"
onclick="return confirm('Are you sure you want to delete this task?');">Delete</a>
    </div>
</li>
<?php } ?>
</ul>
</div>

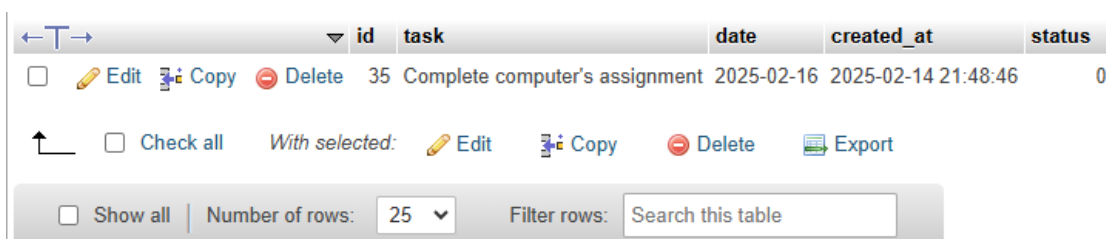
<div class="bg-animation">
    <span></span>
    <span></span>
    <span></span>
    <span></span>
</div>
<script>
let navbar = document.querySelector('.header .navbar');
document.querySelector('#menu-btn').onclick = () => navbar.classList.toggle('active');
window.onscroll = () => navbar.classList.remove('active');
</script>
</body>
</html>

```

UPDATE OPERATION

The Update Task page provides a dedicated interface for modifying existing task details within the To-Do List Project. Accessible by clicking the "UPDATE" button on the All-Tasks page, this page allows users to revise crucial task information. Upon completing the necessary modifications, clicking the "Update Task" button saves the changes and reflects them in the main task list. This page ensures efficient task management by providing a centralized location for keeping task details accurate and up-to-date.

- **Before Update**



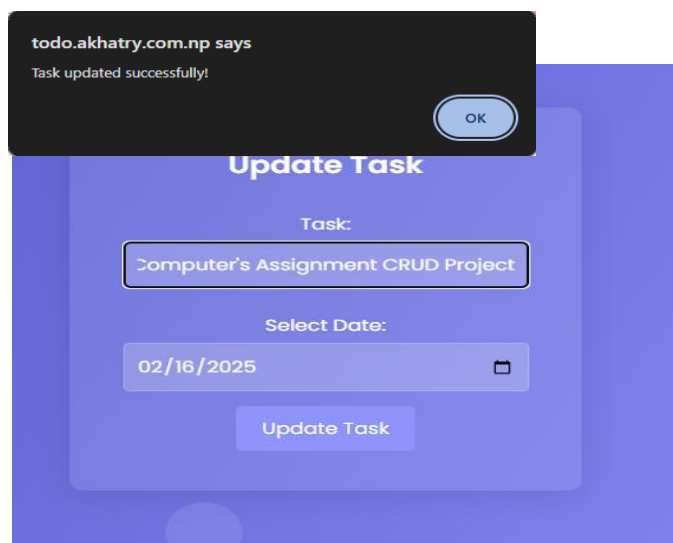
A screenshot of a web application's task list table. The table has columns for 'id', 'task', 'date', 'created_at', and 'status'. The first row shows a task with id 35, 'Complete computer's assignment', date '2025-02-16', created_at '2025-02-14 21:48:46', and status '0'. Above the table are navigation icons and action buttons (Edit, Copy, Delete). Below the table are controls for 'Show all', 'Number of rows' (set to 25), 'Filter rows' (with a search box), and 'Export'.

	id	task	date	created_at	status
<input type="checkbox"/> Edit Copy Delete	35	Complete computer's assignment	2025-02-16	2025-02-14 21:48:46	0

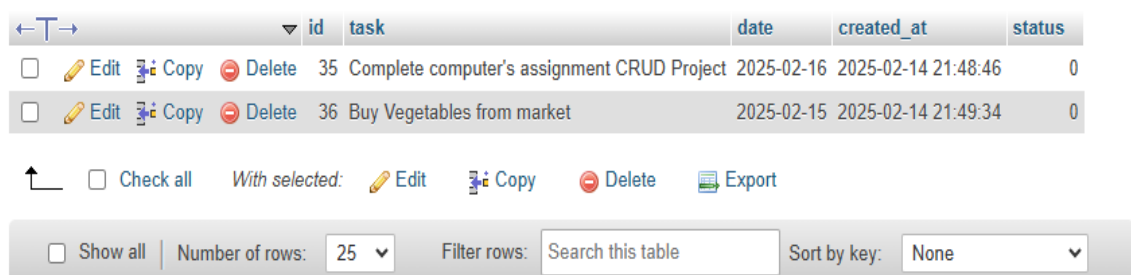
↑ ☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 Filter rows: Search this table

- **After Update**



A screenshot of the 'Update Task' form. A black toast message at the top says 'todo.akhatry.com.np says Task updated successfully!' with an 'OK' button. The form has a title 'Update Task' and a 'Task:' label. The task input field contains 'Computer's Assignment CRUD Project'. Below it is a 'Select Date:' label and a date input field showing '02/16/2025'. At the bottom is an 'Update Task' button.



A screenshot of the task list table after the update. It now shows two rows. The first row is the updated task (id 35, 'Complete computer's assignment CRUD Project', date '2025-02-16'). The second row is a new task (id 36, 'Buy Vegetables from market', date '2025-02-15'). The table controls at the bottom now include a 'Sort by key' dropdown set to 'None'.

	id	task	date	created_at	status
<input type="checkbox"/> Edit Copy Delete	35	Complete computer's assignment CRUD Project	2025-02-16	2025-02-14 21:48:46	0
<input type="checkbox"/> Edit Copy Delete	36	Buy Vegetables from market	2025-02-15	2025-02-14 21:49:34	0

↑ ☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Code used:

```
<?php

session_start();

if (isset($_GET['logout'])) {
    session_destroy();
    header("Location: login.php");
    exit();
}

include('connection.php');
$taskId = $_GET['id'];
$sql = "SELECT * FROM tasks WHERE id = $taskId";
$result = $conn->query($sql);
$taskDetails = $result->fetch_assoc();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $task = ($_POST['task']);
    $date = $_POST['date'];
    $sql = "UPDATE tasks SET task = '$task', date = '$date' WHERE id = $taskId";
    echo $conn->query($sql) ? "<script>alert('Task updated successfully!');
window.location='dashboard.php';</script>" : "<script>alert('Error updating task.');" ;
    $conn->close();
?>

<html lang="en">
<head>
<title>New Task</title>
<link rel="stylesheet" href="style/dash.css">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.5/dist/css/bootstrap.min.css">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.5/font/bootstrap-icons.css">
</head>
<body>
<header class="header">
<a class="logo">TO DO LIST</a>
<div id="menu-btn" class="fas fa-bars"></div>
<nav class="navbar">
<ul>
<li><a href="dashboard.php">ALL TASKS</a></li>
<li><a href="newtask.php">NEW TASK</a></li>
<li><a href="pendingtasks.php">PENDING TASKS</a></li>
<li><a href="completedtasks.php">COMPLETED TASKS</a></li>
<li><a href="dashboard.php?logout=true">LOGOUT</a></li>
</ul>
</nav>
</header>
<div class="bg-animation">
<span></span>
<span></span>
<span></span>
<span></span>
</div>

<div class="task-form">
<h2>Update Task</h2>
<form action="" method="POST">
```



```

        <label for="task">Task:</label>
        <input type="text" id="task" name="task" value="<?php echo
htmlspecialchars($taskDetails['task']); ?>" required>
        <label for="date">Select Date:</label>
        <input type="date" id="date" name="date" value="<?php echo
htmlspecialchars($taskDetails['date']); ?>" required>
        <button type="submit">Update Task</button>
    </form>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js"></script>
<script>

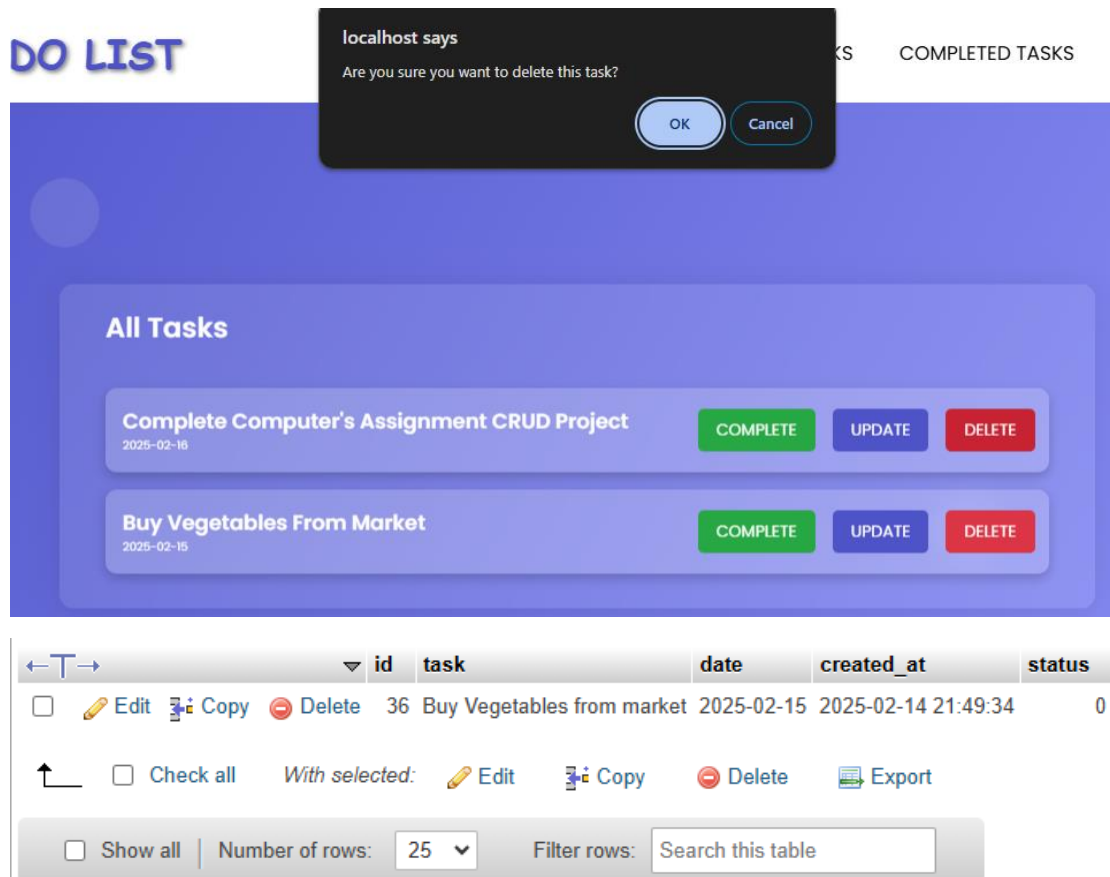
let navbar = document.querySelector('.header .navbar');

if (navbar) {
    document.querySelector('#menu-btn').onclick = () => {
        navbar.classList.toggle('active');
    }
    window.onscroll = () => {
        navbar.classList.remove('active');
    }
}
else {
    console.error("Navbar element not found.");}
</script>
</body>
</html>

```

DELETE OPERATION

The DELETE function provides a means to permanently remove tasks from the All-Tasks list within the To-Do List Project. Upon clicking the DELETE button associated with a specific task, a confirmation dialog appears to prevent accidental deletions.



Code used:

```
if (isset($_GET['complete'])) {  
    $task_id = $_GET['complete'];  
    $conn->query("UPDATE tasks SET status = 1 WHERE id = $task_id");  
    header("Location: dashboard.php");  
    exit();  
}
```

➤ Additional Functions

COMPLETED OPERATION

The Completed Tasks page provides a record of all tasks that have been marked as finished within the To-Do List Project. Accessible through the navigation bar at the top of the interface, this page offers a view-only list of completed tasks, indicated by their name and completion date. The "DELETE" button allows users to permanently remove completed tasks from the record. This page serves as an archive of past work, allowing users to review completed activities and maintain a clean, organized view of their current tasks on the main All Tasks page. It provides a historical perspective on productivity and completed objectives.



				id	task	date	created_at	status
<input type="checkbox"/>	Edit	Copy	Delete	35	Complete computer's assignment CRUD Project	2025-02-16	2025-02-14 21:48:46	1
<input type="checkbox"/>	Edit	Copy	Delete	36	Buy Vegetables from market	2025-02-15	2025-02-14 21:49:34	0

⬆ ☐ Check all With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

☐ Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Code used:

```
<?php
session_start();
if (isset($_GET['logout'])) {
    session_destroy();
    header("Location: login.php");
    exit();
}
include('connection.php');
$sql = "SELECT * FROM tasks WHERE status = 1 ORDER BY date DESC";
$result = $conn->query($sql);
if (isset($_GET['delete'])) {
    $id = $_GET['delete'];
    $conn->query("DELETE FROM tasks WHERE id = $id");
    header("Location: completedtasks.php");
    exit();
}
$conn->close();
?>
```

<html lang="en">

```

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Completed Tasks</title>
<link rel="stylesheet" href="style/dash.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css">
</head>
<body>

<header class="header">
  <a class="logo">TO DO LIST</a>
  <div id="menu-btn" class="fas fa-bars"></div>
  <nav class="navbar">
    <ul>
      <li><a href="dashboard.php">ALL TASKS</a></li>
      <li><a href="newtask.php">NEW TASK</a></li>
      <li><a href="pendingtasks.php">PENDING TASKS</a></li>
      <li><a href="completedtasks.php">COMPLETED TASKS</a></li>
      <li><a href="dashboard.php?logout=true">LOGOUT</a></li>
    </ul>
  </nav>
</header>

<div class="task-list">
  <h2>Completed Tasks</h2>
  <ul class="task-items">
    <?php while ($row = $result->fetch_assoc()) { ?>
      <li class="task-item">
        <div class="task-info">
          <h3><?php echo $row['task']; ?></h3>
          <p><?php echo $row['date']; ?></p>
        </div>
        <div class="task-actions">
          <a href="completedtasks.php?delete=<?php echo $row['id']; ?>" class="btn-delete"
onclick="return confirm('Are you sure you want to delete this task?');">Delete</a>
        </div>
      </li>
    <?php } ?>
  </ul>
</div>

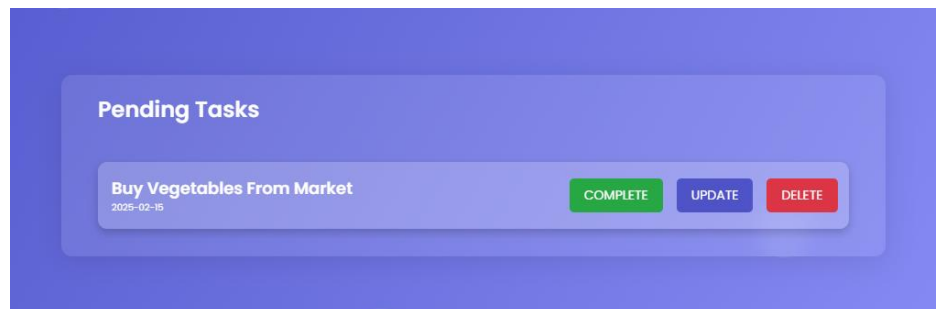
<div class="bg-animation">
  <span></span>
  <span></span>
  <span></span>
  <span></span>
</div>

<script>
let navbar = document.querySelector('.header .navbar');
document.querySelector('#menu-btn').onclick = () => navbar.classList.toggle('active');
window.onscroll = () => navbar.classList.remove('active');
</script>
</body></html>

```

PENDING OPERATION

The Pending Tasks page offers a focused view of all tasks that are currently in progress or awaiting completion within the To-Do List Project. Users can interact with their pending tasks directly from this page, choosing to COMPLETE a task, UPDATE its details, or DELETE it if necessary. This page is designed to enhance productivity by providing a clear and manageable overview of ongoing tasks, allowing users to prioritize and efficiently track their work in progress.



Code used:

```
<?php
session_start();

if (isset($_GET['logout'])) {
    session_destroy();
    header("Location: login.php");
    exit();
}

include('connection.php');

if (isset($_GET['complete'])) {
    $task_id = $_GET['complete'];
    $conn->query("UPDATE tasks SET status = 1 WHERE id = $task_id");
    header("Location: pendingtasks.php");
    exit();
}

$sql = "SELECT * FROM tasks WHERE status = 0 ORDER BY date DESC";
$result = $conn->query($sql);

$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Pending Tasks</title>
<link rel="stylesheet" href="style/dash.css">
</head>
<body>

<header class="header">
```

```

<a class="logo">TO DO LIST</a>
<nav class="navbar">
  <ul>
    <li><a href="dashboard.php">ALL TASKS</a></li>
    <li><a href="newtask.php">NEW TASK</a></li>
    <li><a href="pendingtasks.php">PENDING TASKS</a></li>
    <li><a href="completedtasks.php">COMPLETED TASKS</a></li>
    <li><a href="dashboard.php?logout=true">LOGOUT</a></li>
  </ul>
</nav>
</header>

<div class="task-list">
  <h2>Pending Tasks</h2>
  <ul class="task-items">
    <?php while ($row = $result->fetch_assoc()) { ?>
      <li class="task-item">
        <div class="task-info">
          <h3><?php echo $row['task']; ?></h3>
          <p><?php echo $row['date']; ?></p>
        </div>
        <div class="task-actions">
          <a href="dashboard.php?complete=<?php echo $row['id']; ?>" class="btn-
complete">Complete</a>
          <a href="updatetask.php?id=<?php echo $row['id']; ?>" class="btn-update">Update</a>
          <a href="dashboard.php?delete=<?php echo $row['id']; ?>" class="btn-delete"
onclick="return confirm('Are you sure you want to delete this task?');">Delete</a>
        </div>
      </li>
    <?php } ?>
  </ul>
</div>

<div class="bg-animation">
  <span></span>
  <span></span>
  <span></span>
  <span></span>
</div>

<script>
let navbar = document.querySelector('.header .navbar');

document.querySelector('#menu-btn').onclick = () => navbar.classList.toggle('active');
window.onscroll = () => navbar.classList.remove('active');
</script>

</body>
</html>

```

CONCLUSION

In conclusion, the To-Do List Project delivers a practical and intuitive solution for effective task management. Its user-friendly interface simplifies the process of creating new tasks, updating existing details, marking tasks as complete, and removing them when no longer needed. The organization of tasks into "All Tasks," "Pending Tasks," and "Completed Tasks" views provides users with a clear overview of their workload and facilitates efficient prioritization. Key features such as due dates, comprehensive task descriptions, and a confirmation step for deletions enhance the project's robustness and ensure data integrity. By streamlining task management, this project empowers users to improve their organization, boost productivity, and successfully accomplish their objectives. The straightforward design and clear functionality make it an accessible tool for anyone seeking to enhance their personal or professional task management skills.

BIBLIOGRAPHY

1. Pudasaini, Dipak. Computer Science II. Buddha Publication, 2024.
2. Shrestha, Prachandra. Essentials of Computer Science. Asmita Books Publisher and Distributors , 2078.
3. Step by Step (YouTube),
<https://youtu.be/72U5Af8KUpA?si=3oDVFZHCdt3324R6>
4. Freedial, <https://freedial.in/how-to-create-crud-operation-in-php-and-mysql/>
5. W3Schools, https://www.w3schools.com/php/php_mysql_connect.asp