

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**CHENNAI-602105**

**TASK TRACKER SYSTEM**

**A CAPSTONE PROJECT REPORT**

*Submitted in the partial fulfillment for the completion of the course*

**CSA4309 INTERNET PROGRAMMING FOR WEB SERVICES**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**M.Deepthi(192210113)**

**S.Bhuvaneswari(192211813)**

**Under the Supervision of**

**Dr. K. Jayasakthi Velmurugan**

**NOV 2024**

**DECLARATION**

We, **M.Deepthi, S.Bhuvaneswari.**  students of **Bachelor of Engineering in the Department** of Computer Science and Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha School of Engineering, Chennai, hereby declare that the work presented in this Capstone Project Work entitled **Food Delivery Website** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics.

(M.Deepthi 192210113)

(S.Bhuvaneswari 192211813)

Date:

Place:

**CERTIFICATE**

This is to certify that the project entitled **“TASK TRACKING SYSTEM”** submitted by **M.Deepthi & S.Bhuvaneswari** has been carried out under my supervision. The project has been submitted as per the requirements in the current semester of B.E. Computer Science and Engineering.

Supervisor

Dr. K. Jayasakthi Velmurugan

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPICS** | **PAGE NO.** |
| 1 | **Abstract** | **5** |
| 2 | **Introduction** | **6** |
| 3 | **Project Description** | **7** |
| 4 | **Problem Description** | **8** |
| 5 | **Tool Description** | **9-10** |
| 6 | **Operations** | **11** |
| 7 | **Module Description** | **12** |
| 8 | **Implementation** | **13-30** |
| 9 | **Result** | **31-32** |
| 10 | **Conclusion & References** | **33** |

**ABSTRACT**

This capstone project involves designing and developing a Task Tracker System using the MERN (MongoDB, Express.js, React, and Node.js) stack. The system aims to provide a seamless and efficient task management experience for teams and individuals, enabling them to track and manage tasks, projects, and workflows in real-time.

Our project tackles common challenges in task management systems, such as Managing complex workflows and multiple tasks Ensuring seamless team collaboration and communication Providing real-time updates and notifications Scaling to accommodate growing teams and projects Ensuring data security and integrity

Through this project, we aim to demonstrate the practical application of web development skills acquired during our coursework, utilizing the MERN stack to build a functional and efficient Task Tracker System. This project showcases our ability to design, develop, and deploy a robust task management platform, addressing real-world challenges faced by teams and organizations.

**INTRODUCTION**

In today's fast-paced and dynamic work environment, effective task management is crucial for teams and individuals to achieve their goals and objectives. Traditional methods of task management, such as manual tracking and paper-based systems, are often inefficient, prone to errors, and lack real-time visibility. To address these challenges, a Task Tracker System is proposed to provide a digital platform for creating, assigning, tracking, and managing tasks.

The Task Tracker System aims to provide a robust, scalable, and user-friendly platform for task management. It will enable users to create, assign, and track tasks, set deadlines and priorities, collaborate with team members, receive real-time updates and notifications, and generate reports and analytics. By leveraging cutting-edge technology and best practices in task management, this system will improve operational efficiency, reduce errors, and enhance overall productivity.

To address these challenges, a Task Tracker System is proposed to provide a digital platform for creating, assigning, tracking, and managing tasks. This system aims to streamline task management processes, enhance collaboration, and provide real-time insights into task status and progress. By leveraging cutting-edge technologies and intuitive design principles, the Task Tracker System will empower teams to work more efficiently, effectively, and collaboratively.Streamlined task management and organization Increased team productivity and collaboration Real-time visibility into task status and progress Data-driven decision-making through reporting and analytic the Scalable architecture for future growth and expansion

**PROJECT DESCRIPTION**

This project involves the design and development of a full-stack web application for a Task Tracker System using the MERN stack—MongoDB, Express.js, React, and Node.js. The website offers an interactive and efficient platform for teams and individuals to manage tasks, projects, and workflows, providing a comprehensive digital solution that addresses the needs of both task assigners and assignees.

**Key Features and Functionalities:**

**1. Streamlined Task Management and Organization**

Streamlined task management and organization refers to the efficient and structured process of creating, assigning, tracking, and managing tasks. This involves:

- Simplifying workflows

- Reducing manual errors

- Automating routine tasks

**2. Increased Team Productivity and Collaboration**

Increased team productivity and collaboration refers to the improvement in teamwork efficiency, communication, and output. This involves:

- Enhanced teamwork and cooperation

- Improved communication and feedback

**3. Real-Time Visibility into Task Status and Progress**

Real-time visibility into task status and progress refers to the ability to track and monitor tasks as they are being completed. This involves:

- Live updates on task status

- Progress tracking and monitoring

4. **Data-Driven Decision-Making through Reporting and Analytics**

Data-driven decision-making through reporting and analytics refers to the use of data and insights to inform business decisions. This involves:

- Generating customizable reports

**PROJECT DESCRIPTION**

**Project Objectives:**

**Design and Develop a User-Friendly Task Tracker System:** Create an intuitive and interactive Task Tracker System that enables users to easily navigate, create, assign, and track tasks.

**2. Provide Real-Time Task Tracking and Updates:** Implement real-time updates and notifications to ensure that users are informed of task changes, progress, and completion.

**3. Enhance Team Collaboration and Communication:** Develop features that facilitate collaboration and communication among team members, including commenting, file sharing, and task assignment.

**4. Improve Task Management Efficiency:** Streamline task management processes by providing features such as task prioritization, categorization, and filtering, enabling users to quickly identify and focus on high-priority tasks.

**Tools:**

**1. React:** A JavaScript library for building user interfaces, allowing developers to create reusable UI components.

**2. Angular:** A JavaScript framework for building single-page applications, providing a structured approach to front-end development.

**3. Vue.js:** A progressive and flexible JavaScript framework for building web applications, offering a robust ecosystem and strong community support.

**4. HTML/CSS:** HyperText Markup Language (HTML) is used for structuring content, while Cascading Style Sheets (CSS) are used for styling and layout.

**5. JavaScript:** A high-level programming language used for adding interactivity to web applications.

**User Interface:**

**Login Page**: Username and password fields Forgot password link

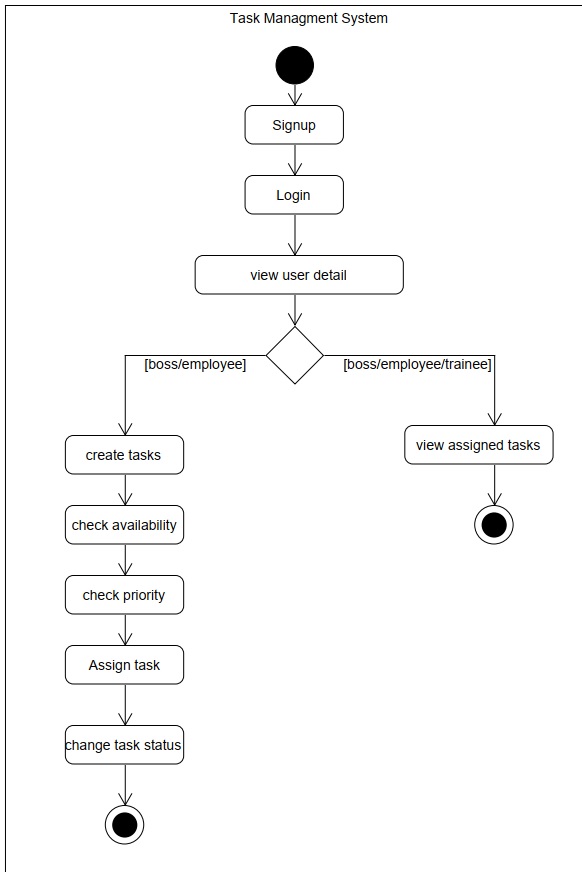
Register button for new users

**Dashboard:** Task list with filtering and sorting options Project overview with progress bars Upcoming deadlines and notifications Quick task creation button

**Task Creation Page:** Task title and description fields Assignment and due date selectors Priority and category dropdowns Attach file and comment options

**Task Details Page:** Task description and comments Progress tracking and status updatesEdit and delete optionsRelated tasks and projects

**BLOCK DIAGRAM:**



**OPERATIONS:**

**User Operations**

**1. User Registration:** The process of creating a new user account, including providing personal details and setting up a username and password.

**2. User Login:** The process of accessing the Task Tracker System using a valid username and password.

**3. User Profile Management:** The ability to view and edit user profile information, including name, email, and password.

**4. Task Assignment:** The process of assigning tasks to team members or individuals.

**5. Task Creation:** The process of creating new tasks, including setting task details such as title, description, and due date.

**6. Task Deletion:** The process of deleting the existing tasks, which has no more use.

**Module Description:**

The Task Tracking System is a comprehensive web-based application designed to streamline task management and team collaboration. The system consists of six modules: User Management, Task Management, Project Management, Reporting and Analytics, Collaboration and Communication, and Security and Access Control. These modules work together to provide a seamless and efficient task tracking experience, enabling users to create and assign tasks, track progress, and collaborate with team members in real-time. The system also provides robust reporting and analytics capabilities, allowing users to gain valuable insights into task and project performance.

**Features:**

* Create new tasks with details such as title, description, due date, and priority.
* Assign tasks to team members or individuals.
* Update the status of tasks as they progress (e.g., in progress, completed, overdue).
* Prioritize tasks based on their importance and urgency.
* Filter and sort tasks by various criteria such as due date, priority, and status.
* Implement role-based access control to restrict access to certain features and tasks based on user roles.
* Encrypt data to ensure it is protected from unauthorized access.
* Perform regular security audits to identify and address potential security vulnerabilities.

**IMPLEMENTATION:**

**Login.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login and Sign Up</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #f4f4f4;

            text-align: center;

        }

        .form-container {

            width: 300px;

            margin: 100px auto;

            background-color: white;

            padding: 30px;

            box-shadow: 0 4px 8px rgba(9, 182, 20, 0.1);

            border-radius: 8px;

        }

        input {

            width: 100%;

            padding: 10px;

            margin: 10px 0;

            border: 1px solid #ccc;

            border-radius: 4px;

        }

        button {

            padding: 10px;

            background-color: #4CAF50;

            color: white;

            border: none;

            border-radius: 4px;

            cursor: pointer;

            width: 100%;

        }

        button:hover {

            background-color: #45a049;

        }

        a {

            color: #4CAF50;

        }

        body {

            background-image: url("https://www.getharvest.com/hs-fs/hubfs/resources/harvest-gif-motivate.gif?width=720&name=harvest-gif-motivate.gif");

            background-size: cover;

            background-repeat: no-repeat;

            background-attachment: fixed;

            font-family: Arial, sans-serif;

        }

    </style>

</head>

<body>

    <!-- Sign Up Form -->

    <div id="signup-form" class="form-container">

        <h2>Create an Account</h2>

        <!-- Signup form -->

        <form action="php/signup.php" method="POST">

            <input type="text" name="Username" placeholder="Username" required>

            <input type="password" name="Password" placeholder="Password" required>

            <input type="email" name="Email" placeholder="Email" required>

            <input type="text" name="PhoneNumber" placeholder="Phone Number" required>

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

            <button type="submit" name="signup">Sign Up</button>

        </form>

        <p>Already have an account? <a href="javascript:showLoginForm()">Login</a></p>

    </div>

    <!-- Login Form -->

    <div id="login-form" class="form-container" style="display: none;">

        <h2>Login</h2>

        <form id="login" method="POST" action="php/login.php">

            <input type="text" name="Username" placeholder="Username" required>

            <input type="password" name="Password" placeholder="Password" required>

            <button type="submit" name="login">Login</button>

        </form>

        <p>Don't have an account? <a href="javascript:showSignUpForm()">Sign Up</a></p>

    </div>

    <script>

        // Show Sign Up Form

        function showSignUpForm() {

            document.getElementById('login-form').style.display = 'none';

            document.getElementById('signup-form').style.display = 'block';

        }

        // Show Login Form

        function showLoginForm() {

            document.getElementById('signup-form').style.display = 'none';

            document.getElementById('login-form').style.display = 'block';

        }

        // Redirect to tracker page on form submission (for demo purposes)

        document.getElementById('signup').onsubmit = function(e) {

            e.preventDefault(); // Prevents default form submission for demonstration

            alert('Sign Up Successful!');

            window.location.href = "traker.html"; // Redirect to tracker page

        };

        document.getElementById('login').onsubmit = function(e) {

            e.preventDefault(); // Prevents default form submission for demonstration

            alert('Login Successful!');

            window.location.href = "traker.html"; // Redirect to tracker page

        };

    </script>

</body>

</html>

**Traker.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>BT Day 1 - Task Tracker</title>

    <link rel="stylesheet" href="style.css">

    <style>

        table, th, td {

            border: 1px solid black;

            border-collapse: collapse;

        }

    </style>

</head>

<body>

    <header>

        <h1 style="text-align:center;">TASK TRACKER</h1>

    </header>

    <div id="table-div">

        <div class="div-form">

            <form id="new-entry">

                <label for="filter">Filter: </label>

                <select id="filter" title="Filter">

                    <option value="all">All</option>

                    <option value="Work">Work</option>

                    <option value="Personal Space">Personal Space</option>

                </select>

                <br>

            </form>

        </div>

        <table class="table">

            <thead>

                <tr>

                    <th>Category</th>

                    <th>Sub-Category</th>

                    <th>Duration</th>

                    <th>Task</th>

                    <th>Tools</th>

                </tr>

            </thead>

            <tbody class="table-body">

                <tr>

                    <td>Work</td>

                    <td>Meeting</td>

                    <td>00:40:21</td>

                    <td>Client Meeting</td>

                    <td>

                        <button onclick="updateRow(this)">Update</button>

                        <button onclick="deleteRow(this)">Delete</button>

                    </td>

                </tr>

                <tr>

                    <td>Personal Space  </td>

                    <td>painting</td>

                    <td>00:26:21  </td>

                    <td>- </td>

                    <td>

                        <button onclick="updateRow(this)">Update</button>

                        <button onclick="deleteRow(this)">Delete</button>

                    </td>

                </tr>

                <tr>

                    <td>Work   </td>

                <td>Project  </td>

                <td>01:30:21</td>

                <td>Mail</td>

                <td>

                    <button onclick="updateRow(this)">Update</button>

                    <button onclick="deleteRow(this)">Delete</button>

                </td>

            </tr>

            <tr>

                <td>Personal Space</td>

                <td>Exercise </td>

                <td>01:10:34 </td>

                <td>Running  </td>

                <td>

                    <button onclick="updateRow(this)">Update</button>

                    <button onclick="deleteRow(this)">Delete</button>

                </td>

            </tr>

                <!-- Add more rows here as needed -->

            </tbody>

        </table>

    </div>

    <div id="form-container">

        <h2 style="text-align:center;">START A NEW TASK</h2>

        <!-- Updated form with POST method and action to php/insertTask.php -->

        <form method="POST" action="php/insertTask.php">

            <label for="category"><strong>Category:</strong></label>

            <select id="category" name="category" title="Category" required>

                <option value="Work">Work</option>

                <option value="Personal Space">Personal Space</option>

            </select>

            <br><br>

            <label for="subCategory"><strong>Sub-Category:</strong></label>

            <input type="text" id="subCategory" name="subCategory" placeholder="Sub-Category" required>

            <br><br>

            <label for="duration"><strong>Duration:</strong></label>

            <div class="stopwatch">00:00:00</div>

            <input type="hidden" id="duration" name="duration" required> <!-- hidden field for stopwatch duration -->

            <button id="startstopbutton" type="button" onclick="startStopButton()">START</button>

            <button id="resetButton" type="button" onclick="resetTimer()">Reset</button>

            <br><br>

            <label for="task"><strong>Task:</strong></label>

            <input type="text" id="task" name="task" placeholder="Task Name" required>

            <br><br>

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

            <button type="button" onclick="goBackHome()" class="home-button">Back to Home</button>

        </form>

    </div>

    <script src="index.js"></script>

    <script>

        function goBackHome() {

            window.location.href = "login.html"; // Adjust with your actual home page path

        }

    </script>

</body>

</html>

**index.js**

// Timer variables and filter function

let seconds = 0;

let minutes = 0;

let hours = 0;

let isRunning = false;

function filterTable() {

    const selected = document.getElementById("filter").value.toLowerCase();

    const tableRows = document.querySelectorAll(".table-body tr");

    tableRows.forEach(row => {

        const category = row.cells[0].innerText.trim().toLowerCase();

        row.style.display = (selected === "all" || category === selected) ? "table-row" : "none";

    });

}

document.getElementById("filter").addEventListener("change", filterTable);

// Stopwatch functionality

function startStopButton() {

    if (!isRunning) {

        isRunning = true;

        intervalId = setInterval(() => {

            seconds++;

            if (seconds >= 60) {

                seconds = 0;

                minutes++;

                if (minutes >= 60) {

                    minutes = 0;

                    hours++;

                }

            }

            document.getElementById('startstopbutton').innerText = 'STOP';

            let formattedTime = `${hours.toString().padStart(2, "0")}:${minutes.toString().padStart(2, "0")}:${seconds.toString().padStart(2, "0")}`;

            document.querySelector('.stopwatch').innerText = formattedTime;

        }, 1000);

    } else {

        isRunning = false;

        clearInterval(intervalId);

        document.getElementById("startstopbutton").innerText = "Start";

    }

}

function resetTimer() {

    clearInterval(intervalId);

    isRunning = false;

    seconds = 0;

    minutes = 0;

    hours = 0;

    document.getElementById("startstopbutton").innerText = "Start";

    document.querySelector('.stopwatch').innerText = "00:00:00";

}

// Add Task Dynamically

document.querySelector("form[action='php/insertTask.php']").addEventListener("submit", function(event) {

    event.preventDefault();

    const category = document.getElementById("category").value;

    const subCategory = document.getElementById("subCategory").value;

    const duration = document.querySelector('.stopwatch').innerText;

    const task = document.getElementById("task").value;

    const newRow = document.createElement("tr");

    newRow.innerHTML = `

        <td>${category}</td>

        <td>${subCategory}</td>

        <td>${duration}</td>

        <td>${task}</td>

        <td>

            <button onclick="updateRow(this)">Update</button>

            <button onclick="deleteRow(this)">Delete</button>

        </td>

    `;

    document.querySelector(".table-body").appendChild(newRow);

    document.getElementById("new-entry").reset();

    resetTimer();

});

// Update and Delete Row Functions

function updateRow(button) {

    const row = button.closest("tr");

    const cells = row.querySelectorAll("td");

    document.getElementById("category").value = cells[0].innerText.trim();

    document.getElementById("subCategory").value = cells[1].innerText.trim();

    document.querySelector('.stopwatch').innerText = cells[2].innerText.trim();

    document.getElementById("task").value = cells[3].innerText.trim();

    row.remove();

}

function deleteRow(button) {

    button.closest("tr").remove();

}

// Login and Signup Event Listeners

document.getElementById('loginFormElement').onsubmit = function(e) {

    e.preventDefault();

    const username = document.querySelector('input[name="Username"]').value;

    const password = document.querySelector('input[name="Password"]').value;

    if (!username || !password) {

        alert("Please fill in both fields!");

        return;

    }

    fetch('php/login.php', {

        method: 'POST',

        body: new URLSearchParams({

            Username: username,

            Password: password

        }),

    })

    .then(response => response.json())

    .then(data => {

        if (data.success) {

            window.location.href = "tracker.html";

        } else {

            alert("Invalid username or password.");

        }

    })

    .catch(error => alert("Error: " + error.message));

};

document.getElementById('signupFormElement').onsubmit = function(e) {

    e.preventDefault();

    const username = document.querySelector('input[name="Username"]').value;

    const password = document.querySelector('input[name="Password"]').value;

    const email = document.querySelector('input[name="Email"]').value;

    const phone = document.querySelector('input[name="PhoneNumber"]').value;

    const dob = document.querySelector('input[name="DateOfBirth"]').value;

    if (!username || !password || !email || !phone || !dob) {

        alert("Please fill in all fields!");

        return;

    }

    fetch('php/signup.php', {

        method: 'POST',

        body: new URLSearchParams({

            Username: username,

            Password: password,

            Email: email,

            PhoneNumber: phone,

            DateOfBirth: dob

        }),

    })

    .then(response => response.json())

    .then(data => {

        if (data.success) {

            window.location.href = "tracker.html";

        } else {

            alert("Error during signup: " + data.message);

        }

    })

    .catch(error => alert("Error: " + error.message));

};

**style.css**

body {

    background-image: url("https://www.getharvest.com/hs-fs/hubfs/resources/harvest-gif-motivate.gif?width=720&name=harvest-gif-motivate.gif");

  }

.div-form{

    float:absolute;

    margin-left: 200px;

    background-color:antiquewhite;

    width: 67%;

    margin: 20px auto;

    padding: 20px;

    border: 1px solid #ccc;

    border-radius: 5px;

}

#form-container {

    background-color:antiquewhite;

    width: 67%;

    margin: 20px auto;

    padding: 20px;

    border: 1px solid #ccc;

    border-radius: 5px;

}

.table {

    margin: 20px auto;

    width: 70%;

    border-collapse: collapse;

    border: 1px solid #000;

}

.table th, .table td {

    border: 1px solid black;

    padding: 8px;

    background-color:antiquewhite;

}

#form-entry label {

    display: block;

    margin-bottom: 5px;

}

#form-entry input[type="text"], #form-entry select {

    width: calc(100% - 10px);

    padding: 8px;

    margin-bottom: 10px;

    border: 1px solid #ccc;

    border-radius: 5px;

}

#form-entry input[type="submit"] {

    background-color: #007bff;

    color: #fff;

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

}

#form-entry input[type="submit"]:hover {

    background-color: #0056b3;

}

.stopwatch {

    font-size: 18px;

    margin-bottom: 10px;

}

.table button {

    padding: 6px 12px;

    margin-right: 5px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

}

.table button:nth-child(odd) {

    background-color: #007bff;

    color: #fff;

}

.table button:nth-child(even) {

    background-color: #ff0000;

    color: #fff;

}

.table button:hover {

    opacity: 0.8;

}

**db.php**

<?php

$servername = "localhost";

$username = "root"; // your MySQL username

$password = ""; // your MySQL password

$dbname = "task\_tracker\_system"; // your database name

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

?>

**signUp.php**

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['signup'])) {

    $username = $\_POST['Username'];

    $password = $\_POST['Password'];

    $email = $\_POST['Email'];

    $phone = $\_POST['PhoneNumber'];

    $dob = $\_POST['DateOfBirth'];

    $conn = new mysqli("localhost", "root", "", "task\_tracker\_system");

    if ($conn->connect\_error) {

        die("Connection failed: " . $conn->connect\_error);

    }

    // Insert user data into the database

    $stmt = $conn->prepare("INSERT INTO users (username, password, email, phone, dob) VALUES (?, ?, ?, ?, ?)");

    $stmt->bind\_param("sssss", $username, $password, $email, $phone, $dob);

    if ($stmt->execute()) {

        header("Location: ../traker.html"); // Redirects to tracker page upon successful signup

        exit();

    } else {

        echo "Error: " . $stmt->error;

    }

    $stmt->close();

    $conn->close();

}

?>

**Login.php**

<?php

if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['login'])) {

    $username = $\_POST['Username'];

    $password = $\_POST['Password'];

    $conn = new mysqli("localhost", "root", "", "task\_tracker\_system");

    if ($conn->connect\_error) {

        die("Connection failed: " . $conn->connect\_error);

    }

    $sql = "SELECT \* FROM users WHERE username = '$username'";

    $result = $conn->query($sql);

    if ($result->num\_rows > 0) {

        $user = $result->fetch\_assoc();

        if ($password === $user['password']) { // For simplicity; consider hashing passwords

            header("Location: ../traker.html"); // Redirect to tracker page on successful login

            exit();

        } else {

            echo "Invalid password.";

        }

    } else {

        echo "No user found with that username.";

    }

    $conn->close();

}

?>

**insertTask.php**

<?php

// Database connection

$servername = "localhost";

$username = "root";

$password = "";

$database = "task\_tracker\_system"; // Replace with your database name

$conn = new mysqli($servername, $username, $password, $database);

// Check connection

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

// Check if form is submitted

if (isset($\_POST['submit'])) {

    // Get form data

    $category = $\_POST['category'];

    $subCategory = $\_POST['subCategory'];

    $duration = $\_POST['duration'];

    $task = $\_POST['task'];

    // Prepare the SQL statement

    $sql = "INSERT INTO tasks (Category, Sub\_Category, Duration, Task) VALUES (?, ?, ?, ?)";

    $stmt = $conn->prepare($sql);

    // Bind parameters and execute the statement

    $stmt->bind\_param("ssss", $category, $subCategory, $duration, $task);

    if ($stmt->execute()) {

        echo "New Task Added!";

        header("Location: ../traker.html"); // Redirect to your tracker page

        exit();

    } else {

        echo "Error: " . $stmt->error;

    }

    // Close the statement and connection

    $stmt->close();

    $conn->close();

}

?>

**Traker.php**

<?php

include('db.php');

// Fetch all tasks

$sql = "SELECT \* FROM tasks";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

    while($row = $result->fetch\_assoc()) {

        echo "<tr>

                <td>".$row['Category']."</td>

                <td>".$row['Sub\_Category']."</td>

                <td>".$row['Duration']."</td>

                <td>".$row['Task']."</td>

                <td>

                    <button onclick='updateRow(this)'>Update</button>

                    <button onclick='deleteRow(this)'>Delete</button>

                </td>

              </tr>";

    }

} else {

    echo "0 results";

}

?>

**RESULT:**

**Fig.1: Sign up page**

**Fig.2: login page**

**Fig.3:Task Tracker Page**

**Conclusion:**

The Task Tracker System is an essential tool for teams and organizations seeking to improve their productivity, efficiency, and collaboration. Its user-friendly interface, robust features, and scalability make it an ideal solution for task management and team collaboration.

The reporting and analytics features provide valuable insights into task and project performance, enabling users to identify areas for improvement and optimize their workflow. The system's security and access control features ensure that sensitive data is protected and only authorized users can access the system.

**Reference:**

[1] Android Studio 2 Development Essentials, Book by Neil Smyth.

[2] PHP and MySQL Web Development, Book by Luke Welling, 2001

[3] D. A. Hillson, "Using a Risk Breakdown Structure in project management", Journal Of Facilities Management, vol. 2, no. 1, pp.

85-97, 2013.

[4] D. A. Hillson, "Using a Risk Breakdown Structure in project management", Journal Of Facilities Management, vol. 2, no. 1, pp.

85-97, 2013.

[5] S. McKenna, "Organisational Complexity and Perceptions of Task", Task Management: An International Journal, vol.

3, no. 2, pp. 53-64, 2013.

[6] A. Aleshin, "Time and Risk Management of International Projects", International Journal of Project Management, no. 19, pp. 207-222, 2014

[7] J. Ellis, L. Kvavilashvili, "Prioritization of tasks in 2000: Past present and future directions", Task Management, vol. 14, pp. 1-9, 2000.

[8] Drake Baer, "Dwight Eisenhower Nailed A Major Insight About Productivity", Business Insider, 2014.