**TaskTrackr: Student Productivity and Project Management Tool**

**Description**

**TaskTrackr** is a modern, web-based task and project management system designed to help individuals and teams organize, track, and complete their work efficiently. Built with PHP and MySQL, TaskTrackr allows users to create and manage projects, assign and monitor tasks, collaborate in groups, and receive timely notifications and reminders. The system features a user-friendly interface, real-time progress tracking, and automated reminders to ensure that deadlines are met and productivity is maximized.

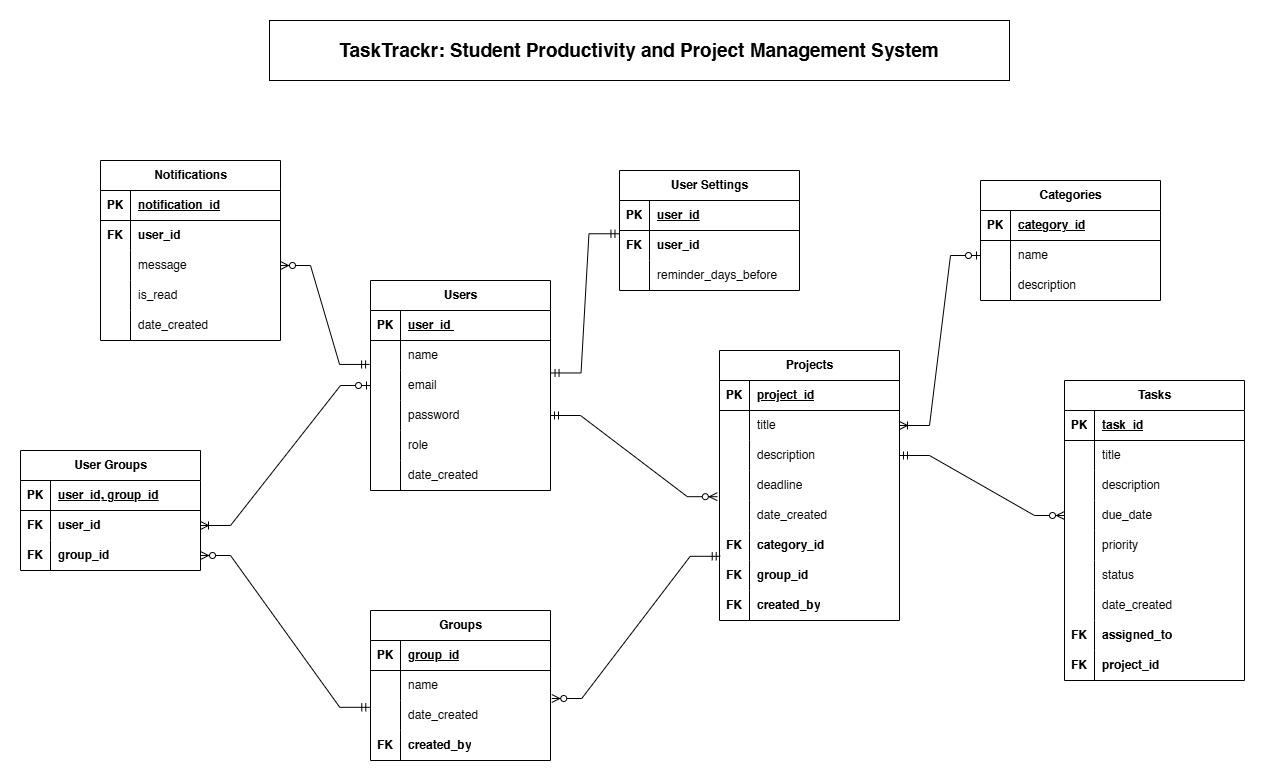
**Objectives**

* To provide an intuitive platform for managing personal and group projects and tasks.
* To enable users to create, assign, update, and track tasks and projects with ease.
* To facilitate collaboration through group management and task assignment features.
* To keep users informed and on schedule with in-app and email notifications and automated reminders.
* To offer a comprehensive dashboard for visualizing progress, upcoming deadlines, and overall productivity.

**Key Features**

* **User Registration & Authentication:** Secure sign-up, login, and session management for all users.
* **Project Management:** Create, edit, and delete projects. Projects can be individual or group-based, with deadlines and categories.
* **Task Management:** Add, assign, update, and delete tasks within projects. Tasks have statuses (Pending, In Progress, Done, Overdue), priorities, and due dates.
* **Group Collaboration:** Create groups, invite users, and manage group membership. Assign group projects and tasks to group members.
* **Dashboard & Progress Tracking:** Visual dashboard with task summary cards, completion charts, and project progress bars. Calendar view for upcoming and overdue tasks.
* **Notifications & Reminders:** In-app and email notifications for assignments, updates, and reminders. Automated reminders for upcoming and overdue tasks.
* **User Profile & Settings:** Update profile information, upload profile pictures, and set personal reminder preferences.
* **Categories:** Organize projects by customizable categories.
* **Responsive, Modern UI:** Clean, mobile-friendly interface using Bootstrap 5 and icons.

**Entity Relationship Diagram (ERD)**



* **Users:** This entity stores all registered users, including their profile information and login credentials. Each user can have multiple settings, notifications, tasks, projects, and group memberships.
* **User\_Settings:** Each user has a single settings record, storing preferences such as how many days in advance they want to receive task reminders. This is a one-to-one relationship with Users.
* **Categories:** Projects can be organized into categories (e.g., Homework, Work, Personal). Each project references a category, but categories are independent and reusable.
* **Groups:** Groups allow for collaborative work. Each group has a creator (a user) and can have multiple members.
* **User\_Groups:** This is a join table that implements a many-to-many relationship between Users and Groups. A user can belong to multiple groups, and a group can have multiple users.
* **Projects:** Projects are the main organizational unit for tasks. Each project has a creator (user), may belong to a group (for group projects), and is assigned a category. Projects can be individual (no group) or group-based.
* **Tasks:** Tasks are the actionable items within projects. Each task belongs to a project and is assigned to a user. Tasks have attributes such as title, description, due date, priority, and status.
* **Notifications:** Notifications are generated for users when important events occur (e.g., task assigned, status updated, reminders). Each notification is linked to a user.

**Screenshots**

|  |
| --- |
|  |
|  |
|  |
|  |

**Sample SQL Queries**

|  |  |
| --- | --- |
| **Task Summary** | SELECT  CASE  WHEN t.status = 'Done' THEN 'Done'  WHEN t.due\_date < CURDATE() THEN 'Overdue'  ELSE t.status  END AS status\_group,  COUNT(\*) AS count  FROM Tasks t LEFT JOIN Projects p  ON t.project\_id = p.project\_id  WHERE t.assigned\_to = ? AND t.project\_id IS NOT NULL  GROUP BY status\_group |
| **Upcoming and Overdue Tasks** | SELECT  t.title AS task\_title,  t.due\_date,  CASE  WHEN t.due\_date < CURDATE() AND t.status != 'Done' THEN 'Overdue'  ELSE 'Upcoming'  END AS task\_status  FROM Tasks t  LEFT JOIN Projects p ON t.project\_id = p.project\_id  WHERE t.assigned\_to = ? AND t.project\_id IS NOT NULL  ORDER BY t.due\_date ASC |
| **Project Progress** | SELECT COUNT(\*) AS total, SUM(status = 'Done') AS done FROM Tasks WHERE project\_id = ? |
| **Fetch Group Members** | SELECT u.name  FROM Users u  JOIN User\_Groups ug ON u.user\_id = ug.user\_id  WHERE ug.group\_id = ? |
| **Fetch User's Projects** | SELECT p.\*, c.name AS category\_name, g.group\_name, u.name AS creator\_name  FROM Projects p  JOIN Categories c ON p.category\_id = c.category\_id  LEFT JOIN `Groups` g ON p.group\_id = g.group\_id  JOIN Users u ON p.created\_by = u.user\_id  WHERE p.created\_by = ?  OR p.group\_id IN (  SELECT group\_id FROM User\_Groups WHERE user\_id = ?) |
| **Insert New Task** | INSERT INTO Tasks (title, description, due\_date, priority, status, project\_id, assigned\_to)  VALUES (?, ?, ?, ?, ?, ?, ?) |
| **Update Task Status** | UPDATE Tasks SET status = ? WHERE task\_id = ? AND assigned\_to = ? |

**Reflection**

Working on TaskTrackr has been a rewarding and educational experience for our team. Throughout the development process, we deepened our understanding of database normalization, efficient schema design, and the importance of clear entity relationships. Implementing advanced SQL queries using JOINs, GROUP BY, and aggregate functions allowed us to see firsthand how powerful and flexible relational databases can be when properly structured.

On the application side, building a full-featured CRUD system with PHP and MySQL challenged us to think carefully about user experience, validation, and error handling. Integrating features such as notifications, reminders, and group collaboration required us to coordinate both backend logic and frontend usability. Designing an intuitive, responsive interface with Bootstrap and incorporating visual elements like charts and calendars helped us appreciate the impact of good UI/UX on user engagement.

Collaboration was key to our success. We divided tasks, communicated regularly, and supported each other in troubleshooting and refining features. We encountered challenges, such as handling edge cases in group assignments and ensuring real-time feedback for users, but overcoming these obstacles strengthened our problem-solving skills and teamwork.

Overall, TaskTrackr not only improved our technical abilities in web development and database management but also taught us valuable lessons in project planning, communication, and delivering a robust, user-friendly product. We are proud of what we accomplished and confident that our system meets the highest standards set by the project rubric.