Task Wise Database Schema

This database model consists of five tables: **users, user_profile_stats, tasks, time_slots and suggestions**. They are interconnected through foreign keys as described below.

“users” Table:

- id: Primary key, auto-incremented integer.
- name: The user's full name.
- email: Unique email address for the user.
- password: User's password.
- created_at: Timestamp indicating when the user record was created.
- updated_at: Timestamp indicating the last update to the user record.

“tasks” Table:

- id: Primary key, auto-incremented integer.
- user_id: Foreign key referencing `users.id`, indicating which user the task belongs to.
- task_name: The name of the task.
- duration: Duration of the task (in minutes).
- deadline: Deadline date and time for task completion.
- status: Task status, noted as either "pending" or "done".
- created_at: Timestamp indicating when the task was created.
- updated_at: Timestamp indicating the last update to the task.

“time_slots” Table:

- id: Primary key, auto-incremented integer.
- task_id: Foreign key referencing `tasks.id`, linking the time slot to its respective task.
- start_time: Timestamp for the start of the scheduled time slot.
- end_time: Timestamp for the end of the scheduled time slot.
- created_at: Timestamp when the time slot record was created.
- updated_at: Timestamp when the time slot record was last updated.

“suggestions” Table:

- id: Primary key, auto-incremented integer.
- user_id: Foreign key referencing `users.id`, indicating the user for whom the suggestion is provided.
- suggestion_text: A 50-word suggestion aimed at improving productivity.
- updated_at: Timestamp indicating when the suggestion was last updated.

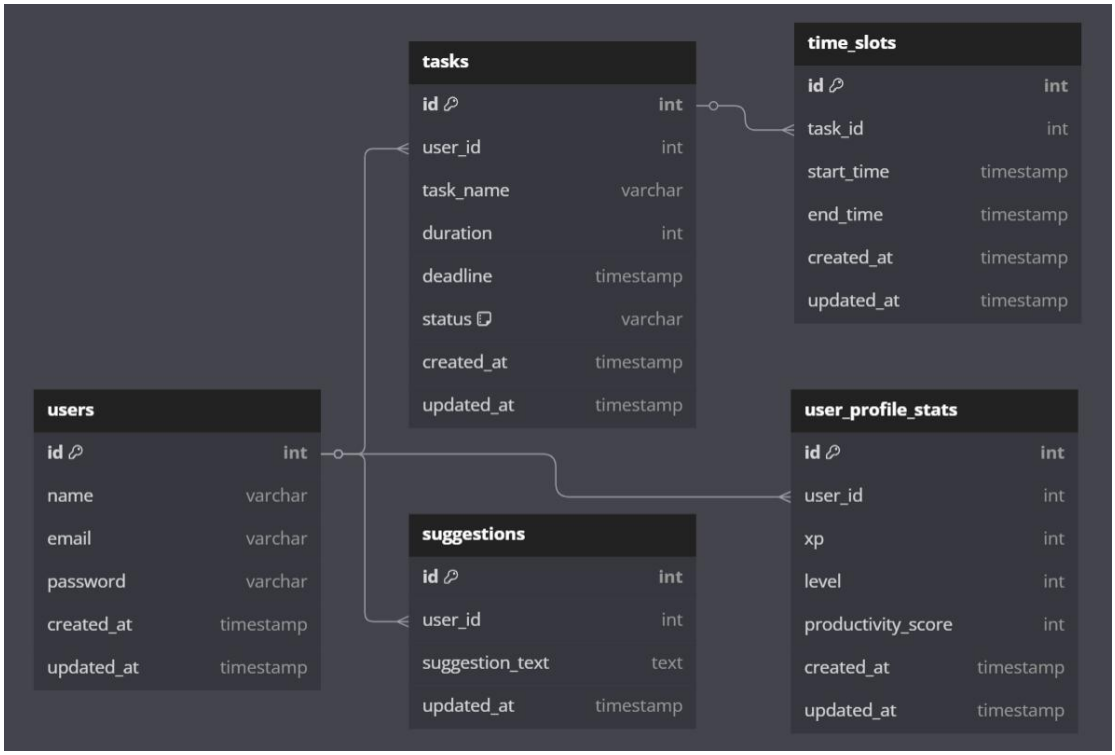
“user_profile_stats” Table:

- id: Primary key, auto-incremented integer.
- user_id: Foreign key referencing `users.id`, linking profile statistics to a specific user.
- xp: Experience points accumulated by the user.
- level: The user's level derived from XP.
- productivity_score: The productivity score for the user (moved from the `users` table).
- created_at: Timestamp indicating when the profile stats record was created.
- updated_at: Timestamp indicating the last update to the profile stats record.

Relationships

- Tasks to Users: Each task is linked to one user through `tasks.user_id > users.id` (many tasks can belong to one user).
- Time Slots to Tasks: Each time slot is linked to one task via `time_slots.task_id > tasks.id`, allowing a task to have multiple scheduled time slots.
- Suggestions to Users: Each suggestion is linked to one user through `suggestions.user_id > users.id` (conceptually one-to-one).
- User Profile Stats to Users: Each user has one associated profile stats record via `user_profile_stats.user_id > users.id`.

Visualization of the schema:



Database Choice and Rationale:
Chosen Database: MongoDB
Reasoning:

- Seamless Integration with MERN: MongoDB is the NoSQL database of choice for MERN (MongoDB, Express, React, Node) applications, offering a natural fit with JavaScript/JSON data structures.
- Flexible Schema Design: MongoDB allows for dynamic schemas, which is ideal for iterative development and evolving requirements like gamification features.
- Scalability: Its document-based model scales well with growing data, making it suitable for an app that handles user tasks, profiles, and suggestions.