One potential improvement for created todo app is adding **user** **authentication and authorization**. This feature will allow multiple users to have their own accounts and manage their own tasks, enhancing the application's functionality and security.

Benefits:

* Personalization: Each user can have their own set of tasks, improving user experience by personalizing the app.
* Security: User-specific tasks are kept private and secure, ensuring that users only access their own data.
* Scalability: Provides a foundation for more advanced features like task sharing or collaboration in the future.

**Implementation Steps:**

Authentication:

Add User Models: Create a new collection in MongoDB for storing user data. Each user document will include fields for username, hashed password, and any other relevant information.

Registration and Login: Implement routes for user registration and login.

Password Hashing: Use a library like bcrypt to hash user passwords before storing them in the database.

Authorization:

Protect Routes: Use Flask-Login's login\_required decorator to protect routes so that only authenticated users can access them.

Associate Tasks with Users: Modify tasks collection to include a reference to the user who created the task.

Frontend Changes:

Login/Logout Forms: Create login and registration forms. Update task views to only show tasks associated with the logged-in user.

User Feedback: Implement user feedback messages for successful registration, login, and logout.

Technologies:

Flask-Login: For handling user sessions and managing authentication.

Flask-WTF: For handling form submissions and validation.

bcrypt: For secure password hashing.

MongoDB: To store user and task data.