Role-Based Task Tracking System (RBTTs)

Backend POC Documentation

Table of Contents

1. Project Overview

RBTTs is a backend-only POC designed to streamline task management by providing role-specific access, accountability, and secure collaboration.

Goals:

- Centralized task tracking
- Role-based access control
- Secure authentication
- Logging and reporting of activities
- Project-based task organization

2. S

2.1 User Roles & Permissions

Admin: Create/update/delete projects & tasks, assign tasks, manage users

Manager: View tasks, assign tasks to developers/testers, update project status

Developer: Update task status, add remarks, view assigned tasks

Tester: Update testing status, add remarks, view assigned tasks

2.2 T

- Create, assign, and update tasks within projects
- Developers/Testers update task status and add remarks
- Admins/Managers track project-wise task completion and progress
- CRUD operations for projects

2.3 A

- JWT-based authentication for secure sessions
- Password hashing using bcrypt
- Logging of login attempts and account locking

2.4

- Generate task reports filtered by project, role, or status
- Admins can audit user and task activity

3. S

Backend-only Architecture:

User (via Postman/HTTP client) -> FastAPI Backend -> MongoDB/MySQL

JWT authentication -> Role-based access -> Project & Task APIs -> Logging

4. D

Users Collection: user_id, name, email, role, password_hash, created_at

Projects Collection: project_id, name, description, status

Tasks Collection: task_id, title, description, project_id, assigned_to_dev, assigned_to_tester, status, remarks, created_at

LoginAttempts Collection: user_email, timestamp, success

5. T

Backend: Python, FastAPI

Database: MongoDB / MySQL

Authentication: JWT, Bcrypt

Testing Tools: Postman / HTTP client

6. A

Endpoint	Method	Role Access	Description
/users/register	POST	Public	Register new user
/users/login	POST	Public	Login and get JWT token
/projects/	POST	Admin/Manager	Create new project
/projects/{id}	GET	All	Get project details
/projects/{id}	PUT	Admin/Manager	Update project info
/projects/{id}	DELETE	Admin	Delete a project
/tasks/	POST	Admin/Manager	Create task within a project
/tasks/{id}	GET	All	Get task details
/tasks/{id}	PUT	Admin/Manager	Update task
/tasks/{id}/status	PUT	Developer/Tester	Update task status/remarks
/tasks/my	GET	Developer/Tester	View assigned tasks within projects

7. Development Workflow

1. Setup FastAPI backend and MongoDB
2. Implement user registration, login, and roles
3. Develop project CRUD APIs
4. Develop task CRUD APIs with project association
5. Implement logging for login and task/project updates

6. Test APIs using Postman / HTTP client

8. C

Secure authentication: JWT + password hashing

Role-based access: Middleware validation

Login logging: Dedicated collection + account lock

Task assignment errors: Validate user existence

Project-task linkage: Ensure tasks are linked to valid projects

9. F

- Push notifications for task/project updates
- Analytics dashboard per project
- Integration with Jira/Trello
- Multi-language support

10.

RBTTs backend-only POC ensures structured task and project management, secure role-based access, and efficient collaboration through APIs.