

Assignment: Team Management Platform

Overview:

Your task is to create a web platform that allows an admin to manage multiple teams and for team members to collaborate. This platform will include two dashboards: one for the admin and one for team members. Both dashboards should have their own set of features, as outlined below.

Functional Requirements:

1. User Authentication:

- Implement proper authentication for both admin and team members.
- Admin and team members should log in to access their respective dashboards.
- Use JWT or any other preferred authentication method.

2. Admin Dashboard:

- **View Teams:** Admin should be able to see all teams and their members.
- **Create/Delete Teams:** Admin can create new teams or delete existing teams.
- **Manage Teams:** Admin can add or remove team members from any team.

3. Team Dashboard:

- **Document Management:**
 - Team members should be able to upload documents, files, and images.
 - You can choose any method to store the files (e.g., locally, using a third-party service like AWS S3, Cloudinary, etc.).
- **To-Do List:**
 - Each team member should be able to create, read, update, and delete (CRUD) items in their personal to-do list.

Technical Requirements:

- **MERN Stack:** Use MongoDB, Express, React, and Node.js to build this platform.
- **Database:** MongoDB should be used for storing user data, team data, and to-do lists.
- **File Storage:** Choose any method for file storage, such as cloud storage (AWS S3, Cloudinary) or local file storage.
- **State Management:** Implement appropriate state management on the front-end (e.g., Redux or Context API).

Evaluation Criteria:

- **Functionality:** Does the platform meet the requirements?
- **Code Quality:** Clean, modular, and well-documented code.
- **User Experience:** The UI/UX of both the admin and team dashboards.
- **Authentication:** Proper implementation of authentication.
- **File Management:** How well file uploads and storage are handled.
- **To-Do List Functionality:** Smooth CRUD operations.

Bonus:

- Implement role-based access control (RBAC) so that users with admin privileges can access the admin dashboard and team members can only access their team's dashboard.

Submission:

- Please submit the GitHub repository link OR project zip file (without node_modules) along with deployment details on Email: hack2hire@skillmingle.in.
- Ensure that the README file contains instructions on how to run the project locally.
- Deadline: 12 October, 2024 (Deadline should be strictly followed)

We look forward to reviewing your submission. Good luck, and feel free to reach out if you have any questions!

Best regards,
Harsh Tiwari
Skill Mingle
hack2hire@skillmingle.in