A company, **TeamCollab**, is building a project management tool that allows teams to collaborate on projects. The tool needs an API to manage users, projects, tasks, and comments. The API will be consumed by their front-end web application and mobile application.

Task Details:

1. Database Plan:

Design the database schema for the project management application. The schema should include the following tables:

- Users: Store user details.

- I'd: Primary Key
- Username: String (Unique)
- Email: String (Unique)
- Password: String
- First_name: String
- Last_name: String
- Date_joined: DateTime

- Projects: Store project details.

- I'd: Primary Key
- Name: String
- Description: Text
- Owner: Foreign Key (to Users)
- Created_at: DateTime

- Project Members: Store project members.

- I'd: Primary Key
- Project: Foreign Key (to Projects)
- User: Foreign Key (to Users)
- Role: String (Admin, Member)

- Tasks: Store task details.

- I'd: Primary Key
- Title: String
- Description: Text
- Status: String (To Do, In Progress, Done)
- Priority: String (Low, Medium, High)
- Assigned_to: Foreign Key (to Users, nullable)
- Project: Foreign Key (to Projects)
- Created_at: DateTime
- Due_date: DateTime

- Comments: Store comments on tasks.

- I'd: Primary Key
- Content: Text
- User: Foreign Key (to Users)
- Task: Foreign Key (to Tasks)
- Created_at: DateTime

2. REST API Endpoints:

Develop the following REST API endpoints using Django and Django REST Framework:

- Users

- **Register User** (POST /api/users/register/): Create a new user.
- **Login User** (POST /api/users/login/): Authenticate a user and return a token.
- **Get User** Details (GET /api/users/{id}/): Retrieve details of a specific user.
- **Update User** (PUT/PATCH /api/users/{id}/): Update user details.
- **Delete User** (DELETE /api/users/{id}/): Delete a user account.

- Projects

- **List Projects** (GET /api/projects/): Retrieve a list of all projects.
- **Create Project** (POST /api/projects/): Create a new project.
- **Retrieve Project** (GET /api/projects/{id}/): Retrieve details of a specific project.
- **Update Project** (PUT/PATCH /api/projects/{id}/): Update project details.
- **Delete Project** (DELETE /api/projects/{id}/): Delete a project.

- Task

- **List Tasks** (GET /api/projects/{project_id}/tasks/): Retrieve a list of all tasks in a project.
- **Create Task** (POST /api/projects/{project_id}/tasks/): Create a new task in a project.
- **Retrieve Task** (GET /api/tasks/{id}/): Retrieve details of a specific task.
- **Update Task** (PUT/PATCH /api/tasks/{id}/): Update task details.
- **Delete Task** (DELETE /api/tasks/{id}/): Delete a task.

- Comments

- **List Comments** (GET /api/tasks/{task_id}/comments/): Retrieve a list of all comments on a task.
- **Create Comment** (POST /api/tasks/{task_id}/comments/): Create a new comment on a task.
- **Retrieve Comment** (GET /api/comments/{id}/): Retrieve details of a specific comment.
- **Update Comment** (PUT/PATCH /api/comments/{id}/): Update comment details.
- **Delete Comment** (DELETE /api/comments/{id}/): Delete a comment.

3. Implementation Steps:

- Set up the Django Project:

- Initialize a new Django project.
- Set up the project configurations and create a new app for the project management functionalities.

- Design the Database Schema:

- Define the models according to the database schema plan.
- Use Django's ORM to create relationships between models.
- Migrate the database to create the necessary tables.

- Implement the REST API:

- Use Django REST Framework to create serializers for each model.
- Develop viewsets for each resource and register them with the router.
- Implement authentication using Django REST Framework or JWT token authentication.

- Documentation:

- Use tools like Swagger or Postman to document the API.
- Provide clear instructions on how to set up and use the API.

4. Submission:

- Push your code to a Git repository.
- Include a README file with instructions on how to set up the project locally, migrate the database, and run the server.
- Provide API documentation.
- Use this google form for code submission [Form Link]

+880 1620-841623
vitasoftsolution@gmail.com
Corporate office: 677, Brothers Tower, east
Dholaipar, kadamtoli, dhaka-1236