API Documentation for User, Role, and Permission Management

**1. Project Setup**

**1.1. Prerequisites**

Before you begin, ensure the following:  
- PHP >= 8.2  
- Composer  
- Laravel 11.x or higher  
- MySQL or any other supported database

**1.2. Installation Instructions**

1. Clone the Repository:  
 Clone the repository to your local machine:  
 git clone <https://github.com/anwarhossain4035/laravel-role-permission-api.git>  
  
2. Install Dependencies:  
 Navigate to the project directory and install dependencies:  
 cd project-directory  
 composer install  
  
3. Set Up .env File:  
 Copy the .env.example file to .env:  
 cp .env.example .env  
 Update the .env file with your database credentials, APP\_KEY, etc.  
  
4. Generate Application Key:  
 php artisan key:generate  
  
5. Run Migrations:  
 Run the migrations to set up the database schema:  
 php artisan migrate  
  
6. Install Passport:  
 Install Laravel Passport for API authentication:  
 composer require laravel/passport  
 php artisan passport:install  
 php artisan migrate  
  
7. Set Up User Authentication:  
 In app/Models/User.php, make sure the HasApiTokens trait is included:  
 use Laravel\Passport\HasApiTokens;  
  
8. Clear Config Cache:  
 php artisan config:clear  
 php artisan cache:clear

**2. API Endpoints**

**2.1. Authentication**

**Login**

POST /api/login  
  
Request Body:  
{  
 "email": "user@example.com",  
 "password": "password"  
}  
  
Response:  
{  
 "access\_token": "your-access-token",  
 "token\_type": "Bearer"  
}

**2.2. User Management**

**Get All Users**

GET /api/users  
  
Headers:  
Authorization: Bearer {access\_token}  
  
Response:  
[  
 {  
 "id": 1,  
 "name": "John Doe",  
 "email": "john@example.com"  
 },  
 {  
 "id": 2,  
 "name": "Jane Doe",  
 "email": "jane@example.com"  
 }  
]

**Create User**

POST /api/users  
  
Request Body:  
{  
 "name": "New User",  
 "email": "newuser@example.com",  
 "password": "password",  
 "role": ["admin", "editor"]  
}  
  
Response:  
{  
 "id": 3,  
 "name": "New User",  
 "email": "newuser@example.com",  
 "roles": ["admin", "editor"]  
}

**2.3. Role Management**

**Get All Roles**

GET /api/roles  
  
Headers:  
Authorization: Bearer {access\_token}  
  
Response:  
[  
 {  
 "id": 1,  
 "name": "Admin"  
 },  
 {  
 "id": 2,  
 "name": "Editor"  
 }  
]

**Create Role**

POST /api/roles  
  
Request Body:  
{  
 "name": "Moderator"  
}  
  
Response:  
{  
 "id": 3,  
 "name": "Moderator"  
}

**2.4. Permission Management**

**Get All Permissions**

GET /api/permissions  
  
Headers:  
Authorization: Bearer {access\_token}  
  
Response:  
[  
 {  
 "id": 1,  
 "name": "view\_posts"  
 },  
 {  
 "id": 2,  
 "name": "edit\_posts"  
 }  
]

**Create Permission**

POST /api/permissions  
  
Request Body:  
{  
 "name": "delete\_posts"  
}  
  
Response:  
{  
 "id": 3,  
 "name": "delete\_posts"  
}

**2.5. Assign/Remove Permissions from Users**

**Get User Permissions**

GET /api/users/{user}/permissions  
  
Headers:  
Authorization: Bearer {access\_token}  
  
Response:  
[  
 {  
 "id": 1,  
 "name": "view\_posts"  
 },  
 {  
 "id": 2,  
 "name": "edit\_posts"  
 }  
]

**Assign Permission to User**

POST /api/users/{user}/permissions  
  
Request Body:  
{  
 "permissions": [  
 "view\_posts",  
 "edit\_posts"  
 ]  
}  
  
Response:  
{  
 "message": "Permissions assigned successfully."  
}