Laravel Developer Pre-Interview Task List

1. Build a Simple API with Authentication

- **Objective**: Create a small RESTful API for a hypothetical e-commerce application.
- **Requirements**:
  - Implement CRUD functionality for products (fields: name, description, price, stock).
  - Secure the API using Laravel Sanctum or Passport for token-based authentication.
  - Provide basic validation for each endpoint.
  - **Deliverables**: Endpoints documentation and sample requests for each API action.

2. Database Optimization & Query Challenge

- **Objective**: Work with a sample dataset to demonstrate query optimization.
- **Requirements**:
  - Given a database table with a large number of rows, optimize queries to fetch the top 5 best-selling products and the most recent customer orders.
  - Add appropriate indexes to improve query performance and explain your indexing strategy.
  - Use Eloquent and query builder for fetching data efficiently.

3. Implement Role-Based Access Control (RBAC)

- **Objective**: Set up roles and permissions for a basic admin dashboard.
- **Requirements**:
  - Create two roles: Admin and Editor.
  - Allow Admins to manage all content and Editors to only update certain sections (e.g., articles).
  - Use Laravel policies or gates to restrict actions based on roles.

4. Implement a Simple Queue for Email Notifications

- **Objective**: Demonstrate familiarity with Laravel's queue system.
- **Requirements**:
  - Create a functionality where users register and receive a welcome email sent via a queued job.
  - Configure a queue driver (e.g., database or Redis) and set up the email job using Laravel’s job dispatching system.

7. Optimize a Complex Query with Eloquent

- **Objective**: Assess optimization skills with complex queries.
- **Requirements**:
  - Given two tables, orders and products, write an optimized query to fetch orders with the details of each product, grouped by product category.
  - Use eager loading to prevent N+1 issues and explain the approach taken.
  - **Deliverables**: Code snippet of the optimized query with a brief explanation of the optimizations.

9. Deploy the Application to a Server

- **Objective**: Deploy the API application to a staging environment.
- **Requirements**:
  - Set up a server (or use a cloud service like AWS or DigitalOcean) and deploy the application.
  - Ensure the application is accessible via a public URL and that environment variables are correctly set.
  - **Deliverables**: Deployment steps documentation and a link to the live application.

Create a Github Repository and upload the project into this. Also create proper documentation to demonstrate the task that you have done. Send the github repository link, application public URL and documentation in my email alifhossain174@gmail.com

Thanks