Laravel MVC Interview Task: “Basic HRM System”

Objective:

Build a simplified Human Resource Management system demonstrating:

- Laravel MVC
- Eloquent relationships
- Resourceful routing
- Validation
- Blade templating
- JavaScript/jQuery for dynamic UI behavior

Requirements:

You are required to build a simplified HRM (Human Resource Management) module using Laravel MVC, demonstrating CRUD operations, Eloquent relationships, validation, Blade templating, and basic JavaScript/jQuery interactivity.

Build an application that manages Employees, Departments, and Skills, where:

1. Implement User Authentication using Laravel’s built-in auth system.
    - The system must include **user registration and login** functionality.
    - Only authenticated users may access the HRM module.
2. Each Employee belongs to one Department.
3. Each Employee can have multiple Skills.
4. The system must include CRUD pages for Employees and create/list pages for Departments and Skills.
5. On the *Employee Create/Edit page*, the user must be able to:
    - Select a Department (dropdown).
    - Use JS/jQuery to **add/remove Skills dynamically** (can be inside a multiselect field or appear as repeatable skill input fields with remove buttons).
6. On the *Employee List page*, implement **JS/jQuery-based filtering** so when the user selects a Department from a dropdown, the list updates dynamically **without reloading the page**.
7. Implement server-side validation for:
    - **Employee**: first_name, last_name (required); email (required, unique, valid); department_id (required); skills (optional array of IDs).
    - **Department**: name (required, unique).

- **Skill**: name (required, unique).

8. Implement model relationship correctly to ensure that skills **sync** correctly when editing an employee using laravel functions.
9. Provide Blade templates for:
    - Listing employees
    - Creating/editing employees
    - Showing employee details (including department and skills)
    - Listing/creating departments
    - Listing/creating skills
    - User login/registration/auth views
10. Include at least **one additional jQuery-based interaction** of your choice (e.g., real-time email uniqueness check via AJAX, show/hide extra info, animations, etc.).

Your final solution should demonstrate **clean MVC separation**, **correct Eloquent usage**, **proper use of JS/jQuery**, **secure authentication**, and **organized Blade views**.