Part 1: Core Models & Relations

* Create basic models with essential fields:

- User: first\_name, last\_name, email, password

- Project: name, status

- Timesheet: task\_name, date, hours

* You need to create the proper relationships between the models, user can be assigned to more than one project, and each project can have many users, every user can log his timesheets to many projects, but each timesheets record should be linked with only one project and only one user.

Part 2: EAV (Entity-Attribute-Value) Implementation

Add a dynamic attributes system for Projects:

- Create models: Attribute, AttributeValue

- Attribute fields: name, type (text, date, number, select)

- AttributeValue fields: attribute\_id, entity\_id, value

- Projects should support dynamic fields like department, start\_date, end\_date through EAV

- Include API endpoints to:

- Create/Update attributes

- Set attribute values when creating/updating projects

- Fetch projects with their dynamic attributes

- Filter projects by dynamic attribute values

Part 3: API Endpoints

Create RESTful endpoints with authentication:

- Auth: /api/register, /api/login, /api/logout

- Standard CRUD for each model:

- GET /api/{model}

- GET /api/{model}/{id}

- POST /api/{model}

- PUT /api/{model}/{id}

- DELETE /api/{model}/{id}

Part 4: Filtering

Implement a flexible filtering system:

- Support filtering on both regular and EAV attributes

- Example: GET /api/projects?filters[name]=ProjectA&filters[department]=IT

- Support basic operators (=, >, <, LIKE)

Requirements:

- Use Laravel Passport for authentication

- Include database migrations and seeders

- Use proper validation and error handling

- Follow PSR standards and Laravel best practices

Deliverables:

- GitHub repository

- SQL dump

- README with:

- Setup instructions

- API documentation

- Example requests/responses

- Test credentials