

Company Management System

A Django-based web application to manage employees, departments, companies, and projects. This system includes features such as scheduling performance reviews, providing feedback, and managing employee performance workflows.

Features

- **Employee Management:** Add, view, update, and manage employees in various departments.
- **Department Management:** Create and manage departments within the company.
- **Company Management:** Manage company information and associate departments with companies.
- **Project Management:** Create, assign, and track projects.
- **Performance Review Workflow:** Schedule and manage performance reviews for employees, including feedback submission, approval, and rejection.

Usage

Admin Panel

To manage the system, access the Django admin panel at:

<http://127.0.0.1:8000/admin/>

Login with the superuser credentials you created.

API Endpoints

This project exposes a set of API endpoints for CRUD operations on the models:

- **Companies API:** `/api/companies/`
- **Departments API:** `/api/departments/`
- **Employees API:** `/api/employees/`
- **Projects API:** `/api/projects/`

Each endpoint supports standard HTTP methods (GET, POST, PUT, DELETE).

Dependencies

- **Django:** Framework used for the web application.
- **Django Rest Framework:** Provides a powerful toolkit for building Web APIs.
- **Viewflow:** Manages stateful workflows, integrated with the performance review system.
- **SQLite:** Database used for persistent storage.
- **Django-FSM:** For handling state transitions in the performance review workflow.

Contributing

Contributions are welcome! If you'd like to contribute, follow these steps:

1. Fork the repository.
2. Create a new branch.
3. Make your changes and commit them.
4. Push to your forked repository.
5. Create a pull request describing your changes.

Checklist

Data Models

- ☒ **User Accounts**
 - ☒ Created User model with fields: username, email, role.
- ☒ **Company**
 - ☒ Created Company model with fields: company_name, num_departments, num_employees, num_projects.
 - ☒ Auto-calculation for num_departments, num_employees, and num_projects.
- ☒ **Department**
 - ☒ Created Department model with fields: company, department_name, num_employees, num_projects.
 - ☒ Auto-calculation for num_employees and num_projects.

- ☒ **Employee**
 - ☒ Created Employee model with fields: company, department, employee_name, email, mobile, address, designation, hired_on, days_employed.
 - ☒ Auto-calculation for days_employed.
- ☒ **Project**
 - ☒ Created Project model with fields: company, department, project_name, description, start_date, end_date, assigned_employees.

Workflow

- ☒ **Employee Performance Review Workflow**
 - ☒ Defined stages: Pending Review, Review Scheduled, Feedback Provided, Under Approval, Review Approved, Review Rejected.
 - ☒ Defined transitions between stages.
 - ☒ Integrated transitions in the workflow for performance review approval.

Security & Permissions

- ☒ Implemented **role-based access control**: Admin, Manager, Employee.
- ☒ Secured authentication using **JWT Tokens**.

APIs

- ☒ Created **RESTful API** endpoints for CRUD operations on Company, Department, Employee, and Project models.
- ☒ Ensured API follows **RESTful conventions** (GET, POST, PATCH, DELETE).
- ☒ Secured the API endpoints and ensured **data security**.

Testing

- ☒ Wrote **unit tests** to validate individual components.

- ☒ Added **integration tests** to ensure smooth interaction between components.

Logging (Bonus)

- ☒ Implemented **logging** to track application behavior and errors.
 - ☒ Logs are detailed but do not expose **sensitive information**.
-

Assumptions and Considerations

1. **Authentication & Authorization:**
The application uses **JWT tokens** for secure authentication. Roles and permissions are defined at the model level to ensure appropriate access.
2. **Performance Review Workflow:**
The transitions in the performance review workflow are designed to handle most real-world scenarios, such as rejections and updates, ensuring that the review process remains organized.
3. **API Security:**
API endpoints are secured and can only be accessed by authorized users based on roles. Sensitive data, such as passwords and personal employee details, are encrypted or not exposed in the API responses.
4. **Deployment:**
This project has been developed with a development setup in mind. Before deployment, it's important to adjust settings such as the database configuration, logging, and debugging settings.

License

This project is licensed under the MIT License - see the [LICENSE](#) file for details.