README Structure:

Project Title: HRManagement System

Description:

This Django-based HRManagement system helps manage departments within an organization.

It allows users to create, view, update, and delete departments.

Departments are marked as active or inactive based on their status.

Features:

- Department creation
- Department listing with edit and delete options
- Department status management

Technologies:

- Django
- Python
- MySQL (for database management)

Installation:

1. Clone the repository
2. Install dependencies using: pip install -r requirements.txt
3. Set up the database
4. Run migrations using: python manage.py migrate
5. Run the server with: python manage.py runserver

Project Structure:

- DepartmentApp/: Contains the main app for department management.
  - models.py: Defines the Department model.
  - views.py: Contains views for CRUD operations on departments.
  - urls.py: URL routing for the Department app.
  - templates/: HTML templates for the views.

- HRManagement/: Main project folder containing configurations.

- manage.py: Django's script to manage the project.

Usage:

- Navigate to the home page to view a list of departments.

- Click "Create Department" to add a new department.

- Click "Edit" next to a department to update its information.

- Click "Delete" next to a department to mark it as inactive.

"""

HR MANAGEMENT System

Overview

The HR MANAGEMENT System is a web-based application built with Django that enables management of employee departments. It allows users to create, update, delete, and list departments within the organization. This project uses Django's powerful ORM (Object-Relational Mapping) to interact with a MySQL database.

Features

- **Home Page**: Displays a list of active departments.
- **Create Department**: A form to create a new department.
- **Update Department**: Allows updating an existing department's name and description.
- **Delete Department**: Marks a department as inactive instead of deleting it from the database.

Technologies Used

- **Django**: Web framework for rapid development and clean, pragmatic design.
- **MySQL**: Relational database management system used for storing department data.
- **HTML/CSS**: For designing user interfaces.

Project Structure

Copy code

```
HRManagement/
 ├── DepartmentApp/
|    ├── migrations/
|    ├── models.py
|    ├── views.py
|   ├── urls.py
|   ├── templates/
|   |   ├── index.html
|   |   ├── createdepartment.html
|   |   ├── updatedepartment.html
|   ├── admin.py
|   ├── apps.py
|   └── tests.py
├── HRManagement/
|   ├── settings.py
|   ├── urls.py
|   └── wsgi.py
└── manage.py
```

Setup Instructions

Prerequisites

- Python 3.x
- Django 5.x
- MySQL

Installation

1. **Clone the repository**:

bash

Copy code

```
git clone <repository-url>
cd HRManagement
```

2. **Create and activate a virtual environment**:

bash

Copy code

```
python -m venv venv
source venv/bin/activate  # On Windows use `venv\Scripts\activate`
```

3. **Install dependencies**:

bash

Copy code

```
pip install -r requirements.txt
```

4. **Configure the database**:
   - Set up a MySQL database and update the DATABASES setting in HRManagement/settings.py to match your database credentials.
5. **Migrate database**:

bash

Copy code

```
python manage.py migrate
```

6. **Run the server**:

bash

Copy code

```
python manage.py runserver
```

7. **Visit the app**: Open your browser and go to http://127.0.0.1:8000/ to view the application.

Application Workflow

1. **Home Page**: Displays all active departments.
2. **Create Department**: You can create a department by entering its name and description.
3. **Update Department**: Editing a department allows you to change its name or description.
4. **Delete Department**: Mark a department as inactive.

Models

Department Model

python

Copy code

```
class Department(models.Model):
    dept_id = models.AutoField(primary_key=True)
    dept_name = models.CharField(max_length=100)
    description = models.CharField(max_length=300)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    status = models.BooleanField(default=True)
    def __str__(self):
        return self.dept_name
```

Views

Home View (home)

Displays the list of active departments.

Create Department View (createDepartment)

Handles the creation of new departments.

Update Department View (updateDepartment)

Allows updating department details.

Delete Department View (Deletedepartment)

Marks a department as inactive.

URL Configuration

The project uses Django's URL routing system for navigation. Here are some key URLs:

- /: Home page showing all active departments.
- /createdepartment: Create new department.
- /edit/<dept_id>: Update department details.
- /delete/<dept_id>: Mark department as inactive.

Templates

- index.html: Displays the list of departments.
- createdepartment.html: Form for creating a department.
- updatedepartment.html: Form for updating a department.

Conclusion

This HR MANAGEMENT System is designed for managing the basic structure of an organization by handling departments. It is easy to extend and modify as needed for other human resource management functionalities.