# Django Multi-Organization Management System

📌 **Table of Contents**

## 🌟 **Project Overview**

What is this Project?
The Django Multi-Organization Management System is a web application designed to help organizations manage users, roles, and access controls efficiently. Whether you're a small startup or a large enterprise, this system provides a flexible solution for managing multiple organizations within a single platform.

**Key Features**
- 🔒 Role-Based Access Control
- 👥 Multi-Organization Support
- 📊 User Management
- 🔄 Easy User Data Import/Export
- 🛡️ Secure Authentication

## 💻 **System Requirements**

Minimum Requirements
- Operating System: Windows, macOS, or Linux
- Python: Version 3.8 or higher
- Disk Space: 500 MB free
- RAM: 4 GB minimum (8 GB recommended)

Required Software
1. Python 3.8+
2. pip (Python Package Installer)
3. Git
4. Virtual Environment tool (venv recommended)

## 🔧 **Installation Guide**

Step 1: Prepare Your Environment

1. Install Python
   - Download from official Python website: https://www.python.org/downloads/
   - Ensure "Add Python to PATH" is checked during installation
   - Verify installation by running in terminal/command prompt:
   ```bash
   python --version
   pip --version
   ```

2. Install Git
  - Download from: https://git-scm.com/downloads
  - Verify installation:
   ```bash
   git --version
   ```

Step 2: Clone the Project

```bash
#Open terminal/command prompt
git clone https://github.com/Nitishgithub2005/Multi_org_mng.git
cd multi-org-management
```

Step 3: Set Up Virtual Environment

```bash
# Create virtual environment
python -m venv venv

Activate virtual environment
 On Windows
venv\Scripts\activate

# On macOS/Linux
source venv/bin/activate
```

Step 4: Install Dependencies

```bash
# Upgrade pip
pip install --upgrade pip

# Install project dependencies
pip install -r requirements.txt

# Install additional required package
pip install django-import-export
```

🚀 **Project Setup**

Database Configuration

1. Apply Migrations:
  ```bash
  python manage.py makemigrations
  python manage.py migrate
  ```

2. Create Superuser(A superuser has already been provided by me below, so this step is optional):

```bash
python manage.py createsuperuser
```

Follow the prompts to create an admin account

3.Update Settings

Open `multi_org_mgmt/settings.py` and ensure `import_export` is in `INSTALLED_APPS`:

```python
INSTALLED_APPS = [
 # Other apps...
 'import_export',
]
```

🌐 **Running the Application**

```bash
# Start development server
python manage.py runserver
```

🔗 Access the application:
- Local URL: http://127.0.0.1:8000
- Admin Pane: http://127.0.0.1:8000/admin

🔐 **User Roles and Access**

Default User Credentials

Superadmin
- Username: `mainadmin`
- Password: `admin@123`

IT Department

1. Admin
   - Username: `person1`
   - Password: `nitish@123`

2. Editor
   - Username: `editor1`
   - Password: `nitish@123`

3. Viewer
   - Username: `viewer1`
   - Password: `nitish@123`

Research Department

1. Admin
   - Username: `person2`
   - Password: `nitish@123`

2.Editor
- Username: `editor2`
- Password: `nitish@123`

3. Viewer
- Username: `viewer2`
- Password: `nitish@123`

**Role Permissions**
- Admin: Full system access, can add/edit users and organizations
- Editor: Can modify limited user and organizational data
- Viewer: Read-only access to system information

## 🛠️ Troubleshooting

<u>Common Issues & Solutions</u>

1. Dependency Conflicts
   - Ensure you're in the virtual environment
   - Update pip: `pip install --upgrade pip`
   - Recreate virtual environment if persistent issues occur

2. Migration Errors
   - Delete existing migration files in `organizations/migrations/`
   - Run `python manage.py makemigrations`
   - Then `python manage.py migrate`

3. Import/Export Problems
   - Verify data format matches expected schema
   - Check django-import-export documentation

🔬 Advanced Configuration

<u>Performance Optimization</u>
- Use `python manage.py check --deploy` for production readiness check
- Configure static file handling
- Set up proper logging

## 🤝 Contributing
1. Fork the repository
2. Create a new branch
3. Make your changes
4. Submit a pull request

## 📋 Project Structure

```
multi_org_mgmt/
│
├── manage.py
├── multi_org_mgmt/
│   ├── settings.py
│   ├── urls.py
│   └── ...
│
├── organizations/
│   ├── models.py
│   ├── admin.py
│   ├── views.py
│   └── ...
│
└── templates/
 ├── base.html
 ├── organization_list.html
 └── user_list.html
```

## 📞 Support

For issues or questions, please open a GitHub issue or contact nitishmaladakar@gmail.com.