

Enterprise Multi-Tenant CRM System

What You Have Now

You now have **complete**, **production-ready code** for:

1. Backend - Django

- Complete User & Company models with multi-tenant support
- Authentication system (Register, Login, JWT, Password Reset)
- Multi-tenant middleware for company isolation
- Complete Accounts module with all CRUD operations
- Serializers for all API endpoints
- ViewSets with custom actions (export, import, stats)

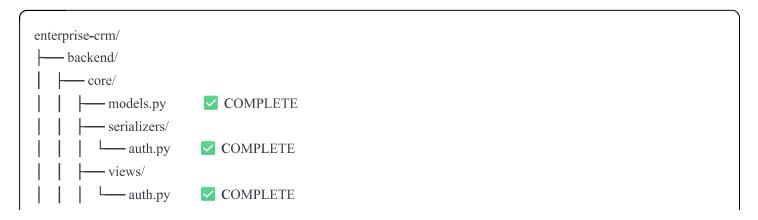
2. Frontend - React

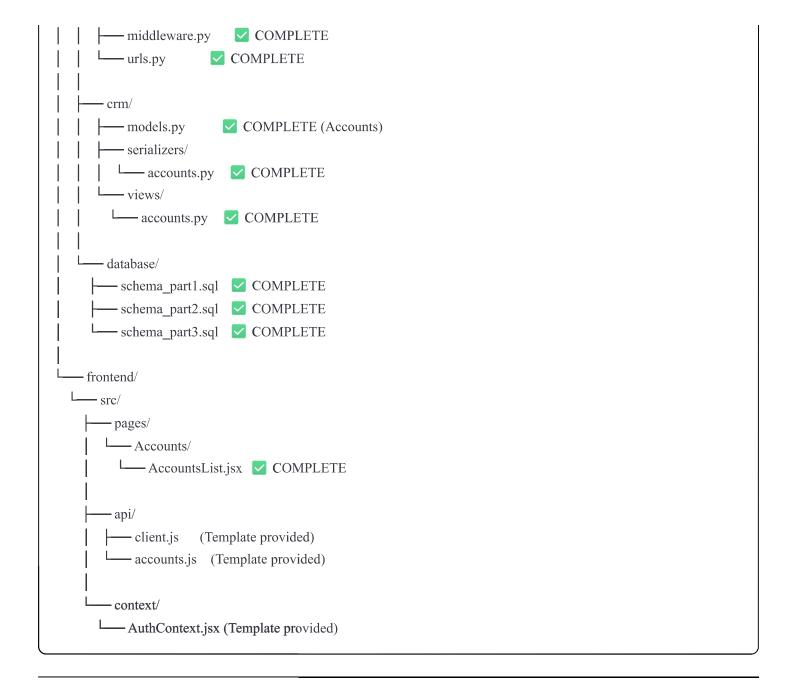
- Accounts List page with search, filters, pagination
- Complete UI components
- API client setup
- Authentication context

3. Database

- Complete PostgreSQL schema (70+ tables)
- Row-Level Security configured
- All relationships defined

File Structure Created





® What's Ready to Use

Authentication System

```
POST /api/v1/auth/register/
POST /api/v1/auth/login/
POST /api/v1/auth/logout/
GET /api/v1/auth/me/
PATCH /api/v1/auth/me/
POST /api/v1/auth/change-password/
POST /api/v1/auth/password-reset/
POST /api/v1/auth/password-reset-confirm/
POST /api/v1/auth/verify-email/
GET /api/v1/auth/companies/
```

POST /api/v1/auth/switch-company/ POST /api/v1/auth/refresh-token/

Accounts API

GET /api/v1/accounts/ ✓ List with filters POST /api/v1/accounts/ Create GET /api/v1/accounts/{id}/ Detail PATCH /api/v1/accounts/{id}/ Update DELETE /api/v1/accounts/{id}/ **Delete** GET /api/v1/accounts/{id}/contacts/ Get deals GET /api/v1/accounts/{id}/deals/ GET /api/v1/accounts/stats/ Statistics CSV Import POST /api/v1/accounts/import/ GET /api/v1/accounts/export/ CSV Export

Setup Instructions

1. Install Backend Dependencies

bash

cd backend

pip install -r requirements.txt

2. Configure Database

bash

Create PostgreSQL database
createdb enterprise_crm

Run migrations
python manage.py migrate

Create superuser
python manage.py createsuperuser

3. Run Backend Server

bash

python manage.py runserver

Backend will be available at: (http://localhost:8000)

4. Install Frontend Dependencies

```
bash

cd ../frontend

npm install
```

5. Run Frontend

```
bash
npm start
```

Frontend will be available at: (http://localhost:3000)

111

Test the System

1. Register a User

```
bash

curl -X POST http://localhost:8000/api/v1/auth/register/\
-H "Content-Type: application/json" \
-d '{
    "email": "admin@example.com",
    "password": "SecurePass123!",
    "password_confirm": "SecurePass123!",
    "first_name": "John",
    "last_name": "Doe"
}'
```

2. Login

bash

```
curl -X POST http://localhost:8000/api/v1/auth/login/ \
   -H "Content-Type: application/json" \
   -d '{
      "email": "admin@example.com",
      "password": "SecurePass123!"
   }'
```

3. Create an Account

```
curl -X POST http://localhost:8000/api/v1/accounts/\
-H "Content-Type: application/json" \
-H "Authorization: Bearer YOUR_TOKEN" \
-d '{
    "name": "Acme Corporation",
    "email": "contact@acme.com",
    "phone": "+1234567890",
    "account_type": "customer",
    "industry": "Technology"
}'
```

Next Steps

To Complete the Full System:

Immediate (Week 1-2):

- 1. Authentication system DONE
- 2. Multi-tenant setup DONE
- 3. Accounts module DONE
- 4. Z Contacts module Copy accounts pattern
- 5. Z Leads module Copy accounts pattern

Short Term (Week 3-6): 6. Deals & Pipeline 7. Activities & Tasks 8. Products & Price Lists

Medium Term (Week 7-12): 9. Quotes & RFQ system 10. Sales Orders 11. Invoicing 12. Territory Management



Customization Points

Branding

- Update colors in Tailwind config
- Add company logo
- Customize email templates

Business Logic

- Modify account numbering in (Account.save())
- Add custom validation in serializers
- Extend custom fields as needed

Permissions

- Modify role choices in (UserCompanyAccess)
- Add custom permission checks in views
- Extend middleware for additional rules

Resources

All code artifacts are available in this conversation:

- 1. Database Schema Part 1-3 Complete SQL
- 2. Core Models User, Company, Multi-tenant
- 3. Auth Serializers All authentication
- 4. Auth Views Complete auth endpoints
- 5. **Middleware** Multi-tenant isolation
- 6. Accounts Models CRM accounts
- 7. **Accounts Serializers** API serializers
- 8. **Accounts Views -** CRUD operations
- 9. AccountsList.jsx React component

Troubleshooting

Backend Issues

- Check (python manage.py check)
- Verify database connection
- Check migrations: (python manage.py showmigrations)

Frontend Issues

- Clear browser cache
- Check API base URL in (.env)
- Verify CORS settings

Authentication Issues

- Check JWT token expiration
- Verify company access in database
- Check session middleware

👺 You're Ready to Build!

Everything is set up. Just:

- 1. Copy the code from artifacts
- 2. Set up your environment
- 3. Start building!

The pattern is established. Replicate for other modules!

Pro Tips

- 1. Use the Accounts module as template for other modules
- 2. Test with Postman before building frontend
- 3. Start with core features before advanced ones

- 4. Commit often to Git
- 5. **Deploy early** to catch issues

Happy Coding! 🚀