**You are aiming to build a comprehensive, self-hosted cybersecurity training platform inspired by KnowBe4, with multi-tenancy (each installation is separate), licensing, demo mode, super admin and admin roles, and flexible feature enablement.**

**This is a significant project. Here’s how I’ll help you proceed:**

1. **Project Structure Proposal**
2. **Key Feature Breakdown**
3. **Stepwise Implementation Plan**
4. **Example Skeleton Implementation**
5. **Documentation Approach**
6. **How to Continue & Expand**

**1. Project Structure Proposal**

**Code**

**cybersecurity-training-platform/**

**│**

**├── app/**

**│ ├── \_\_init\_\_.py**

**│ ├── models.py**

**│ ├── routes/**

**│ │ ├── auth.py**

**│ │ ├── admin.py**

**│ │ ├── superadmin.py**

**│ │ └── training.py**

**│ ├── templates/**

**│ ├── static/**

**│ ├── utils.py**

**│ └── config.py**

**│**

**├── migrations/**

**│**

**├── tests/**

**│**

**├── install/**

**│ └── setup.py**

**│**

**├── requirements.txt**

**├── README.md**

**└── run.py**

**2. Key Feature Breakdown**

* **Modular Training Management (CRUD for modules, assignments)**
* **User Progress Tracking (per user, per module)**
* **Security Reporting (CSV/PDF export, dashboard)**
* **Local SQLite Database (one per installation)**
* **Flask Web UI**
* **Super Admin Features (user/license limits, enable/disable features, demo mode, create admins)**
* **Admin Role (access to enabled features)**
* **Installation Package (pip installable or Docker)**
* **Demo Mode (limits certain features as per super admin config)**

**3. Stepwise Implementation Plan**

1. **Initialize Flask project**
2. **Design User, Role, Feature, License, Module, Progress models**
3. **Implement authentication & role management**
4. **Super admin panel (feature toggles, license management, demo mode)**
5. **Admin panel (filtered by enabled features)**
6. **Training module CRUD & assignment**
7. **Progress tracking & reporting**
8. **Demo mode logic**
9. **Installer script (setup.py or Docker)**
10. **Documentation with explanations for each script**

**4. Example Skeleton Implementation**

**Below is a simplified version of the essential components to get you started. Full implementation will require iteration and expansion.**

**app/\_\_init\_\_.py**

**from flask import Flask**

**from flask\_sqlalchemy import SQLAlchemy**

**from flask\_login import LoginManager**

**db = SQLAlchemy()**

**login\_manager = LoginManager()**

**app/models.py**

**from . import db**

**from flask\_login import UserMixin**

**class User(UserMixin, db.Model):**

**id = db.Column(db.Integer, primary\_key=True)**

**username = db.Column(db.String(150), unique=True)**

**run.py**

**from app import create\_app**

**app = create\_app()**

**if \_\_name\_\_ == "\_\_main\_\_":**

**app.run(debug=True)**

**app/config.py**

**import os**

**class Config:**

**SECRET\_KEY = os.environ.get("SECRET\_KEY", "change\_this\_in\_production")**

**SQLALCHEMY\_DATABASE\_URI = 'sqlite:///cybersec\_training.db'**

**SQLALCHEMY\_TRACK\_MODIFICATIONS = False**

**install/setup.py**

**from setuptools import setup, find\_packages**

**setup(**

**name='cybersecurity-training-platform',**

**version='0.1',**

**packages=find\_packages(),**

**5. Documentation Example**

**README.md**

**# Cybersecurity Training Platform**

**## Overview**

**A self-hosted platform for cybersecurity awareness training, inspired by KnowBe4. Features modular training management, user progress tracking, security reporting, and robust admin controls.**

**6. How to Continue & Expand**

1. **Fill out routes in app/routes/ for authentication, admin, superadmin, and training logic.**
2. **Implement templates for UI in app/templates/.**
3. **Build out reporting logic for progress exports.**
4. **Expand models as needed for more KnowBe4-like modules (phishing simulations, quizzes, etc.).**
5. **Add tests in /tests.**
6. **Provide detailed in-line comments and script-level docstrings for each Python file.**