
✅ 1. Project Structure

```
Project/
|-- app.py
|-- models.py
|-- database.db
|-- templates/
|   |-- home.html
|-- static/
|   |-- style.css
|-- requirements.txt
```

✅ 2. Install Required Libraries

Bash

```
Pip install flask flask_sqlalchemy flask_login
```

✅ 3. Models.py

Python

```
From flask_sqlalchemy import SQLAlchemy
From flask_login import UserMixin

Db = SQLAlchemy()

# Association table for many-to-many relationship
User_roles = db.Table(‘user_roles’,
   Db.Column(‘user_id’, db.Integer, db.ForeignKey(‘user.id’)),
   Db.Column(‘role_id’, db.Integer, db.ForeignKey(‘role.id’))
)

Class User(db.Model, UserMixin):
   Id = db.Column(db.Integer, primary_key=True)
   Username = db.Column(db.String(50), unique=True, nullable=False)
   Password = db.Column(db.String(100))
   Roles = db.relationship(‘Role’, secondary=user_roles, backref=’users’)

Class Role(db.Model):
   Id = db.Column(db.Integer, primary_key=True)
   Name = db.Column(db.String(50), unique=True)
```

✅ 4. App.py

Python

```
From flask import Flask, render_template, redirect, url_for
From models import db, User, Role
From flask_login import LoginManager, login_user, logout_user, current_user, login_required


App = Flask(__name__)
App.config[‘SECRET_KEY’] = ‘secret’
App.config[‘SQLALCHEMY_DATABASE_URI’] = ‘sqlite:///database.db’
Db.init_app(app)


Login_manager = LoginManager()
Login_manager.init_app(app)


@login_manager.user_loader
Def load_user(user_id):
    Return User.query.get(int(user_id))


@app.route(‘/’)
Def home():
    Return render_template(‘home.html’, user=current_user)


@app.route(‘/admin’)
@login_required
Def admin_page():
    If not any(role.name == ‘Admin’ for role in current_user.roles):
        Return “Access Denied: Admins Only”
    Return “Welcome Admin!”
```

```
@app.route(‘/create’)
Def create_roles_users():
  Db.create_all()

  Admin_role = Role(name=’Admin’)
  User_role = Role(name=’User’)

  Admin = User(username=’admin’, password=’admin123’, roles=[admin_role])
  User = User(username=’user’, password=’user123’, roles=[user_role])

  Db.session.add(admin_role)
  Db.session.add(user_role)
  Db.session.add(admin)
  Db.session.add(user)
  Db.session.commit()
  Return “Database and roles created successfully!”
```

Would you like me to send this as a Word or text file now instead of PDF?

✅ 🔚 Final Output / Result Part

These are the responses and results a user or admin will see on the website after performing actions:

🔷 1. Home Page Result

Python

When a user opens the site:

```
@app.route(‘/’)
Def home():
   Return render_template(‘home.html’, user=current_user)
```

📌 Result: Shows a simple homepage UI (home.html), optionally with user info if logged in.

🔷 2. Admin Page Access Result

Python

```
@app.route(‘/admin’)
@login_required
Def admin_page():
   If not any(role.name == ‘Admin’ for role in current_user.roles):
      Return “Access Denied: Admins Only”
   Return “Welcome Admin!”
```

📌 Result:

If the user is not an admin → “Access Denied: Admins Only”

If the user is an admin → “Welcome Admin!”

🔷 3. Database & Role Creation Result

Python

```
@app.route(‘/create’)
Def create_roles_users():
   # ...creates roles and users
   Return “Database and roles created successfully!”
```

📌 Result:

✅ Outputs this message after creating initial data:

“Database and roles created successfully!”

🖼️ Visual Outputs

If you add HTML in templates/home.html, you can display:

User name

User roles

Logout/login links

Navigation buttons based on role