Creating a website to connect landlords and tenants involves multiple steps, from setting up the development environment to deploying the website. Here’s a detailed guide using Python, Flask, and other essential tools:

**1. Setting Up the Development Environment**

1. **Install Python:**
   * Ensure Python is installed on your system. Download it from the [official Python website](https://www.python.org/).
2. **Set Up a Virtual Environment:**

bash

Copy code

python -m venv myenv

source myenv/bin/activate # On Windows, use `myenv\Scripts\activate`

1. **Install Necessary Packages:**

bash

Copy code

pip install Flask Flask-SQLAlchemy Flask-Migrate Flask-Login

**2. Create the Flask Application**

1. **Project Structure:**

arduino

Copy code

landlord\_tenant\_app/

├── app/

│ ├── \_\_init\_\_.py

│ ├── models.py

│ ├── routes.py

│ ├── templates/

│ │ ├── layout.html

│ │ ├── index.html

│ │ ├── register.html

│ │ ├── login.html

│ │ ├── properties.html

│ └── static/

├── migrations/

├── config.py

├── run.py

└── requirements.txt

1. **Initialize Flask App (app/init.py):**

python

Copy code

from flask import Flask

from flask\_sqlalchemy import SQLAlchemy

from flask\_migrate import Migrate

from flask\_login import LoginManager

app = Flask(\_\_name\_\_)

app.config['SECRET\_KEY'] = 'your\_secret\_key'

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///site.db'

db = SQLAlchemy(app)

migrate = Migrate(app, db)

login\_manager = LoginManager(app)

login\_manager.login\_view = 'login'

from app import routes, models

1. **Database Models (app/models.py):**

python

Copy code

from datetime import datetime

from app import db, login\_manager

from flask\_login import UserMixin

@login\_manager.user\_loader

def load\_user(user\_id):

return User.query.get(int(user\_id))

class User(db.Model, UserMixin):

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

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

email = db.Column(db.String(120), unique=True, nullable=False)

password = db.Column(db.String(60), nullable=False)

properties = db.relationship('Property', backref='owner', lazy=True)

class Property(db.Model):

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

title = db.Column(db.String(100), nullable=False)

description = db.Column(db.Text, nullable=False)

price = db.Column(db.Float, nullable=False)

date\_posted = db.Column(db.DateTime, nullable=False, default=datetime.utcnow)

owner\_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)

1. **Routes (app/routes.py):**

python

Copy code

from flask import render\_template, url\_for, flash, redirect

from app import app, db

from app.models import User, Property

from flask\_login import login\_user, current\_user, logout\_user, login\_required

from app.forms import RegistrationForm, LoginForm, PropertyForm

@app.route("/")

@app.route("/home")

def home():

properties = Property.query.all()

return render\_template('index.html', properties=properties)

@app.route("/register", methods=['GET', 'POST'])

def register():

if current\_user.is\_authenticated:

return redirect(url\_for('home'))

form = RegistrationForm()

if form.validate\_on\_submit():

user = User(username=form.username.data, email=form.email.data, password=form.password.data)

db.session.add(user)

db.session.commit()

flash('Your account has been created!', 'success')

return redirect(url\_for('login'))

return render\_template('register.html', form=form)

@app.route("/login", methods=['GET', 'POST'])

def login():

if current\_user.is\_authenticated:

return redirect(url\_for('home'))

form = LoginForm()

if form.validate\_on\_submit():

user = User.query.filter\_by(email=form.email.data).first()

if user and user.password == form.password.data:

login\_user(user)

return redirect(url\_for('home'))

else:

flash('Login Unsuccessful. Please check email and password', 'danger')

return render\_template('login.html', form=form)

@app.route("/logout")

def logout():

logout\_user()

return redirect(url\_for('home'))

@app.route("/property/new", methods=['GET', 'POST'])

@login\_required

def new\_property():

form = PropertyForm()

if form.validate\_on\_submit():

property = Property(title=form.title.data, description=form.description.data, price=form.price.data, owner=current\_user)

db.session.add(property)

db.session.commit()

flash('Your property has been listed!', 'success')

return redirect(url\_for('home'))

return render\_template('property.html', form=form)

1. **Forms (app/forms.py):**

python

Copy code

from flask\_wtf import FlaskForm

from wtforms import StringField, PasswordField, SubmitField, TextAreaField, FloatField

from wtforms.validators import DataRequired, Length, Email, EqualTo

class RegistrationForm(FlaskForm):

username = StringField('Username', validators=[DataRequired(), Length(min=2, max=20)])

email = StringField('Email', validators=[DataRequired(), Email()])

password = PasswordField('Password', validators=[DataRequired()])

confirm\_password = PasswordField('Confirm Password', validators=[DataRequired(), EqualTo('password')])

submit = SubmitField('Sign Up')

class LoginForm(FlaskForm):

email = StringField('Email', validators=[DataRequired(), Email()])

password = PasswordField('Password', validators=[DataRequired()])

submit = SubmitField('Login')

class PropertyForm(FlaskForm):

title = StringField('Title', validators=[DataRequired()])

description = TextAreaField('Description', validators=[DataRequired()])

price = FloatField('Price', validators=[DataRequired()])

submit = SubmitField('List Property')

**3. Templates (HTML)**

1. **Base Template (app/templates/layout.html):**

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>{% block title %}Landlord-Tenant App{% endblock %}</title>

<link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">

</head>

<body>

<nav>

<a href="{{ url\_for('home') }}">Home</a>

{% if current\_user.is\_authenticated %}

<a href="{{ url\_for('logout') }}">Logout</a>

<a href="{{ url\_for('new\_property') }}">List Property</a>

{% else %}

<a href="{{ url\_for('login') }}">Login</a>

<a href="{{ url\_for('register') }}">Register</a>

{% endif %}

</nav>

<div class="container">

{% with messages = get\_flashed\_messages(with\_categories=true) %}

{% if messages %}

{% for category, message in messages %}

<div class="flash {{ category }}">{{ message }}</div>

{% endfor %}

{% endif %}

{% endwith %}

{% block content %}{% endblock %}

</div>

</body>

</html>

1. **Home Page Template (app/templates/index.html):**

html

Copy code

{% extends "layout.html" %}

{% block content %}

<h1>Available Properties</h1>

{% for property in properties %}

<div class="property">

<h2>{{ property.title }}</h2>

<p>{{ property.description }}</p>

<p>Price: ${{ property.price }}</p>

<p>Owner: {{ property.owner.username }}</p>

</div>

{% endfor %}

{% endblock %}

1. **Register Template (app/templates/register.html):**

html

Copy code

{% extends "layout.html" %}

{% block content %}

<h1>Register</h1>

<form method="POST">

{{ form.hidden\_tag() }}

<div>{{ form.username.label }} {{ form.username }}</div>

<div>{{ form.email.label }} {{ form.email }}</div>

<div>{{ form.password.label }} {{ form.password }}</div>

<div>{{ form.confirm\_password.label }} {{ form.confirm\_password }}</div>

<div>{{ form.submit }}</div>

</form>

{% endblock %}

1. **Login Template (app/templates/login.html):**

html

Copy code

{% extends "layout.html" %}

{% block content %}

<h1>Login</h1>

<form method="POST">

{{ form.hidden\_tag() }}

<div>{{ form.email.label }} {{ form.email }}</div>

<div>{{ form.password.label }} {{ form.password }}</div>

<div>{{ form.submit }}</

4o