File structure

```
HomeProperty/
│── .sass-cache/
│── scss/
│── static/
│── templates/
│── app.py
│── accounts.db
│── listings.db
│── gov_data.db
│── resale_data.csv
│── license.txt
```

Skeleton Code of app.py (placed here in case not allowed to upload .py files)

```
from flask import Flask, render_template, request, redirect, url_for, session, flash, jsonify
import sqlite3
import re
import json
import random
from twilio.rest import Client  # For SMS OTP



#summary:
"""
using Flask framework for backend
using html/css and javascript for the frontend
using twilio SMS API for OTP verification for account handling
each route is a separate webpage to be displayed, they can link to each other
if needed
"""
app = Flask(__name__)
app.secret_key = "your_secret_key_here"

# Database initialization
def init_db():
    # Creates SQLite databases for user accounts and property listings
    # Users table stores seller/agent accounts
    # Listings table stores property details and bids

# OTP Functions
def generate_otp():
    # Generates 6-digit OTP
```

```
def send_otp_sms(phone_number):
   # Sends OTP via Twilio SMS

# User Authentication
@app.route('/login', methods=['GET', 'POST'])
def login():
   # Handles user login with username/password
   # Sets session variables and redirects to appropriate dashboard

@app.route('/create_account', methods=['GET', 'POST'])
def create_account():
   # Handles account creation for sellers/agents
   # Validates OTP, password strength, and agent IDs via government API

# Seller Functions
@app.route('/seller_dashboard')
def seller_dashboard():
   # Seller homepage showing their listings

@app.route('/list-property', methods=['GET', 'POST'])
def list_property():
   # Form for sellers to list new properties
   # Shows similar historical prices

@app.route('/view_your_property')
def view_your_property():
   # Shows all properties listed by current seller

# Agent Functions
@app.route('/agent_dashboard')
def agent_dashboard():
   # Agent homepage showing available listings

@app.route('/view_listed_property')
def view_listed_property():
   # Shows all active listings agents can bid on

@app.route('/submit_bid', methods=['POST'])
def submit_bid():
   # Handles agent bids on properties
   # Validates against max commission rate

# Property Management
@app.route('/get_similar_prices')
```

```
def get_similar_prices():
    # Returns similar historical property prices for reference

@app.route('/accept_bid', methods=['POST'])
def accept_bid():
    # Lets sellers accept an agent's bid
    # Marks property as sold

# Helper Functions
def find_similar_past_prices():
    # Queries CSV data to find comparable property sales
    # Used for price suggestions

if __name__ == '__main__':
    init_db()
    app.run(debug=True)
```