Business Problem

“Can we build a simple web application that stores house details in a database and uses machine learning to predict the next selling price of a house?”

This application must allow users to:

✓ Enter house details

(Size, BHK, Age, Last Sold Price, etc.)

✓ Store the details safely in MySQL

(Like a mini real-world property management system)

✓ Perform CRUD operations

- Add
- View
- Update
- Delete

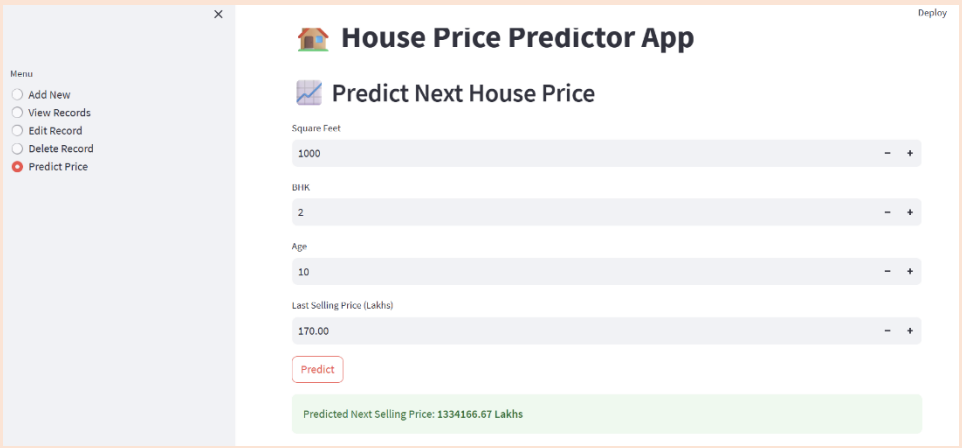
✓ Train a machine learning model

using historical house prices

✓ Predict the *next* price

based on patterns learned from old prices

✓ And all this should run as a clean Web App using Streamlit



Step 0 —

“Open PowerShell (Windows) or Terminal (macOS/Linux).

We’ll create a fresh project folder. I’ll give exact commands — copy/paste them.”

Student prompt (run all lines together):

create project folder and enter it

mkdir house_price_predictor

cd house_price_predictor

create venv (force Python 3.11)

py -3.11 -m venv venv

.\venv\Scripts\Activate.ps1

Make sure you see (venv) in the prompt — that means the virtual environment is active.”
Check / expected: Prompt shows (venv) PS C:\...\house_price_predictor>.

```
PS C:\Users\askpr\house_price_predictor> py -3.11 -m venv venv
PS C:\Users\askpr\house_price_predictor> .\venv\Scripts\Activate.ps1
(venv) PS C:\Users\askpr\house_price_predictor>
```

Step 1 — Install required packages

@"

streamlit==1.26.0

SQLAlchemy==2.0.20

pandas==2.1.2

scikit-learn==1.3.2

joblib==1.3.2

mysql-connector-python==8.1.0

python-dotenv==1.0.0

"@ > requirements.txt

(venv) PS C:\Users\askpr\house_price_predictor> pip install -r requirements.txt

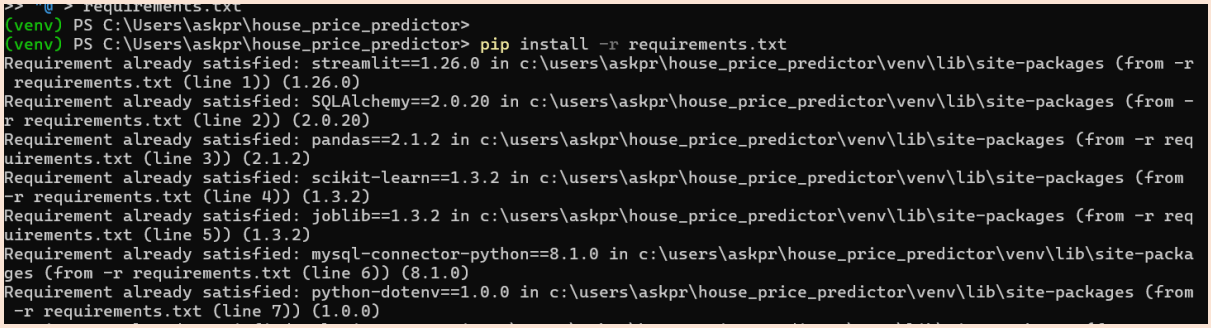
“This installs the exact packages we’ll use. If pip fails, check venv activation or Python version.”

Chatgpt may recommend specific version to install, and follow instruction.

Check / expected: pip finishes with no errors and packages listed by pip list

Output will be installation begin, this will take some time, an important step, wait till command prompt you get .

Output sample , different for different computer system



Anything in red color comes, that indicates, that is giving error during installation.

It is possible that , during all list of requirements file , 3-4 installed and 1 gave error. If error , copy all text including what ever came , give to chatgpt and ask to resolve error. Warning need reading of solution with patience, try one by one option, and finally it will compelte. For me it took 15-20 minutes, for you it may be less or more time as well

Step 2 — open mysql workbench , (you must know your mysql password to run it) Create MySQL database & user

“Open MySQL Workbench . Run these SQL statements to create the database and a user. Use these exact names so our .env matches the code.”

CREATE DATABASE house_price_db;

CREATE USER 'house_user'@'localhost' IDENTIFIED BY 'House@1234';

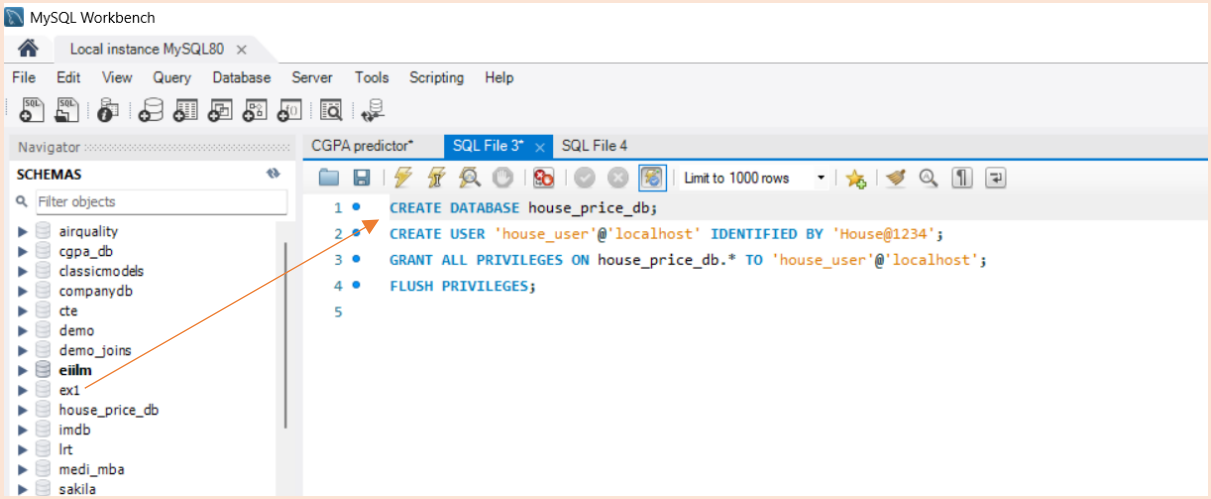
GRANT ALL PRIVILEGES ON house_price_db.* TO 'house_user'@'localhost';

FLUSH PRIVILEGES;

Select one line and run or select all line and run. If you run more than once , it will throw error.

You can check in explorer database , by refresh , data base created or not

Output



Step 3 — Create .env file (so SQLAlchemy can read connection)

Now PowerShell command to write a UTF-8 .env file containing the DB URL and model path.

Write following command in prompt (PowerShell inside project folder):

$db = "DATABASE_URL=mysql+mysqlconnector://house_user:House%401234@localhost:3306/house_price_db"

$m = "MODEL_PATH=house_model.joblib"

Set-Content -Path .env -Value "$db`n$m" -Encoding UTF8

Get-Content .env

What to say: "You should see the DATABASE_URL (with %40 encoded) in the printed .env text."
Check / expected output :

DATABASE_URL=mysql+mysqlconnector://house_user:House%401234@localhost:3306/house_price_db

MODEL_PATH=house_model.joblib

```
(venv) PS C:\Users\askpr\house_price_predictor> $db = "DATABASE_URL=mysql+mysqlconnector://house_user:House%401234@local
host:3306/house_price_db"
(venv) PS C:\Users\askpr\house_price_predictor> $m = "MODEL_PATH=house_model.joblib"
(venv) PS C:\Users\askpr\house_price_predictor> Set-Content -Path .env -Value "$db`n$m" -Encoding UTF8
(venv) PS C:\Users\askpr\house_price_predictor> Get-Content .env
DATABASE_URL=mysql+mysqlconnector://house_user:House%401234@localhost:3306/house_price_db
MODEL_PATH=house_model.joblib
(venv) PS C:\Users\askpr\house_price_predictor>
```

Here I got error due to UTF8 related, and I paste command and error as text to chatgpt to solve error, and it resolved after 2-3 back forth command and prompt.

Step 4 — Create ORM schema (models.py)

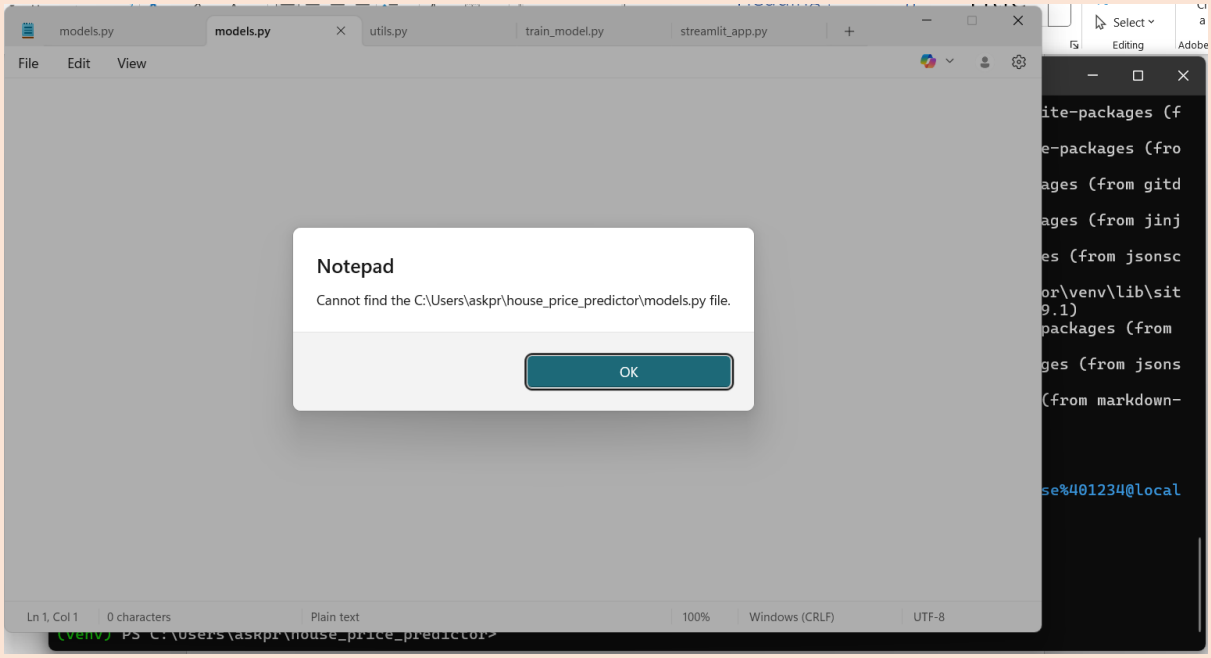
"Now we create the table schema. This script reads .env and creates the MySQL table. Run it after confirming .env exists."

Step 4.1

(venv) PS C:\Users\askpr\house_price_predictor> notepad models.py

```
(venv) PS C:\Users\askpr\house_price_predictor> notepad models.py
(venv) PS C:\Users\askpr\house_price_predictor>
```

This will open notepad. With file name models.py , copy and paste code I am giving below. Save and close file.



Step 4.2 copy paste code

Paste exactly this content into models.py, save and close:

```
import os
from dotenv import load_dotenv
from sqlalchemy import create_engine, Column, Integer, Float, String
from sqlalchemy.orm import declarative_base, sessionmaker

load_dotenv()
```

```
DATABASE_URL = os.getenv("DATABASE_URL")
engine = create_engine(DATABASE_URL)
Base = declarative_base()

def init_db():
    Base.metadata.create_all(engine)

class House(Base):
    __tablename__ = "houses"
    id = Column(Integer, primary_key=True, autoincrement=True)
    location = Column(String(100))
    size_sqft = Column(Integer)
    bedrooms = Column(Integer)
    age_years = Column(Float)
    last_price = Column(Float)
    next_price = Column(Float, nullable=True)

SessionLocal = sessionmaker(bind=engine)

if __name__ == "__main__":
    print("📌 Creating tables in MySQL...")
    Base.metadata.create_all(engine)
    print("✅ Tables created successfully!")
```

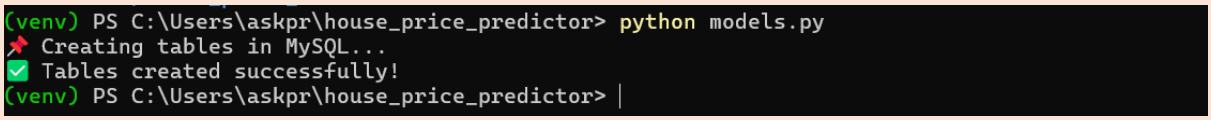
Step 4.3 Run following code on prompt.

python models.py

like below

(venv) PS C:\Users\askpr\house_price_predictor> python models.py

This will give output like



@@@@@@@@@@@@@@@@@@@@

Above step was giving error or not created table due to one of various following reason.

Now next all steps are you will be running first time

@@@@@@@@@@@@@@@@@@@@@@

But If giving error here is what you will do .

Preflight checklist (run **before** Step 4.3)

1) Make sure venv is active

.\venv\Scripts\Activate.ps1

2) Confirm .env exists and is exactly the right file (show lines)

Write-Output ">>> SHOW .env <<<"

Get-Content .env -Encoding UTF8 | % { "LINE: '$_'" }

3) Quick Python check that python-dotenv reads it

python -c "import os; from dotenv import load_dotenv; load_dotenv(); print('DATABASE_URL ->', repr(os.getenv('DATABASE_URL')))"

Expected outputs:

- (venv) visible in prompt after step 1.
- Step 2 prints exactly two lines: DATABASE_URL=... and MODEL_PATH=... (no extra characters).
- Step 3 prints the DB URL string (not None).

If all good — proceed to Step 4.3.

Super-short Troubleshooting Card

If python models.py fails with Expected string or URL object, got None or DATABASE_URL -> None:

1. Ensure venv active:

.\venv\Scripts\Activate.ps1

2. Re-write .env safely (this forces UTF-8 no-BOM):

$db = "DATABASE_URL=mysql+mysqlconnector://house_user:House%401234@localhost:3306/house_price_db"

$m = "MODEL_PATH=house_model.joblib"

Set-Content -Path .env -Value "$db`n$m" -Encoding UTF8

Get-Content .env -Encoding UTF8

3. Quick Python verification:

python -c "import os; from dotenv import load_dotenv; load_dotenv(); print('DATABASE_URL ->', repr(os.getenv('DATABASE_URL')))"

If that prints the DB URL (not None) — re-run:

python models.py

If step 2/3 still give None, run this temporary session fix (lets you continue immediately while you debug):

$env:DATABASE_URL = "mysql+mysqlconnector://house_user:House%401234@localhost:3306/house_price_db"

python models.py

(This sets the env var only for the current shell; it proves the script works when the env variable exists.)

One-line explanation

“If Python says it can’t find DATABASE_URL even though .env looks right, it’s almost always an encoding issue: Notepad sometimes saves files with a Byte Order Mark (BOM) or UTF-16 encoding that python-dotenv won't parse. Rewriting .env with Set-Content -Encoding UTF8 fixes it. If that still fails, we set the env var in the shell temporarily and continue the lesson — then debug offline.”

Short troubleshooting FAQ you can put in the doc

ModuleNotFoundError: No module named 'dotenv'
Fix: activate venv and run pip install python-dotenv.

- **Expected string or URL object, got None**
  Fix: .env not read. Recreate .env with Set-Content .env -Encoding UTF8 and test with the Python one-liner above.
- **pandas install/build fails**
  Fix: ensure students use Python 3.11; py -3.11 -m venv venv then pip install -r requirements.txt.
- **Streamlit port already in use**
  Run streamlit run streamlit_app.py --server.port 8502.

Let us continue now our project

=-

▶ STEP 4 — Insert Sample House Records

1) Open Python REPL

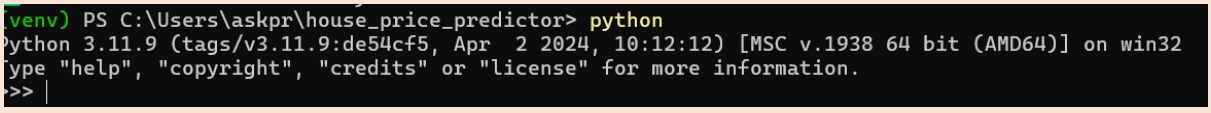
In PowerShell (venv active):

python

You will get:

>>>

Like this



2) Copy-paste this inside the Python REPL

from utils import add_house

add_house("House A", 1200, 3, 2, 10, 550000)

add_house("House B", 900, 2, 1, 15, 320000)

add_house("House C", 1500, 4, 3, 5, 780000)

add_house("House D", 800, 2, 1, 20, 280000)

add_house("House E", 1800, 4, 3, 2, 950000)

print("Inserted 5 sample rows!")

3) Exit from python prompt and back to working directory

>>> exit()

```
Type "help", "copyright", "credits" or "license" for
>>> from utils import add_house
>>>
>>> add_house("House A", 1200, 3, 2, 10, 550000)
1
>>> add_house("House B", 900, 2, 1, 15, 320000)
2
>>> add_house("House C", 1500, 4, 3, 5, 780000)
3
>>> add_house("House D", 800, 2, 1, 20, 280000)
4
>>> add_house("House E", 1800, 4, 3, 2, 950000)
5
>>>
>>> print("Inserted 5 sample rows!")
Inserted 5 sample rows!
>>> exit
Use exit() or Ctrl-Z plus Return to exit
>>> exit()
(venv) PS C:\Users\askpr\house_price_predictor>
```

Step 5 : Train the ML model (train_model.py)

"Now run the training script which reads labeled rows and saves a model."

```
venv) PS C:\Users\askpr\house_price_predictor> notepad train_model.py
venv) PS C:\Users\askpr\house_price_predictor>
```

This will open notepad and copy and paste following code and save and close file

```
import os
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
from joblib import dump
from dotenv import load_dotenv
from sqlalchemy import select
from models import House, SessionLocal

load_dotenv()
MODEL_PATH = os.getenv("MODEL_PATH", "house_model.joblib")

def load_training_data():
    with SessionLocal() as session:
        rows = session.execute(select(House)).scalars().all()
    data = []
    for r in rows:
        if r.next_price is not None:
            data.append({
                "size_sqft": r.size_sqft,
                "bedrooms": r.bedrooms,
                "age_years": r.age_years,
                "last_price": r.last_price,
                "next_price": r.next_price
```

```
            })
    if len(data) == 0:
        raise ValueError("No training data found. Insert some rows with next_price filled (labelled data).")
    df = pd.DataFrame(data)
    X = df[["size_sqft","bedrooms","age_years","last_price"]]
    y = df["next_price"]
    return X, y


def train_and_save_model():
    X, y = load_training_data()
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    model = LinearRegression()
    model.fit(X_train, y_train)
    preds = model.predict(X_test)
    mse = mean_squared_error(y_test, preds)
    mae = mean_absolute_error(y_test, preds)
    print(f"Model trained successfully! Test MSE: {mse:.4f}, MAE: {mae:.4f}")
    dump(model, MODEL_PATH)
    print(f"Model saved at: {MODEL_PATH}")


if __name__ == "__main__":
    train_and_save_model()
```

now run python program , which you have saved and closed. Like this

(venv) PS C:\Users\askpr\house_price_predictor> python train_model.py

```
(venv) PS C:\Users\askpr\house_price_predictor> python train_model.py
Model trained successfully! Test MSE: 0.0000, MAE: 0.0000
Model saved at: house_model.joblib
(venv) PS C:\Users\askpr\house_price_predictor>
```

You may get different numbers in MSE and MAE and that is perfectly okay. It is just because of limited data . as you give more data, it will change in future.

Next step : step 6

1) Create the Streamlit file

In PowerShell (project folder, venv active): write following command

notepad streamlit_app.py

see following output

```
(venv) PS C:\Users\askpr\house_price_predictor> notepad streamlit_app.py
(venv) PS C:\Users\askpr\house_price_predictor>
```

Paste the exact code below into streamlit_app.py, save and close.

2) Full streamlit_app.py (copy-paste exactly)

```
# streamlit_app.py
import os
import streamlit as st
import pandas as pd
import joblib
from dotenv import load_dotenv

from models import init_db, House, SessionLocal
from utils import add_house, get_all_houses, update_house, delete_house

# --- Setup ---
load_dotenv()
MODEL_PATH = os.getenv("MODEL_PATH", "house_model.joblib")
init_db()

st.set_page_config(page_title="House Price Predictor", layout="wide", page_icon="🏠")

# local image path (your Notepad encoding screenshot)
ENCODING_IMAGE = r"/mnt/data/4fff8675-e089-4047-970e-f5428f6da01c.png"
```

```
# --- helper functions ---
def load_model():
    if os.path.exists(MODEL_PATH):
        try:
            model = joblib.load(MODEL_PATH)
            return model
        except Exception as e:
            st.error(f"Failed to load model: {e}")
            return None
    else:
        return None

def houses_to_df(houses):
    rows = []
    for r in houses:
        rows.append({
            "id": r.id,
            "location": r.location,
            "size_sqft": r.size_sqft,
            "bedrooms": r.bedrooms,
            "age_years": r.age_years,
            "last_price": r.last_price,
            "next_price": r.next_price
        })
    return pd.DataFrame(rows)

# --- Layout ---
st.title("🏠 House Price Predictor — Demo")
st.markdown(
    "Add house data, predict next price with the trained model, and save predictions back to the DB."
)
```

```
col1, col2 = st.columns([1, 2])


with col1:
    st.subheader("Add new house (or Predict)")
    with st.form("add_form", clear_on_submit=True):
        location = st.text_input("Location", value="MG Road")
        size_sqft = st.number_input("Size (sqft)", min_value=200, max_value=10000, value=1000)
        bedrooms = st.number_input("Bedrooms", min_value=0, max_value=10, value=2)
        age_years = st.number_input("Age (years)", min_value=0.0, max_value=100.0, value=5.0,
step=0.5)
        last_price = st.number_input("Last price (INR)", min_value=0.0, value=500000.0, step=1000.0)
        submitted = st.form_submit_button("Add to DB")
        if submitted:
            new_id = add_house(location, size_sqft, int(bedrooms), float(age_years), float(last_price),
None)
            st.success(f"Inserted id: {new_id}")


    st.markdown("---")
    st.subheader("Predict next price (without saving)")
    model = load_model()
    if model is None:
        st.warning("Model not found. Train model (train_model.py) to enable predictions.")
    else:
        loc2 = st.text_input("Location (predict)", value="MG Road", key="loc2")
        sqft2 = st.number_input("Size (sqft) (predict)", min_value=200, max_value=10000, value=1000,
key="sqft2")
        beds2 = st.number_input("Bedrooms (predict)", min_value=0, max_value=10, value=2,
key="beds2")
        age2 = st.number_input("Age (years) (predict)", min_value=0.0, max_value=100.0, value=5.0,
step=0.5, key="age2")
        lastp2 = st.number_input("Last price (INR) (predict)", min_value=0.0, value=500000.0,
step=1000.0, key="lastp2")
```

```
        if st.button("Predict"):
            X = [[int(sqft2), int(beds2), float(age2), float(lastp2)]]
            try:
                pred = model.predict(X)[0]
                st.metric("Predicted next_price", f"{pred:,.0f}")
                if st.button("Save predicted value to DB"):
                    added_id = add_house(loc2, int(sqft2), int(beds2), float(age2), float(lastp2), float(pred))
                    st.success(f"Saved predicted value as new record id {added_id}")
            except Exception as e:
                st.error(f"Prediction failed: {e}")

with col2:
    st.subheader("Dataset — View / Edit / Delete")
    houses = get_all_houses()
    df = houses_to_df(houses)
    if df.empty:
        st.info("No records found. Add some houses first.")
    else:
        st.dataframe(df.sort_values("id", ascending=False), use_container_width=True)

    st.markdown("---")
    st.subheader("Edit / Delete an entry")
    ids = [int(r.id) for r in houses] if houses else []
    if ids:
        sel_id = st.selectbox("Choose ID to edit/delete", options=ids)
        sel_rec = None
        for r in houses:
            if r.id == sel_id:
                sel_rec = r
                break
        if sel_rec:
```

```
        with st.form("edit_form"):
            loc_e = st.text_input("Location", value=sel_rec.location)
            sqft_e = st.number_input("Size (sqft)", min_value=200, max_value=10000, value=sel_rec.size_sqft)
            beds_e = st.number_input("Bedrooms", min_value=0, max_value=10, value=sel_rec.bedrooms)
            age_e = st.number_input("Age (years)", min_value=0.0, max_value=100.0, value=sel_rec.age_years, step=0.5)
            lastp_e = st.number_input("Last price", min_value=0.0, value=sel_rec.last_price, step=1000.0)
            nextp_e = st.number_input("Next price", min_value=0.0, value=sel_rec.next_price or 0.0, step=1000.0)
            save_btn = st.form_submit_button("Save changes")
            del_btn = st.button("Delete record")
            if save_btn:
                update_house(sel_rec.id, loc_e, int(sqft_e), int(beds_e), float(age_e), float(lastp_e), float(nextp_e))
                st.success("Record updated — refresh the page to see changes.")
            if del_btn:
                delete_house(sel_rec.id)
                st.success("Record deleted — refresh the page to see changes.")
    else:
        st.info("No IDs available to edit/delete.")

st.markdown("---")
st.sidebar.image(ENCODING_IMAGE, caption="Notepad encodings — watch for BOM", use_column_width=True)
st.sidebar.markdown("**Quick actions**")
if st.sidebar.button("Reload model"):
    st.experimental_rerun()

st.sidebar.markdown("**Tips**\n- If model not loaded, run `python train_model.py`.\n- If env issues, ensure `.env` is UTF-8 (no BOM).")
```

3) Run the app

In PowerShell (project folder, venv active):

streamlit run streamlit_app.py

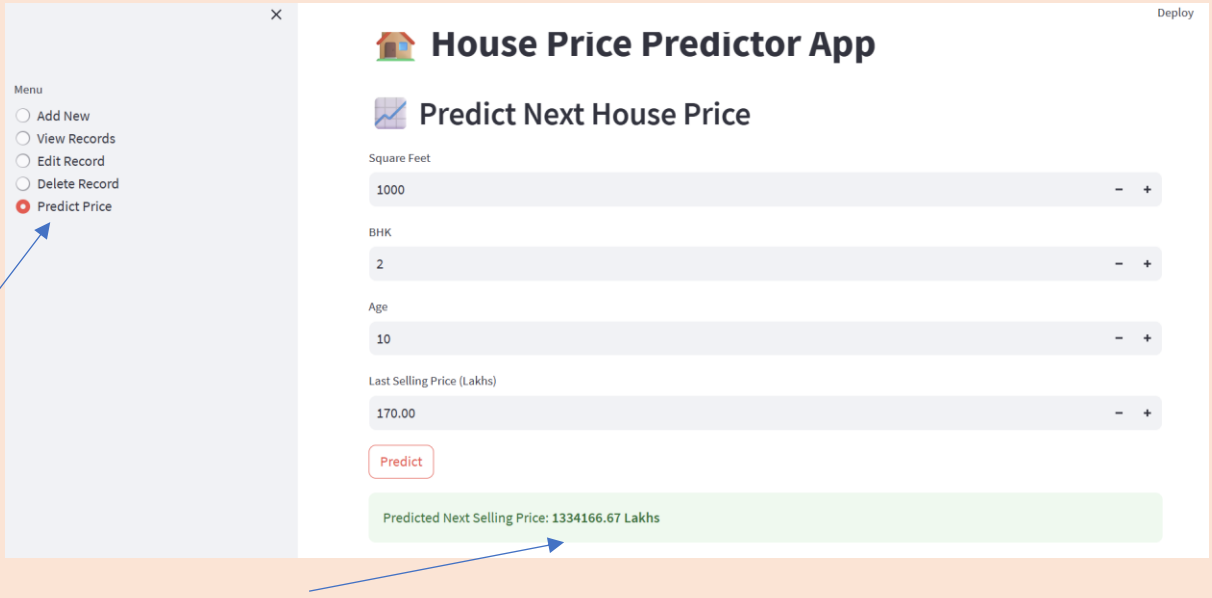
just like this

```
(venv) PS C:\Users\askpr\house_price_predictor> notepad streamlit_app.py
(venv) PS C:\Users\askpr\house_price_predictor> streamlit run streamlit_app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.29.32:8501
```

Congratulations !!! your app is ready



In case any of above functionality gives your run time error, one can use copy paste and send to chatgpt , such that revised code of streamlit python file you can get.

To close, one can use ctrl +c on windowshell prompt , will stop and then close browser.

Prepared by Pragnesh shah , Freelancer trainer, Academician, Data science consultant

Now here is some hidden prompt I used first to begin with

Prompt I used

want to develop python program ,which takes input from user , store in mysql with crud operations and then create deployment version to show . input is hours studied, assginments compelte out of 10 , attendecne in terms of percentage, last semester result CGPA(out of 10) . this will store all values i wanted. then i apply machine learning to predict for next semester CGPA . what will be plan and what information is do you requeire , let me know.

As it give solution, all in once, I said as follows

Prompt I used

no ,i am getting confuse with multiple instructions. give one at a time and check it is performed successfully ? then go for second.

This way I got one instruction at a time and I was pasting output to confirm , step executed correctly or not.

Somewhere I got following choice and I choose option A

✅ **Option A (Recommended non-Docker): Use local MySQL + local Python (like we are doing now)**

✅ **Option B: Use a hosted MySQL (Cloud) + local Python**

✅ **Option C: Pre-create a Portable MySQL bundle**