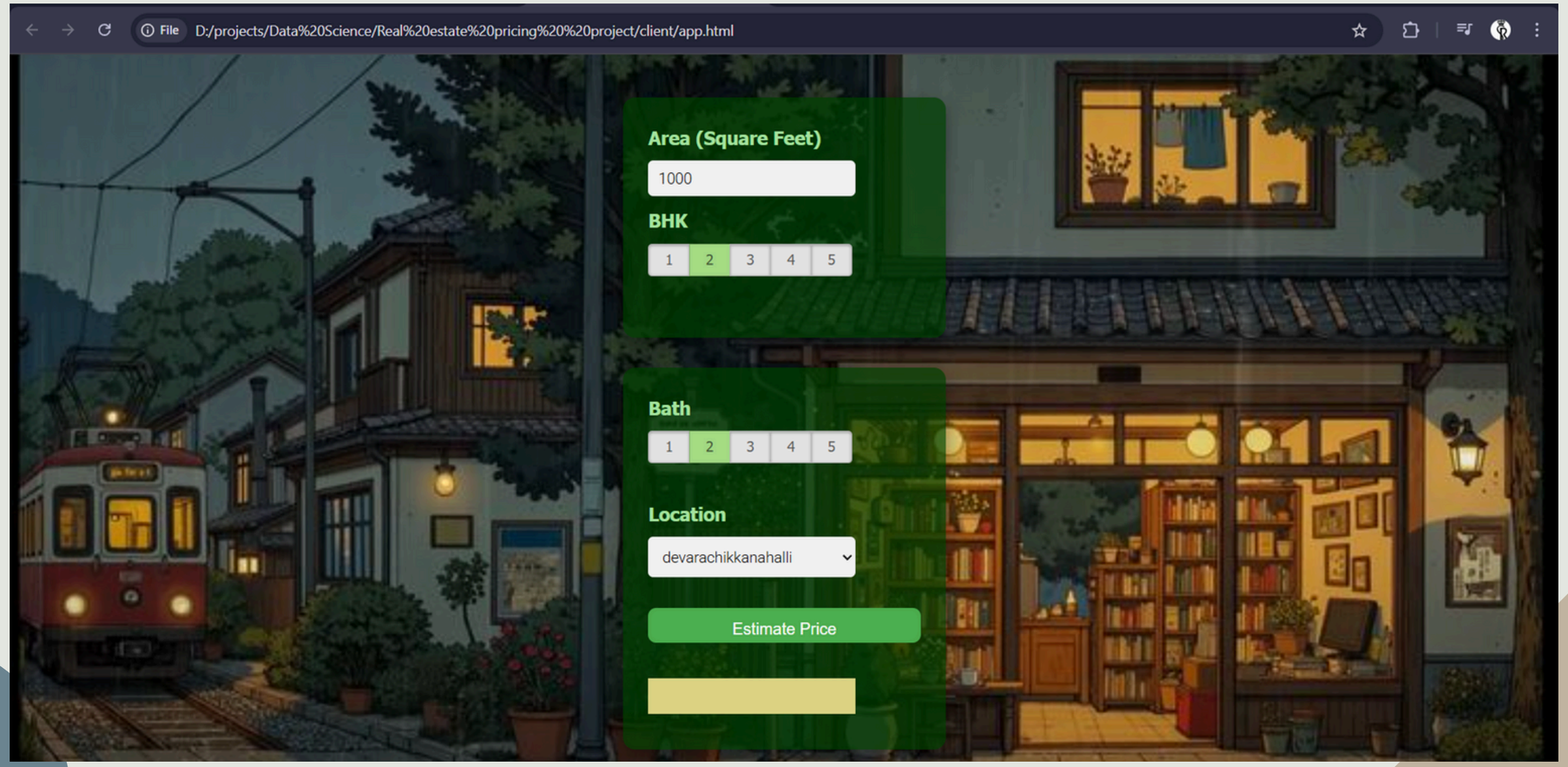
PROJECT: BANGALORE HOME PRICE PREDICTION

Making property pricing simple and data-driven

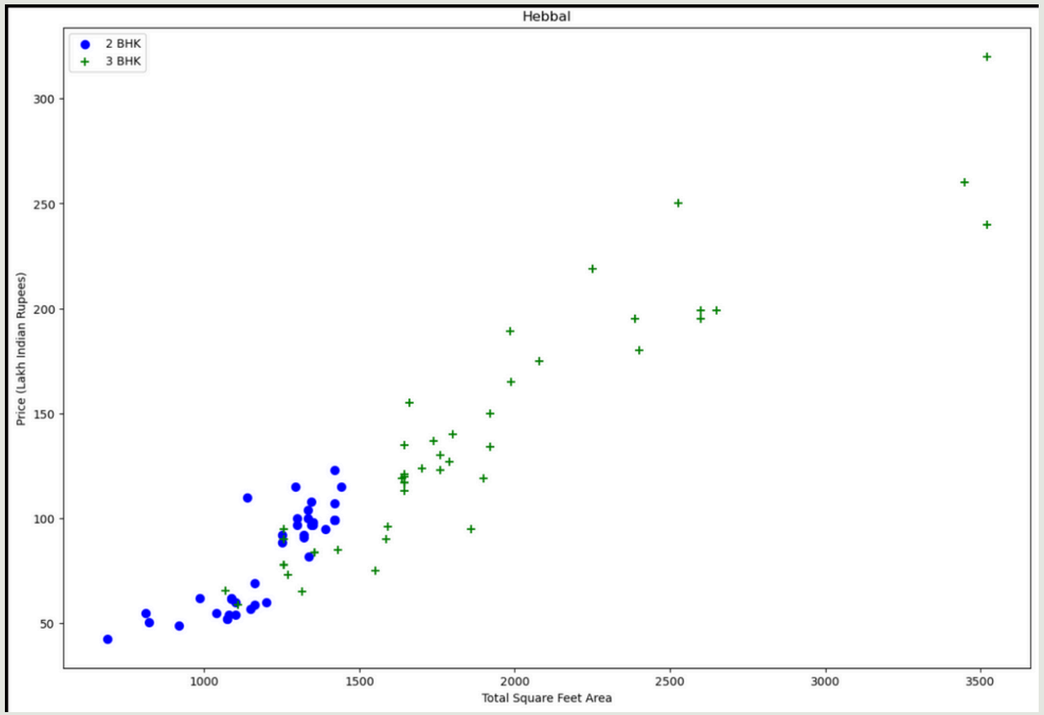


WHY IT’S USEFUL

Many buyers overpay or miss good opportunities because property pricing information is scattered and unclear.

Even slight differences in area, BHK, bathrooms, or location can significantly impact the price.

The tool simplifies this by providing instant, accurate price estimates, saving time and reducing uncertainty for buyers.



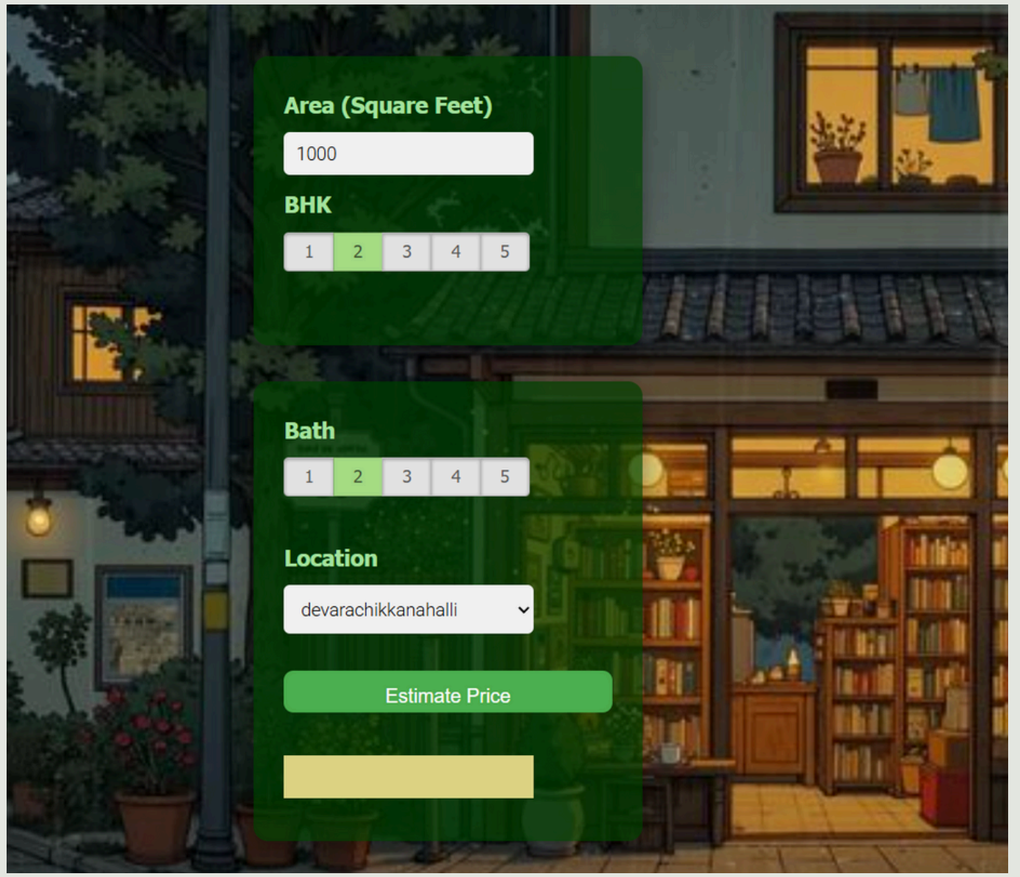
USER INPUTS

The system is designed to be simple and intuitive.

Users provide just four inputs:

- Area in square feet
- Number of bedrooms (BHK)
- Number of bathrooms
- Location

With these details, the model predicts a realistic home price almost instantly.



MODEL & APPROACH

I chose Random Forest because it handles non-linear relationships between features like area, BHK, and location.
It is also robust against outliers and variations in the dataset, providing consistent predictions.
The Flask API connects the frontend with the model, taking inputs and returning predictions instantly.
The jQuery frontend ensures results are displayed dynamically without reloading the page.

TECHNICAL STACK

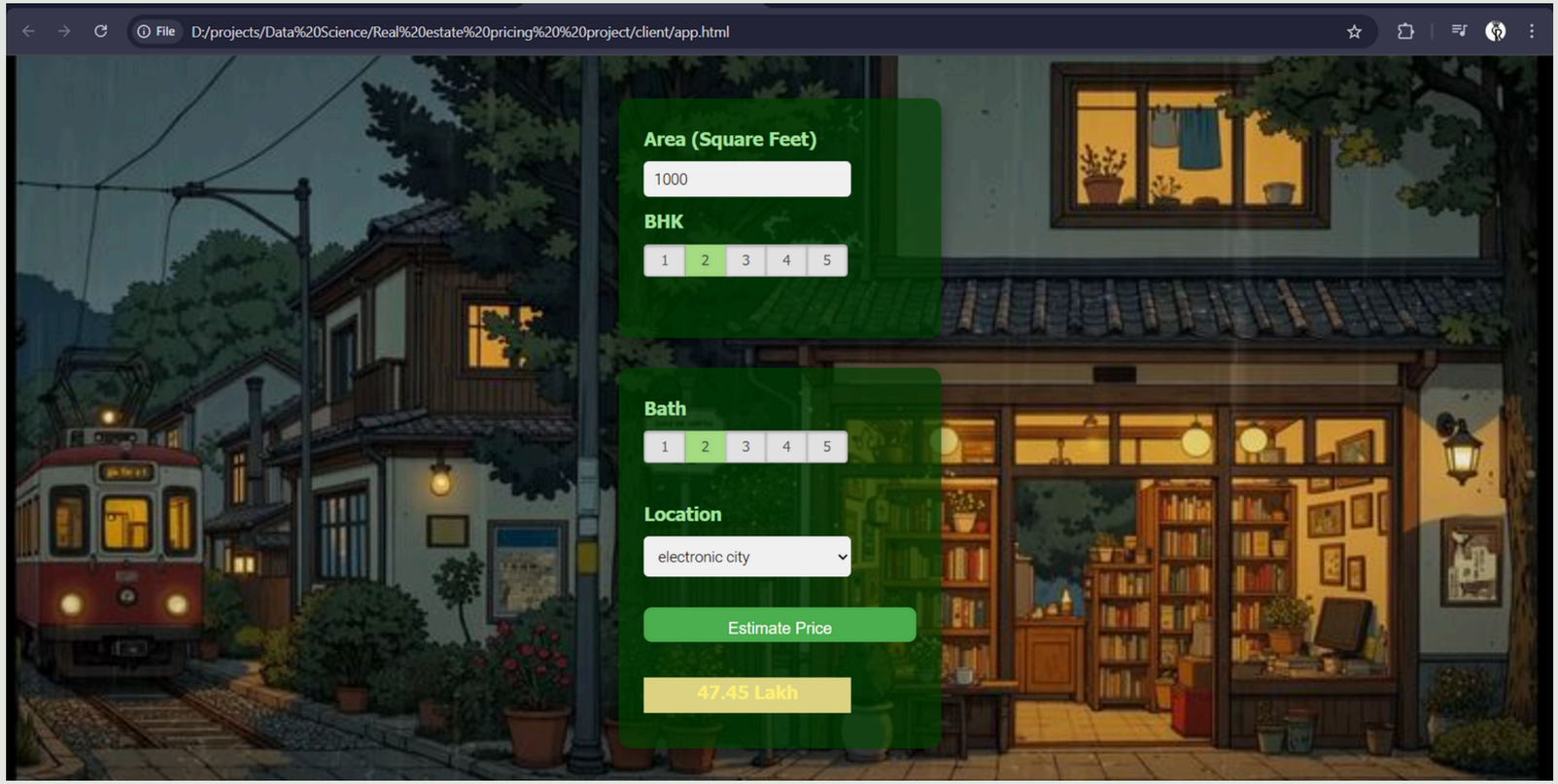
The project uses a combination of web development and machine learning:

- Frontend: HTML, CSS, JavaScript, jQuery (for dynamic updates)
- Backend: Python Flask (handles API requests and serves predictions)
- Modeling: Random Forest (robust, handles non-linear relationships)
- Libraries: Pandas, NumPy (for data preprocessing and calculations)

This stack ensures the system is responsive and easy to interact with.

HOW IT WORKS

- User enters property details on the interface.
- Frontend sends the data to the Flask backend API.
- The backend runs the inputs through the trained Random Forest model.
- The predicted price is sent back and displayed on the page instantly.



LEARNINGS & INSIGHTS

This project taught me:

- How to integrate machine learning with web applications
- The importance of data preprocessing and choosing the right model
- How to design an API that connects user inputs to ML predictions
- Building interactive and user-friendly interfaces
- Future improvements could include more cities, advanced visualizations, or a more polished UI.