HOUSE PRICE PREDICTION

B.Harshasri (1602-23-737-085)
G.Suhasini Reddy (1602-23-737-312)

Abstract

TheReal Estate Price Prediction Websiteis designed to provide users with a reliable tool to estimate property prices based on key inputs such as location, square footage, and the number of bedrooms and bathrooms. The project integrates machine learning techniques with web development to create a seamless and user-friendly platform.

The project is structured into three main components:

Model Building:
A machine learning model, trained using a dataset of Bangalore home prices from Kaggle, leverages linear to ensure accuracy.

Backend Development:
A Python Flask server connects the machine learning model with the web interface, enabling efficient data exchange and predictions.

Frontend Development:
An interactive user interface developed with HTML, CSS, and JavaScript allows users to input property details and view the predicted prices.

This project not only addresses the growing demand for data-driven tools in the real estate sector but also demonstrates the practical application of data science concepts in real-world scenarios.

INTRODUCTION

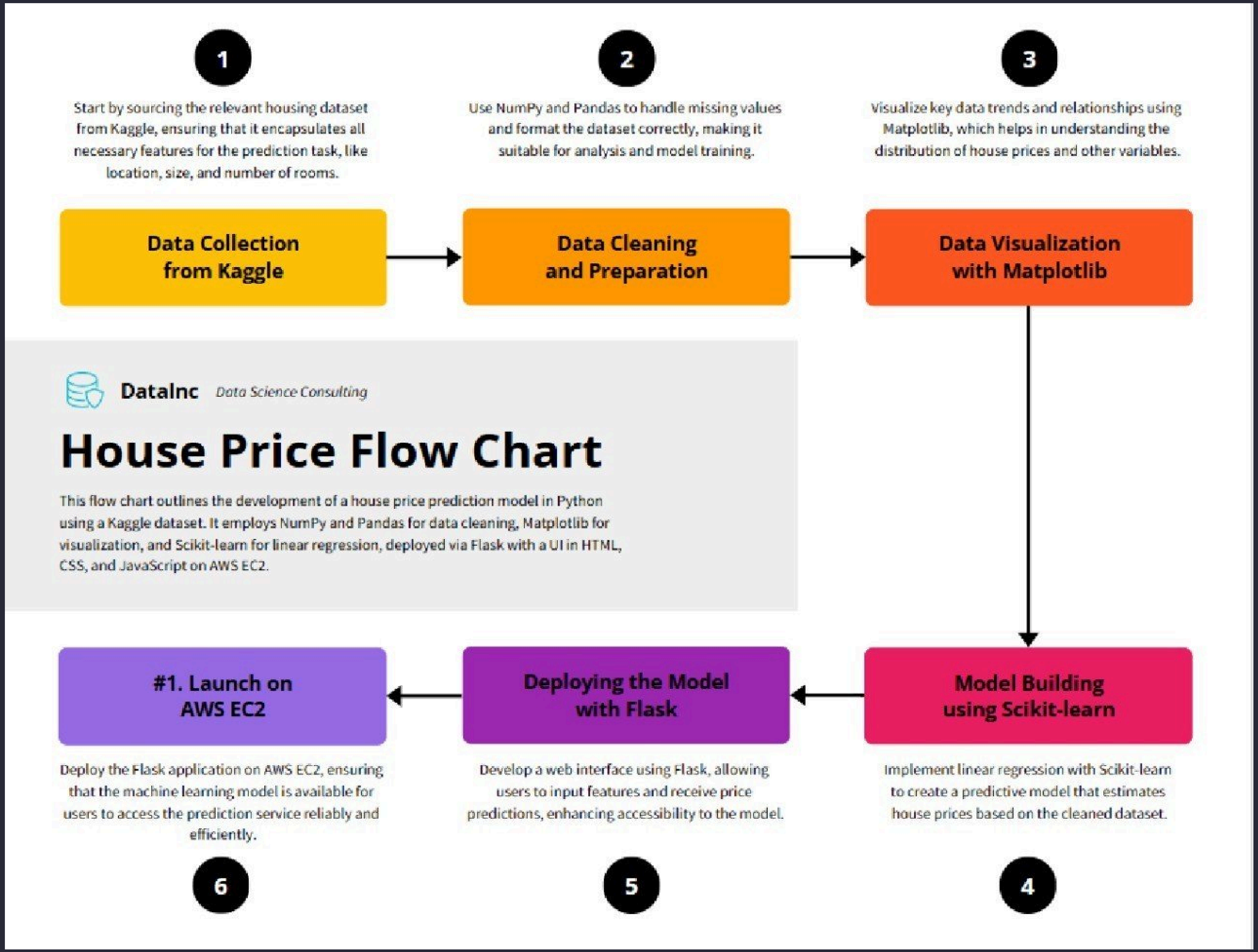
a. Information About the Project Domain

The real estate sector is a cornerstone of economic growth, and accurate property valuation is critical for buyers, sellers, and investors. Traditional property valuation methods often rely on subjective analysis, leading to inconsistent and unreliable pricing. This project addresses this challenge by using data science to provide a more scientific and accurate approach to price prediction.The domain of real estate prediction is inherently data-driven, with factors like location, property size, and amenities influencing prices. Machine learning techniques enable the extraction of patterns and trends from historical data, leading to more precise predictions. This project, specifically focused on Bangalore's real estate market, demonstrates the application of predictive analytics to address these challenges.

b. Features Specific to the Project

The project is designed with the following prioritized features:

1. Accurate Price Prediction
2. User-Friendly Interface
3. Backend Integration with Flask
4. Data-Driven Approach
5. Scalability and Extendibility


1
Start by sourcing the relevant housing dataset from Kaggle, ensuring that it encapsulates all necessary features for the prediction task, like location, size, and number of rooms.
2
Use NumPy and Pandas to handle missing values and format the dataset correctly, making it suitable for analysis and model training.
3
Visualize key data trends and relationships using Matplotlib, which helps in understanding the distribution of house prices and other variables.
Data Collection from Kaggle
Data Cleaning and Preparation
Data Visualization with Matplotlib
DataInc Data Science Consulting
House Price Flow Chart
This flow chart outlines the development of a house price prediction model in Python using a Kaggle dataset. It employs NumPy and Pandas for data cleaning, Matplotlib for visualization, and Scikit-learn for linear regression, deployed via Flask with a UI in HTML, CSS, and JavaScript on AWS EC2.
#1. Launch on AWS EC2
Deploying the Model with Flask
Model Building using Scikit-learn
Deploy the Flask application on AWS EC2, ensuring that the machine learning model is available for users to access the prediction service reliably and efficiently.
Develop a web interface using Flask, allowing users to input features and receive price predictions, enhancing accessibility to the model.
Implement linear regression with Scikit-learn to create a predictive model that estimates house prices based on the cleaned dataset.
6
5
4

Technology

Software Requirements

a. Software Requirements

The project relies on the following software tools and technologies:

1. Python

Primary language for data preprocessing, machine learning model development, andbackend integration.

2. Libraries and Frameworkso Numpy and Pandas:
For data cleaning, preprocessing, and manipulation.o Matplotlib:For
data visualization and exploratory data analysis.o Sklearn (Scikit-learn):For building the machine learning model,
performing and implementing linear regression. Flask:To create a lightweight web server for
handling HTTP requests and serving predictions.

3. Integrated Development Environments (IDEs): Jupyter Notebook:For
exploratory data analysis and model building.o Visual Studio CodeFor Flask server development and integrating
the backend with the frontend.

4. Frontend Technologies:o HTML:To structure the web pages.o CSS:For
styling the user interface.8o JavaScript:To
handle dynamic interactions and send requests to the Flask server.

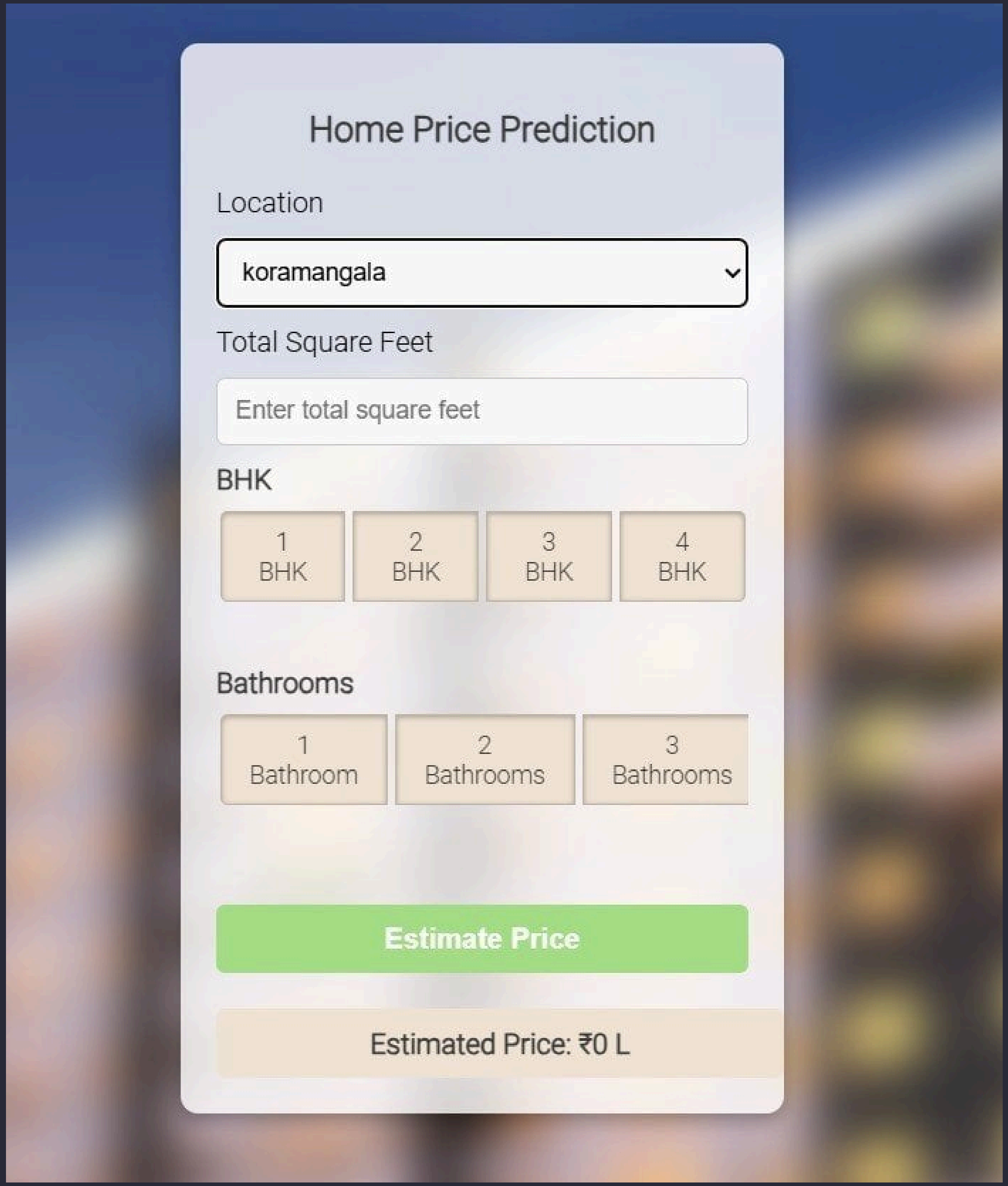
5. Other Tools:o
Kaggle:To source the Bangalore home prices dataset.o Git/GitHub:o For
version control and repository management.

Windows PowerShell

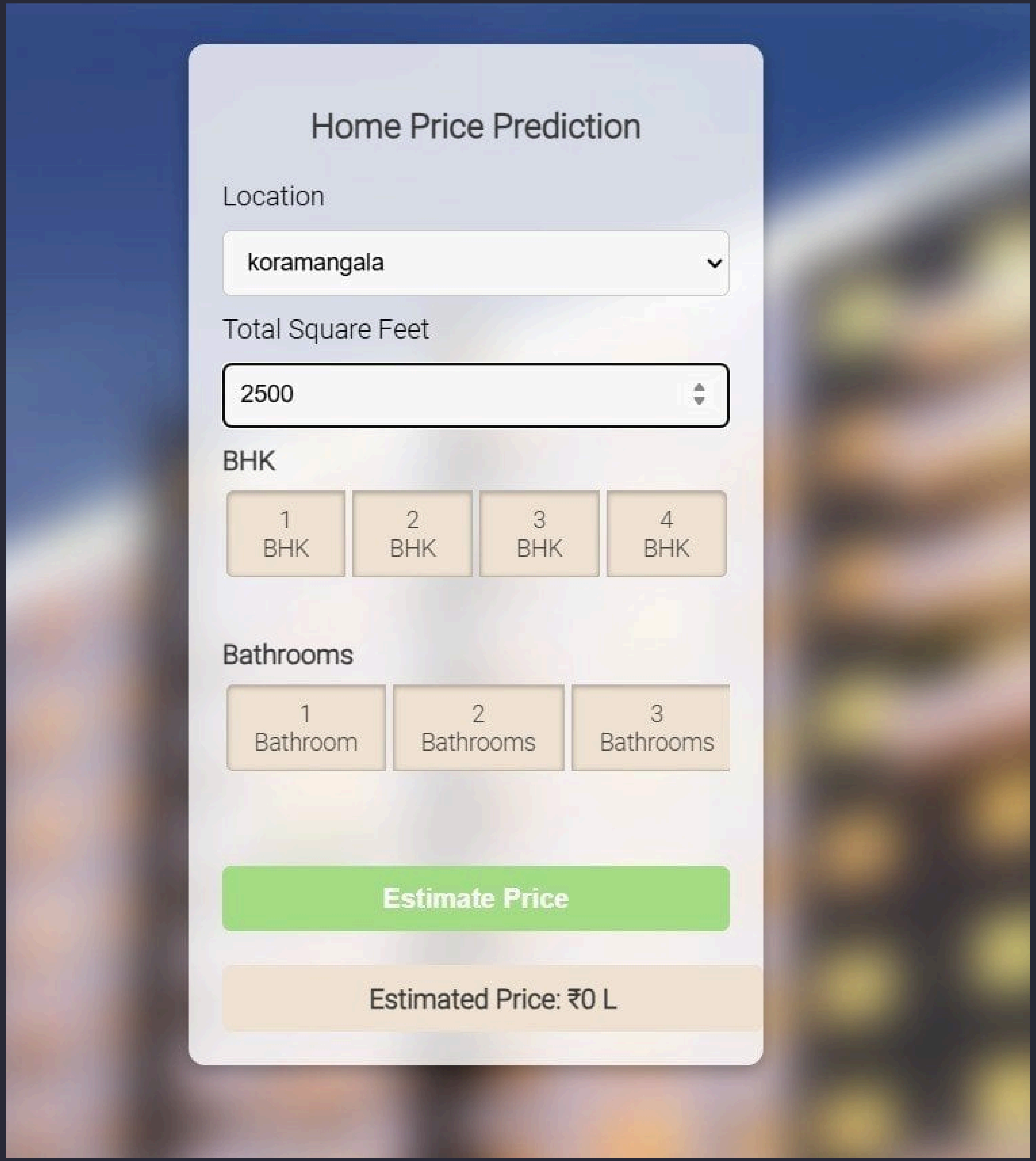
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\harsh\OneDrive\Desktop\miniproject\server> python app.py
```

Windows PowerShell

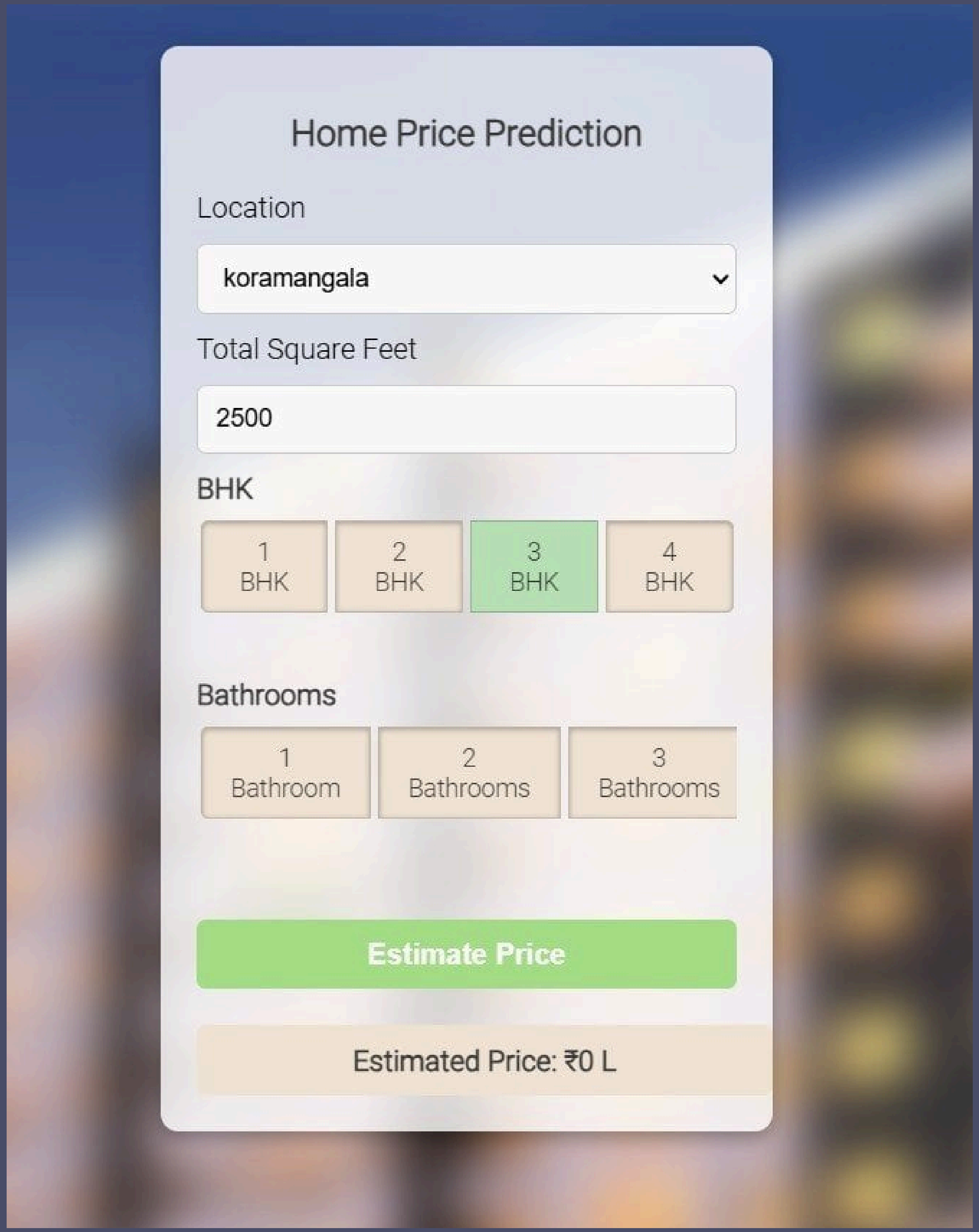
```
  eps=np.finfo(np.float).eps, random_state=None,
C:\Users\harsh\AppData\Local\Programs\Python\Python37-32\lib\site-packages\sklearn\linear_model\randomized_l1.py:580: De
precationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itse
lf. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float
64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecation
s
  eps=4 * np.finfo(np.float).eps, n_jobs=None,
loading saved artifacts...done
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
Starting Python Flask Server For Home Price Prediction...
loading saved artifacts...start
C:\Users\harsh\AppData\Local\Programs\Python\Python37-32\lib\site-packages\sklearn\linear_model\least_angle.py:35: Depre
cationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself.
 Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64`
 here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecation
s
  eps=np.finfo(np.float).eps,
C:\Users\harsh\AppData\Local\Programs\Python\Python37-32\lib\site-packages\sklearn\linear_model\least_angle.py:597: Depr
ecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, use `float` by itself
. Doing this will not modify any behavior and is safe. If you specifically wanted the numpy scalar type, use `np.float64
```



enter location

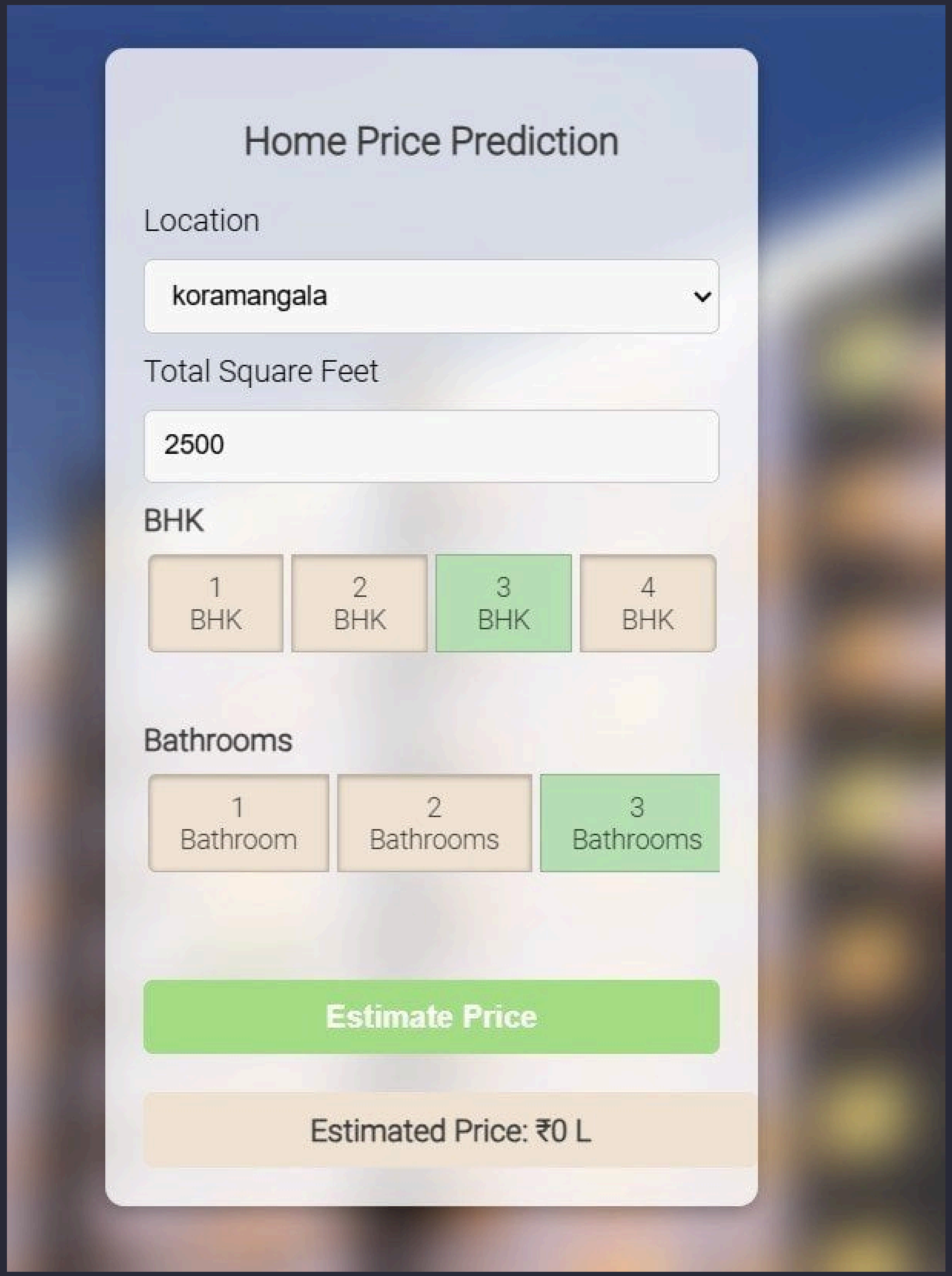


enter square fetts

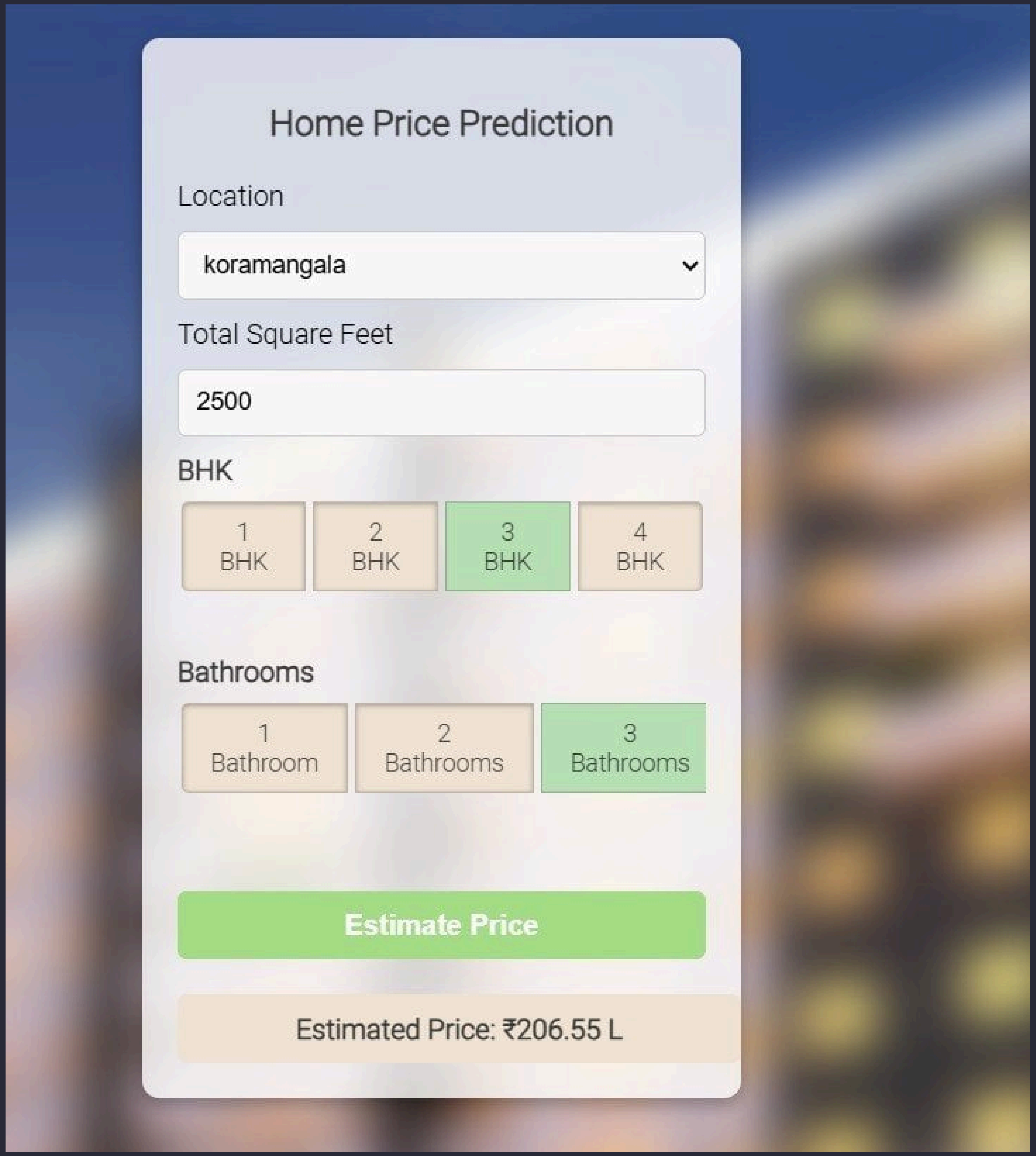


Home Price Prediction
Location
koramangala
Total Square Feet
2500
BHK
1 BHK
2 BHK
3 BHK
4 BHK
Bathrooms
1 Bathroom
2 Bathrooms
3 Bathrooms
Estimate Price
Estimated Price: ₹0 L

enter bhk



enter no of bathrooms


Home Price Prediction
Location
koramangala
Total Square Feet
2500
BHK
1 BHK
2 BHK
3 BHK
4 BHK
Bathrooms
1 Bathroom
2 Bathrooms
3 Bathrooms
Estimate Price
Estimated Price: ₹206.55 L

Conclusion

The house price prediction mini-project has been an insightful and valuable learning experience. By combining various aspects of programming, data science, machine learning, and web development, it has provided a comprehensive understanding of how to build an end-to-end predictive system. The project successfully demonstrated the full workflow, starting from data preprocessing and model training, to deploying the model via a Flask backend, and integrating it with a frontend for real-time predictions. The use of linear regression, data cleaning, feature engineering, and model evaluation through techniques helped in achieving a robust and efficient prediction model. Additionally, building a dynamic web interface in HTML, CSS, and JavaScript provided hands-on experience in developing full-stack applications. Through this project, I have gained valuable knowledge in machine learning, data manipulation, and full-stack development. The integration of machine learning with web technologies is crucial in the current landscape of software applications, and this project has helped me bridge the gap between theoretical knowledge and practical implementation.

Future Work

Although the current implementation serves its purpose effectively, there are several areas where the project can be enhanced:

1.Model Improvement

oAdvanced Machine Learning Models:

The current project uses linear regression. Exploring more sophisticated models like Random Forest, Gradient Boosting, or XGBoost could potentially improve prediction accuracy

oEnsemble Methods:

Combining multiple models through techniques like bagging and boosting could also help enhance the overall model performance.

2.Data Expansion

oCurrently, the model is based on a limited dataset for Bangalore. Expanding the dataset to include more cities or regions would allow for more generalized predictions, improving the versatility of the model.

3.Real-Time Data Integration oTo make the predictions even more accurate, integrating real-time data such as market trends, interest rates, or economic factors could provide more context for the pricing predictions.
4.User Interface Enhancements 30
oThe current web interface can be made more interactive by implementing real-time validation of user inputs, and providing more detailed feedback (e.g., range of predicted prices based on varying input features). oThe website could be optimized for mobile users to provide a better experience across devices.
5.Deployment and Scaling
oDeploying the project on a cloud platform (such as AWS, Azure, or Heroku) could make the application more accessible. This would also allow for scalability, supporting more users and requests without performance degradation. oImplementing Docker could help containerize the application, ensuring consistent deployment across different environments.
6.Additional Features oUser Account Management: Allow users to save their predicted results or access historical data. oData Visualization: Adding graphs and charts for visual representation of predicted house prices based on location, area, etc., would make the results more insightful for users.

Thank you!!