**1.Introduction**

We propose to make a Real estate price predictor web app for Major Indian Cities.

* 1. **Purpose**

The product aims to help the average consumer have a tool they can use to check the price of a property in a given demographic area.

* 1. **Scope**

The product currently aims to be a web app that consumers can visit and check prices, however, over time as the ML model changes and the dataset grows, it could be converted into an extension that people can use on their own website or an API.

* 1. **Definitions, Abbreviations**

ML-Machine Learning

**1.4 References**

1.4.1. R. S. Pressman, “Software Engineering: A Practitioners Approach”, McGraw Hill, 2009.

1.4.2 N. N. Ghosalkar and S. N. Dhage, "Real Estate Value Prediction Using Linear Regression” 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-5, doi: 10.1109/ICCUBEA.2018.8697639.

1.4.3 J. C. Soares Silva and A. T. de Almeida Filho, "Performing hierarchical Bayesian regression to assess the best districts for building new residential real estate developments," 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2020, pp. 2411-2416, doi: 10.1109/SMC42975.2020.9282906.

**2.General Description**

**2.1 Product Perspective**

The product is supposed to be an open source, under the GNU general Public License. It is a web-based system implementing an ML model. It will be hosted on a cloud platform to increase its accessibility and limit host-side hardware requirements.

**2.2 Product Functions**

**2.3 User Characteristics**

We assume the user is truthful and does not misuse the product, they are somebody that is trying to sell their property or are trying to buy a property.

**2.4 General Constraints**

We aim to achieve an accuracy of 95% in our predictions at the current stage, this implies the possibility of some edge cases where the prediction is not accurate.

**2.5 Assumptions and Limitations**

We assume that the user does not input dummy data to ruin the ML model, to counter this, we propose maintenance every 15 days, however in the long term, we will need to deploy a Deep Learning system instead.

**3.Specific Functions**

**3.1 City Selector**

**3.1.1 Introduction**

This is the first page of the product that users use to choose their city

**3.1.2 Input**

String input through website

**3.1.3 Processing**

The processing is handled on the back end by the ML model.

**3.1.4 Outputs**

Specific City Website is shown.

**3.2 Simple Predictor (Specific City Website)**

**3.2.1 Introduction**

This is the second page of the product that users can use to get a better estimate by specifying more information

**3.2.2 Input**

We plan on having a form to accept inputs, The parameters/features (from an ML point of view) will take research. We can assume it takes values like BHK and Locality.

**3.2.3 Processing**

The processing is handled on the back end by the ML model.

**3.2.4 Outputs**

The output is an estimate of what the price of the property will be.

**3.3 Submit your own data**

**3.3.1 Introduction**

This is the last page of the product that users can use to add their own data to the model

**3.3.2 Input**

We plan on having a form to accept inputs, The parameters/features (from an ML point of view) will take research. We can assume it takes values like BHK, amenities, schools nearby, Locality etc.

**3.3.3 Processing**

The data is stored in the backend

**3.3.4 Outputs**

The ML model will improve, A more accurate price prediction will be seen

**3.2 External Interface Requirements**

**3.2.1 User Interface**

We plan on designing a web application.

**3.2.2 Software Interface**

The user must have a browser.

3.3 Performance Requirement  
 3.3.1 On the cloud: 8 GB Ram and 2 GB GPU  
 3.3.2 On the user end: A web browser with strong internet connection

**3.4 Design Constraint**

**3.4.1 Standard Compliance**

On the user end: A web browser with strong internet connection

**3.4.2 Hardware Compliance:**

On the cloud: 8 GB Ram and 2 GB GPU

**3.5 Attributes**

**3.5.1 Security**

Handled by Cloud platform

**3.5.2 Maintainability**

Will currently be manually updated every 15 days or whenever a large chunk of unadded data is seen

**3.6 Other Requirements**

3.6.1 Cloud platform subscription   
 3.6.2 Database  
 3.6.3 Compute Resources like VMs or Kubernetes Container

**3.7 Testing**  
 Model is trained and tested using cross-val score