

PRODUCT REQUIREMENTS DOCUMENT

Version 1.0 | Confidential

# NaviEstimate

# House Price Prediction System

*NaviEstimate — An ML-Driven Real Estate Valuation Platform*

<b>Prepared By</b> Senior PM / ML Architect / UX Lead	<b>Date</b> June 2025
<b>Status</b> Draft — For Review	<b>Domain</b> PropTech / Machine Learning

# 1. Executive Summary

*The Navi Mumbai real estate market is growing at double-digit annual rates yet operates with chronic price opacity. This document defines the product, technical, and design requirements for NaviEstimate — a machine learning-powered house price prediction platform that brings data-driven transparency to residential property transactions.*

## 1.1 The Problem

Navi Mumbai's residential real estate market — spanning nodes like Kharghar, Vashi, Nerul, Panvel, Airoli, and Belapur — processes thousands of property transactions annually. However, price discovery remains opaque, inconsistent, and heavily broker-dependent. Three structural failures define the current state:

- Asymmetric information: Sellers and buyers access radically different price signals. Brokers arbitrage this gap, often inflating or deflating quotes by 15–30%.
- Absence of objective benchmarks: No publicly accessible, locality-specific, feature-adjusted valuation model exists for Navi Mumbai.
- Static and lagging listings: Platforms like MagicBricks and 99acres display listed prices, not fair market values. These listings are often stale, unverified, and stripped of context.

## 1.2 Target Users

User Segment	Core Need
Home Buyers	Verify fair price before making offer; avoid overpaying
Sellers	Set competitive, data-backed asking price; reduce time on market
Real Estate Brokers	Generate instant valuation reports for client pitches
Property Investors	Assess ROI potential; compare sub-market performance
Academic Researchers	Access clean, structured real estate data for studies
Housing Finance Companies	Automated collateral valuation for loan underwriting

## 1.3 Product Value Proposition

- Instant, ML-powered price predictions with confidence intervals — not just listing averages.
- Granular locality intelligence covering 30+ Navi Mumbai nodes.
- Transparent feature attribution: users understand why a property is valued at a given price.
- Downloadable PDF valuation reports for formal use cases (loan applications, negotiations).
- A neutral, non-broker-influenced platform. No listings. No commissions. No noise.

## 1.4 Expected Impact

Metric	Baseline (Current)	Target (6 Months)
--------	--------------------	-------------------

Price Prediction Accuracy ( $R^2$ )	—	$\geq 0.88$
Mean Absolute Error	—	$\leq ₹4.5$ Lakhs
Prediction Latency	—	$< 800$ ms
Monthly Active Users (MAU)	—	5,000+
Broker Adoption Rate	—	20% within region

## 2. Market & Domain Analysis

### 2.1 Navi Mumbai Real Estate Ecosystem

Navi Mumbai, developed by CIDCO (City and Industrial Development Corporation), is one of India's most planned urban agglomerations. The residential market is stratified across multiple transit-oriented nodes, each with distinct price points driven by infrastructure maturity, connectivity, and commercial density.

Key residential nodes and their market positioning:

Node	Market Positioning & Key Drivers
Kharghar	Mid-premium; IT corridor proximity, NMSEZ, golf course, Central Park
Vashi	Established prime node; APMC market, Seawoods, proximity to Mumbai
Nerul	Premium residential; DY Patil, Belapur CBD, railway connectivity
Panvel	High-growth corridor; NAINA, New Airport, NH-48 connectivity
Airoli	Emerging IT hub; TCS, Infosys campuses, Thane connectivity
Belapur	CBD of Navi Mumbai; CIDCO HQ, court complexes, corporate density
Ulwe	Affordable + growth play; NMIA (new airport) land appreciation zone
Taloja / Kamothe	Affordable housing; MIDC industrial proximity

### 2.2 Key Price Influencing Factors

Feature analysis from domain research identifies the following as statistically significant price drivers:

#### Infrastructure & Connectivity

- Distance to nearest railway station (Harbour Line, Trans-Harbour Line)
- Proximity to Navi Mumbai Metro Line 1 (NMML1) stations — currently operational Belapur to Pendhar
- Access to NH-48, Sion-Panvel Expressway, and upcoming NAINA corridors
- Airport proximity premium (NMIA expected 2025-26 — Ulwe, Panvel corridor)

#### Social Infrastructure

- Distance to top schools: DAV, Apeejay, Ryan International, CBSE/ICSE affiliates
- Proximity to hospitals: MGM Hospital, Apollo, DY Patil Medical
- Retail and entertainment: Seawoods Grand Central, Inorbit, Nexus

#### Employment & Commercial Density

- Distance to IT parks: Mindspace (Airoli), DLF (Kharghar), Turbhe MIDC
- Proximity to Belapur CBD, APMC market, JNPT port corridor

#### Property Intrinsic Features

- BHK configuration, carpet area (sq ft), floor level, total floors

- Property age, construction quality tier (CIDCO/private developer)
- Amenities: swimming pool, gym, club house, 24/7 security, parking
- Furnishing status: bare shell / semi-furnished / fully furnished

## 2.3 Competitive Landscape

Platform	Strength	Weakness	Differentiation Gap
MagicBricks	Large inventory, brand trust	Listed prices only, no ML valuation	No prediction engine
99acres	Search depth, agent network	No confidence intervals, broker-skewed	No Navi Mumbai granularity
Housing.com	UI quality, map view	Macro-level estimates only	No feature-level transparency
PropTiger	Research reports	Not real-time, subscription gated	Not accessible to public
NoBroker	No-commission model	No price intelligence layer	No ML valuation

## 2.4 Market Opportunity

- Navi Mumbai real estate transactions: ~₹8,200 Cr annually (CREDAI estimates).
- PropTech SaaS penetration in Tier-2 Indian metro segments is < 5% — massive whitespace.
- No purpose-built, open-access ML valuation tool exists for Navi Mumbai specifically.
- Growing demand from housing finance companies for automated valuation models (AVMs).
- Academic and government bodies (CIDCO, NMMC) lack structured price benchmarking tools.

### 3. Product Vision & Goals

*Vision: To become the definitive, ML-powered real estate valuation intelligence layer for Navi Mumbai — trusted by buyers, relied upon by brokers, and cited by researchers.*

#### 3.1 MVP Goals (0–3 Months)

1. Deploy a trained regression model achieving  $R^2 \geq 0.85$  on held-out Navi Mumbai property data.
2. Build a responsive web interface with prediction form, results dashboard, and basic locality comparison.
3. Serve predictions with  $< 1$  second API response time under normal load.
4. Support 20+ localities with locality-specific feature encoding.
5. Enable PDF export of prediction reports.
6. Launch admin panel for dataset management and model re-training triggers.

#### 3.2 Long-Term Vision (6–18 Months)

- Expand to all 111 NAINA planning nodes with hyperlocal models.
- Launch rental prediction model with yield calculator.
- Build price forecasting using time-series models (Prophet, LSTM).
- Integrate AI chatbot for natural language property queries.
- Mobile app (React Native) with AR-based property inspection overlays.
- API monetization for housing finance companies and institutional clients.
- Partnership with CIDCO and NMMC for verified transaction data feeds.

#### 3.3 Measurable KPIs

KPI	Target & Measurement Method
Mean Absolute Error (MAE)	$\leq ₹4.5$ Lakhs on test set (sklearn metrics)
Root Mean Square Error (RMSE)	$\leq ₹7.2$ Lakhs (penalizes large errors)
$R^2$ Score	$\geq 0.88$ (% variance explained by model)
Prediction Latency (P95)	$< 800$ ms end-to-end API response
Model Retraining Cycle	Monthly, triggered by data drift detection
User Engagement (DAU/MAU)	$\geq 25\%$ ratio within 3 months of launch
Prediction Report Downloads	$\geq 500$ /month within 90 days of launch
Model Confidence Interval Width	Avg $\pm 8\%$ of predicted value
Feature Coverage Rate	$\geq 95\%$ of input submissions fully parsed

## 4. Functional Requirements

### 4.1 User Input Interface

The prediction form must collect the following structured inputs. All fields validated client-side before API submission.

Input Field	Type / Constraints
Locality / Node	Dropdown — 30+ Navi Mumbai localities (Kharghar, Vashi, Nerul, etc.)
BHK Configuration	Single select: 1 BHK, 1.5 BHK, 2 BHK, 2.5 BHK, 3 BHK, 4 BHK, 4+ BHK
Carpet Area (sq ft)	Numeric input — Range: 200–10,000 sq ft
Floor Number	Numeric — 0 (Ground) to 50
Total Floors in Building	Numeric — 1 to 60
Property Age (years)	Numeric — 0 (under construction) to 50
Amenities	Multi-select checkboxes: Pool, Gym, Clubhouse, Lift, Security, Parking, Garden
Furnishing Status	Radio: Bare Shell / Semi-Furnished / Fully Furnished
Transaction Type	Toggle: Sale / Resale
Developer Type	Radio: CIDCO / Private Developer

### 4.2 Prediction Output

- Predicted price (₹ Lakhs) — primary output rendered prominently.
- Confidence interval: lower and upper bound at 90% confidence.
- Price per sq ft — derived metric displayed alongside total price.
- Feature contribution chart: SHAP-style bar chart showing top 5 value drivers.
- Comparable properties: 3–5 similar listings from dataset with price + locality.
- Price trend visualization: 12-month price trend line chart for the selected locality.
- Market position indicator: gauge showing predicted price vs. locality average.

### 4.3 Admin Dashboard

- Secure login (JWT-based, role-separated: Admin / Analyst).
- Dataset upload: CSV ingestion with schema validation and duplicate detection.
- Data quality metrics: missing value counts, outlier flagging, feature distribution plots.
- Model management: view current model version, trigger retraining, compare model versions.
- Usage analytics: prediction volume by date, locality heatmap, popular BHK types.
- Error log viewer: API errors, failed predictions, validation rejections.

---

## 4.4 Downloadable Prediction Report (PDF)

- One-click PDF export from prediction results page.

Report contents:

- Property summary (all input parameters)
- Predicted price with confidence interval
- Feature importance breakdown
- Locality price trend chart
- Comparable properties table
- Model metadata (version, training date, accuracy metrics)
- Disclaimer and methodology note

## 5. Non-Functional Requirements

Requirement	Specification
Prediction API Latency	P50 < 400ms   P95 < 800ms   P99 < 1500ms
Concurrent Users	Support 500 concurrent sessions without degradation at MVP
Scalability	Horizontal scaling via containerized deployment (Docker + orchestration)
Availability	99.5% uptime SLA for production; planned maintenance windows < 2hr/month
Data Privacy	No PII collected; prediction inputs not stored without explicit user consent
Security	HTTPS enforced, CORS locked, rate limiting (100 req/min/IP), JWT auth for admin
Input Validation	Server-side schema validation on all API endpoints (Pydantic models)
Error Handling	Standardized error responses: HTTP codes + human-readable messages
Browser Support	Chrome 90+, Firefox 88+, Safari 14+, Edge 90+; fully responsive mobile
Accessibility	WCAG 2.1 AA compliance for form elements and data visualizations
Audit Logging	All admin actions logged with timestamp and user ID
Model Versioning	MLflow or custom versioning — rollback capability within 2 model versions

## 6. Dataset & Data Engineering Plan

### 6.1 Data Sources

Source	Data Type & Acquisition Strategy
MagicBricks / 99acres / Housing.com	Web scraping (Python + Scrapy/Selenium) — listing price, area, BHK, locality
Maharashtra IGR (Stamp Duty Records)	Public registration data — verified transaction prices (source of truth)
CIDCO Land Records	Plot allocation data, developer classification, node-level pricing
Google Maps API	Distance calculations: nearest station, school, hospital, IT park
OpenStreetMap (Overpass API)	Amenity density mapping per locality
Manual Broker Data Collection	Supplementary ground-truth data for edge localities

### 6.2 Required Features (Final Feature Set)

Feature	Type	Engineering Note
locality_encoded	Categorical → OHE/Target Enc.	30+ categories; use target encoding
bhk	Ordinal Integer	1–6 scale
carpet_area_sqft	Continuous	Log-transform for right-skewed distribution
floor_number	Continuous	Normalize by total_floors
floor_ratio	Derived	floor_number / total_floors — relative position
property_age_years	Continuous	0 = under construction; cap at 50
furnishing_status	Ordinal Encoded	0=bare, 1=semi, 2=fully
amenity_score	Derived Integer	Count of amenities selected (0–7)
developer_type	Binary	0=CIDCO, 1=Private
dist_railway_km	Continuous	From Google Maps API
dist_metro_km	Continuous	NMML1 station proximity
dist_school_km	Continuous	Nearest rated school (≥3.5 stars)
dist_hospital_km	Continuous	Nearest multi-specialty hospital
dist_it_park_km	Continuous	Nearest IT campus or MIDC IT zone

price_per_sqft_locality_avg	Aggregated	Rolling 6-month locality avg from dataset
transaction_type	Binary	0=new sale, 1=resale

### 6.3 Data Pipeline Workflow

The end-to-end data pipeline follows a structured ETL + feature engineering pattern:

```
[Raw Sources] → [Scraper / API Collectors] → [Raw Data Lake (S3/PostgreSQL)]
  → [Validation Layer: Schema Check, Dedup, Outlier Flag]
  → [Feature Engineering Layer: Encoding, Scaling, Distance Calc]
  → [Processed Feature Store] → [Model Training] → [Serialized Model]
```

### 6.4 Data Quality Engineering

#### Missing Value Strategy

- District-level median imputation for continuous features (carpet\_area, floor\_number).
- Mode imputation for categorical features (furnishing\_status, developer\_type).
- Distance features: compute via API at runtime if not pre-cached.

#### Outlier Detection

- IQR method: cap price outliers beyond  $Q1 - 1.5 \times IQR$  and  $Q3 + 1.5 \times IQR$  per locality.
- Z-score threshold ( $|z| > 3.5$ ) for continuous numeric features.
- Business rule validation: price < ₹1 Lakh or > ₹50 Crore flagged for manual review.

#### Categorical Encoding

- Locality: Target encoding (smoothed) — handles high cardinality without OHE explosion.
- BHK, furnishing\_status: Ordinal encoding preserving inherent ordering.
- Amenity multi-select: Aggregated to single amenity\_score integer.

#### Feature Scaling

- StandardScaler applied to all continuous features before tree models (for linear baselines).
- Log1p transformation on carpet\_area\_sqft and price target to normalize skew.

## 7. Machine Learning Approach

### 7.1 Problem Formulation

**Problem Type:** Supervised Regression

**Target Variable:** property\_price (₹ Lakhs) — log-transformed during training, inverse-transformed at output

**Evaluation Split:** 70% Train / 15% Validation / 15% Test (stratified by locality)

**Primary Metric:** R<sup>2</sup> Score (variance explained) + MAE (absolute error in ₹)

### 7.2 Model Comparison Framework

Model	Pros	Cons	Expected R <sup>2</sup>
Linear Regression	Interpretable, fast, baseline	Cannot capture non-linear interactions	0.55–0.65
Ridge / Lasso	Regularization, feature selection	Still linear limitations	0.60–0.68
Random Forest	Non-linear, robust to outliers	Slower inference, less interpretable	0.80–0.86
Gradient Boosting (sklearn)	High accuracy, handles mixed types	Sensitive to hyperparameters	0.82–0.87
XGBoost	Speed + accuracy, GPU support	Black-box without SHAP	0.85–0.91
LightGBM	Fastest training, large datasets	Overfits on small data without tuning	0.85–0.92

### 7.3 Final Model Selection Rationale

*Primary model: XGBoost Regressor with SHAP explainability. Selected for its superior accuracy on tabular real estate data, native handling of missing values, and compatibility with SHAP for feature attribution — a core product requirement.*

Selection criteria:

- XGBoost outperforms linear baselines by  $\geq 20$  R<sup>2</sup> points on real estate datasets.
- SHAP (SHapley Additive exPlanations) integration enables per-prediction feature attribution.
- Inference time: < 50ms for single prediction — well within API latency budget.
- Ensemble fallback: If XGBoost fails health check, fall back to Random Forest serving layer.

### 7.4 Cross-Validation & Hyperparameter Tuning

- Cross-validation: 5-fold stratified KFold, stratified by locality cluster.
- Hyperparameter search: Optuna Bayesian optimization (faster than GridSearch, better than RandomSearch).

---

**Key hyperparameters tuned:**

- `n_estimators`: 100–1000 (optimal range ~400–600)
- `max_depth`: 3–10 (real estate data: typically 5–7)
- `learning_rate`: 0.01–0.3 (with early stopping at 50 rounds)
- `subsample`: 0.6–1.0
- `colsample_bytree`: 0.6–1.0
- `reg_alpha` (L1) and `reg_lambda` (L2): 0.01–10.0

## 7.5 Feature Importance & Explainability

- SHAP TreeExplainer: generates per-prediction Shapley values for the top 5 features.
- Global importance: SHAP summary plot used to validate model is not relying on spurious correlations.
- Locality-level permutation importance: identify which features drive variance within each node.
- Bias audit: check for systematic over/under-prediction by BHK tier and locality income segment.

## 7.6 Confidence Interval Generation

- Quantile Regression on top of XGBoost: train separate models for 5th and 95th percentile.
- Alternatively: Conformal Prediction wrapper for distribution-free coverage guarantees.
- Output: predicted price  $\pm$  interval displayed to user with methodology tooltip.

## 7.7 Model Monitoring & Drift Detection

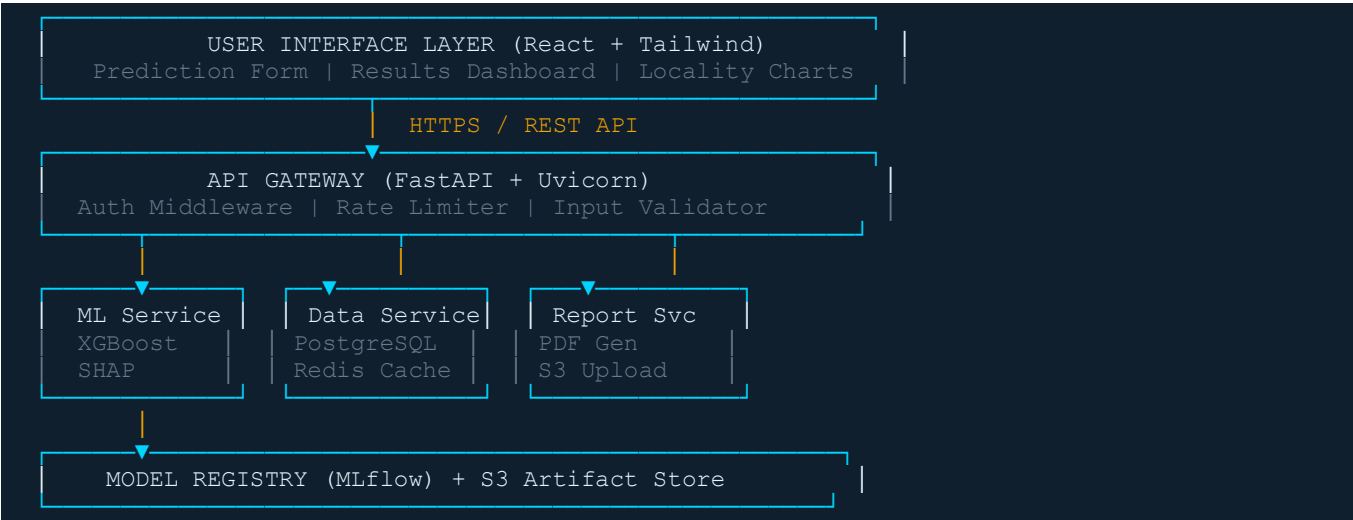
- Population Stability Index (PSI) computed monthly on incoming feature distributions.
- Prediction drift: monitor rolling 30-day MAE against baseline test set MAE.
- Alert threshold: trigger retraining if  $PSI > 0.2$  or rolling MAE degrades  $> 15\%$ .
- MLflow tracks all experiments: parameters, metrics, artifacts, and model lineage.

## 8. Technical Architecture

### 8.1 Stack Overview

Layer	Technology Choices
Frontend (Primary)	React 18 + Vite + Tailwind CSS + Recharts + React Query
Frontend (MVP Alt)	Streamlit — rapid prototype for internal/academic demo
Backend API	Python 3.11 + FastAPI + Uvicorn (ASGI)
ML Runtime	scikit-learn + XGBoost + SHAP + joblib (model serialization)
Data Store	PostgreSQL (structured data) + Redis (prediction cache, rate limiting)
File Storage	AWS S3 (datasets, PDF reports, model artifacts)
Authentication	JWT (python-jose) + bcrypt for admin panel
Containerization	Docker + Docker Compose (dev) + Kubernetes (prod roadmap)
CI/CD	GitHub Actions — lint → test → build → deploy
Hosting (MVP)	Render.com (API + frontend) / Vercel (static frontend)
Hosting (Scale)	AWS EC2 + ALB + RDS + ElastiCache
Monitoring	Prometheus + Grafana (metrics) + Sentry (error tracking)
Experiment Tracking	MLflow (self-hosted or Dagshub)

### 8.2 System Architecture Flow



### 8.3 API Endpoints (FastAPI)

Endpoint	Description
POST /api/v1/predict	Submit property features → returns predicted price + SHAP values
GET /api/v1/localities	Returns list of supported localities with avg price per sqft
GET /api/v1/trends/{locality}	Returns 12-month price trend data for a locality
GET /api/v1/comparables	Returns similar properties from dataset matching input profile
POST /api/v1/reports/generate	Generates and returns PDF report URL (S3 signed URL)
POST /admin/v1/dataset/upload	Admin: Upload new CSV dataset with schema validation
POST /admin/v1/model/retrain	Admin: Trigger model retraining pipeline
GET /admin/v1/analytics	Admin: Usage statistics and model performance metrics
GET /health	Health check endpoint for load balancer

## 9. UI/UX Design Specification

*Design Philosophy: Data-forward minimalism. Every visual element earns its place by reducing cognitive load or revealing insight. Inspired by Bloomberg Terminal clarity, Stripe Dashboard elegance, and Linear.app precision.*

### 9.1 Color System

Token	HEX Value & Usage
--color-primary	#1A3C5E — Deep Navy. Headers, primary text, brand elements
--color-accent	#0D8ABC — Teal Blue. CTA buttons, active states, interactive links
--color-accent-warm	#F4A300 — Amber Gold. Highlights, confidence bands, alerts
--color-bg-base	#F3F6F9 — Off-white. Page background, form backgrounds
--color-bg-card	#FFFFFF — Pure white. Card surfaces, modals
--color-bg-dark	#0F1F2E — Near-black. Code blocks, dark nav variant
--color-text-primary	#0F1F2E — Primary body text
--color-text-secondary	#4A6785 — Labels, captions, helper text
--color-text-muted	#6C7A8A — Placeholder text, disabled states
--color-border	#C8DAEA — Default border color for cards and inputs
--color-success	#1E8449 — Positive delta indicators
--color-error	#C0392B — Validation errors, negative deltas
--color-surface-ice	#E8F4FB — Light teal tint for callout boxes

### 9.2 Typography System

Role	Specification
Font Family (Primary)	Inter — variable font, loaded via Google Fonts CDN
Font Family (Mono)	JetBrains Mono — code blocks, metric values
Display XL (H1)	40px / 600 weight / -0.5px letter-spacing / line-height 1.15
Display L (H2)	28px / 600 weight / -0.3px letter-spacing / line-height 1.2
Display M (H3)	20px / 600 weight / 0px letter-spacing / line-height 1.3
Body Large	16px / 400 weight / 0.1px letter-spacing / line-height 1.65
Body Regular	14px / 400 weight / 0.1px letter-spacing / line-height 1.6
Caption / Label	12px / 500 weight / 0.3px letter-spacing / UPPERCASE for labels
Metric Display	JetBrains Mono 32px / 700 weight — for price output

Code / Data

JetBrains Mono 13px / 400 weight

## 9.3 Spacing & Grid System

- Base unit: 4px. All spacing uses multiples: 4, 8, 12, 16, 24, 32, 48, 64, 96.
- Grid: 12-column grid with 24px gutters. Max content width: 1280px. Sidebar: 280px fixed.
- Card padding: 24px all sides. Section padding: 48px vertical, 32px horizontal.
- Form field height: 44px. Icon size: 20px (inline), 24px (standalone).
- Border radius: 8px (cards), 6px (inputs/buttons), 4px (tags/badges), 12px (modals).

## 9.4 Component Library

### Primary CTA Button

- Background: #0D8ABC | Text: #FFFFFF | Height: 44px | Padding: 0 24px
- Border-radius: 6px | Font: Inter 14px/600 | Letter-spacing: 0.2px
- Hover: background #0A6E96, transform translateY(-1px), box-shadow 0 4px 12px rgba(13,138,188,0.3)
- Active: transform translateY(0), box-shadow none
- Disabled: opacity 0.4, cursor not-allowed
- Transition: all 150ms cubic-bezier(0.4, 0, 0.2, 1)

### Input Fields

- Height: 44px | Border: 1.5px solid #C8DAEA | Border-radius: 6px
- Focus ring: 2px outline #0D8ABC with 2px offset | Background: #FFFFFF
- Error state: border #C0392B, error icon right-aligned, error message 12px below
- Placeholder: color #6C7A8A | Padding: 0 12px | Font: Inter 14px/400

### Prediction Result Card

- Background: #FFFFFF | Border: 1px solid #C8DAEA | Border-radius: 8px
- Price header: 40px JetBrains Mono, #1A3C5E, with animated count-up on load
- Confidence band: #F4A300 amber bar showing lower–upper range
- Left border accent: 4px solid #0D8ABC
- Hover: box-shadow 0 8px 24px rgba(26,60,94,0.12), border-color #0D8ABC

### Data Visualization Rules

- Chart library: Recharts (React) for line/bar charts; custom SVG for gauge.
- Price trend line: color #0D8ABC, stroke 2px, area fill gradient #0D8ABC→transparent
- Bar charts (feature importance): horizontal, color scale from #1A3C5E to #0D8ABC
- Comparable properties: data grid with alternating row shading #F3F6F9
- Loading skeleton: animated shimmer in #E8F4FB for all async chart zones

## 9.5 Micro-interactions & Motion

- Form field focus: smooth border color transition 150ms + subtle scale(1.005) on label

- Submit button: spinner replaces text on prediction loading (not disabled, shows progress)
- Result reveal: price card animates in with fadeIn + slideUp 300ms ease-out
- Price count-up: numeric ticker animation 800ms from 0 to predicted value (easeOut)
- Chart data: paths draw from left-to-right with 600ms stroke-dashoffset animation
- Hover tooltips: appear 150ms delay, fade in 100ms, offset 8px from cursor
- Tab switching: sliding underline indicator 200ms ease transition

## 9.6 Loading, Empty & Error States

### Loading State

- Skeleton screens — never spinners alone. Match exact layout of loaded content.
- Prediction in progress: animated 3-step progress bar (Validating → Processing → Complete)

### Empty State

- Illustration + 16px body text + CTA. Example: 'No comparable properties found. Try adjusting your BHK or area range.' with 'Adjust Filters' button.

### Error State

- Inline validation: red border + error icon + specific message (not generic 'Something went wrong').
- API errors: toast notification (top-right, 4s timeout) with error code and retry action.
- Network failure: full-page fallback with offline illustration and refresh CTA.

## 9.7 Dark Mode Specification

Token (Dark Mode)	Dark Value
--color-bg-base	#0D1B2A — Deep slate background
--color-bg-card	#162032 — Raised card surface
--color-text-primary	#E8F4FB — Near-white body text
--color-text-secondary	#8BABC4 — Subdued secondary text
--color-border	#1E3348 — Subtle card borders
--color-accent	#1AADDCC — Brighter teal for dark backgrounds
Chart colors	All chart colors lightened by 15% for dark bg contrast compliance

- Dark mode triggered via system preference (prefers-color-scheme: dark) + manual toggle.
- Transition: CSS transition on background-color, color, border-color — 200ms ease.

## 9.8 Responsive Design Breakpoints

Breakpoint	Layout Behavior
< 640px (Mobile)	Single column. Prediction form full-width. Charts collapse to simplified sparklines.

640–1024px (Tablet)	2-column grid. Sidebar collapses to top nav drawer. Charts resize to 100% width.
1024–1280px (Desktop)	Full 12-column grid. Sidebar visible. Charts side-by-side in results.
> 1280px (Wide)	Max-width 1280px centered. Whitespace increases. No layout change.

---

## 10. User Experience Flow

---

### 10.1 Primary User Journey (Buyer/Seller)

7. Landing Page: Hero with value prop, animated price ticker, and single CTA: 'Get Your Property Valuation'.
8. Prediction Form: 3-step wizard — Step 1: Location + BHK. Step 2: Area + Floor + Age. Step 3: Amenities + Furnishing. Progress bar at top. Each step validates before proceeding.
9. Processing State: Animated 3-stage progress indicator. Loading skeleton behind glass morphism overlay.
10. Results Dashboard: Price card reveals with count-up animation. Feature importance chart, locality trend chart, and comparables table load sequentially (staggered 150ms).
11. Deep Dive: User can toggle between 'My Property' and 'Locality Insights' tabs. Comparable properties expand with map pins.
12. Export: 'Download Report' button. PDF generates server-side, download begins within 3 seconds.
13. Re-predict: 'Adjust Inputs' returns to pre-filled form at Step 1.

### 10.2 Admin Workflow

14. Admin Login: Separate /admin route. JWT-secured. No public link in nav.
15. Dashboard Home: KPI tiles — total predictions today, model version, last training date, data rows count.
16. Dataset Management: Drag-and-drop CSV upload. Schema preview. Validation report before commit.
17. Model Management: Trigger retraining. View training logs in real-time (WebSocket stream). Compare metrics between current and candidate model.
18. Analytics: Prediction volume chart, top localities by request volume, BHK distribution pie chart.

### 10.3 Wireframe-Level Screen Descriptions

#### Landing Page

- Full-width hero: dark navy gradient, large headline, animated real-time price ticker strip.
- Feature highlights: 3 icon cards (Instant Valuation / ML Powered / Free Report).
- Trust signals: sample report preview, accuracy metric badges.

#### Prediction Results Page

- Left panel (40%): Input summary + predicted price card + confidence band.
- Right panel (60%): Tabbed — 'Price Drivers' (SHAP chart), 'Locality Trend' (line chart), 'Comparables' (data grid).
- Sticky bottom bar on mobile: price + download button always visible.

# 11. Risk Analysis

Risk	Category	Likelihood	Mitigation
Data Bias — Broker-listed prices skewed high	Data Quality	High	Cross-validate with IGR stamp duty data. Apply locality-level price ceiling/floor rules.
Market Volatility — Model trained on 2023-24 data becomes stale	ML Drift	Medium	Monthly PSI monitoring. Automated retraining triggers. Confidence interval widens when drift detected.
Overfitting — XGBoost memorizes locality-level anomalies	ML Accuracy	Medium	L1/L2 regularization. Strict train/test split. Locality-stratified cross-validation.
Incomplete Listings — Missing amenity/age data for 30%+ records	Data Quality	High	Robust imputation pipeline. Model trained on incomplete records with missingness as feature.
Scraped Data — Platform ToS compliance risk	Legal	Medium	Supplement with Maharashtra IGR public records. Frame scraping as academic/research use.
Cold Start — Low confidence for rare localities	ML Coverage	Medium	Cluster rare localities with similar nodes. Report lower confidence when training data < 50 records.
API Latency Spikes — SHAP computation expensive	Performance	Low	Pre-compute SHAP for cached predictions. Async SHAP computation post-response.

## 12. Development Roadmap

8-Week Sprint Plan — Parallel tracks for ML, Backend, and Frontend

Week	Milestones & Deliverables
Week 1 — Foundation	Data collection pipeline setup (scrapers + IGR data). PostgreSQL schema design. Project repo, CI/CD skeleton, Docker setup. Exploratory Data Analysis (EDA) notebook.
Week 2 — Data Engineering	Feature engineering pipeline (encoding, scaling, distance calculation). Missing value and outlier handling. Final cleaned dataset (target: 5,000+ records). Feature store schema.
Week 3 — ML Development	Baseline models (Linear, Ridge). XGBoost training. Optuna hyperparameter tuning. SHAP integration. MLflow experiment tracking. Model evaluation report.
Week 4 — API Development	FastAPI project structure. /predict endpoint with Pydantic validation. Model loading + inference service. Redis caching layer. Unit tests for all endpoints.
Week 5 — Frontend Core	React + Tailwind project setup. Prediction form (3-step wizard). API integration. Results dashboard (price card + basic charts). Responsive mobile layout.
Week 6 — Features + Admin	PDF report generation (ReportLab / WeasyPrint). Admin panel (dataset upload, model trigger). Locality trends chart. Comparable properties grid. Dark mode.
Week 7 — Integration & QA	End-to-end integration testing. Performance benchmarking (load test with Locust). Cross-browser testing. Accessibility audit. Bug fixes and polish.
Week 8 — Deployment	Production deployment (Render/AWS). Domain setup + SSL. Monitoring (Sentry + Prometheus). User acceptance testing. Documentation. Demo video + portfolio write-up.

### 12.1 Key Milestones

- Day 7: Clean dataset with 5,000+ Navi Mumbai records ready for training.
- Day 14: XGBoost model achieving  $R^2 \geq 0.85$  on validation set.
- Day 21: FastAPI prediction endpoint live on local environment.
- Day 35: Full-stack MVP running locally (form → API → results).
- Day 49: Admin panel complete. PDF export functional.
- Day 56: Production deployment live. Monitoring active.

## 13. Future Enhancements

Enhancement	Description & Technical Approach
Rental Prediction Model	Separate XGBoost model trained on rental listings. Inputs: same property features + expected rental yield. Output: monthly rent estimate + gross yield %.
ROI Calculator	Financial modeling module: purchase price + rental income - EMI - maintenance = net ROI. Visualized as 5-year projection chart.
Price Forecasting (Time Series)	Prophet or LSTM model on locality-level price index data. Input: historical quarterly prices. Output: 12-month forward price forecast with confidence bands.
AI Chatbot for Property Queries	RAG-based chatbot using LangChain + OpenAI API. Indexed on property database, locality guides, and CIDCO regulations. Natural language price queries.
Location Heatmap Analytics	Leaflet.js interactive map with price density heatmap overlay. Filter by BHK, price range, amenities. Click-to-predict on map zones.
Mobile Application	React Native app (iOS + Android). Offline-capable with cached locality data. Camera-based property feature detection using CV model.
AVM API (B2B Product)	REST API product for housing finance companies. Automated collateral valuation. SLA-backed, white-labeled, priced per-call with volume tiers.
Builder / Developer Dashboard	Invite-only portal for real estate developers. Track price competitiveness of their projects vs. market. Pre-launch pricing intelligence.

## Appendix

### A. Glossary

Term	Definition
MAE	Mean Absolute Error — average absolute difference between predicted and actual price
RMSE	Root Mean Square Error — penalizes large prediction errors more than MAE
R <sup>2</sup> Score	Coefficient of determination — proportion of price variance explained by model
SHAP	SHapley Additive exPlanations — framework for interpreting ML model predictions
PSI	Population Stability Index — measure of feature distribution drift over time
AVM	Automated Valuation Model — ML-based property valuation, standard in mortgage lending
OHE	One-Hot Encoding — categorical variable transformation for ML models
IGR	Inspector General of Registration — Maharashtra government body recording property transactions
CIDCO	City and Industrial Development Corporation — Navi Mumbai's planning and development authority
NAINA	Navi Mumbai Airport Influence Notified Area — expanded development zone around new airport

### B. Technology License Summary

Technology	License
React 18	MIT License
FastAPI	MIT License
XGBoost	Apache License 2.0
scikit-learn	BSD License
SHAP	MIT License
MLflow	Apache License 2.0
PostgreSQL	PostgreSQL License (open source)
Tailwind CSS	MIT License
Recharts	MIT License

---

— *End of Document* —

NaviEstimate — House Price Prediction System | PRD v1.0 | Confidential