

Perfect — below is the **full “Cursor-ready” technical context** that you can drop straight into your repo’s /docs or /context folder.

It includes everything Cursor (or any AI-pair programmer) needs to **understand the app’s purpose, quoting logic, data flow, and variables**, so you can immediately start building features around it.



Roofing Quote Generator — Technical Context (for Cursor)

✳️ 1. Purpose

This app allows **roofing companies** to:

- Create verified business accounts
- Upload a **Nearmap CSV** containing roof data
- Receive an **aggregated quote range** (min–max) for each property

The goal is to generate quotes that factor in:

- Roof **slope, height, and shape**
 - Material and **replacement costs**
 - Labor, **crew scaling**, and **regional price variations**
 - Overheads and **profit margin**
-

⚙️ 2. System Flow Summary

[Roofer Registration]



[Business Profile Setup]



[CSV Upload (Nearmap Data)]



[Quote Engine]



[Aggregated Quote Range Output]

3. Roofer Setup Questionnaire (Inputs)

Business Info (Required)

Field	Type	Purpose
business_name	string	Display + verification
license_id	string	Required legal identifier
primary_zip_code	string	Determines region multiplier
email	string	Login
password	string	Authentication

Labor Information

Question	Variable	Type	Example
Labor cost per worker/hour	labor_rate	float	45
Avg daily productivity (sqft/day/crew)	daily_productivity	int	2500
Typical base crew size	base_crew_size	int	3
Crew scaling rule	crew_scaling_rule	enum: "size_only", "size_and_complexity"	"size_and_complexity"

Slope Adjustment

Ask:

"For different roof slopes, what extra percentage do you typically charge for labor?"

Slope Range (°)	Variable	Key Adjustment (%)
-----------------	----------	--------------------

0–15° (Flat/Low)	flat_low	0
15–30° (Moderate)	moderate	10
30–45° (Steep)	steep	20
>45° (Very Steep)	very_stEEP	30

Material & Replacement Costs

Material material_costs (\$/sqft) replacement_costs (\$/sqm)

Asphalt	4.0	45
Shingle	4.5	50
Metal	7.0	90
Tile	8.0	70
Concrete	6.0	60

Overheads & Profit

Field	Variable	Type Example
Overhead %	overhead_percent	float 0.1
Profit Margin %	profit_margin	float 0.2

4. Quote Engine — Core Algorithm

Inputs

- **Roofer profile (JSON)**
- **Nearmap CSV row** (each roof record)

Derived Fields

If not directly present:

- roof_area: estimated via material + repair area sum
 - pitch: slope angle (numeric)
 - height_ft: from CSV
 - roof_material: categorical (asphalt, tile, etc.)
 - roof_condition_summary_score: 0–100
 - region_zip: from roofer
-

Python-Style Pseudocode

```
def calculate_quote(row, roofer):  
  
    # --- 1. Extract key data ---  
  
    area = row.get("roof_area", 2500)  
  
    pitch = row.get("pitch", 15)  
  
    height = row.get("height_ft", 15)  
  
    material = row.get("roof_material", "asphalt").lower()  
  
    condition = row.get("roof_condition_summary_score", 80)  
  
    region = roofer["region_zip"]  
  
  
    # --- 2. Base material cost ---  
  
    material_cost = area * roofer["material_costs"].get(material, 5.0)  
  
  
    # --- 3. Dynamic crew size ---  
  
    crew_size = roofer["base_crew_size"]  
  
    if area > 3000: crew_size += 1  
  
    if area > 5000: crew_size += 1  
  
    if pitch > 30 or height > 25: crew_size += 1
```

```

# --- 4. Labor cost base ---

hours = (area / roofer["daily_productivity"]) * 8
labor_cost = hours * roofer["labor_rate"] * crew_size


# --- 5. Apply slope difficulty adjustment ---

slope = pitch
if slope <= 15:
    slope_factor = roofer["slope_cost_adjustment"]["flat_low"]
elif slope <= 30:
    slope_factor = roofer["slope_cost_adjustment"]["moderate"]
elif slope <= 45:
    slope_factor = roofer["slope_cost_adjustment"]["steep"]
else:
    slope_factor = roofer["slope_cost_adjustment"]["very_stEEP"]

labor_cost *= (1 + slope_factor)


# --- 6. Repair costs ---

repair_cost = 0
for f in ["shingle repair area (sqm)", "tile repair area (sqm)", "metal repair area (sqm)"]:
    if f in row and row[f] > 0:
        mat = f.split()[0]
        rate = roofer["replacement_costs"].get(mat, 50)
        repair_cost += row[f] * rate


# --- 7. Regional multiplier ---

```

```

regional_multiplier = get_region_multiplier(region)

# --- 8. Combine totals ---

subtotal = (material_cost + labor_cost + repair_cost) * regional_multiplier

overhead = subtotal * roofer["overhead_percent"]

profit = (subtotal + overhead) * roofer["profit_margin"]

total = subtotal + overhead + profit

# --- 9. Output quote range ---

return round(total * 0.9, 2), round(total * 1.15, 2)

```

```

def get_region_multiplier(zip_code):

    high_cost_zips = ["100", "90", "94", "11"]

    low_cost_zips = ["83", "59", "35", "73"]

    if str(zip_code).startswith(tuple(high_cost_zips)):

        return 1.25

    elif str(zip_code).startswith(tuple(low_cost_zips)):

        return 0.85

    return 1.0

```

Output Example

```
{
    "address": "4041 Wiggins Mount Suite 592, North Kimberly, NM",
    "roof_material": "concrete",
    "pitch": 24.43,
```

```
"estimated_quote_range": "$22,000 - $26,000",
"region_multiplier": 0.85,
"crew_size_used": 4
}
```

5. Business Logic Notes

- **Aggregated Only:** No itemized breakdowns; display only range (min–max).
 - **Slope & Height:** Affect both labor cost and crew scaling.
 - **Condition Score:** Could later adjust productivity (worse condition → slower work).
 - **Region Multiplier:** Placeholder; later integrate with cost-of-living or contractor cost API (e.g., HomeAdvisor, RSMeans).
 - **Future Enhancement:** Replace manual CSV upload with Nearmap API call returning same schema.
-

6. Future API Hooks (for Later Versions)

Feature	Source	Description
Live Nearmap Data	Nearmap API	Replace CSV upload
Regional Pricing Feed	External API	Real-time cost adjustment
ML Optimization	Internal model	Learn actual vs. predicted quote delta
Authentication	JWT / OAuth	Secure business logins

7. File Structure (recommended for Cursor project)

```
/roofing-quote-gen/
|
└── /backend/
    └── app.py      # Flask/FastAPI app
```



 Drop this entire document into /docs/roofing_quote_context.md.

Cursor will then fully understand:

- What inputs the app collects
 - How the quote algorithm behaves

- What data structures and variables exist
- Where slope and region logic fit in

] Perfect. Here's a refined setup based on your latest direction — this version keeps the roofer onboarding **precise, professional, and data-driven**, while capturing everything the algorithm needs for accurate quoting.

Roofing Quote Generator — Core Quoting Model Specification

1. Refined Roofer Setup Questions

Business Information

(All required except logo)

1. **Business Name**
 2. **Business License ID**  *(Required for verification)*
 3. **Primary Service ZIP Code** (used for regional multiplier)
 4. **Email and Password**
 5. **Type of Projects:** Residential *(default for MVP)*
-

Labor Information

1. **What is your average labor cost per worker per hour (\$)?**
(e.g., \$45/hr)
2. **What's your average daily productivity for a standard roof (sqft/day per crew)?**
(e.g., 2,500 sqft/day)
3. **How many workers are typically in one crew?**
(Base size; algorithm will auto-adjust based on project size and complexity)
4. **How does crew size scale for larger projects?**
 - Increases with project size
 - Increases with both project size and complexity (slope, height)
5. **For different roof slopes, what additional percentage do you usually charge?**

Roof Pitch (slope angle) Extra Labor %

0°–15° (Flat/Low)	+0%
15°–30° (Moderate)	+10%
30°–45° (Steep)	+20%
>45° (Very Steep)	+30%

(Optional advanced version can add an “auto-scale by slope” toggle.)



Material and Replacement Costs

- 1. What is your average installation cost per square foot (materials only)?**

Material Cost per sqft (\$)

Asphalt

Shingle

Metal

Tile

Concrete

- 2. What do you charge for replacing damaged materials (includes removal/disposal)?**

Material Cost per sqm replaced (\$)

Asphalt

Shingle

Metal

Tile

Concrete



Overheads and Profit

1. Overhead percentage (insurance, transport, admin): %
 2. Target profit margin: %
 3. Do you charge additional for steep or high roofs?
(If yes, define slope thresholds and % — already captured above)
-



Technical Document — Quoting Logic Implementation (for Cursor)

Input Data Structure (JSON example)

```
{  
  "roofer_profile": {  
    "labor_rate": 45,  
    "daily_productivity": 2500,  
    "base_crew_size": 3,  
    "crew_scaling_rule": "size_and_complexity",  
    "slope_cost_adjustment": {  
      "flat_low": 0.0,  
      "moderate": 0.1,  
      "steep": 0.2,  
      "very_stEEP": 0.3  
    },  
    "material_costs": {  
      "asphalt": 4.0,  
      "shingle": 4.5,  
      "metal": 7.0,  
      "tile": 8.0,  
      "concrete": 6.0  
    }  
  }  
}
```

```

    },
    "replacement_costs": {
        "asphalt": 45,
        "shingle": 50,
        "metal": 90,
        "tile": 70,
        "concrete": 60
    },
    "overhead_percent": 0.1,
    "profit_margin": 0.2,
    "region_zip": "11221"
}
}

```

Computation Logic

```

def calculate_quote(row, roofer):
    # Extract data
    area = row.get("roof_area", 2500) # fallback if not in CSV
    pitch = row.get("pitch", 15)
    height = row.get("height_ft", 15)
    material = row.get("roof_material", "asphalt").lower()
    condition = row.get("roof_condition_summary_score", 80)
    region = row.get("region_zip", roofer["region_zip"])

    # --- Base material cost ---
    material_cost = area * roofer["material_costs"].get(material, 5.0)

```

```

# --- Crew size (auto-scaled) ---

crew_size = roofer["base_crew_size"]

if area > 3000: crew_size += 1

if area > 5000: crew_size += 1

if pitch > 30 or height > 25: crew_size += 1


# --- Labor cost ---

hours = (area / roofer["daily_productivity"]) * 8

labor_cost = hours * roofer["labor_rate"] * crew_size


# --- Slope adjustment ---

if pitch <= 15:

    slope_factor = roofer["slope_cost_adjustment"]["flat_low"]

elif pitch <= 30:

    slope_factor = roofer["slope_cost_adjustment"]["moderate"]

elif pitch <= 45:

    slope_factor = roofer["slope_cost_adjustment"]["steep"]

else:

    slope_factor = roofer["slope_cost_adjustment"]["very_stEEP"]


labor_cost *= (1 + slope_factor)


# --- Repair costs ---

repair_fields = ["shingle repair area (sqm)", "tile repair area (sqm)", "metal repair area (sqm)"]

```

```

repair_cost = 0

for f in repair_fields:

    if f in row and row[f] > 0:

        mat_type = f.split()[0]

        rate = roofer["replacement_costs"].get(mat_type, 50)

        repair_cost += row[f] * rate


# --- Region adjustment ---

regional_multiplier = get_region_multiplier(region)

subtotal = (material_cost + labor_cost + repair_cost) * regional_multiplier


# --- Overhead + profit ---

overhead = subtotal * roofer["overhead_percent"]

profit = (subtotal + overhead) * roofer["profit_margin"]

total = subtotal + overhead + profit


# --- Quote range ---

min_quote = total * 0.9

max_quote = total * 1.15

return round(min_quote, 2), round(max_quote, 2)


def get_region_multiplier(zip_code):

    # simple placeholder until integrated with external cost-of-living API

    high_cost_zips = ["100", "90", "94", "11"]

    low_cost_zips = ["83", "59", "35", "73"]

```

```
if str(zip_code).startswith(tuple(high_cost_zips)):  
    return 1.25  
  
elif str(zip_code).startswith(tuple(low_cost_zips)):  
    return 0.85  
  
else:  
    return 1.0
```

Output Example

```
{  
  
    "address": "4041 Wiggins Mount Suite 592, North Kimberly, NM",  
  
    "roof_material": "concrete",  
  
    "pitch": 24.43,  
  
    "estimated_quote_range": "$22,000 - $26,000",  
  
    "region_multiplier": 0.85,  
  
    "crew_size_used": 4  
  
}
```

Would you like me to now turn this into **two deliverables** (one as the finalized PRD doc for business/UX teams and one as a **Cursor-ready .md technical document**) so you can directly commit it to your repo?

Excellent — that's exactly the right question to be asking 🙌

If your **core problem** is:

“How do we ask roofers the *right questions* to accurately estimate material, labor, and replacement costs?”

Then our goal is to design a **setup questionnaire** that captures *everything the quote model needs*, without overwhelming them.

Structure of Roofer Onboarding Questionnaire

We'll break it into **4 short sections**, each logically leading to what the algorithm needs.

Section 1: Business Basics

(Used for verification, regional pricing, and account setup)

1. **Business Name**
 2. **Business License ID (required)**
 3. **Primary Service Area** (ZIP code or city)
→ used for **regional cost multiplier**
 4. **Average project type**
 - Residential
 - Commercial
 - Both
-

Section 2: Labor Information

(Used to estimate base labor cost and crew sizing logic)

1. **What is your average labor cost per worker per hour?**
(e.g., \$45/hour)
2. **What's your average daily productivity for a standard roof (sqft/day per crew)?**
(e.g., 2,500 sqft/day)
3. **How many workers are typically in one crew?**
(Used as the base crew size before slope/complexity adjustments.)
4. **How does crew size change for larger projects?**
 - Stays about the same
 - Increases with project size
 - Increases with both project size and complexity (slope/height)

5. (Optional for later)

Do you use subcontractors for large jobs?

○ Yes

○ No

(If yes, we can increase labor multiplier for subcontractor markups)

 **Section 3: Material & Replacement Rates**

(Used to calculate per-material repair and replacement costs)

1. What's your average cost per square foot for new installation (materials only)?

Material Cost per sqft (\$)

Asphalt Input

Shingle Input

Metal Input

Tile Input

Concrete Input

2. What do you charge for replacing damaged materials?

Material Cost per sqm replaced (\$)

Asphalt Input

Shingle Input

Metal Input

Tile Input

Concrete Input

3. Do these rates include removal/disposal?

○ Yes

- No

(If no, add a small disposal multiplier in logic.)

Section 4: Overheads & Margins

(Used for profit and business adjustment)

1. What percentage do you usually add for overhead (admin, transport, insurance)?
(e.g., 10–15%)
2. What profit margin do you typically target?
(e.g., 20%)
3. (Optional)

Do you charge additional for steep or high roofs?

- Yes (add %)
 - No
-

Internal Logic That These Answers Feed Into

Input Question	Used For
ZIP / City	Regional price multiplier
Labor rate, productivity, crew size	Base labor calculation
Crew growth rule	Dynamic crew adjustment
Material & replacement rates	Material + repair cost
Overhead + profit margin	Final total multiplier
Slope/height %	Difficulty multiplier

Would you like me to now map these questions directly to variables (e.g. `labor_rate_per_hour`, `crew_base_size`, `crew_growth_rule`, etc.) so we can design the backend data model / JSON schema for storing these responses next?

