This detailed expansion of your project plan focuses specifically on the **Fixed Reinforced Concrete (RC) Beam** scenario and integrates the "Corrosion" variable as your unique contribution.
This document is structured to serve as the **Chapter 3 (Methodology)** and **Chapter 4 (Implementation)** blueprint for your thesis.

Detailed Project Plan: Predictive Modeling of Natural Frequency Shifts in Corroded Fixed RC Beams

1. The Core Physics & Modeling Strategy

1.1. Why "Fixed" Beams?

Most student papers use "Simply Supported" beams because the math is easy. However, in real infrastructure (bridges, continuous building frames), beams are often **Fixed-Fixed** (clamped at both ends).

- **The Physics:** Fixed ends prevent both *translation* (moving up/down) and *rotation* (bending at the joint).
- **The Result:** Fixed beams are stiffer than simply supported ones, meaning they have **higher natural frequencies**.
- **The Challenge:** Corrosion at the fixed ends (where bending moment is highest) causes a drastic reduction in stiffness, making this a critical safety study.

1.2. The Unique Contribution: "The Stiffness Reduction Method"

Instead of modeling complex chemical rust, you will model the *mechanical consequence* of corrosion.

- **Reference:** Zhang et al. (2021).
- **Concept:** Corrosion eats away the steel rebar and cracks the surrounding concrete. This results in a lower **Effective Moment of Inertia (I_{eff})**.
- **Implementation Logic:**
    - **Pristine State ($C = 0\%$):** Use the full transformed section stiffness.
    - Corroded State ($C > 0\%$): You will introduce a damage factor (α).

      $$EI_{corroded} = EI_{original} \times (1 - \alpha)$$
    - *Where α is proportional to the corrosion level (e.g., 5% corrosion $\approx$ 8% stiffness loss, based on your references).*

---

2. Phase 1: Finite Element Model (FEM) Development

Goal: Build a Python-based simulation engine that calculates frequencies.

2.1. The Algorithm (Matrix Structural Analysis)

You don't need expensive software like ANSYS for the *dataset*; you can code the FEM in Python using `numpy` and `scipy`. This is faster and allows you to generate 2,000 samples in minutes.
The Equation of Motion:

$$[K]\{u\} - \omega^2 [M]\{u\} = 0$$

- $[K]$ = Global Stiffness Matrix (Global stiffness of the beam).
- $[M]$ = Global Mass Matrix.
- ω = Natural Frequency (eigenvalue).

The Step-by-Step Python Workflow:

1. **Discretization:** Divide the beam into **20 elements** (finite segments).
2. **Element Matrices:** For each element, calculate the local stiffness matrix (k) and mass matrix (m) based on input L, E, I, ρ, A.
3. **Assembly:** Assemble these into the Global Matrices $[K]$ and $[M]$.
4. **Applying Boundary Conditions (The "Fixed" Part):**
   - In a Fixed-Fixed beam, the Degrees of Freedom (DOF) at Node 0 and Node N are **locked**.
   - **Action:** Delete the rows and columns in $[K]$ and $[M]$ corresponding to the first two and last two DOFs.
5. **Solution:** Use `scipy.linalg.eigh(K, M)` to solve for Eigenvalues.
   - $\text{Frequency (Hz)} = \frac{\sqrt{\text{Eigenvalue}}}{2\pi}$

2.2. Validation Checkpoints

Before generating the full dataset, run these two specific tests:

- **Test A (Theoretical):** Run a standard steel beam ($L=3m$). Compare the result to the Euler-Bernoulli formula for Fixed-Fixed beams. **Error must be < 1%.**
- **Test B (Experimental - Sivasuriyan):** Input the dimensions from the Sivasuriyan paper. Even though their beam might be Simply Supported, change your code's BCs to match theirs just for this one test. **Target: Match their "Mode 1" frequency.**

---

3. Phase 2: Dataset Generation Strategy

To train a robust ML model, you need a dataset that covers "The Design Space."

3.1. Sampling Method: Latin Hypercube Sampling (LHS)

Don't just use random numbers. Use **LHS**. It ensures you don't accidentally bunch all your samples in one corner (e.g., only short, thick beams).

3.2. The Dataset Matrix (2,000 Rows)

Your CSV file will look like this. Each row is one simulation.

ID	**Length (m)**	**Width (m)**	**Depth (m)**	**Conc. Strength (f'c)**	**Corrosion Level (%)**	**Freq Mode 1 (Hz)**	**Freq Mode 2 (Hz)**
1	4.5	0.3	0.45	30 MPa	**0% (Pristine)**	45.2	120.5
2	4.5	0.3	0.45	30 MPa	**10% (Damaged)**	**41.8**	115.2

ID	Length (m)	Width (m)	Depth (m)	Conc. Strength (f'c)	Corrosion Level (%)	Freq Mode 1 (Hz)	Freq Mode 2 (Hz)
...	...	...	...	...	...	...	...

- **Pristine Data (1,500 rows):** Vary $L, b, h, f'c$. Set Corrosion = 0.
- **Corroded Data (500 rows):** Take 500 random existing beams and re-run them with Corrosion = 5%, 10%, 15%, 20%.

---

4. Phase 3: Machine Learning Model Development

4.1. The Recommended Model: XGBoost (or CatBoost)

While Neural Networks (ANN) are popular in your references (Avcar), **XGBoost** is currently the industry leader for this type of tabular engineering data.

Why XGBoost fits your thesis:

1. **Handles Non-Linearity:** It easily learns the "kink" in the curve where corrosion starts affecting stiffness.
2. **Feature Importance:** It can automatically generate a chart showing which factor (Length vs. Corrosion) impacts frequency the most. This is excellent for your "Results" chapter.
3. **Speed:** It trains in seconds.

4.2. Training Architecture

1. **Input Features (X):** Length, Width, Depth, Concrete Strength, Corrosion Level.
2. **Target (Y):** Natural Frequency (Mode 1). *Note: Train separate models for Mode 2 and Mode 3 if needed.*
3. **Splitting:**
   - **Training Set (70%):** Teaches the model.
   - **Validation Set (15%):** Tunes the parameters (learning rate, depth).
   - **Test Set (15%): Crucial.** This data is never seen by the model until the very end.

5. Phase 4: Validation & Results (The "Thesis Story")

This is how you prove your project worked.

5.1. Accuracy Metrics

You will present a table comparing your ML predictions to the Ground Truth (FEM values) in the Test Set.

- **R^2 Score:** Measures correlation. **Target: > 0.99**.
- **MAE (Mean Absolute Error):** Measures average Hz error. **Target: < 0.5 Hz**.
- **MAPE (Percentage Error): Target: < 2%** (This beats the Horváth reference).

5.2. The "Corrosion Detection" Demonstration (The Zhang Trend)

This is the most important graph in your thesis.

1. Take a single beam (e.g., $L=5m$).

2. Ask the ML model to predict frequencies for Corrosion = 0%, 5%, 10%, 15%, 20%.
3. **Plot:** Corrosion Level (X-axis) vs. Frequency Drop (Y-axis).
4. **Result:** The curve should show a downward trend. You will validate this by overlaying the experimental curve from the **Zhang et al.** paper. If they match, you have successfully modeled reality.

6. Immediate Next Steps for You

1. **Week 1:** Set up your Python environment (Install `numpy`, `pandas`, `scikit-learn`, `xgboost`).
2. **Week 2:** Write the Python script to calculate the frequency of *one* single fixed beam and compare it to the hand-calculated formula.
3. **Week 3:** Wrap that script in a loop to generate the CSV dataset.

Would you like me to generate the **Python code for Week 2 (The Single Fixed Beam FEM Calculator)** so you can verify your physics immediately?