PROJECT TITLE

Automation of Structural Model Generation in STAAD.Pro using Python

OVERVIEW:

This project focuses on automating the preprocessing stage of structural analysis in STAAD.Pro. Raw node data provided in CSV or Excel format is validated and converted into a STAAD-compatible input file using Python.

The goal is to eliminate manual modeling, reduce human error, and create a reusable and scalable automation pipeline.

PROBLEM STATEMENT:

In research and industry workflows, structural geometry is often shared as raw tabular data. Manual conversion of this data into STAAD.Pro models is time-consuming, error-prone, and difficult to reproduce.

SOLUTION APPROACH:

A Python-based automation pipeline was developed with the following steps:

1. Read raw node data from CSV or Excel files
2. Validate input data for:
   - Missing required columns
   - Duplicate node IDs
   - Duplicate node coordinates
   - Formatting inconsistencies
3. Generate a STAAD.Pro-compatible input file (.std) automatically

INPUT:

Input file format: CSV / Excel

Required columns:

- node_id
- x
- y
- z

Example:

```
node_id, x, y, z
1, 1532.14, 1578.57, 0.0
```

OUTPUT:

The output of the script is a STAAD.Pro input file named:

model.std

This file contains:

- Job information
- Unit definition
- Joint coordinates in correct STAAD syntax

The generated file is analysis-ready and can be opened directly in STAAD.Pro on any licensed system.

HOW TO RUN:

1. Place the node data file (nodes.csv or Excel file) in this folder
2. Open terminal in this folder
3. Run the command:

   ```
   python node_to_staad.py
   ```
4. When prompted, enter the input file name
5. The output file model.std will be generated

KEY CONTRIBUTION:

The key contribution of this work is the implementation of strict input validation prior to model generation. This ensures geometric correctness and prevents incorrect structural models from being created.

LIMITATIONS:

- Analysis execution requires licensed STAAD.Pro software
- Current implementation focuses on node generation

FUTURE SCOPE:

- Automation of member connectivity
- Support and load assignment
- Automated analysis execution
- Extraction of stresses and reactions
- Integration with optimization workflows

AUTHOR

Ishita Parmar
B.Tech Civil Engineering
NIT Goa