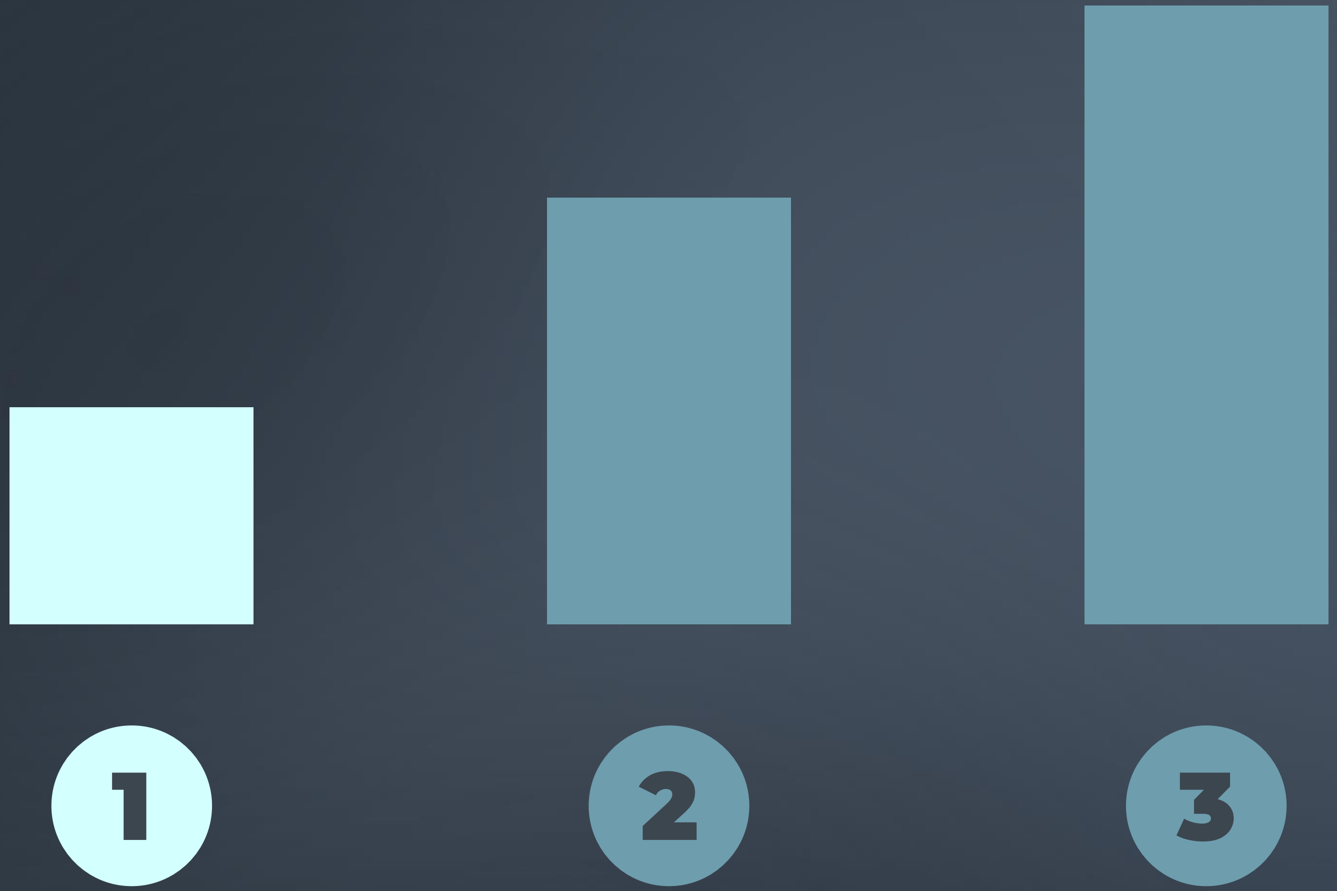
GROUP 6

41934 – ADVANCED BUILDING INFORMATION MODELLING

COST ESTIMATION ANALYSIS

Eliza Anna Grinberga / S223262 Rihards Zamoidiks / S186415

FOR WHO?



LEVEL 1 - BEGINNERS

USE CASE

STRUCTURAL ELEMENTS
Columns, load bearing walls, beams and slabs.

QUANTITY TAKEOFFS
Streamline the process of precise quantity takeoffs.

COST ESTIMATION
Cost estimation for construction projects.

The primary use case for these BIM tools is cost estimation, specifically focusing on structural elements. The main objective of this tool is to streamline the process of precise quantity takeoffs, provided that an IFC model of an appropriate development level is available. This tool aims to improve the decision-making and accuracy in cost estimation for construction projects.

WHO CAN USE THIS TOOL?

PROJECT MANAGERS

Stakeholders involved in construction projects, especially those interested in the structural aspects

CONTRACTORS

Efficiently estimate material costs for any given project.

DESIGNERS

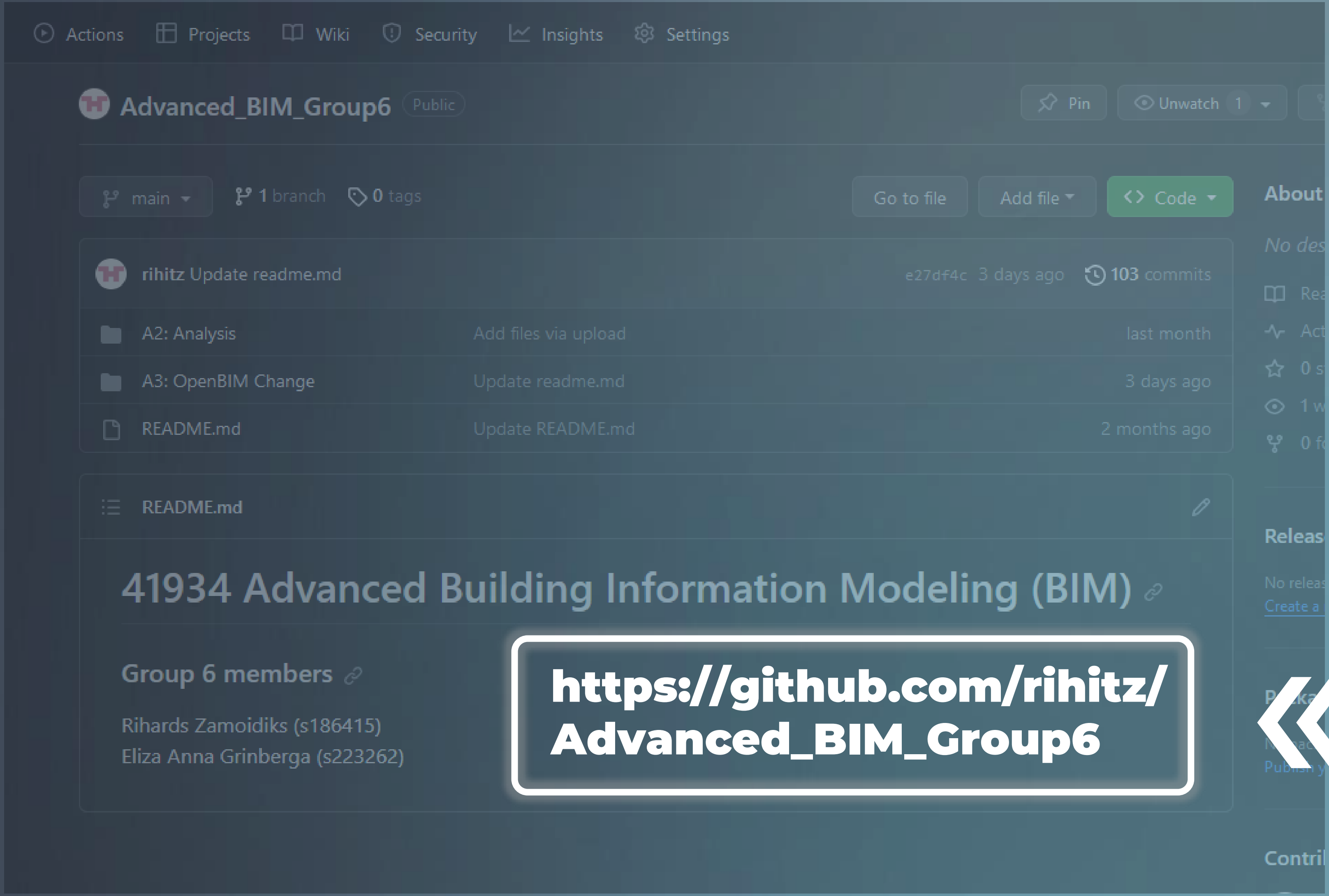
Compare various design concepts in terms of projected expenses, facilitating quick and informed decision-making

LCA

Material quantity takeoffs for precise LCA calculations.

HOW TO USE OUR TOOLS?

STEP 1 – OPEN GITHUB



Actions Projects Wiki Security Insights Settings

Advanced_BIM_Group6 Public

Pin Unwatch 1

main 1 branch 0 tags Go to file Add file Code

rihitz Update readme.md e27df4c 3 days ago 103 commits

A2: Analysis	Add files via upload	last month
A3: OpenBIM Change	Update readme.md	3 days ago
README.md	Update README.md	2 months ago

README.md

41934 Advanced Building Information Modeling (BIM)

Group 6 members

Rihards Zamoidiks (s186415)
Eliza Anna Grinberga (s223262)

About
No des
Rea
Act
0 s
1 w
0 f

Releas
No releas
Create a

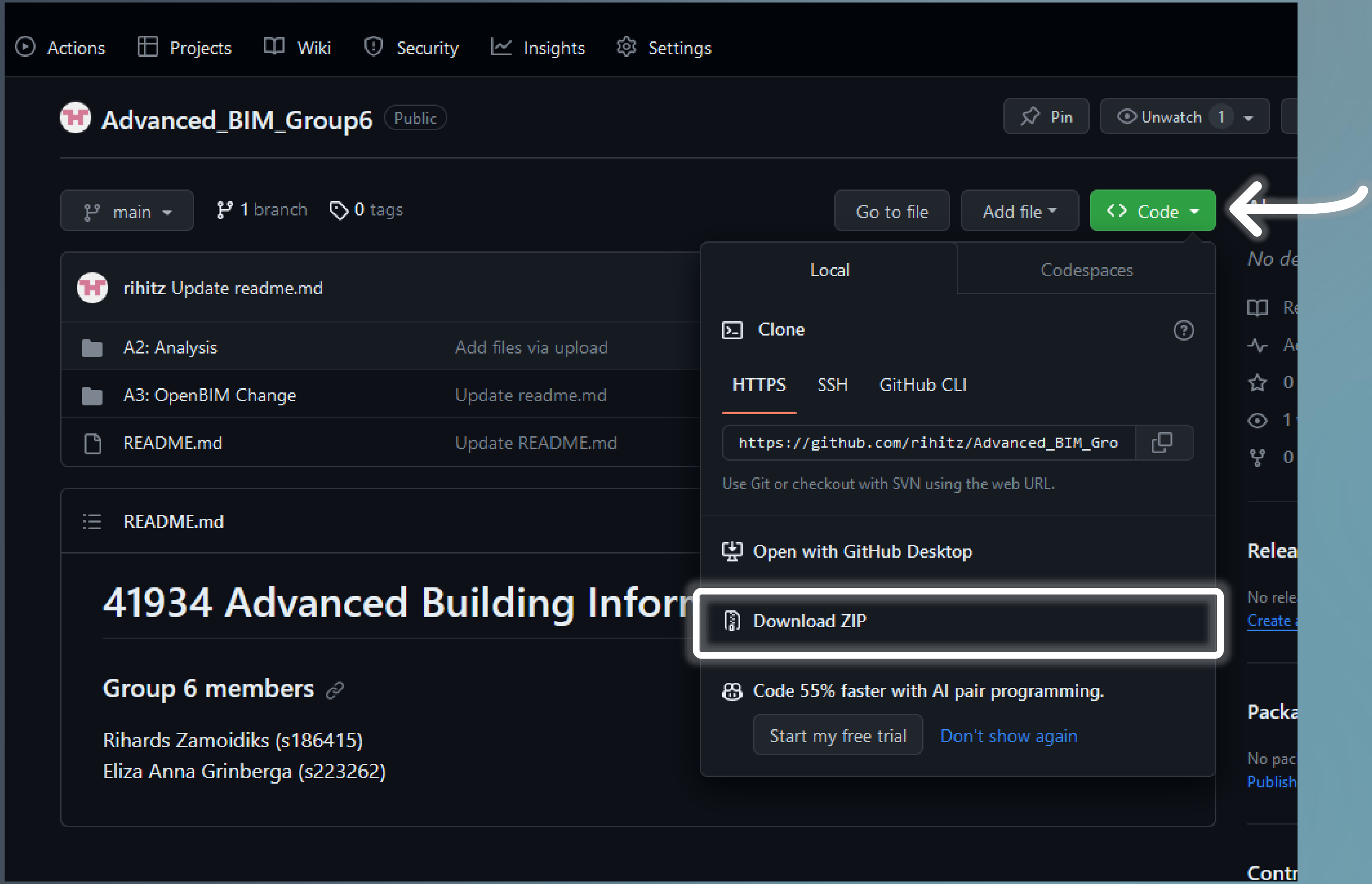
Contri

https://github.com/rihitz/Advanced_BIM_Group6

GO TO

First things first, download the project directory from Github, by following the URL: https://github.com/rihitz/Advanced_BIM_Group6

STEP 2 - DOWNLOAD



rihitz Update readme.md

A2: Analysis	Add files via upload
A3: OpenBIM Change	Update readme.md
README.md	Update README.md

https://github.com/rihitz/Advanced_BIM_Gro

Use Git or checkout with SVN using the web URL.

Code 55% faster with AI pair programming.

README.md

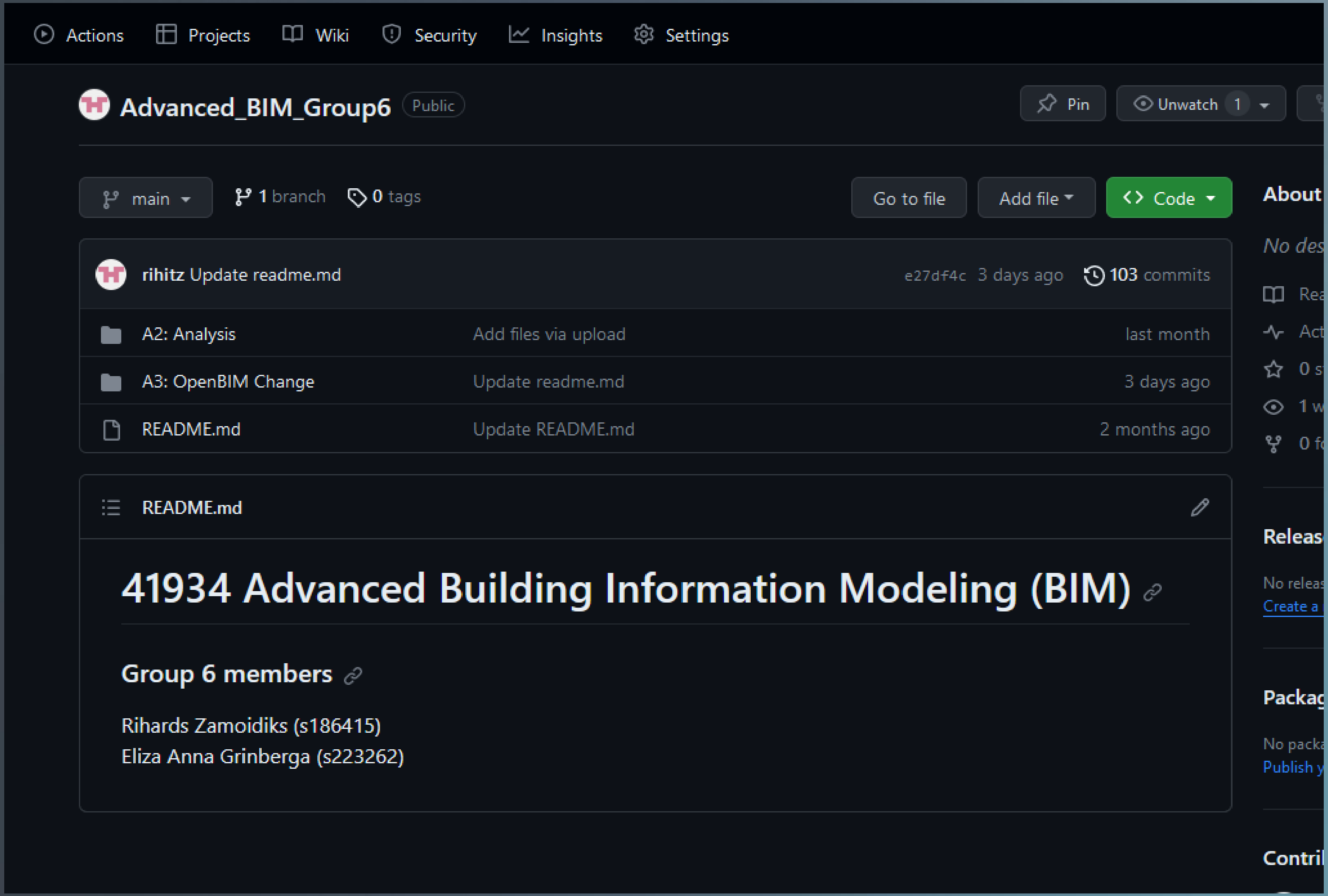
41934 Advanced Building Infor

Group 6 members

Rihards Zamoidiks (s186415)
Eliza Anna Grinberga (s223262)

Once you're at the main page of the project repository, hover over to the big green button that says "Code" and click on it. From the dropdown menu, choose the option 'Download zip file.' This will automatically handle all the necessary setup tasks for the tool, including folder hierarchies, input files, and more

STEP 3 - OPEN



Advanced_BIM_Group6 Public

rihitz Update readme.md e27df4c 3 days ago 103 commits

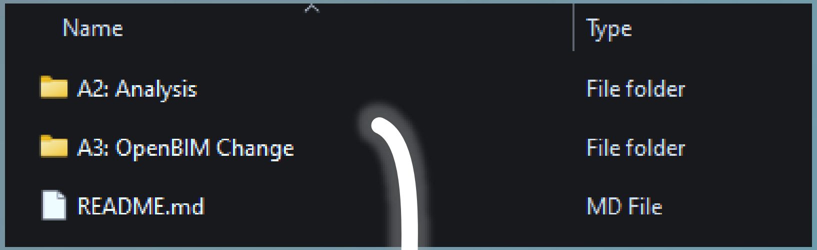
A2: Analysis	Add files via upload	last month
A3: OpenBIM Change	Update readme.md	3 days ago
README.md	Update README.md	2 months ago

README.md

41934 Advanced Building Information Modeling (BIM)

Group 6 members

Rihards Zamoidiks (s186415)
Eliza Anna Grinberga (s223262)



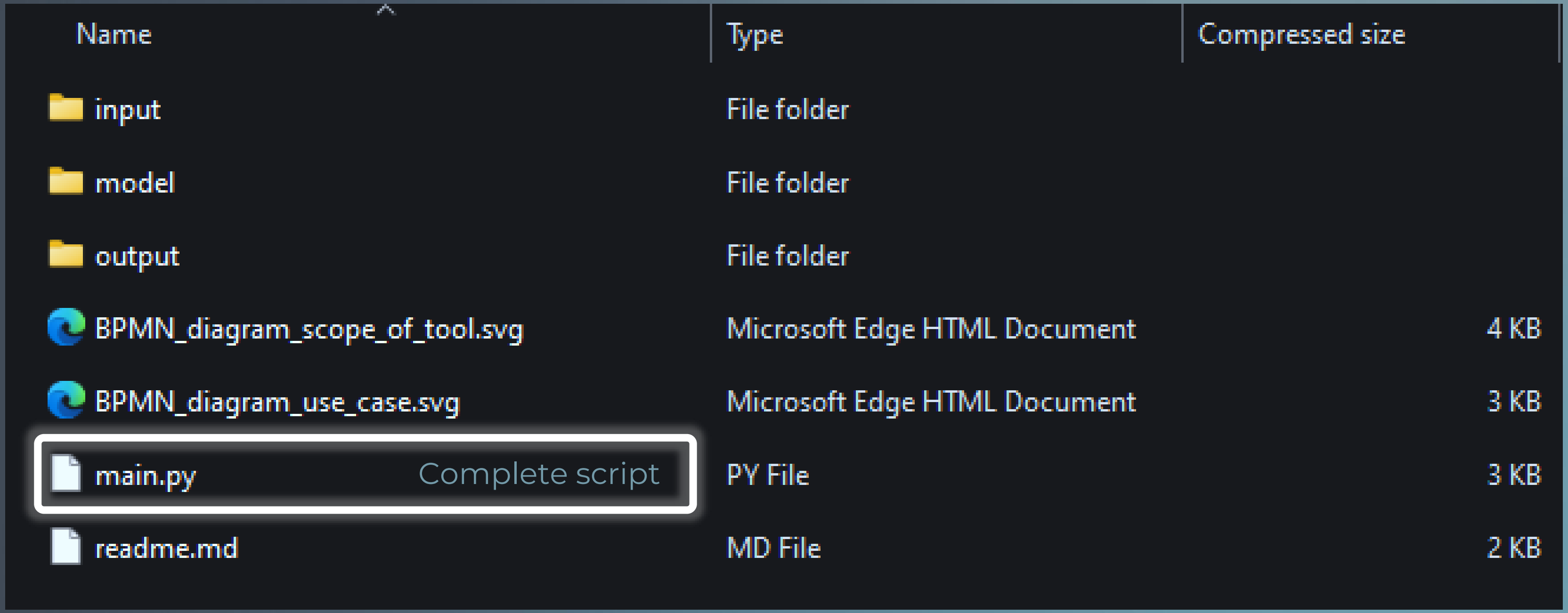
Name	Type
A2: Analysis	File folder
A3: OpenBIM Change	File folder
README.md	MD File

A2
General cost estimation.

A3
Supplements IFC model.

After extracting the downloaded file, you'll find two primary folders, labeled as A2 and A3, corresponding to the two developed tools. The tool in the A2 folder serves as a general cost estimator, while the tool in the A3 folder supplements the IFC model with crucial information, typically present in the architectural models but often missing in structural ones, namely material types and their properties.

STEP 4 – COST ESTIMATION TOOL



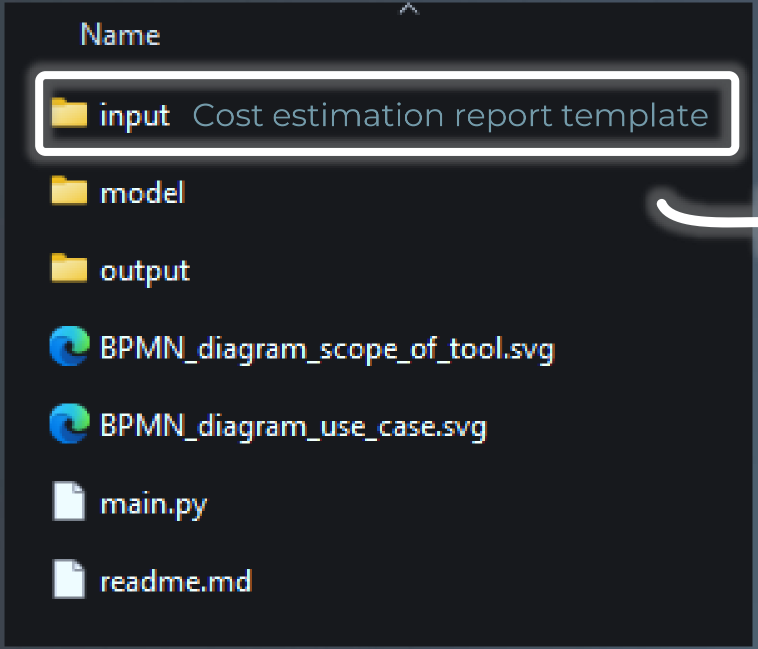
Name	Type	Compressed size
input	File folder	
model	File folder	
output	File folder	
BPMN_diagram_scope_of_tool.svg	Microsoft Edge HTML Document	4 KB
BPMN_diagram_use_case.svg	Microsoft Edge HTML Document	3 KB
main.py	PY File	3 KB
readme.md	MD File	2 KB

Complete script

GitHub > spyder

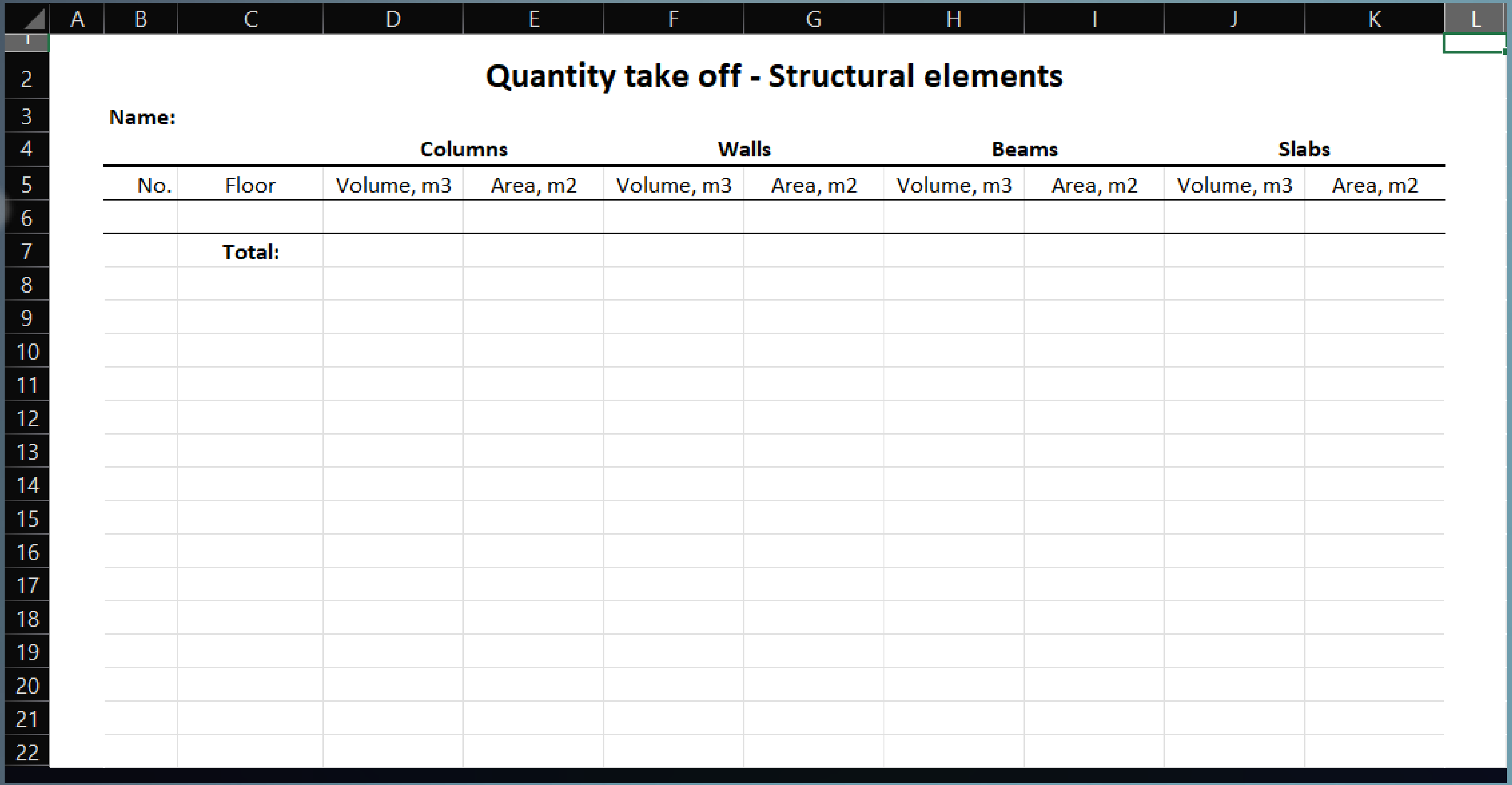
In the tool directory, you'll find various folders and files. The BPMN diagrams provide a general overview of how the tool functions, and the complete script is in 'main.py'. This tool was developed in a Python IDE, so for the most seamless experience, it's recommended to run it on one rather than pasting it into BlenderBIM.

STEP 5 – COST ESTIMATION TOOL



Name

- input Cost estimation report template
- model
- output
- BPMN_diagram_scope_of_tool.svg
- BPMN_diagram_use_case.svg
- main.py
- readme.md



Quantity take off - Structural elements

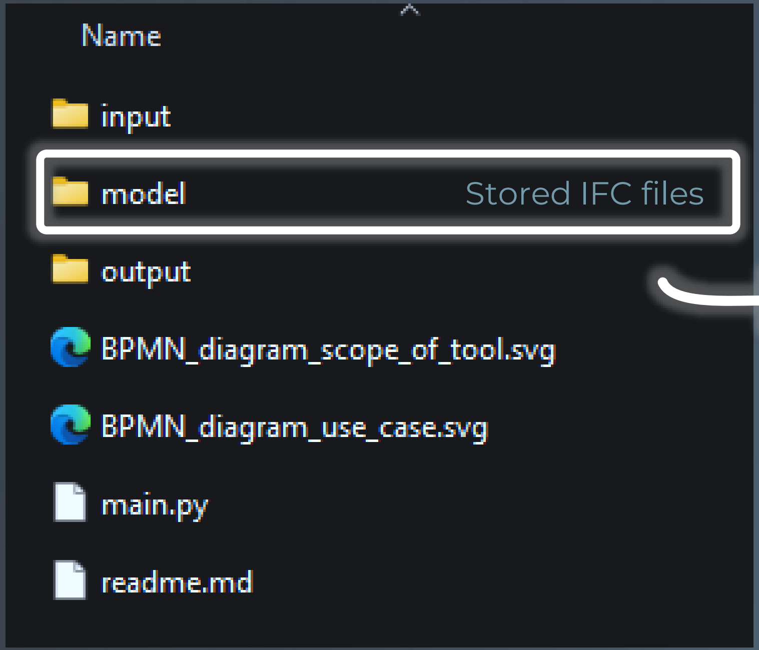
Name:

		Columns		**Walls**		**Beams**		**Slabs**	
No.	Floor	Volume, m3	Area, m2	Volume, m3	Area, m2	Volume, m3	Area, m2	Volume, m3	Area, m2
	Total:								

for optimal results, it's advisable to keep the template unchanged.

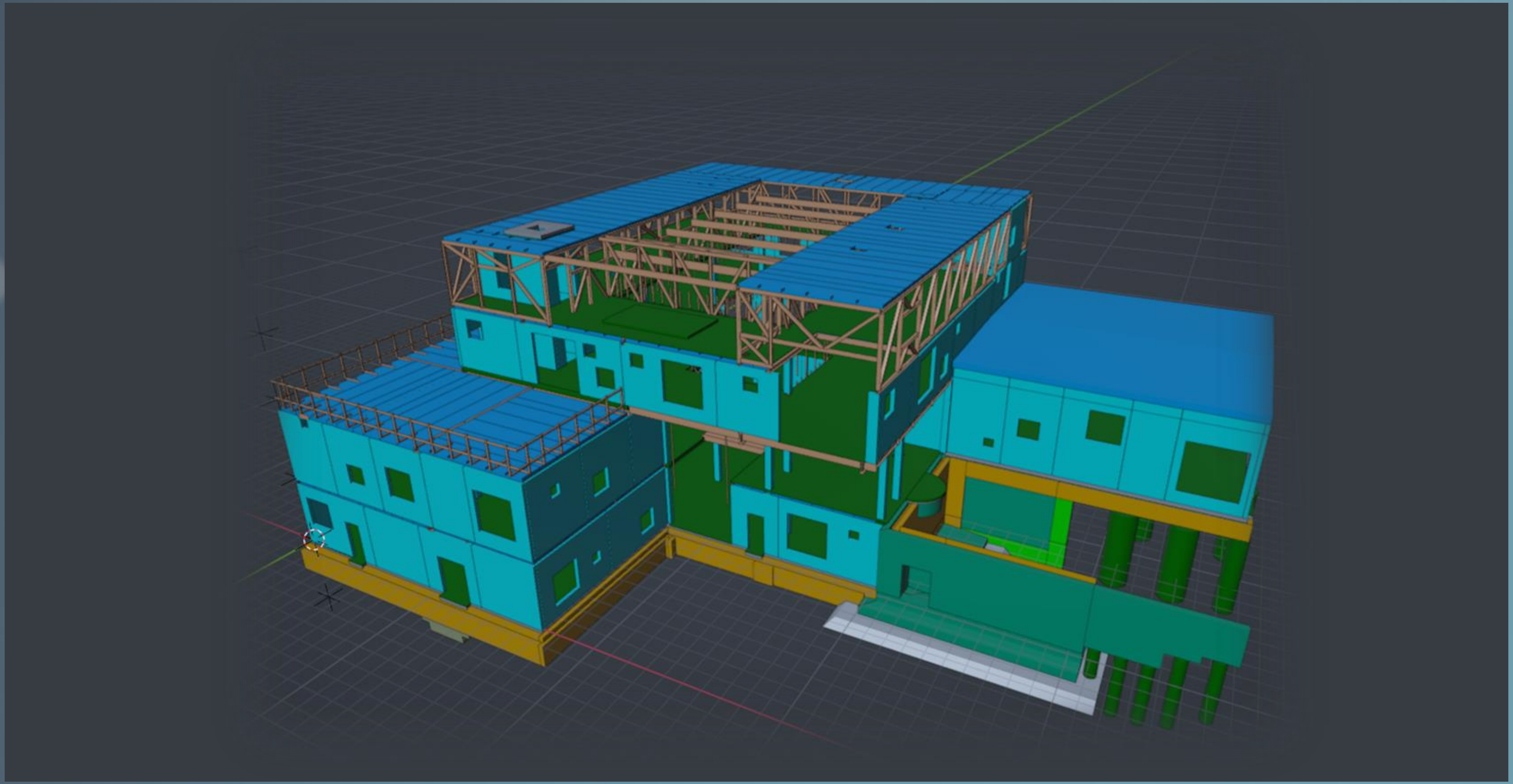
The input folder contains a cost estimation report template used during the tool's operation. This document is formatted in a specific way to remain undistorted when populated with findings from the IFC file. Therefore, for optimal results it's advisable to keep the template unchanged.

STEP 6 – COST ESTIMATION TOOL



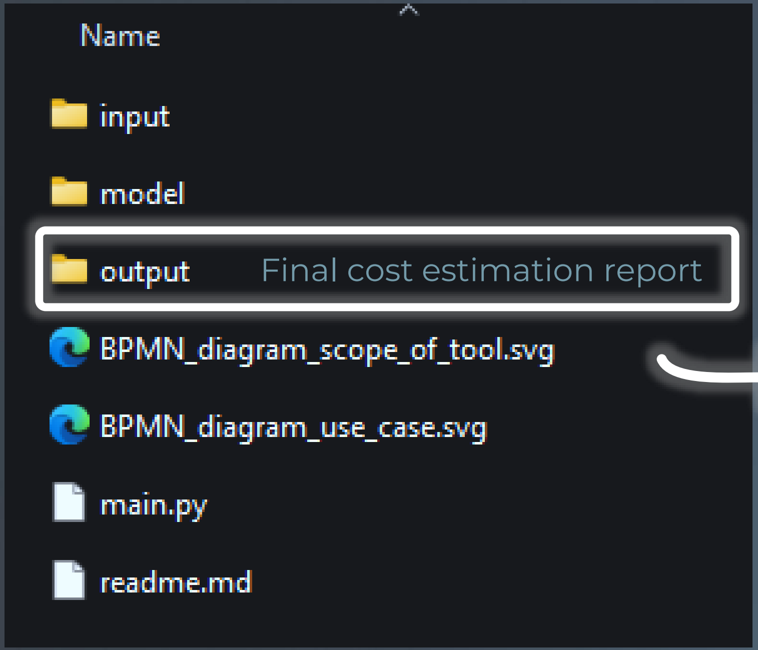
Name

- input
- model — Stored IFC files
- output
- BPMN_diagram_scope_of_tool.svg
- BPMN_diagram_use_case.svg
- main.py
- readme.md



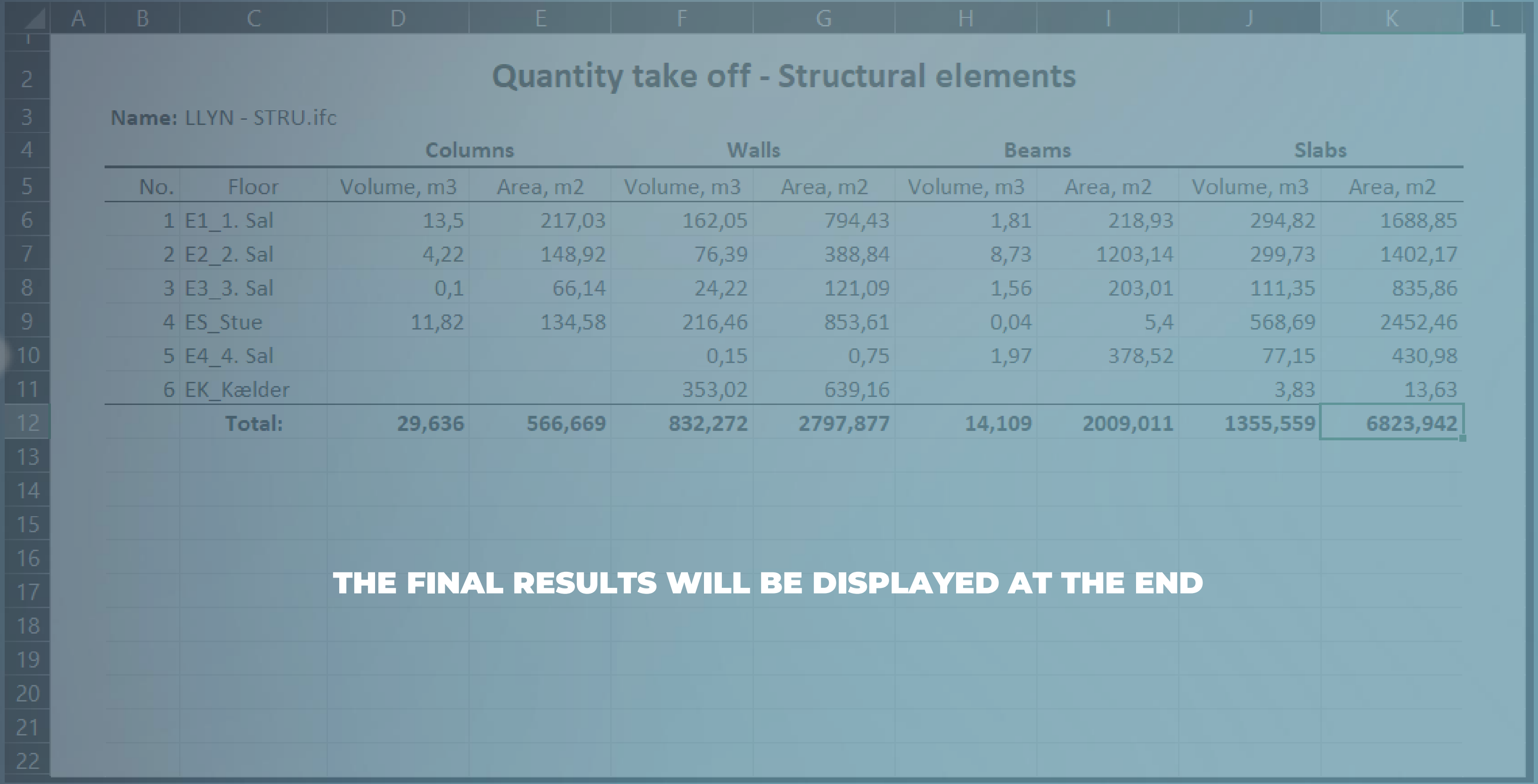
The 'model' folder is designated for storing the IFC files you plan to process using this tool. The script will search for project files in this folder based on the provided filename during the operation. For showcase purposes, the structural model of SkyLab has been pre-uploaded to this folder.

STEP 7 – COST ESTIMATION TOOL



Name

- input
- model
- output Final cost estimation report
- BPMN_diagram_scope_of_tool.svg
- BPMN_diagram_use_case.svg
- main.py
- readme.md



Quantity take off - Structural elements

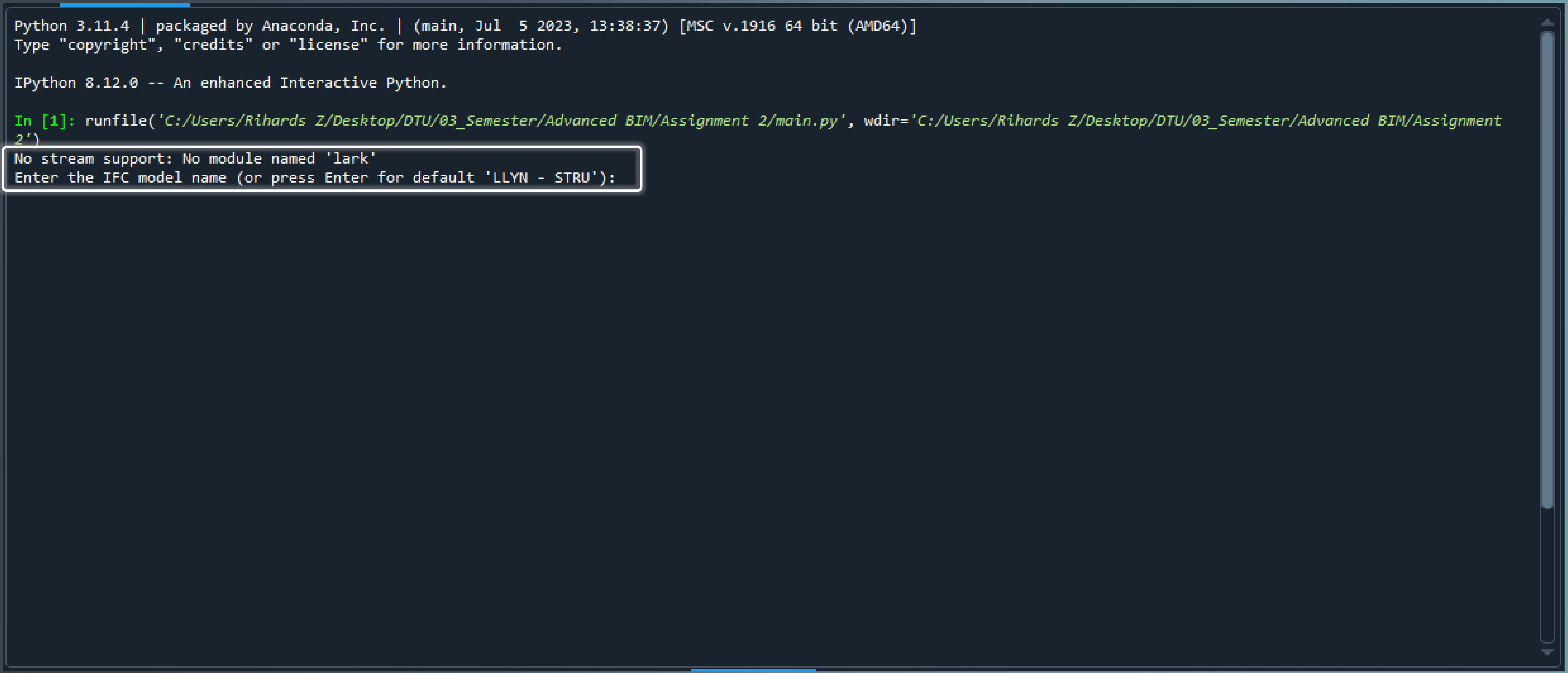
Name: LLYN - STRU.ifc

		Columns		Walls		Beams		Slabs	
No.	Floor	Volume, m3	Area, m2	Volume, m3	Area, m2	Volume, m3	Area, m2	Volume, m3	Area, m2
1	E1_1. Sal	13,5	217,03	162,05	794,43	1,81	218,93	294,82	1688,85
2	E2_2. Sal	4,22	148,92	76,39	388,84	8,73	1203,14	299,73	1402,17
3	E3_3. Sal	0,1	66,14	24,22	121,09	1,56	203,01	111,35	835,86
4	ES_Stue	11,82	134,58	216,46	853,61	0,04	5,4	568,69	2452,46
5	E4_4. Sal			0,15	0,75	1,97	378,52	77,15	430,98
6	EK_Kælder			353,02	639,16			3,83	13,63
	Total:	**29,636**	**566,669**	**832,272**	**2797,877**	**14,109**	**2009,011**	**1355,559**	**6823,942**

THE FINAL RESULTS WILL BE DISPLAYED AT THE END

As suggested by its name, the 'output' folder is where the final cost estimation report will be stored once the script has completed its operations.

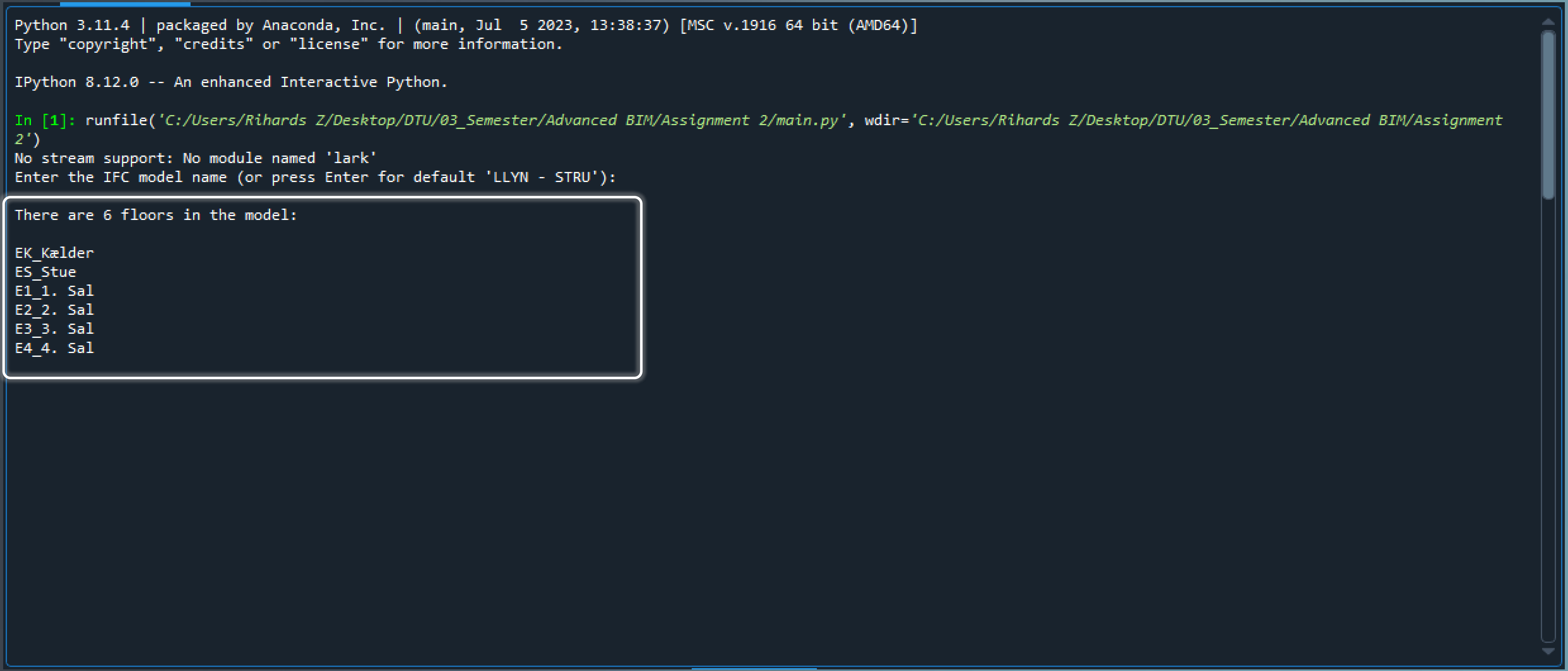
STEP 8 – COST ESTIMATION TOOL



```
Python 3.11.4 | packaged by Anaconda, Inc. | (main, Jul  5 2023, 13:38:37) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.12.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Rihards Z/Desktop/DTU/03_Semester/Advanced BIM/Assignment 2/main.py', wdir='C:/Users/Rihards Z/Desktop/DTU/03_Semester/Advanced BIM/Assignment
2')
No stream support: No module named 'lark'
Enter the IFC model name (or press Enter for default 'LLYN - STRU'):
```

When you execute the 'main.py' script, it will prompt you to enter the filename. Leaving it blank will load the default preset, which is the structural export of the SkyLab model.

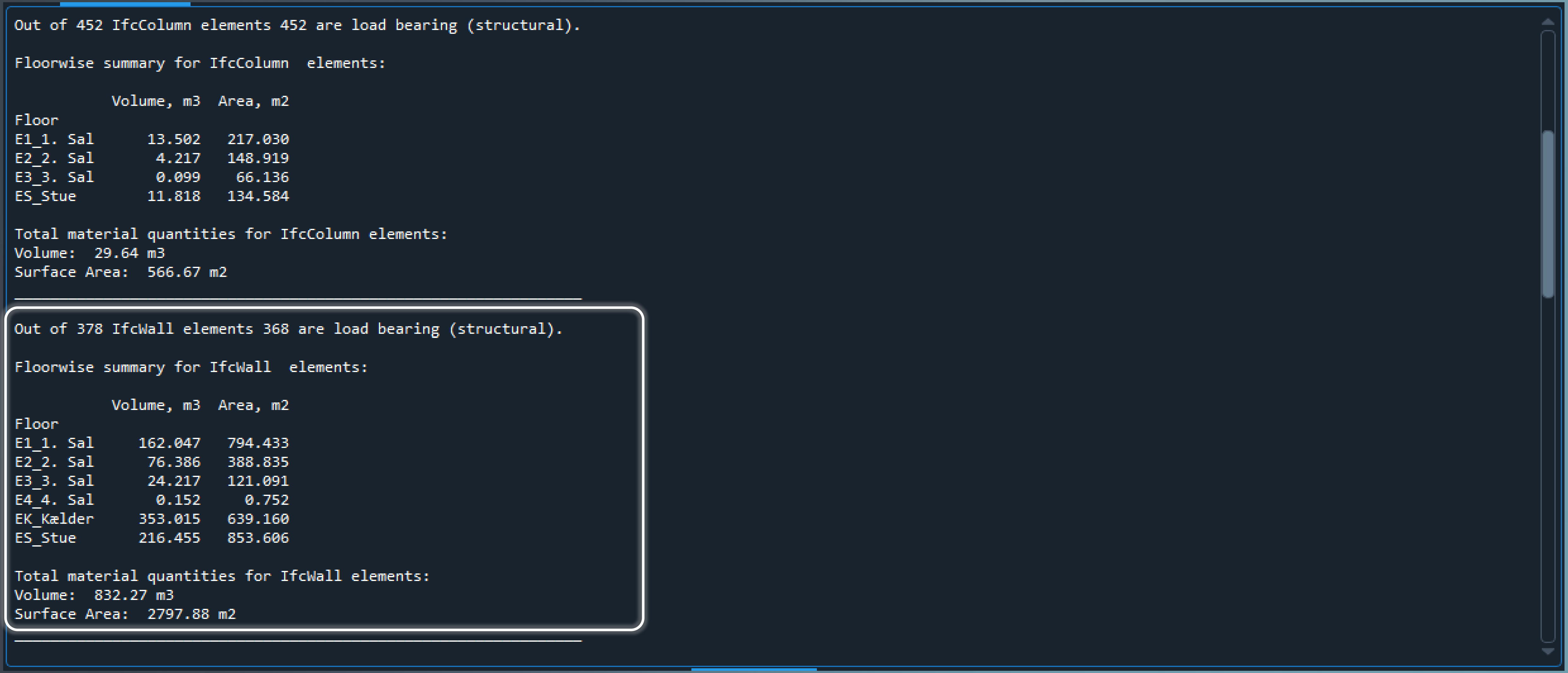
STEP 9 – COST ESTIMATION TOOL



```
Python 3.11.4 | packaged by Anaconda, Inc. | (main, Jul  5 2023, 13:38:37) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.12.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Rihards Z/Desktop/DTU/03_Semester/Advanced BIM/Assignment 2/main.py', wdir='C:/Users/Rihards Z/Desktop/DTU/03_Semester/Advanced BIM/Assignment
2')
No stream support: No module named 'lark'
Enter the IFC model name (or press Enter for default 'LLYN - STRU'):

There are 6 floors in the model:

EK_Kælder
ES_Stue
E1_1. Sal
E2_2. Sal
E3_3. Sal
E4_4. Sal
```

If a file with a matching name is found and successfully loaded, the tool will begin its operations. Initially, it will report the number and names of the stories present in the IFC file. Then, the tool will iterate through all the available structural elements, namely columns, load-bearing walls, beams, and slabs. The script will extract element quantities - volume and surface area, which are instrumental for accurate price estimations.

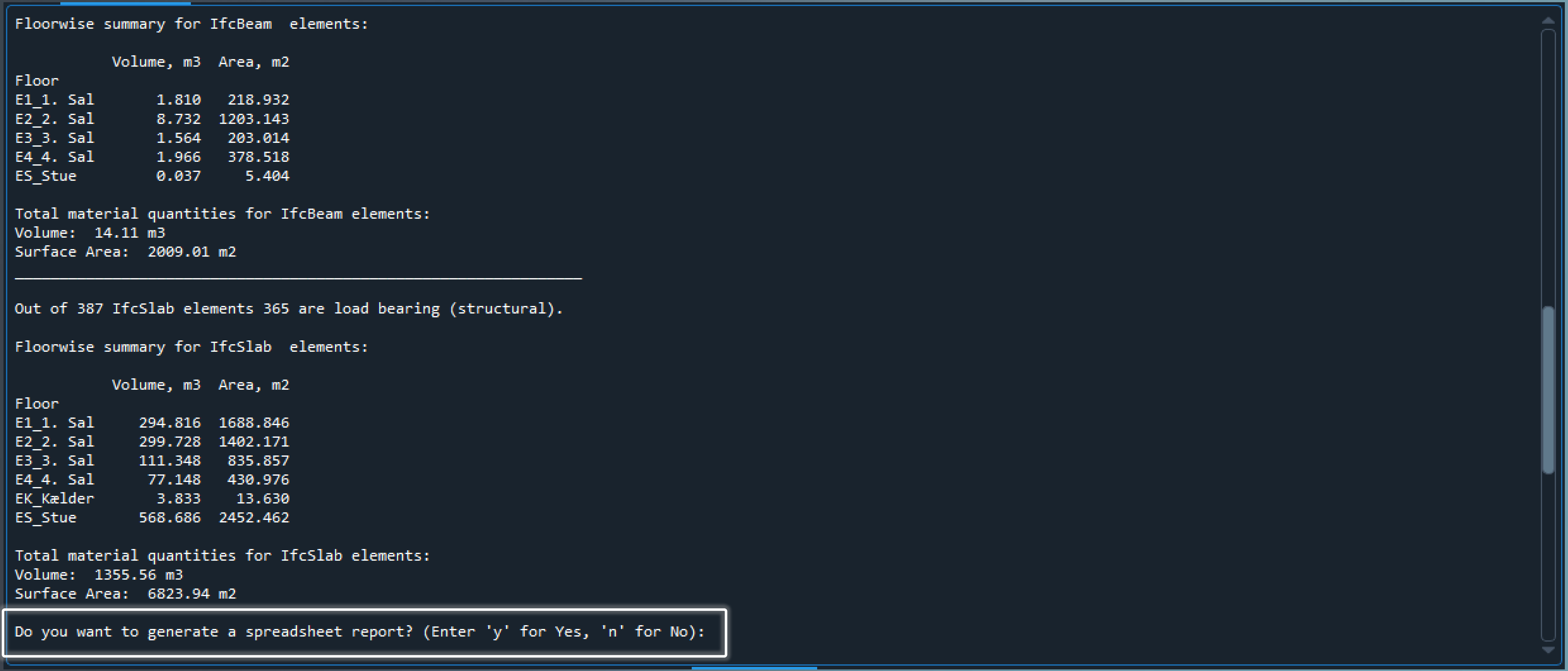
STEP 10 – COST ESTIMATION TOOL



```
Out of 452 IfcColumn elements 452 are load bearing (structural).

Floorwise summary for IfcColumn  elements:

           Volume, m3  Area, m2
Floor
E1_1. Sal      13.502   217.030
E2_2. Sal       4.217   148.919
E3_3. Sal       0.099    66.136
ES_Stue        11.818   134.584

Total material quantities for IfcColumn elements:
Volume:  29.64 m3
Surface Area:  566.67 m2

Out of 378 IfcWall elements 368 are load bearing (structural).

Floorwise summary for IfcWall  elements:

           Volume, m3  Area, m2
Floor
E1_1. Sal     162.047   794.433
E2_2. Sal      76.386   388.835
E3_3. Sal      24.217   121.091
E4_4. Sal       0.152     0.752
EK_Kælder     353.015   639.160
ES_Stue       216.455   853.606

Total material quantities for IfcWall elements:
Volume:  832.27 m3
Surface Area:  2797.88 m2
```

As the script progresses, a preliminary floorwise report will be directly displayed in the Python IDE. This report will showcase quantities of each structural element type by floor and in total.

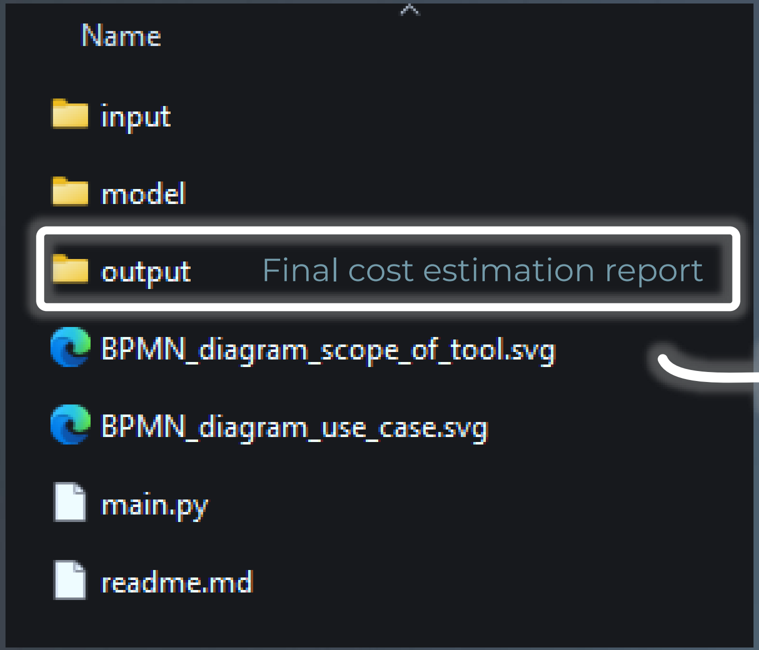
STEP 11 – COST ESTIMATION TOOL



```
Floorwise summary for IfcBeam  elements:

           Volume, m3  Area, m2
Floor
E1_1. Sal       1.810   218.932
E2_2. Sal       8.732  1203.143
E3_3. Sal       1.564   203.014
E4_4. Sal       1.966   378.518
ES_Stue         0.037     5.404

Total material quantities for IfcBeam elements:
Volume:  14.11 m3
Surface Area:  2009.01 m2

________________________________________________________________

Out of 387 IfcSlab elements 365 are load bearing (structural).

Floorwise summary for IfcSlab  elements:

           Volume, m3  Area, m2
Floor
E1_1. Sal     294.816  1688.846
E2_2. Sal     299.728  1402.171
E3_3. Sal     111.348   835.857
E4_4. Sal      77.148   430.976
EK_Kælder       3.833    13.630
ES_Stue       568.686  2452.462

Total material quantities for IfcSlab elements:
Volume:  1355.56 m3
Surface Area:  6823.94 m2

Do you want to generate a spreadsheet report? (Enter 'y' for Yes, 'n' for No):
```

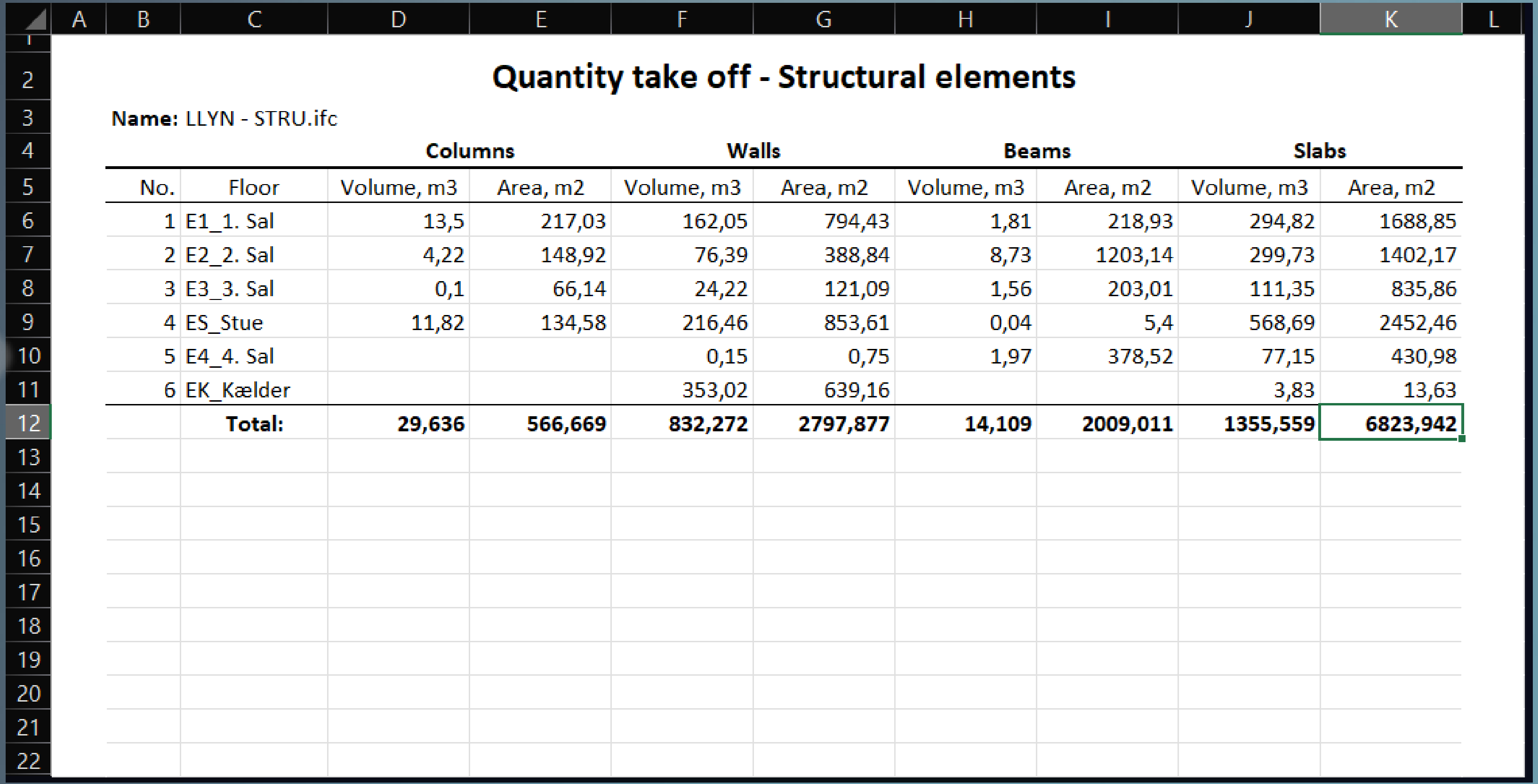
After the operations conclude, the tool will prompt you to generate a quantity takeoff report, which, as mentioned earlier, will be stored in the 'output' folder.

STEP 12 – COST ESTIMATION TOOL



Name

- input
- model
- output — Final cost estimation report
- BPMN_diagram_scope_of_tool.svg
- BPMN_diagram_use_case.svg
- main.py
- readme.md



Quantity take off - Structural elements

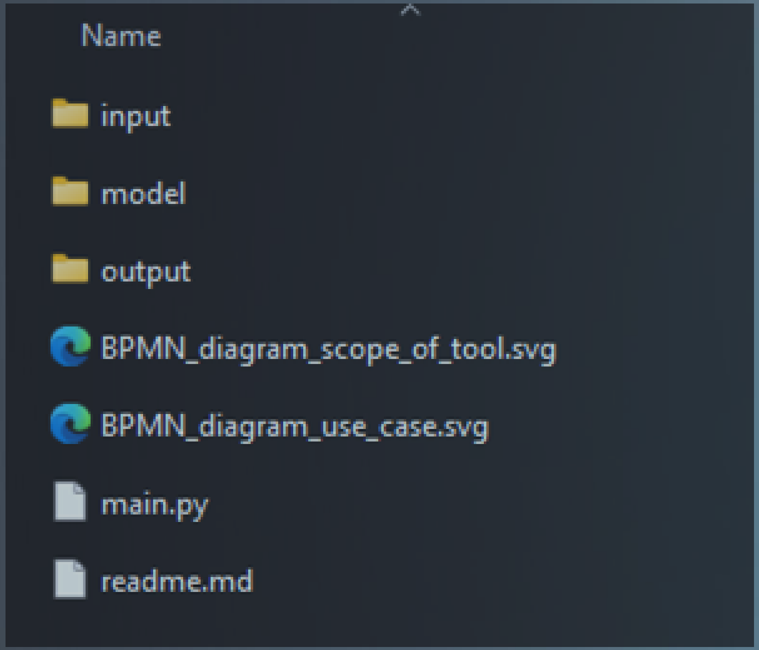
Name: LLYN - STRU.ifc

		Columns		Walls		Beams		Slabs	
No.	Floor	Volume, m3	Area, m2	Volume, m3	Area, m2	Volume, m3	Area, m2	Volume, m3	Area, m2
1	E1_1. Sal	13,5	217,03	162,05	794,43	1,81	218,93	294,82	1688,85
2	E2_2. Sal	4,22	148,92	76,39	388,84	8,73	1203,14	299,73	1402,17
3	E3_3. Sal	0,1	66,14	24,22	121,09	1,56	203,01	111,35	835,86
4	ES_Stue	11,82	134,58	216,46	853,61	0,04	5,4	568,69	2452,46
5	E4_4. Sal			0,15	0,75	1,97	378,52	77,15	430,98
6	EK_Kælder			353,02	639,16			3,83	13,63
	Total:	**29,636**	**566,669**	**832,272**	**2797,877**	**14,109**	**2009,011**	**1355,559**	**6823,942**

In its current state, the tool generates a report documenting the quantities of each structural element by floor. However, the next step in its development will involve adding another input document - a price list. This addition will seamlessly allow for quick price estimations.

STEP 13 – STRUCTURAL PROPERTY TOOL

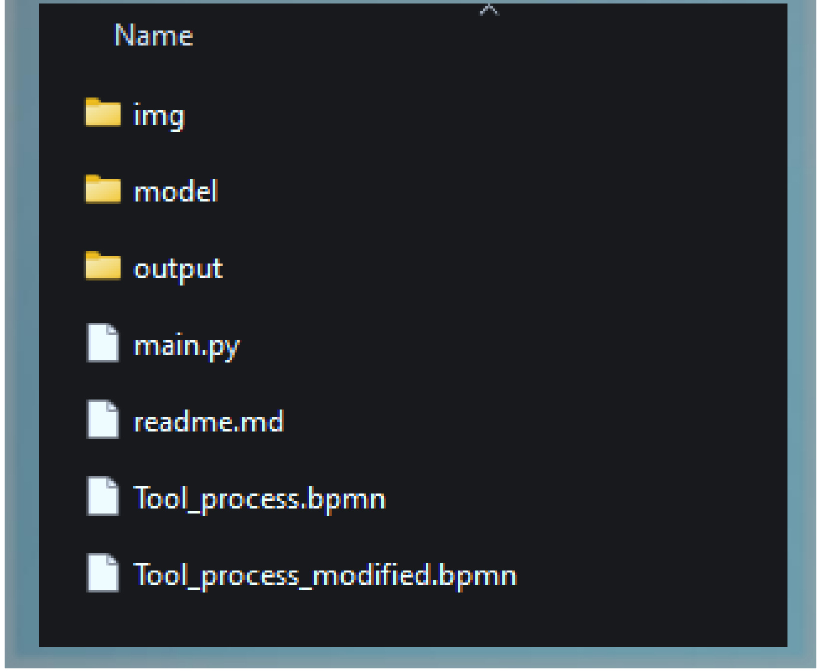
A2



Name

- input
- model
- output
- BPMN_diagram_scope_of_tool.svg
- BPMN_diagram_use_case.svg
- main.py
- readme.md

A3

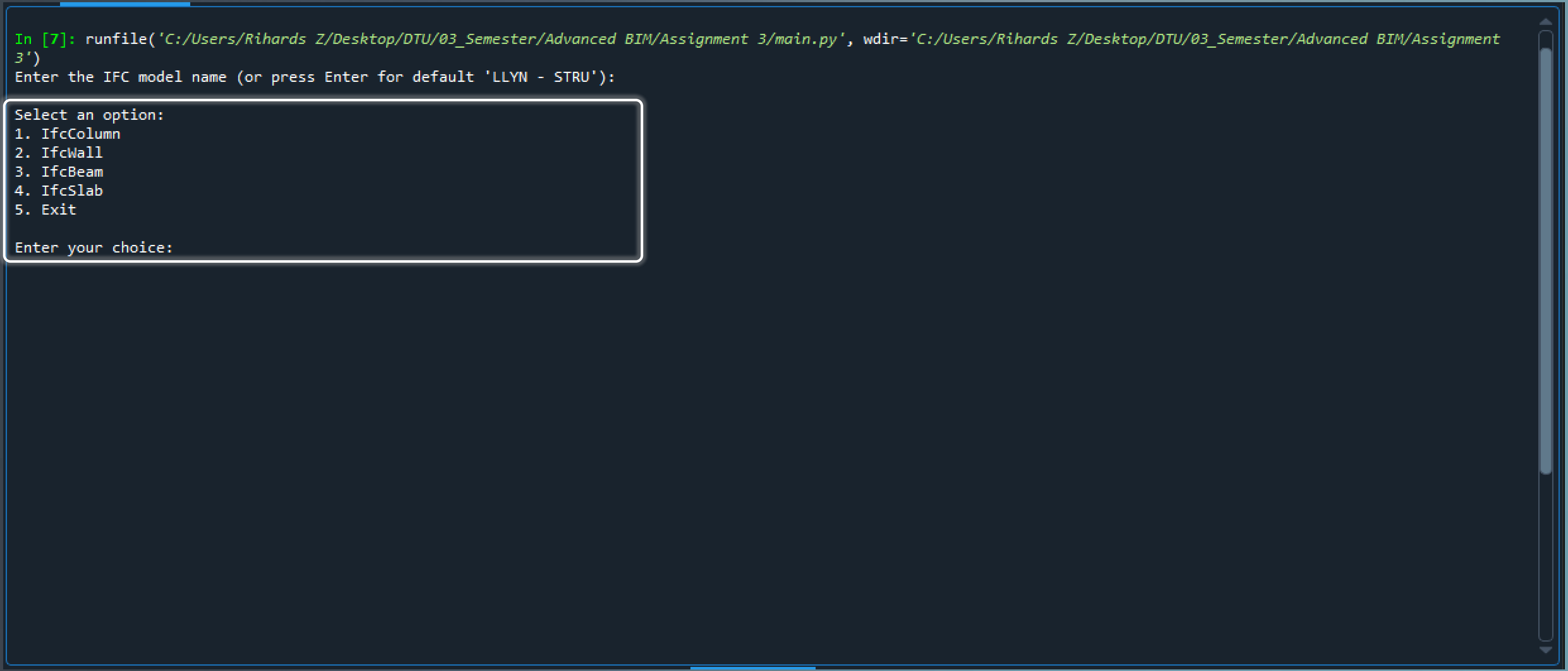


Name

- img
- model
- output
- main.py
- readme.md
- Tool_process.bpmn
- Tool_process_modified.bpmn

Now lets have a quick look at the other tool. The folder hierarchy follows the same principle as for the other tool, including the BPMN diagrams for a general overview of the tools processes.

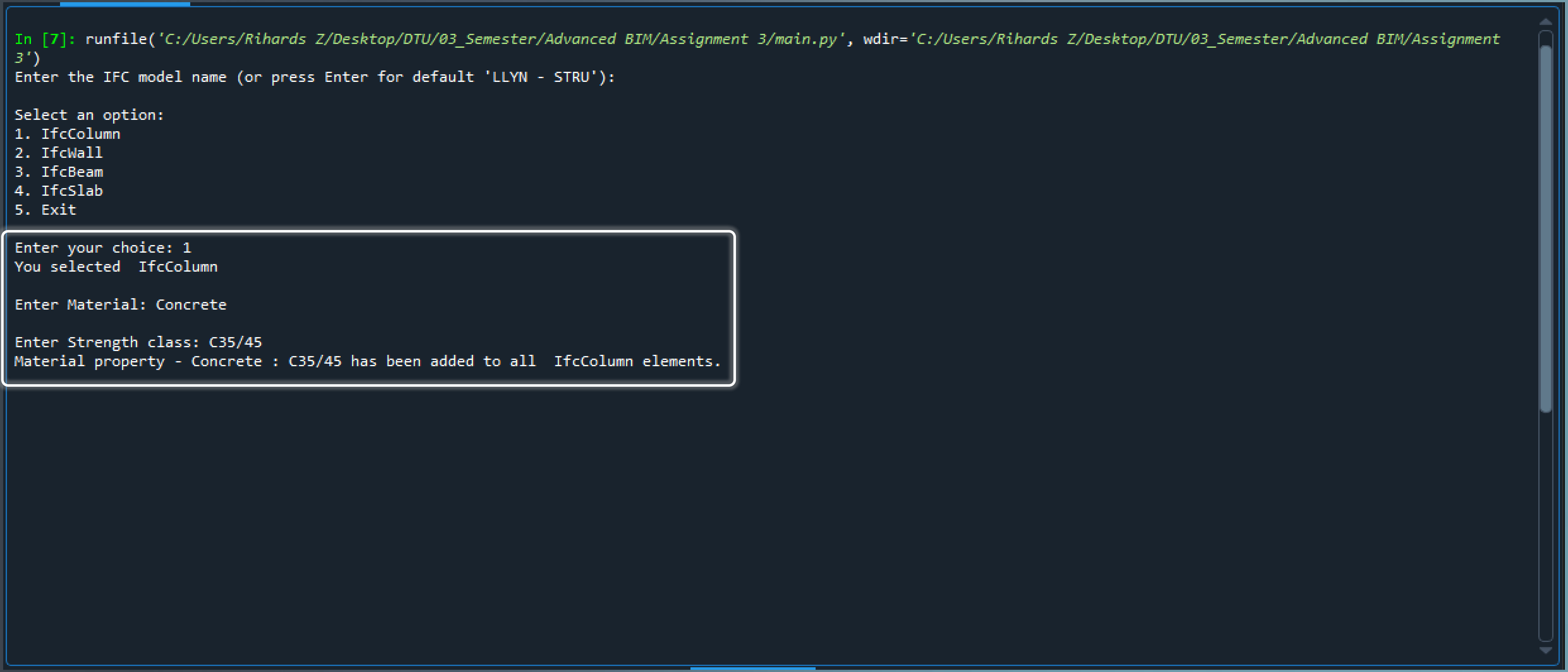
STEP 14 – STRUCTURAL PROPERTY TOOL



```
In [7]: runfile('C:/Users/Rihards Z/Desktop/DTU/03_Semester/Advanced BIM/Assignment 3/main.py', wdir='C:/Users/Rihards Z/Desktop/DTU/03_Semester/Advanced BIM/Assignment
3')
Enter the IFC model name (or press Enter for default 'LLYN - STRU'):

Select an option:
1. IfcColumn
2. IfcWall
3. IfcBeam
4. IfcSlab
5. Exit

Enter your choice:
```

When you execute the 'main.py' script for this tool, you encounter the same prompt as the previous tool – to enter the filename. However, this time, upon successful loading, a new menu appears. It enables you to select a type of structural element and create a new PropertySet for it. This is particularly useful because material properties are frequently missing from a structural IFC model export but are commonly found in an architectural model. Therefore, this tool facilitates a quick solution for correcting this error without requiring an entirely new export, which could potentially cause delays if executed by a third party.

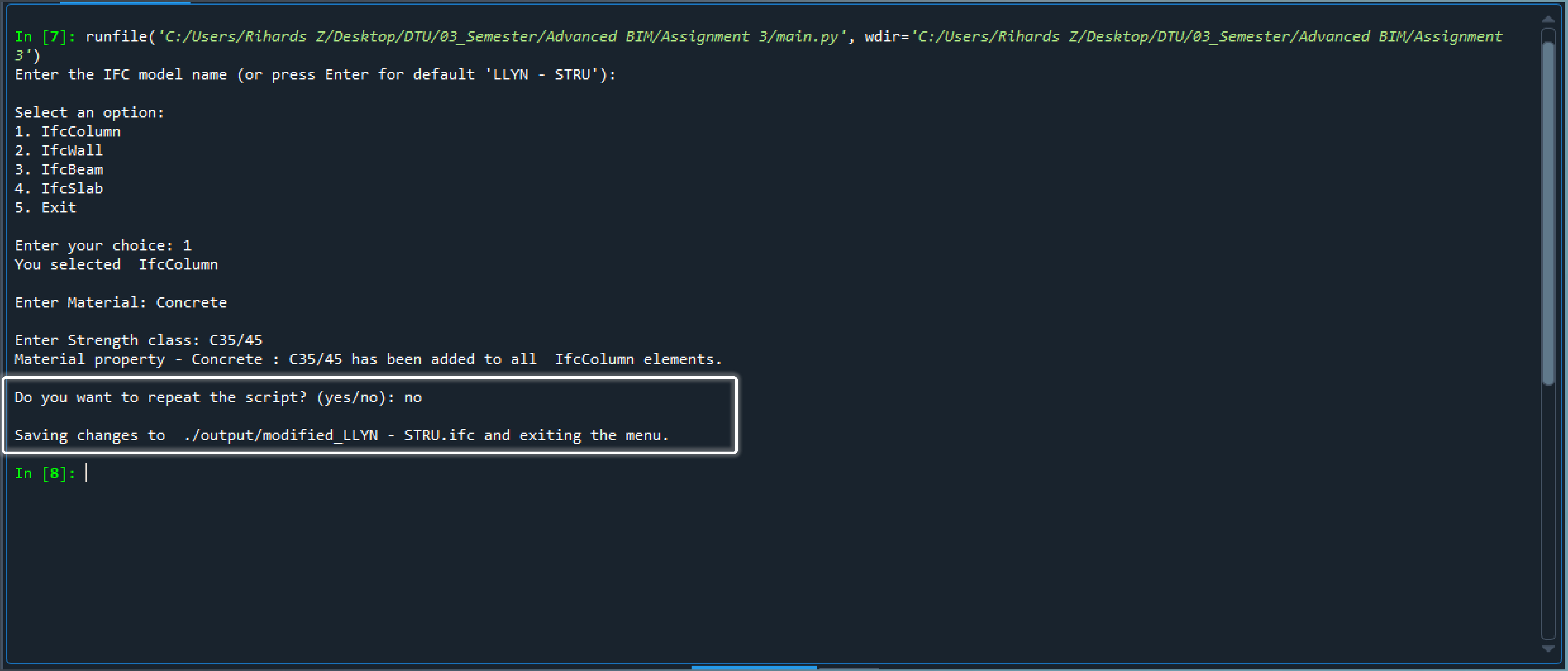
STEP 15 – STRUCTURAL PROPERTY TOOL



```
In [7]: runfile('C:/Users/Rihards Z/Desktop/DTU/03_Semester/Advanced BIM/Assignment 3/main.py', wdir='C:/Users/Rihards Z/Desktop/DTU/03_Semester/Advanced BIM/Assignment
3')
Enter the IFC model name (or press Enter for default 'LLYN - STRU'):

Select an option:
1. IfcColumn
2. IfcWall
3. IfcBeam
4. IfcSlab
5. Exit

Enter your choice: 1
You selected  IfcColumn

Enter Material: Concrete

Enter Strength class: C35/45
Material property - Concrete : C35/45 has been added to all  IfcColumn elements.
```

In this particular example, after choosing an element type, the script will ask you to input a material type and a strength class. Once these values are provided, a new PropertySet is generated and assigned to all elements of the selected type.

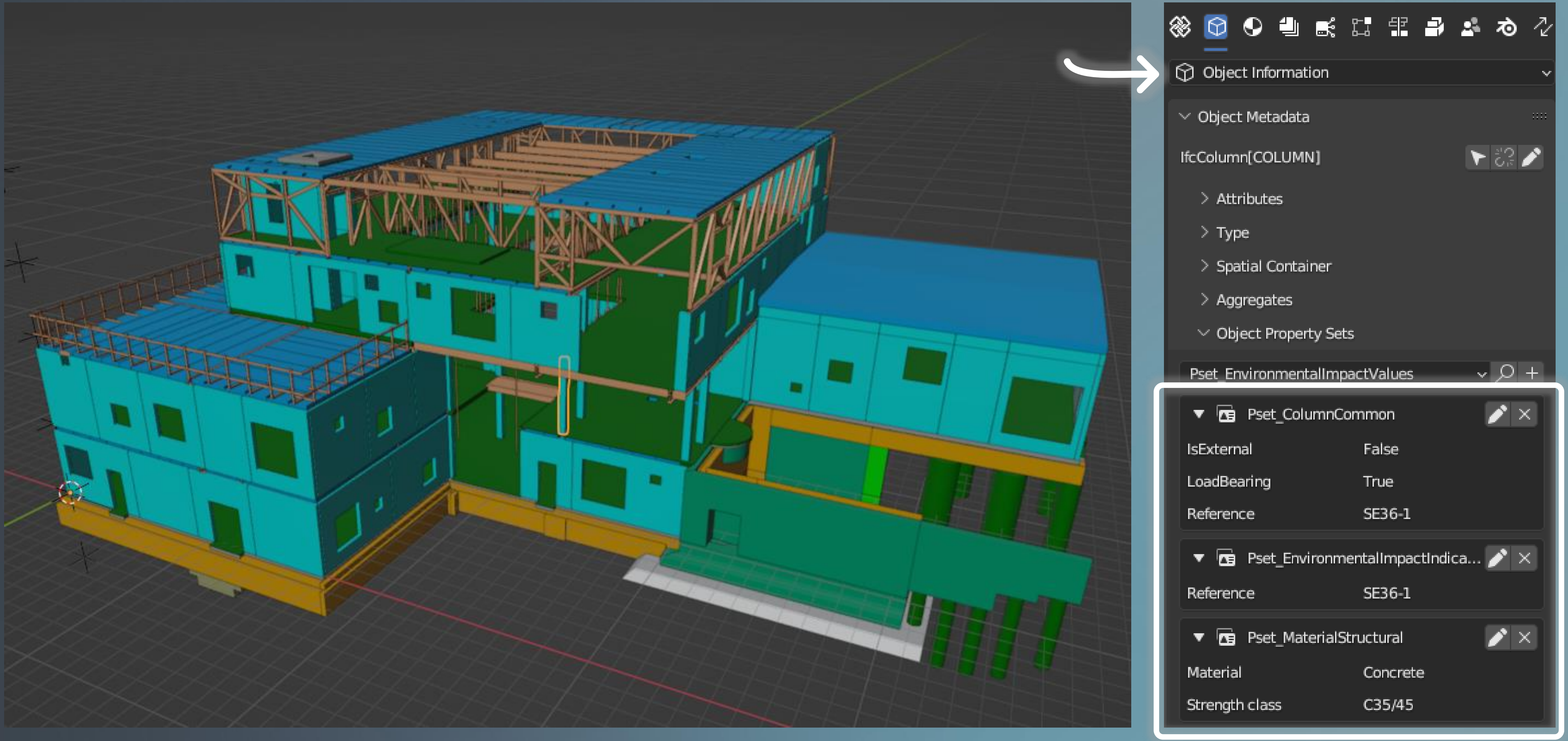
STEP 16 – STRUCTURAL PROPERTY TOOL



```
In [7]: runfile('C:/Users/Rihards Z/Desktop/DTU/03_Semester/Advanced BIM/Assignment 3/main.py', wdir='C:/Users/Rihards Z/Desktop/DTU/03_Semester/Advanced BIM/Assignment
3')
Enter the IFC model name (or press Enter for default 'LLYN - STRU'):

Select an option:
1. IfcColumn
2. IfcWall
3. IfcBeam
4. IfcSlab
5. Exit

Enter your choice: 1
You selected  IfcColumn

Enter Material: Concrete

Enter Strength class: C35/45
Material property - Concrete : C35/45 has been added to all  IfcColumn elements.

Do you want to repeat the script? (yes/no): no

Saving changes to  ./output/modified_LLYN - STRU.ifc and exiting the menu.

In [8]: 
```

If at any point you wish to stop the IFC modifications, you can either choose 'no' when prompted to repeat the element type selection menu or select 'exit' from the menu. This action will lead the script to save the modified IFC file to the ‘output’ folder and exit the tool.

STEP 17 – STRUCTURAL PROPERTY TOOL



IfcColumn[COLUMN]

Pset_ColumnCommon

IsExternal	False
LoadBearing	True
Reference	SE36-1

Pset_EnvironmentalImpactIndica...

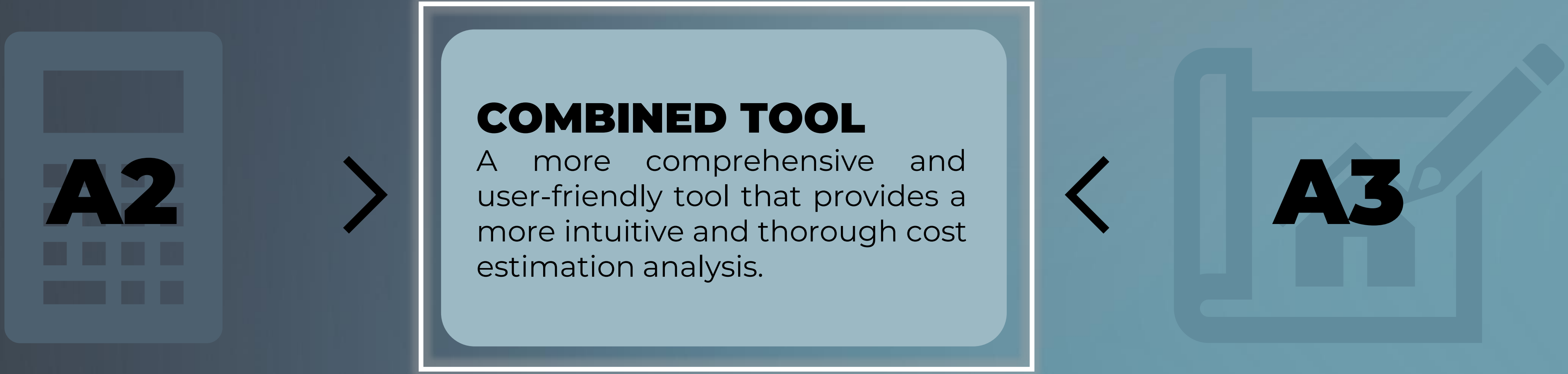
Reference	SE36-1

Pset_MaterialStructural

Material	Concrete
Strength class	C35/45

After loading the modified IFC file into BlenderBIM, you can verify the addition of the newly generated features under the Object Information tab.

STEP 18 – STRUCTURAL PROPERTY TOOL

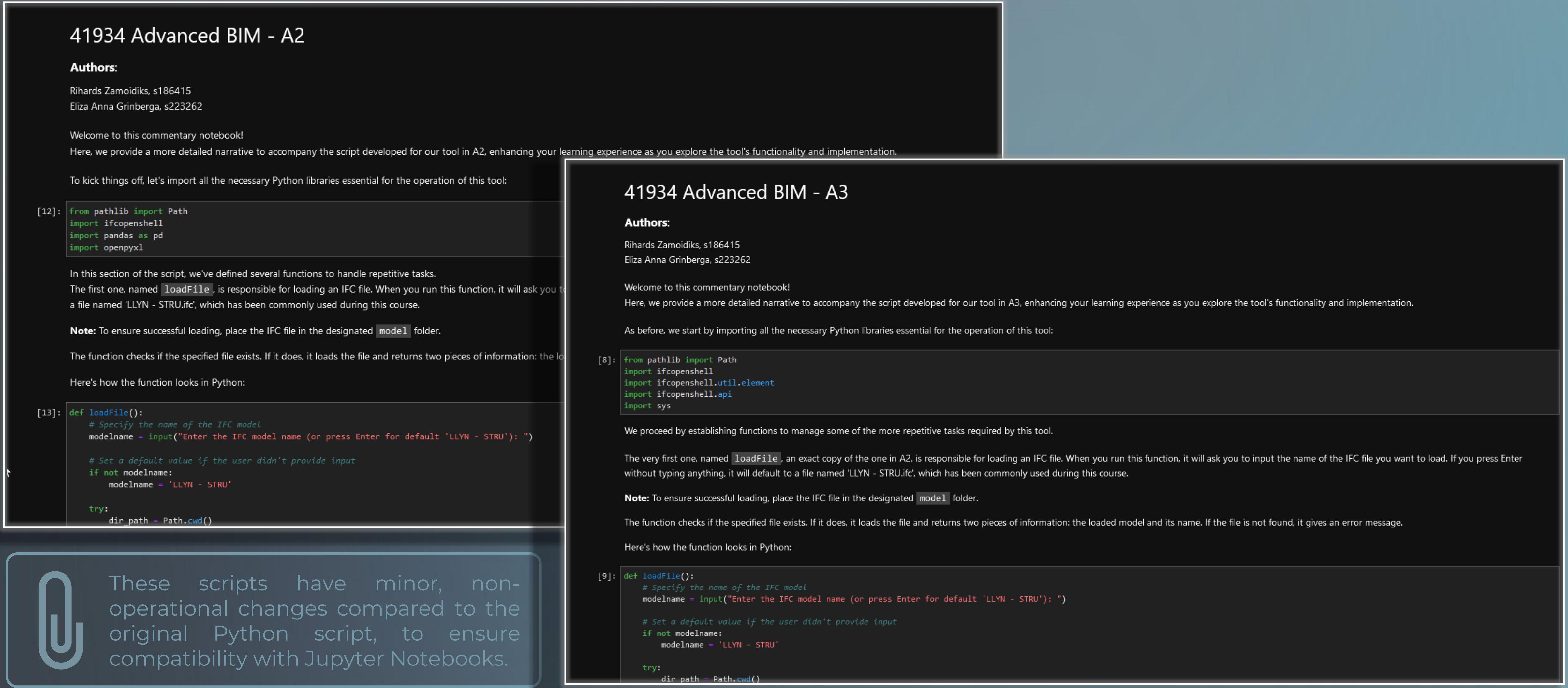


COMBINED TOOL

A more comprehensive and user-friendly tool that provides a more intuitive and thorough cost estimation analysis.

While these tools were designed to address different OpenBIM aspects and were thus developed independently of each other, the second tool aims to enhance the functionality of the first. The end goal is to create a more comprehensive and user-friendly tool that provides a more intuitive and thorough cost estimation analysis. The next phase of development involves integrating both tools into one – a more advanced solution.

STEP 18



41934 Advanced BIM - A2

Authors:

Rihards Zamoidiks, s186415
Eliza Anna Grinberga, s223262

Welcome to this commentary notebook!
Here, we provide a more detailed narrative to accompany the script developed for our tool in A2, enhancing your learning experience as you explore the tool's functionality and implementation.

To kick things off, let's import all the necessary Python libraries essential for the operation of this tool:

```
[12]: from pathlib import Path
      import ifcopenshell
      import pandas as pd
      import openpyxl
```

In this section of the script, we've defined several functions to handle repetitive tasks.
The first one, named `loadFile`, is responsible for loading an IFC file. When you run this function, it will ask you t
a file named 'LLYN - STRU.ifc', which has been commonly used during this course.

Note: To ensure successful loading, place the IFC file in the designated `model` folder.

The function checks if the specified file exists. If it does, it loads the file and returns two pieces of information: the lo

Here's how the function looks in Python:

```
[13]: def loadFile():
          # Specify the name of the IFC model
          modelname = input("Enter the IFC model name (or press Enter for default 'LLYN - STRU'): ")

          # Set a default value if the user didn't provide input
          if not modelname:
              modelname = 'LLYN - STRU'

          try:
              dir_path = Path.cwd()
```

41934 Advanced BIM - A3

Authors:

Rihards Zamoidiks, s186415
Eliza Anna Grinberga, s223262

Welcome to this commentary notebook!
Here, we provide a more detailed narrative to accompany the script developed for our tool in A3, enhancing your learning experience as you explore the tool's functionality and implementation.

As before, we start by importing all the necessary Python libraries essential for the operation of this tool:

```
[8]: from pathlib import Path
     import ifcopenshell
     import ifcopenshell.util.element
     import ifcopenshell.api
     import sys
```

We proceed by establishing functions to manage some of the more repetitive tasks required by this tool.

The very first one, named `loadFile`, an exact copy of the one in A2, is responsible for loading an IFC file. When you run this function, it will ask you to input the name of the IFC file you want to load. If you press Enter without typing anything, it will default to a file named 'LLYN - STRU.ifc', which has been commonly used during this course.

Note: To ensure successful loading, place the IFC file in the designated `model` folder.

The function checks if the specified file exists. If it does, it loads the file and returns two pieces of information: the loaded model and its name. If the file is not found, it gives an error message.

Here's how the function looks in Python:

```
[9]: def loadFile():
         # Specify the name of the IFC model
         modelname = input("Enter the IFC model name (or press Enter for default 'LLYN - STRU'): ")

         # Set a default value if the user didn't provide input
         if not modelname:
             modelname = 'LLYN - STRU'

         try:
             dir_path = Path.cwd()
```

These scripts have minor, non-operational changes compared to the original Python script, to ensure compatibility with Jupyter Notebooks.

And with that, we conclude our brief overview of our OpenBIM tools. To gain a deeper understanding of the OpenBIM concepts, we encourage exploring the provided Python scripts and becoming familiar with the ifcopenshell library which is instrumental for use in OpenBIM. For an enhanced learning experience, the project repository contains a Jupyter Notebook file, offering more comprehensive explanations than the plain comments within the script.

GOOD LUCK!