

Concrete Compressive Strength Prediction

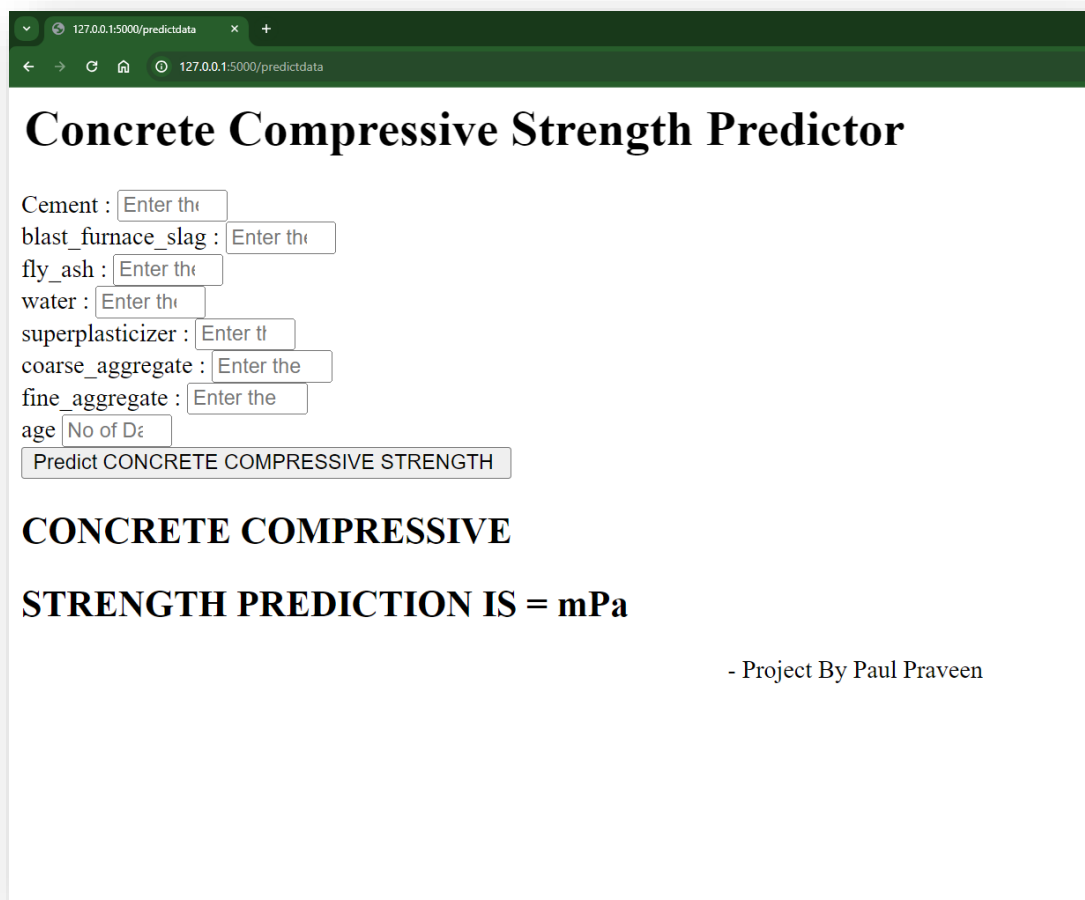
Wireframe Documentation

PAUL PRAVEEN

User Interface of the Project

User Interface or the Frontend of the project is created with the help of HTML and CSS.

The Flask Framework is used to integrate frontend with the backend.



The screenshot shows a web browser window with the URL `127.0.0.1:5000/predictdata`. The page title is "Concrete Compressive Strength Predictor". The form contains the following fields and labels:

- Cement :
- blast_furnace_slag :
- fly_ash :
- water :
- superplasticizer :
- coarse_aggregate :
- fine_aggregate :
- age :

Below the fields is a button labeled "Predict CONCRETE COMPRESSIVE STRENGTH".

Below the button, the text "CONCRETE COMPRESSIVE STRENGTH PREDICTION IS = mPa" is displayed.

At the bottom right, the text "- Project By Paul Praveen" is shown.

Home Page

This is the Homepage of the Project.

To predict the Compressive Strength, you must enter all the field values.

[*Here all the field are required]

Concrete Compressive Strength Predictor

Cement : 540
blast_furnace_slag : 142
fly_ash : 5
water : 228
superplasticizer : 2.5
coarse_aggregate : 1055
fine_aggregate : 676
age : 360

Predict CONCRETE COMPRESSIVE STRENGTH

CONCRETE COMPRESSIVE STRENGTH PREDICTION IS = mPa

- Project By Paul Praveen

Prediction Page

Concrete Compressive Strength Predictor

Cement : Enter the
blast_furnace_slag : Enter the
fly_ash : Enter the
water : Enter the
superplasticizer : Enter the
coarse_aggregate : Enter the
fine_aggregate : Enter the
age : No of Days

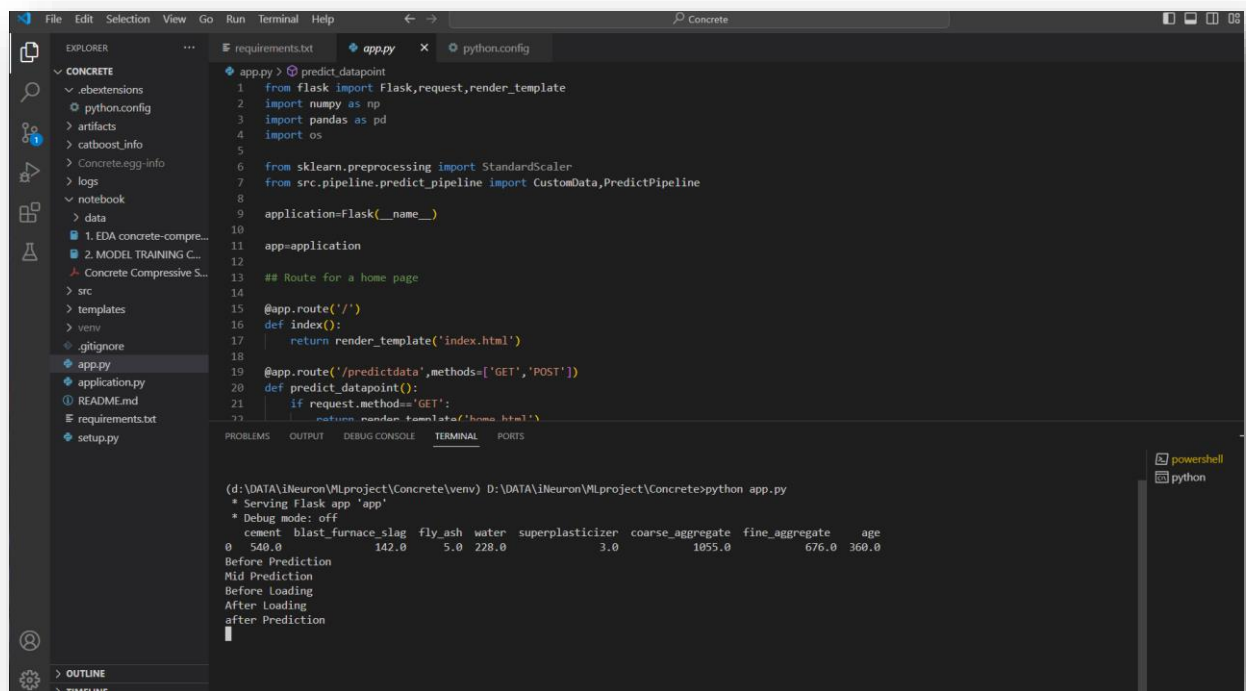
Predict CONCRETE COMPRESSIVE STRENGTH

CONCRETE COMPRESSIVE STRENGTH PREDICTION IS = 66.38167557642474 mPa

- Project By Paul Praveen

This is the prediction page of the project.

Here is the VS code screen during and after the deployment



The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** A file tree on the left showing the project structure. The file `app.py` is selected.
- EDITOR:** The main workspace showing the `app.py` file. The code is a Flask application that uses a machine learning model to predict concrete compressive strength.
- TERMINAL:** The bottom panel shows the output of the command `python app.py`. It displays the Flask app starting, the debug mode being off, and a table of input features and their corresponding predicted values.

```
1 from flask import Flask, request, render_template
2 import numpy as np
3 import pandas as pd
4 import os
5
6 from sklearn.preprocessing import StandardScaler
7 from src.pipeline.predict_pipeline import CustomData, PredictPipeline
8
9 application = Flask(__name__)
10
11 app = application
12
13 ## Route for a home page
14
15 @app.route('/')
16 def index():
17     return render_template('index.html')
18
19 @app.route('/predictdata', methods=['GET', 'POST'])
20 def predict_datapoint():
21     if request.method == 'GET':
22         return render_template('index.html')
```

Terminal Output:

```
(d:\DATA\iNeuron\MLproject\Concrete\venv) D:\DATA\iNeuron\MLproject\Concrete>python app.py
* Serving Flask app 'app'
* Debug mode: off
   cement  blast_furnace_slag  fly_ash  water  superplasticizer  coarse_aggregate  fine_aggregate  age
0    540.0             142.0      5.0   228.0                3.0             1055.0           676.0    360.0
Before Prediction
Mid Prediction
Before Loading
After Loading
After Prediction
```