# CSCB20
# Introduction to Databases and Web Application

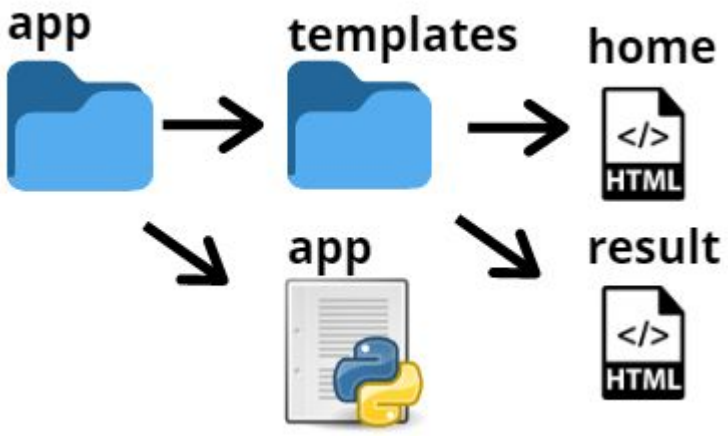
## Optional Python Project

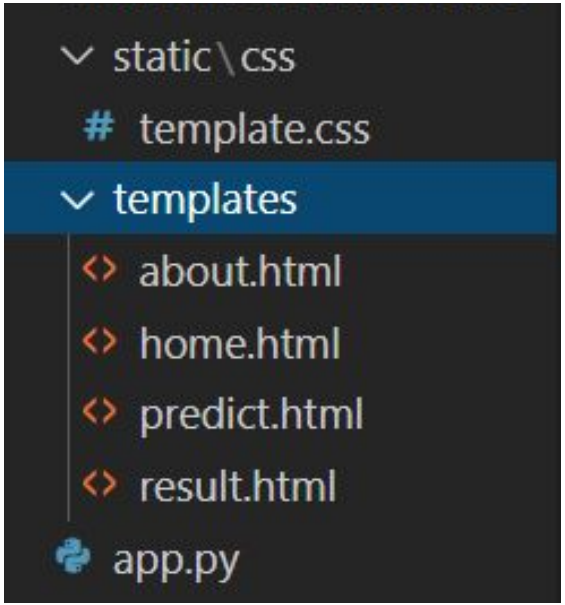
Dr. Purva R Gawde

# Optional Project to Try:
# Sentiment Analysis App using Flask

# Sentiment Analysis App

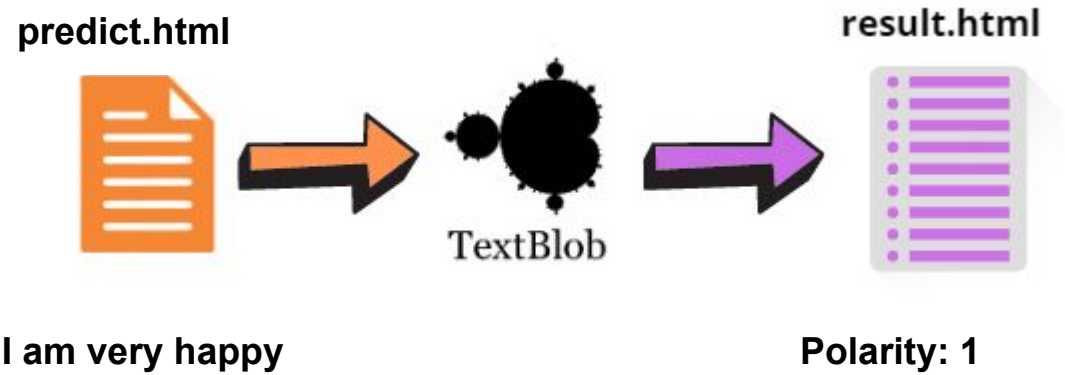
- We are going to build a simple sentiment analysis application and serve it, using flask.



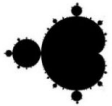
# Game plan

- Text input is captured via form on predict.html
- TextBlob detects the language of text input
- TextBlob results are deciphered by result.html



# TextBlob

Source: https://textblob.readthedocs.io/en/dev/




- TextBlob is a Python (2 and 3) library for processing textual data.

## Sentiment Analysis

The **`sentiment`** property returns a namedtuple of the form `Sentiment(polarity, subjectivity)`. The polarity score is a float within the range [-1.0, 1.0]. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective.

```
>>> testimonial = TextBlob("Textblob is amazingly simple to use. What great
>>> testimonial.sentiment
Sentiment(polarity=0.39166666666666666, subjectivity=0.4357142857142857)
>>> testimonial.sentiment.polarity
0.39166666666666666
```

# Flask HTTP methods

- Flask has different decorators to handle http requests. Http protocol is the basis for data communication in the World Wide Web.
- Different methods for retrieving data from a specified URL are defined in this protocol.

| Request | Purpose |
| --- | --- |
| GET | The most common method. A GET message is send, and the server returns data |
| POST | Used to send HTML form data to the server. The data received by the POST method is not cached by the server. |
| HEAD | Same as GET method, but no response body. |
| PUT | Replace all current representations of the target resource with uploaded content. |
| DELETE | Deletes all current representations of the target resource given by the URL. |

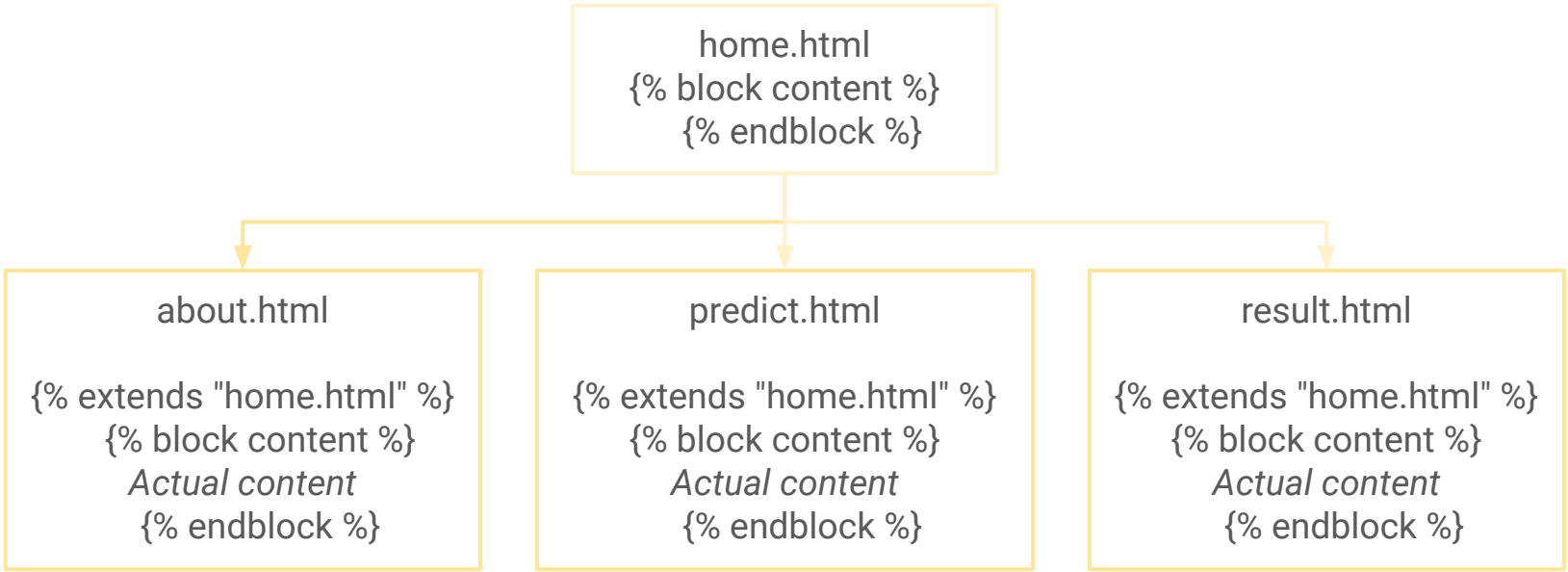
# app.py

```python
from flask import Flask, render_template, url_for, request
from textblob import TextBlob, Word, Blobber

app = Flask(__name__)

@app.route('/')
def home():
        return render_template('home.html')

@app.route("/about")
def about():
    return render_template("about.html")

@app.route('/predict',methods=['GET','POST'])
def predict():
    if request.method == 'GET':
        return render_template('predict.html')
    if request.method == 'POST':
        message = request.form['message']
        blob = TextBlob(message)
        for sentence in blob.sentences:
            detect = sentence.sentiment.polarity
        return render_template('result.html',prediction = detect)

if __name__ == '__main__':
        app.run(debug=True)
```

# Template Inheritance

- allows you to build a base “skeleton” template that contains all the common elements of your site and defines blocks that child templates can override.

Source: https://flask.palletsprojects.com/en/1.1.x/patterns/templateinheritance/



# home.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>About Flask</title>
        <link rel="stylesheet" href="{{ url_for('static', filename='css/template.css') }}">
      </head>
    <body>
      <div class="logo">
        <h1>
          Sentiment Analysis App using TextBlob
        </h1>
      </div>

      <div class="menu-container">
        <div class="container">
          <div class = "links">
            <div class="home"><a href="{{ url_for('home') }}">Home</a></div>
            <div class="about"><a href="{{ url_for('about') }}">About</a></div>
            <div class="predict"><a href="{{ url_for('predict') }}">Predict</a></div>
          </div>
        </div>
    </div>

    {% block content %}
    <p class="para"> This page is my home page. I created a very simple app of sentiment analysis using TextBlob
      <br>I am also using a Natural Language Toolkit named nltk
    {% endblock %}
    </body>
</html>
```

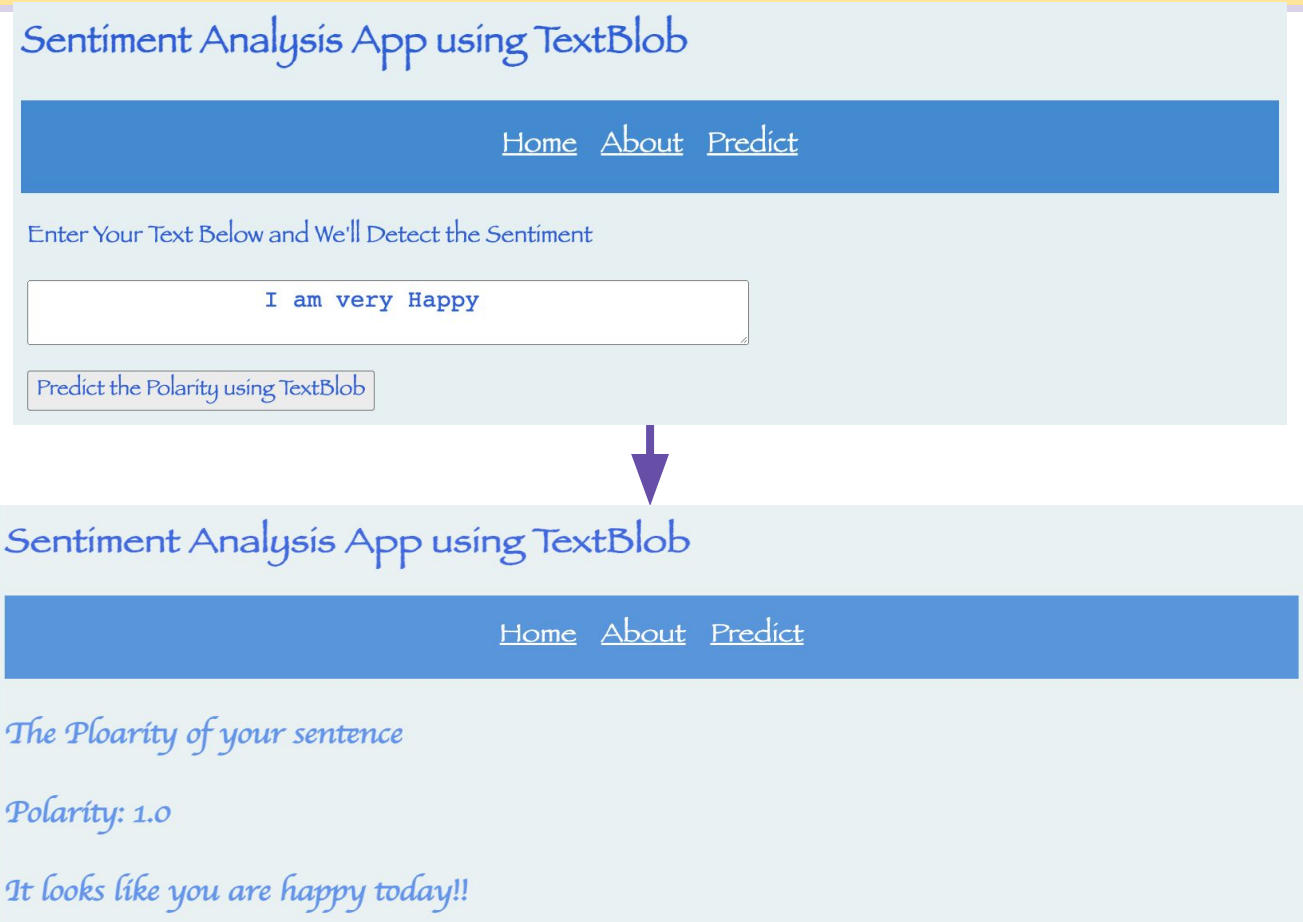
# predict.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Predict Flask</title>
      </head>
    <body>
        {% extends "home.html" %}
        {% block content %}
        <!-- Text Input -->
        <div class="sub-container">
            <form action="{{ url_for('predict') }}" method="POST">
                <p class = "para">Enter Your Text Below and We'll Detect the Sentiment</p>
                <!-- <input type="text" name="comment"/> -->
                <textarea name="message" rows="4" cols="180">
                </textarea>
                <br/>
                <input type="submit" value="Predict the Polarity using TextBlob" padding="16px">
            </form>
        {% endblock %}
    </body>
</html>
```

# result.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
 <head>
   <meta charset="utf-8">
   <title>Predicted Sentiment</title>
 </head>
<body>
    <header>
        {% extends "home.html" %}
        {% block content %}
        <div class="results">
            <h3>The Ploarity of your sentence </h3>
            <h3>Polarity: {{ prediction }}</h3>
                {% if prediction < 0 %}
                <h3>Maybe you need some coffee</h3>
                {% elif prediction == 0 %}
                <h3>You are not in bad mood today!!</h3>
                {% elif prediction > 0 %}
                <h3>It looks like you are happy today!!</h3>
                {% else %}
                <h3>Not sure how you feel today!!</h3>
                {% endif %}
        </div>
    </header>
    {% endblock %}
</body>
</html>
```

# Result


Sentiment Analysis App using TextBlob
Home About Predict
Enter Your Text Below and We'll Detect the Sentiment
I am very Happy
Predict the Polarity using TextBlob
Sentiment Analysis App using TextBlob
Home About Predict
The Ploarity of your sentence
Polarity: 1.0
It looks like you are happy today!!