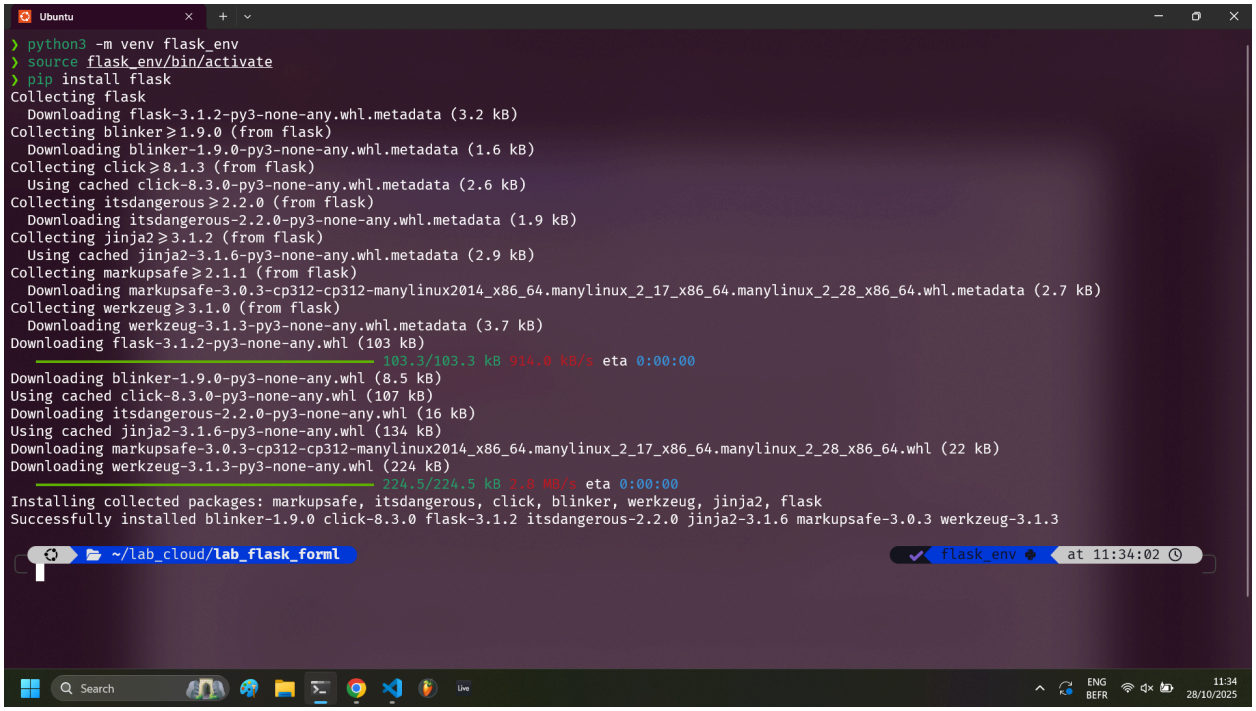
Zakaria CHOUKRI

Lab Flask for ML - Cloud Computing - M311

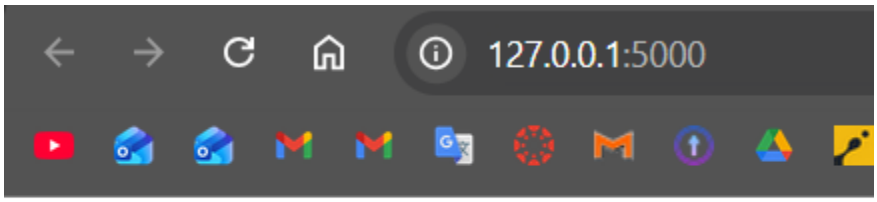
Installing Flask



Example 1:

hello.py

```python
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

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



Hello, World!

Example 2:

Zakaria CHOUKRI

app.py

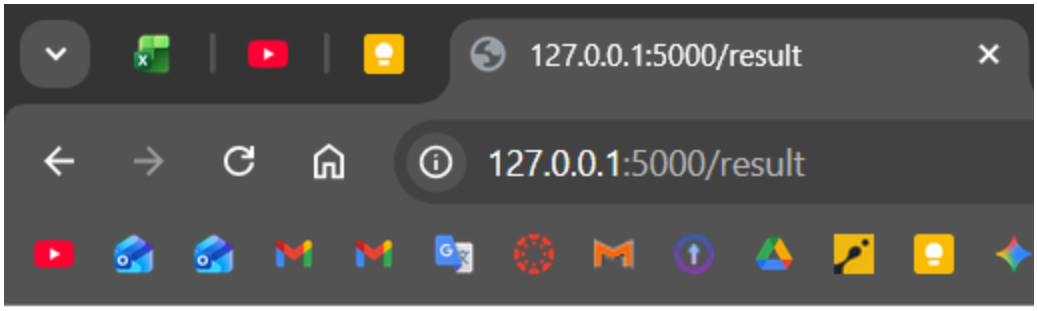
```python
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/result')
def result():
    dict = {'phy':50, 'che':60, 'maths':70}
    return render_template('result.html', result=dict)

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

app.py results.html

templates > results.html > ...

```html
<!doctype html>
<html>
    <body>
        <table border = 1>
            {% for key, value in result.items() %}
            <tr>
                <th> {{ key }} </th>
                <td> {{ value }} </td>
            </tr>
        {% endfor %}
        </table>
    </body>
</html>
```

Zakaria CHOUKRI



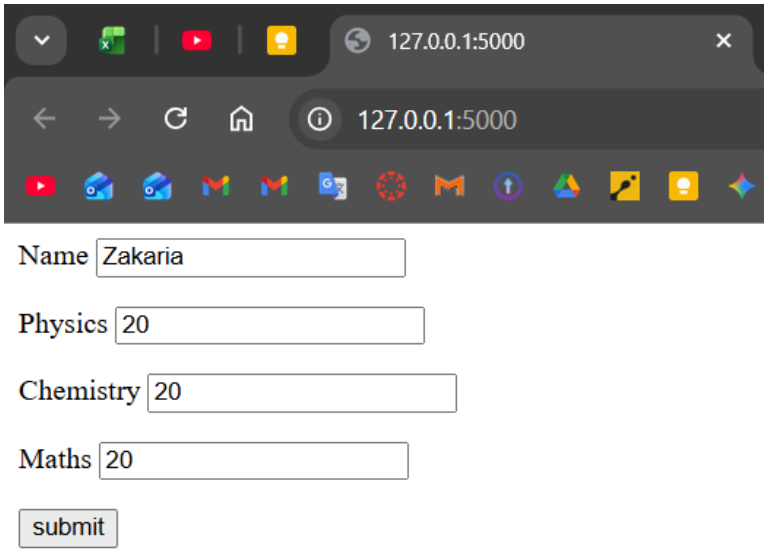
phy	50
che	60
maths	70

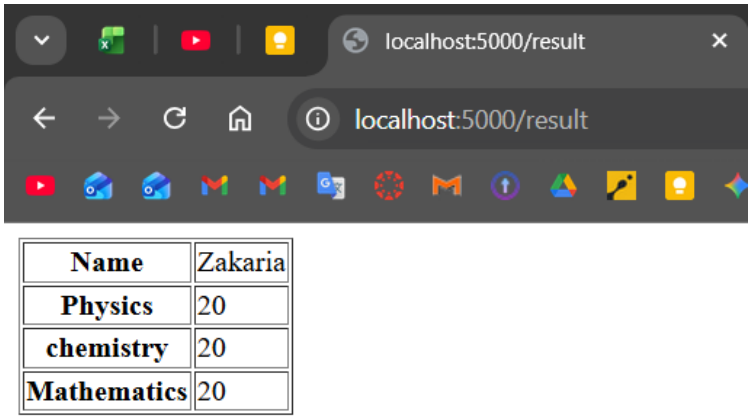
Example 3:

app.py

```python
from flask import Flask, render_template, request

app = Flask(__name__)

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

@app.route('/result', methods=['POST', 'GET'])
def result():
    if request.method == 'POST':
        result = request.form
        return render_template('result.html', result=result)

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

Zakaria CHOUKRI





Heroku part:
I couldn't do the Heroku part because it needs a credit card:
heroku create furniture-prediction-app --region eu

Assignment:

1. Choose a dataset, containing tabular data, of your choice
I chose the Iris dataset

```python
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
import joblib

iris = load_iris()
X = pd.DataFrame(iris.data, columns=iris.feature_names)
y = pd.Series(iris.target, name='target')

print(f"Dataset shape: {X.shape}")
print(f"Missing values: {X.isnull().sum().sum()}")
print(f"Target distribution:\n{y.value_counts()}")
```

2. Prepare your data (treatment of missing values, etc.)

There is nothing to be done

```
Dataset shape: (150, 4)
Missing values: 0
Target distribution:
target
0    50
1    50
2    50
Name: count, dtype: int64
```

4. Use these characteristics to create the prediction page (like in the example above)

5. Compare the performance of multiple ML models to keep the best one at the end.

```
1  joblib.dump(best_model, 'model.pkl')
2  joblib.dump(scaler, 'scaler.pkl')
3  joblib.dump(iris.target_names, 'target_names.pkl')
4  print("Model, scaler, and target names saved!")
```

✓ 0.0s

```
Model, scaler, and target names saved!
```

Zakaria CHOUKRI

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

models = {
    'LogisticRegression': LogisticRegression(max_iter=200, random_state=0),
    'RandomForest': RandomForestClassifier(n_estimators=100, random_state=0),
    'SVM': SVC(kernel='rbf', random_state=0)
}

scores = {}
for name, model in models.items():
    model.fit(X_train_scaled, y_train)
    pred = model.predict(X_test_scaled)
    acc = accuracy_score(y_test, pred)
    scores[name] = acc
    print(f"{name}: {acc:.4f}")

best_model_name = max(scores, key=scores.get)
best_model = models[best_model_name]
print(f"\nBest model: {best_model_name} with accuracy {scores[best_model_name]:.4f}")
```

[2] Python

```
LogisticRegression: 1.0000
RandomForest: 1.0000
SVM: 1.0000

Best model: LogisticRegression with accuracy 1.0000
```

6. Create an application (i.e., a form) that predicts the target (value or class) of a record.

Zakaria CHOUKRI