Hands-on Lab: Building and Deploying a Web App using Flask

Introduction

In this lab, we createa a basic application of mathematical functions and deploy it over a web interface using Flask. The purpose is to connect all the pieces of knowledge gained in the course till now, and see the application development and deployment steps in action.

Estimated time needed: **30** minutes

Objectives

In this assignment you will:

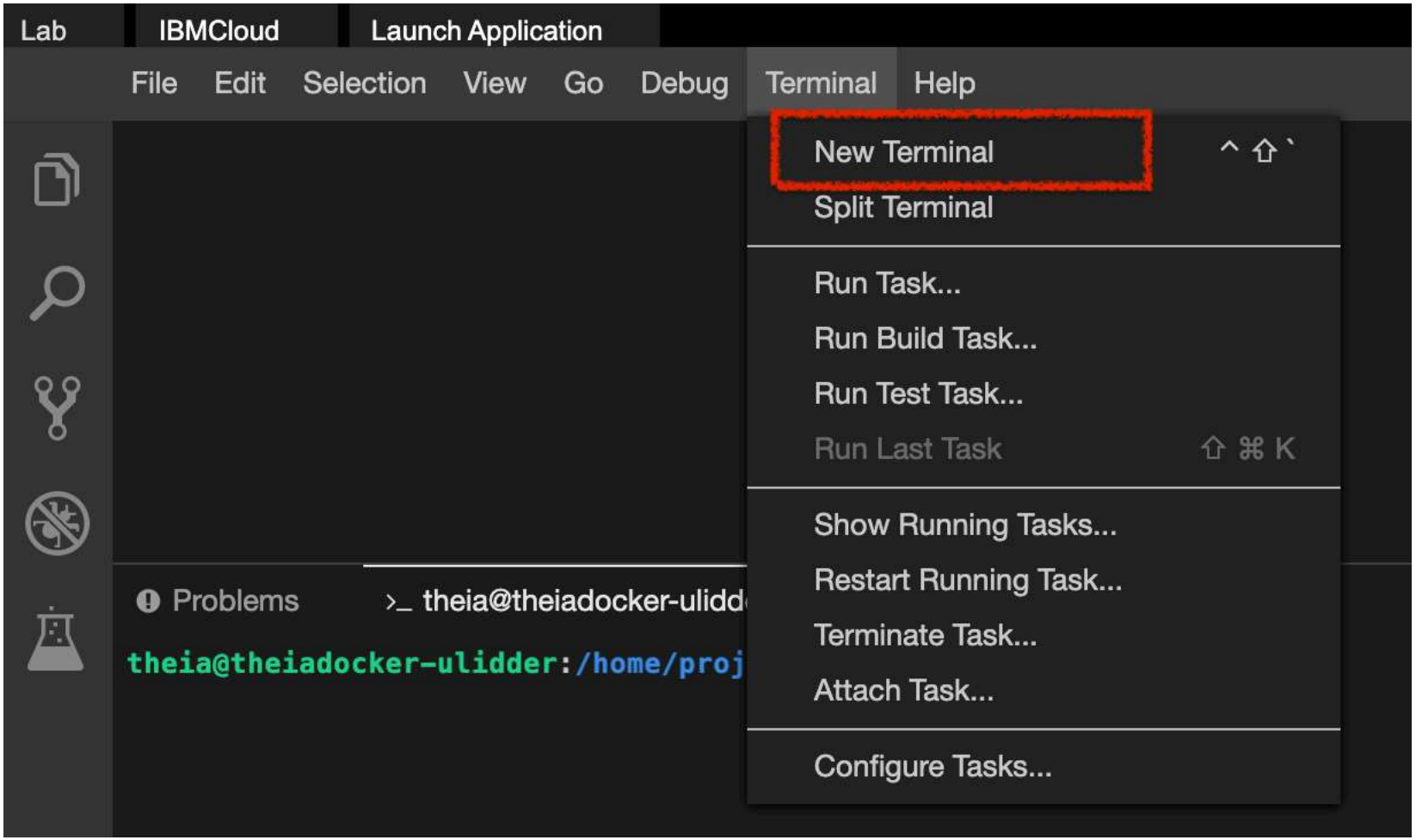
- Task 1: Create the mathematical functions.
- Task 2: Package the functions and test the package.
- Task 3: Web Deployment of the application package using Flask.

Task 1: Write the mathematical functions

In this task, you arre required to write a script that has functions to add, subtract and multiply two values. Let's call this script `mathematics.py`

Follow the steps for this task.

1. Open a terminal window by using the menu in the editor: Terminal > New Terminal.



2. Go to the project home directory.

```
cd /home/project
```

3. Run the following command to Git clone the project directory from the clone URL you had copied in the prework lab.

```
git clone https://github.com/ibm-developer-skills-network/hjbsk-build_deploy_app_flask
```

4. Change to the `practice_project` folder.

```
cd /home/project/hjbsk-build_deploy_app_flask
```

5. Create folder named `Maths` and change to that directory.

```
mkdir Maths
cd Maths
```

6. In the explorer, go to the `Maths` directory and create a new file called `mathematics.py`.

7. Add function **summation** which takes in the `a` and `b` as a number arguments, in `mathematics.py`.

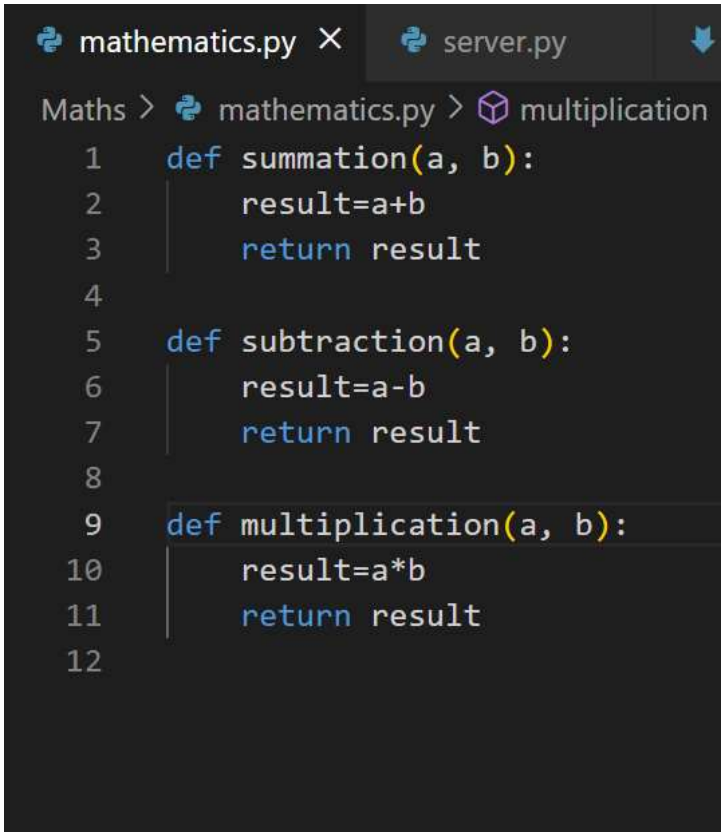
▶ Click here for solution

8. Add function **subtraction** which takes in the `a` and `b` as a number arguments, in `mathematics.py`.

▶ Click here for solution

9. Add function **multiplication** which takes in the `a` and `b` as a number arguments, in `mathematics.py`.

▶ Click here for solution



Task 2: Package the functions

1. Create `__init__.py` file in the directory `Maths`.
2. Import the file `mathematics.py` to the `__init__.py` file.

```
from . import mathematics
```

3. Import the package `Maths` in server.py.

```
from Maths.mathematics import summation, subtraction, multiplication
```

4. In the server.py, for end-point `/`, implement a method that renders the `index.html`.

```
@app.route("/")
def render_index_page():
    return render_template('index.html')
```

5. In the space provided in server.py for the endpoint `/sum` implement a function that uses the appropriate summation function from the package you created in the previous part. The function should retrieve `num1` and `num2` as float inputs from the request parameters. It should then check if the result is a whole number using the `is_integer()` method. If it is, convert the result to an integer before returning it as a string.

6. In the space provided in server.py for the endpoint `/sub` implement a function that uses the appropriate subtraction function from the package you created in the previous part. The function should retrieve `num1` and `num2` as float inputs from the request parameters. It should then check if the result is a whole number using the `is_integer()` method. If it is, convert the result to an integer before returning it as a string.

7. In the space provided in server.py for the endpoint `/mul` implement a function that uses the appropriate multiplication function from the package you created in the previous part. The function should retrieve `num1` and `num2` as float inputs from the request parameters. It should then check if the result is a whole number using the `is_integer()` method. If it is, convert the result to an integer before returning it as a string.

server.py M ×

hjbsk-build_deploy_app_flask > server.py

```python
from flask import Flask, render_template, request
from Maths.mathematics import summation, subtraction, multiplication

app = Flask("Mathematics Problem Solver")

@app.route("/sum")
def sum_route():
    num1 = float(request.args.get('num1'))
    num2 = float(request.args.get('num2'))
    result = summation(num1, num2)
    if result.is_integer():
        result = int(result)
    return str(result)

@app.route("/sub")
def sub_route():
    num1 = float(request.args.get('num1'))
    num2 = float(request.args.get('num2'))
    result = subtraction(num1, num2)
    if result.is_integer():
        result = int(result)
    return str(result)

@app.route("/mul")
def mul_route():
    num1 = float(request.args.get('num1'))
    num2 = float(request.args.get('num2'))
    result = multiplication(num1, num2)
    if result.is_integer():
        result = int(result)
    return str(result)

@app.route("/")
def render_index_page():
    return render_template('index.html')

```

```
37  if __name__ == "__main__":
38      app.run(host="0.0.0.0", port=8080)
```

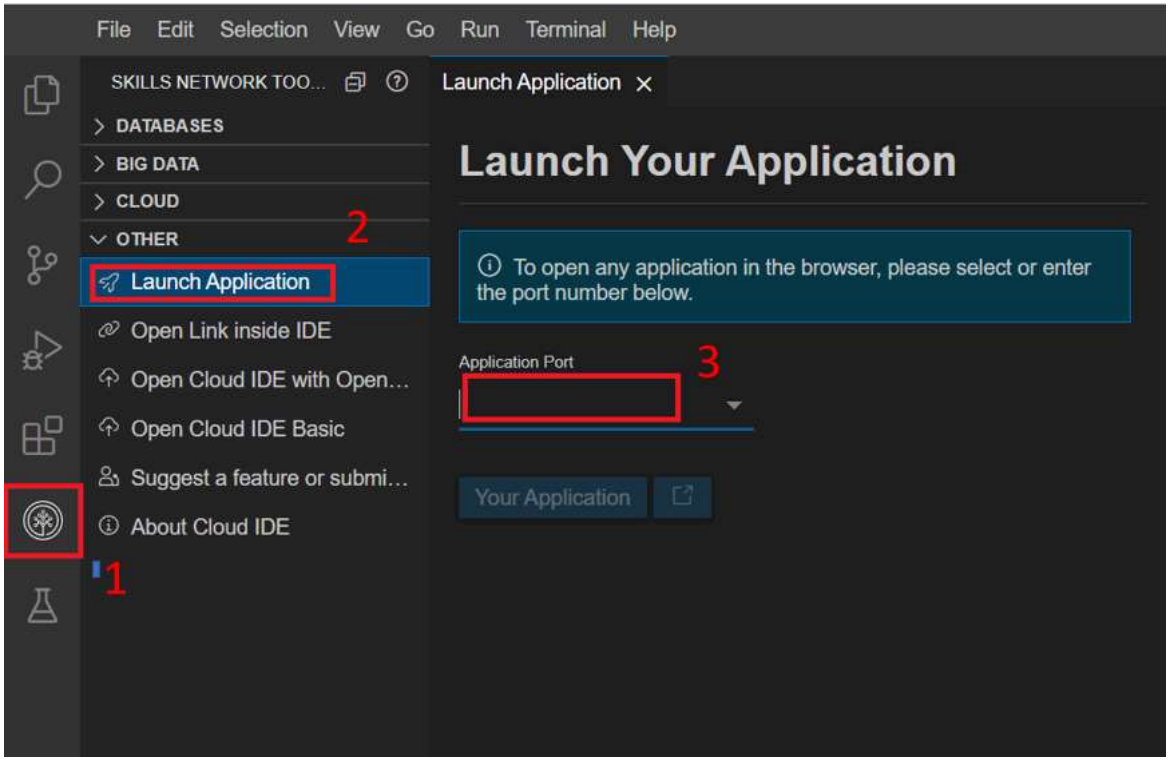
Task 3: Web Deployment of the application package using Flask

1. Change current directory on the terminal to the hjbsk-build_deploy_app_flask directory and run the server from your terminal.

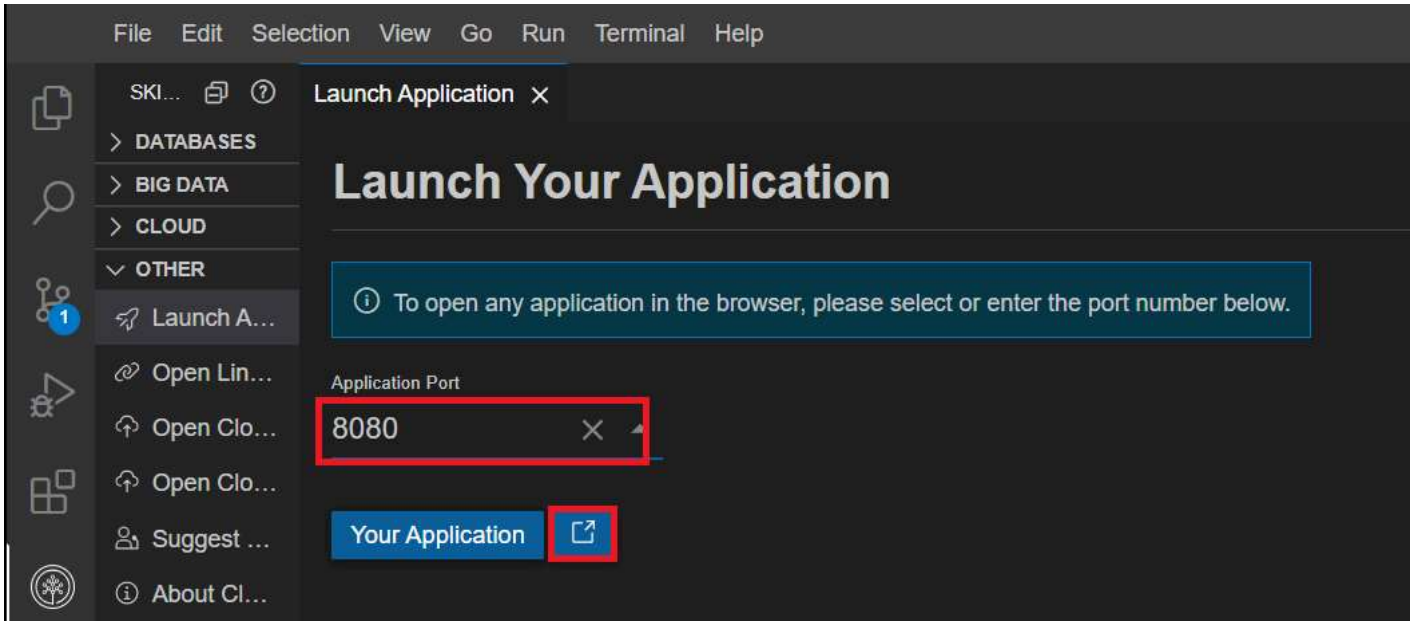
   ```
   cd /home/project/hjbsk-build_deploy_app_flask && python3.11 server.py
   ```

2. You will see that the server starts up in port 8080.

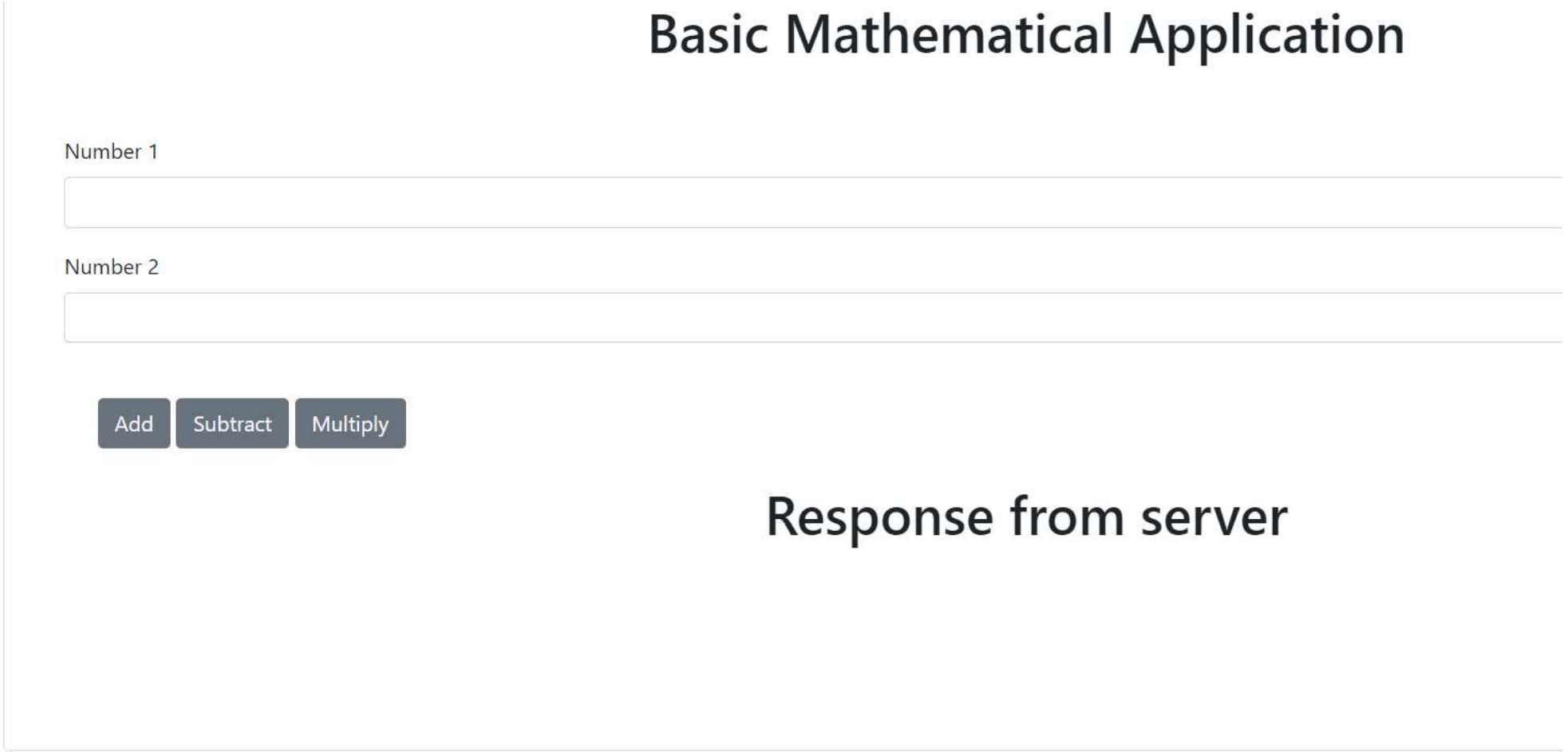
3. Click on the `Skills Network button` on the left, it will open the `Skills Network Toolbox`. Then click the `Other` then `Launch Application`. From there you should be able to enter the port number.



Connect to port `8080`and click `Launch` button.



4. A new browser window opens up with the index page as shown below.



Test your application for the desired outputs. Some examples are shown below.

Basic Mathematical Application

Number 1

10

Number 2

20

Add Subtract Multiply

Response from server

Result: 30

Basic Mathematical Application

Number 1

20

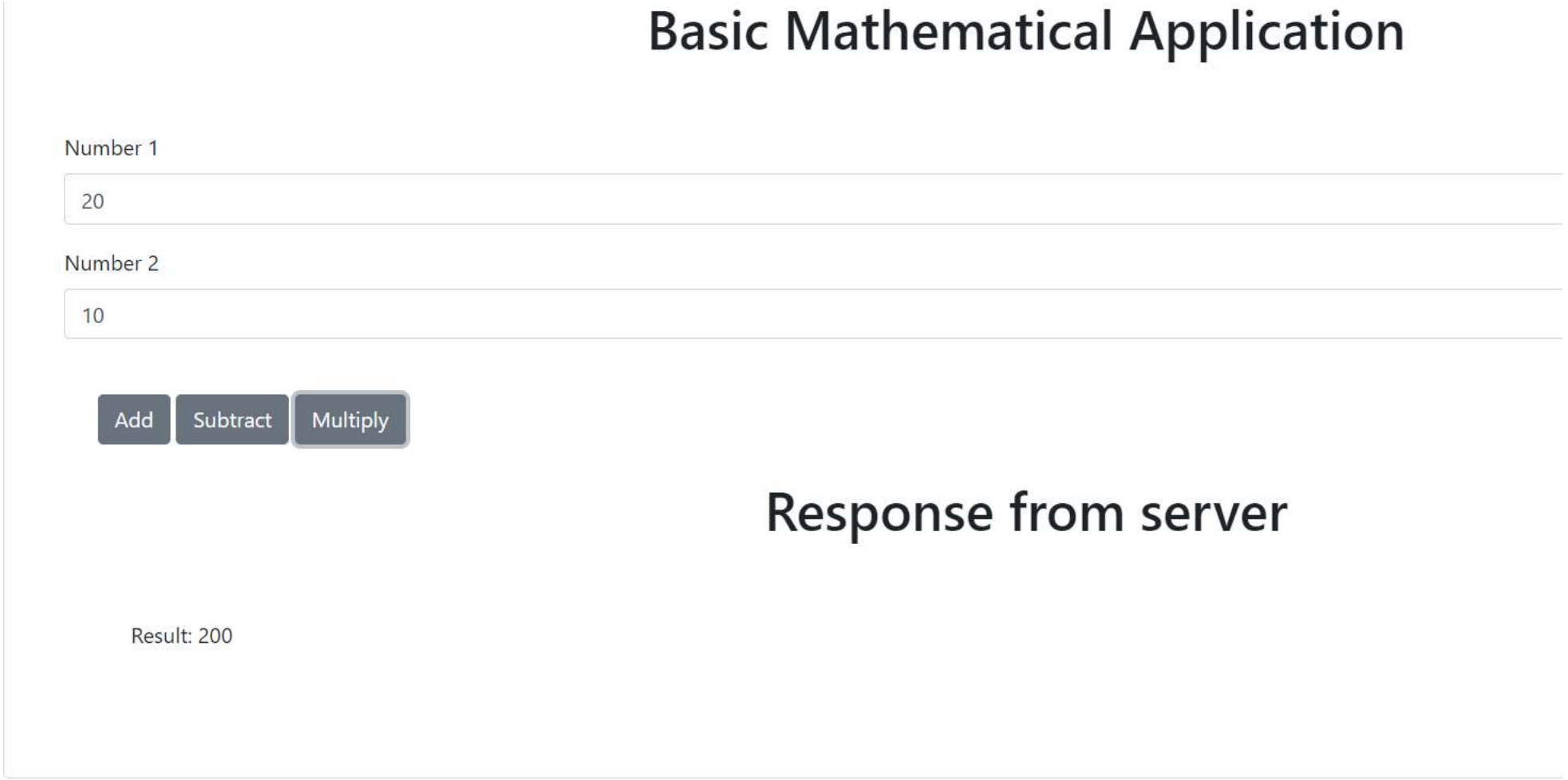
Number 2

10

Add Subtract Multiply

Response from server

Result: 10



(Optional) Practice exercise

Interested learners can try to incorporate error handling capability in this deployed application. For e.g. in case the interface receives non numerical entries for mathematical operations, what should the system response be?

Conclusion

Congratulations! You have completed the tasks for this project.

By the end of this lab, you have:

1. Created functions that perform mathematical operations.
2. Created a package for these functions.
3. Deployed the application that uses this package on localhost using Flask.

Authors

Shivam

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2025-05-22	1.2	Ritika Joshi	Modified the instructions as part of Content Analysis
2023-07-13	1.1	Abhishek Gagneja	Modified the instruction set
2023-06-28	1.0	Shivam	Created initial version of the lab